

Lesson 10: Maximizing System Reliability

45 minutes

Overview

What issues and impacts result from software bugs, and what are strategies to maximize system reliability?

Students explore the issues and impacts resulting from open source software bugs. Students discover strategies for maximizing system reliability.

Standards

Full Course Alignment

CSA Conceptual Framework

- **IOC-1** - While computing innovations are typically designed to achieve a specific purpose, they may have unintended consequences.

Agenda

Warm Up (5 minutes)

Bugs

Activity (35 minutes)

Open Source Bugs

Maximizing Reliability

Wrap Up (5 minutes)

Show What You Know Week

Assessment: Check for Understanding

Objectives

Students will be able to:

- Describe issues and impacts resulting from software bugs
- Identify strategies for maximizing system reliability

Preparation

- Print copies of the Open Source Bugs handouts (two copies of one version per pair of students)
- Print copies of the Unit 7 Study Guide (one for each student)
- Check the **Teacher's Lounge** for verified teachers on the CSA Forum to find additional strategies or resources shared by fellow teachers

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the students

- **Open Source Bugs (Version A)** - Handout
- **Open Source Bugs (Version B)** - Handout
- **Open Source Bugs (Version C)** - Handout
- **The Rise of Open Source Software** - Video
- **U7L10 Extra Practice** - Handout
- **Unit 7 Study Guide** - Resource

Vocabulary

- **System Reliability** - A system's ability to function without failure

Teaching Guide

Warm Up (5 minutes)

Bugs

Remarks

Throughout this year, you've used your software engineer skills to debug your code. Let's think about some examples of bugs in the real world.

 **Discuss:** Use the Retrieve-Pair-Share strategy to discuss the prompt.

- *When have you experienced a bug in an application or program that you use that affected system reliability - a system's ability to function without failure?*


Discussion Goal: Students share examples of bugs they have experienced, such as a website being down or apps that they use crashing.

Activity (35 minutes)


Open Source Bugs (20 minutes)

Remarks

It's impossible to completely prevent all bugs in a program. However, as software engineers, there are actions we can take to quickly identify and resolve bugs before they cause significant issues for users.

 **Do This:** Review the lesson objectives.

Group: Place students in pairs.

 **Distribute:** Give each pair two copies of the same version of the Open Source Bugs handout.

 **Do This:** Review the instructions for the Open Source activity.

Remarks


Now that you are an expert on your Open Source Bug, let's learn about the other bugs.

 **Discuss:** Click through the animated slide to display the prompts.

- *What was the bug?*
- *How widespread was it?*
- *What is being done to maximize system reliability in the future?*


Discussion Goal: Students share the bug they reviewed and the impacts resulting from the bug. Students identify the actions that are being taken to prevent these issues in the future and may also suggest additional actions that can be taken to prevent these issues.

Maximizing Reliability (15 minutes)

 **Display:** Show the video – *The Rise of Open Source Software*.

Remarks

Software engineers often use existing open source software to develop their own programs. They also may choose to develop new programs and make them open source for other people and software engineers to use in their programs.

 **Discuss:** *What can you do as software engineers to maximize the reliability of open source software?*


Discussion Goal: Students suggest ways to maximize the reliability of open source software, such as contributing code to existing projects or identifying and submitting bugs found in existing projects.


Wrap Up (5 minutes)

Show What You Know Week


Remarks

The next five lessons are an opportunity to Show What You Know! We complete our unit project, then spend a day practicing answering AP-style free-response questions and wrapping up the unit with a multiple-choice assessment. You've learned so much in this unit, and now you're ready to show what you know!

 **Distribute:** Give each student a copy of the Unit 7 Study Guide.

 **Do This:** Introduce the Show What You Know week.

 **Do This:** Explain the goals of the project workdays.

 **Do This:** Have students update their Project Planning Board and Project Backlog with any tasks they completed, changed, or added.

 **Do This:** Review the concepts covered in this lesson.

 **Display:** Key Vocabulary

Assessment: Check for Understanding

Check For Understanding Question(s) and solutions can be found in each lesson on Code Studio. These questions can be used for an exit ticket.



Check for Understanding



This curriculum is available under a
Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes **contact us**.