

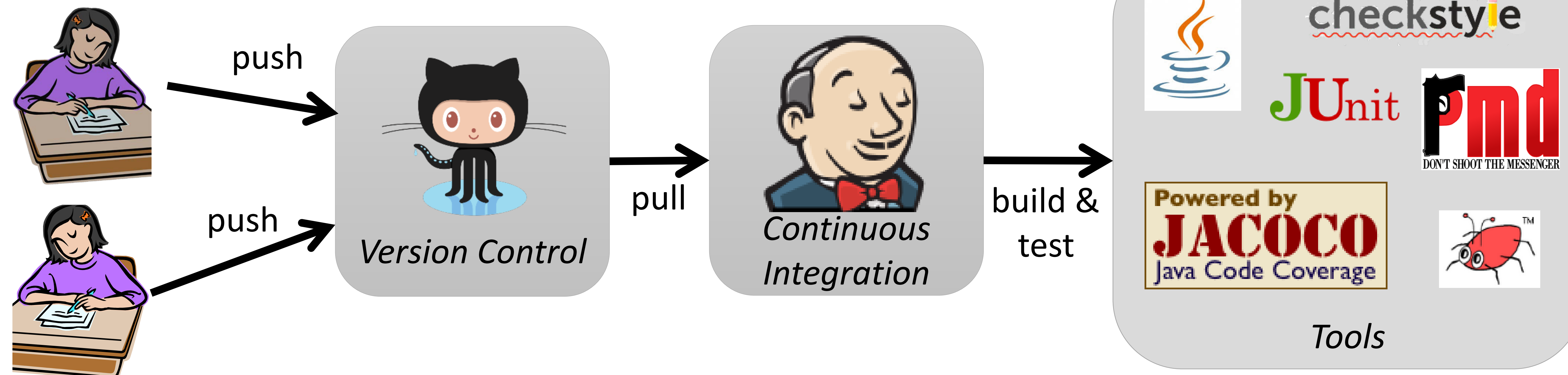
A Continuous Integration Framework for Promoting Software Engineering Best Practices

