



Improving Requirements Through Testing

Richard Bender
Bender RBT Inc.
17 Cardinale Lane
Queensbury, NY 12804
rbender@BenderRBT.com

Copyright 2003 Bender RBT Inc.

1

The Challenge

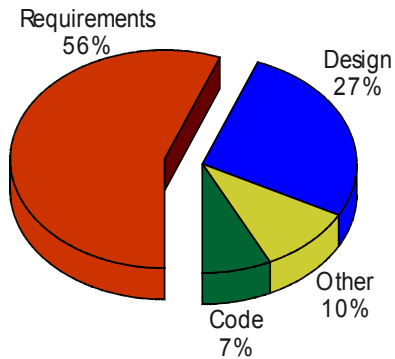


- Faster, Better, Cheaper
- Residual Defect Rates
 - Software: 5 to 7 defects per KLOC
 - Hardware circuits: far less than 1 per million gates

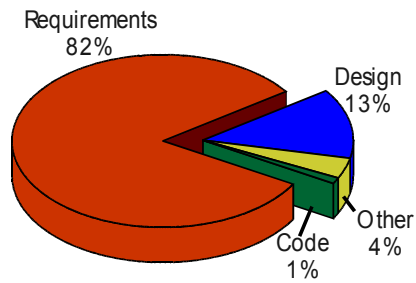
Copyright 2003 Bender RBT Inc.

2

Distribution of Bugs



Distribution of Effort to Fix Bugs



(James Martin)

Relative Cost to Fix an Error



<u>Phase in which found:</u>	<u>Cost Ratio:</u>
Requirements	1
Design	3-6
Coding	10
Development Testing	15-40
Acceptance Testing	30-70
Operation	40-1000

(IBM, GTE, et.al.)

The RBT Process



1. **Validate** requirements against objectives.
2. Apply scenarios against requirements.
3. Perform initial Ambiguity Review.
4. Perform domain expert reviews.
5. Create cause-effect graph.
6. Logical consistency check and test cases designed by BenderRBT.
7. Review test cases with requirements author.
8. **Validate** test cases with users/domain experts.
9. Review test cases with developers.
10. Walk test cases through design.
11. Walk test cases through code.
12. **Verify** code against test cases designed from the requirements.

Validate Requirements Against Objectives



Objectives define **WHY** the system is being created.

Fully Qualified Objectives:

Identify goals/desired return on investment.

Identify constraints - e.g., time, resources.

Example:

Reduce operational expenses by 10% by year end 2003 with project expenditures not to exceed \$2 million.

Validate Requirements Against Objectives



Objective:

Comply with a regulatory requirement to supply the government with requested data within 5 days of the request.

Delays are subject to fines.

Initial Solution:

A real-time, on-line, mighty-fine, database application costing

hundreds of thousands of dollars.

Alternative Solution:

Two part-time clerks.

Validate Requirements Against Objectives



- Ensures proper scope for high-level requirements.
- Ensures application rules stay focused.
- Critical for managing scope changes.
- This is an on-going activity against each iteration of the requirements [and design].

Apply Scenarios Against Use Cases / Requirements



- Scenario:
 - A “what if” action by the user
- Use Case:
 - A task oriented user’s view of the system.
 - A completed unit of work.
 - Users may be people or other systems outside the scope of this system.
- Goal:
 - Verify that the requirements are robust enough to handle all of the Use Cases.
 - Also important in verifying usability.

Apply Scenarios Against Requirements



Examples:

Add a New Hire:

- Requires one or more transactions for the payroll system.
- Requires one or more transactions for the personnel system.
- Requires one or more transactions for the hiring system.

Transfer an Employee:

- Requires one or more transactions for the payroll system.
- Requires one or more transactions for the personnel system.

*Each scenario requires one or more transactions.
Each transaction requires one or more functions.*

Ambiguity Reviews



- Testing, by definition, is comparing an expected result against an observed result.
- Rarely are specifications detailed/clear enough to predict the expected results.
- Result: most software testing is not true testing.

Ambiguity Reviews



- Requirements are written primarily in natural language.
- All natural languages are inherently ambiguous.
- Formal specification languages are not a viable alternative.
- Using structured, natural language helps considerably.
- Most developers do not know how to write detailed unambiguous requirements.

Ambiguity Reviews



One half of two and two = ??

Ambiguity Reviews

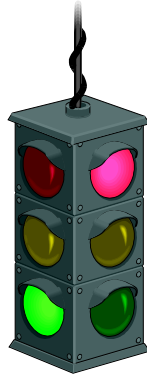


A difference between version I and version II exists
only when mixed data types are used, and then
only when operand lengths differ, and then only
sometimes.

Ambiguity Reviews



EXAMPLE:



If the light is **red**, then stop.

Ambiguity Reviews Dangling Else



MUST BE, WILL BE, IS ONE OF,
SHOULD BE,
COULD BE, CAN BE, SHALL.

EXAMPLE:

“The Marriage Status must be
either Married, Single, or
Divorced.”

Else? An error condition?

Ambiguity Reviews Ambiguity of Reference



IT, SUCH, THE ABOVE, THE PREVIOUS, THEM,
THESE, THEY

EXAMPLE 1:

“Add field A to field B.

