andrew Bolton

8/14/21

when I looked at the requirements I approached. My. Software development based on. The

requirements that were given to me. As a starting point, but not as an ending point. For me, I wanted to

make it an actual functioning program. Which did cause me some trouble down the road with the. J unit

testing. I did find that even though there were some functions that were in there that. Didn't have any

particular uses. I got dinged for. In the J unit. Testing overall. Because it didn't meet the coverage. Some

of my test cases, even though I thought they would. Hit that line of code. They did not. And after several

times of going back. For some help. I ended up having to comment out the code in order to get my

desired number. With a little bit of explanation given to me, I did finally understand that. A high number

doesn't always mean a good working program. Because there are some things that you can't just test

for. And the higher the number. The harder and more test cases you have to try to come up with too

come up and or develop or make that line of code be used? I think the number is more of a baseline and

we shouldn't go off of that whole number. We should mainly look to see if. Our test case scenarios.

Match up with our code. And not the percentage that is actually covered. I have found and talked to

other people that have said. you can have a high percentage of coverage. But your code and actual.

Production may fail horribly. Because there are some test cases you cannot predict or right. For J unit

testing or in any other testing for that matter. So in order to achieve my number in this assignment. I

commented out part of my code. So I could achieve my 80%. Even though. That piece of code would

need to be there. To make the whole thing function properly in production.

Right here is a piece of commented code. That I took out. And it raised my. Coverage level by a lot. But I am not sure how to make a test case for this.Graphical user interface, text, application, email

Description automatically generated

As for my 3 approaches. I tested, debug and ran my program to see if it worked. Even though I only hit 12% coverage. The program itself ran great. Had no issues and accepted anything I pretty much thrown at it.

I ensured my code was technically sound. By Writing Pacific test cases. They either throw exceptions or

errors in my code. And if it could not handle it. My coverage would be lower or so I thought. But in the

end I ended up testing specific functions of the code. To make sure it worked. And that's how I made

sure it was technically sound. There were some test cases. I honestly did not know how to test for since

it test line by line. As I wrote my test cases, I tested strictly towards functionality. Because I wasn't too

sure how they test some of the other pieces of my code, such as security and performance via unit

testing. Down below here is a sample of my test case that I used. For appointment services.

Graphical user interface, text, application, email

Description automatically generatedtested the calendar piece. The adding of an appointment and the test of deleting an appointment. With

variables that could not be returned as null. Now this I know tested sections of my code. That were

functioning. In that respect, where it would target a Pacific area like add appointment and delete

appointment.

Some of the other testing methods which I am interesting in learning more about and have been doing a

little bit of reading on. Is security testing. And how I would write a case for that? Along with

performance testing. And how I would measure that? Because in my mind I'm not too sure if I even did it

right by testing how long it took for the program to run start to finish. It was security testing on currently

out the ballpark on how to test that/measure that. What I do know about performance testing. Is that?

It is a non functional software technique that determines. They state ability, speed and scalability and

responsiveness. Of an application. When given a heavy workload. But how to duplicate that workload?

By myself in the IDE, I'm a little confused on. Also, I'm not sure by what standard I should look at. Now,

regarding security testing. I'm somewhat familiar about. How that works? And what I'm looking for like

penetration testing, risk assessment, security audit are all part of security testing. But I'm not sure how

to make test cases to be able to accurately perform these actions.

For security testing would be actually very good for. A lot of applications that are going to be housing

personal information. Or sensitive data like your Social Security number. And medical records. This is

something that would want to be done with every program. Weather outsourced or in-house. The other

part that I talked about. The performance testing is something that should be done as a standard due to

the fact that it's testing to see how well it handles and performs with a workload if not coded properly in

A to much large volume of information flowing through it. Could cause the program to either falter.

Abend or crash and worst case scenario. I would have to say out of. The security testing of performance.

Our security testing would actually take number one because we do not want our data vulnerable to

other people. But at the same time, performance is a big factor because sometimes our job. That we

made is relying. On another job, and if we can't finish. And output the data in time for the other process

to pick it up. It will cause the program the app end in failure.

While it is important to exercise caution when writing your code. You also shouldn't be too cautious at

the same time. Because sometimes by being too cautious you end up missing key points that could be

crucial later on. For example, the requirement. In week 5 Appointment services requirements states the following.

Graphical user interface, text, application

Description automatically generatedNow if we were to code exactly to the 30 of the requirements, the program would not operate properly.

It would have some actual serious issues. Along with. Some very big issues on handling. Some other

scenarios? That are not outlined in their requirements. But are somewhat given when you're making an

appointment. Even though this is just a basic requirement, it is something to also build off of as a

starting point.

Graphical user interface, text, application

Description automatically generated

Center Sumption that is made is that they've made an appointment before, but how do we reference

that appointment? We're not just making a date because we can have more than one appointment on

the same date to multiple. Doctors. Or office visits. So how do you separate those? Different people.

And appointments that they are going to. As seen on line 13, the process starts by making a unique ID.

Which is returned. In a random string. Which is then. Using a public. Appointment service to generate

this and add it to the appointment book along with the date of the appointment. Now nowhere in our

requirements was this outline. But when you think of appointments. What do you think of just one

appointment today? No, you think of multiple appointments, so you need to. Take what you're given

sometimes and expand and expect. Maybe there are multiple things happening at once and not just one

appointment being added, but maybe five or six? And how are you going to keep track of those? The

same goes with deleting the appointment. There was a very hard part for me during. The actual testing

of my own code and that was keeping the biased. A bias. Review is being totally open and not saying

something is totally wrong even. If it could be right, it's a different way of someone else doing it. I've

talked to people that have done some code reviews, like my mother and my Internship boss. They both

tell me you have. To be careful. And very strict. With the way things are done. Because someone. May

not be as experienced with writing code and you got be careful with the way you word or your

recommendations or reject their work. Because sometimes it's simply in the air that they missed. Or

that they didn't think of. Both people have told me that they've seen it happen before. Along

with sometimes they just simply overcomplicated it when it could have really only been three lines of

code change, but. They ended up. Writing an entire new system when it wasn't even needed. Which

takes up space. So you have to limit to what does it need to do pacifically? And does it get the job done?

The right way. And by what I mean right way? I mean, with the littlest effort by not making something

overcomplex. Simplify it. Make it easy for anyone to understand. Just in case there needs to be a

modification down the road. And as I evaluate. My. Own testing strategy. Of my code. I found that I was.

Two in experience and I overcomplicated my code even though. It was designed to actually be running. I

did not write my test cases for that and did I quite understand how the test cases targeted my code. The

best example is what I could give what I had before is line 13 where I wrote for a unique ID, so that way I

could separate the different appointments that were being made for the same day. But I did not end up

testing for that, nor was I sure how to test for that. In the end that ended up hurting me versus helping

me achieve my 80% coverage.