The Accessible Learning Labs: Supporting Accessibility Education

Heather Moses, Jaden Wedner, Samuel A. Malachowsky and Daniel E. Krutz
Rochester Institute of Technology, Rochester, NY, USA
Email: {hlm8500, jnw7417, samvse, dxkvse}@rit.edu

***Abstract*—Mature and robust software applications should demonstrate several crucial qualities; namely, they should supply the functionality expected by the stakeholder(s), perform efficiently, be secure, and be accessible. However, despite government legislation and growing demonstrated need, accessibility is not a priority for a compelling amount of software applications being developed currently. The primary objective of our *Accessible Learning Labs* is to not only educate participants on how to properly create accessible software, but also effectively illustrate the need to create accessible software. Our experiential labs enable practitioners, instructors, and students to engage with the material using only a browser. This tutorial will benefit a wide spectrum of participants in the software engineering community, ranging from beginning-level students to experienced practitioners who want to ensure that they are properly creating inclusive and accessible software. Complete project material is publicly available on the project website: https://all.rit.edu**

***Index Terms*—Accessibility Education, Computing Education, Computing Accessibility**

I. Instructor Experience

Authors Malachowsky and Krutz have led the creation of the Accessible Learning Labs since the project's inception in 2018.

II. Tutorial Information

1) **Title:** The Accessible Learning Labs: Supporting Accessibility Education
2) **Objective:** Inform participants about creating accessible software and available resources for developing accessible software.
3) **Duration:** 1.5h (but 3h is also ok)

III. Tutorial Outline

A. Lab Structure

The instructional activities found in each lab utilize a structure that supports students in learning about foundational concepts in developing accessible software. Furthermore, the activities allow students to experience the implications of inaccessible software and make repairs based on their experience in order to make the software accessible. An example lab activity is represented in Figure 1, where in Figure 1a the participant experiences the software as a 'non-disabled' user would. Figure 1b demonstrates the simulated repair process that the participant completes and Figure 1c demonstrates how the participant made the software more accessible. The phases in this structure are further described in the steps below.

1) **Participant learns about foundational accessibility issue:** The participant is guided through the appropriate background information regarding the accessibility issue.
2) **Participant experiences accessibility issue:** The participant experiences the accessibility challenge using an emulation feature.
3) **Participant repairs accessibility issue:** Using a simulated, browser-based code editor, the participant repairs the inaccessible software feature.
4) **Participant experiences the benefits/impact of their repair:** After making repairs, the participant completes the activity with their repairs incorporated into the software (Figure 1c). Ideally, with the repairs, the software should be considerably more accessible, even with the accessibility emulation feature applied.

B. Tutorial Session Agenda

The tutorial session will include the following items:

Activity 1: Overview of Labs (10 minutes): An overview of the labs, their objectives, and their components.

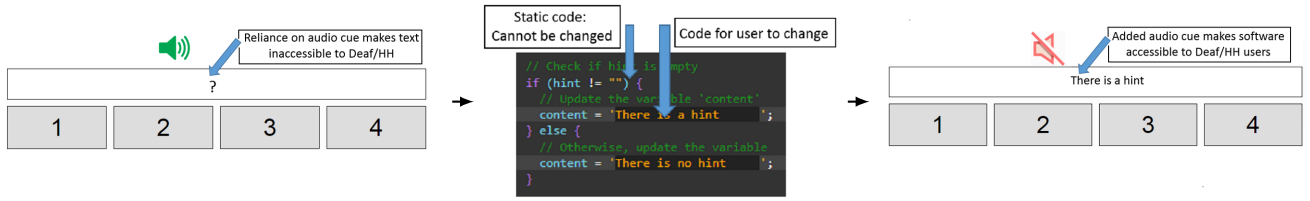
Activity 2: Lab # 1: Hearing (40 minutes):

- **Learning Objective:** Introduce the concept of making software accessible to users who are deaf or hard-of-hearing (DHH) and demonstrate its importance.
- **Difficulty Level (for students):** Introductory
- **Activity:** In this lab, participants play a game where they are tasked with locating a random, hidden item. Points are awarded for finding the item quickly.