This number must be positive.”

EXAMPLE 2:

“Transaction 1 displays the customer’s name and address.

Transaction 2 displays the customer’s account numbers.

Transaction 3 displays the customer’s account balances.

Such transactions require the security code.”

Ambiguity Review Checklist



- Dangling Else
- Ambiguity Of Reference
- Scope Of Action
- Omissions
 - Causes Without Effects
 - Missing Effects
 - Effects Without Causes
 - Complete Omissions
 - Missing Causes
- Ambiguous Logical Operators
 - Or, And, Nor, Nand
 - Implicit Connectors
 - Compound Operators
- Negation
 - Scope Of Negation
 - Unnecessary Negation
 - Double Negation
- Ambiguous Statements
 - Verbs, Adverbs, Adjectives
 - Variables, Unnecessary Aliases
- Random Organization
 - Mixed Causes And Effects
 - Random Case Sequence
- Built-In Assumptions
 - Functional/Environmental Knowledge
- Ambiguous Precedence Relationships
- Implicit Cases
- Etc.
- I.E. Versus E.G.
- Temporal Ambiguity
- Boundary Ambiguity

Additional Benefits from Ambiguity Reviews



- Timely feedback reduces issue resolution time.
- Explicit feedback leads to defect avoidance.
- Writing is accelerated.

Savings Via Ambiguity Reviews



Ambiguity Reviews of Specifications

(source: Bender RBT Inc)

- Defects found
 - Mutual Funds project
 - # Functions** 180
 - # Issues** 1,713
 - Embedded code project
 - # Functions** 65
 - # Issues** 595

Savings Via Ambiguity Reviews



Ambiguity Reviews of Specifications

(source: Bender RBT Inc.)

- Costs per defects found
 - .85 hour/defect
 - \$75 hour fully burdened rate (\$150K year)
 - \$63.75 per defect
- Costs if found in integration test/system test
 - \$750 to \$3,000 per defect (SEI)
- Cost if found in production
 - \$10,000 per defect (HP)
 - \$140,000 per defect (IBM)

Copyright 2003 Bender RBT Inc.

21

Cause-Effect Graphing

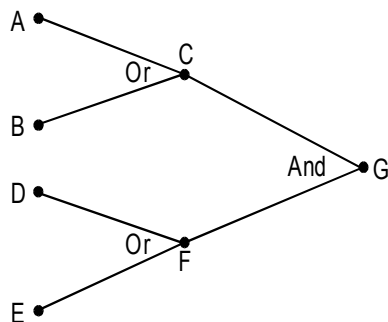


If A or B,
then C.

If D or E,
then F.

If C and F,
then G.

- Resolve Aliases
- Clarify Precedence Rules
- Clarifies Implicit Information
- Begin Integration Test



Copyright 2003 Bender RBT Inc.

22

Cause-Effect Graphing



- Superficially intimidating.
- You only need to know the definition of:
AND, OR, NOT.
- Forces the issue on the level of detail in specifications.
- Allows us to apply hardware logic testing algorithms to software testing.

Logical Consistency Check



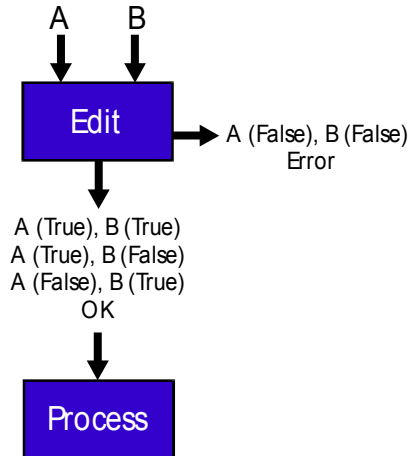
1. Causes of Environmental Constraints
 - a. Editing Rules
 - b. Physical Structure Of The Data
2. Boundary Constraints
 - a. Exclusive
 - b. One
 - c. Inclusive
 - d. Requires
 - e. Attribute Mask
 - f. Anchor
3. Processing Sequence Constraints
 - a. Termination Mask
 - b. Alternative Path Mask

Logical Consistency Check



1a. Editing Rules

Process Will Never Receive A (False), B (False) Case.



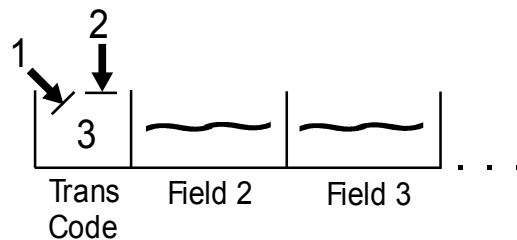
Logical Consistency Check



1b. Physical structure of the data.

The transaction code may be 1, 2, or 3.

The transaction code field is 1 byte.

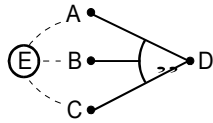


Cannot have more than one value per transaction code.

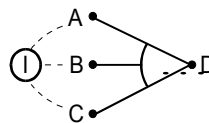
Logical Consistency Check



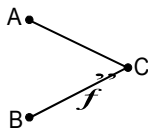
1.



