



Principles of Software Testing for Testers

Module 2: Core Concepts of Software Testing

Objectives

- ◆ Introduce foundation topics of functional testing
- ◆ Provide stakeholder-centric visions of quality and defect
- ◆ Explain test ideas
- ◆ Introduce test matrices

Module 2 Content Outline

→ Definitions

- Defining functional testing
 - Definitions of quality
 - A pragmatic definition of defect
 - Dimensions of quality
- ◆ Test ideas
 - ◆ Test idea catalogs
 - ◆ Test matrices

Functional Testing

- ◆ In this course, we adopt a common, broad current meaning for functional testing. It is
 - Black box
 - Interested in any externally visible or measurable attributes of the software other than performance.
- ◆ In functional testing, we think of the program as a collection of functions
 - We test it in terms of its inputs and outputs.

Discussion Exercise 2.1: Define *Quality*

- ◆ Form pairs
- ◆ Define *quality*
 - Write the definition down
 - Is this your company's definition?
 - Is this your partner's company's definition?
 - Does your partner agree with his/her corporate definition of quality?

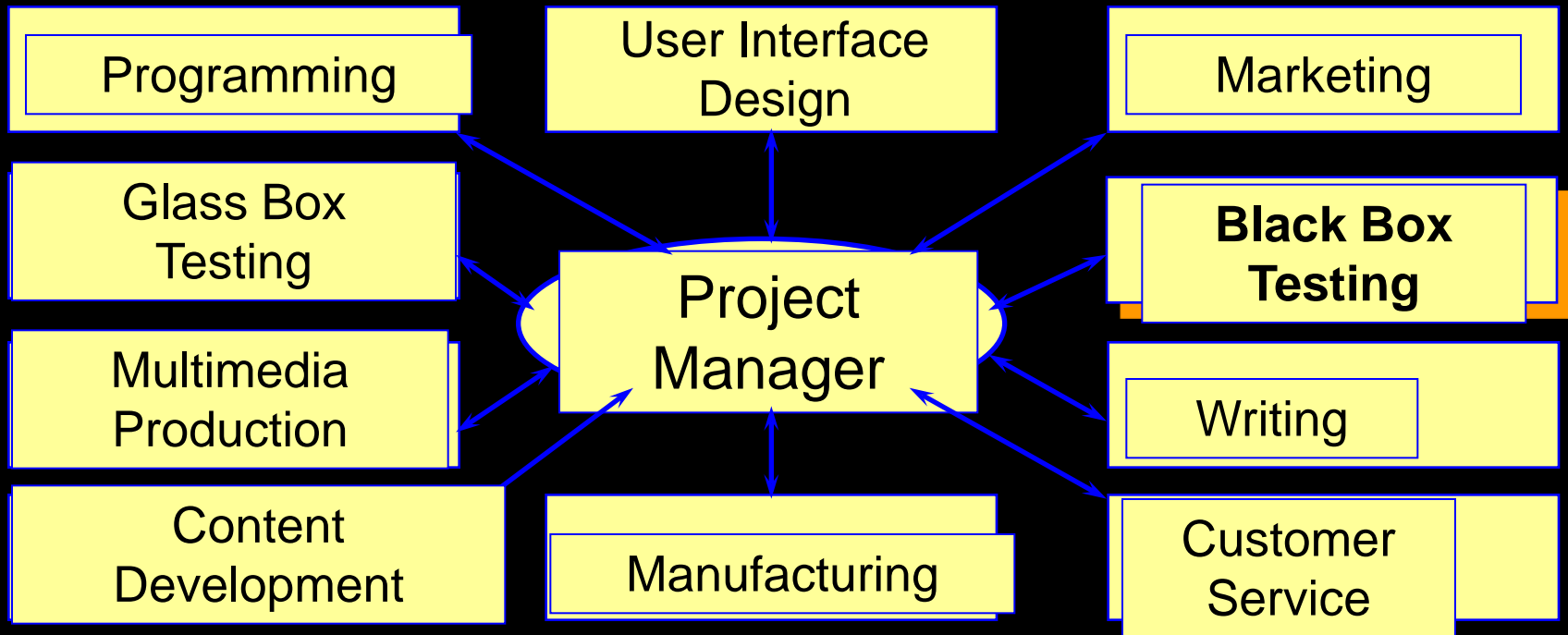
How Some Experts Have Defined Quality

- Fitness for use (Dr. Joseph M. Juran)
 - The totality of features and characteristics of a product that bear on its ability to satisfy a given need (American Society for Quality)
 - Conformance with requirements (Philip Crosby)
 - The total composite product and service characteristics of marketing, engineering, manufacturing and maintenance through which the product and service in use will meet expectations of the customer (Armand V. Feigenbaum)
- ◆ Note absence of “conforms to specifications.”

Quality As Satisfiers and Dissatisfiers

