An Overview of the Quality Process

As presented at

CSC 2009 CONNECT SEPTEMBER 13-17 • LAKE BUENA VISTA, FLORIDA SEIZE EVERY DAY

by

Peter Briggs Jr.
New Instruction LLC
615 Valley Road - Upper Montclair, NJ 07043
(973) 744-3339
www.newinstruction.com
pbriggs@newinstruction.com

Outline

Introduction: Defect Detection or Defect Prevention?

The objectives of testing are examined and the responsibilities of testers at all levels are defined. Quality concepts are introduced along with a list of success factors that will facilitate decision-making at each stage of the testing process. Numerous tests may be conducted during the SDLC and each is the primary responsibility of a specific group. As each group completes a testing phase a formal transition process is initiated before the next group begins. This section identifies entrance and exit criteria by phase for each component of the development team. The object is to eliminate as much testing overlap as possible.

- Objectives / Observations
- Impediments to Quality
- Roles/Responsibilities of the Tester
- Early Testing vs. Late Testing
- Quality Improvement Suggestions
- Quality Tools and Steps
- Opportunities to Improve the Testing Process
- System Development Life Cycles - Waterfall SDLC / Spiral SDLC / V-Model
- Phase Objectives
- Performance / Reliability Metrics
- Testing Success Factors
- Product Development and Testing Phases

Test Methodologies and Checklists

Testing methodologies enable testers to compute their test coverage and have confidence that all requirements will be tested. The use of methodologies in testing is an essential element of a quality assurance organization.

- Setting Test Objectives and Identifying Tests
- Test Planning
- Methodologies
- Test Coverage Computation
- Boundary Value Analysis
- Decision Tables
- Exploratory Testing
- Checklists – (Table and Array Testing, Date Edits, Screen, Button, and Character Entry Checklists)

Test Planning

Testing begins with a plan that unambiguously states the objectives. A suitable methodology is selected to provide adequate test coverage and to deliver the desired level of confidence that the software will perform as advertised. Testing is treated as a dynamic process that may continue after delivery and will certainly play a role in future system modifications. Appropriate record keeping is initiated and maintained through the life of the product.

- Unit Testing (Early Testing)
- White Box Test Case Sources
- Sample Unit Test Plan Table of Contents
- Unit Testing Scenario
- Integration Testing and System Testing
- System / Acceptance Testing
- Sample System (or Acceptance) Test Plan Table of Contents
- Sample System (or Acceptance) Test Script
- Possible Test Plan Elements
- Sample System (or Acceptance) Test Plan
- Creating the Regression Test / Regression Test Alternatives
- Traceability Matrix
- Usability Testing
- Test Notebook

Section 1

Defect Detection or Defect Prevention

Defect Detection

Finding Defects after they have been introduced into our applications.

Defect Prevention

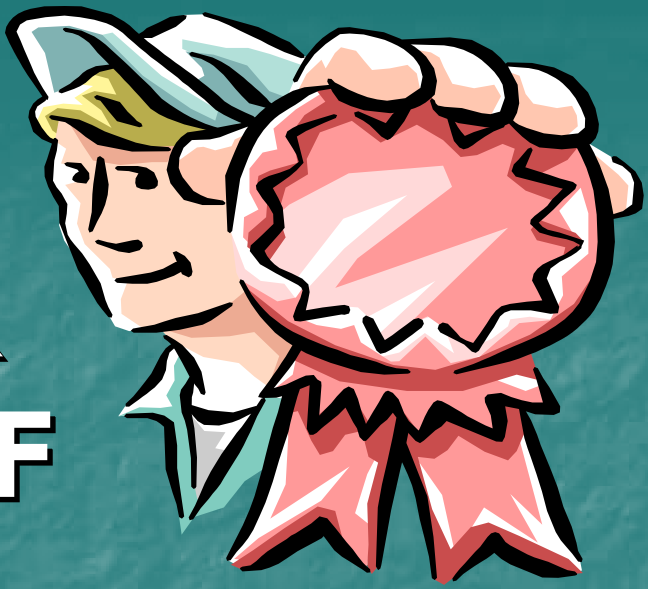
Preventing the defects from getting into our applications from the start.

WHAT IS YOUR PRIMARY RESPONSIBILITY?

Defect Detection

Defect Prevention

WHAT IS YOUR DEFINITION OF QUALITY?



WHAT DOES QUALITY MEAN TO YOU?

QUALITY QUESTIONS

1. What is quality?
2. What does quality cost?
3. How is quality measured?
4. Where does quality come from?
5. Can you test quality into your products?
6. How will we know that we have quality?
7. What projects have been quality efforts?
8. Who is responsible for quality?

QUALITY MESSAGE

To ensure that we are progressing set a low water mark and a high water mark for your application.

Where are we currently in this application and where would we like to be? This year? Next year? Five years?

GOALS

1. Automate the testing process
2. A structured review process
3. Structured development with reusable code and reusable tests
4. Measured progress and performance metrics
5. On-going quality initiatives

GOALS

6. Methods for identifying testable conditions
7. Organized testing process
8. Teamwork
9. Communications
10. Enjoy your job

OBSERVATIONS - 1

Nothing is obvious

- Specifications must be written
- Examples
- Graphics
- Quantify everything

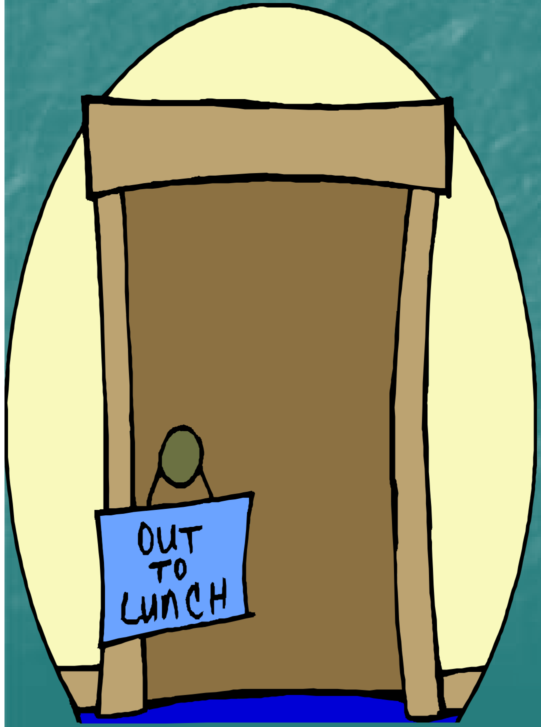
OBSERVATIONS - 2

Do it twice

- Test and retest
- Test initialization & re-initialization
- Design all tests to be repeatable
- Test bed should be maintainable

OBSERVATIONS - 3

Everything has a limit and they will be reached at the worst possible time.



- Identify the limits
- Test the limits, document the limits
- Language imposed limitations
- Platform imposed limitations
- Specification imposed limitations

OBSERVATIONS - 4

Design systems with testing in mind

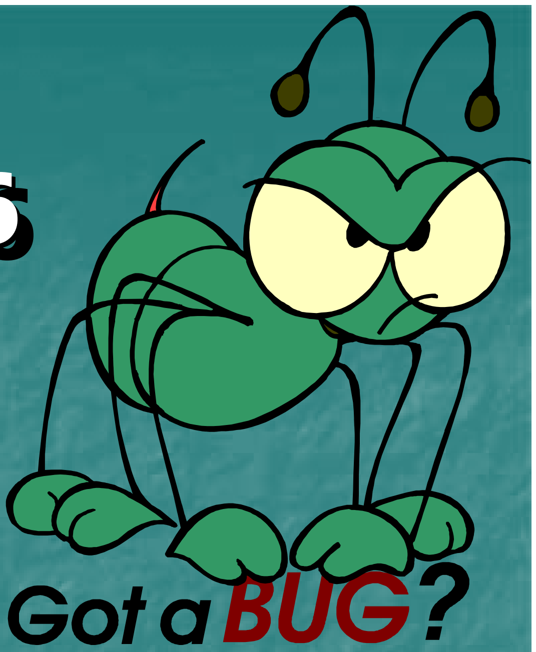
- Insert diagnostic tools (instrumentation)
- Control totals
- Audit trails
- Balancing routines
- File comparisons

OBSERVATIONS - 5

Practice tact and diplomacy

- Don't be critical all of the time
- Offer positive comments
- Encourage the right behavior
- It is better to find agreement than to win

OBSERVATIONS - 6

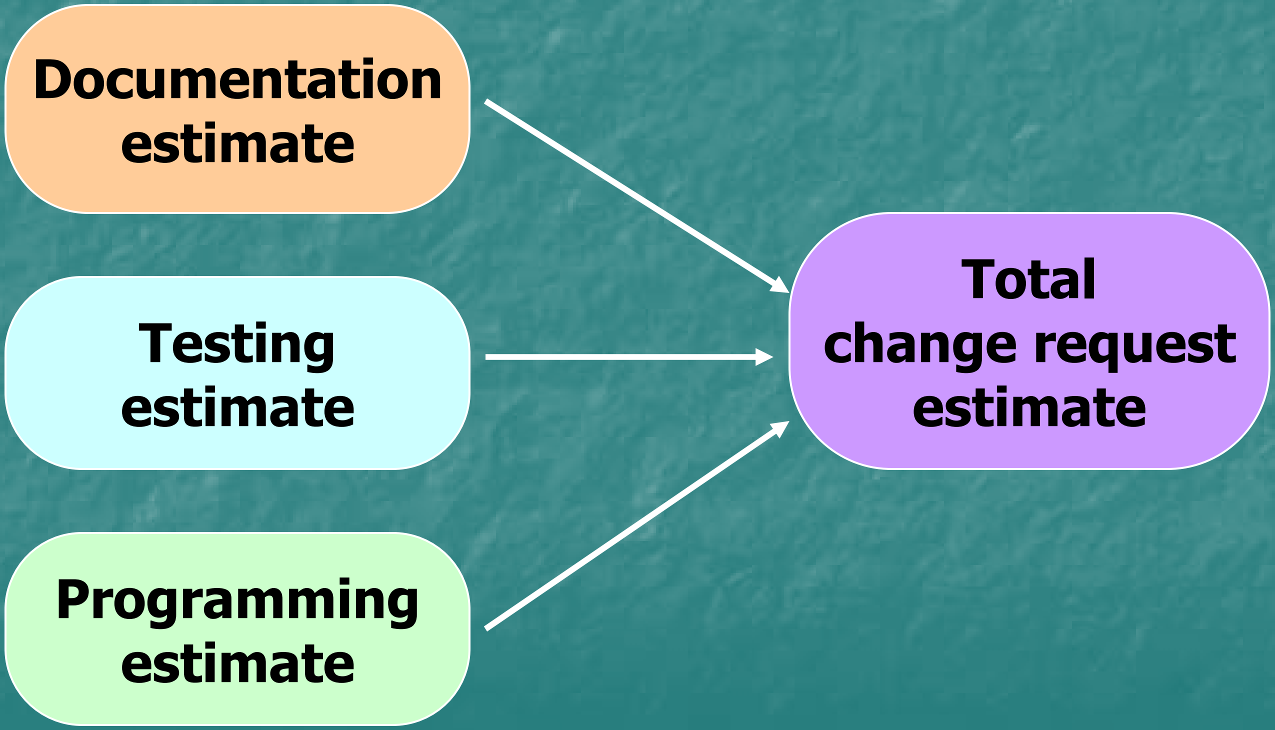


The specifications may be wrong

- All specifications will change
- Include test plans in the specs
- Identify a formal change process
- Don't fight change
- Get better estimates of change impact

CHANGE REQUESTS:

. . . must be in writing and require 3 estimates.



IMPEDIMENTS, OPPORTUNITIES, AND MANAGING

Inertia

- **"Things aren't so bad, why should I want to do anything differently. No one has yelled at me in over two weeks."**

IMPEDIMENTS, OPPORTUNITIES, AND MANAGING

No time

- **"We just don't have time to change the way we test or develop systems. Quality improvement is a great idea, but we don't have anyone available right now."**

IMPEDIMENTS, OPPORTUNITIES, AND MANAGING

Need a management buy-in

- **"If management doesn't tell us to improve quality, gives us time, and a budget, nothing is going to happen. Discussing quality with us is simply preaching to the choir."**

IMPEDIMENTS, OPPORTUNITIES, AND MANAGING

We're not ready and we need training

- **"We have to get the rest of the shop in order, before we can consider quality improvement suggestions. When will we have time for training?"**

IMPEDIMENTS, OPPORTUNITIES, AND MANAGING

Unrealistic Commitments

- **"Someone upstairs promised the clients that they would have it by next week. It HAS to go out by next week, no matter what."**

IMPEDIMENTS, OPPORTUNITIES, AND MANAGING

Lack of Domain Knowledge

- **"Everyone should just know how the business operates, that is part of your job. You should know everything about the business, after all you work here don't you."**

WHO IS RESPONSIBLE FOR TESTING?

- **Customer**
  They know what they want better than I do.
- **Business Analysts**
  They should do all of the testing. They're always talking to the customers.
- **Programmers**
  If they do their job properly, no one else has to test.
- **Quality Assurance**
  It's their job.