Sarah Heckman, North Carolina State University
sarah_heckman@ncsu.edu

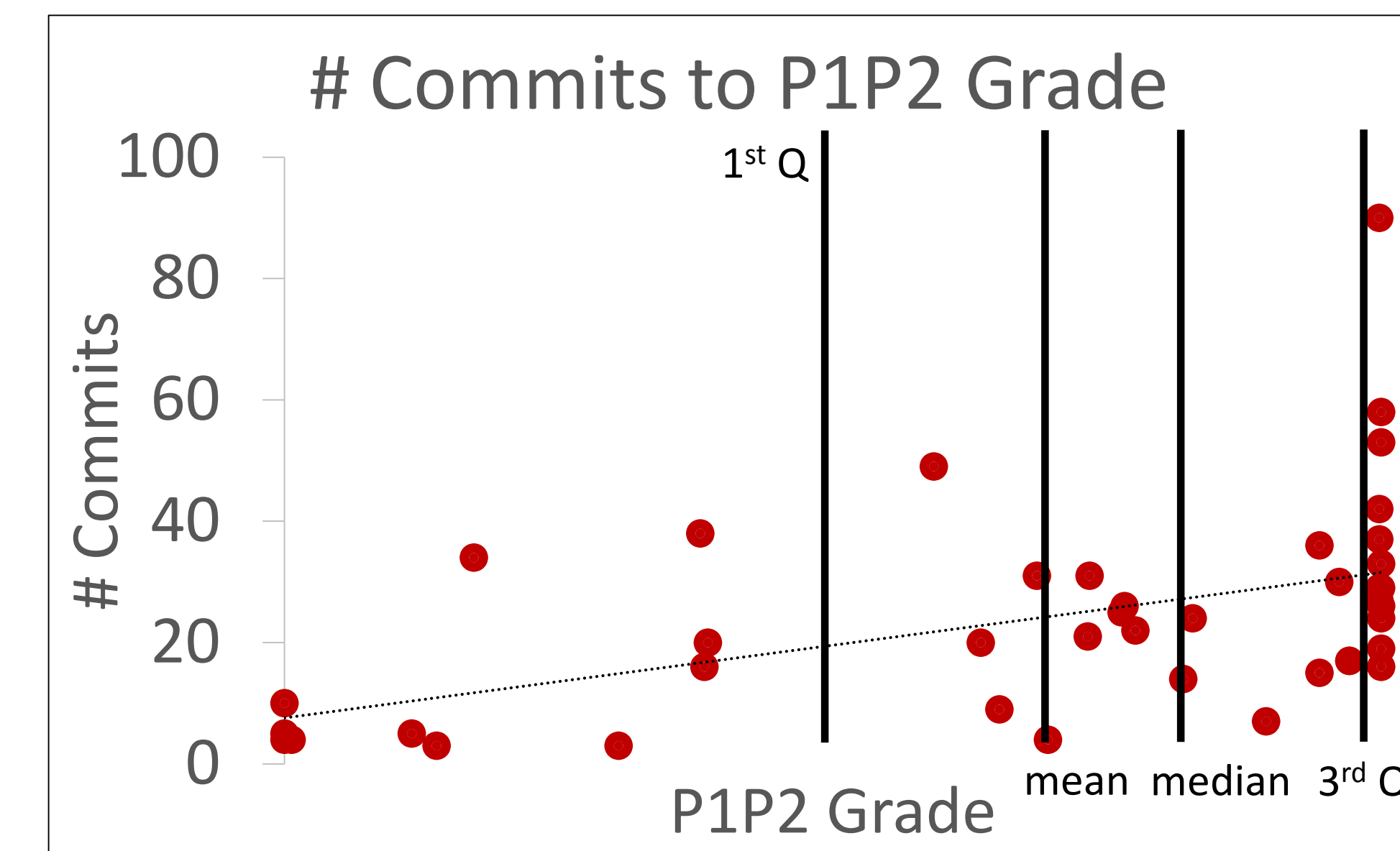
Overview

Goal: to combine and modify *industrial-strength continuous integration tools* and *best practice adoption systems* to encourage and enforce student *adoption* and *use* of software engineering best practices during in-class labs and individual assignments.