- ◆ Joseph Juran distinguishes between Customer *Satisfiers* and *Dissatisfiers* as key dimensions of quality:
 - Customer Satisfiers
 - the right features
 - adequate instruction
 - Dissatisfiers
 - unreliable
 - hard to use
 - too slow
 - incompatible with the customer's equipment

Quality Involves Many Stakeholders



- ◆ In a project team meeting, each person in the room has a different vision of what a “quality” product would be. Fixing defects is just one issue.

Exercise 2.2: Quality Has Many Stakeholders (1/2)

- ◆ A programming group agrees on variable naming conventions and follows them. Then they hire a new senior programmer, who won't follow the conventions. Programming the way he did in the 1970's, he gives variables names like Mabel and Al. Some of the other programmers are unhappy about this and so they want to enter a bug report into the bug tracking system, saying that "Mabel is not a proper variable name."
- ◆ **Question:** Does this report belong in the change request system? Why or why not?

Exercise 2.2: Quality Has Many Stakeholders (2/2)

- ◆ A company is developing a product that they will release in English first. The company expects to create other language versions by changing strings and other resource variables from the English version. They plan to localize versions without recompiling or relinking the code. However, the English code has a hyphenation error that doesn't affect English words but will mishandle some German words.
- ◆ The localization manager wants this filed as a bug against the English base code. Fixing the problem in the German version would require a code change and recompilation. The American project manager says that it is not a bug because the English version works OK. Who is right? Should the change request go with the English product, the German product, or both?

