

CS577b Spring 2001

February 26, 2001

Software Quality Management – 3

A. Winsor Brown

Goals of Presentation

You should get a brief overview of the "software testing" area

- Types (domain or level) of testing
- Testing techniques
- Organizational approaches and implications for testing
- Testing in CS577ab
- CeBASE and software testing

Invite you to participate in a some "stress testing" and learn about defect reduction

CeBASE eWorkshop Stress Test

Tuesday, February 27, 12:00 to 1:00PM

Stress-test goal: have many users access the tool simultaneously, thus "stressing" the system.

CeBASE eWorkshop

- Using the PortaChat tool
- Connect to <http://cebase.org/chat3/>
- Participate in the discussion and "stress test" exercises
 - Preparation: Read "Software Defect Reduction Top-10 List" available at <http://cebase.org/defectreduction/top10/>
 - Be on for most of the hour; participate in the "stress test" [or inform why unable to participate]

WIIFM: Maximum credit replaces your lowest quiz score.

Outline

CeBASE and software testing

Types (domain or level) of testing

Testing techniques

Organizational approaches & implications for testing

Testing in CS577ab

CeBASE and Software Testing

The "Software Defect Reduction Top-10 List" mentions testing only in passing. Why?

The CeBASE "roadshow" presentation says:
Examples of Useful Empirical Results
Technique Selection Guidance

“Under specified conditions, ...”

- Peer reviews are more effective than functional testing for faults of omission and incorrect specification (UMD, USC)
- Functional testing is more effective than reviews for faults concerning numerical approximations and control flow (UMD, USC)

