A Survey on Software Testability

This survey is an attempt to understand testability from software practitioners' perspective. It has 13 questions and it may take 10-15 minutes of your time. Please note that the survey is anonymous.

Please direct your questions and concerns, if any, to Tushar Sharma (tushar@dal.ca), Stefanos Georgiou (stefanos1316@gmail.com), Tahir Ghaleb (taher.a.ghaleb@gmail.com), Maria Kechagia (m.kechagia@ucl.ac.uk), or Federica Sarro (f.sarro@ucl.ac.uk).

* Required

1. What best describes your role in software development? *

- [ ] Software developer
- [ ] QA/Tester
- [ ] Project manager/Team lead
- [ ] DevOps Engineer
- [ ] Other

2. Please specify your software development experience in years. It is intended to gauge the experience level of the participants. *

- ◯ 0
- ◯ 1-2
- ◯ 3-5

- 6-10
- 10-20
- >20

Testability - Definition

3. How do you define 'software testability'? *

Testability smells

We define software **testability** as the "*ease of testing enabled by software design*". In other words, to what extent the chosen software design decisions enable easy automated software testing.

Please note that testability smells are different from **test smells** (such as *assertion roulette* or *lazy test*). Test smells are quality issues *in* tests; testability smells are quality issues that make writing tests difficult.

4. **Hard-wired dependency**
   A hard-wired dependency is created when a concrete class is instantiated and used in a class. For example:

```
class Painting {
    HardFrame frame;
    public Painting() {
        frame = new HardFrame();
    }
/*other methods*/
}
```

   A hard-wired dependency makes the class difficult to test because the newly instantiate objects are not replaceable with test doubles (such as

mocks and fakes). Hence, such dependencies reduces testability.

To what extent do you agree with the above statement? *

- ◯ Completely agree
- ◯ Somewhat agree
- ◯ Neither agree nor disagree
- ◯ Somewhat disagree
- ◯ Completely disagree

5. If you have selected either 'Completely agree' or 'Somewhat agree', why in **your experience** it would be difficult for you to test a class with **Hard-wired dependency**?

   Or, if you have selected one of the other options, how do you typically test a class with **Hard-wired dependency?**

6. **Global state**
   Global state refers to a global variable or Singleton object. Global variables create hidden channels of communication among abstractions in the system even when they don't depend on each other explicitly. Global variables introduce unpredictability and hence makes tests difficult to write.

   To what extent do you agree with the above statement? *

   - ◯ Completely agree
   - ◯ Somewhat agree
   - ◯ Neither agree nor disagree
   - ◯ Somewhat disagree
   - ◯ Completely disagree

7. If you have selected either 'Completely agree' or 'Somewhat agree', why in **your experience** it would be difficult for you to test a class with **Global state**?

   Or, if you have selected one of the other options, how do you typically test a class with **Global state?**

8. **Excessive dependency**
   Dependencies make testing harder. A large number of dependencies makes it difficult to write tests for the class under test in isolation.

   To what extent do you agree with the above statement? *

   - ◯ Completely agree
   - ◯ Somewhat agree
   - ◯ Neither agree nor disagree
   - ◯ Somewhat disagree
   - ◯ Completely disagree

9. If you have selected either 'Completely agree' or 'Somewhat agree', why in **your experience** it would be difficult for you to test a class with **Excessive dependency**?

   Or, if you have selected one of the other options, how do you typically test a class with **Excessive dependency?**

10. **Law of Demeter violation**
    A class that is interacting with objects that are not specified as the class member or parameters of its methods violates the law of Demeter. In other words, the class has a chain of method calls such as

type.getPackage().getProject().parse().

Violation of the law of Demeter creates additional dependencies that the test has to take care. Moreover, some of the dependencies may not be relevant for the class under test. Hence, violation of the law reduces testability.

To what extent do you agree with the above statement? *

- ◯ Completely agree
- ◯ Somewhat agree
- ◯ Neither agree nor disagree
- ◯ Somewhat disagree
- ◯ [illegible]

11. If you have selected either 'Completely agree' or 'Somewhat agree', why in **your experience** it would be difficult for you to test a class with **Law of Demeter violation?**

    Or, if you have selected one of the other options, how do you typically test a class with **Law of Demeter violation?**

12. Kindly list other practices from your experience that you consider impact **software testability**. Feel free to provide example code or scenario in which such practices occur.

13. Thank you very much for taking this survey. Feel free to provide any suggestion or comment about this survey.