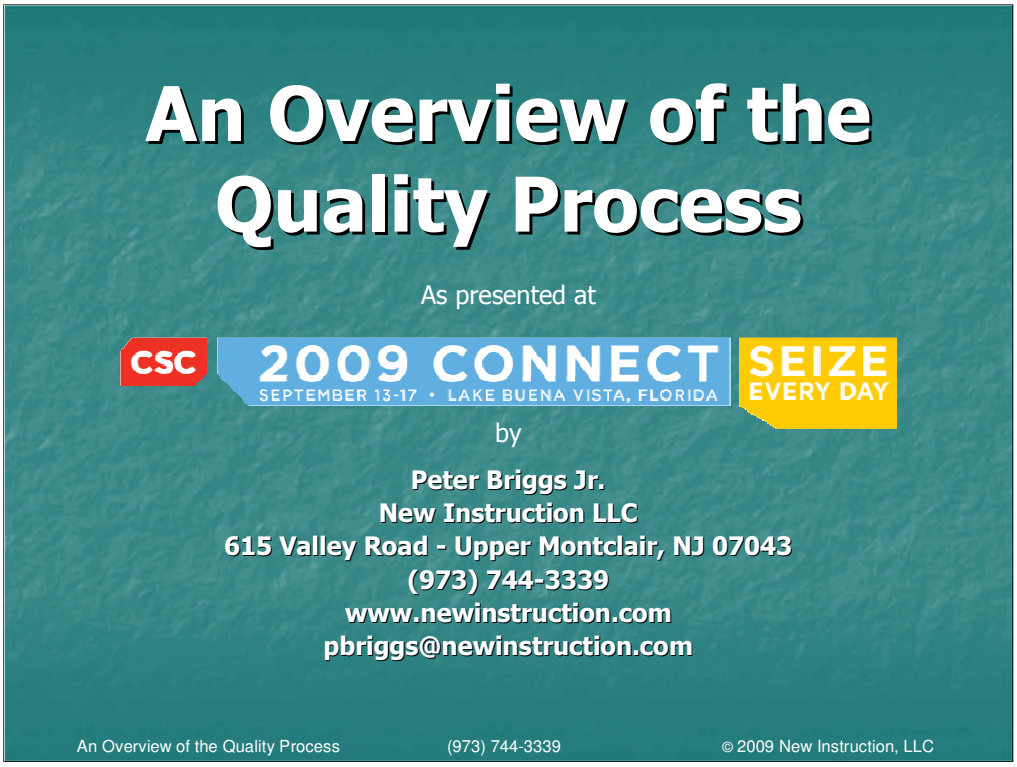


All slide notes are courtesy of …

Ginny A. Eiwen, PMP

CSC Exceed Quality Office Manager

CSC Financial Services Group

Ginny's knowledge of this presentation was augmented through the previous exposure of bringing New Instruction's 3-day workshop in-house for her team.

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.

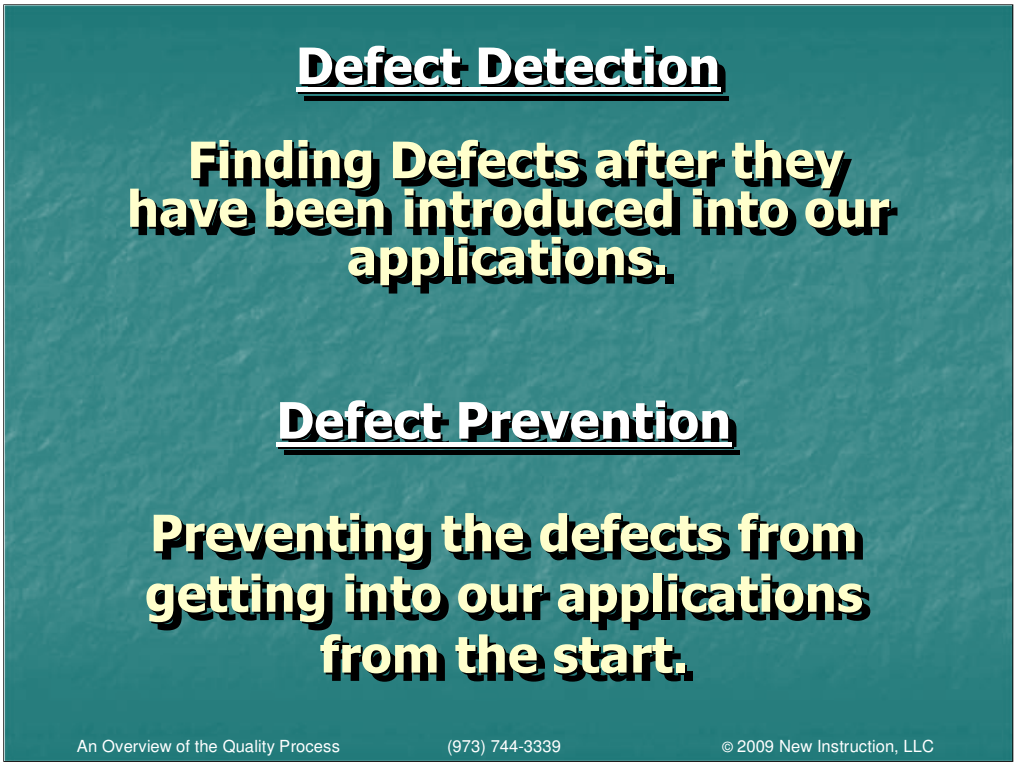
- Unit Testing (Early Testing)
- White Box Test Case Sources
- Sample Unit Test Plan Table of Contents
- Unit Testing Scenario
- Integration Testing and System Testing
- System / Acceptance Testing
- Sample System (or Acceptance) Test Plan Table of Contents
- Sample System (or Acceptance) Test Script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

An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC

Section 1

Defect Detection or Defect Prevention

An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC



Detection is very costly because you have paid someone to do something incorrectly, paid another person to find the problem, another to fix the problem, and another to re-test the function.

Prevention is MUCH more effective.

We should be talking about testing and defect prevention from the very beginning of every effort.

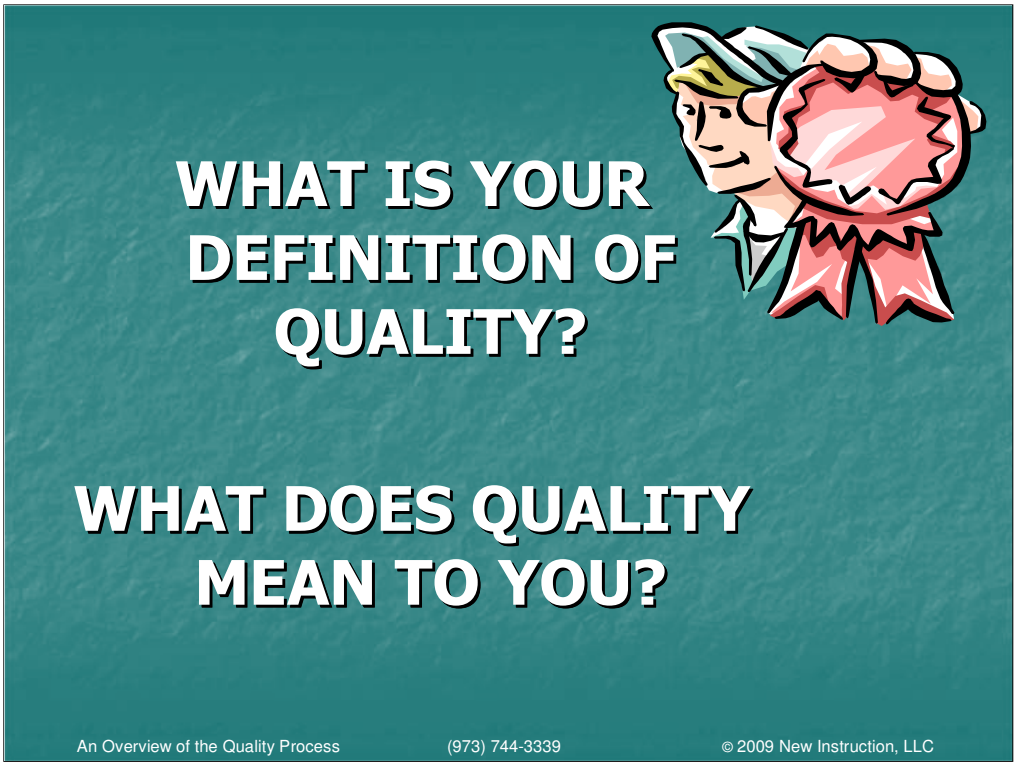
We need to understand the nature of a requirement in order to develop effectively.

WHAT IS YOUR PRIMARY RESPONSIBILITY?

Defect Detection

Defect Prevention

An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC



Everyone needs to work toward the same definition to reach a common objective.

Quality can be defined many different ways – correctness, reliability, maintainability, completeness, usability.

There are many different views of quality – and they all need to work together to get to a shared objective, all need to have a part in defining quality.

- Customer
- Business Analysts – product will solve business issue at hand
- Project Managers – concerned about costs, delivery on time on budget
- Programmers / Developers – maintainability
- QA / Testers – working with other applications
- Technical Support team – customer concerns / upgrades
- Trainers

Cost of Quality – Pay me now / pay me later – spend a little more up front to spend less / have less back end work to deal with.

Quality comes from within systems – it is nurtured and grown. Can't test quality IN to a system.

One way you would suggest that you have quality is when the number of defects begins to DECLINE. Need to do "Lessons Learned" activities to identify those characteristics of successful projects so they can be repeated in future activities.

Need a low water and high water mark for each application to help define 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?

An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC

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

Automate the testing process – allows faster, better, more consistent testing. Manual testing can result in different outcomes if testers conduct tests differently / review results differently. Need to be prepared for automated testing – "A fool with a tool is still a fool!" You need to know what and how to test before you can start automating. Automated testing takes THREE TIMES AS LONG as manual testing. But you 'never' test things just once – so time savings comes from repeated test execution of the test. Very helpful for Regression testing (a retest of the entire system to ensure that new function doesn't break existing function).

Structured reviews – should take place at ALL stages of the Life Cycle. Stopping to check work products at each phase before moving on. Includes reviews of requirements, tests, installation plan, etc. SUGGESTION – have someone other than the author do presentation in a review based on their understanding of code / spec. This gets another person familiar with the spec / code, and helps ensure completeness of spec / code – because if presenter can't answer a question from the review, code / documentation update most likely needed. Also, include QA and customer in code review for verification, and to ensure that code logic is clear enough for non-technicians to understand, which will ensure that technicians in the room thoroughly understand the requirements.

Structured development with reusable code and reusable tests – Antidote to employee turnover.

Measured progress and performance metrics – for example, mean time to defect, mean time to repair (how long does it take to diagnose error, and deliver fix to production).

On-going quality initiatives – how do we make the next cycle better?

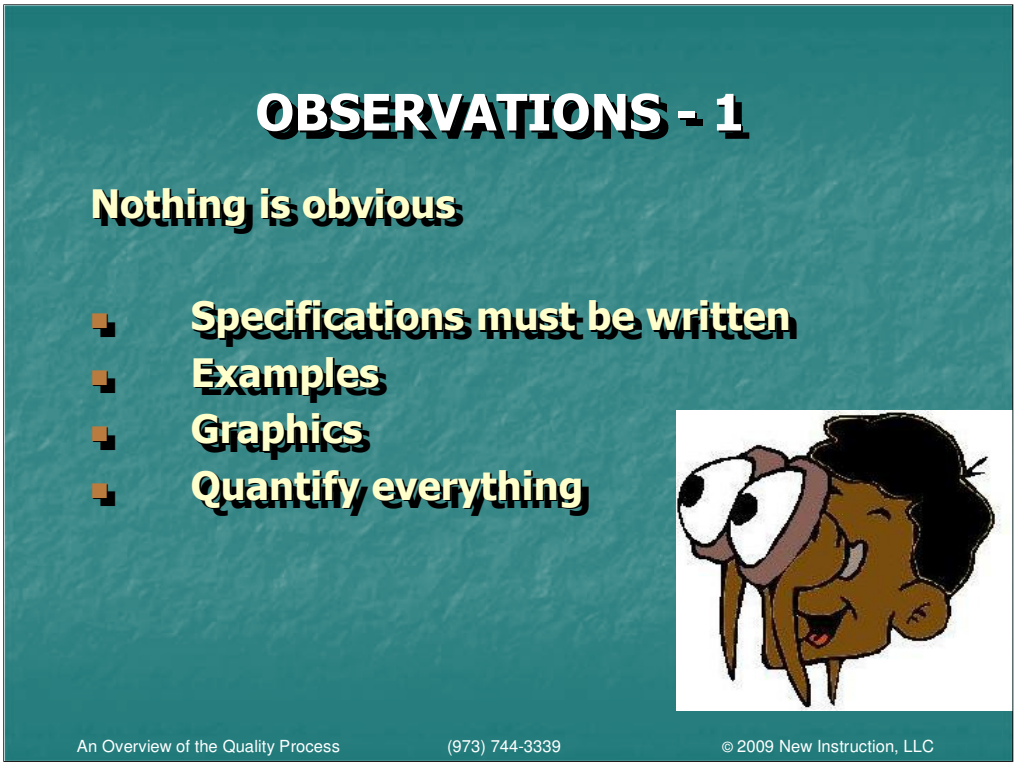
Methods for identifying testable conditions – how do you decide what to test? Need a consistent methodology for identifying testable conditions.

Organized, documented testing process.

Teamwork – No one person knows it all, can solve all problems.

Communications – between all stakeholders.

Enjoy your job – increases motivation.



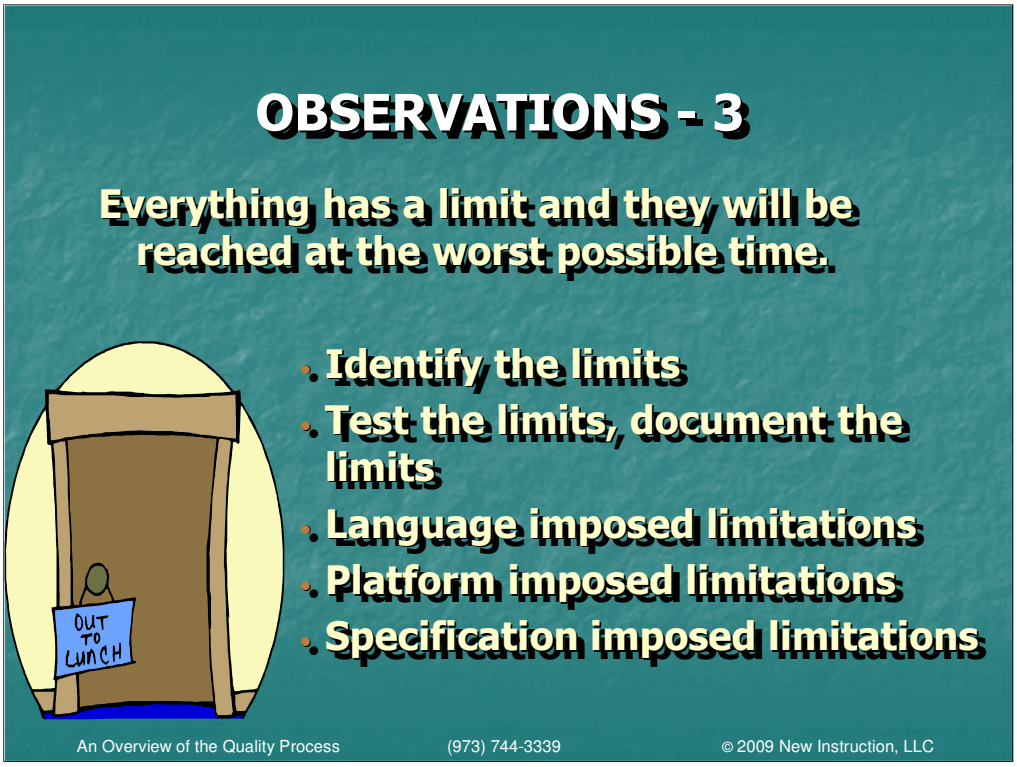
Nothing is obvious – specific examples are very helpful in defining requirements, building specs; graphics are particularly effective in communicating.

