**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**

---

**Unit Testing**

---

**Deployment**

---

**Acknowledgements**
This work is funded by a Google CS Engagement Award, a DELTA Course Redesign Grant. This material is based upon work supported by the National Science Foundation under Grant No 1525173. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

---

**Fall 2014 Software Engineering Skills**

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

**Fall 2015 Software Engineering Skills**

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

---

**In-Class Labs**

**Collections**
- Debugger
- Finite State Machines
- Recursive Lists
- GUIs

**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.