A Working Definition of Quality

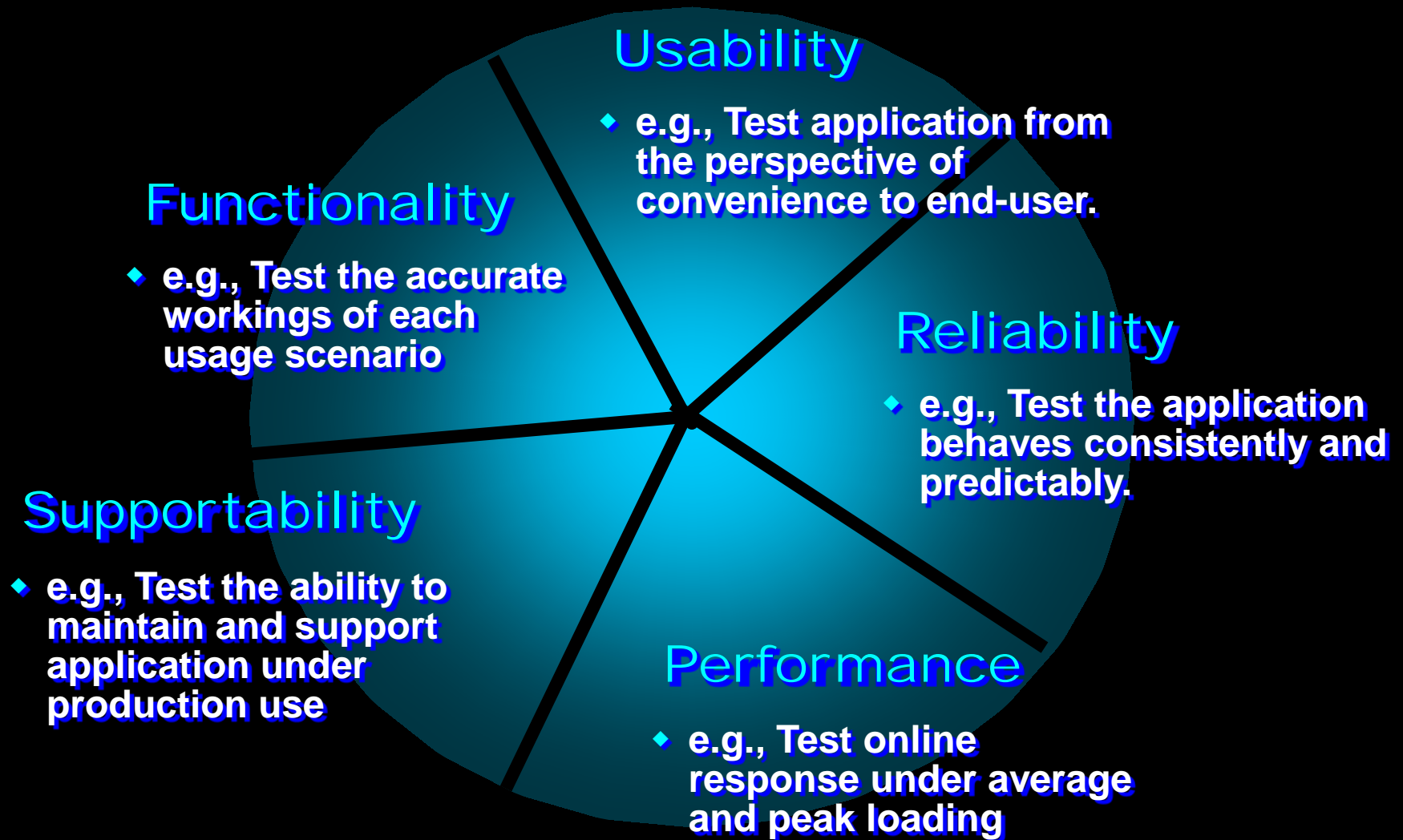
Quality is value to some person.

---- Gerald M. Weinberg

Change Requests and Quality

- ◆ A “defect” – in the eyes of a project stakeholder– can include anything about the program that causes the program to have lower value.
- ◆ It’s appropriate to report any aspect of the software that, in your opinion (or in the opinion of a stakeholder whose interests you advocate) causes the program to have lower value.

Dimensions of Quality: FURPS



A Broader Definition of Dimensions of Quality

- ◆ Accessibility
- ◆ Capability
- ◆ Compatibility
- ◆ Concurrency
- ◆ Conformance to standards
- ◆ Efficiency
- ◆ Installability and uninstallability
- ◆ Localizability
- ◆ Maintainability
- ◆ Performance
- ◆ Portability
- ◆ Reliability
- ◆ Scalability
- ◆ Security
- ◆ Supportability
- ◆ Testability
- ◆ Usability

Collectively, these are often called *Qualities of Service*, *Nonfunctional Requirements*, *Attributes*, or simply *the -ilities*

Module 2 Content Outline

- ◆ Definitions
 - Defining functional testing
 - Definitions of quality
 - A pragmatic definition of defect
 - Dimensions of quality

→ Test ideas

- ◆ Test idea catalogs
- ◆ Test matrices

Test Ideas

- ◆ A test idea is a brief statement that identifies a test that might be useful.
- ◆ A test idea differs from a test case, in that the test idea contains no specification of the test workings, only the essence of the idea behind the test.
- ◆ Test ideas are generators for test cases: potential test cases are derived from a test ideas list.
- ◆ A key question for the tester or test analyst is which ones are the ones worth trying.

