The Craftsman: 27
Dosage Tracking IV
Carole's way, or the HighWay.

Robert C. Martin
28 May 2004

...Continued from last month.

---

21 Feb 2002, 1000

In the first decade of the 20th century, Percival Lowell predicted the existence of a ninth planet by studying the motion of Uranus. Though he searched, he died before he could complete the effort. In 1929 a young man named Clyde Tombaugh came to Lowell Observatory and resumed Lowell's search for planet X. The procedure was both delicate and tedious. Two photographic plates, taken of the same stretch of sky, but on different nights, were put into a device known as a "blinker". The blinker displayed the two plates one at a time, quickly alternating between them. Any object on one plate that was in a different position on the other would appear to blink. Clyde Tombaugh found Pluto on the 18th of February, 1930. In the summer of 1935 he found the angel of death.

The newspapers named the rock "Clyde", and had a few days poking fun at the notion that Clyde might actually hit the Earth in 1959. But the world was headed for war, and even a possible doomsday rock couldn't hold the papers' attention for long. Besides, as the astronomers kept saying, the odds of a collision were millions to one against.

Carole and Jasper grabbed another workstation to work on the next acceptance test, while Avery, Jerry, and I started working on making the *Register Normal Suit* test run.

"OK, the first thing we need to do is write the fixture for the first table." Said Jerry.

"What's a fixture?" I asked.

"A fixture is a Java class that binds the table to the DTrack application." Jerry replied.

"But there is no DTrack application." Avery complained.

"True." Said Jerry. "We write the tests and fixtures first to make sure that the application is designed to be testable."

"Oh, sort of like writing unit tests first." I said.

"Yes, it's a bit like that", replied Jerry, "except we write the whole test and fixture before writing any of the application. Indeed, we write many tests and fixtures before writing the application."

"What does a fixture look like?" Avery asked.

Jerry pulled up the test that he and Carole had just finished. "Look at the first table." He said.

```
DTrack Context
Today's date
2/21/2002
```

"Now watch what happens to that table when I hit the *Test* button."

DTrack Context
Could not find fixture: DTrackContext.
Today's date
2/21/2002

"Does that mean we have to write a class named DTrackContext?" I asked.

"That's exactly what it means." Jerry said, passing me the keyboard. "Put it in the dtrack.fixtures package.

So I wrote:

```
package dtrack.fixtures;

public class DtrackContext {
}
```

"Great." Said Jerry. "Now compile that and run the test again."

It compiled without a problem, of course; but when I hit the *Test* button, I got the same error as before.

"Do you know why?" Jerry asked.

I thought I did, and I started to answer, but Avery beat me to it by saying: "It's a classpath issue; FitNesse doesn't know where the `DTrackContext.class` file is."

"Right you are, Avery." Jerry beamed. I could see a smug little smile flicker on Avery's face.

Jerry took the keyboard and made the following changes to the test page:

```
!path C:\MyProjects\DosageTrackingSystem\classes

!|DTrack Context|
|Today's date|
|2/21/2002|
```

"This tells FitNesse what the classpath of the fixtures is." Said Jerry.

"But it still fails." Complained Avery, who had just hit the *Test* button.

Jerry looked expectantly at us.

"Oh!" I said. "The name of the class is `dtrack.fixtures.DTrackContext`, not just `DTrackContext`."

"Right again!" Said Jerry as Avery Scowled. "Why don't you make that change?"

So I changed the page to look like this:

```
!path C:\MyProjects\DosageTrackingSystem\classes

!|dtrack.fixtures.DTrackContext|
|Today's date|
|2/21/2002|
```

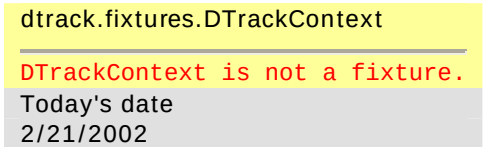
And it displayed like this:

dtrack.fixtures.DTrackContext
Today's date
2/21/2002

"I can sort of see the syntax of this." I said. The strokes are table cell separators. But what is the bang (!) at the beginning?"

"Don't worry about that for now." Said Jerry. "You can read up on FitNesse in your spare time. The syntax is pretty easy to get used to. For now, let's just concentrate on getting this fixture written. So run the test."

I pushed the Test button, and saw a different kind of failure.



"Good!" Said Jerry. "It found the fixture class."

