Seminar Software Quality

Preliminary meeting

We will start at 3:32 pm

Fabian Leinen (Orga)

Jakob Rott
Roland Würsching
Dr. Benjamin Poppinga
Dr. Markus Schnappinger
Kohei Dozono
Michael Wolf
Maximilian Jungwirth
Software Quality

Quality of Tests

Quality of Code

Quality in Operations
Participating

Apply via matching tool

Application with us: Online form
- Letter of motivation
- Study program and semester
- Optional: CV + grade report
- Your 3+ favorite topics

July 16th, 23:59

http://go.tum.de/070420
Schedule

- Application
- Matching
- Matching results and topic assignment
- Kickoff
- Individual phase
- Block seminar

Events:
- Literature research
- How to thesis?
- Effective presentations

Timeline:
- Until 16th of July
- October
- February
Grading

Thesis

- Seminar paper: max. 15 pages
- Content: Theory + application of the topic (results, experiences, problems and limitations)
- Initial submission
- Final submission: 1 week after presentation

Presentation

- 20 min + 10 min discussion
- Mandatory dry run (1 week before seminar)

50/50
Questions about the organization?
Detecting Flaky Test Failures in Continuous Integration Using Failure Symptoms

**Goal:** Distinguish between *regression failures* and *flaky failures* using failure symptoms

**Data:** Billions of test executions from open-source projects

**Approach:** Brittle rule based and advanced NLP techniques

Flaky Tests

- Waste developer time
- Bugs might go to production
CodeLM as Test Scheduler

"Summarize Python: def inc_value(x):..."
"Generate Python: increment value"
"Defect: if x=0: x += 1"
"Refine: if x=0: x += 1"
"Translate Python to C: if x==0: x += 1"

"increment value"
"def inc_value(x):..."
"true"
"if x == 0: x += 1"
"if (x==0) {x += 1;}

Does break

Priorization

Selection
Speed up Testing by Identifying Code Refactorings

Which types of refactorings can we safely exclude?

```java
public class A {
    public int m(int x, int y){ return x - y; }
    + public int m(int x, int y){ return x / y; }
}
class B {
    public int m(int x, int y){ return x + y; }
}
class C {
    public int m(int x, int y) throws Exception {
        Object a = Class.forName("A").newInstance();
        Method m = a.getClass().getMethod("m", ...);
        return m.invoke(a, x, y);
    }
}

public class Base64Test {
    - private static final String[] BASE64_IMPOSSIBLE_CASES = {
        + static final String[] BASE64_IMPOSSIBLE_CASES = {
            "ZE="", "ZmC="", "Zm9vYE==", "Zm9vYmC="", "AB",)
    }
```
Non-Code Dependencies in Software Testing

How common are non-code deps in open-source projects?
Can we trace changes in resource files back to a test case?
Scenario-Based Testing of Cyber-Physical Systems I

Free Driving

Following

Lane Change

Obstacle

Cut In

Traffic Jam

Lane Violation

Leave Lane

Ego car  Other car
Scenario-Based Testing of Cyber-Physical Systems II

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting positions $s_{0,c_i}$ [m]</td>
<td>0</td>
<td>300</td>
</tr>
<tr>
<td>Starting times $t_{start,c_i}$ [s]</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Target velocities $v_i$ [km/h]</td>
<td>80</td>
<td>140</td>
</tr>
<tr>
<td>Lane change time $t_{lc,c_1}$ [s]</td>
<td>6</td>
<td>13</td>
</tr>
</tbody>
</table>
Scenario-Based Testing of Cyber-Physical Systems II
Scenario-Based Testing of Cyber-Physical Systems

Summary

- Scenario based testing is used to test hard to test dynamic CPS such as cars
- In compare test cases within a parameterized scenario, and to optimize for the most critical test case, we need performance metrics such as such as fitness functions
- Simple fitness functions are easy to design, for example minimizing the distance of the System-under-Test to other objects optimizes for a collision.
- This topic will focus on improving on these simple fitness functions.
Test Gap Analysis