System

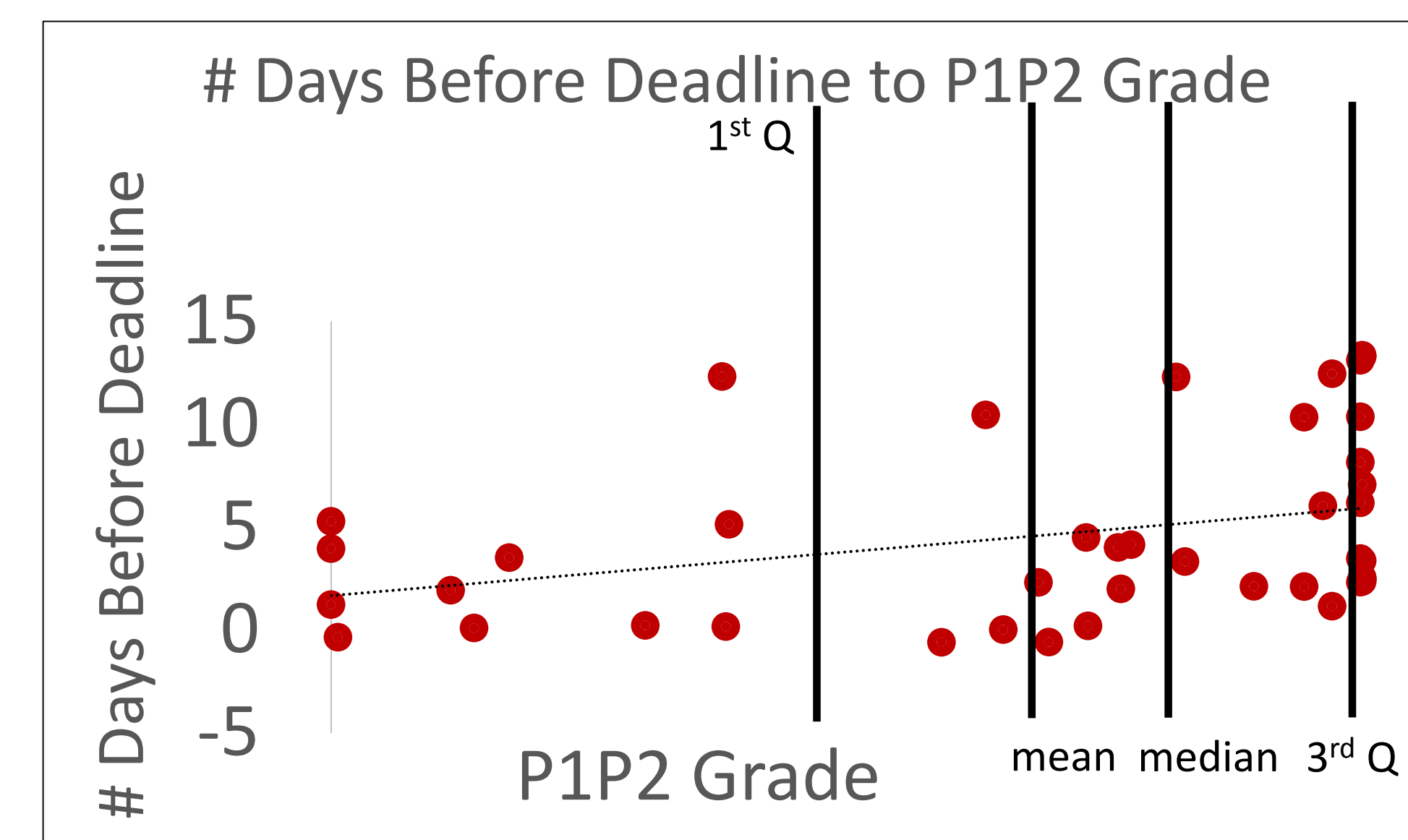


Observations of Software Engineering Skills – CSC216



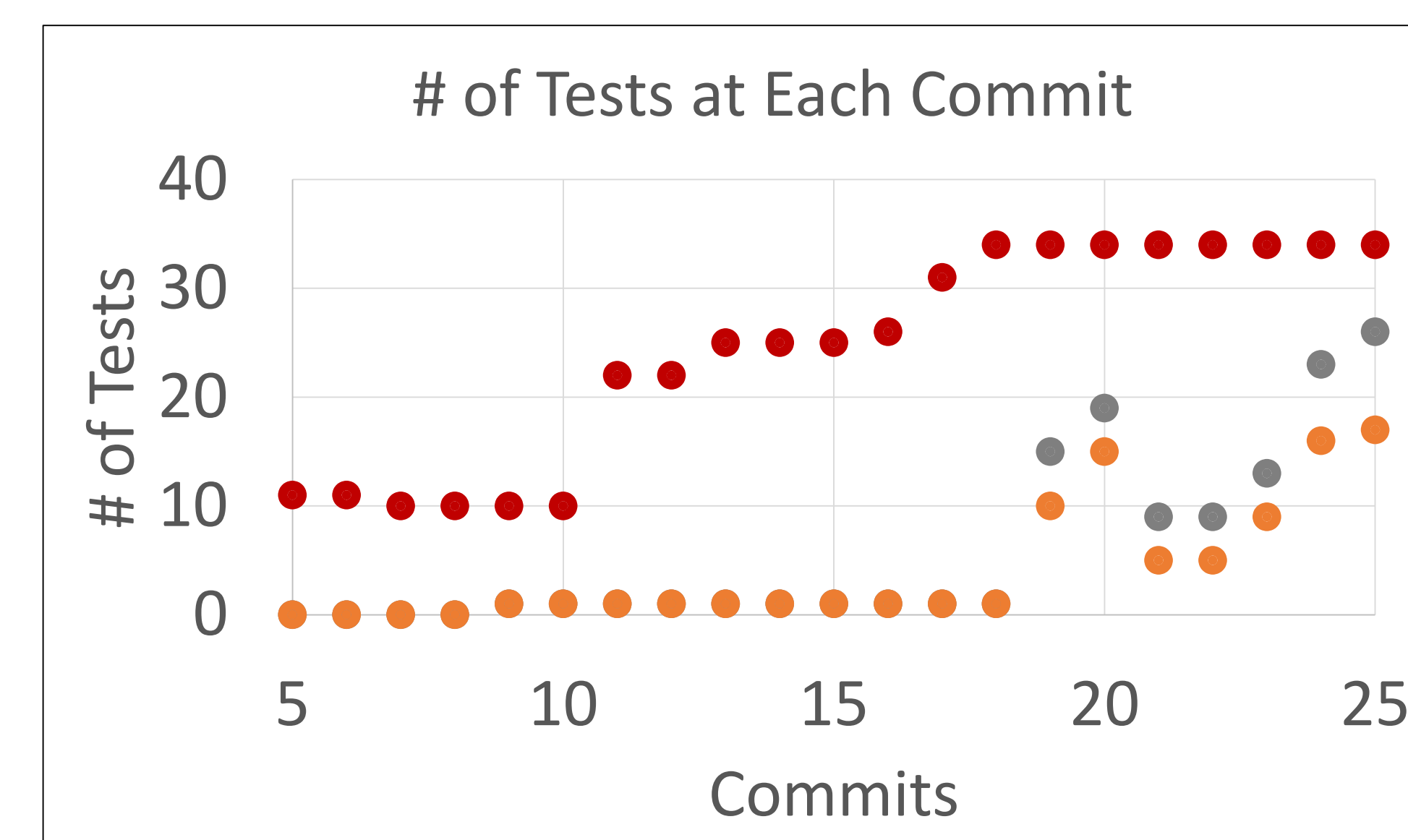
| Grade | Avg. Commits |
|-------|--------------|
| F | 12.9 |
| D | 34.5 |
| C | 14.7 |
| B | 25 |
| A | 30.5 |

