Software Reliability

Xiangyu Zhang

Pronounced as Shang You Zang

The Goals of CS590F

- ❑ Get to know this area.
  - What are the topics?
  - How people solve problems? Hopefully some of them will be inspiring.
- ❑ Use program analysis to solve some interesting problems.
  - Hands-on experience on designing and implementing program analysis.
- ❑ Paper
  - Not necessarily a conference paper.

Why Reliable Software is Important?

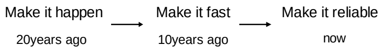
- ❑ Software bugs cost the U.S. economy about $59.5 billion each year (0.6% of the GDP) [NIST 02].
- ❑ The worldwide economic loss caused by all forms of overt attacks is $226 billion. [CRS 03].
- ❑ Software errors can cause human death.
- ❑ Stories
  - The Role of Software in Spacecraft Accidents (http://sunnyday.mit.edu/papers/jsr.pdf)

Why ? – FSE'06 Experience (Nov. 7-9, 2006)

- ❑ Data mining - 5 papers.
  - Mining api, bug patterns, associate failure inducing changes with failures.
- ❑ Debugging - 4 papers.
- ❑ Testing - 3 papers
  - testing web services, SQL programs, distributed applications;
- ❑ Software verification – 3 papers.
- ❑ Security – 2 papers.
- ❑ Program Analysis – 3 papers.
- ❑ …

Why? –The Relevant Areas

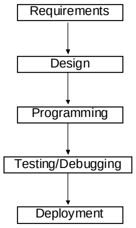
- ❑ Software Engineering
  - covers all topics in software reliability
  - conferences (FSE, ICSE, ASE, ISSTA, FASE, ICSM…)
- ❑ Programming Languages
  - language design, language support, program analysis
  - conferences (PLDI, POPL, OOPSLA,…)
- ❑ Computer Architecture
  - Architecture support for reliability
  - Conferences (ISCA, MICRO, ASPLOS,…)
- ❑ OS, Security.



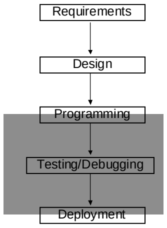
What is Software Reliability

- ❑ IEEE 610.12-1990 defines reliability as "The ability of a system or component to perform its required functions under stated conditions for a specified period of time."
- ❑ IEEE 982.1-1988 defines Software Reliability Management as "The process of optimizing the reliability of software through a program that emphasizes software error prevention, fault detection and removal, and the use of measurements to maximize reliability in light of project constraints such as resources, schedule and performance.“
- ❑ Using these definitions, software reliability is comprised of three activities:
  - Error prevention.
  - Fault detection and removal.
  - Measurements to maximize reliability, specifically measures that support the first two activities.

Software Reliability >> Debugging



The Scope of CS 590F



- ❑ The essence of the this course:
  - Using program analysis (both static and dynamic) to detect and fix program defects.
    - ❖ Given a program, with or without test inputs, can you ...
- ❑ Therefore it covers
  - Debugging
  - Security
  - Testing
  - Program analysis for fun
- ❑ Does not cover:
  - Requirements, design, metrics, …

Course Organization

- ❑ Instructor will lecture the first four weeks.
  - (week 1) introduction, program representations.
  - (week 2) program analysis.
  - (week 3) tools and implementation.
  - (week 4) testing and program slicing.
- ❑ Students will be presenting papers from week 5 to week 14.
- ❑ Final project presentation will be scheduled in the last week.

Course Requirements

- ❑ Two paper presentations (40%, 20% each)
  - 75 minutes each, may contain one or two papers in each presentation;
  - send me your preferences of papers and time slots by Jan. 22.
  - I prefer both presentations in the same topic or in two closely related topics;
  - send me your discussion part of slides the night before you present, send me your presentation slides after the talk.
- ❑ Presentation format
  - Text book concept review in case some fellow students do not have the background (up to 15 mins, NOT REQUIRED)
  - The technical paper, besides the main technical content, clearly identify the following if possible:
    - ❖ the tool/system used;
    - ❖ the benchmark used;
      - ✓ is it standard compared to similar papers?
      - ✓ is it publicly available?
  - Discussion (up to 15 mins)
    - ❖ What is most inspiring about this paper (what your fellow students should learn from the paper)?
    - ❖ What are the problems of the presented work?
    - ❖ Do you have any new ideas to share ?
      - ✓ Can the same problem be solved differently?
      - ✓ Can you use the same technique to solve a different problem?