OBSERVATIONS - 2

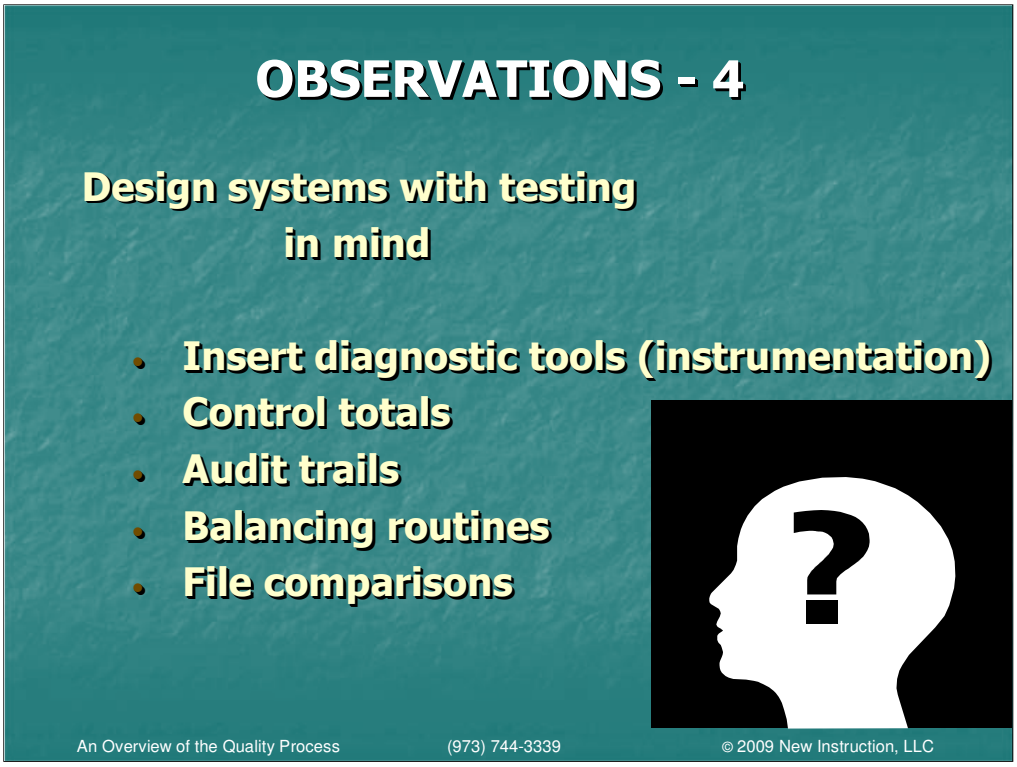
Do it twice

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

An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC



Limitations MUST be identified, documented, and tested. Instrumentation – alerts that a limit is being approached are particularly helpful.



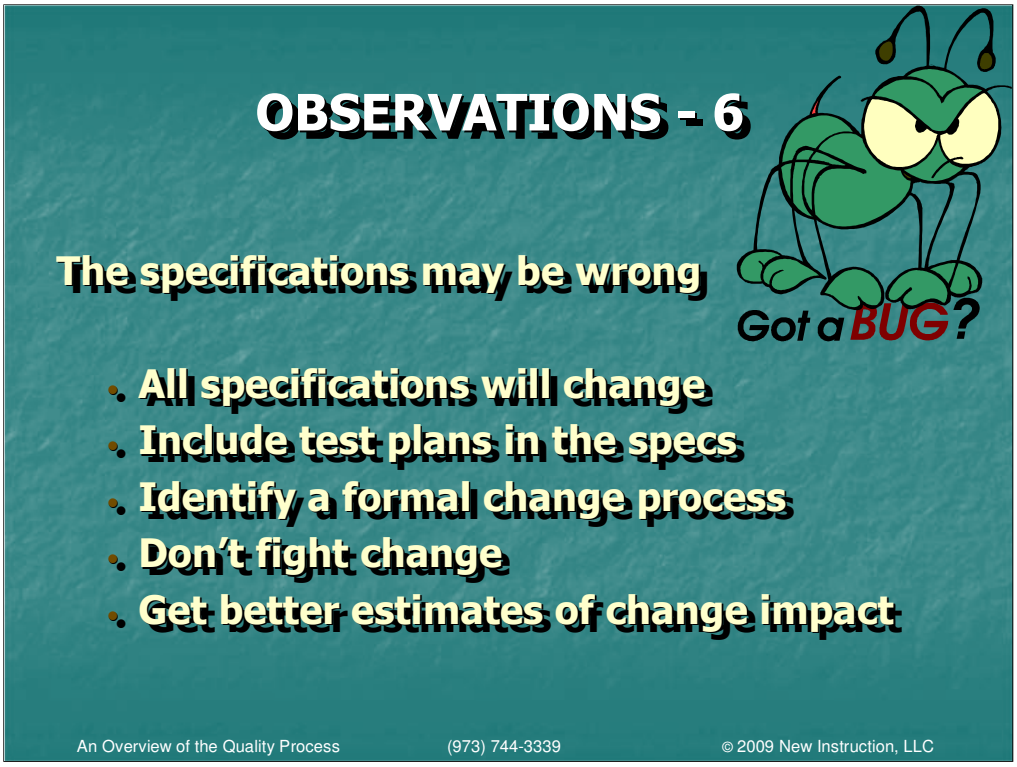
ALL systems have to be tested, so design system to support testing, build testing assistance into the system, for ex., balancing routines, diagnostics.

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

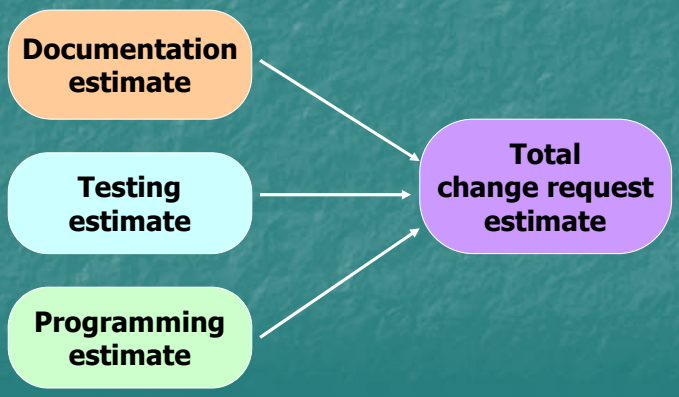
An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC



Specs may be wrong – techs should ensure requirements / specs are clear BEFORE they start coding.

CHANGE REQUESTS:

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



An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC

Grass roots approach to quality improvement WILL work – if everyone does just a tiny bit, overall quality WILL improve. Management dictates will NOT improve quality unless everyone does their bit to make improvements.

EVERYONE can start today to improve quality.

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."

An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC

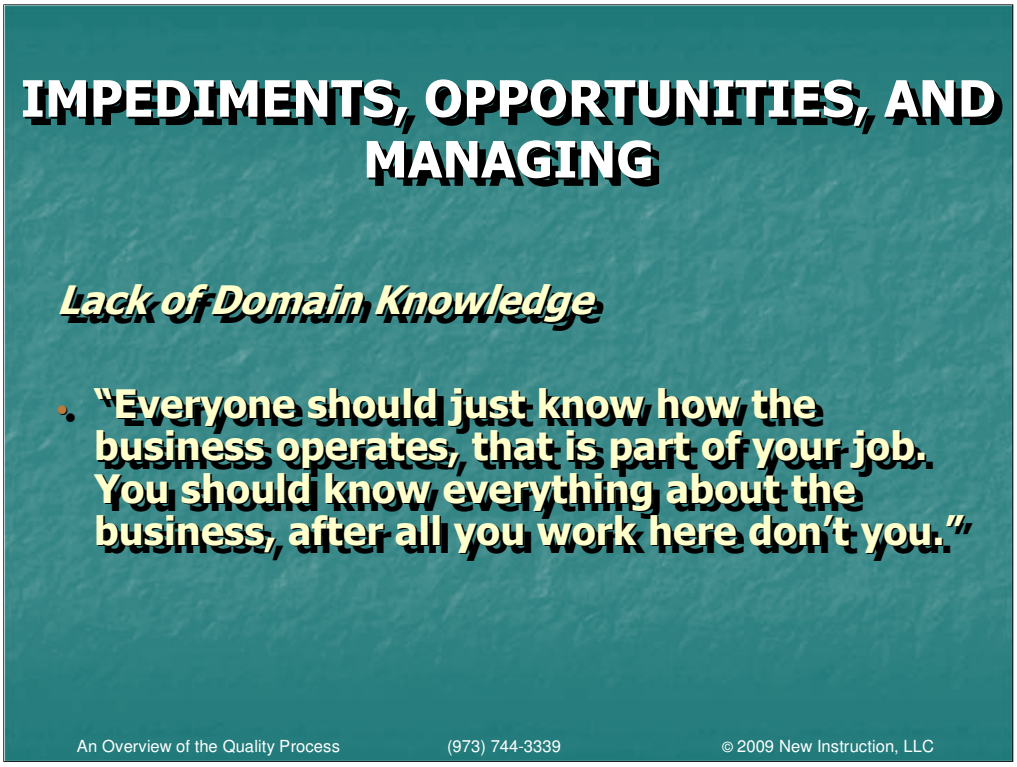
IMPEDIMENTS, OPPORTUNITIES, AND MANAGING

We're not ready and we need training

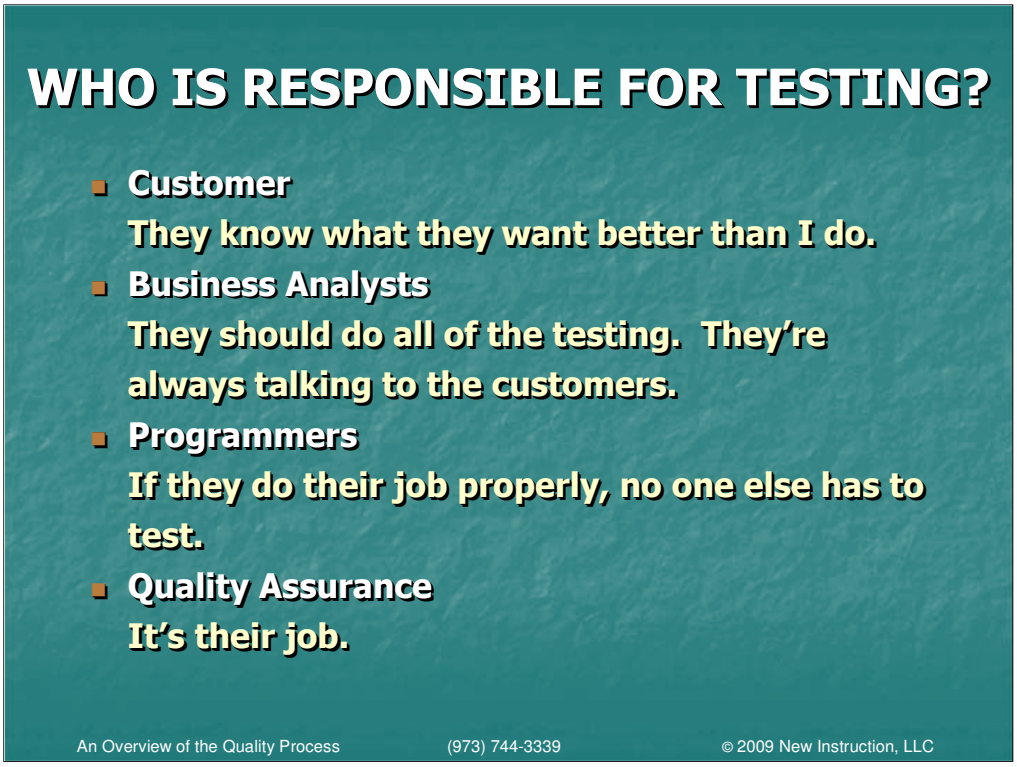
- "We have to get the rest of the shop in order, before we can consider quality improvement suggestions. When will we have time for training?"

An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC

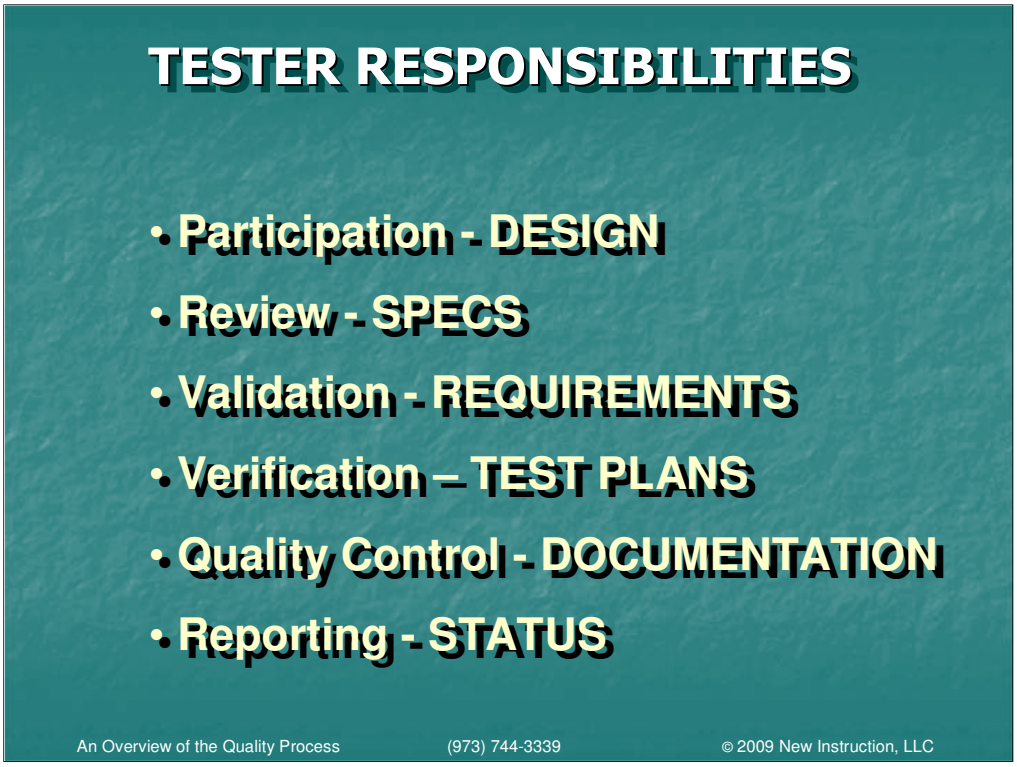
Unrealistic commitments are significant impediments to improving quality. Need to document impact of management commitments that are unrealistic. Always develop personal estimate for work to be done and compare it to provided commitments.