Outline

CeBASE And Software Testing

Types (Domain Or Level) Of Testing

Testing Techniques

Testing: Organizational Approaches & Implications

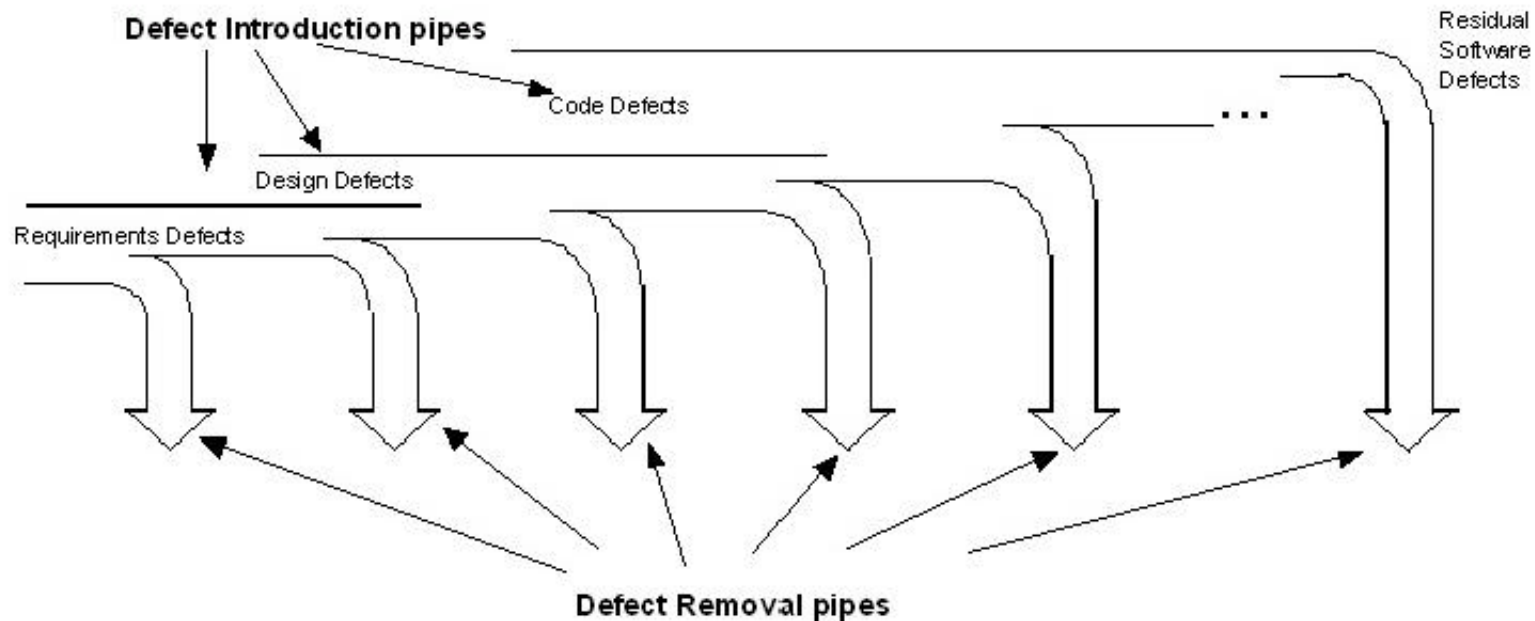
Testing In CS577ab

Testing in Perspective

Testing is one of many Quality Assessment techniques
[AKA defect detection approaches]

Other approaches identified in the COQUALMO model:

- Automated Analysis
- People Reviews
- Execution Testing and Tools



Testing vs. Debugging

Debugging – the act of attempting to determine the cause of the symptoms of malfunctions detected by testing or by [frenzied] user complaints¹

Testing – the process of establishing confidence that a program or system does what it is supposed to²

Testing – the process of *executing a program* or system with the intent of finding errors³

Testing – *any* [authors opinion] activity aimed at evaluating an attribute or capability of a program or system and determining that it meets it required results⁴

¹ Beizer, Boris; Software Testing Techniques 2nd Edition, 1990, Van Nostrand Reinhold

² Hetzel, William; Program Test Methods, 1973, Prentice Hall

³ Myers, Glen; The Art of Software Testing 1979

⁴ Hetzel, Bill; The Complete Guide to Software Testing 2nd Edition, 1988, QED Information Sciences, Inc.

Testing vs. Debugging (cont.)

The last definition "includes" people reviews; MBASE keeps them separate.

The *purpose of testing* is to show that a program has bugs⁵.

The *purpose of debugging* is to find the error or misconception that lead to the program's failure and to design and implement the program changes that correct the error⁶.

NOTE: "designing and implementing the program changes" is sometimes referred to as the "hidden factory"

5

Hetzel, Bill; [The Complete Guide to Software Testing](#) 2nd Edition, 1988, QED Information Sciences, Inc.

6

Hetzel, Bill; [The Complete Guide to Software Testing](#) 2nd Edition, 1988, QED Information Sciences, Inc.

Types of Testing

Formal vs. Informal:

- **Formal:** planned tests, cases & harnesses retained, results tracked; "Process of conducting testing activities and reporting test results in accordance with an approved test plan."⁷
- **Informal:** cases & harnesses not retained; results not tracked

Levels of testing

- Unit testing
- Integration testing
- System testing
- Acceptance testing

Not really "testing" [per MBASE], but often included as part of the "discipline": Inspection, Walkthrough, Audits, Desk checking, ...

⁷ Hetzel, Bill; The Complete Guide to Software Testing 2nd Edition, 1988, QED Information Sciences, Inc.
© 2001 USC Center for Software Engineering

Typical Unit Testing Practices

- Objective:* **Confirm that module is coded correctly**
- Who does it:* **Usually the programmer**
- What is tested:* **Functions are exercised (black box). Code may be exercised (white or clear box). Extremes and boundaries are explored.**
- When complete:* **Usually when programmer feels comfortable and has no known defects**
- Tools or aids:* **Not commonly employed**
- Records:* **Usually not recorded**

Integration and Other Testing Definitions

Integration testing – An orderly progression of testing in which software and/or hardware elements are combined and tested until the entire system has been integrated.

