The Craftsman: 38
Dosage Tracking XV
Test Independence

Robert C. Martin
11 May 2005

...Continued from last month.

President Henry Wallace was sworn into office on the 13th of November, 1942. Wallace was a scientist and a visionary. Before he took office he had dreamed of conquering fascism, and creating a just and lasting peace throughout the world. He felt the Earth was on the verge of "The Century of the Common Man".

A weaker man might have given up hope upon learning that "the common man" had less than twenty years to live. But Wallace was not cowed. He had read the Contingent's reports on the use of Atomic Energy to deflect, or at least escape, Clyde. He determined that the resources of the entire world must be brought to bear on this problem, and he was determined to lead the U.S. in making that happen.

Only one thing was in his way... The Eastern Hemisphere.

---

22 Feb 2002, 0830

"The first thing to do, my dear Fonse, is to get all these tests to pass. Do you know why they are failing?" Jasper's toothy grin was both congenial and grating.

"I think so." I said, trying not to appear befuddled. "Apparently the first test to run changes the database in a way that the next test doesn't expect."

"You got it, buddy! I'm pretty sure that's the case. So we need to clear the database between each test case."

This sounded strange to me, like extra work. "Why don't we just change the second test to expect what the first test leaves behind? That way we don't have to clean up. Even better, it means that the tests can build on each other. Each test leaves the database all set up for the next test." I was proud of myself for thinking this through.

Jasper waggled his finger at me. His thick blond hair, cut of at his ears, jiggled as he shook his head. "Nosirree, Fonse ol' pal. We don't let tests depend on each other. We want to be able to run any test at any time. We don't want to have to run them in sequence."

In the back of my mind I could see Avery rolling his eyes at this thought. I suppressed that gesture, and simply asked: "Why is that? Wouldn't it be much easier to run the tests in sequence?"

"It might be a little bit easier *right now*", Jasper said with sugary emphasis on the last two words, "but it would make it very hard to diagnose problems later on." Jasper put his arm around my shoulder as he continued his syrupy lecture: "Think about what would happen, Fonse, if one of the tests in your sequence failed? *All* the downstream tests would fail too, wouldn't they?"

I politely disengaged from his arm and said: "Well, sure, but so what? You'd know what test failed."

A glimmer of regret passed over Jasper's face as I backed away from him. "True, but you wouldn't know if any of the downstream tests would have passed. To find out you'll have to get that first failing test to pass. If other, later, tests fail, you'll have to get them all working one by one."

Jasper paused for a second and then said: “It’d be like compiling a C++ program. Have you ever done that?”

I shook my head.

“Never mind. You can see that it would be inconvenient. It’s better for each test to fail for it’s own reasons, than because a previous test failed. Also, it’s very convenient to be able to run any test at any time. If we put them in a sequence, then we wouldn’t be able run individual tests.”

I sighed. This actually made some sense. I wondered why I didn’t like hearing sense from Jasper. Perhaps it was because the more he talked, the wider his grin got, making it look like his head was split in half.

“OK.” I said, “So we need to clear the database in front of each test case.”

Jasper winked and said: “Attaboy, Fonse!”

This was torture. I didn’t know how much longer I could keep cool. So I faced the keyboard and ran all the unit tests again. Two unit test files were failing: `RegistrarTest`, and `UtilitiesTest`. I opened each, and looked them over. Sure enough, each of the failing test cases made use of the `SuitGateway`, our front end to the database. I opened `SuitGateway` and saw this:

```
public class SuitGateway {
  // todo move initialization somewhere else.
  public static final ISuitGateway instance = new InMemorySuitGateway();

  public static int getNumberOfSuits() {
    return instance.getNumberOfSuits();
  }

  public static Object[] getArrayOfSuits() {
    return instance.getArrayOfSuits();
  }

  public static void add(Suit suit) {
    instance.add(suit);
  }
}
```

I thought to myself that perhaps it was time to follow the advice in that `todo` comment. So I changed the initialization of the `instance` variable to `null`, and ran all the unit tests. Of course I got lots of null pointer exceptions, but only in `RegistrarTest` and `UtilitiesTest`. So I opened `RegistrarTest` and modified it as follows:

```
public class RegistrarTest extends TestCase {
  protected void setUp() throws Exception {
    SuitGateway.instance = new InMemorySuitGateway();
  }

  ...
}
```

The `setUp` function is called in front of every test function, so now each test function will start with an empty database. I pushed the test button and *all* the unit tests now passed, even the ones in `UtilitiesTest`!

“That’s strange.” I said.

Jasper grinned at me like he’d just heard a funny joke. “C’mon, Fonse, use your head there buddy. The last test case in `RegistrarTest` just left the database empty.”

I could feel the heat rise in my face. I hoped the flush wasn’t visible. I took a deep breath and said: ”I see what you mean. But I still need to clear the database in `UtilitiesTest`, just to make sure.”

“Oh, absolutely Fonse! I won’t argue with that!”

His Cheshire cat smile hung in space behind my closed eyes. When it faded I made the same change to `UtilitiesTest`, and the unit tests all continued to pass.

Jasper gave me a gentle elbow in the ribs and said: “Nice sailing captain, now let’s fix those acceptance tests.”

Something cold began to rise in me, replacing my flush with ice, or perhaps steel. I took another deep breath, and realized it was the last one I was willing to take. I ran the acceptance tests. Sure enough they all failed for null pointer violations.

The first fixture in both acceptance tests was `DTrackContext`. This fixture was made for initialization. So I made the following changes.

```
public class DTrackContext extends ColumnFixture {
  public Date todaysDate;

  public void execute() throws Exception {
    Utilities.testDate = todaysDate;
    SuitGateway.instance = new InMemorySuitGateway();
  }
}
```

This made sure that each of the two test pages now started with a cleared database. I pushed the suite button, and saw green; all the acceptance tests pass.

I knew it was coming without even looking. I could feel his enthusiastic congratulations before they left his mouth. I waited for it with ice water in my veins, knowing what I had to do.

He patted me on the head. “Nice work Fonse!”

He patted me on the head! I closed my eyes. I was done. There would be *no more* of this. I tightened my stomach and visceral muscles and turned towards him.

“Jasper, my name is *Alphonse*. Please don’t call me Fonse anymore. And *please* stop being so patronizing. I’m not going to be able to work with you if you keep treating me like a pet. I know you are more experienced than I am, but that doesn’t make me a low grade moron who needs constant cuddling. Now, if you’ll excuse me, I’m going to the washroom.”

Had I really just said all that? Had I really been so calm and articulate? As I approached the washroom I could feel the ice-water drain away to be replaced with fear and embarrassment. I stood in the washroom and waited for the shakes to stop. Jean was going to be furious with me. She was going to take me to the woodshed. But I couldn’t go on like that. I had to stop him.

Didn’t I?

The code for this article can be located at:

http://www.objectmentor.com/resources/articles/CraftsmanCode/Craftsman_38_DosageTrackingSystem.zip

To be continued...