TESTER RESPONSIBILITIES

- Participation - DESIGN
- Review - SPECS
- Validation - REQUIREMENTS
- Verification – TEST PLANS
- Quality Control - DOCUMENTATION
- Reporting - STATUS

QUALITY IMPROVEMENT SUGGESTIONS 1

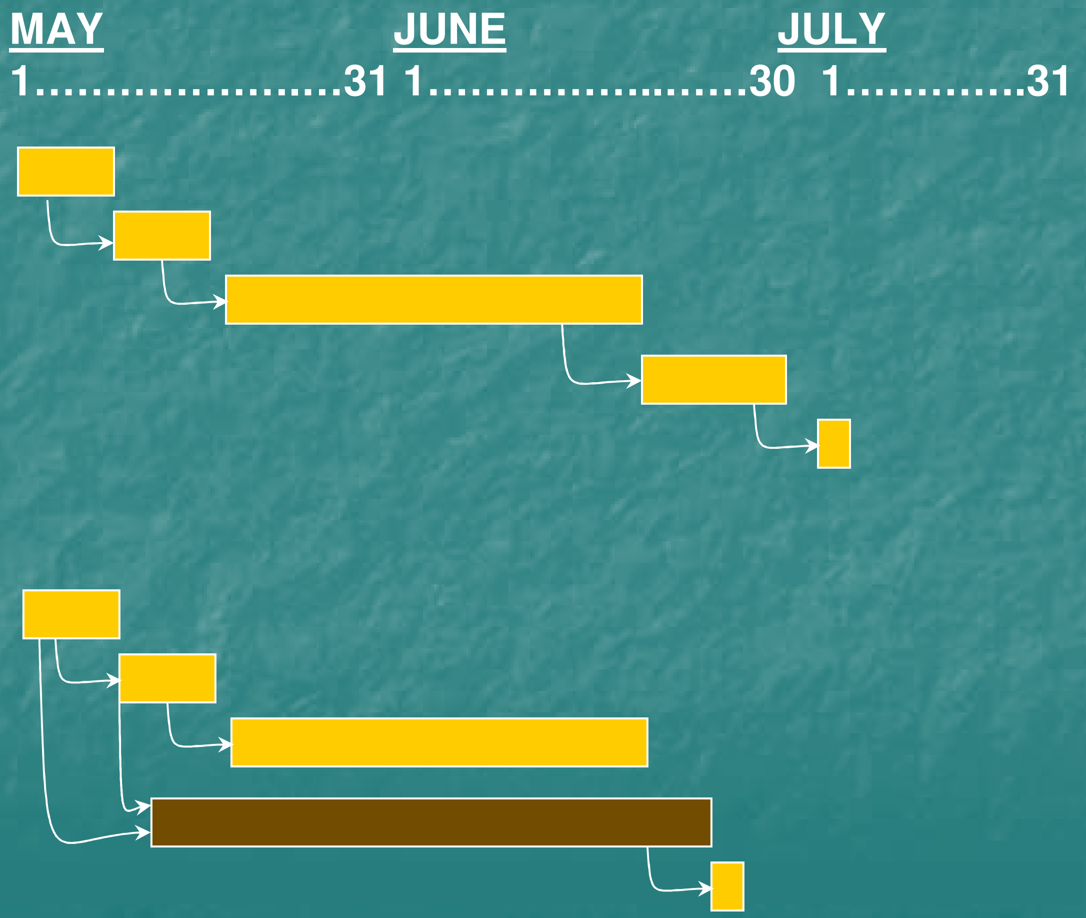
- Joint Application Design Sessions (JAD/JAR)
- Well Defined Business Objectives
- Insist On Written Specifications
- Use Prototyping Tools / Write User Manuals
- Written Unit/System Test Plans (*before coding*)
- Estimate The Coding/Testing Efforts First
- Perform Risk Analysis and Contingency Planning
- Assess The Corporate Readiness To Automate
- Allocate Resources To Automate Testing

QUALITY IMPROVEMENT SUGGESTIONS 2

- Earlier Involvement Of QA Personnel
- Promote Team Involvement (Including Users)
- Implement A Process For Improvement Ideas
- Teach Development Methodologies
- Enforce Coding/Testing Standards
- Capture And Report Metrics
- Instrumentation
- Structured Walkthroughs
- Version Control

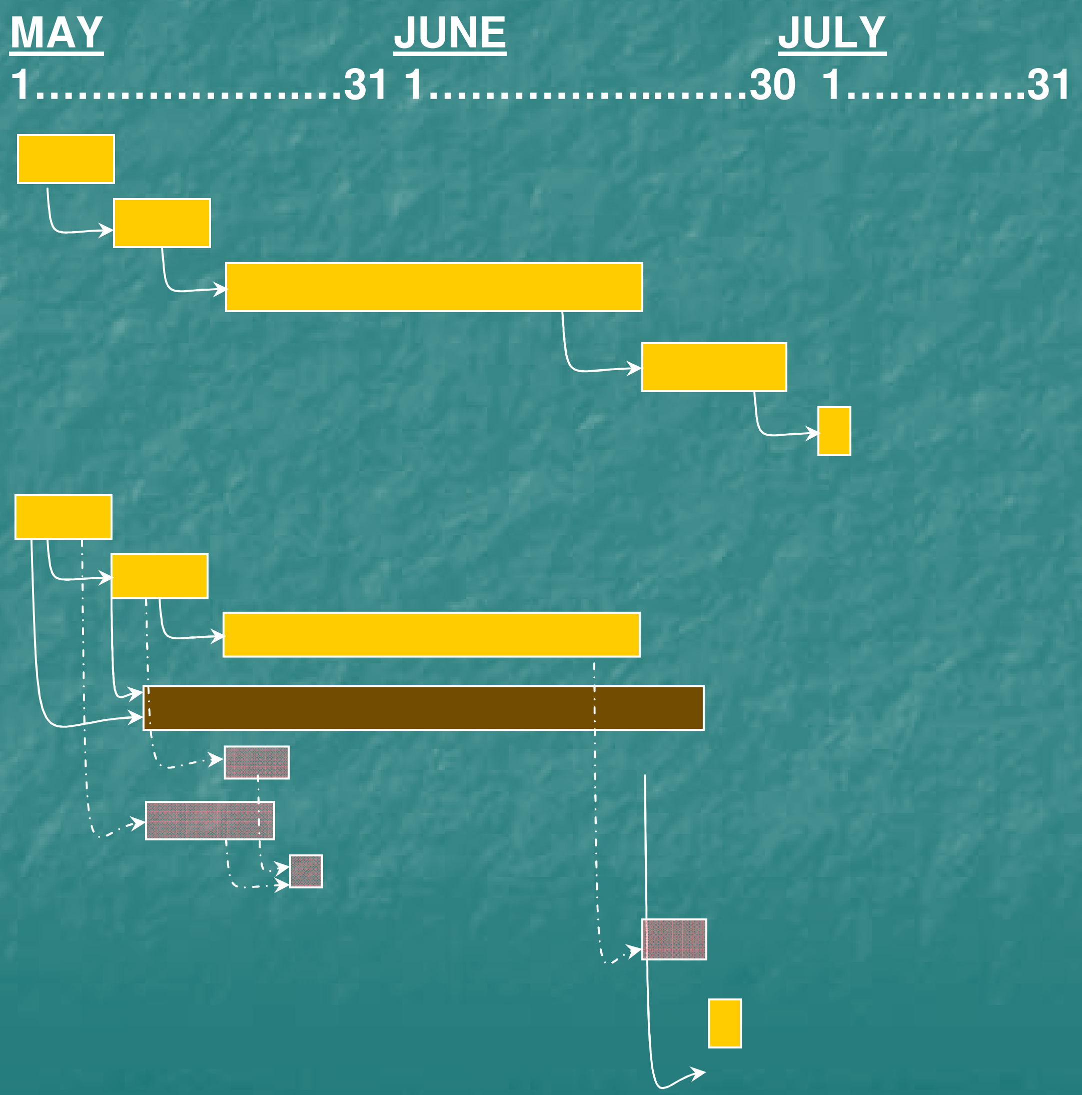
SYSTEM DEVELOPMENT LIFE CYCLE
Late start vs. early start testing

TASK NAME	DURATION
Analysis	5 Days
Design	5 Days
Code	25 Days
Test	10 Days
Install	2 Days
Analysis	5 Days
Design	5 Days
Code	25 Days
Test	28 Days
Install	2 Days



SYSTEM DEVELOPMENT LIFE CYCLE
Late Start vs. Early Start Testing

TASK NAME	DURATION
Analysis	**5 Days**
Design	**5 Days**
Code	**25 Days**
Test	**10 Days**
Install	**2 Days**
Analysis	**5 Days**
Design	**5 Days**
Code	**25 Days**
Test	**28 Days**
• Write Unit Test Plan—(2 DAYS)	
• Write System Test Plan – (4 DAYS)	
• Review Test Plans – (1 DAY)	
• Execute System Tests – (3 DAYS)	
Install	**2 Days**



QUALITY TOOLS AND STEPS

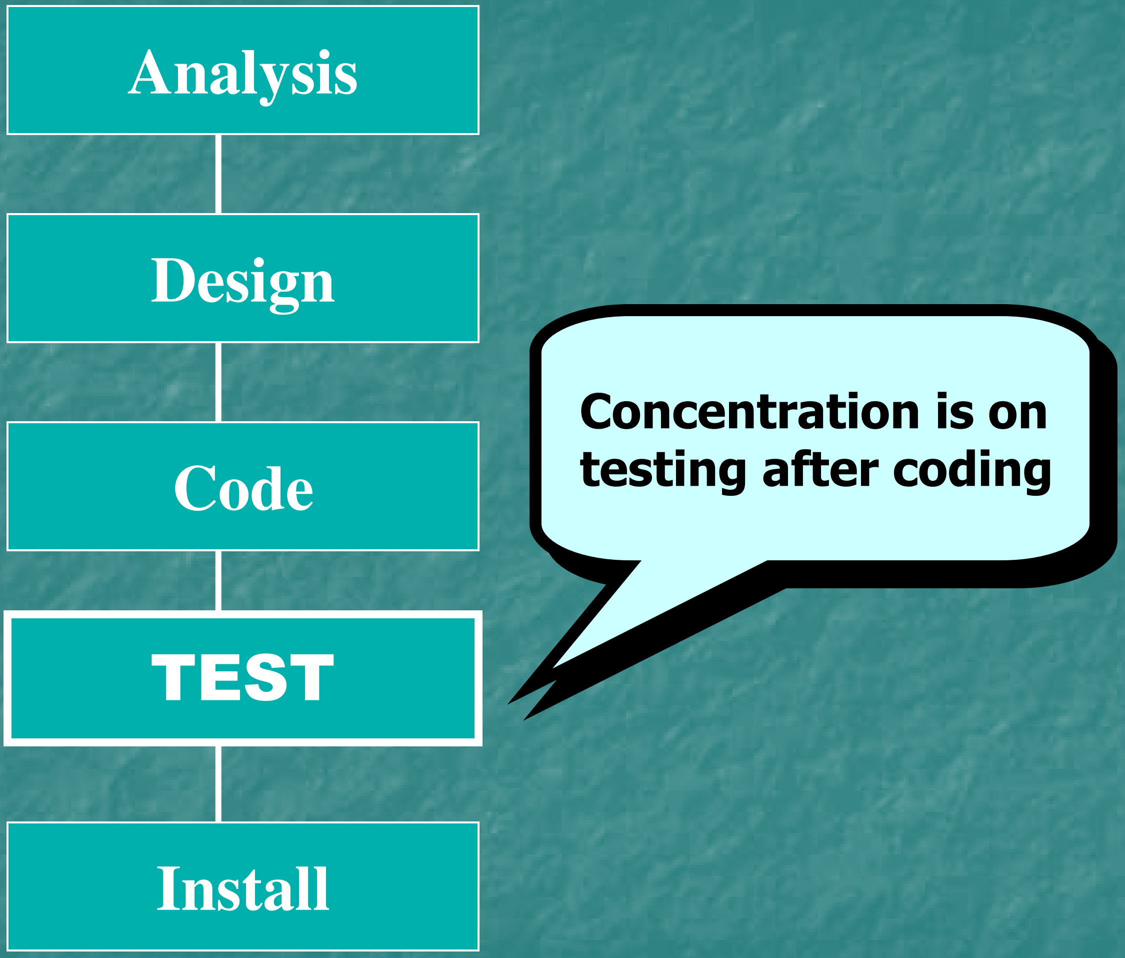
Test Data Generators
Automated Regression Testers
Complexity Measurement / Path Analyzers
Millennium Tools
Network Performance Simulators
Protocol Analyzers
Network Modeling & Simulation Tools
Application partitioning Tools
Network Management Platforms
System Auditors
Defect Tracking and Resolution Managers
Database Integrity Checkers
Comparators
Back-up & Disaster Recovery Tools
System Configuration Managers
Error Handling & Recovery Systems
Software Reliability and Defect Predictors
Standard Benchmarks
CASE Tools
Prototypers
Traceability Matrix Maintenance Tools
Code Optimizers
Performance Measurement & Prediction Tools

Test Case / Script Generators
Automated Code Reviewers
Version Control
Performance Analyzers
Network Diagnostic Tools
Probes & Traffic Monitors
Transaction Processing Monitors
Server Database Monitoring Tool
Memory Leak Detectors
Software Re-engineering Tools
Test Management Tools
Real-Time Test Tools
Maintainability Evaluators
Coverage Analyzers
Logic Emulators
Communications Emulators
Image Quality Checkers
State Transition Diagrammer
Test Data Managers
Data Dictionaries
Pre-Compilers
Report Generators

QUALITY ASSURANCE QUALITY CONTROL INFORMATION SYSTEMS QA

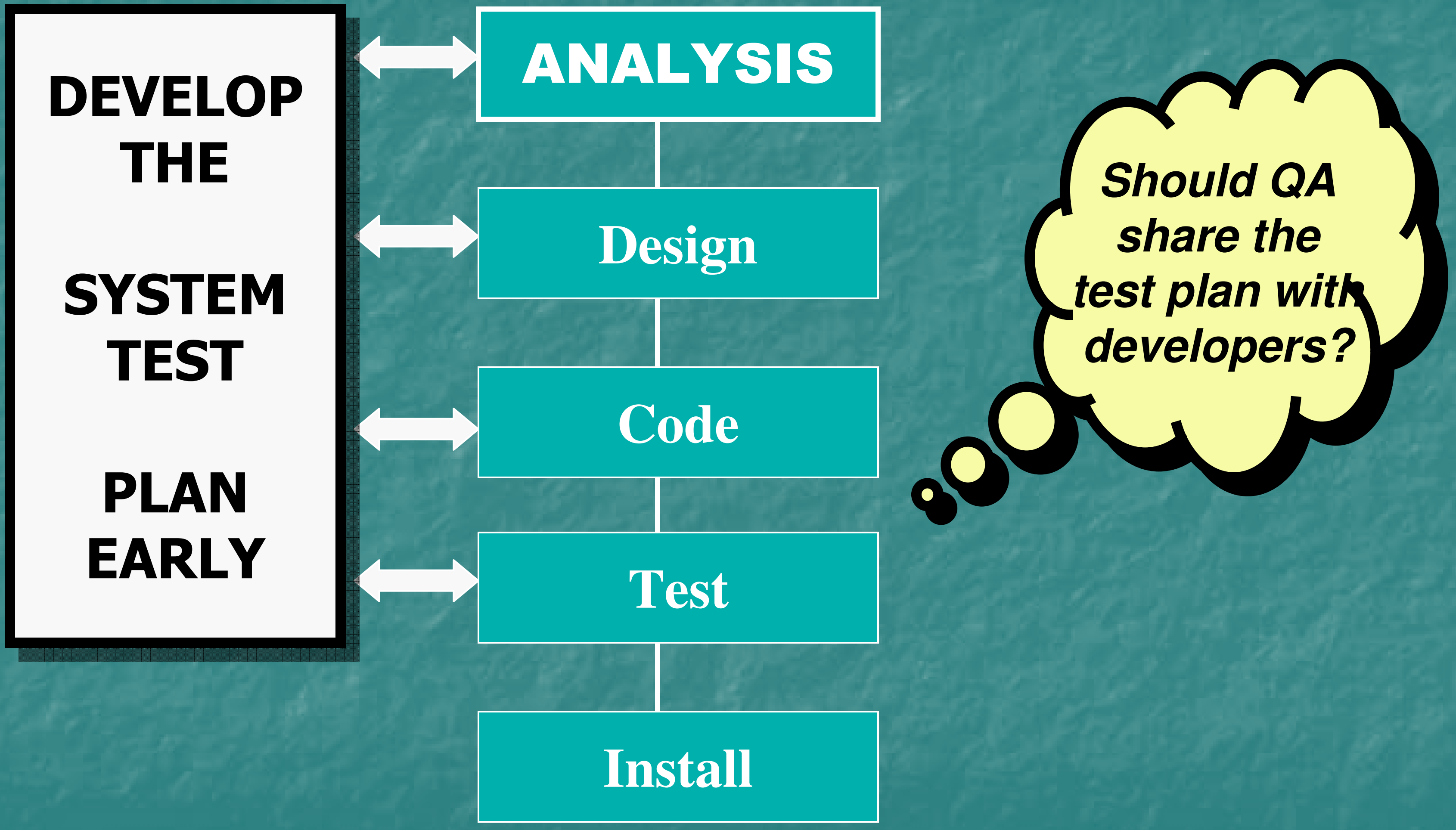
QA	QC	ISQA
• Detached • Guiders • Usually Does Not Test • Works With The Customers	• Internal • Part Of Development • Responsible For Testing	• Combined Functionality • Process Knowledge • Application Knowledge • Internal Group w/ Power

