STEM Teaching with Active Learning

A Model Extracted from Software Engineering Classes

Kesina Baral, Paul Ammann, Jeff Offutt

September, 2020

ACTIVE LEARNING

PREPARATION

ENGAGEMENT AND LEARNING

ASSESSMENT

PREPARATION

OFFLINE READING

FLIPPED VIDEOS

READING QUIZZES

SWE 437 Course Schedule
URL: https://cs.gmu.edu/~pammann/437-sched.html

Software Testing
Fall 2018

Meeting	Date	Topic	Reading	Assignment	Quizzes	In Class	ShowMes
1	Aug 30	Class Overview; Why Test?	AO 1			In-Class #1 In-Class #2	
2	Sep 6	Faults, Errors, and Failures	more AO 1	Assignment 1 Due	Guide 1	In-Class #3 In-Class #4	Faults, Errors, Failures
3	Sep 13	Test Automation: JUnit	AO 3.1-3-3,		Guide 2	In-Class #5 In-Class #6	JUnit Theories
4	Sep 20	Traditional Maintenance	Offutt's essay: Overview of Maintenance (pptx) Singer's article: Practices of Software Maintenance	Assignment 2 Due	Guide 3	In-Class #7 In-Class #8	
5	Sep 27	Test Doubles; Legacy Code; Putting Testing First	AO 12.2 K4.3, 4.4, K4.6 AO 4 Videos available		Guide 4	In-Class #9 In-Class #10	
6	Oct 4	Agile Methods; Test Driven Development; Test Patterns	K1 K2 K4.1, 4.2, 4.5;	Assignment 3 Due	Guide 5	In-Class #11 In-Class #12	
7	Oct 11	Coverage Criteria; Input Space Partition Testing	AO 5 AO 6.1-6.3 Videos available		Guide 6	In-Class #13 In-Class #14	

Using this model in (other) STEM classes

- Software engineering is STEM

    - Particularly this class

- Our exercises look/feel like math exercises
  - Example: "hashCode() is consistent with equals()"
  - Extract from API documentation
  - Formalize constraint
    - If x, y equal, they must share hash
    - But not the converse!
  - Analyze artifacts for compliance
    - "Proofs" and counterexamples (test cases)
- We hope the model can work in your class!