Exercise 2.3: Brainstorm Test Ideas (1/2)

- ◆ We're about to brainstorm, so let's review...
- ◆ **Ground Rules for Brainstorming**
 - The goal is to get lots of ideas. You brainstorm together to discover categories of possible tests—good ideas that you can refine later.
 - There are more great ideas out there than you think.
 - Don't criticize others' contributions.
 - Jokes are OK, and are often valuable.
 - Work later, alone or in a much smaller group, to eliminate redundancy, cut bad ideas, and refine and optimize the specific tests.
 - Often, these meetings have a facilitator (who runs the meeting) and a recorder (who writes good stuff onto flipcharts). These two keep their opinions to themselves.

Exercise 2.3: Brainstorm Test Ideas (2/2)

- ◆ A field can accept integer values between 20 and 50.
- ◆ What tests should you try?

A Test Ideas List for Integer-Input Tests

- ◆ Common answers to the exercise would include:

Test	Why it's interesting	Expected result
20	Smallest valid value	Accepts it
19	Smallest -1	Reject, error msg
0	0 is always interesting	Reject, error msg
Blank	Empty field, what's it do?	Reject? Ignore?
49	Valid value	Accepts it
50	Largest valid value	Accepts it
51	Largest +1	Reject, error msg
-1	Negative number	Reject, error msg
4294967296	2^{32} , overflow integer?	Reject, error msg

Discussion 2.4: Where Do Test Ideas Come From?

- ◆ Where would you derive Test Ideas Lists?
 - Models
 - Specifications
 - Customer complaints
 - Brainstorm sessions among colleagues
- ◆ How do you create them in your company?

Module 2 Content Outline

- ◆ Definitions
 - Defining functional testing
 - Definitions of quality
 - A pragmatic definition of defect
 - Dimensions of quality
- ◆ Test ideas
 - ➔ **Test idea catalogs**
 - **Test matrices**

A Catalog of Test Ideas for Integer-Input tests

- ◆ Nothing
- ◆ Valid value
- ◆ At LB of value
- ◆ At UB of value
- ◆ At LB of value - 1
- ◆ At UB of value + 1
- ◆ Outside of LB of value
- ◆ Outside of UB of value
- ◆ 0
- ◆ Negative
- ◆ At LB number of digits or chars
- ◆ At UB number of digits or chars
- ◆ Empty field (clear the default value)
- ◆ Outside of UB number of digits or chars
- ◆ Non-digits
- ◆ Wrong data type (e.g. decimal into integer)
- ◆ Expressions
- ◆ Space
- ◆ Non-printing char (e.g., Ctrl+char)
- ◆ DOS filename reserved chars (e.g., "\ * . :")
- ◆ Upper ASCII (128-254)
- ◆ Upper case chars
- ◆ Lower case chars
- ◆ Modifiers (e.g., Ctrl, Alt, Shift-Ctrl, etc.)
- ◆ Function key (F2, F3, F4, etc.)

The Test-Ideas Catalog

- ◆ A test-ideas catalog is a list of related test ideas that are usable under many circumstances.
 - For example, the test ideas for numeric input fields can be catalogued together and used for any numeric input field.
- ◆ In many situations, these catalogs are sufficient test documentation. That is, an experienced tester can often proceed with testing directly from these without creating documented test cases.

Apply a Test Ideas Catalog Using a Test Matrix

	Lower Bound	LB-1	Upper Bound	UB+1	Zero	Spaces	Nothing	Negative
<i>Field name</i>								
<i>Field name</i>								
<i>Field name</i>								

Exercise 2.5: Your Own Test Ideas Lists

- ◆ **Now (in class)**
 - Pick a topic of interest for a test ideas list
- ◆ **Homework**
 - Expand into test ideas list
 - Tomorrow we will discuss this
- ◆ **Homework follow-up in class**
 - Develop a matrix from these ideas

Review: Core Concepts of Software Testing

- ◆ What is Quality?
- ◆ Who are the Stakeholders?
- ◆ What is a Defect?
- ◆ What are Dimensions of Quality?
- ◆ What are Test Ideas?
- ◆ Where are Test Ideas useful?
- ◆ Give some examples of a Test Ideas.
- ◆ Explain how a catalog of Test Ideas could be applied to a Test Matrix.