Traditional Waterfall View of Development & Testing

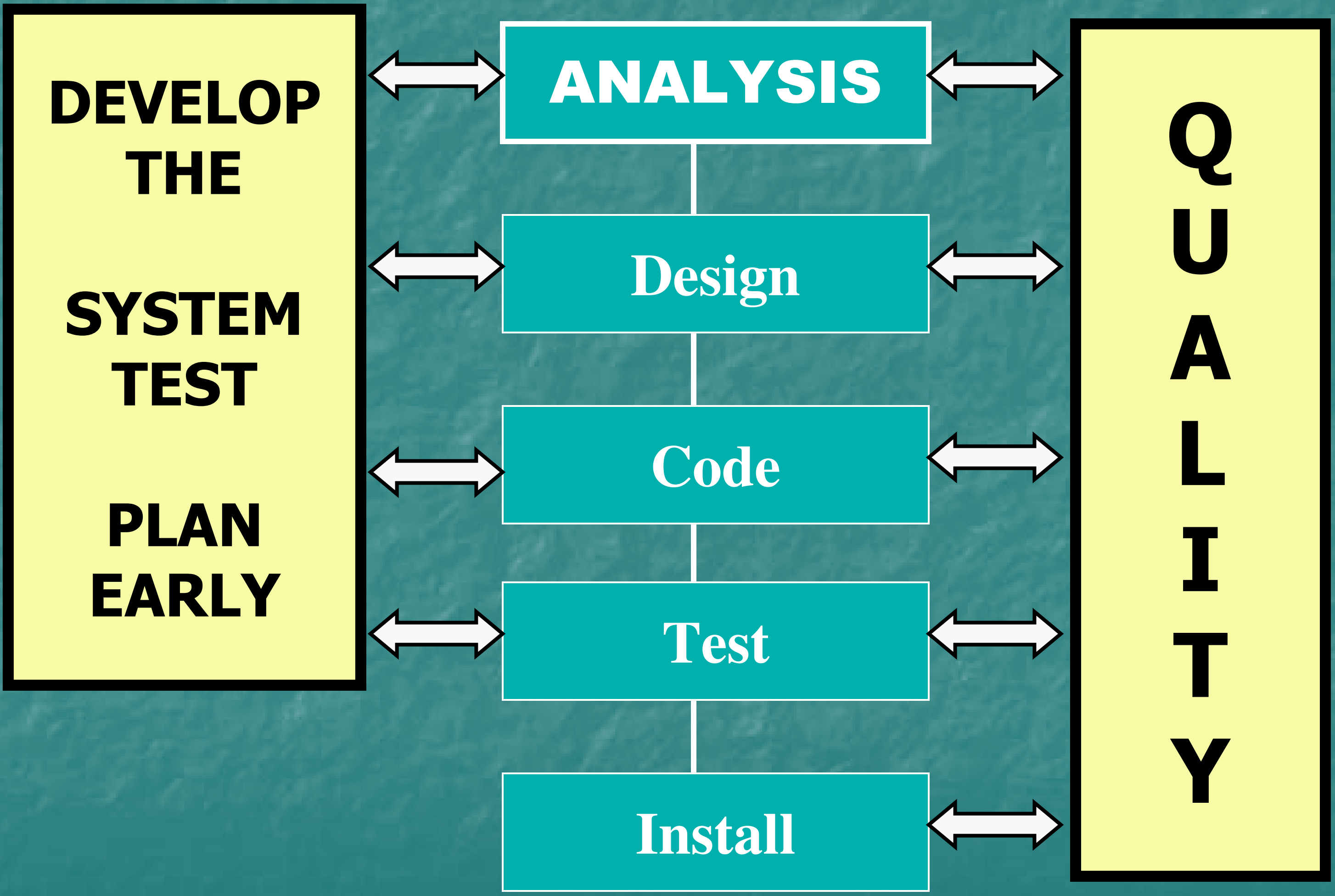


Only the programmer knows the project status

IMPROVED VIEW OF TESTING


DEVELOP THE SYSTEM TEST PLAN EARLY
ANALYSIS
Design
Code
Test
Install
Should QA share the test plan with developers?

BEST VIEW OF TESTING



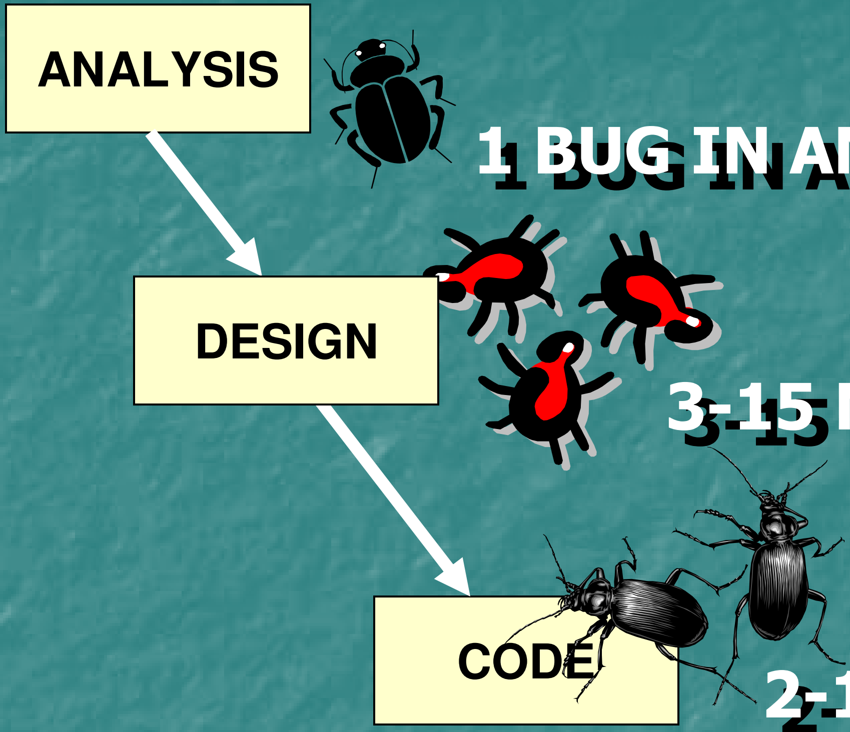
COSTS OF DEFECTS

FIND DEFECTS EARLY !!

Analysis	$10.00
Design	$100.00
Code	$1,000.00
TEST	$10,000.00
Install	$100,000.00

BUG PROPAGATION

How many will you find in testing?



1 BUG IN ANALYSIS

3-15 MORE ADDED IN DESIGN

2-10 MORE ADDED IN CODING

Worst Case: 1 bug in analysis causes 15 bugs in design and they in turn create 10 bugs in the code.

Total = 150 code defects.

	Front End	Coding	Back End
Traditional	35%	15%	50%
Quality Process	42%	18%	28%

- 12% Productivity Increase
- 50+% Fewer Defects
- 12-15% Faster To Market

It may cost less to leave the defects out of the system, than to pay to put them in, pay to find them, and then pay to take them out again.

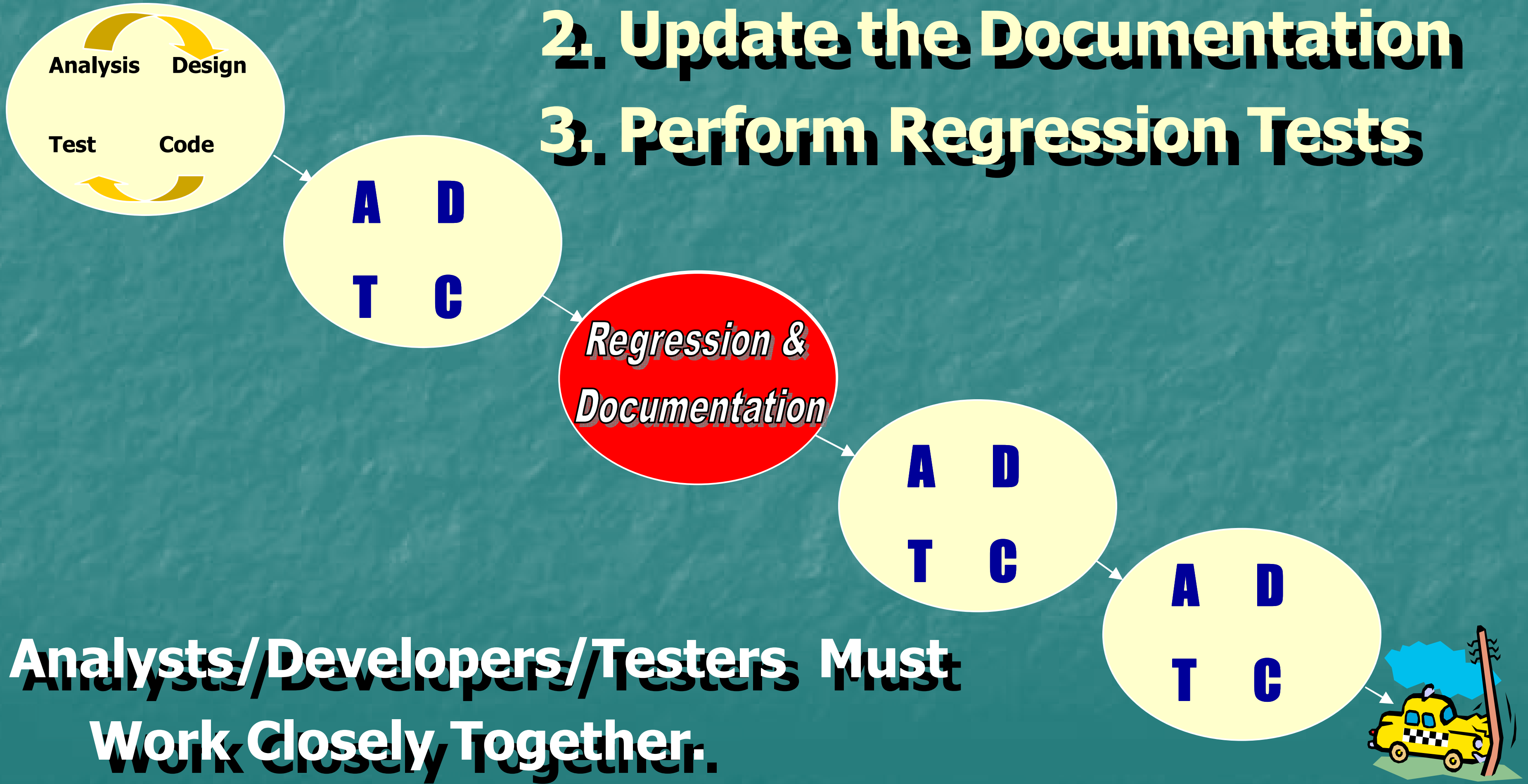
We are still working out the numbers?

CONCISE PROJECT MANAGEMENT

- Milestones
    - 40 hour rule
    - 80 hour rule
- Deliverables
    - Measurable
    - Reviewable
    - Achievable
- Document everything
- Review everything promptly

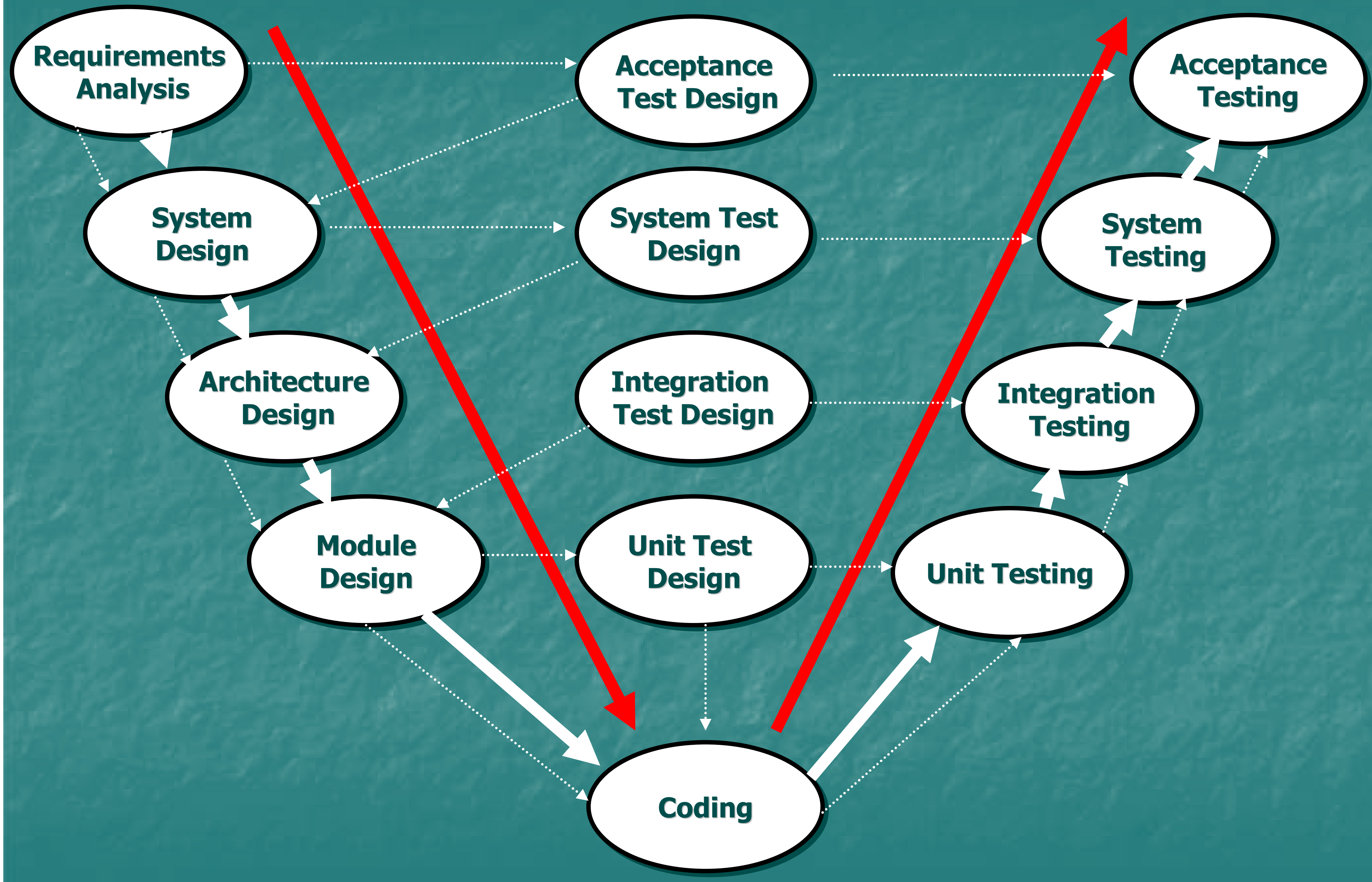
Spiral Cycles & RAD (Rapid Application Development)

1. Skip Cycles On A Schedule
2. Update the Documentation
3. Perform Regression Tests



Analysts/Developers/Testers Must Work Closely Together.

