

Example Size and Time Estimates for Project 5

Working Set

Assume Moderate size large project

5,000,000 lines of code

500 lines of code per module (10 functions at 50 lines each)

=> 10,000 modules in finished project

50 modules per developer

=> 200 developers

Schedule

5,000,000 lines / 30 lines/developer day = 166,667 developer days

166,667 developer days / 200 developers = 833.3 days

833.3 days / 240 working days /year = 3.47 years

Testing

Test strategy: every time we add a module to baseline, we run tests on all modules in baseline, so:

1 module => 1 test

2 modules => 2 more tests

3 modules => 3 more tests

n modules => n more tests

total tests = 1+2+3 + .. + n + .. + number of modules

=> total tests = $N(N+1)/2$ where N = number of modules

=> 50,005,000 tests

50,005,000 tests / 833.3 days = 60,006 tests /day average

24 hrs/day * 60 min/hr * 60 sec/min = 86,400 secs/day

Sequential Tests

60,006 test /day / 86,400 secs/day = 0.695 tests/sec

=> avg test time must be < $1/0.695$ tests/sec = 1.44 secs/test

=> we will want to run tests concurrently.

If we run tests on two threads with one processor, that may nearly double the test rate, since each test thread will spend a lot of time blocked on I/O, file access, etc. This won't go on increasing linearly with each thread, but at least, the allowable average time per test can be relaxed somewhat.

File Transfers

Check-in and Check-out

10,000 modules completed / 833.3 project days = 12 modules /day average
Avg of 3 versions per module => 36 modules/day
=> no problem

Extraction to Testbed for testing

Assume that the Testbed will only keep one version of any file in its cache, but won't remove files, overwriting older version when a new one is extracted. Adding new files will be roughly the same as check-in, e.g., about 36 modules/day => no problem.

Extracting and building an earlier version of some component will require more file transfers, as earlier versions are not kept in the file cache. Assume that 10 percent of the modules changed from the earlier version to the latest (obviously this is speculative, but probably not too far off). Since the full system build consists of 10,000 modules, this extraction will require the transfer of about 1000 modules.

Using the TimedFileStreamService example, we find a "typical" module transfer takes about 25 milliseconds for a 3500 byte file.
=> 25 seconds for the entire extraction. => no problem since this is a fairly rare occurrence.