Regression testing – Selective testing to verify that modifications have not caused unintended adverse side effects or to verify that a modified system still meets requirements.

Functional testing – Process of testing to verify that the functions of a system are present as specified.

Operational testing – Testing by the end user on software in its normal operating environment (DoD)

Typical System Testing Practices

- Objective:* **Assemble modules into a working system. Determine readiness for acceptance test.**
- Who does it:* **Team leader or test group**
- What is tested:* **System requirements and functions; system interfaces**
- When complete:* **Usually when majority of requirements met, no major defects remaining**
- Tools or aids:* **Library system and test case library. Test data generators, comparators, and simulators.**
- Records:* **Defects discovered usually logged; test cases maintained**

Typical Acceptance Testing Practices

- Objective:* **Evaluate readiness for use**
- Who does it:* **End user or his agent**
- What is tested:* **Major functions, documentation, and procedures**
- When complete:* **Usually when user is comfortable or when test set runs successfully**
- Tools or aids:* **Comparators**
- Records:* **Spotty; some highly formalized**

Outline

CeBASE and Software Testing

Types (Domain Or Level) Of Testing

Testing Techniques

Testing: Organizational Approaches & Implications

Testing In CS577ab

Testing Techniques

Path Testing – a family of tests based on testing selected paths in the control flow

Transaction-Flow Testing – testing based on a behavioral model that leads to functional testing

Data-Flow Testing – any testing strategy based on data flows [where a "flow" is an abstract representation of the possible changes of state (creation, usage, destruction) of data objects]

Domain Testing – testing based on the set defined over the components input space

Syntax Testing – testing based on validating system inputs

Logic-based Testing – testing of functional requirements that can be specified via decision tables

Transition Testing – testing transitions based on a finite-state machine model

Box Views

Black Box Testing – Testing based on a comparison of a component's actual behavior with its specified behavior with out regard to the component's implementation [AKA functional testing]⁸

White Box Testing – Testing aimed at exploring the consistency of a component's implementation with its design [AKA structural testing]⁹

8

Hetzel, Bill; The Complete Guide to Software Testing 2nd Edition, 1988, QED Information Sciences, Inc.

9

Beizer, Boris; Software Testing Techniques 2nd Edition, 1990, Van Nostrand Reinhold

Outline

CeBASE and Software Testing

Types (Domain Or Level) Of Testing

Testing Techniques

Testing: Organizational Approaches & Implications

Testing In CS577ab

Testing: Organizational Approaches & Implications

CM

- Applies to code and test cases
- Each must cross identify to the others

Separate vs. part of team

- Independence of test developers and executers
- Buddy testing: dependent

Small projects vs. Large

- Small: often use team members for test procedure and case development
- Large: usually use independent test teams

Testing: Organizational Implications (cont.)

Standards:

ISO 12207 – (ISO/IEC 12207) Standard for Information Technology—Software life cycle processes—Life cycle data

IEEE J-STD-016 1995 – Trial-Use Standard //
Standard for Information Technology //
Software Life Cycle Processes Software
Development Acquirer-Supplier Agreement

EIA-498 Mil-STD for Software Development &
Documentation



Outline

CeBASE and Software Testing

Types (Domain Or Level) Of Testing

Testing Techniques

Testing: Organizational Approaches & Implications

Testing In CS577ab

Testing In CS577ab

A few of many forms of quality "assessment"

