

# *Black Box Software Testing*

## *(Academic Course - Fall 2001)*

**Cem Kaner, J.D., Ph.D.**

Florida Institute of Technology

**Section: 16 :**

**Stress Testing Paradigm**

Contact Information:

**kaner@kaner.com**

**www.kaner.com (testing practitioners)**

**www.badsoftware.com (software law)**

**www.testingeducation.org (education research)**

Copyright (c) Cem Kaner 2001.

I grant permission to make digital or hard copies of this work for personal or classroom use, without fee, provided that (a) copies are not made or distributed for profit or commercial advantage, (b) copies bear this notice and full citation on the first page, and if you distribute the work in portions, the notice and citation must appear on the first page of each portion. Abstracting with credit is permitted. The proper citation for this work is Cem Kaner, *A Course in Black Box Software Testing (Academic Version)*, Fall 2001, [www.testing-education.org](http://www.testing-education.org). To copy otherwise, to republish or post on servers, or to distribute to lists requires prior specific permission and a fee. Request permission to republish from kaner@kaner.com.

Copyright (c) Cem Kaner 2001

# *Black Box Software Testing*

## **Stress Testing Paradigm**

Assigned Reading:

Schneier, *Cryptogram*

Whittaker & Jorgenson, *How to Break Software*

# *Stress Testing*

- **Tag line**
  - “Overwhelm the product.”
- **Fundamental question or goal**
  - Learn about the capabilities and weaknesses of the product by driving it through failure and beyond. What does failure at extremes tell us about changes needed in the program’s handling of normal cases?
- **Paradigmatic case(s)**
  - Buffer overflow bugs
  - High volumes of data, device connections, long transaction chains
  - Low memory conditions, device failures, viruses, other crises.
- **Strengths**
  - Expose weaknesses that will arise in the field.
  - Expose security risks.
- **Blind spots**
  - Weaknesses that are not made more visible by stress.