Maddy Cook

9-9-16

Software testing

Pete tucker

EX01\_01

1. College I use this program to this program to input people into the Whitworth database and to update records that already exist this includes addresses, phone numbers, emails, children, other relationships, and how they are related to Whitworth. I use this program almost every day. It is very critical that the software is high quality because it is the majority of where my time is spent when I am at work, I rarely have long projects that do not involve college. If the software fails essentially I cannot work and much of the university cannot do their job either. In fact, when college is down for a long period of time people just go home because no one can do their jobs.
2. Snap chat I use this App to communicate using pictures I have taken, the app has filters you can put over your face distorting it or making you look better. It is much like texting but with pictures instead of just words. It is not very critical that this software is high quality but I would not use it if it were not a decent quality. So I guess in a way it is important that it is decent quality. If this app fails it’s not a big deal to me, it does often for me and I keep using it, it caused me mild frustration but nothing that really would make me stop using it. The majority of the reasons snap chat crashes on my phone is because my phone does not process fast enough.
3. Spotify I use this program/app every day to play music in the car, at work, at home, and even doing homework. It has the ability to download music onto my phone and to stream it from the internet. The quality of this software must be high because it is important that the music quality is clear and not choppy. If the program fails it causes me a lot of frustration because most of the time I cannot concentrate if I don’t have music. Also because it does not crash or not work so little when Spotify is down it is rare and very frustrating.

EX01\_04

B) Ambiguities- it does not tell what to do with the overlapping numbers, it does not say how large either of the vectors are, and it does not say whether they are the same size vector or not.

Defects- Because it does not say what to do with overlapping numbers in my code they are only counted once and not twice, which throws off the count cycle, also because we do not know how large the vectors are we need to use the count and iteration process which causes issues when you go to count when adding two numbers. If the vectors are the same size one can just stop when you have reached double the size of vector a or vector b. if the vectors are not the same size things get more complicated because one vector has to stop adding numbers to the union vector once it has reached the end, if it doesn’t it will cause an error.

C) –Test if the union is correct –test if the end vector is the correct size –test if the overlap between the two vectors is correct –Test if all of the numbers made it into the union vector –Test if one vector is larger than the other that all of the second vector gets into the union vector –Test if the vectors are the same the union vector is the correct size –Test to make sure the numbers are being pushed to the union vector –Test to make sure there are no extra numbers that are not in Vectors a or b –Test whether or not the function returns the correct vector –Test if the vectors a and b get into the function correctly.