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?
<table>
<thead>
<tr>
<th>Static analysis vs. Dynamic analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Static analysis</strong></td>
</tr>
<tr>
<td>● Build an abstraction of runtime states.</td>
</tr>
<tr>
<td>● Reason about the program without execution.</td>
</tr>
<tr>
<td><strong>Dynamic analysis</strong></td>
</tr>
<tr>
<td>● Execute the program with some inputs.</td>
</tr>
<tr>
<td>● Observe behavior.</td>
</tr>
</tbody>
</table>

### Static analysis: examples

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

#### 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.
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;
}
```

What is the CFG for this avg function?

Static analysis: examples

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

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.

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

- Are Automated Debugging Techniques Actually Helping Programmers?
  ISSTA 2011.
  ASE 2015.
- Ask the Mutants: Mutating Faulty Programs for Fault Localization.
  JSTVR 2014.