Activity 3: Lab # 2: Colorblindness (40 minutes):

- **Learning Objective:** Inform students about the *Distinguishable Content* accessibility guideline.
- **Difficulty Level (for students):** Introductory
- **Activity:** This lab introduces the concept of making software accessible to users that are color blind. In this game, participants are tasked with clicking on specifically colored circles when they appear on the screen.

Activity 4: Lab # 3: Blindness (40 minutes):



(a) Software is inaccessible since user cannot hear the notification and the visual message is not relevant.

(b) Simulated code editor used through browser.

(c) Software is made accessible by participant adding informative visual message.

Fig. 1: Example of participant repairing accessibility problem using simulated code editor.

- **Learning Objective:** Inform participants about the importance of creating software that is accessible to users with blindness or a visual impairment.
- **Difficulty Level (for students):** Medium
- **Activity:** This lab demonstrates the impact that inaccessible software can have on blind users.

Activity 5: Lab Discussion (15 minutes): We will discuss the labs that were presented as well as the future of the project. This discussion will include: A) What new labs should be created and what topics are most important for instructors to be addressing, B) The alterations that need to be made to existing labs, and C) How to make our material more effective and easily adoptable.

Activity 6: Feedback on Accessible Learning Labs (20 minutes): Participants will provide feedback on the created material and offer guidance to presenters on the future direction of the labs. This feedback will be incorporated into the design of future labs and to modify existing labs.

Activity 7: Conclusion/Q&A (15 minutes): Recap of the labs and a question and answer period.

IV. Target Audience

The target audience for this tutorial includes:

- **Practitioners:** Despite the fact that ≈ 15% - 20% of the world's population has a disability, much of the software created today is inaccessible to people with visual, cognitive, hearing, dexterity, and other disabilities. The material presented in the tutorial will support practitioners in creating accessible software by providing resources that inform them on how to develop accessible software.
- **Educators:** Instructors will be educated on how to best include the created resources in their curriculum.

Tutorial participants will not require any special knowledge or skills prior to attending the workshop.

V. Attendee Background

Attendees that may benefit from the tutorial include practitioners, students, instructors and anyone else who will want to learn how to create accessible and inclusive software.

VI. Instructor Biography

Daniel Krutz and Samuel Malachowsky are the PI's of the NSF-funded projects (#1825023 and #2111152) that are devoted to creating the presented labs. Heather Moses is a third year Software Engineering student who has been with the Accessible Learning Labs project for two years. Additionally, Jaden Wedner is a first year Computing Security student who has been with this project for one year.

VII. Tutorial History

A similar tutorial has been presented at other venues. However, due to COVID restrictions, no presentations have been given in-person. Several research-focused papers have been published from this effort [2], [1].

VIII. Technical Requirements

Participants should have laptops with internet connection. The presenter should have access to a projector that material can be shared from. No other special accommodations are required.

IX. Preferences for Tutorial Logistics

The most appropriate duration for the tutorial is 3hr. However, a 1.5h or full-day tutorial could also be easily supported. Other than typical accommodations (*e.g.,* room with presentation screen, table, etc.), there are no preferences.

Acknowledgements

This material is based upon work supported by the United States National Science Foundation under grant #1825023, #2111152 and #2145010.

References

[1] W. Shi, S. Khan, Y. El-Glaly, S. Malachowsky, Q. Yu, and D. E. Krutz. Experiential learning in computing accessibility education. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Companion Proceedings*, ICSE '20, page 250251, New York, NY, USA, 2020. Association for Computing Machinery.

[2] W. Shi, S. Malachowsky, Y. El-Glaly, Q. Yu, and D. E. Krutz. Presenting and evaluating the impact of experiential learning in computing accessibility education. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering Education and Training*, ICSE-SEET '20, page 4960, New York, NY, USA, 2020. Association for Computing Machinery. Distinguished Paper Award.