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



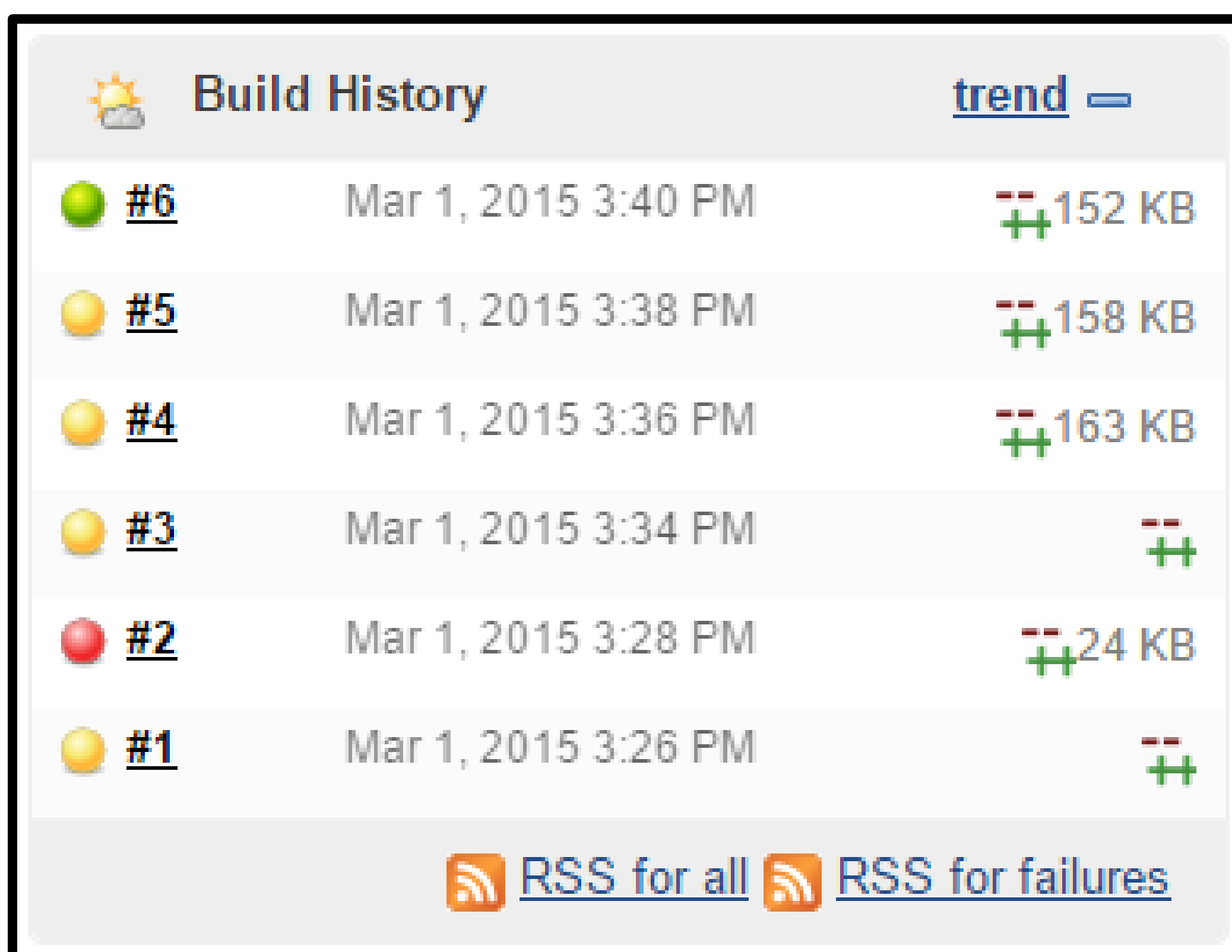
| Grade | Avg. Days Before DL |
|-------|---------------------|
| F | 3.0 |
| D | 4.9 |
| C | 0.6 |
| B | 3.0 |
| A | 6.4 |