Use review sessions as opportunity for knowledge transfer / training sessions. Code review sessions should be used to share background information and application structure, interfaces, etc.



Everyone in the process is responsible for testing – customers, BAs, programmers, QA.



Participate in Design sessions to understand critical application aspects.

Review Specs for accuracy and thoroughness. JAD – Joint Application Design – key team members meet to come to same understanding of desired outcome of the project. Provides focus and input to a project so you don't get to the middle of the project and discover problems / disagreements.

Validate Requirements – prioritize requirements coming out of the JAD session. Validate to ensure that each requirement is testable (specific, measurable).

Verify Test Plans – initial creation of written unit and system test plans.

Quality Control of Documentation – does Doc meet organizational standards? Do test plans follow standards, use correct template?

Reporting - conveys status / progress to management.

Joint Application Design Sessions (JAD / JAR). Jim Putka, participant *(Dir. of Software Development, Indiana Farm Bureau Insurance)* – "...these can take up to a week, not necessarily easy to do – but VERY effective in ensuring completeness and correctness of requirements and in defect prevention in the long run. They include Business members, QA members in these sessions along with techs. Development staff trained in and use UML. Users also trained so they understand and know how to proof documents. Do JAD session, document results, and provide to business unit for validation. It's HARD to get started in this process, but well worth the effort. Having users involved from the beginning results in better defined workflows and more complete, correct user manuals, because people who will operate the systems have been involved in defining systems..."

From a class, pick THREE things to implement over six months, then pick the next area(s) to address and implement them.

Well defined Business Objectives – why are you doing this? What is the purpose behind this? Without this information people struggle to understand what needs to be accomplished

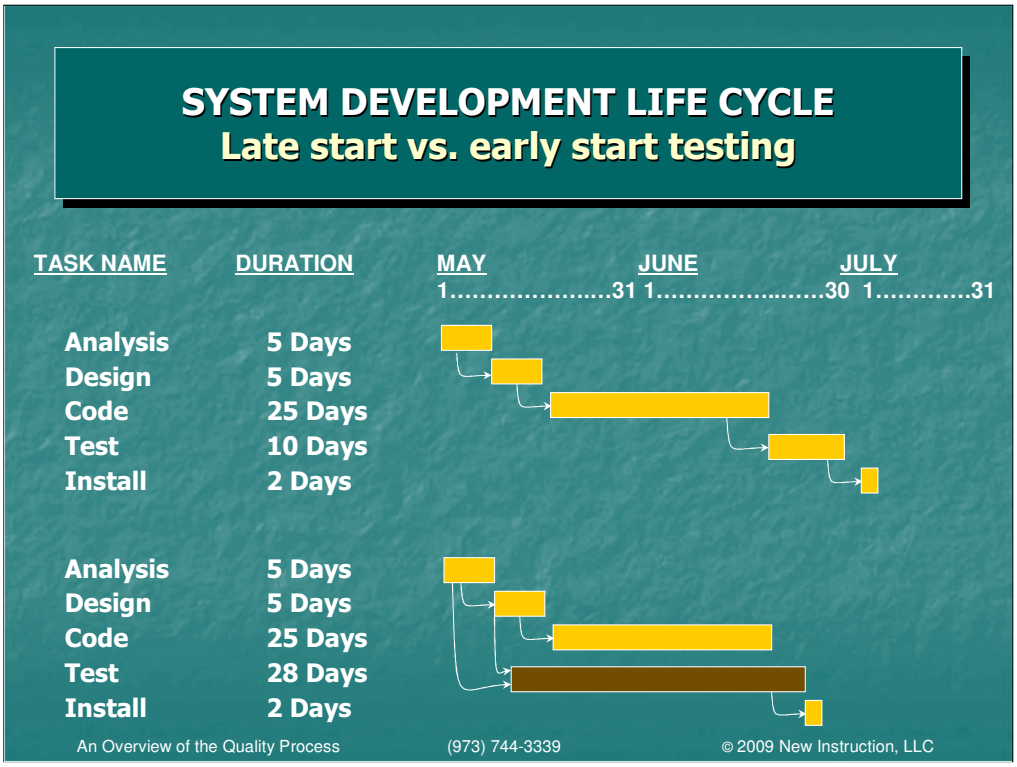
Written specs are the ONLY way to get accountability.

Written test plans are absolutely necessary – for use on initial delivery, then for verification of correctness after product changes – which may be more important of the two reasons

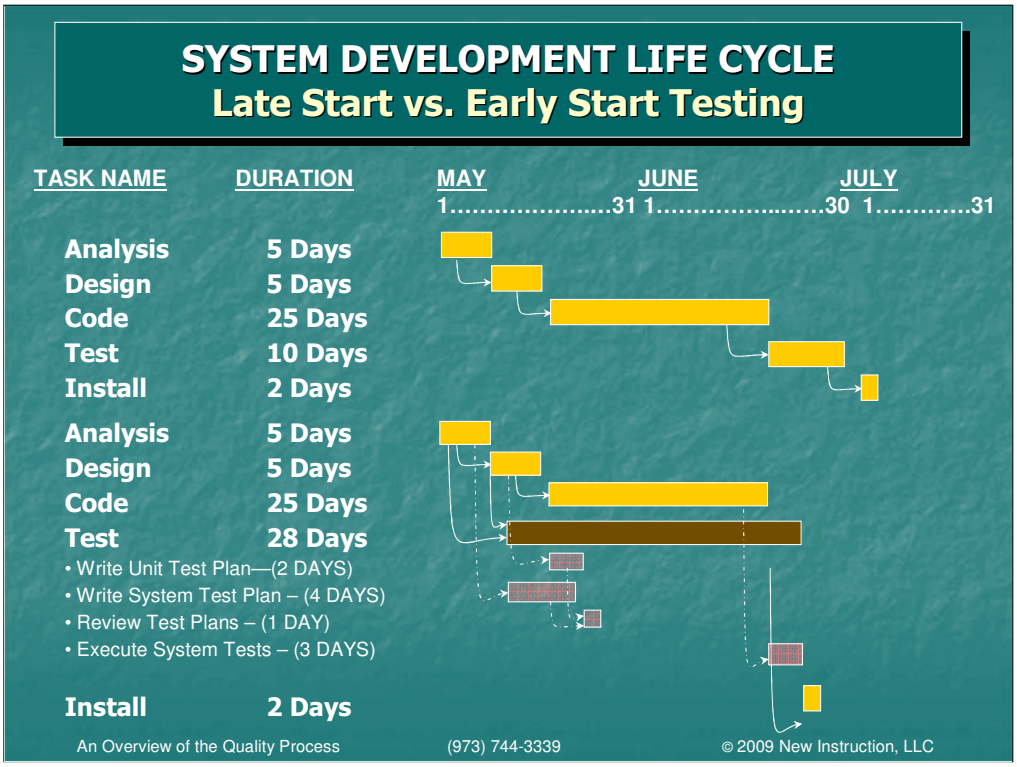
Risk assessment may include ensuring that critical functions are tested completely before a fix moves to production and then on-going testing of other functions after move to production

Allocate resources to automate testing. Don't have all automation knowledge / experience in one or two resources. As much as possible, ensure there is more than one person with knowledge of all functions / tools, etc.

Distribute analysis information to coders and testers at the same time, so that tests are written and ready to be executed once code is completed – including testing of pieces as code is completed.

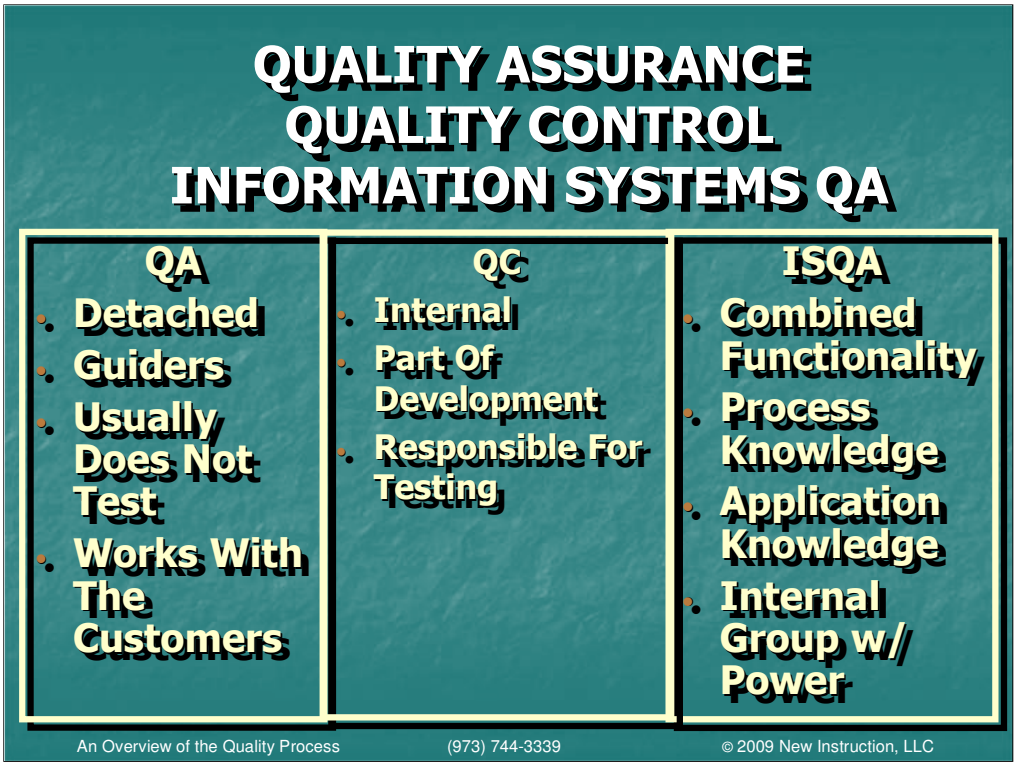


Having test plan developed during coding window allows sharing of test plan with coders to ensure they are coding to meet the test plan, reducing the iterations (code/test/ find defect/ fix defect /test) required.



Can compare unit test to system test plan – to ensure consistent understanding of functionality. Quality should be a factor through entire life cycle.

Tools – can't use ALL the tools available – find what will best fit organization needs – focus on ONE area.



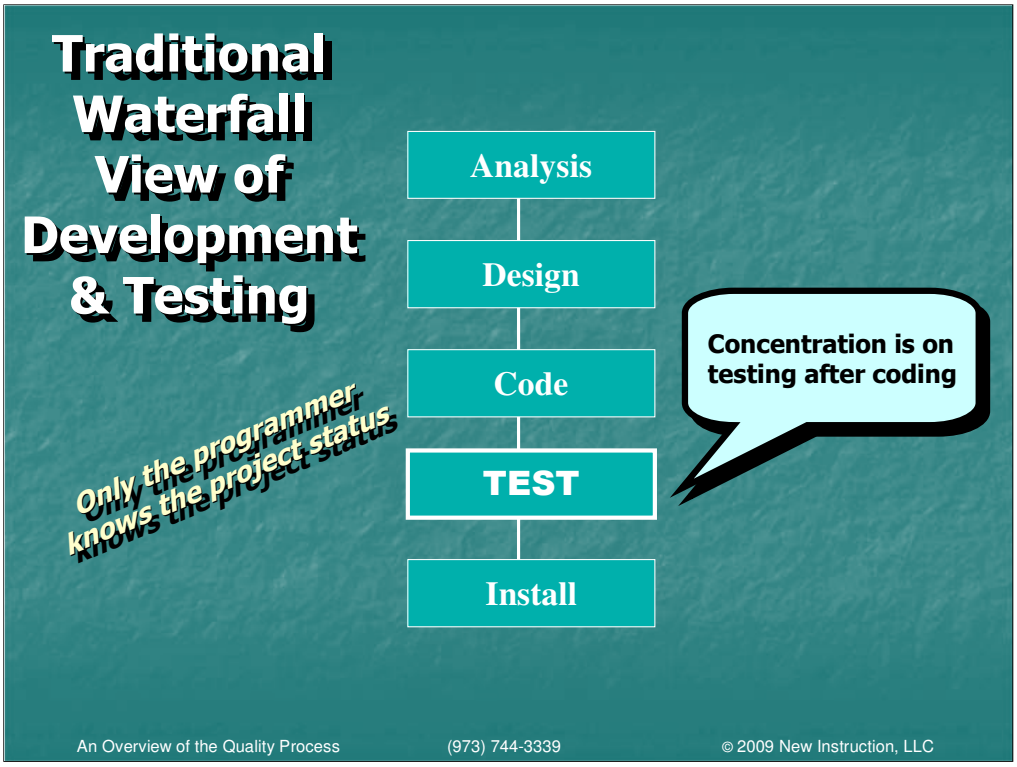
Terminology:

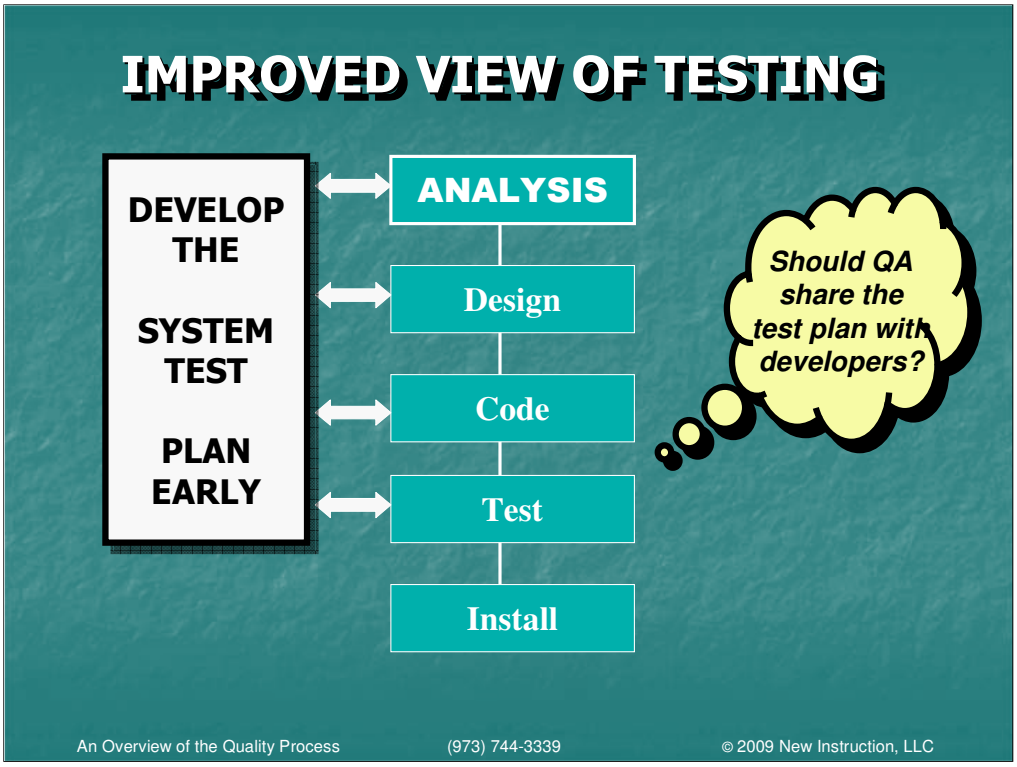
Quality Assurance – detached / independent from IT group. Doesn't do testing, doesn't have application knowledge. Recommend tools, process, procedures.

