

Software Development Evaluation and Grading Strategies

Mark Choman
Professor - Computer Information Systems

Bob Dushok
Director of Infrastructure/Network Operations

Luzerne County Community College - Nanticoke, PA
<http://www.luzerne.edu>



HI-TEC Conference
07/21-22/2021



Today's Session

- This session (pre-recorded) is targeted to Information Systems professors desiring to expand their evaluation and grading strategies of student projects. Examples are drawn from mobile, web, desktop GUI, and console -based projects. We want to share what we feel are effective techniques to consider when evaluating programming assignments as well as building useful assessment feedback.

Starting Template

- Because students often receive the same project, having a .docx with common feedback is helpful.
- Tag likely to change areas in yellow highlight.
- Incorporate "next steps" feedback that will...
 - keep students aware of what is next in the course's sequence.
 - remind students of active communication.
 - cite IT workforce-related references, program advisory board feedback, college or department events applicable to the IT student, etc..
- Live demonstration.

Starting Template (cont.)

- Use an image capture tool when seeing feedback in action is better or complements what written feedback provides.
- Apply a macro to remove highlighting and format images with borders.
- Live demonstration.

Unzipping Submissions And Pinning

- Your LMS may allow you to download all submissions as a single .zip file. If so, consider a script to unzip submissions on a per-folder basis.
- Pin a shortcut to the root folder containing the submissions.
- Live demonstration.

In-app Workflows

- Reset the IDE if needed.
- Take advantage of screen splitting to see non-adjacent content.
- Adjust the font settings to make code or results easier to understand.
- Capture contextual feedback.
- Use multimedia feedback when applicable.
- Live demonstration.

Starting Template (revisited)

- Build it as you're grading.
- Use to track number of instances of specific problems.

Consistent Environment

- Regardless of the language in use, select an appropriate IDE.
- Require students to all use the same version of the product you've selected.
- Although we're teaching a specific language, we are also supporting the IDE we're using.

No Beta or Preview Products!

- Some students feel it's interesting to test unreleased products and experience new features.
- This might work well in an higher level programming classes.
- This is not helpful in introductory programming classes.
- Many students don't understand these products aren't finished and may include significant bugs.

Submission of Screenshots

- Useful for early project in introductory classes.
- Enables instructors to determine each student has a working environment.
- May be submitted in addition to code or instead of code for simple projects.

Thanks for Attending!

- Mark Choman
mchoman@luzerne.edu
@MarkChoman
- Bob Dushok
bdushok@luzerne.edu