Teaching Software Engineering Skills in CS1.5: Incorporating Real-world Practices and Tools

Sarah Heckman and Jason King, North Carolina State University
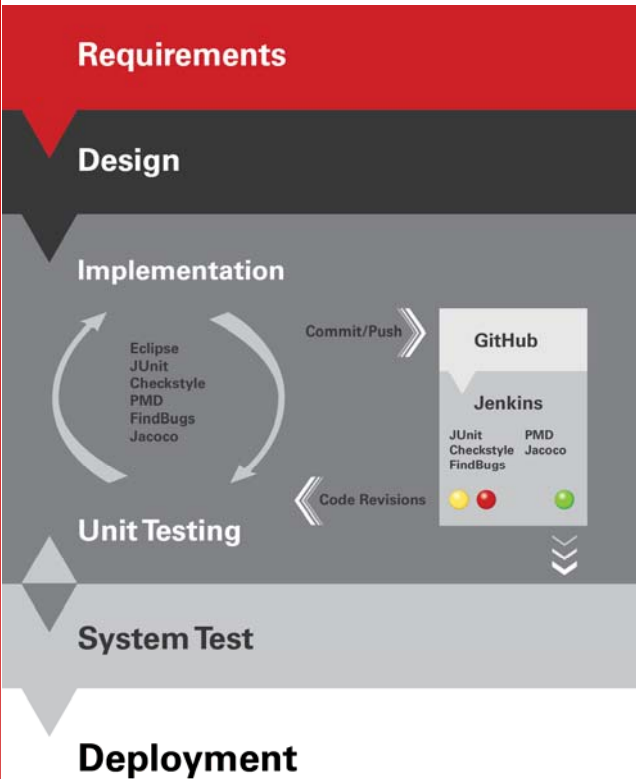sarah_heckman@ncsu.edu and jtking@ncsu.edu

Goal

To improve student learning, engagement in the course and profession, and retention through the use of software engineering practices and tools that introduce students to the software engineering profession.
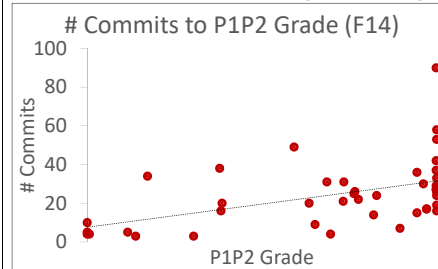
System

Requirements
Design
Implementation

Eclipse
JUnit
Checkstyle
PMD
FindBugs
Jacoco

Commit/Push → GitHub

Jenkins

JUnit Checkstyle FindBugs · PMD Jacoco

Code Revisions

Unit Testing
System Test
Deployment

Fall 2014 Software Engineering Skills

Commits to P1P2 Grade (F14)

Grade	Avg. Commits
F	12.9
D	34.5
C	14.7
B	25
A	30.5

Students who commit more frequently tend to score better on the assignment. *Encourage more frequent commits.*

Days Before Deadline to P1P2 Grade
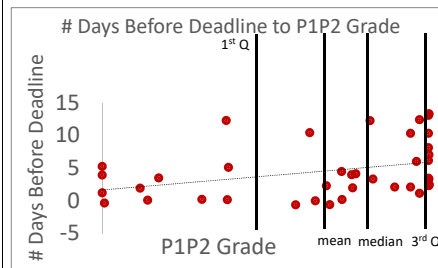
Grade	Avg. Days Before DL
F	3.0
D	4.9
C	0.6
B	3.0
A	6.4

A students start much earlier than other students. *Require early commits for a grade, add milestones, or indications of progress.*


# of Tests at Each Commit
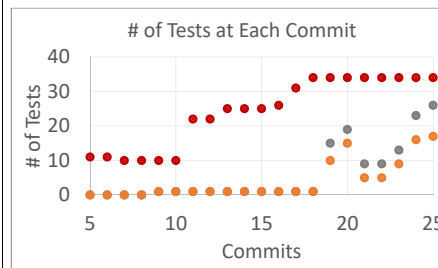
 # Passing Teaching Staff Tests
All TS tests passing BEFORE student really writes tests. *Better checks of student tests: methods should contain asserts.*

 # Passing Student Tests
Student doesn't even pass their own tests! But passes the teaching staff tests.
• *Require passing student tests before TS tests are run.*
• *Note that passing own tests are easy points. While still enforcing quality.*

Fall 2015 Software Engineering Skills

Commits to P1P2 Grade (F15)
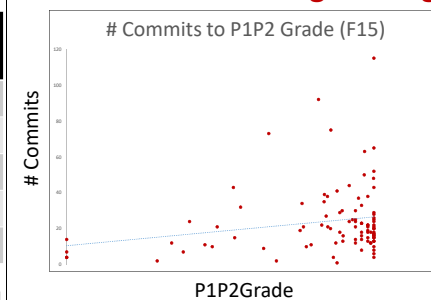
Grade	Avg. Commits
F	14.7
D	28
C	19
B	30.4
A	25.3

We continue to see, that students who commit more frequently tend to score better on the assignment. *Encourage more frequent commits.*

In-Class Labs

Fall 2014	Fall 2015 – Spring 2016
Collections	Debugger
Array List (2)	*Finite State Machines*
Linked List (2)	Recursive Lists
Iterators	GUIs
Inspections	

Spring 2016 Updates

• Process Points – due 1 week before project deadline
  • Teaching staff tests compile against the skeleton
  • Fully commented (no CheckStyle notifications)
  • Meaningful commit messages
• Scaffolding script for Project 3

Future Work

• Identify additional patterns with tests, coverage, and static analysis.
• Identify or create adoption systems that support SE best patterns.
• Incorporate the adoption systems into appropriate classes and measure change.