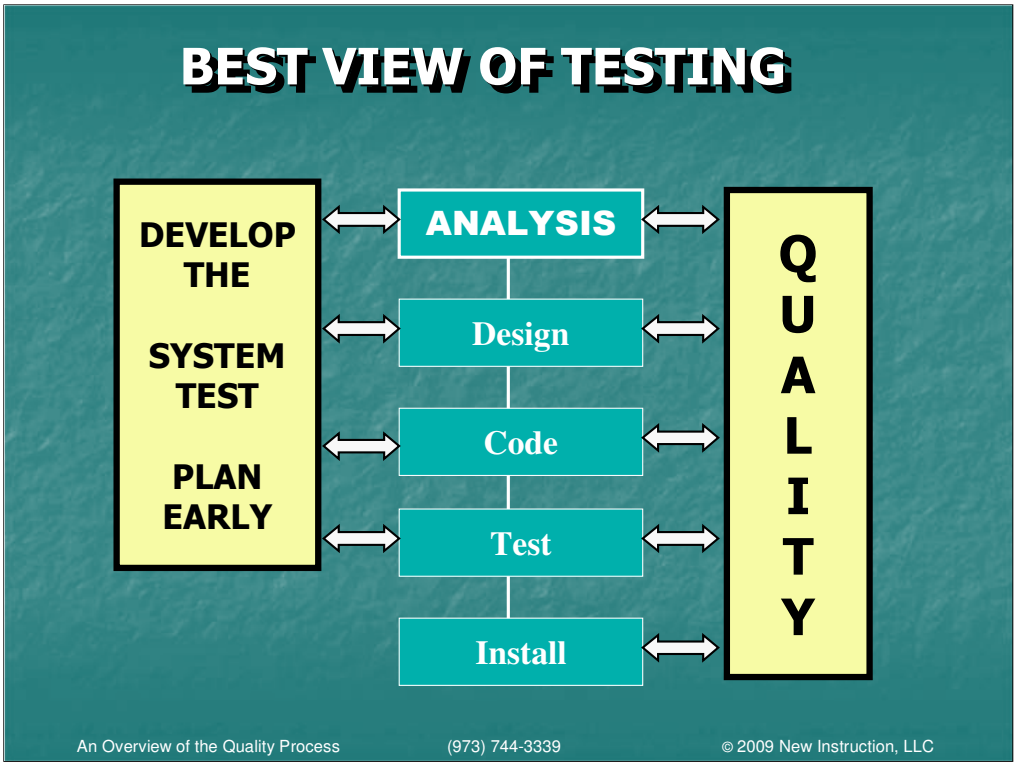
Quality Control – does testing, part of IT umbrella, have application knowledge.

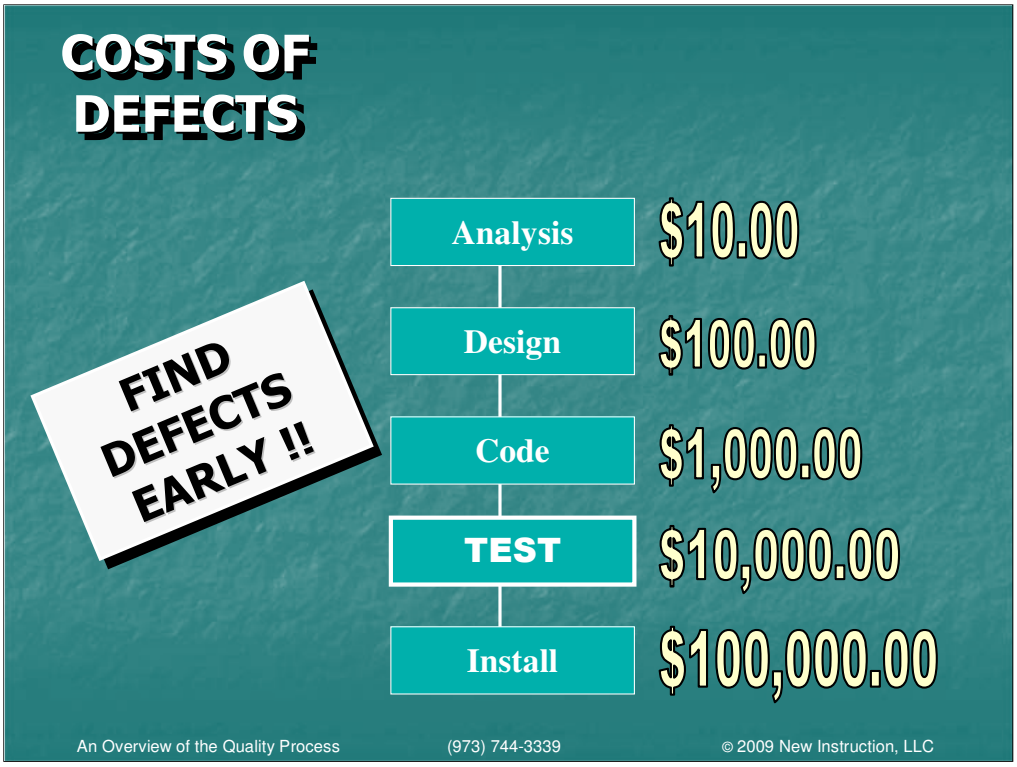
Information Systems QA – does both functions. Most organizations have ISQA.

QA / ISQA should NOT report to Development Manager to allow freedom from conflicting priorities.

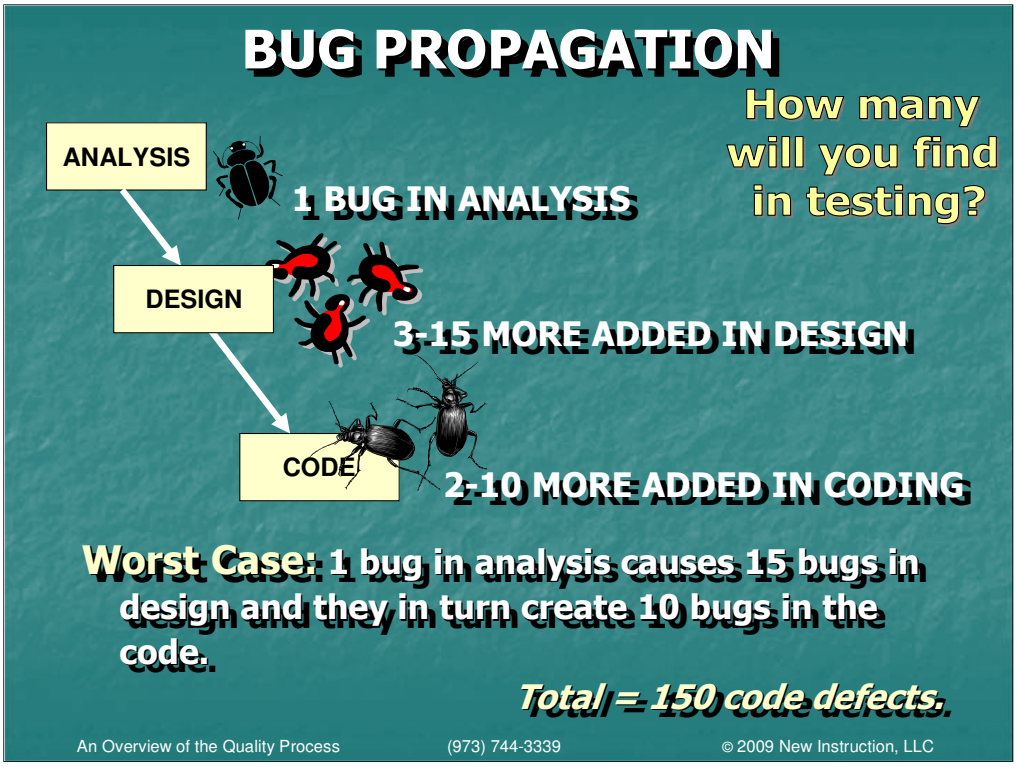

Traditional Waterfall View of Development & Testing
Analysis
Design
Code
TEST
Install
Concentration is on testing after coding
Only the programmer knows the project status
An Overview of the Quality Process
(973) 744-3339
© 2009 New Instruction, LLC


IMPROVED VIEW OF TESTING
DEVELOP THE SYSTEM TEST PLAN EARLY
ANALYSIS
Design
Code
Test
Install
Should QA share the test plan with developers?
An Overview of the Quality Process
(973) 744-3339
© 2009 New Instruction, LLC

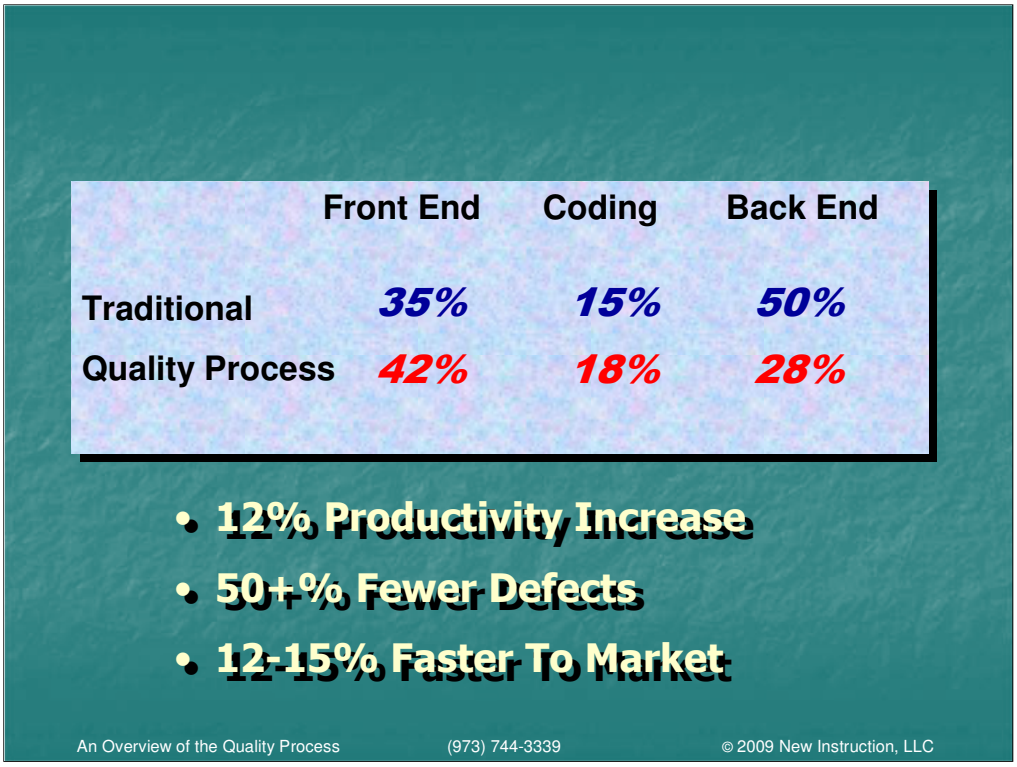

BEST VIEW OF TESTING
DEVELOP THE SYSTEM TEST PLAN EARLY
ANALYSIS
Design
Code
Test
Install
QUALITY
An Overview of the Quality Process
(973) 744-3339
© 2009 New Instruction, LLC



Increases tenfold for every life cycle passed without finding the defect – paying team twice for every stage that has to be repeated. Worst case scenario – don't find until it's moved to production.



Undiscovered bugs multiply across life cycle phases. Testers usually can't find all bugs introduced during the life cycle.



Traditional life cycle has 35% planning, 15% of time spent in coding, 50% in testing/installation.

Improved life cycle spends 42% of time in planning, 18% in coding, and 28% in testing/installation with a 12% productivity increase along with a 50% reduction in defects – because defects are identified and removed before they can multiply.

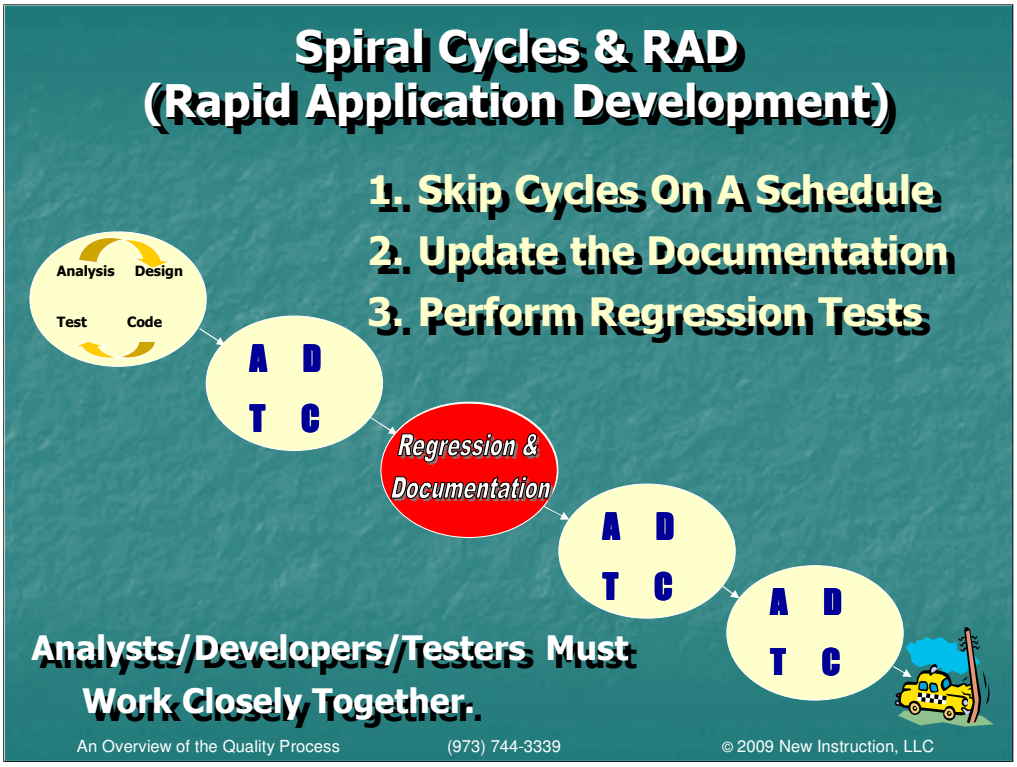
It may cost less to leave the defects out of the system, than to pay to put them in, pay to find them, and then pay to take them out again.