Course Requirements

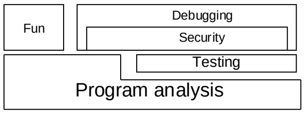
- ❑ Term project (50%)
  - in groups of 1 or 2.
    - ❖ Form your group and decide your project by Feb. 15.
  - one proposal presentation (5%).
    - ❖ 15 mins.
  - one final presentation (10%).
    - ❖ The length of time to be decided.
  - one final report (35%).
    - ❖ Due on Apr. 29 midnight.
    - ❖ 10-18 pages, single column.
    - ❖ Suggested format:
      - ✓ the problem you are solving;
      - ✓ a motivation example;
      - ✓ your solution;
      - ✓ empirical results;
      - ✓ related work.

Course Requirements

- ❑ Attendance and class participation (10%)
  - You are HIGHLY RECOMMENDED to read the papers beforehand.
  - An active role in discussion will earn extra credits.

Topics

Overview



Fun

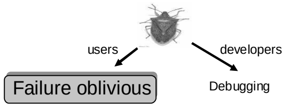
Debugging

Security

Testing

Program analysis

Debugging



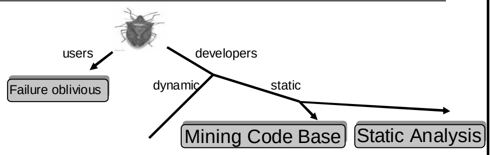
users

developers

Failure oblivious

Debugging

Debugging



users

developers

Failure oblivious

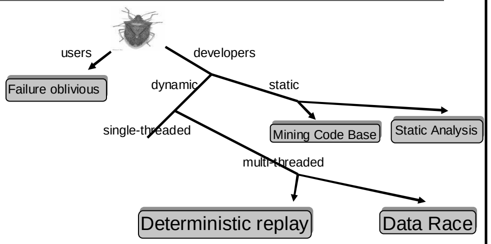
dynamic

static

Mining Code Base

Static Analysis

Debugging



users

developers

Failure oblivious

dynamic

static

single-threaded

Mining Code Base

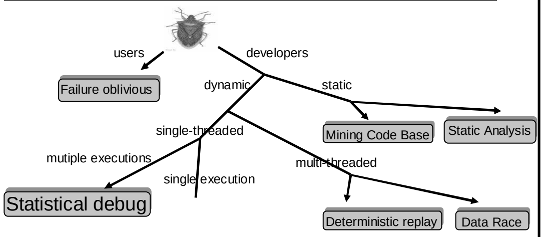
Static Analysis

multi-threaded

Deterministic replay

Data Race

Debugging



users

developers

Failure oblivious

dynamic

static

single-threaded

Mining Code Base

Static Analysis

mutiple executions

multi-threaded

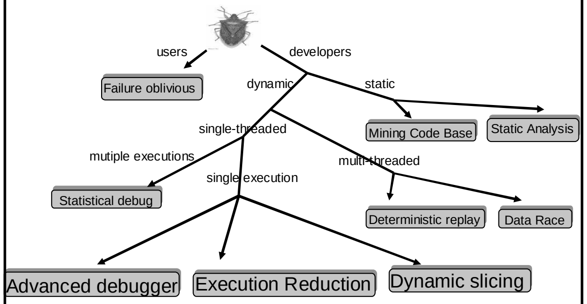
single execution

Statistical debug

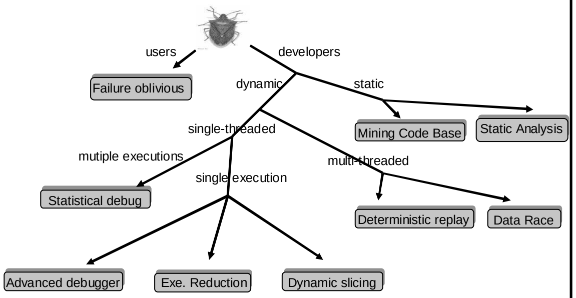
Deterministic replay

Data Race

Debugging



Debugging



Not Covered: model checking, performance bugs, …

Security

- ❑ Covered: security issues that are related to programs or program executions.
  - Information flow;
  - Static vulnerability detection;
    - ❖ Security holes in many cases are essentially specific type of software defects.
  - Secure execution (dynamic vulnerability detection);
  - SQL injection attacks.
- ❑ Not Covered:
  - Cryptography;
  - Protocol design, access control;

Testing

- ❑ Test generation
  - Test generation by symbolic execution;
  - Test generation by concrete execution.
- ❑ Interesting Directions
  - Testing + verification
  - Testing + security
  - (Haven’t seen) Testing + debugging

Program Analysis for Fun

- ❑ Matching program executions.
- ❑ Treating program executions as database.
- ❑ Data lineage.
- ❑ Handle the bug that caused the mars orbiter crash.

Wrap Up

- ❑ This course is about
  - ANALYZING PROGRAMS AND PROGRAM EXECUTIONS to expose defects.
  - All topics are not covered.
- ❑ Next lecture – program representations.
- ❑ Make it 75 mins (twice a week)?