CS 520/620
Advanced Software Engineering
Spring 2016

February 18, 2016
Logistics

- **No class** on **02/23** -- finalize project plan.
- **Paper selection** due **02/25**.
- **Paper presentations** between **03/01** and **03/31**.
Today

A brief overview
● Program analysis (static vs. dynamic).
● Software testing and debugging.

Research paper presentations
● Overview of paper suggestions.
● How to choose a paper?
● How to present a paper?
Program analysis

Why program analysis?
Program analysis

Why program analysis?
● Increase confidence in correctness.
● Understand/prove behavior and properties.
● ...

Static analysis vs. Dynamic analysis
Program analysis

Why program analysis?
- Increase confidence in correctness.
- Understand/prove behavior and properties.
- ...

Static analysis
- Build an abstraction of runtime states.
- Reason about the program without execution.

Dynamic analysis
- Execute the program with some inputs.
- Observe behavior.
Static analysis: examples

Manual

- Code/design review.
- Informal walkthrough.
- Formal inspection.

Anything that could be improved in this code?

```java
double avg(double[] nums) {
    int n = nums.length;
    double sum = 0;

    int i = 0;
    while (i < n)
        sum = sum + nums[i];
    i = i + 1;

    double avg = sum / n;
    return avg;
}
```
Manual static analysis

Pros
● Can be applied at any step in the development process.
● Improves confidence and communication.

Cons
● Time-consuming.
● Mostly informal.
● Not repeatable.
Static analysis: examples

Automated static analysis
- Rule/pattern-based analysis (PMD, Findbugs, etc.).

```java
double avg(double[] nums) {
    int n = nums.length;
    double sum = 0;
    int i = 0;
    while (i < n) {
        sum = sum + nums[i];
        i = i + 1;
    }
    double avg = sum / n;
    return avg;
}
```
Static analysis: examples

Automated static analysis

- Rule/pattern-based analysis (PMD, Findbugs, etc.).

double avg(double[] nums) {
    int n = nums.length;
    double sum = 0;

    int i = 0;
    while (i < n) {
        sum = sum + nums[i];
        i = i + 1;
    }

    double avg = sum / n;
    return avg;
}
Static analysis: examples

Automated static analysis
  ● Control-flow analysis
  ● Data-flow analysis

What is the CFG for this avg function?

double avg(double[] nums) {
    int n = nums.length;
    double sum = 0;

    int i = 0;
    while (i<n) {
        sum = sum + nums[i];
        i = i + 1;
    }
    double avg = sum / n;

    return avg;
}
Program analysis

Static analysis
● Build an abstraction of runtime states.
● Reason about the program without execution.

Dynamic analysis
● Execute the program with some inputs.
● Observe behavior.

Should we use static or dynamic analysis?
Research paper presentations

Logistics:
● 15 papers to be presented in March.
● Papers are grouped by high-level topics.
● Each paper has a presentation date.
● **Paper assignments** through Moodle *(02/23, 9am)*.

Presentations:
● 2 or 3 students per paper.
● 1 presentation per student (group).
● 20 minutes per paper.
Research paper presentations

How to choose a paper:
- Take a brief look at each paper.
- Read the abstract (and conclusions).
- Rank the papers.
Research paper presentations

How to present a paper:
● Give a brief overview of the area/domain and problem.
● Outline research questions and contributions.
● Describe the approach/methodology.
● Highlight key results (evaluation).
● Outline and discuss conclusions.
● Discuss limitations and directions for future work.
Research paper presentations

How to present a paper:

- Give a brief overview of the area/domain and problem.
- Outline research questions and contributions.
- Describe the approach/methodology.
- Highlight key results (evaluation).
- Outline and discuss conclusions.
- Discuss limitations and directions for future work.

Outline questions or controversial statements for classroom discussion.
Selected papers

Empirical software engineering -- 03/01
- Views on internal and external validity in empirical software engineering. ICSE 2015.

Distributed systems -- 03/01
- “To Share or not to Share” in Client-Side Encrypted Clouds. 2014.
Selected papers cont.

**Structural testing and code coverage -- 03/03**

- Coverage Is Not Strongly Correlated with Test Suite Effectiveness. ICSE 2014.
- Assertions Are Strongly Correlated with Test Suite Effectiveness. FSE 2015.
Selected papers cont.

**Mutation testing -- 03/08**

- Mutations: How close are they to real faults? ISSRE 2014.
Selected papers cont.

**Test generation -- 03/22**

- Mutation-driven Generation of Unit Tests and Oracles. TSE 2012.
- Test Input Generation with Java PathFinder. ISSTA 2004
Selected papers cont.

**Fault localization -- 03/29**