What is the difference between fault seeding and fault injection?

Fault seeding is performed by actually changing the code being tested in order to check test coverage. Fault injection involves introducing errors “on the fly” in order to perturb the normal flow of a program either with the purpose of extending test coverage or stress testing the system.

Software survivability and reliability both have to do with software performance under various conditions; briefly describe the difference between these two software testing goals.

Software reliability is the ability of a program to perform correctly under certain conditions for a specific period of time.

Software survivability is the ability of a program to continue functioning in any situation at any point in time.

In our study of fault injection, we discussed three different threats to software survivability. What were these three threats, and why is it important to consider each of them separately when testing software survivability?

The three threats to software survivability are: software flaws, malicious attacks, and anomalous behavior of third-party software. Software flaws in program source code are faults that have been inadvertently introduced to the system by the programmer. Malicious attacks exploit security flaws in the software – these are intelligent, purposeful actions performed by a user or other third party to serve some ultimate goal other than what was originally intended by the programmers. Anomalous behaviour from third-party software encompasses component failure and the fact that commercial off-the-shelf (COTS) applications may be flawed. These threats to survivability come from very different sources and manifest in software performance in a variety of ways. Because of these differences, it is important to consider each type of threat when testing software survivability; the same methods will not uncover faults that are vulnerable to separate threats.