

# *Black Box Software Testing*

## *(Academic Course - Fall 2001)*

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

Florida Institute of Technology

**Section: 30:**

Status Reporting

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](mailto:kaner@kaner.com).

# *Project Status*

- Bug Statistics are Only One Part of Status.
  - Product status reporting is a long-term opportunity to communicate project information to a range of managers.
  - Put the key issues that you want people to see on page 1. People will flip a page to get to your statistics.

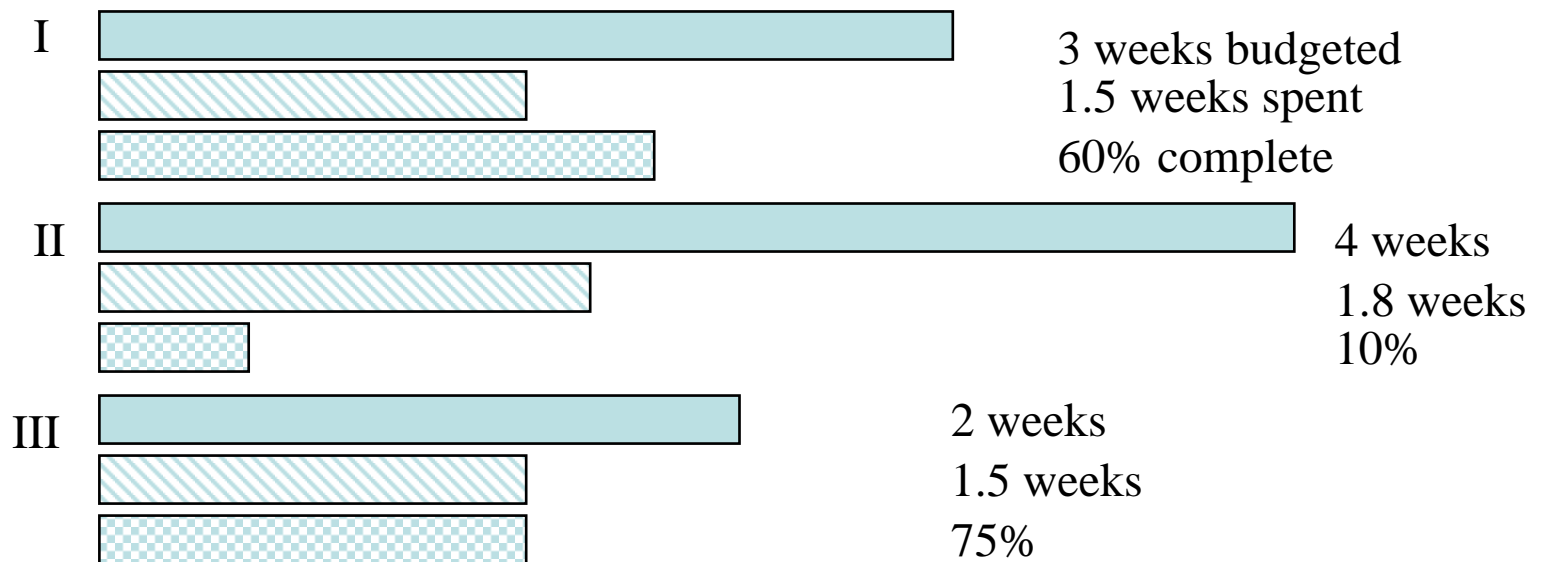
# Project Status

- **My weekly status reports look like this:**
- **A. Key Issues**
  - Deliveries needed
    - What promises are still open
    - What documents are still needed
    - What functions are still uncoded
    - What tools are not yet working
  - Decisions needed
  - Bug fixes needed
  - Unexpected problems

# Project Status

- B. Progress Against Plan
  - Milestones
  - Weekly goals
  - Percent complete
- C. Bug Numbers
- D. List of Bugs Not Fixed

# Project Status



# Project Status

- Bug Numbers
- I. Bug Curves (weekly progress)
  - Total open
  - New open
  - New closed/fixed
  - New closed/not-fixed
- II. Weekly Table
  - Show this week and last week:
  - New bugs this week
  - New closed
  - New closed/fixed
  - New closed/not-fixed
  - New resolved (not yet fixed)
  - Total resolved

# Project Status

- More Bug Numbers
- III. Another Weekly Table
  - Show last week and this week
  - Total open (x severity)
  - Total closed fixed (x severity)
  - Total closed not-fixed (x severity)
  - New open (x severity)
  - New closed fixed (x severity)
  - New closed not-fixed (x severity)

# Project Status

This is an example of the type of chart that you might create. This one can be used to flag a serious mismatch of opinion between testers (who assign severity) and programmers (who assign priority).

	Priority				
		Urgent	Important	Want to fix	Can defer
Severity	Awful				
	Bad				
	Pretty bad				
	Needs improvement				
	Minor				