We are still working out the numbers?

An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC

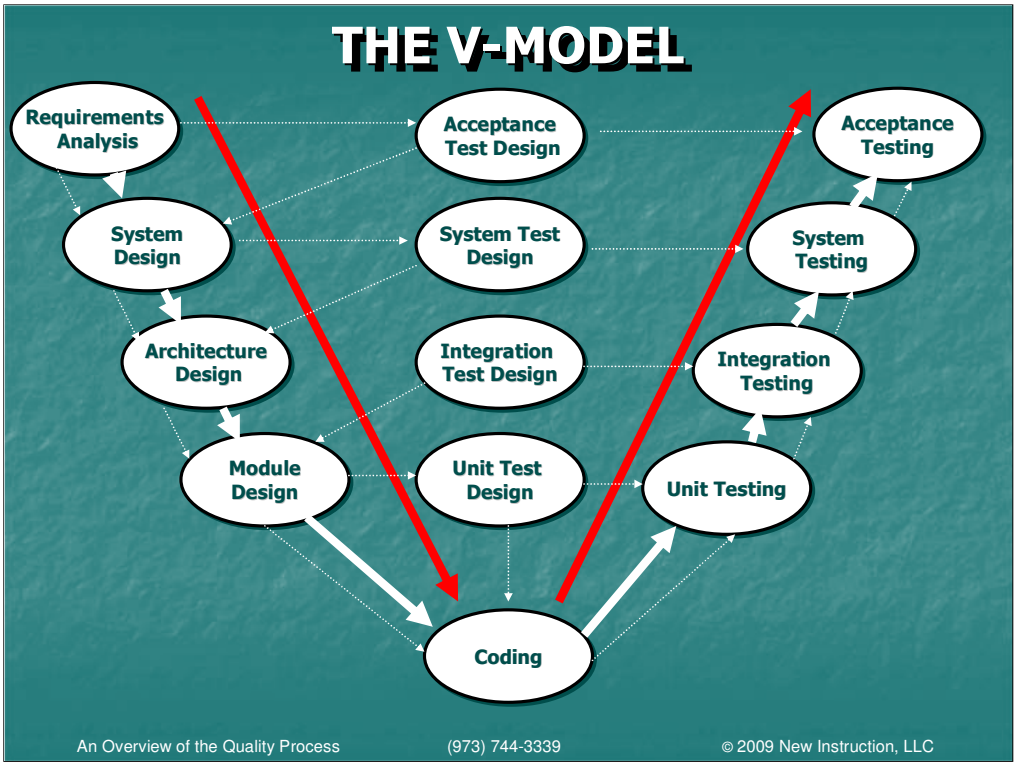
Tasks in 40 – 80 hour range.

Deliverables are defined to be measurable, reviewable, achievable – and reviewed promptly.

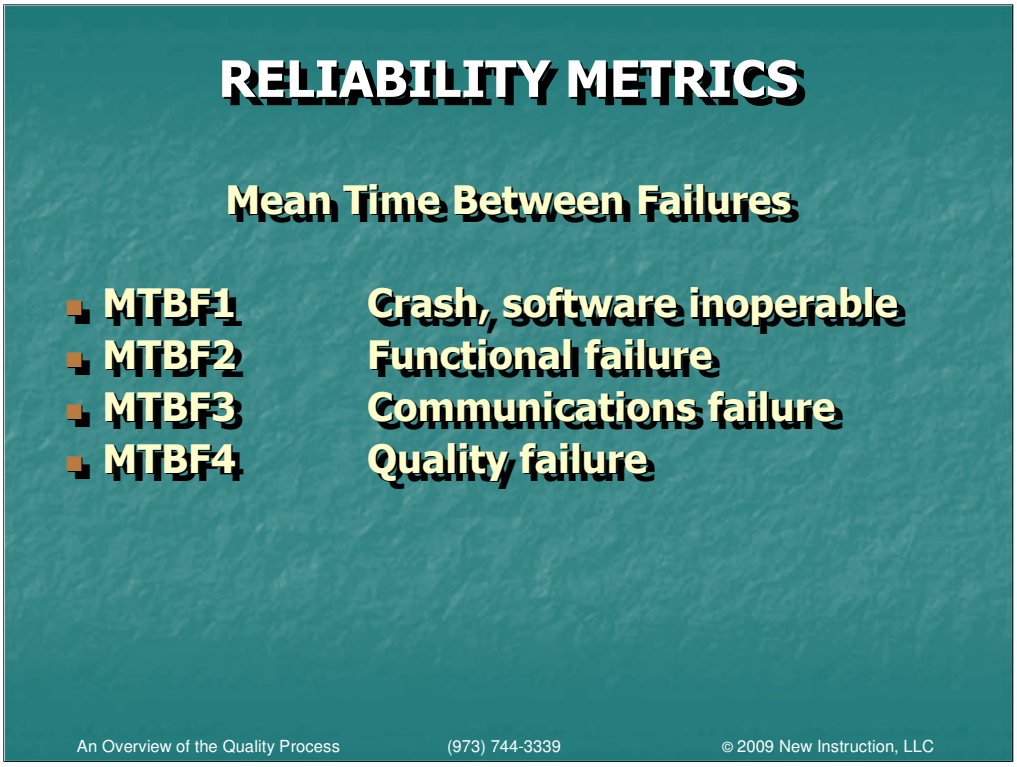


Development becomes a series of small lifecycles – Analysis, Design, Code, Test, Deliver in very small time blocks.

Often ends in a crash – defect was introduced in system – and don't know which cycle introduced it. Must skip cycles on a schedule – to do regression test and documentation. This gives a baseline to help identify where the defect was introduced – like a balance point in a checkbook..



Working together to do early testing is key to the model.



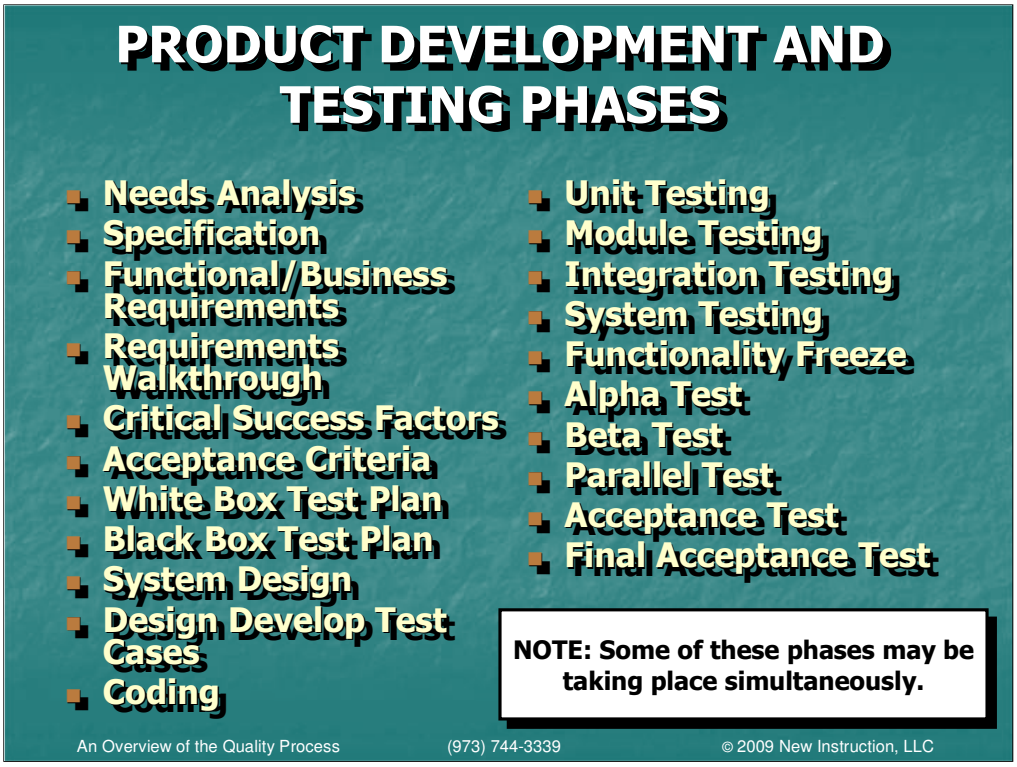
Used to measure quality improvement over time.

RELIABILITY METRICS

Mean Time To Repair

- MTTR1 — Actual time to fix
- MTTR2 — Total time in queue

An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC



White Box – look at brake pads, drums.

Black Box – get in car and go 60, and then apply brakes – what happens?

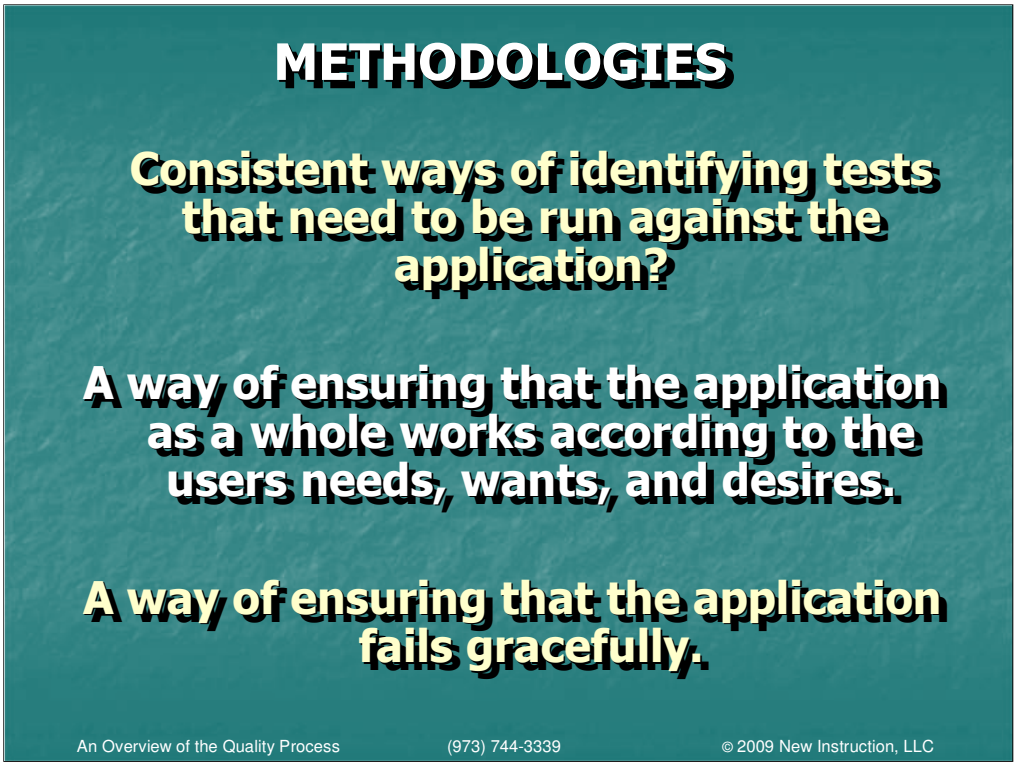
Module testing / string testing – making sure that a module works with those directly linked to it.

Final Acceptance Test – usually associated with payment.

Section 2

Test Methodologies and Checklists

An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC



Random testing typically hits about 30% of an application. Using a methodology is a way to ensure that entire system is tested, works, and 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.

An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC

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.

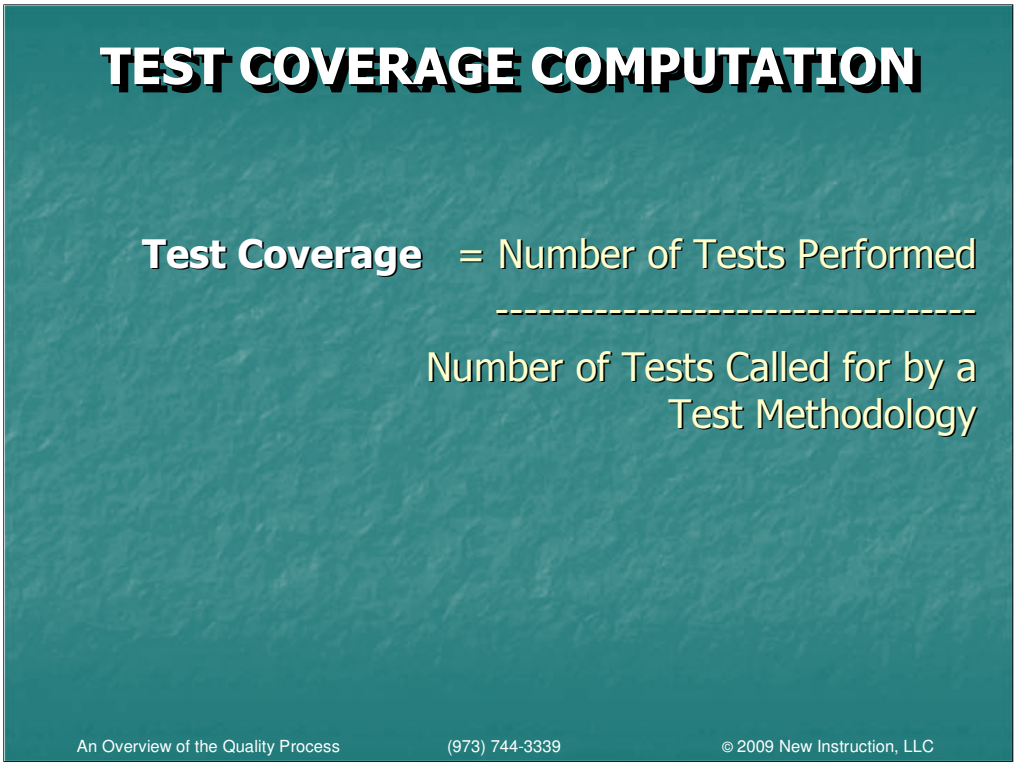
An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC

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

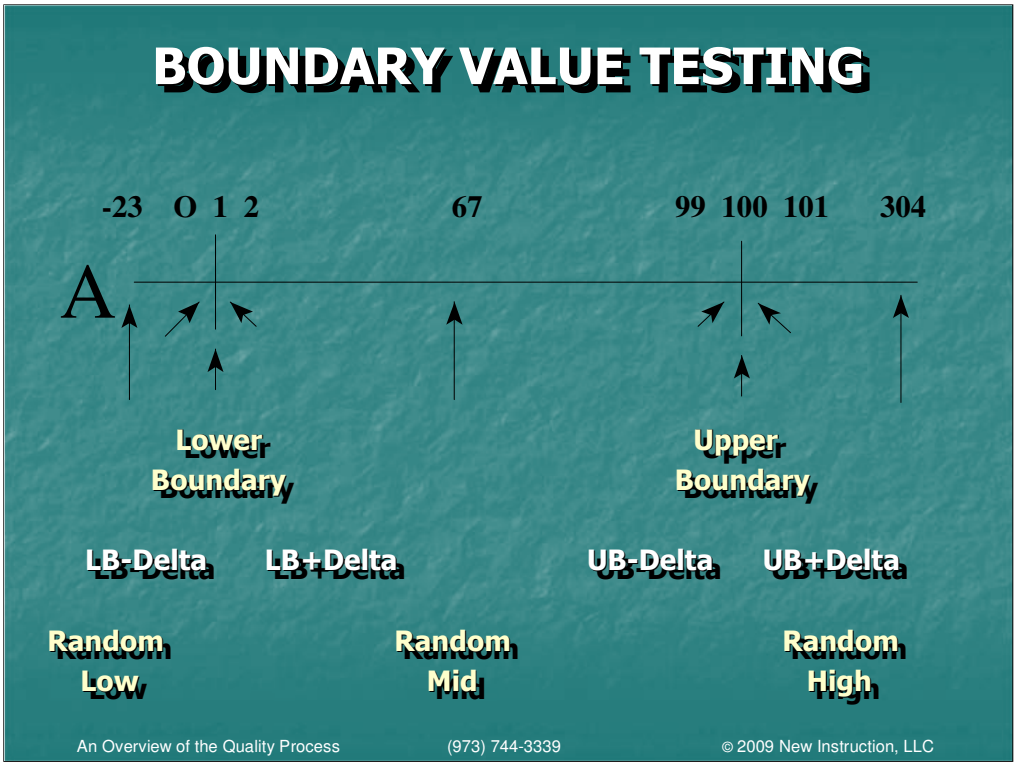
- VISUAL INSPECTION OF MY CAR'S BRAKING SYSTEM.
- ROAD TEST MY CAR ON THE FREEWAY.

