Review 5

2/1/12

CS6V81.502

Seungtack Baek

**An Empirical Study of Build Maintenance Effort**

Macintosh et al. addresses that build files are susceptible to the changes to either changes in test files or changes in production files. They studied this by finding the logical coupling between production code, test code and build code. The researchers classified each files as either production file, test file or build file. Also, at the same time, they looked at the each revision and analyzed what types of files was associated with each revision. Through this process, they were able to find that

1. Build file takes about 9% of the total project size,
2. Churn rates of build and source files are tends to be similar, and
3. Build code changes introduce more code change than those of source code.

In addition, they found that there are two types of ownership for build files: centralized and dispersed. They found this by labeling each author as production, test or build and observe the distribution of build-marked authors in relation to other type of authors. They could not draw any conclusion regarding relationship between type of build file ownership and the build maintenance efficacy. However, in conclusion, they claimed that “the overhead of build maintenance for individual developers is limited.” (can you explain this sentence???)

First of all, I noticed that they used median, instead of mean, to describe the center. I can see the point of choosing median over mean, for median is more resilient to outliers than mean. But the real question is, from 10 samples, how do you know which one is real outliers?

Also I like the fact that they start with how important build maintenance is and start analyzing when to be careful regarding build maintenance process (or outside of build maintenance process, actually – such as source code commit and test code commit). In their analysis, there is still some ambiguity. For example, in java project, we use “.properties” file to change project parameters. However, “the parameter” can be source code parameter, such as database location or number of connections, etc. At the same time, although rarely, it can also be regarded as build file parameters as well, such as files to link for either production or test. Classifying this sort of files can require subjective decision.

**Question:**

1. This article was not too hard to read once you know what “churn rate” means. However, still there are a lot of statistics (such as chi-square test), and new terminologies (such as Conf(A=>B) and Conv(A=>B)) are we supposed to know all of this, as a MSCS/MSCE?