THE V-MODEL


Requirements Analysis
Acceptance Test Design
Acceptance Testing
System Design
System Test Design
System Testing
Architecture Design
Integration Test Design
Integration Testing
Module Design
Unit Test Design
Unit Testing
Coding

RELIABILITY METRICS

Mean Time Between Failures

MTBF1	Crash, software inoperable
MTBF2	Functional failure
MTBF3	Communications failure
MTBF4	Quality failure

RELIABILITY METRICS

Mean Time To Repair

MTTR1	Actual time to fix
MTTR2	Total time in queue

PRODUCT DEVELOPMENT AND TESTING PHASES

- Needs Analysis
- Specification
- Functional/Business Requirements
- Requirements Walkthrough
- Critical Success Factors
- Acceptance Criteria
- White Box Test Plan
- Black Box Test Plan
- System Design
- Design Develop Test Cases
- Coding
- Unit Testing
- Module Testing
- Integration Testing
- System Testing
- Functionality Freeze
- Alpha Test
- Beta Test
- Parallel Test
- Acceptance Test
- Final Acceptance Test

NOTE: Some of these phases may be taking place simultaneously.

Section 2

Test Methodologies and Checklists

METHODOLOGIES

Consistent ways of identifying tests that need to be run against the application?

A way of ensuring that the application as a whole works according to the users needs, wants, and desires.

A way of ensuring that the application fails gracefully.

WHITE BOX TESTING (STATIC)

A way of ensuring that the components of an application work individually according to the users needs, wants, and desires.

Ensuring that no matter how the application is coded, that the functionality works as intended.

A way of ensuring that the application fails gracefully.

BLACK BOX TESTING (DYNAMIC)

A way of ensuring that the components of an application work together as a whole according to the users needs, wants, and desires.

Ensuring that no matter how the application is coded, that the application works as the user desires.

A way of ensuring that the application does not fail.

WHITE BOX TESTING (STATIC) VS. BLACK BOX TESTING (DYNAMIC)

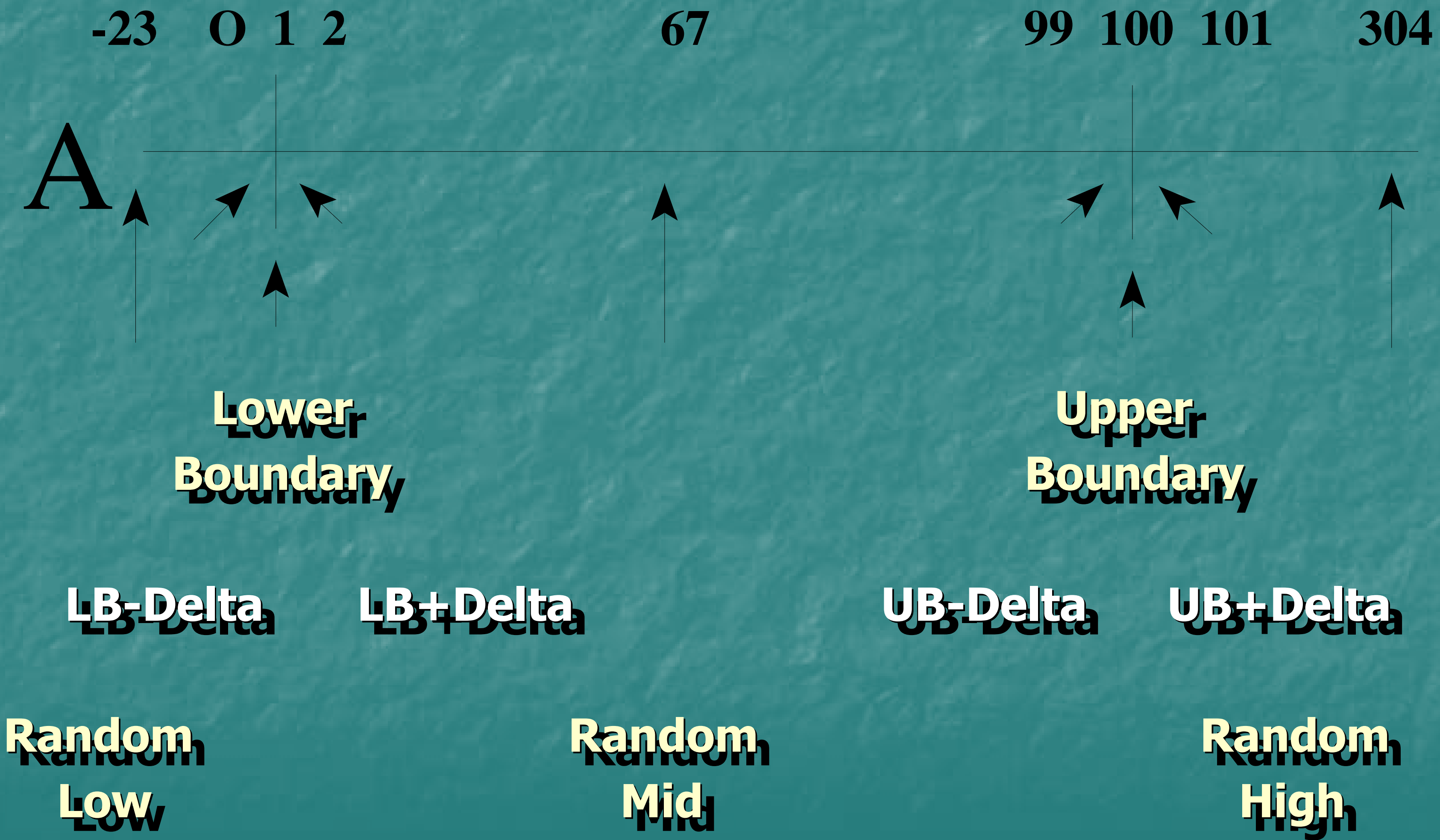
- VISUAL INSPECTION OF MY CAR'S BRAKING SYSTEM.
- ROAD TEST MY CAR ON THE FREEWAY.

Which one should be completed first?

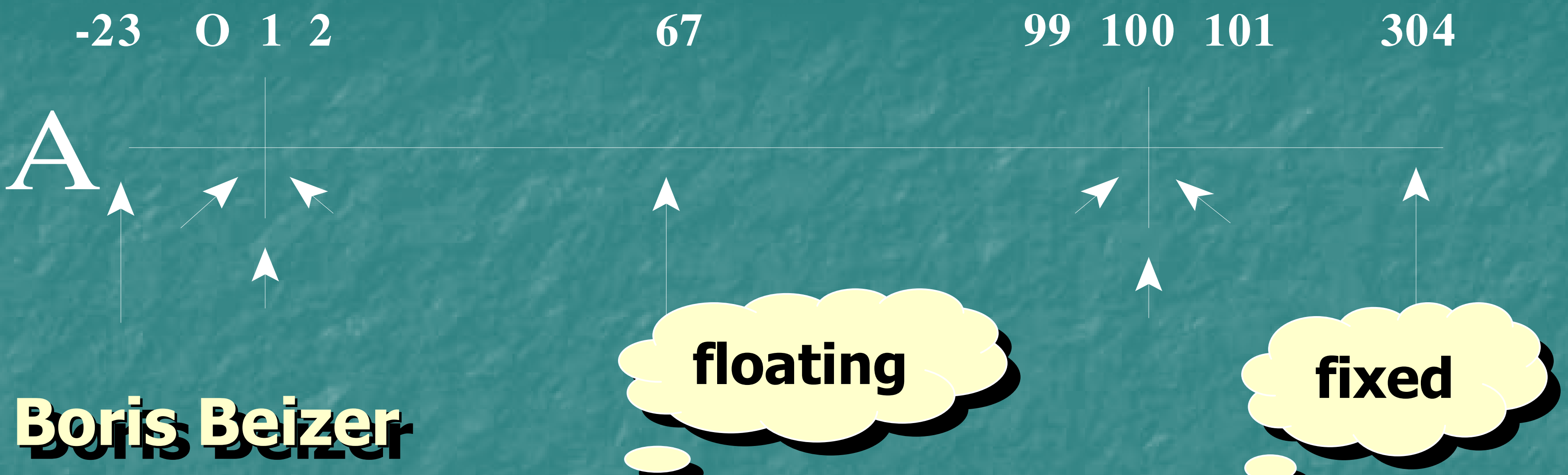
TEST COVERAGE COMPUTATION

$$\textbf{Test Coverage} = \frac{\text{Number of Tests Performed}}{\text{Number of Tests Called for by a Test Methodology}}$$

BOUNDARY VALUE TESTING



BOUNDARY VALUE TESTING



Boris Beizer

	X > 0		x <= 100
	OPEN		CLOSED
POS	NEG	POS	NEG
1	0	100	101

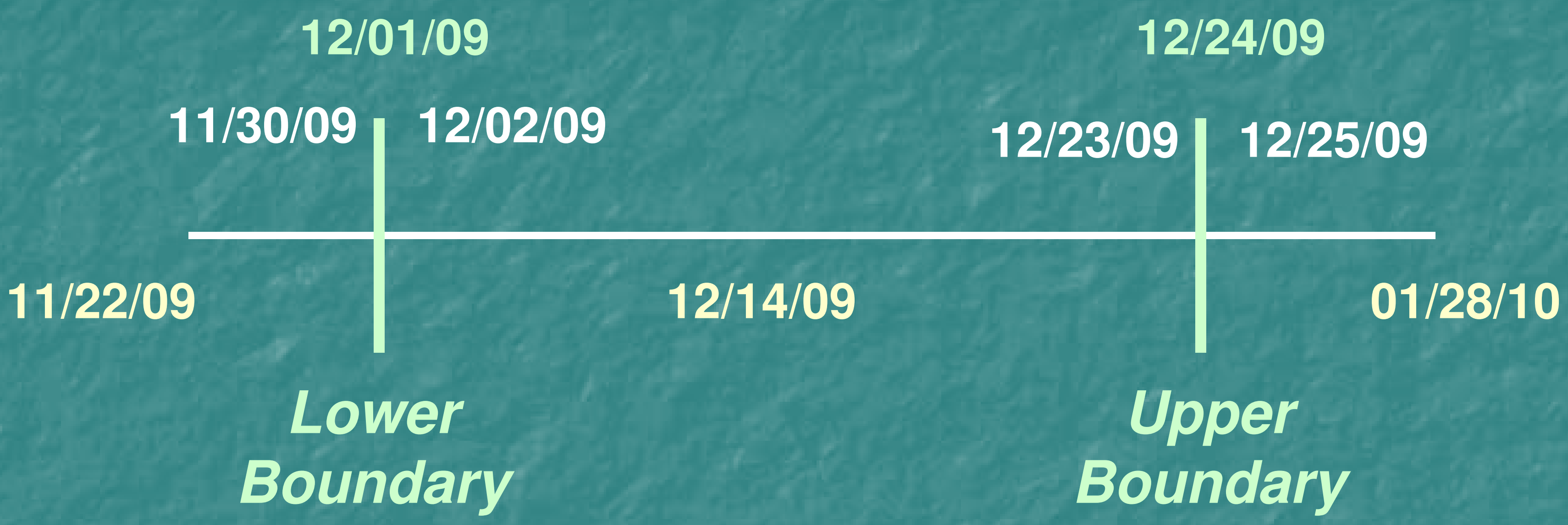
2 TESTS/BOUNDARY = 90 - 95%

Boundary Value Testing with Independent Numeric Fields

Input A	Input B	Expected C
-23	3	Error
0	3	Error
1	3	3
2	3	6
67	3	201
99	3	297
100	3	300
101	3	Error
304	3	Error

Input A	Input B	Expected C
10	-31	Error
10	-6	Error
10	-5	-50
10	-4	-40
10	2	20
10	4	40
10	5	50
10	6	Error
10	25	Error

DATES BOUNDARY ANALYSIS



9 tests/range = 99%
4 tests/range (pos/neg) = 90-95%
2 tests/range (pos/neg) = 50-65%

BOUNDARIES #1

The discount offer is only valid between January 1 and January 31.

BOUNDARIES #2

All calls are rounded to the nearest minute.

BOUNDARIES #3

A warning light will go on at speeds in excess of 75 mph.

BOUNDARIES #4

Customers are permitted to make no more than 2 ATM withdrawals in a 24 hour period regardless of the account balance.