Which one should be completed first?

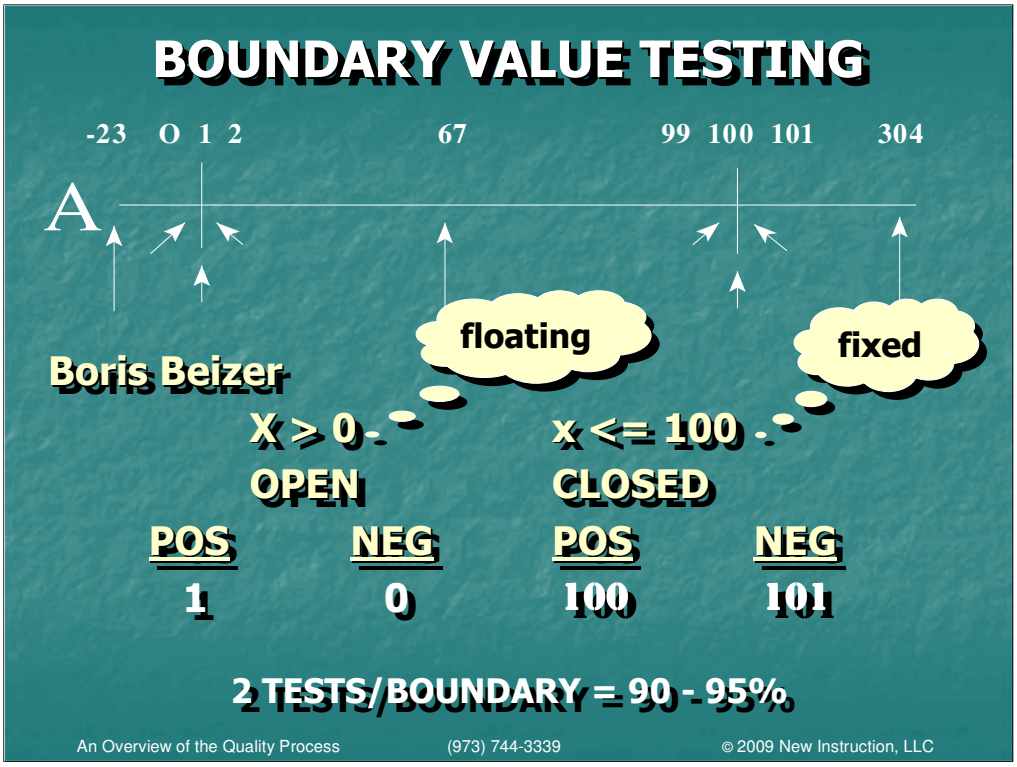
An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC



Measures out of the total number of possible tests, how many am I running. Number of tests performed divided by the number of tests called for by a test methodology.



Done because defects tend to cluster around boundaries / limits. For example, in a field where values 1 to 100 are valid, test values right at and next to boundaries (0,1,2,99,100,101) and then do equivalence class partitioning – test below, inside, and above the range for every range tested (so for this example, test the lowest possible number you can think of, the highest possible number you can think of, and the middle of the range: -23, 67, 304) Running these 9 tests should find 99% of any defects associated with boundary conditions.



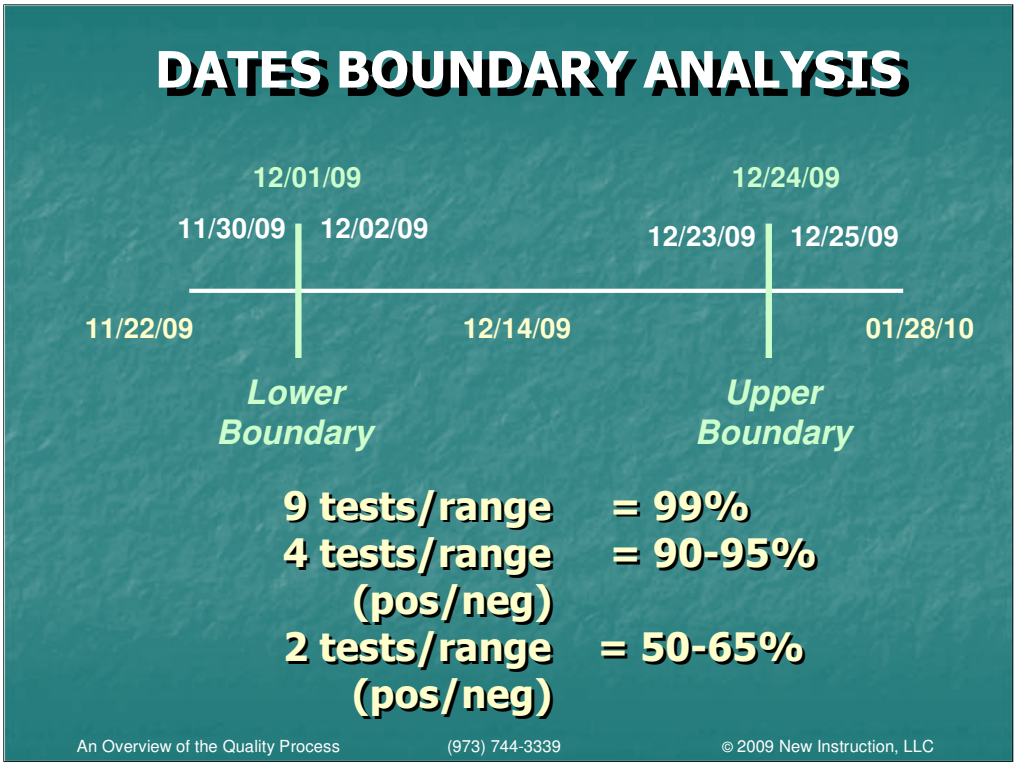
Boris Beizer found that conducting positive and a negative test for each boundary condition 101 – four tests would result in 90 – 95% defects hit. Floating = anything greater than a point, Fixed = anything with exact limit (< = 100), so test 1,0,100.

Boundary Value Testing with Independent Numeric Fields

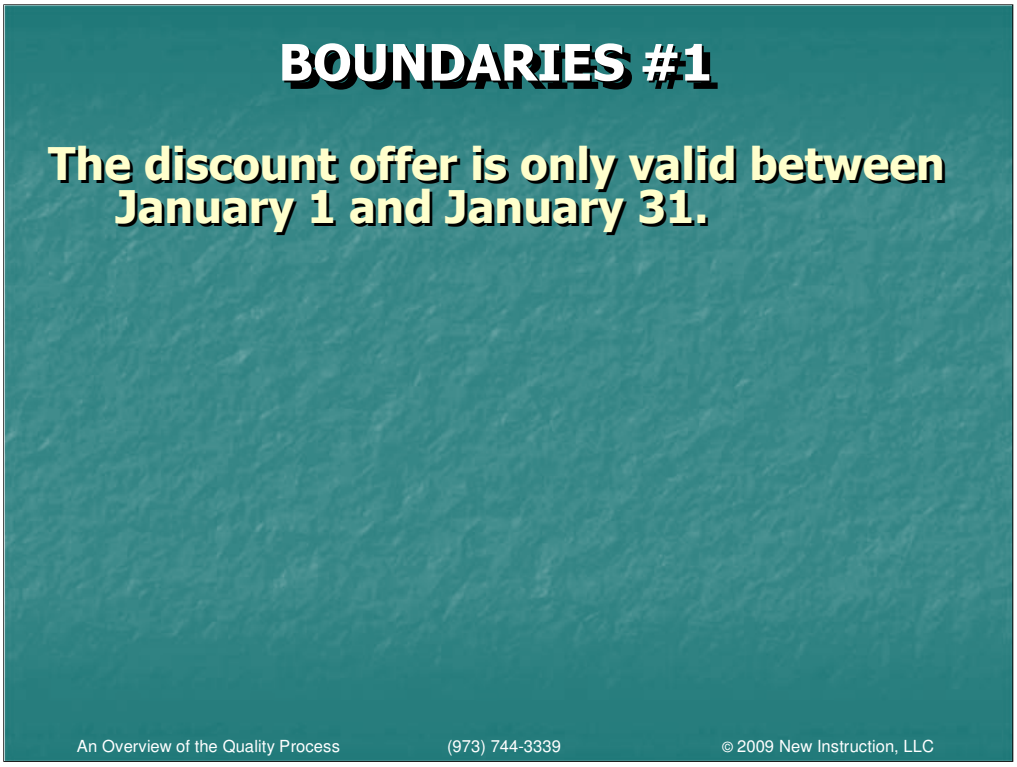
Input A	Input B	Expected C
-23	3	Error
0	3	Error
1	3	3
2	3	6
67	3	201
99	3	297
100	3	300
101	3	Error
304	3	Error

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

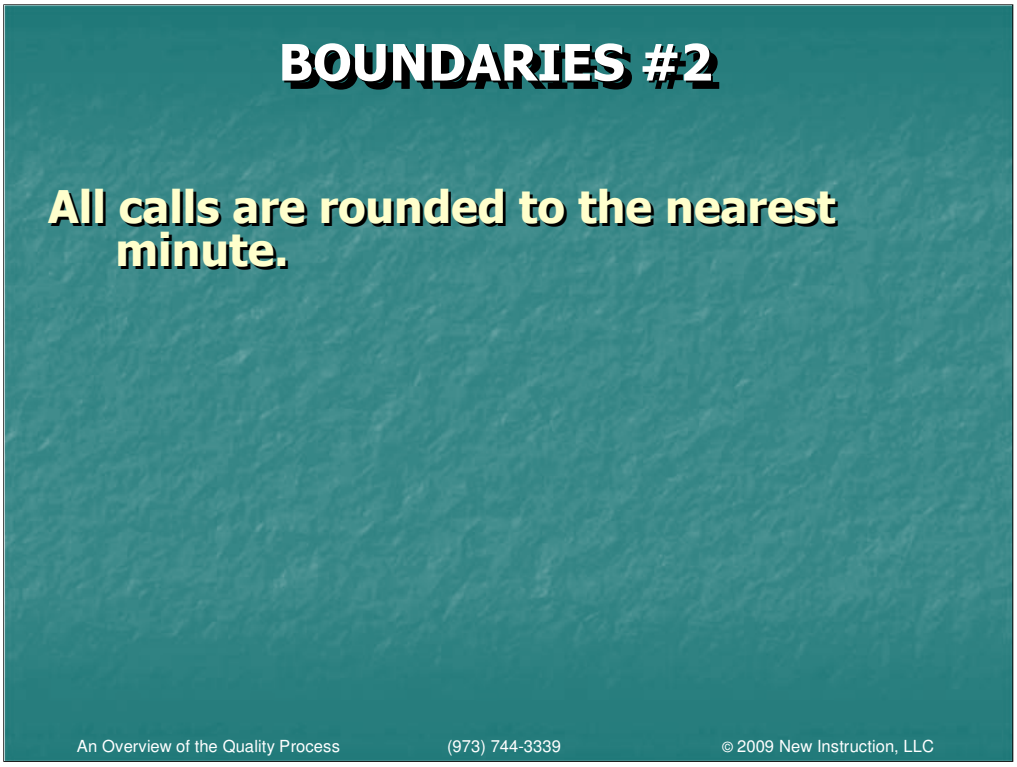
An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC



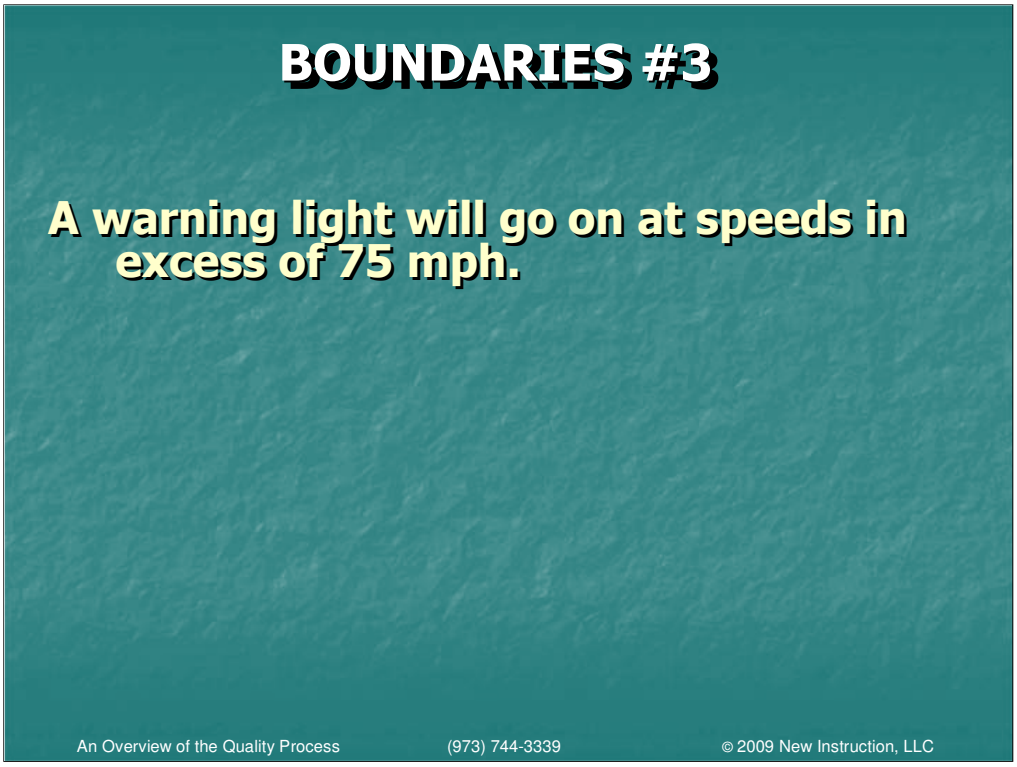
Apply same theories / tests to date fields.



