Ahram Kim

AppDetex Engineering

CS 498

February 9, 2018

Seminar 4 (02.09.18)

Today, I had the seminar from AppDetex Company. AppDetex is the global brand protection leader in combating brand infringement, fraud, and piracy within the increasingly complex worlds of mobile apps, marketplaces, and domains. They were talking about the topic of ‘Code Smells’. The work is also unfamiliar to me. According to the wikipedia.com, code small is any symptom in the source code of a program that possibly indicates a deeper problem in computer programming. The people from AppDetex have lots of information and knowledge.

Frontend Smells is using large dependencies for simple things. It is non semantic HTML. The copy paste is to think about your approach before you copy/paste functionality. Long/Complex function make the code not easy to understand. This code need to be shrinked for simple function. Long parameter list also makes the code difficult to read and refactor. To fix those problem, it should pass the original object and introduce parameter object. Bad and same name make it confusing and vague. Dead code means unused code, so it should be removed it. You can use VCS to look at what the old code.

Minimize Mutability is immutable objects are easier to design and implement. They have less error prone and secure. When making classes immutable, It does not provide the methods that modify the objects state. Ensure the class can not be extended, and make all field final and private. Also, do not expose any mutable objects, and make defensive copies of passed objects that could be mutable.

Avoiding the code smell, there is SOLID principles. The ‘S’ is single responsibility, it do one thing well. The ‘O’ is open and closed. The ‘L’ is like of substitution, objects should be able to be swapped. The ‘I’ is interface segregation, client specific interfaces better than general. The ‘D’ is dependency inversion, it depend on abstraction. In order to avoid high maintenance systems (continuous integration(AKA CI), it should continuous deployment(AKA CD), )

Their seminar was great for me. I got lots of information from the lecture. They said that good developers is needed to check in frequently. Do not check in broken code, untested code, when the build is broken. Jez Humble also said that “Essentially, it is the practice of releasing every good build to users.”. I liked all contents of today’s seminar. In other ways, their seminar made me want to get work at AppDetex due to presenter, their present made me have less concentration and confused, because they presented each section per each person.