CS577a: No testing? Debugging of functional prototype

CS577b: Planned, tracked and results reported

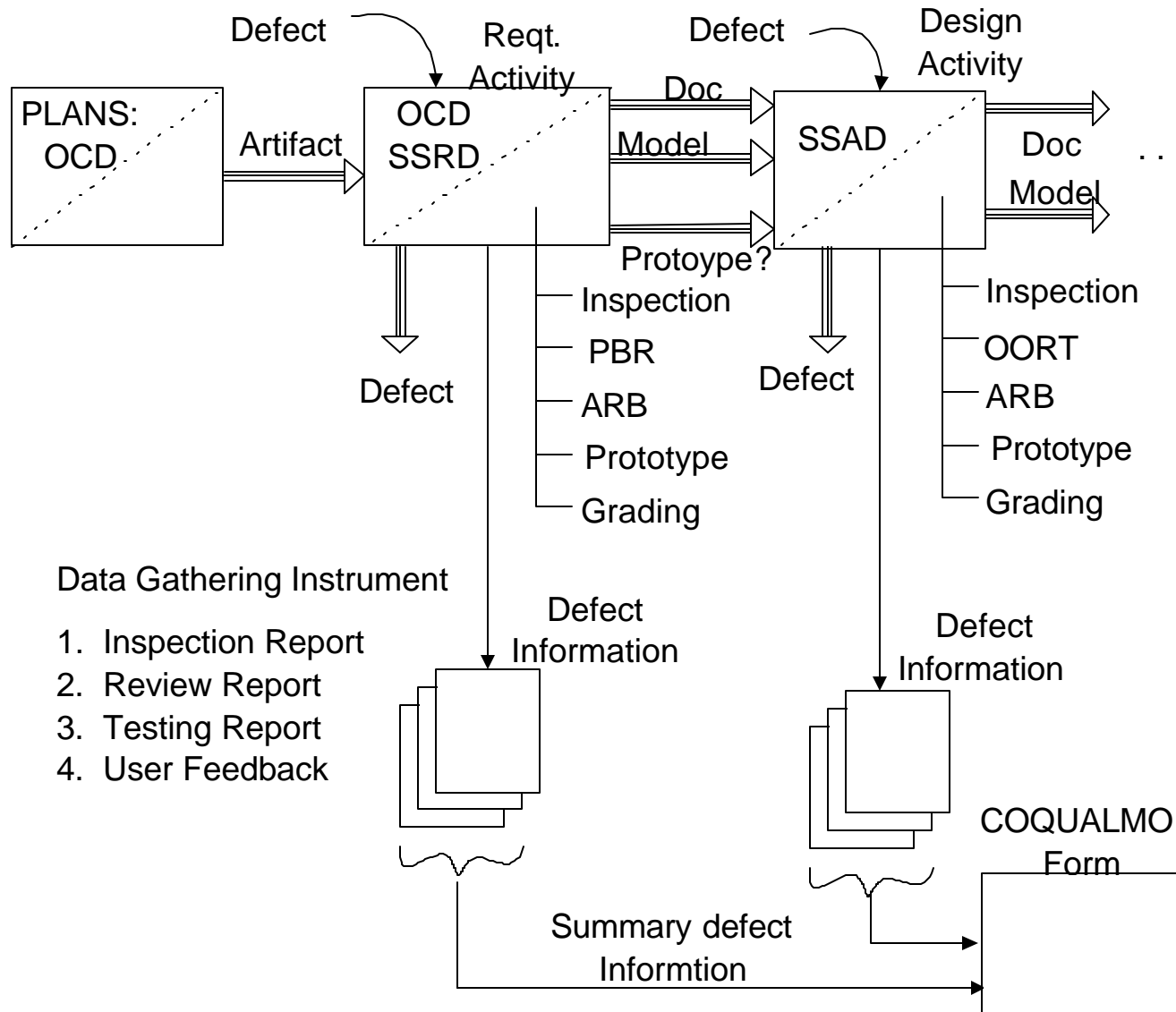
- Integration testing? [recommended]
- System testing
- Acceptance testing

**TeamTest: a Rational tool being installed in SAL200;
preferred way to develop & run system/acceptance
tests, and possibly the integration tests**

- Test case capture
- Test case executer
- Test result comparator

[Test] Defect tracking tool: GNATS

CS577 Model/Document Assessment



577B Activity Diagram