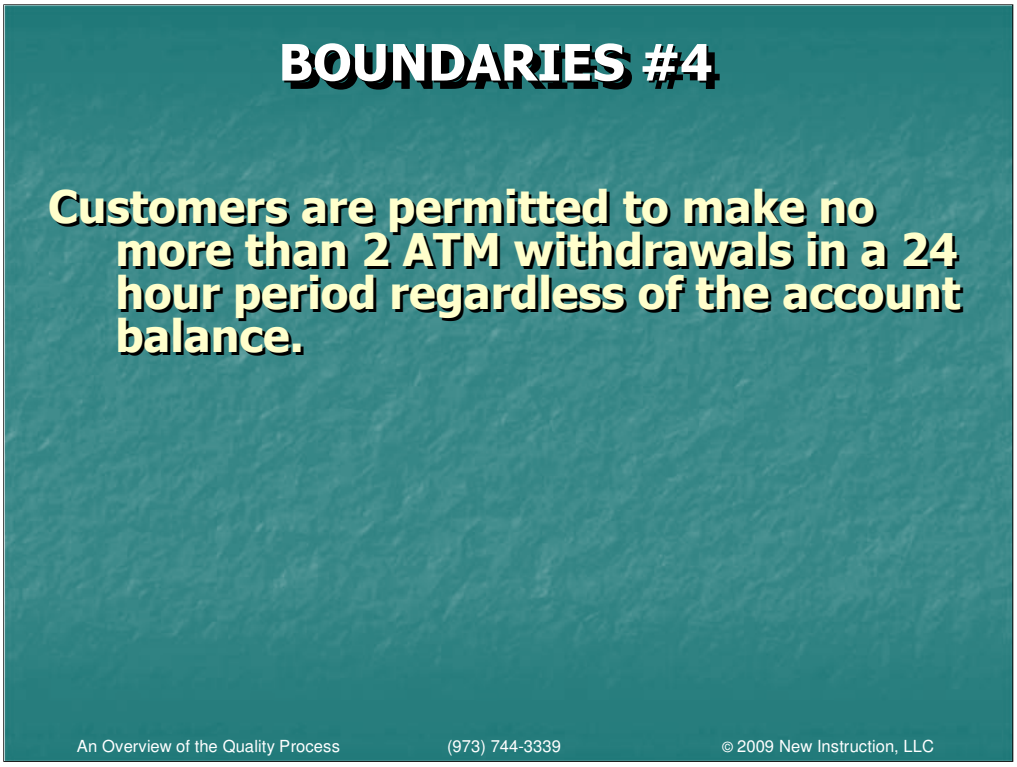
Boundary analysis could bring up questions that need to be answered during requirements / spec phase – for example, if a boundary date is a holiday or weekend, does special logic need to happen? Exactly what date is the boundary defined by – system date, effective date? Exactly what is the boundary date? (In example, between 1/1 and 1/31 – are 1/1 and 1/31 valid or not?)



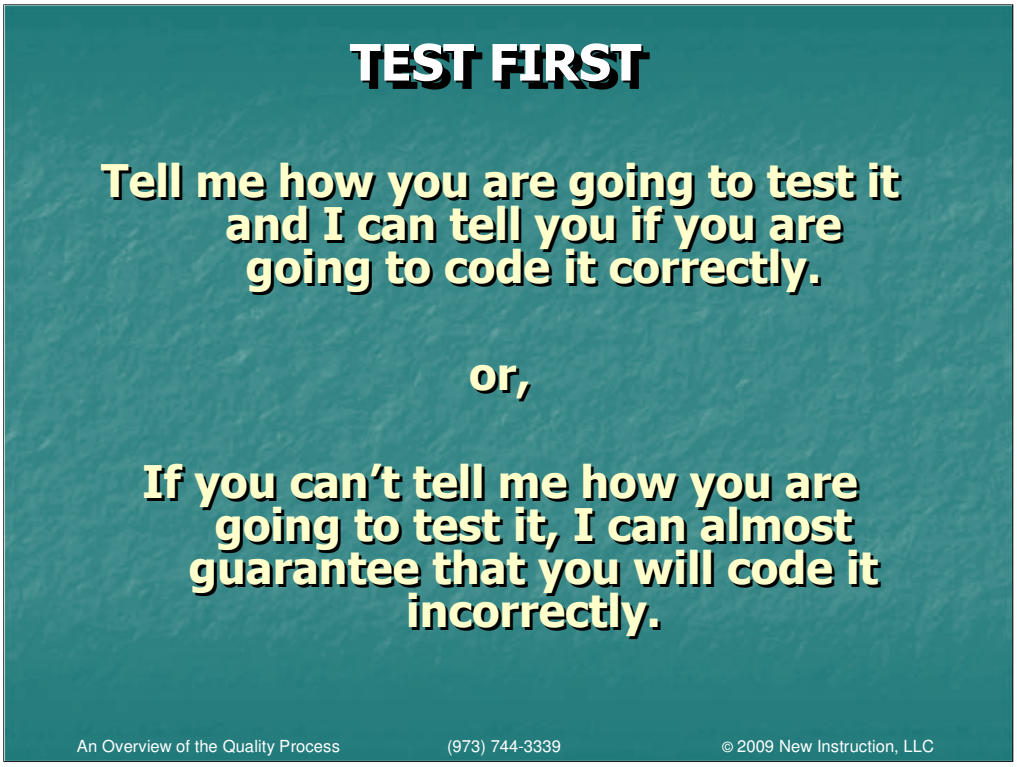
Test 0, 59, -1, 1, 58, 60, 30. (Test -1 to ensure graceful failure! Need to know how to round 30 – up or down!) Are first and subsequent minutes rounded the same way?



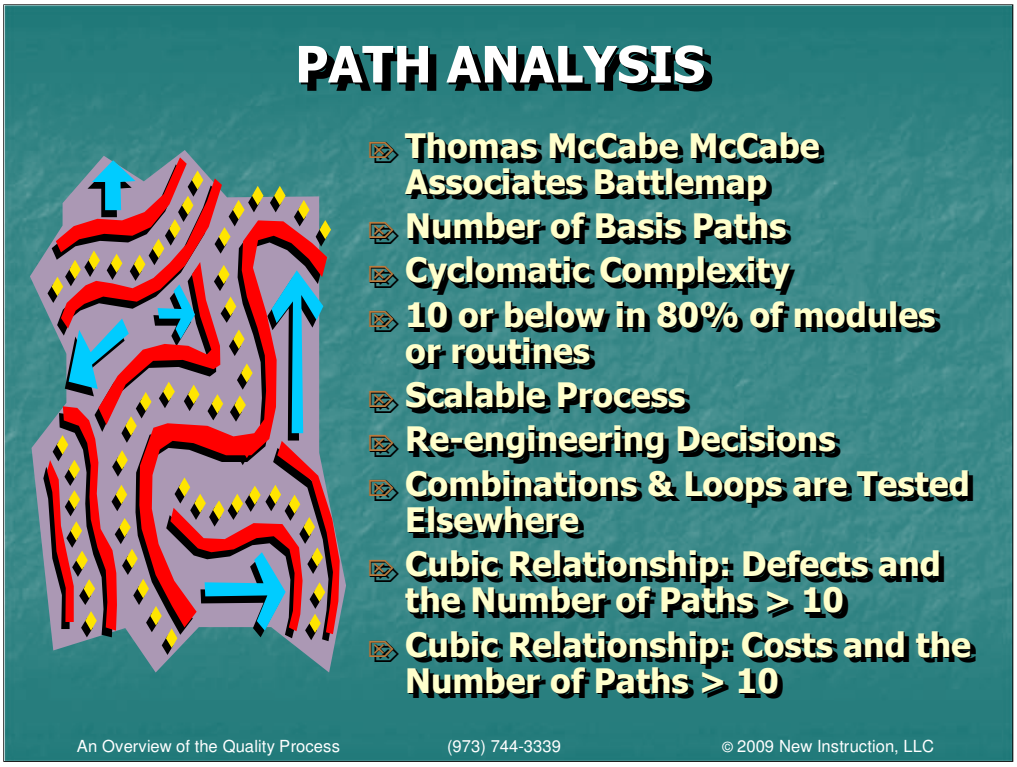
(Boris Beizer says we should test 76, 75, 74 (76 and 75 would be critical tests, 74 extra verification) -76 would be a great test as well – because it checks to ensure that the field is checking for absolute values. The idea of testing is to identify failure points, so sometimes you have to test things that might not make sense.



Looking at an example of tests to be conducted can be very effective in illustrating requirements and helping to clarify requirements. When you start to apply test definition to a requirement, you can often identify holes in the requirement definition.



If you can tell me how you are going to test it, I can tell if you are going to code it correctly!



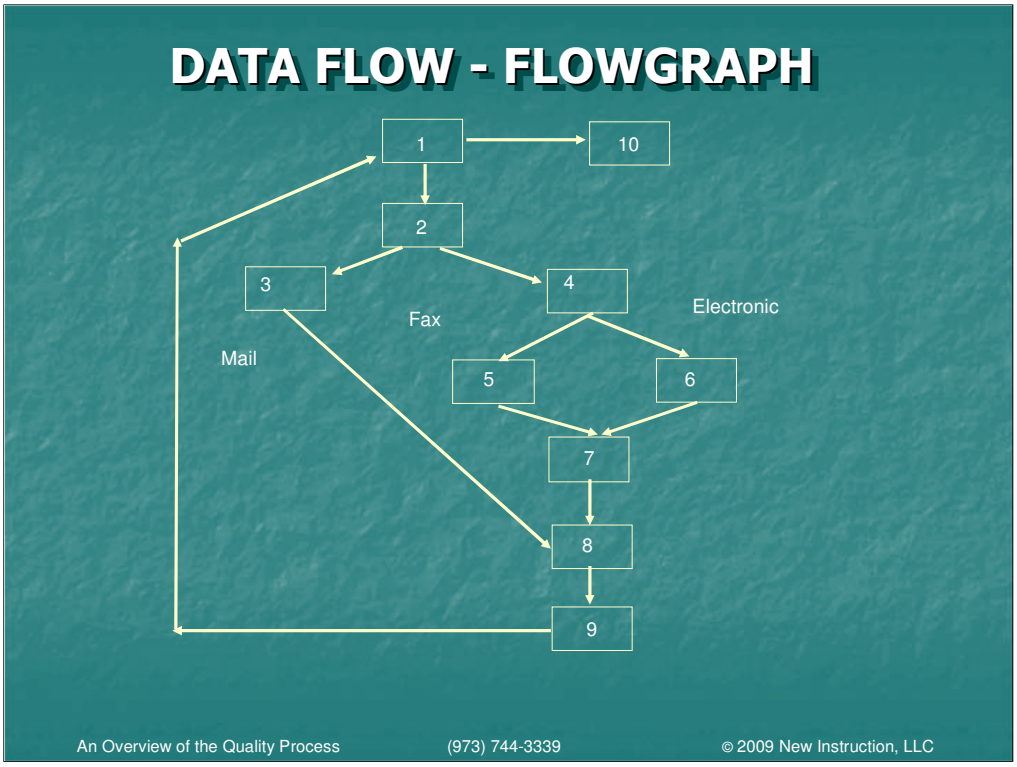
Developed by Thomas McCabe. Battlemap – automation tool. Measures cyclomatic complexity – the number of different paths available to get through any given set of modules – increased complexity leads to increased defects. When the number of paths is greater than 10, the number and costs of defects multiply in a cubic relationship. The process can be used in white or black box environments (scalable). Helps in re-engineering decisions. In Agile, this is helpful in refactoring. Complex logic is broken into smaller, less complex pieces.

PATH ANALYSIS

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

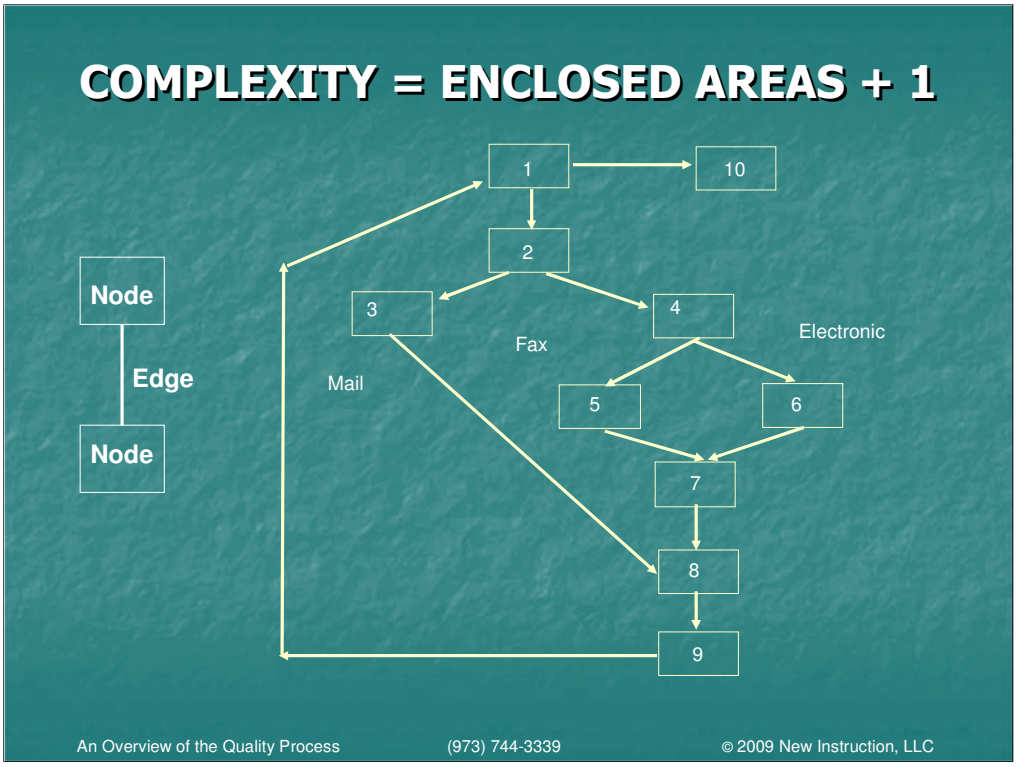
```
1   WHILE NOT END-OF-FILE READ TRANSACTION
2   IF ORDER = "MAIL"
3       ADD 1 TO MAIL-CTR
4           ELSE IF ORDER = "FAX"
5               ADD 1 TO FAX-CTR
6               ELSE ADD 1 TO ELEC-CTR
7           ENDIF
8       ENDIF
9   ENDWHILE
10  RETURN
```


