Seminar Software Quality

Preliminary meeting

We will start at 1:02 pm

Fabian Leinen (Orga)

Jakob Rott
Lena Gregor
Roland Würsching
Martin Gruber
Fabian Leinen
Prof. Andrea Stocco
Dr. Markus Schnappinger
Dr. Andrea Nutsi
Software Quality

- Quality of User Interfaces
- Quality of Code, Data and Models
- Quality of Tests
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

July 19th

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

- Literature research
- How to thesis?
- Effective presentations

Application

Matching results and topic assignment

Matching

Kickoff

Individual phase

Block seminar

Until 19th of July

October

February
Grading

**Thesis**
- Seminar paper: max. 15 pages (text)
- 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?
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().resetCurrentRenderer();
    }
    if (found & nomsg != null & nomsg.length() > 0) {
        res.append(HTML.italics(nomsg));
    }
    return res.toString();
}
// 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().resetCurrentRenderer();
    }
    if ((found && nomsg != null && nomsg.length() > 0) {
        res.append(HTML.italics(nomsg));
    }
    return res.toString();
}
Test Gap Analysis

"Have all changes since the last release been tested?"
Deep Dive into Practical Contract-Based-Testing

Service A -> Service B

Consumer

Provider

Contract

1. Consumer unit tests its behavior against provider mock
2. Requests in contract captured into a contract between systems
3. Contract is shared amongst teams to enable collaboration, using tools like Pactflow
4. Requests in contract replayed against provider API and verified against consumer's expectations
5. Provider tests mock out any other systems, so it can be tested in isolation

PACTS / MICROCKS / etc.

[https://docs.pact.io/]
Prioritizing Natural Language Tests

```java
import org.junit.Assert;
import org.junit.Test;

public class CalculatorTest {
    @Test
    public void testAddition() {
        Calculator calculator = new Calculator();
        int num1 = 5;
        int num2 = 10;
        int result = calculator.add(num1, num2);
        int expectedResult = 15;
        Assert.assertEquals(expectedResult, result);
    }
}
```

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>Expected Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Launch the browser.</td>
<td>Browser is started.</td>
</tr>
<tr>
<td>2</td>
<td>Click menu ➔ select &quot;Customize&quot;</td>
<td>The &quot;Customize&quot; window is opened.</td>
</tr>
<tr>
<td>3</td>
<td>Drag 3 new items from the palette or menu panel and drop them onto the Navigation toolbar</td>
<td>All items are added onto the Navigation toolbar.</td>
</tr>
<tr>
<td>4</td>
<td>Exit &quot;Customize&quot;</td>
<td>The changes are applied.</td>
</tr>
</tbody>
</table>
| 5    | Wait at least 15 seconds, after exiting "Customize", then restart the browser. | Browser is restarted and the previously made customizations are in place.
Prioritizing Natural Language Tests

- Word2Vec
- Language models (BERT)
- Clustering

Evaluating on real-world data
Comparing to other techniques
Non-deterministically failing tests are called *flaky tests*. Flaky tests hinder efficient continuous integration. Current solutions mostly limit to rerunning tests. Data driven solutions tailored to CI:
- Detection
- Root causing
- Determine origin
- …
Swiftly Detecting Flaky Failures

Goal: distinguish FLAKY failures from REGRESSION failures.

Why? Culprit finding

How? Train models based on
- Flip rate
- Duration history
- Code Changes
- Error message (new)

Literature:
Practical Flaky Test Prediction using Common Code Evolution and Test History Data – ICST 2023

Novel contributions:
- Dataset of flaky failures in CI in open-source projects
- Replication of existing approach
- Introduce new features
An Empirical Assessment of Neural Embeddings Techniques for Web Testing

Problem
Redundant + Incomplete
An Empirical Assessment of Neural Embeddings Techniques for Web Testing

Clone pages
Structurally/visually similar pages that represent the same functionality

- The State Abstraction Function determines the amount of redundancy
  - DOM Tree similarity (e.g., with RTED)
  - Visual screenshot similarity (e.g., with PDiff)
  - Several shortcomings (hard to generalize, need thresholds)
An Empirical Assessment of Neural Embeddings Techniques for Web Testing

- Propose novel state abstraction functions based on neural embeddings
  - DOC2Vec
  - Code2Vec
  - BERT

- Compare them with existing approaches (e.g., RTED and PDiff)

- Evaluate them within an existing crawler in the context of web testing
  - Accuracy of model inference (redundancy and completeness)
  - Code Coverage of the web app
  - Number of faults retrieved

In this thesis you will learn how to

- ... develop / fine-tune real-world neural embedding models to a new domain (AI)
- ... automatically test a web app (SE)
- ... perform rigorous empirical experiments
- ... report scientific findings (positive outcomes -> scientific publication)
Benchmarking
LLM-based Code Generators

Large Language Models like ChatGPT become more and more powerful. They can be used for a multitude of purposes including programming. But – is this actually true?

- How can we even measure their performance?
- What are suitable and realistic benchmarks?
- And how different do the models perform?
Accessibility tasks during software development

Our challenge:
- countless resources on accessibility
- low level of knowledge among most software developers and designers
- develop and deliver accessible software

Our approach:
- provide guidance during the software development process: phase by phase, matching the context
- task-driven: clear descriptions, helpful resources
How can we achieve Accessibility?

Your topic:

• What tasks do software developers and designers have to perform to build accessible software?

• Convert existing literature into hands-on descriptions of tasks
Participating

1. Apply via matching tool

2. Apply with us: Online form
   - Letter of motivation
   - Study Program and Semester
   - Optional: CV + grade report
   - Your 3+ favorite topics

July 19th

http://go.tum.de/070420