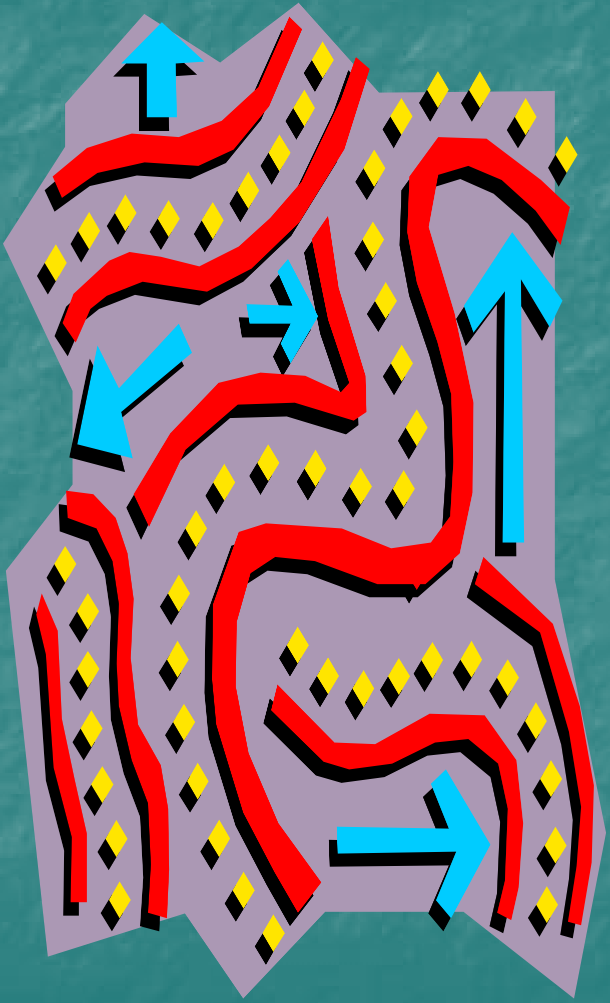
TEST FIRST

Tell me how you are going to test it and I can tell you if you are going to code it correctly.

or,

If you can't tell me how you are going to test it, I can almost guarantee that you will code it incorrectly.

PATH ANALYSIS



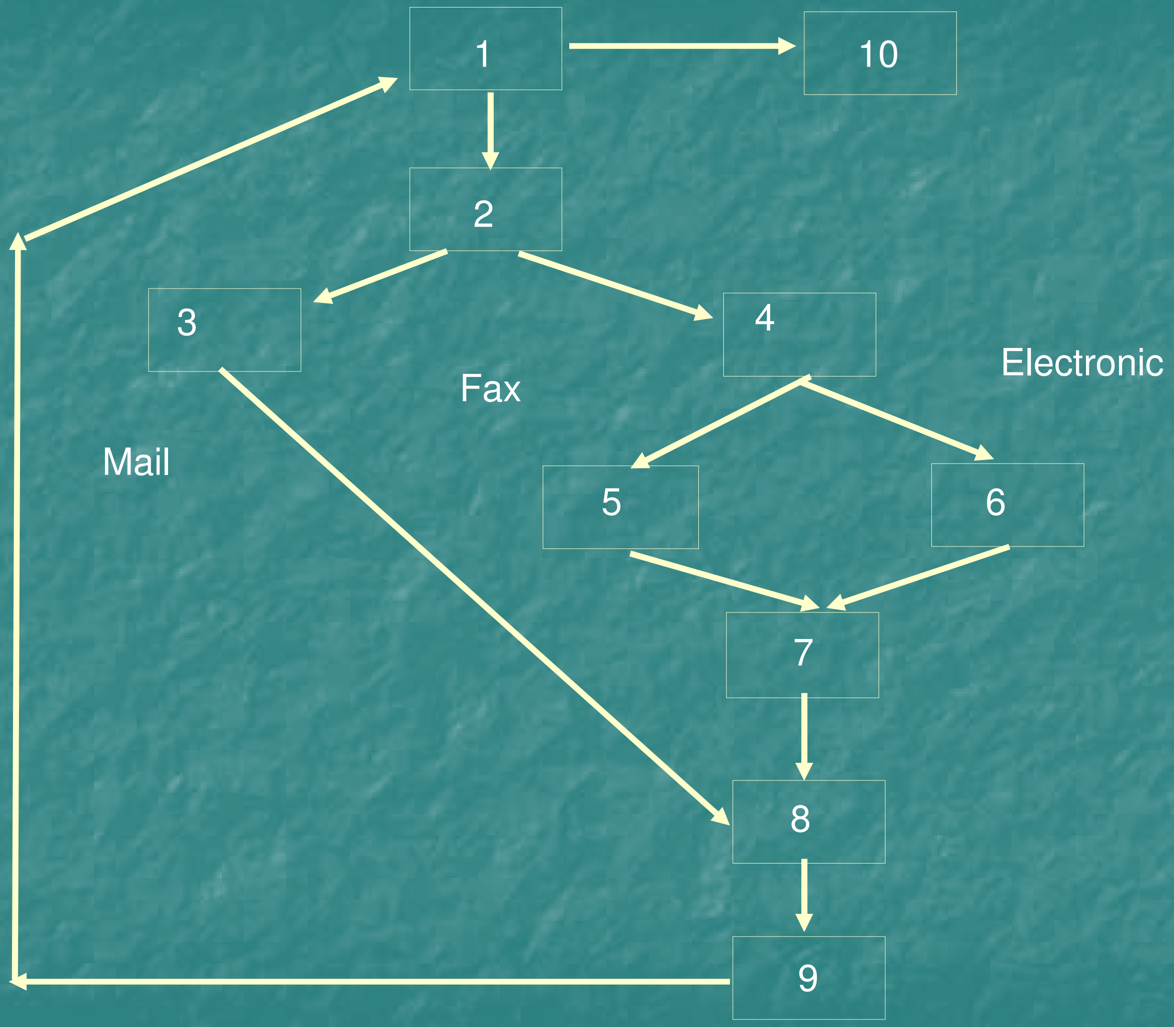
- Thomas McCabe McCabe Associates Battlemap
- Number of Basis Paths
- Cyclomatic Complexity
- 10 or below in 80% of modules or routines
- Scalable Process
- Re-engineering Decisions
- Combinations & Loops are Tested Elsewhere
- Cubic Relationship: Defects and the Number of Paths > 10
- Cubic Relationship: Costs and the Number of Paths > 10

PATH ANALYSIS

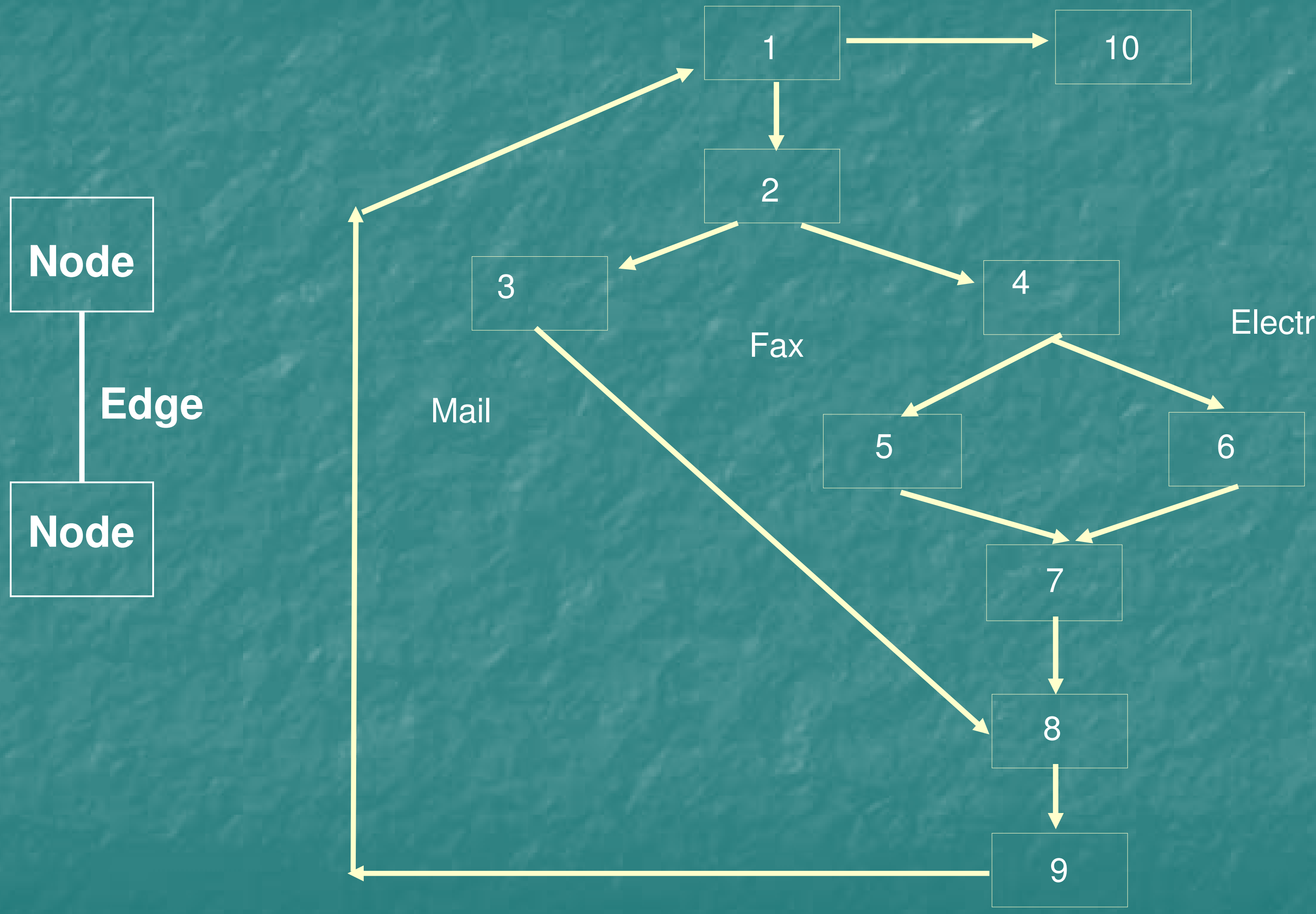
Specification: Count the number of orders received via fax, mail, or electronic submission. No other transactions are processed.

```
1    WHILE NOT END-OF-FILE READ TRANSACTION
2    IF ORDER = “MAIL”
3        ADD 1 TO MAIL-CTR
4                ELSE IF ORDER = “FAX”
5                        ADD 1 TO FAX-CTR
6                        ELSE ADD 1 TO ELEC-CTR
7                ENDIF
8        ENDIF
9    ENDWHILE
10   RETURN
```

