CS 528 : Object Oriented Programming and Design **Summer 2014**

Problem Set 7

Dan McQuillan *Handed In: June 29, 2014*

Part 0 Click the links above and read the original blog post, and the response from Kent Beck. Then, answer the following briefly (2-3 sentences).

(a) What are the major complaints that David brings against TDD?

(b) What does he propose as being the solution?

(c) Pick two of the things Kent Beck would miss without using TDD. Can you accomplish those things without the use of TDD?

Part 1 (a) What is a Mock Object? Give an example of the use for a mock.

(b) At one point, Kent says the following:

Even if I don?t know how to implement something I can almost always figure out how to write a test for it. And if I can?t figure out how to write a test, I have no business writing it in the first place.

Do you agree with this sentiment? Why or why not?

(c) What is the ?red/green/refactor? loop?

(d) Which way of arriving at a test suite is better, in your opinion: ?going through the tests? (test-first) or ?going to the tests? (test-after)? Does it matter at all? Does this depend on the situation? (If so, in what way/ways?)

(e) In what cases is TDD not (easily) applicable? Give at least one example. What can you do in these cases?

(f) In what ways can an overzealous demand for isolation in unit testing create problems? Is this inherent in unit-testing or TDD?

(g) Why should you have an automated unit test suite? Give at least two reasons.

Part 2 (a) What is an MVC application? Where did this design pattern originate?

(b) Can an application have a worse design because of an attempt to make it more testable? Or is it the case that something that is more testable is always a better design?

(c) How does TDD influence the design of code? Give a positive and a negative example.

(d) David correlates the size of a codebase with how easy it is to change it. Does this correlation ring true, in your experience?

(e) Compare the situation David brings up of ?test-induced design damage? coming from a desire for isolation in TDD with the concept of ?speculative generality?. Are the similar? the same? why/why not?

(f) What is cohesion? What is coupling? In what ways can more of one mean less of the other?

(g) Kent says,

Difficulty testing is a symptom of poor design.

Do you agree?

(h) Kent says he is optimistic in that he believes that there is always some design insight that exists that will result in a design that is simultaneously more easily testable as well as better structured.

Do you share his optimism?

Part 3 (a) Martin identifies three important forms of feedback for software development. What were they, and how well is each one served by having a unit test suite?

(b) How has Quality Assurance (QA) for programming changed over time? In particular, what role did TDD and self-testing code play in this change?

(c) What does David mean when he refers to ?criticality?? How should the ?criticality? of your software influence your testing?

(d) Describe how it's possible for code with 100% test coverage with all tests passing can still have bugs.

Part 4 (a) Kent mentions a good measure for understanding how important a given test is for a codebase. What is it? How could you determine its value for a given test? Using this measure, how could you numerically define ?overtesting??

(b) Describe Martin?s trick used to identify if a given line of code is tested by the current test suite.

(c) How is the ratio of functional code to test code influenced by coupling in the system?

(d) What are the symptoms of overtested code? undertested code?

(e) How does the ?audience? of a codebase change its testing tradeoffs?

(f) Describe the difference between placing unit tests as the ?authority? of the system differs from placing the functional code as the ?authority? of the system. Which places a larger focus on refactoring?

(g) What is one situation where you would be more happy to lose the tests and keep the code? What is one situation where you would be more happy to lose the code and keep the tests? What is different about these two situations?

(h) David thinks the testing battle has been won, and Martin isn?t so convinced. Do you believe that we?ve ?won? the battle to get people to unit test their code? Why or why not?