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



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.*



```

15:28:17 build_student:
15:28:17 [echo] Building Student Project...
15:28:17 [javac] /var/lib/jenkins/Summer2014/jobs/TT_sesmith5/workspace/CashRegister_MASTER/build
to build.sysclasspath=last; set to false for repeatable builds
15:28:17 [javac] Compiling 7 source files to /var/lib/jenkins/Summer2014/jobs/TT_sesmith5/works
15:28:19 [javac] /var/lib/jenkins/Summer2014/jobs/TT_sesmith5/workspace/CashRegister_MASTER/src
error: cannot find symbol
15:28:19 [javac]         currency[0] = new Currency(PENNY_VALUE, PENNY_NAME, initialCount);
15:28:19 [javac]         ^
  
```

Build #3 (Mar 1, 2015 3:34:06 PM)

- Changes: 1. Fixed compiler error, but failing tests and static analysis (commit: 1174302eb2ee13c...)
- Started by an SCM change
- Revision: 1174302eb2ee13cd7f9591ac20f99f...
 - refs/remotes/origin/master
- Checkstyle: 2 warnings from one analysis.
 - 2 new warnings
 - Plug-in Result: -2 warnings exceeded
- FindBugs: 7 warnings from one analysis.
 - 7 new warnings
 - Plug-in Result: -7 warnings exceeded the threshold of 0 by 7 (Reference build: #1)
- PMD: 1 warning from one analysis.
 - 1 new warning
 - 3 fixed warnings
 - Plug-in Result: -1 warning exceeded the threshold of 0 by 1 (Reference build: #1)
- Test Result (20 failures / +20)

| name | instruction | branch | complexity | line | method | class |
|-------|--------------|-------------|-------------|------------|------------|-----------|
| all | 95% | 76% | 76% | 91% | 100% | 100% |
| class | M: 21 C: 388 | M: 11 C: 33 | M: 10 C: 31 | M: 0 C: 85 | M: 0 C: 19 | M: 0 C: 3 |

Build #6 (Mar 1, 2015 3:40:11 PM)

- Changes: 1. Should have everything fixed! (commit: 55c754eb3e6021c3e37f30d2c1b8c...)
- Started by an SCM change
- Revision: 55c754eb3e6021c3e37f30d2c1b8c...
 - refs/remotes/origin/master
- Checkstyle: 0 warnings from one analysis.
 - No warnings since build 5.
 - New zero warnings highscore: no warnings since yesterday!
- FindBugs: 0 warnings from one analysis.
 - No warnings since build 4.
 - New zero warnings highscore: no warnings since yesterday!
- PMD: 0 warnings from one analysis.
 - 1 fixed warning
 - Plug-in Result: -
 - New highscore: 0
- Test Result (no failures)

| name | instruction | branch | complexity | line | method | class |
|-------|--------------|-------------|-------------|------------|------------|-----------|
| all | 95% | 76% | 76% | 91% | 100% | 100% |
| class | M: 21 C: 388 | M: 11 C: 33 | M: 10 C: 31 | M: 0 C: 85 | M: 0 C: 19 | M: 0 C: 3 |

Acknowledgements

Jason King, Michael Winters, Barry Peddycord, III,
Chris Theisen
Virtual Computing Lab (VCL)
CSC IT Staff
Other CSC216/CSC230/CSC326 instructors

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.