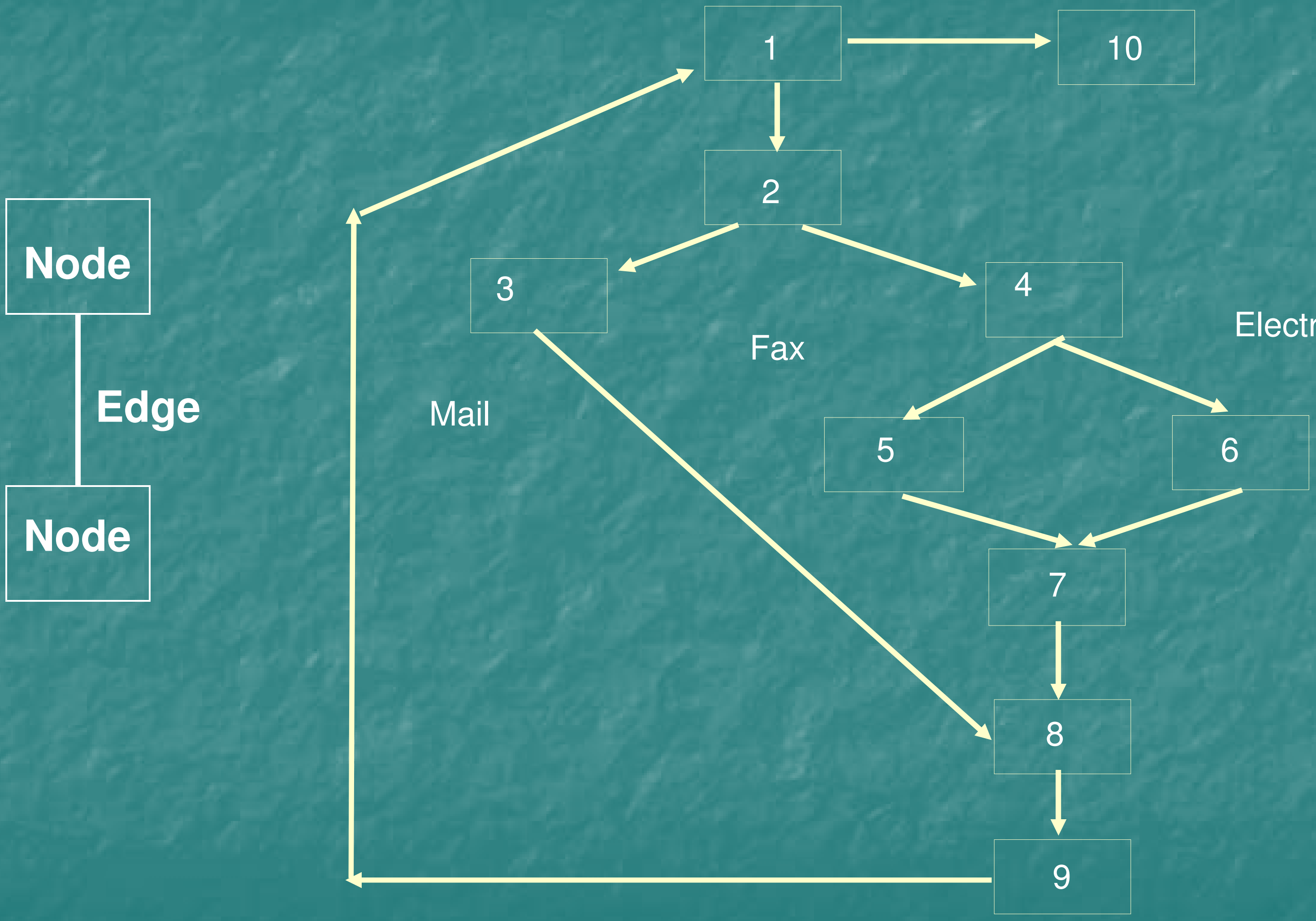
DATA FLOW - FLOWGRAPH

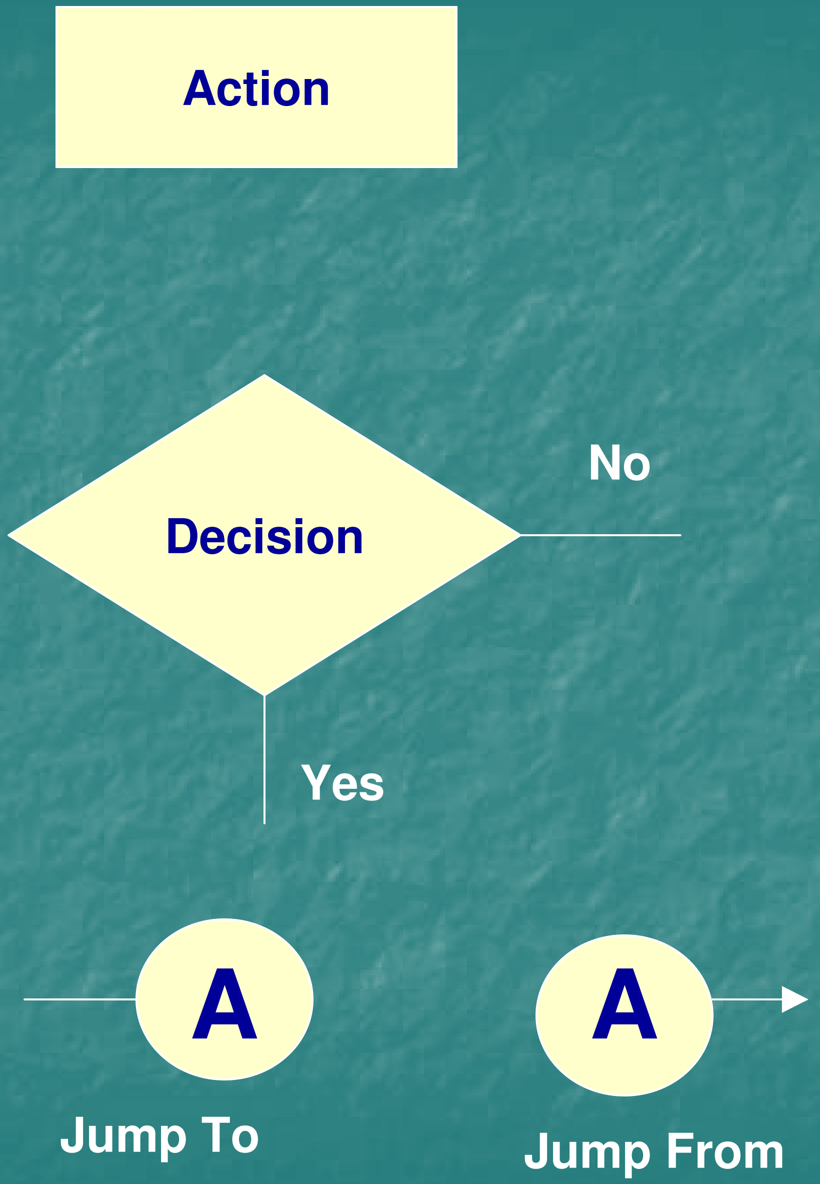


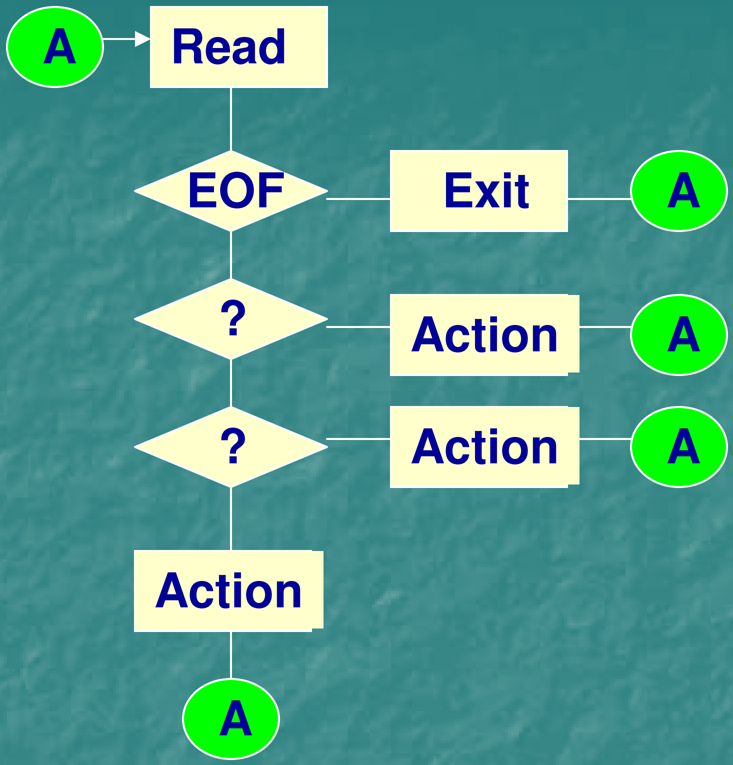
COMPLEXITY = ENCLOSED AREAS + 1



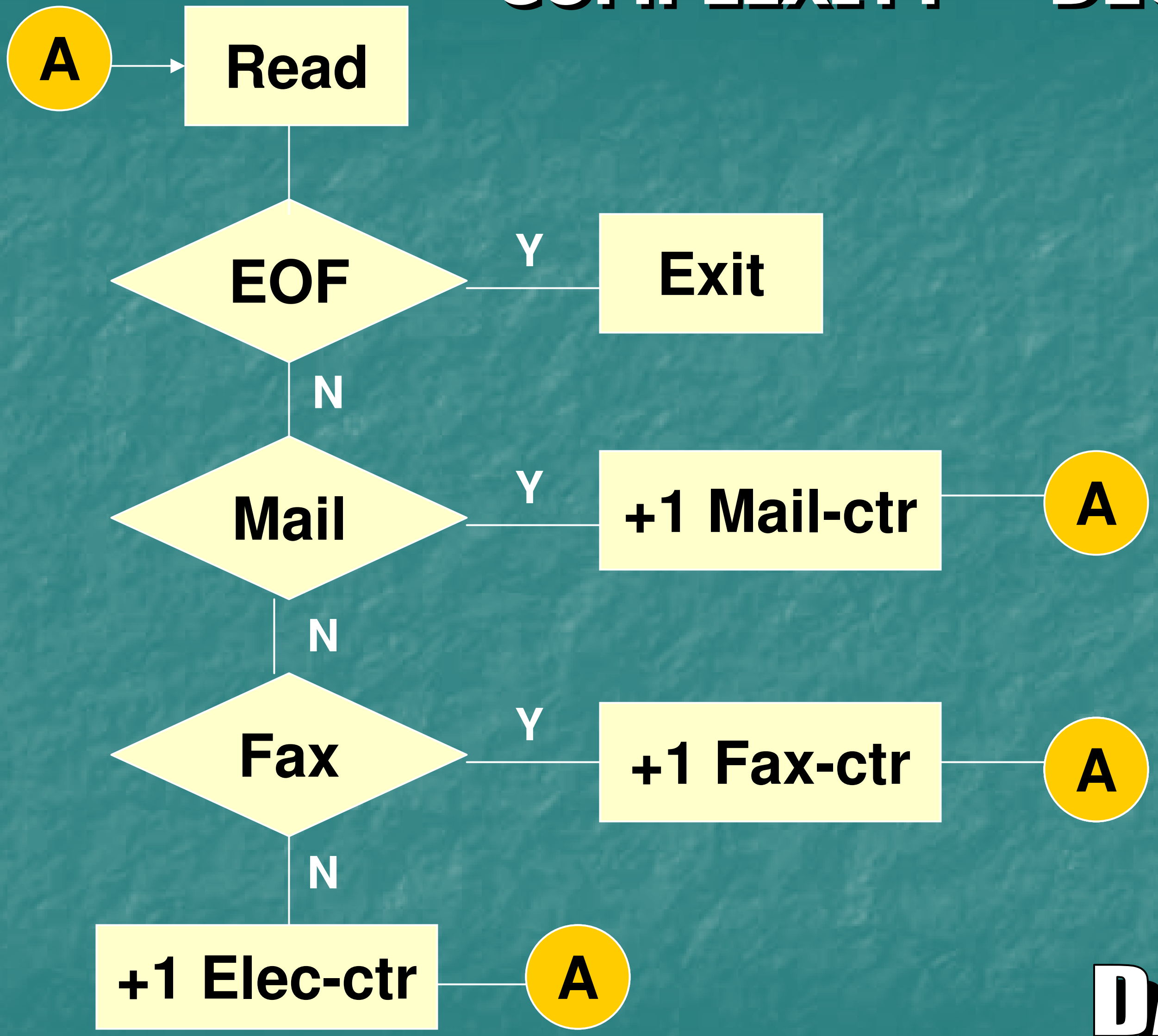
COMPLEXITY = EDGES – NODES + 2




Action
Decision
No
Yes
A
Jump To
A
Jump From

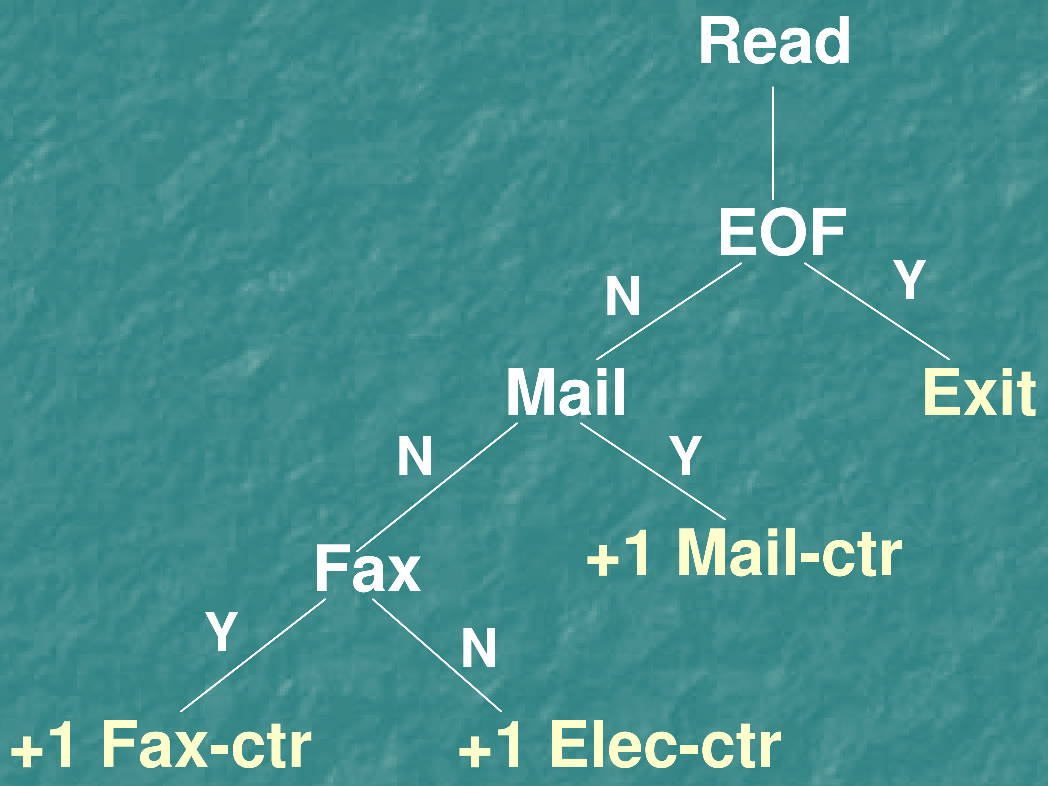

A
Read
EOF
Exit
A
?
Action
A
?
Action
A
Action
A

COMPLEXITY = DECISIONS + 1



DATA FLOW

DECISION TREE

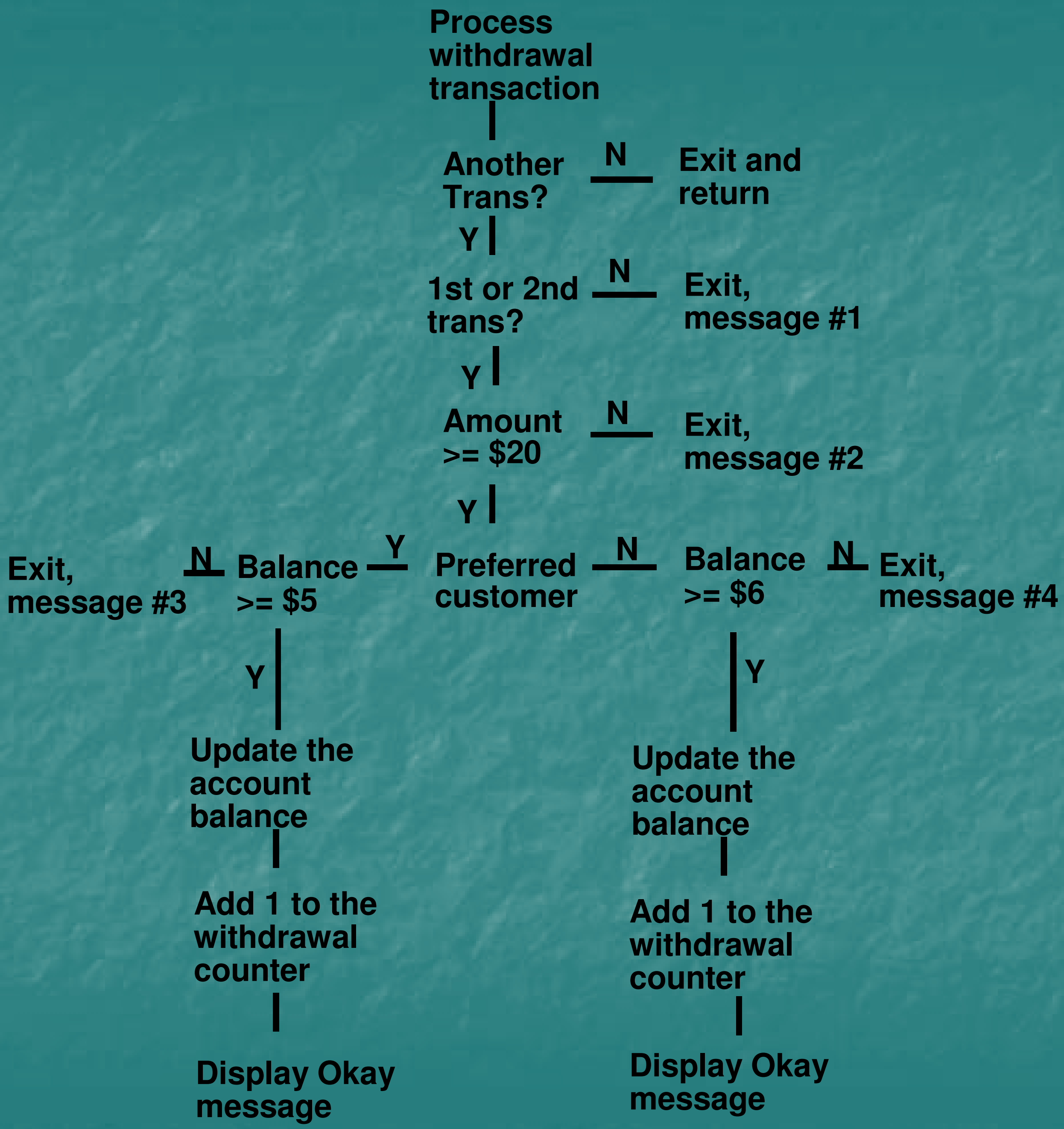


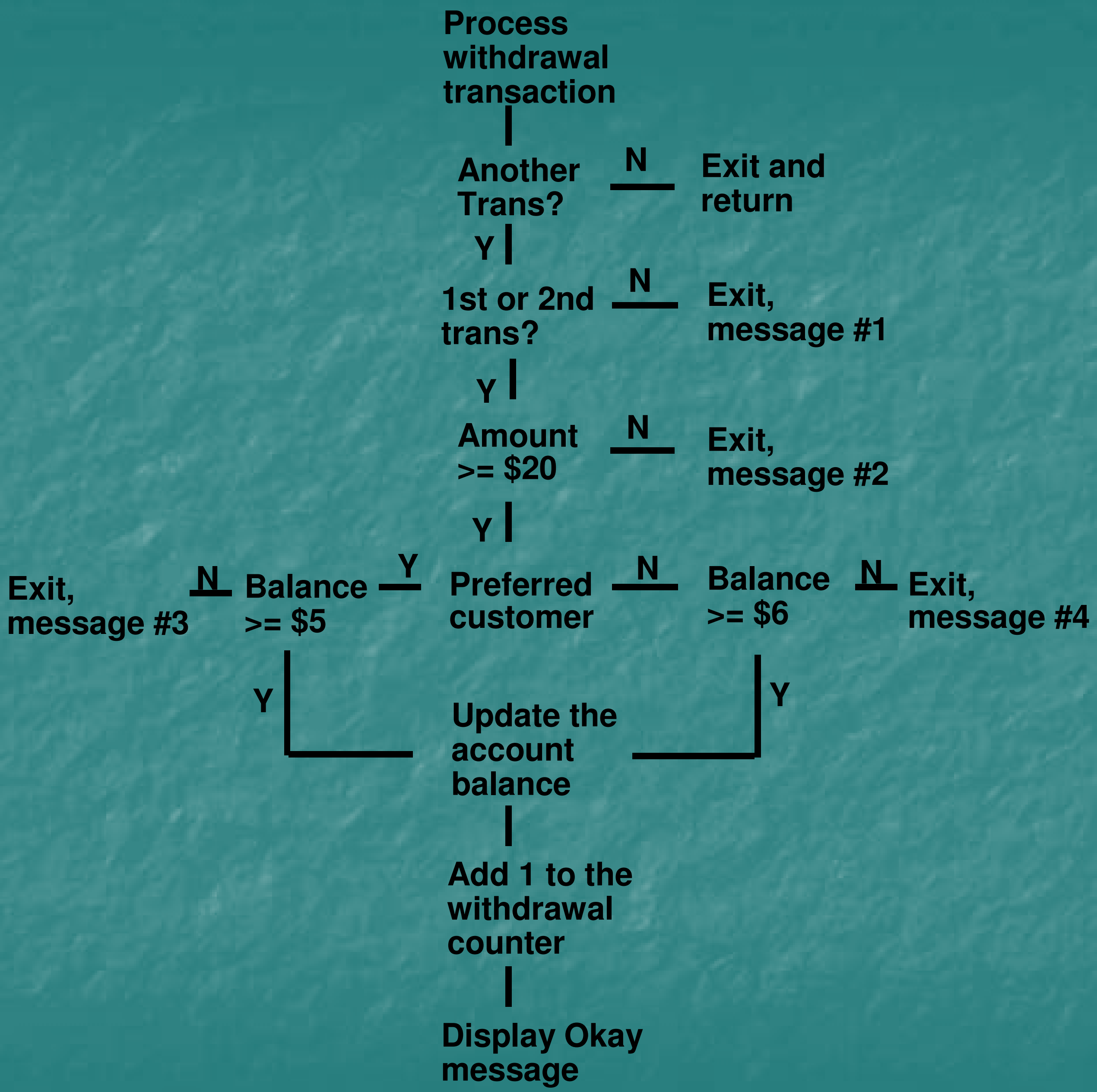
COMPLEXITY = ENDPOINTS

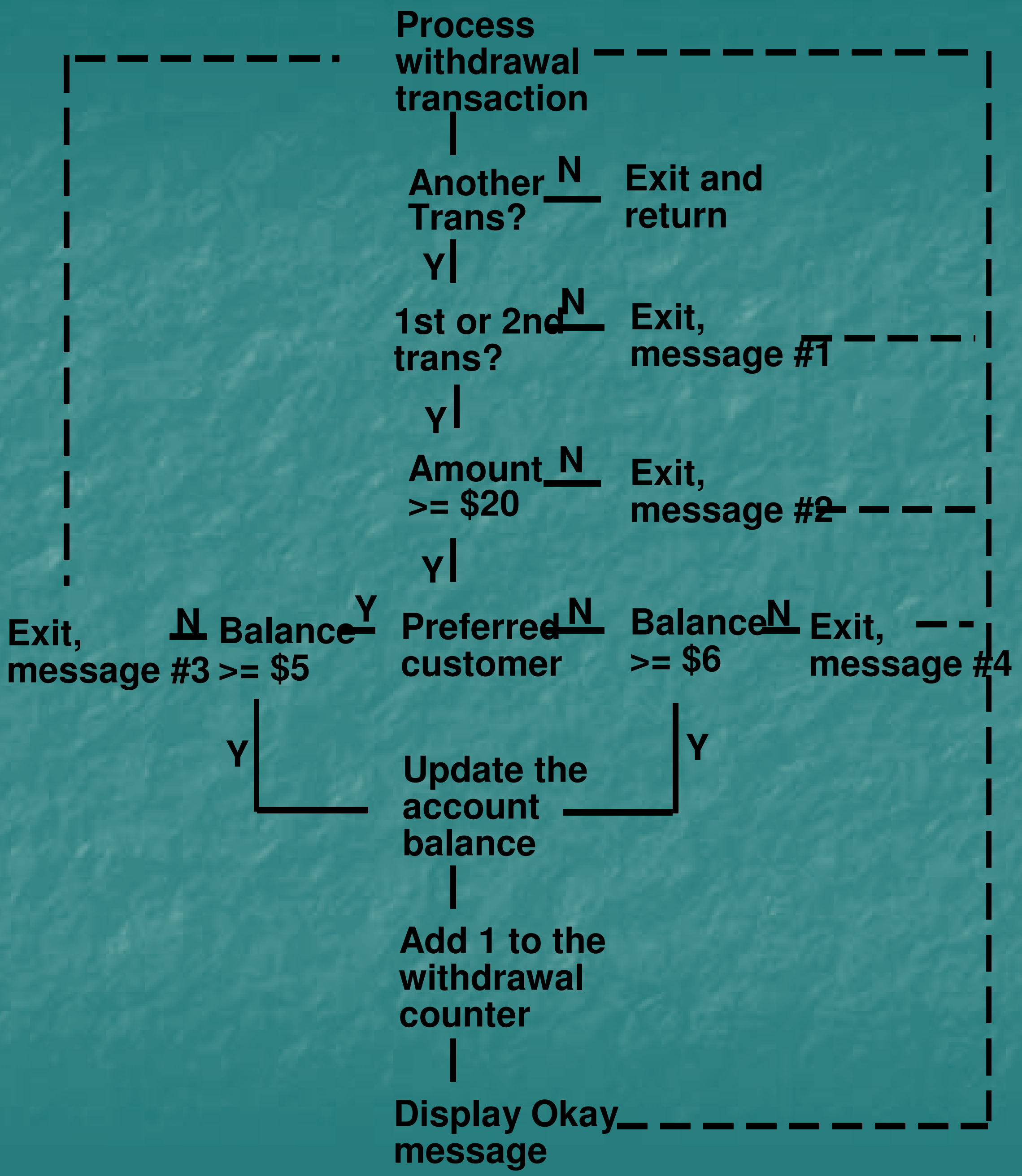
ATM Withdrawal Example

The following specification identifies account balance requirements for making an ATM withdrawal. The account balance must satisfy the following requirements before an ATM withdrawal will be approved:

A. Minimum withdrawal amount is $20.
B. Preferred customers (identified by account number) will not pay a transaction fee.
C. The transaction fee for non-preferred customers is $1 per withdrawal.
D. The account balance must be at least $5 after the transaction.
E. A maximum of 2 withdrawals can be made from an account in a 24-hour period.


Process withdrawal transaction
Another Trans?
N
Exit and return
Y
1st or 2nd trans?
N
Exit, message #1
Y
Amount >= $20
N
Exit, message #2
Y
Preferred customer
Y
Balance >= $5
N
Exit, message #3
Y
Update the account balance
Add 1 to the withdrawal counter
Display Okay message
N
Balance >= $6
N
Exit, message #4
Y
Update the account balance