"Have all changes since the last release been tested?"
unchanged
new & not tested
changed & not tested
changed & executed in test
Clone Detection:

"Where can identical (copied) parts be found in source code?"
// Utilities for arrays of elements

public String showElements(ModelElement[] elements, String nomsg) {
    boolean found = false;
    StringBuffer res = new StringBuffer();
    if (elements != null) {
        Index.getInstance().setCurrentRenderer(
            FlatReferenceRenderer.getInstance());
        for (int i = 0; i < elements.length; i++) {
            ModelElement el = elements[i];
            res.append(showElementLink(el)).append(HTML.LINE_BREAK);
            found = true;
        }
        Index.getInstance().setCurrentRenderer();
    }
    if (!found && nomsg != null && nomsg.length() > 0) {
        res.append(HTML.ITALICS(nomsg));
    }
    return res.toString();
}
public String showElements(ModelElement[] elements, String nomsg) {
    boolean found = false;
    StringBuffer res = new StringBuffer();
    if (elements != null) {
        Index.getInstance().setCurrentRenderer(
            FlatReferenceRenderer.getInstance());
        for (int i = 0; i < elements.length; i++) {
            ModelElement el = elements[i];
            res.append(showElementLink(el)).append(HTML.LINE_BREAK);
            found = true;
        }
        Index.getInstance().resetCurrentRenderer();
    }
    if ((found & amp; nomsg != null & amp; nomsg.length() > 0) {
        res.append(HTML.italics(nomsg));
    }
    return res.toString();
}
Using AI to generate test cases: Reality or Illusion?

- Are code snippets sufficient to create test cases?
- Which type of bugs can be detected?
- Which benefits do we gain when providing the requirements, too?

And how „good“ is that approach actually?
Impact of Cybersecurity Incidents

35 million Xfinity customers have data leaked in breach tied to Citrix Bleed bug

Xfinity confirmed more than 35 million of its customers were affected by a data breach linked to the Citrix Bleed vulnerability. The company, which is part of Comcast Corporation, notified customers Monday that usernames and hashed passwords were stolen in a mid-October cyberattack.

Citrix announced the discovery of a critical vulnerability tracked as CVE-2023-4956 on Oct. 10, and released a patch the same day. The bug, nicknamed Citrix Bleed, is a buffer overflow

British Airways faces record £183m fine for data breach

British Airways is facing a record fine of £183m for last year's breach of its security systems.

The airline, owned by IAG, says it is "surprised and disappointed" by the penalty from the Information Commissioner's Office (ICO).
Are LLMs the Future of Code Security Analysis?

- Can LLM detect and classify software vulnerability (C/C++) \textit{without fine-tuning}?
  - Prompt engineering?
  - Retrieval-Augmented Generation?
  - etc...

- How to evaluate the effectiveness?
  - One of the prominent LLMs vs One of the Static Application Security Testing (SAST) tools.
Dependency Communication Assistant

Background
- Reliability is essential for Google Cloud, if something breaks we need to mitigate quickly
- Multiple teams involved in incident mitigation, good communication is critical for success
- Most friction can be observed in communicating with upstream/downstream dependencies

Idea: Use LLMs to help in communication
- Help downstream customers to assess impact of production incident on services early ("Am I affected?")
- Help communicators summarizing most critical insights ("We are certain that it is affecting cell XY only")
- Show communicators potential gaps in comms ("Stakeholders might want to know if this is affecting their cell")
- Many more use cases ...

Approach: Hands-on with what is publicly available
- GCP deployment configurations - from Github or mocked
- Gemini, ChatGPT, Gemma, etc. - any LLM you find interesting
- Incident communication logs – from Google, other cloud providers
Participating

1. Apply via matching tool

2. Application with us: Online form
   - Letter of motivation
   - Study program and semester
   - Optional: CV + grade report
   - Your 3+ favorite topics

http://go.tum.de/070420

July 16th, 23:59