2.



3.



A (T) requires B (T)

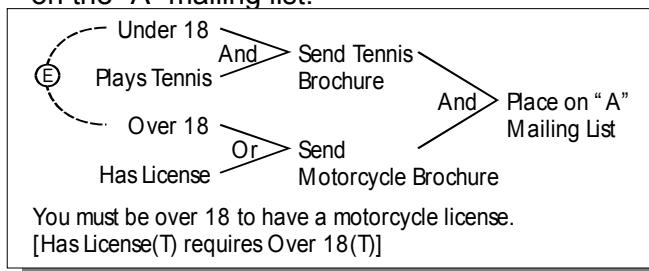
Logical Consistency Check



If the person is under 18, and plays tennis,
then send them a tennis club brochure.

If the person is 18 or older, or has a motorcycle license,
then send them a motorcycle club brochure.

If the person was sent both brochures, then put them
on the "A" mailing list.



Validate Test Cases



Dentists with membership codes of 2, 3, or 9 are member dentists.

For claims referencing a non-member dentist or for procedures not within the referenced dentist's record, a system table is used to calculate the amount paid. Otherwise the amount submitted is paid.

However, an override code of 1 or 9 allows the amount submitted to be paid for non-member dentists or for procedures not within the referenced dentist's record. When an override code is used an entry is made on the paid claims report.

Validate Test Cases



TEST 1

Cause States:
The Dentist is a Member Dentist
The procedure was not preauthorized
An override code was entered

Effect States:
Override the partial payment
Pay the full amount of the claim
Make an entry on the paid claims report

TEST 2

Cause States:
The Dentist is a Member Dentist
The procedure was preauthorized

Effect States:
Pay the full amount of the claim
Do not make an entry on the paid claims report

Validate Test Cases



Tester Reviews Test Cases Produced by BenderRBT

Verify tests match tester's understanding of the requirements.

Tester identifies errors and omissions.

Tests Reviewed With Requirements Author

Insures that tester and author are in sync.

Author identifies missing cases and corrects the requirements.

Validate Test Cases



Tests Validated by Users/Domain Experts

Users identify errors and omissions.

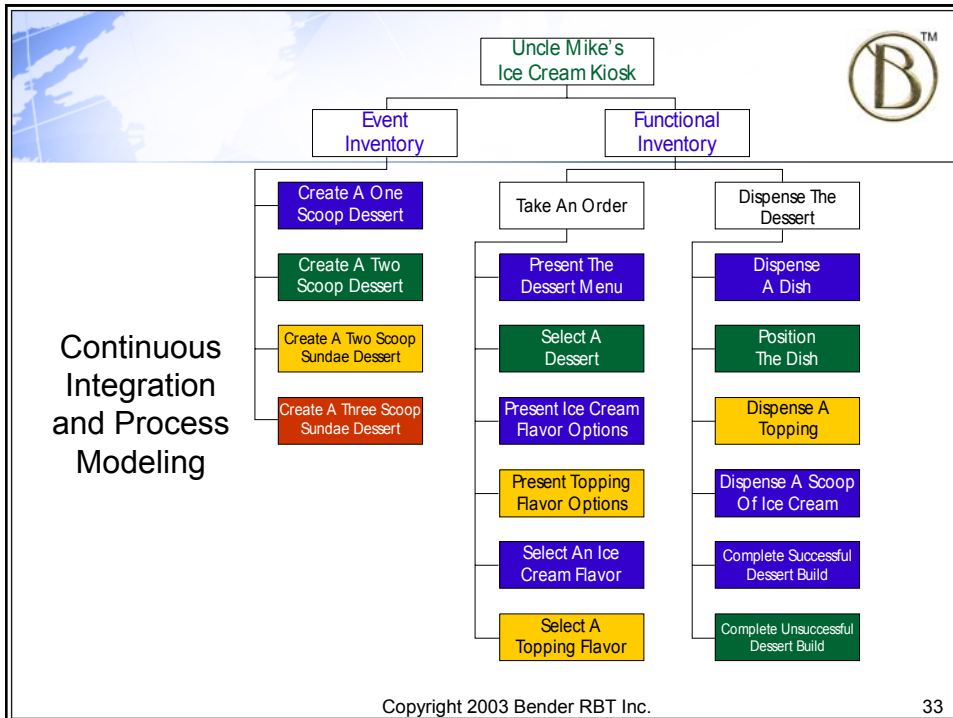
Tests Reviewed With Developers

Clarifies developers' understanding of the requirements.

Ensures design/code will match the requirements.

Note:

Sometimes the tests are the only version of the detailed rules available for review.



Continuous Integration and Process Modeling

- Perform Ambiguity Reviews of each process.
- Design Test Cases for each individual process.
- String the tests together into end-to-end test suites.
- Sequence the requirements writing to allow for early full thread test definitions.
- Sequence design and coding to continue to exploit early integration opportunities.

Copyright 2003 Bender RBT Inc. 34

Concluding Comments



- There is no way to significantly improve software quality or productivity without first addressing the quality of the requirements