Add 1 to the withdrawal counter
Display Okay message

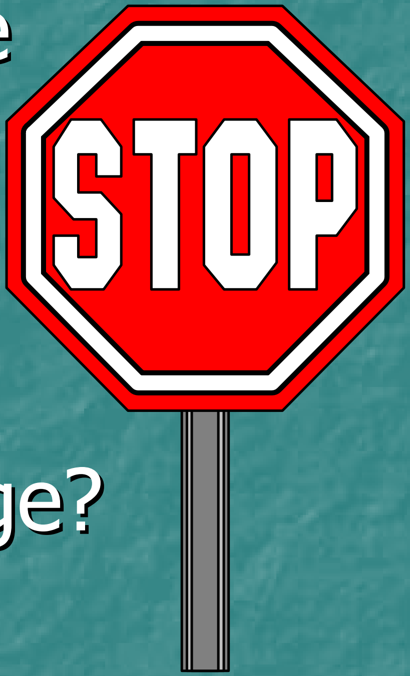

Process withdrawal transaction
Another Trans?
N
Exit and return
Y
1st or 2nd trans?
N
Exit, message #1
Y
Amount >= $20
N
Exit, message #2
Y
Preferred customer
Y
Balance >= $5
N
Exit, message #3
Y
N
Balance >= $6
N
Exit, message #4
Y
Update the account balance
Add 1 to the withdrawal counter
Display Okay message



Process withdrawal transaction
Another Trans?
N
Exit and return
Y
1st or 2nd trans?
N
Exit, message #1
Y
Amount >= $20
N
Exit, message #2
Y
Preferred customer
Y
Balance >= $5
N
Exit, message #3
Y
N
Balance >= $6
N
Exit, message #4
Y
Update the account balance
Add 1 to the withdrawal counter
Display Okay message

DID YOU REMEMBER TO...

- ask questions after reading the spec?
- confirm your understanding of the spec?
- diagram the problem?
- develop a testing strategy?
- determine acceptable test coverage?
- make this project a team effort?
- request a review of your work?



AIRLINE UPGRADE - EXAMPLE

TBC Airlines is trying a new 1st Class upgrade program during the month of January. If 1st Class seats are available on a flight and are requested by passengers, upgrades will be offered under the following conditions:

1) Elite frequent flyers must have their Gold card to be upgraded.

2) Non-elite flyers must have an eligible ticket as well as an upgrade certificate.

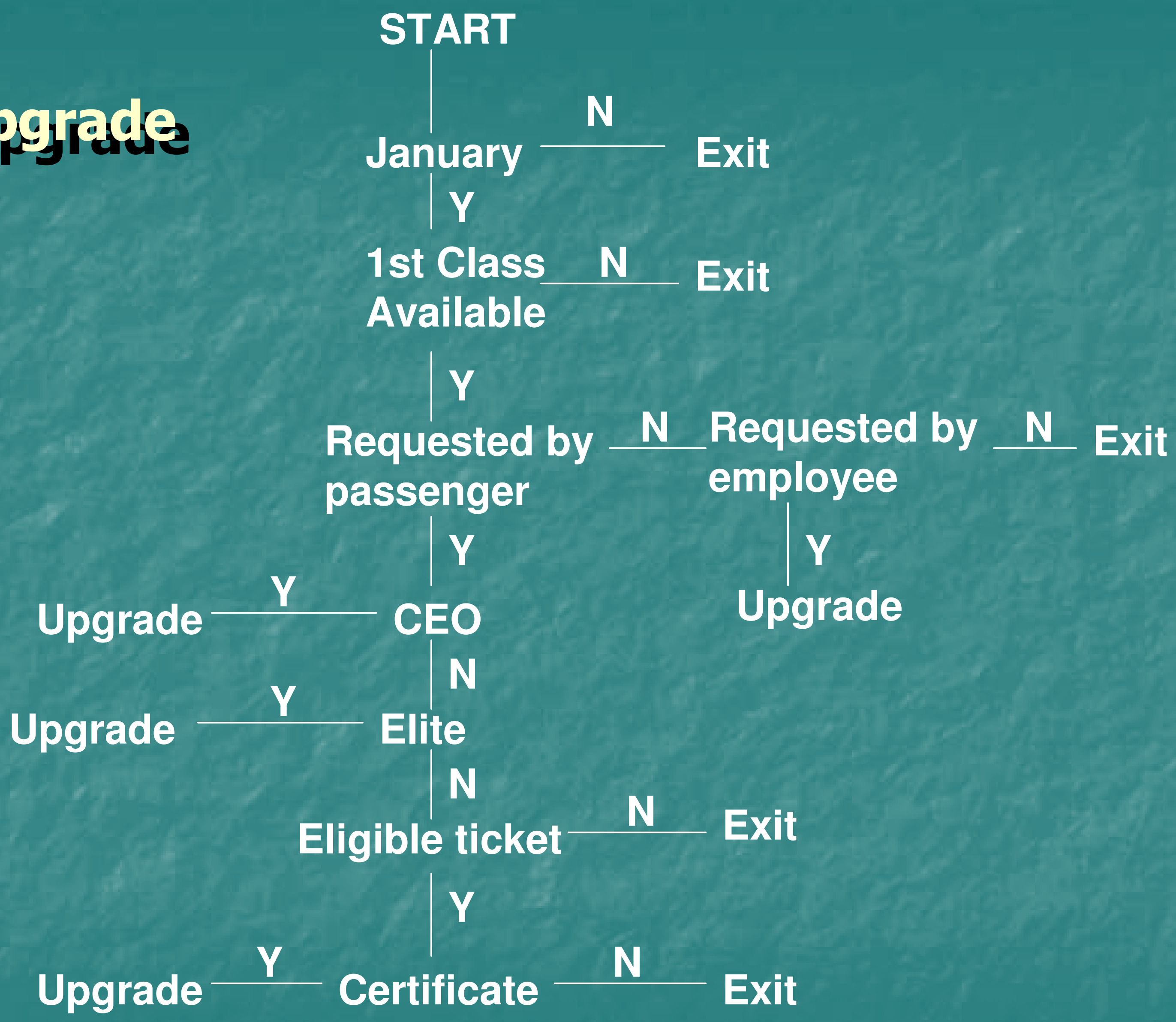
3) To encourage new business, CEO's are always upgraded regardless of their elite status.

