



## Crash test your application



Image courtesy: theinspirationroom.com

Sometime ago I [wrote](#) about a quick way to shutdown a hanged application. But, in fact, shutting down a working application could help you exposing certain types of resource management defects. This is how ends become means.

Sample tests, suggested below, were tried on different Windows Desktop applications, sometimes helping to reveal serious issues.

### Local Tests

- Start an application. Begin working process then shutdown the application in the middle of it. Start the application. See if it reports than the previous session wasn't properly finished (and does it suggest something about it, like "start in safe mode").
- Start an application. Begin working process then shutdown the application in the middle of it. Start the application. See if it can access any previously opened files, and how is it handling and reporting the issues.
- Change settings in an application, apply changes. Shutdown the application. If the application saves settings only at exit point, all changes will be lost.
- Change settings in an application, apply changes and shutdown the application simultaneously. Start the application. See if it can access configuration files, does it report that they might be corrupted, does it suggest repairing them? (you can also try corrupting the files manually).
- If an application allows running 2 copies: open 2, then shutdown one of them. Observe behavior of the remaining one.

### Server Tests

- Start a client. Shutdown a client. Try to login as the same user: a) immediately b) after a timeout given.
- Start a client. Shutdown a client. Repeat this process (you may want to automate that) while monitoring how Server side of an application allocates and releases (does it?) RAM and hard-drive resources.
- Start a client. Shutdown a client. Repeat this process (you may want to automate that) for a while - you may see errors like "max number of connections exceeded", etc.

Let us know results of your exploration.

Filed under: [Heuristics](#), [Testing Techniques](#)

[Leave a comment](#)

Comments (0)

Trackbacks (0)

( subscribe to comments on this post )

No comments yet.

### Categories

- [Accessibility](#) (2)
- [Automation](#) (17)
- [Availability](#) (1)
- [Bias](#) (7)
- [Bug Reports](#) (18)
- [Career Tips](#) (6)
- [Compliance](#) (1)
- [Database](#) (3)
- [Disaster Recovery](#) (2)
- [Documentation](#) (14)
- [Estimation](#) (1)
- [Exploratory Testing](#) (24)
- [Free Tools](#) (68)
- [Heuristics](#) (47)
- [Internationalization](#) (4)
- [Learning about the product](#) (7)
- [Mind Mapping](#) (9)
- [Performance Testing](#) (33)
- [Playing Well With Others](#) (39)
- [Practicing Testing](#) (8)
- [Regression Testing](#) (2)
- [Security Testing](#) (13)
- [SEO](#) (1)
- [Skilled Bug Investigation](#) (13)
- [Stress Testing](#) (1)
- [Technical Tricks](#) (1)
- [Test Data](#) (15)
- [Test Management](#) (33)
- [Test Oracles](#) (2)
- [Test Planning](#) (31)
- [Test Theory](#) (4)
- [Testability](#) (1)
- [Testing mobile](#) (6)
- [Testing Techniques](#) (14)
- [Time Savers](#) (27)
- [Tools under \\$100](#) (9)
- [Uncategorized](#) (2)
- [Unit Testing](#) (2)
- [Usability](#) (12)
- [Web Testing](#) (27)

### Authors

- [Albert Gareev's website](#)
- [Anne-Marie Charrett's blog](#)
- [Jonathan Kohl's blog](#)
- [Julian Harty's website](#)
- [Karen Johnson's blog](#)

## Leave a comment

Name (required)

Mail (will not be published) (required)

Website

**Submit**

[« Unlearning for testers](#)

[Get yourself out of Inbox »](#)

[Mike Kelly's blog](#)

### Pages

[About](#)