Prof. Dr. Oscar Nierstrasz
Pascal Gadient, Pooja Rani

Solution
Assignment 12 — 05/12/2018 – v1.0
Code and Test Smells

Please submit this exercise by mail to sma@list.inf.unibe.ch before 12 December 2018, 10:15am.

Exercise 1: Code smells (3 Points)

- What is the fundamental problem in developers bad code smell perception? **Answer:**

  The perception is a subjective matter. Hence, it is very difficult for developers to reach an agreement regarding the "smelliness" of code. The perception is even highly context dependent, e.g., an internal application for cleaning log files is in general less exposed to security threats than web endpoints, thus some smells might become neglected. Moreover, different programmers have different programming habits shaped by their culture, and previous experiences. Hence, a code smell in one project might not be considered as a code smell in another project. Anaïs Nin sums it up quite nicely: "We don't see things as they are, we see things as we are."

- What is *association rule mining* in the context of the HIST code smell paper you can find here? **Answer:**

  More general, association rules are the correlations between elements that co-occur frequently within a dataset consisting of multiple independent selections of elements. In the context of the HIST paper, the association rules represent correlations between subsets of methods in the same class that frequently change together. The discovery of all association rules within one or more projects is then called "association rule mining".

- Provide a test code example that includes a test smell and explain where the smell is, and why it is a smell? **Answer:**

```
public void testDataIsVariable() throws Throwable {
  JSTerm term = new JSTerm();
  term.makeVariable();
  term.add((Object) "");
  jSTerm0.matches(jSTerm0);
  assertEquals(false, term.isGround());
  assertEquals(true, term.isVariable());
}
```

  This test uses two different assert statements each testing a different method within the same test case. A test containing multiple (independent) asserts is also known as "EagerTest". This complicates not only the implementation of the test, it also misleads developers when the test breaks due to the inappropriate test name, and finally, it makes it harder to fix code that breaks such a test because it is not obvious what the cause of the failure is.

Exercise 2: Change of roles (6 Points)

Please craft three potential exam questions with their corresponding answers. You might want to think carefully, because some of them might appear in the upcoming exam. The questions should not take more than three minutes to solve. You will retrieve feedback about each question, and deducted points if your answer is wrong or incomplete. **Answer:**

Please have a look at the Q&A summary on the SCG webpage.

Exercise 3: Exam preparation (1 Point)

Please start reviewing the content of this course and ask questions, if any, by mail. The most important questions will be discussed during next practical session which will be a Q&A session for the final exam. Deadline for questions: upcoming Monday, 10/12/2018, 12:00pm (midnight). **Answer:**

Please have a look at the Q&A summary on the SCG webpage.