

How to Increase *STRESS* in any Software Testing Project

By Scott G. Ames

Over the years, I have seen many stress testing projects, and one question that I am often asked is how to go about ramping up the amount of stress in a project. Now, please understand, I know that what is really being asked is how to go about ramping up the amount of *load*, not stress, in a test. Since I, however, am a quality engineer by trade and like specifics, this article will attempt to answer the question, not as it was intended, but rather, as it was asked. If you really want to increase the overall stress on your people and in your testing project, these are some of the best practices to follow.

- Skip any “unnecessary” steps.**

Defining goals, requirements, and test specifications takes a long time. These steps not only cause test engineers to create tests that will execute properly with something other than just simple test data but seriously cuts into their functional testing of Bejeweled and load testing of internet web servers. If a test works with any possible type of data, it significantly reduces the amount of stress in a testing project. Not taking the time to do this right increases stress by forcing people to have to do it over, and possibly over again, under deadline pressure. Also, defined goals, requirements, and specifications remove any possibility of scapegoating outsourced resources for any failures that occur in the test project. This is especially important when validation that a system works is more important than actually fixing any problems that might arise.

- Have a meeting.**

Better still, have lots of meetings. Require that everyone connected to the project attend, no matter how small his or her role, and without regard to his or her other responsibilities. The meetings should require preparation and follow up tasks that take at least twice as long as the meetings themselves. To increase stress, it is far more important that everyone know everybody else's responsibilities rather than actually accomplishing anything productive. This process creates stress two ways: it increases micromanagement, and it aids in choosing potential scapegoats.

- Delegate, Delegate, Delegate.**

You probably think this actually reduces stress, but proper delegation to increase stress is almost an art form. Delegate critical tasks to people who are completely inappropriate for those tasks. For example, make your testing tool consultants, who will have little or no knowledge of your business processes, responsible for collecting the data necessary to execute the tests. Give them no direction as to how to collect the data and no authority to obtain assistance from the subject matter experts who know how to collect and validate the data. Be careful to not go overboard here. Later on, when the project is approaching its deadline, you can re-delegate this task to people who are capable of actually

accomplishing it. In this manner, you can increase stress on multiple fronts and still keep the project from failing. If the project does fail, you can always scapegoat any of the people to whom these tasks were originally delegated. In the case above, since it is impossible for the consultants to deliver a successful project, it may even be possible to avoid paying them for their work. This is best accomplished when used in combination with the previous items.

- **Don't bother to validate the test environment.**

This is really an important step for increasing the stress in a testing project. It's even better to ignore any piddling little details like test tool environmental requirements, such as the operating system, networking protocols, and any other necessary software and system configuration settings. Just make blanket statements like:

"All of the machines have exactly identical configurations."
"The test system is an exact duplicate of the system we have here."
"Everything has been set up as you requested."

Later, when these statements are proven inaccurate, the project stress will greatly increase. The third statement above was once combined with "Our people spent the whole weekend working on it." Of course, that was before an automatic system restore early the next morning undid all of the system configuration work that had just been completed. Having to rebuild an environment two or three times while working on a project deadline is a wonderful way to increase stress. You will be amazed by the amount of stress added to a project by not having a properly configured environment and by making people scramble to make inadequate work-arounds in a short amount of time.

Better yet, don't bother with a test system at all. Simply use the production system as the test bed, and don't bother to back anything up. Remember, backups are only for people who make mistakes. And finally,

- **Ignore the stupid questions. Just make assumptions.**

This is a combination of all the above items. What assumptions, made at all levels, keep the project from running smoothly and greatly increase the amount of stress on the project? As complex and detailed as a software testing project is now, there are no stupid questions, only bad assumptions.

On second thought, just do it right the first time; otherwise, you may not be around to do it over. Who needs all that extra stress anyway?

SCOTT G. AMES is an ASQ Certified Software Quality Engineer.

He is the Quality Assurance and Software Testing Manager for AutoTester, Inc., a provider of software quality solutions in Dallas, Texas and is their SAP and Stress & Load Testing expert. He can be reached at sames@autotester.com or scottames@att.net.