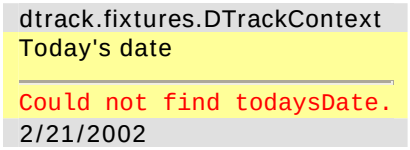
"Yeah, but it says it isn't a fixture." Avery whined.

"I can fix that." Jerry said, as he took the keyboard. He made the following changes to the fixture class.

```
package dtrack.fixtures;
import fit.ColumnFixture;

public class DTrackContext extends ColumnFixture {
}
```

"That's better!" Jerry said as he pushed the *Test* button.



"Not much!" Said Avery with a smirk.

"What is it looking for?" I asked? "A variable?"

"Bingo!" Said Jerry, who continued typing.

```
package dtrack.fixtures;
import fit.ColumnFixture;
import java.util.Date;

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

"Ick!" Cried Avery. "A public variable! That's not a very OO construct!"

Jerry looked calmly over at Avery and said: "So what?"

"It breaks encapsulation!" Avery sputtered with righteous indignation.

"Fixture classes aren't encapsulated in the usual manner." Jerry explained. "Public variables are one of the ways we communicate with them. Anyway this isn't the time for a lesson on the true principles of OO. Right now we want to get this fixture done." So he pushed the *Test* button as Avery rolled his eyes impatiently.

dtrack.fixtures.DTrackContext
Today's date
2/21/2002

Avery burst out with a huge guffaw: “Oh great! All that work to make it look normal again!”.

“Right!” Said Jerry. “Now we know that the fixture is being found and that the data is getting into it as expected.”

“It doesn’t look as nice as it did before.” I said. “That package name dirties things up a bit. I like the way the variable name uses spaces and punctuation. Can’t the fixture name do the same?”

“Indeed it can!” Said Jerry, as he opened the test page and made the following changes:

```
!path C:\MyProjects\DosageTrackingSystem\classes

!3 Normal suit registration.

!|Import|
|dtrack.fixtures|

'' * We assume that today is 2/21/2002.''
!|DTrack Context|
|Today's date|
|2/21/2002|
```

This displayed as:

classpath: C:\MyProjects\DosageTrackingSystem\classes

Normal suit registration.

Import
dtrack.fixtures

- *We assume that today is 2/21/2002.*

DTrack Context
Today's date
2/21/2002

“OK, that’s much nicer, I said. Now what do we do with that date?”

“Good point.” Said Jerry. “We need to put that date somewhere that the rest of the DTrack system can get it from.”

Avery assumed a superior air and said: “Wouldn’t it be easier if DTrack just used the regular `Date` class to get today’s date? Why do we have to invent a whole new mechanism for something as simple as today’s date?”

“Because the tests need to control the date in order to make sure the application manages it correctly.” Jerry replied, a little annoyed.

“Yeah, but that’s just extra work that slows us down! We need to have this done in two months!” Avery had raised his voice enough for Carole to overhear. She quickly came over, looked Avery in the eye, and said:

“No, Avery, it’s not extra work, and it doesn’t slow us down. We go much faster when we write these

tests and build systems that are testable. You are right, Avery, we've only got two months. And the only way we'll make it is if we write the tests and follow our disciplines. We've done it both ways -- haven't we Jerry? (he grimaced, but nodded) -- and I can tell you that as long as *I'm* the customer on this project, *we're* going to be writing acceptance tests, and *you* will be doing it the way *Jerry* leads." And she strode back over to Jasper in a huff.

Avery had paled under Carole's tongue lashing. Now he looked at Jerry and me and said: "Whoa!"

"Yeah, she can be a little intense at times." Jerry said calmly. "The point is that we've decided to do things this way, and if you want to be part of the team, you'll have to go along."

"I still think it's a waste of time." Avery said under his breath.

"Suspend your disbelief for awhile." Said Jerry. "Trust me, we're not fools. This is the best way for us to proceed."

Avery shrugged, but nodded. He looked at me and rolled his eyes. I just stayed out of it.

"OK." I said. "Now what about that date? How does our fixture communicate it to the rest of the system, and how should the rest of the system gain access to the date?"

Jerry looked at the two of us, then glanced over his shoulder at Carole who had busied herself with Jasper, and then sheepishly he said: "I think it's time for a break. Let's get out of here for a few minutes and we can talk about why public variables are sometimes appropriate."

So the three of us left the lab and headed for the nearest lounge.

To be continued...
