Document of Project 2

Shiyang Zhang SUID:785878565

In project 1, I design some packages having specific functions to organize the operation of test harness, like repository, test executive, logging, appdomain, loader, file manager, etc. For some packages, I designed many functions for them. But in project 2, I divide some functions into pieces. And several packages can realize one function. So it seems like that some of my arrangement in project 1 is inappropriate.

In project 1, I have a storage named repository. It will store .dll files and logs. I assumed it’s virtual, so users have to upload .dll files into it. But in project 2, repository is some local files where users can copy files in it. It can save test requests, .dll files, logs separately. It’s responsible for file management and storage. In addition, in project 1, I assume logs have 4 different situation: success with output; success without output; fail with output; fail without output. However, in project 2, every success or failure has a output. The user can only know if it is success or not. If failed, they have to review the code to test and test driver.

Some detailed and complex progress in project 1 is just a few codes in project 2. For example, in project 1, loader will parse xml files, get dependency information, send dependency information to repository, and get .dll files list. However, in project 2, after we parsing xml files, information will be passed as references to other projects. Test harness will use some information to run the test. Sometimes codes are more concise and convincing.

In addition, I am not very clear about the relationship between queue, appdomain and thread. In project 2, I understand more of them. In my design, a test request will be parsed into several tests. These tests will enqueue orderly. When they stay in queue, main thread will create a child thread, and then test dequeues. Child thread will create child appdomain, then some test information about test will be loaded into it.

In project 1, I design a database to store user information and log information for convenient management. But in project 2, database is not realized. Because it is not involved with limiting and permission management.

When I start to programming, the meaning of project seems to be significant. If I did not write this OCD, I have no idea where to begin. Because I partition the whole function into several pieces, I know each function of package. So I can connect code pieces in project 2 and realize each small function. Finally, the whole function can be realized. Because my incorrect partition in project 1, some packages have too much responsibility, while some packages can be integrated with other packages. If I can consider something about code realization, maybe my package partition will be more accurate. Meanwhile, coding is a concise and specific way to describe functions, which can also help me figure out some logic problems of a project. So before we begin project 3, trying to think about the design of project 4 may help us finish project 3 better.