4) Employees will be offered upgrades only if no passenger requests are outstanding and all passengers are seated. They are not required to have certificates.

Airline Upgrade Program

		1	2	3	4	5	6	7	8	9
Decisions:										
	January	N	Y	Y	Y	Y	Y	Y	Y	Y
	1st Class available	--	N	Y	Y	Y	Y	Y	Y	Y
	Requested by passenger	--	--	Y	N	N	Y	Y	Y	Y
	Requested by employee	--	--	--	Y	N	--	--	--	--
	CEO	--	--	Y	--	--	N	N	N	N
	Elite status	--	--	--	--	--	Y	N	N	N
	Eligible ticket	--	--	--	--	--	--	N	Y	Y
	Certificate	--	--	--	--	--	--	--	N	Y
Actions:										
	Upgrade	--	--	Y	Y	--	Y	N	N	Y

Airline Upgrade Program



EXPLORATORY TESTING

Exploratory testing is a method of manual testing that is concisely described as simultaneous learning, test design and test execution.

While the software is being tested, the tester learns things that together with experience and creativity generates new good tests to run.

SCREEN EDITS

- Screen defaults
- Function keys
- Escape key
- Minimize screen
- Maximize screen
- Drag screen
- resize screen
- Initial screen size
- Initial screen location
- O/S Characteristics
- Resolution
- Color
- Fonts
- Menu bars
- Button bars
- Navigation bars
- Slide bars
- Screen title

BUTTON EDITS

- Single click
- Double click
- Look & feel
- Space bar
- Tab
- Enter
- Hot key
- Other events that can be triggered
- Escape
- Default setting
- Color
- Relational edits
- Focus box

CHARACTER ENTRY

- Leading spaces
- Trailing spaces
- Embedded spaces (multiples)
- Permitted spaces
- Special characters (numeric, CTRL, ALT, SHIFT, foreign)
- Specific valid or invalid characters
- Font
- Color
- Case sensitivity
- Entry templates
- Minimum field length
- Maximum field length

Section 3

Test Planning

SETTING TEST OBJECTIVES

How will you know when to stop testing?

1. Programmers say so.

2. Time is up.

QUESTION

Which activity will enable you to make the greatest contribution to your organization?

1. **Running tests**

2. **Identifying testable conditions**

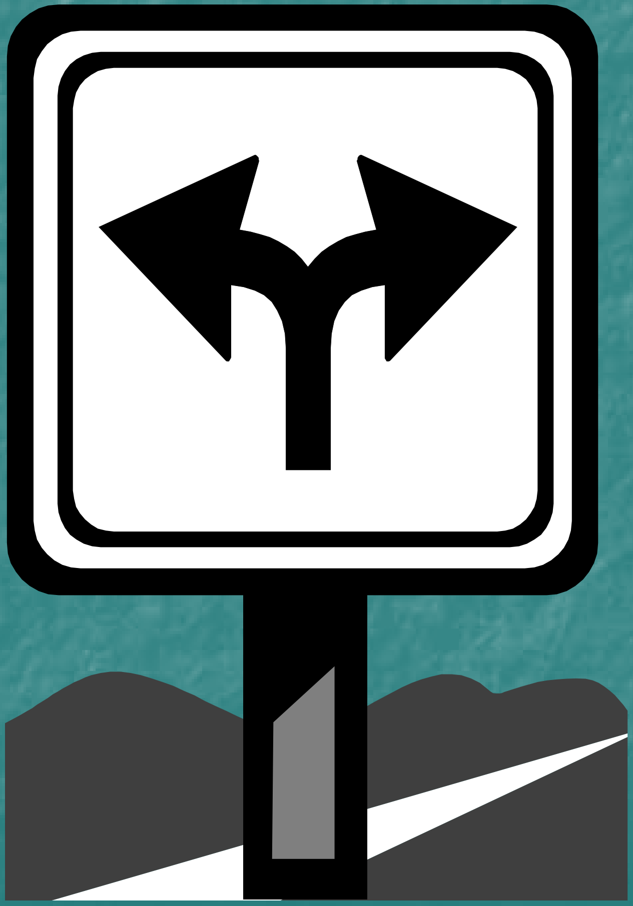
TEST PLANNING

How does the organization know the status of the testing process?

1. Ask the programmers?
2. Ask the users?
3. Ask the testers?

PROJECT SPECIFICATIONS

- Major Source Of Defects
- A Bad Start Can Only Give Bad Results
- Definitions That Are Clear To Everyone



TEST SCRIPT vs. TEST CASE

TEST SCRIPT

- A description of the test that is about to be performed.

TEST CASE

- Actual data and setup requirements used in the execution of a test script.

TEST SCRIPTS (GENERIC TESTS)

Unit Test Scripts

- Component level tests

System and Acceptance Test Scripts

- Functional tests
- Test threads
- End-to-end tests
- Start-to-finish

UNIT TEST vs. SYSTEM TEST

Verify that by failing to enter one or more of the required fields the error message 01-"One or more required fields missing" is displayed when trying to save the record.

Verify that if a level 3 user retrieves an existing record and changes the current address, after saving the record, if a paycheck is printed that the paycheck prints with the modified address.

UNIT TEST SCRIPT - EXAMPLES

Verify that new job descriptions are appended to the Job table after entry.

Verify that the entry date is a valid date.

Demonstrate that the name field is present and does not exceed 30 characters.

SYSTEM / ACCEPTANCE TEST SCRIPTS

- **Verify** that each paycheck and W2 contains the complete employee name and address and that these are the same on the master record.

- **Verify** that the reimbursed amount is equal to or less than the claimed amount and that this amount appears with the current date in the check register.

UNIT TEST CASES

Verify that all subscription dates are valid-
(Use the system date: 03/01/1999)

01/05/1998
13/10/1997
12/31/1999
02/29/2000
02/29/1999

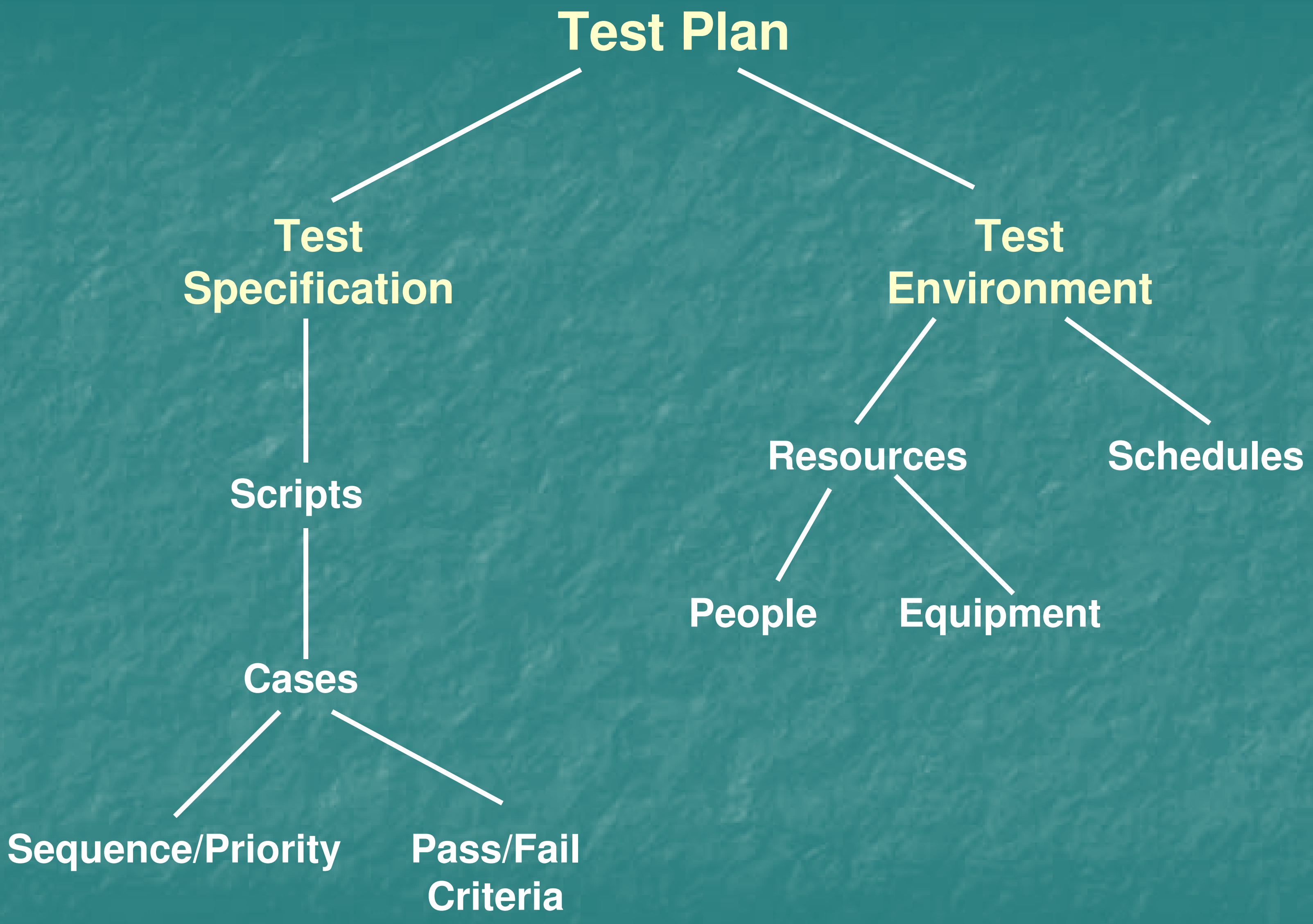
SYSTEM / ACCEPTANCE TEST CASES

Script:

Verify that no paychecks are prepared for anyone with a release date in the Employee Master File.

Procedure:

- Enter a release date for employee No. 10
- Verify release date is before the system date
- Perform production payroll run for exempt employees
- Perform production payroll run for non-exempt employees


Test Plan
Test Specification
Test Environment
Scripts
Resources
Schedules
People
Equipment
Cases
Sequence/Priority
Pass/Fail Criteria

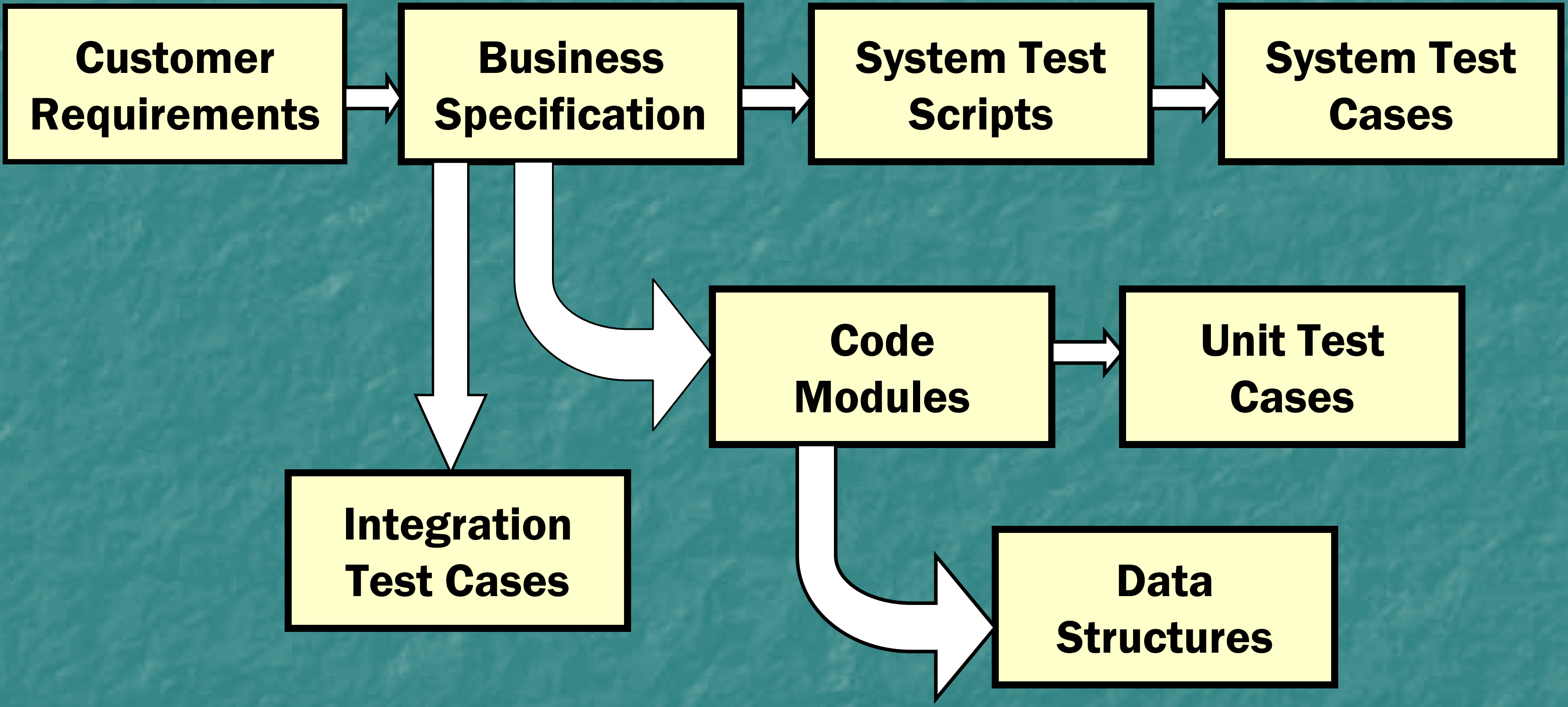
POSITIVE & NEGATIVE TESTING

- Positive
  (within the expected range)
- Negative
  (outside of the expected range)
- Must be a balance
- Use the 80/20 rule
- Easier for Programmers
- Easier for Testers

QUOTES

- It compiled, it's got to be good.
- It usually works.
- No reasonable customer would ever do that.
- Trust me, it's okay.
- It worked yesterday.
- It works on my machine.
- I tested it for you.
- Of course it doesn't pass that test.
- What could go wrong?

TRACEABILITY MATRIX



TESTER'S NOTEBOOK

A personal logbook for the tester that documents the progress made during testing.

Updated twice a day.

A record of all activities performed to that point during the day.