An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC

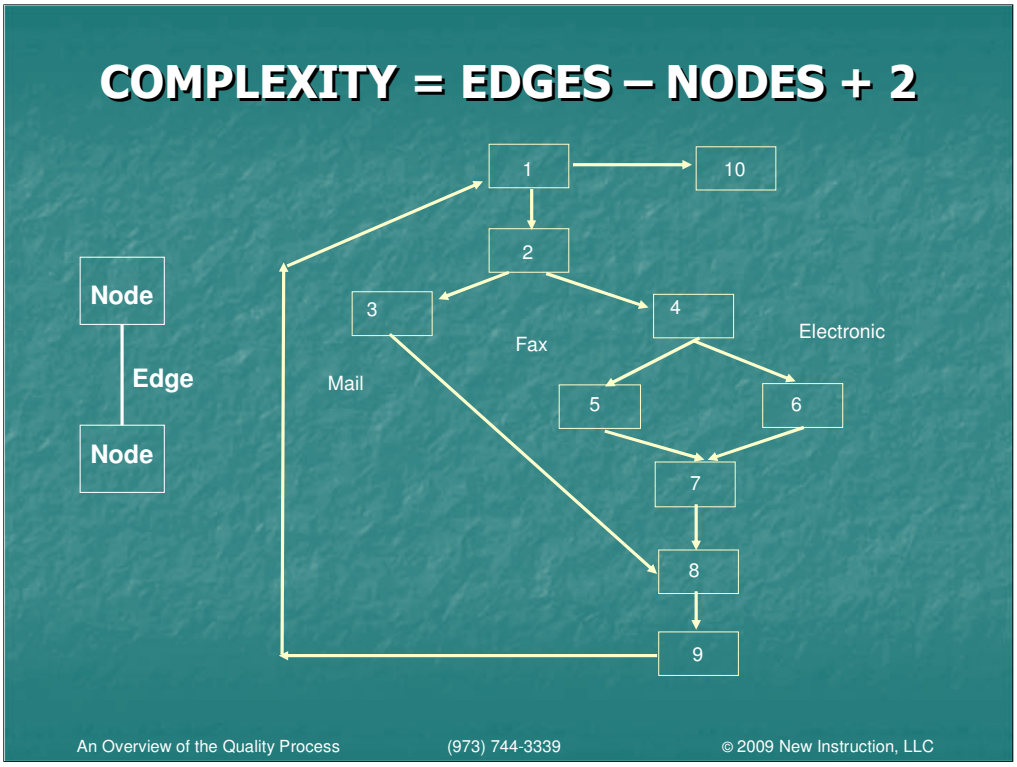


Basis paths are counted multiple ways. Try three different methods – because using only one or two will not guarantee correct results. If done correctly, each method should result in the same number of paths, run one test for each path.

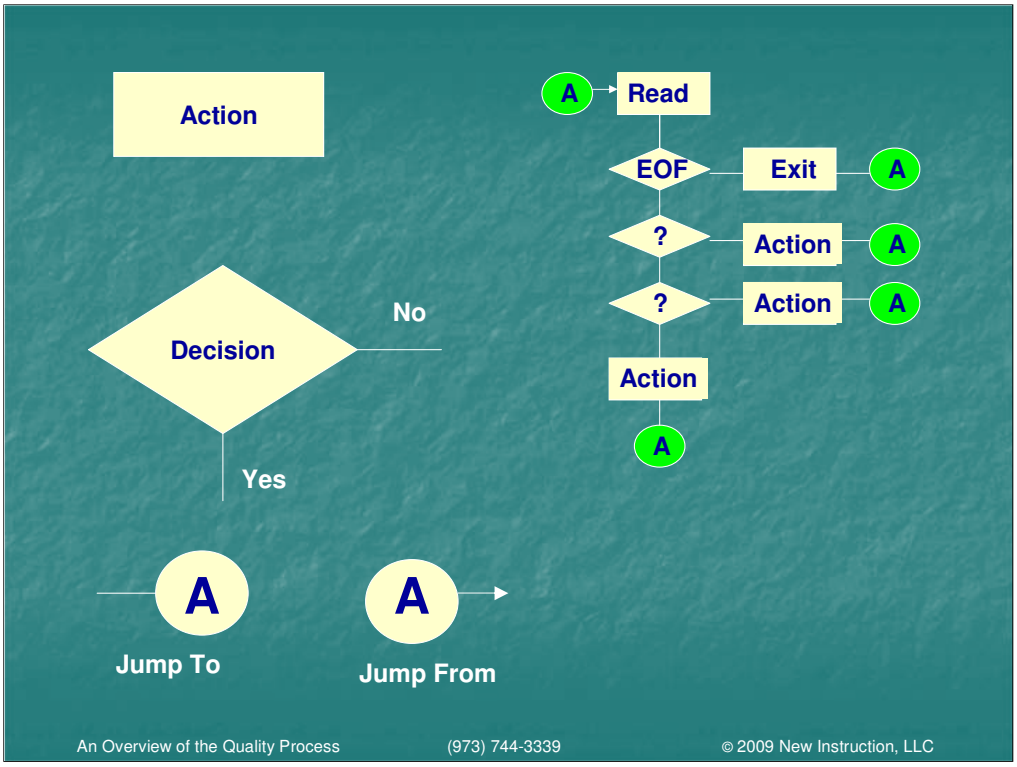
Trace a path through the application diagram, count the number of possible paths.



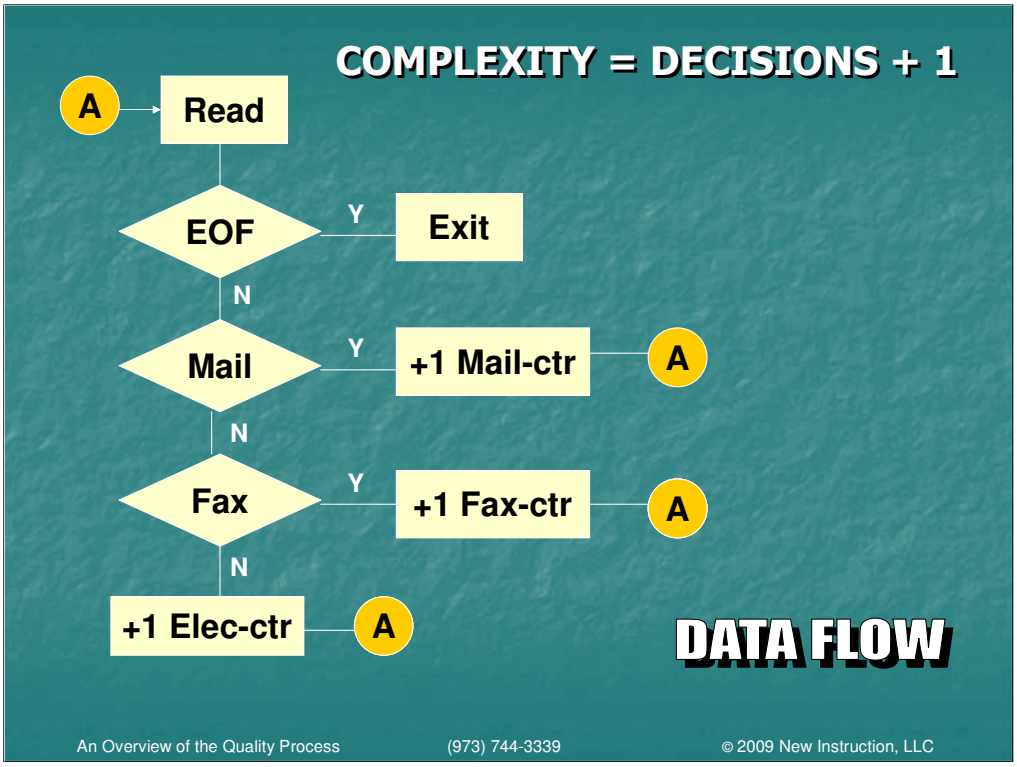
Count areas created by flow lines, and also count the entire area – total indicates the total number of test cases to be run



Count ‘edges’ (lines connecting boxes/nodes), count ‘nodes’ (boxes) – calculate Basis Paths = Edges – Notes + 2.



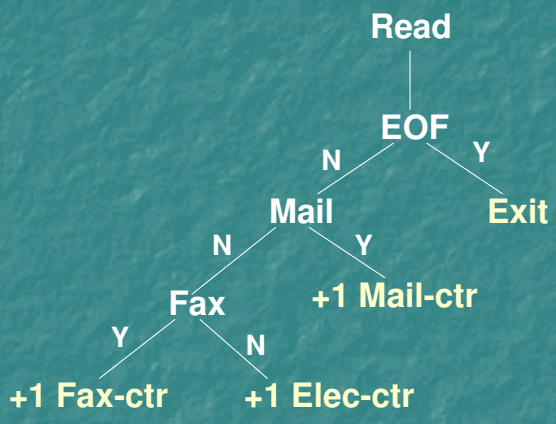
Count decision boxes – every place where there are two possible choices / paths.
Basis Paths = Decisions +1.



Count endpoints from decisions. Basis Paths = the number of end points.

Numbers would not agree if diagram is wrong, if the spec is wrong, or if you can't count - or where one end point is arrived at from two flows – as shown on the second data flow slide in the deck.

DECISION TREE



COMPLEXITY = ENDPOINTS

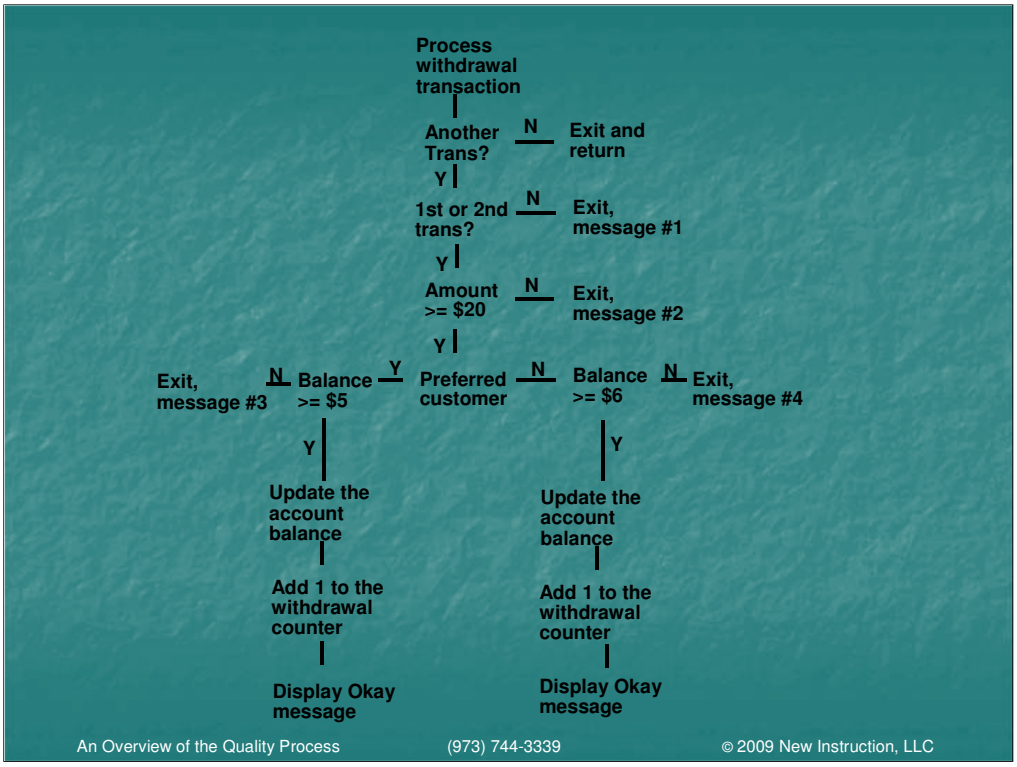
An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC

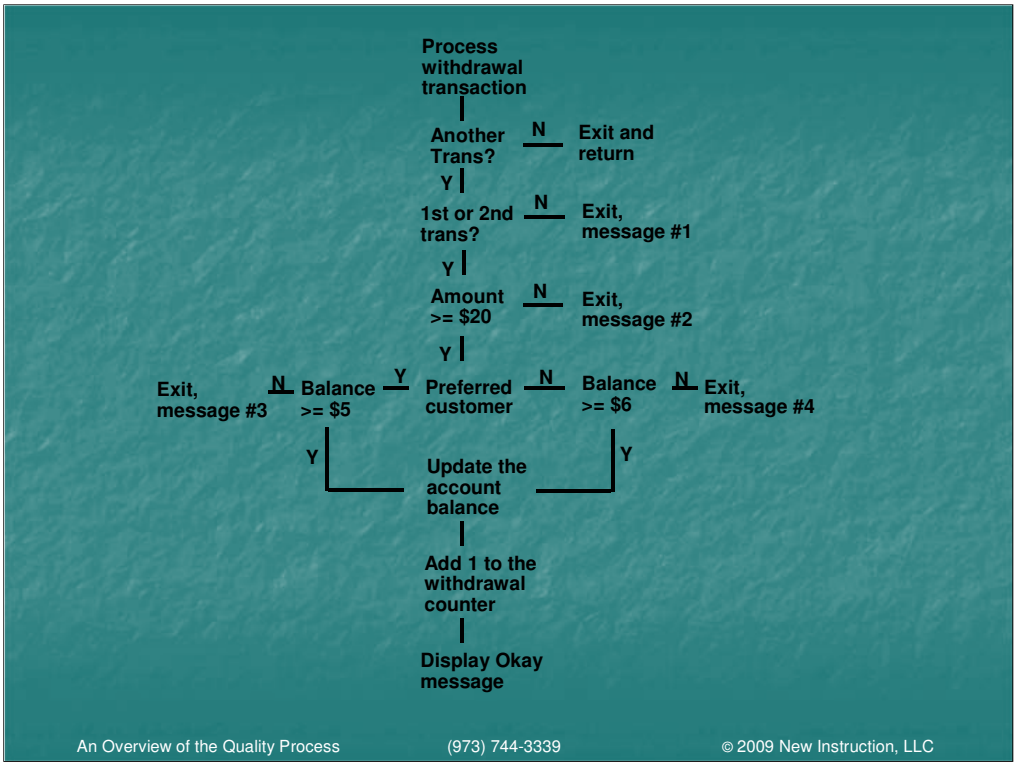
ATM Withdrawal Example

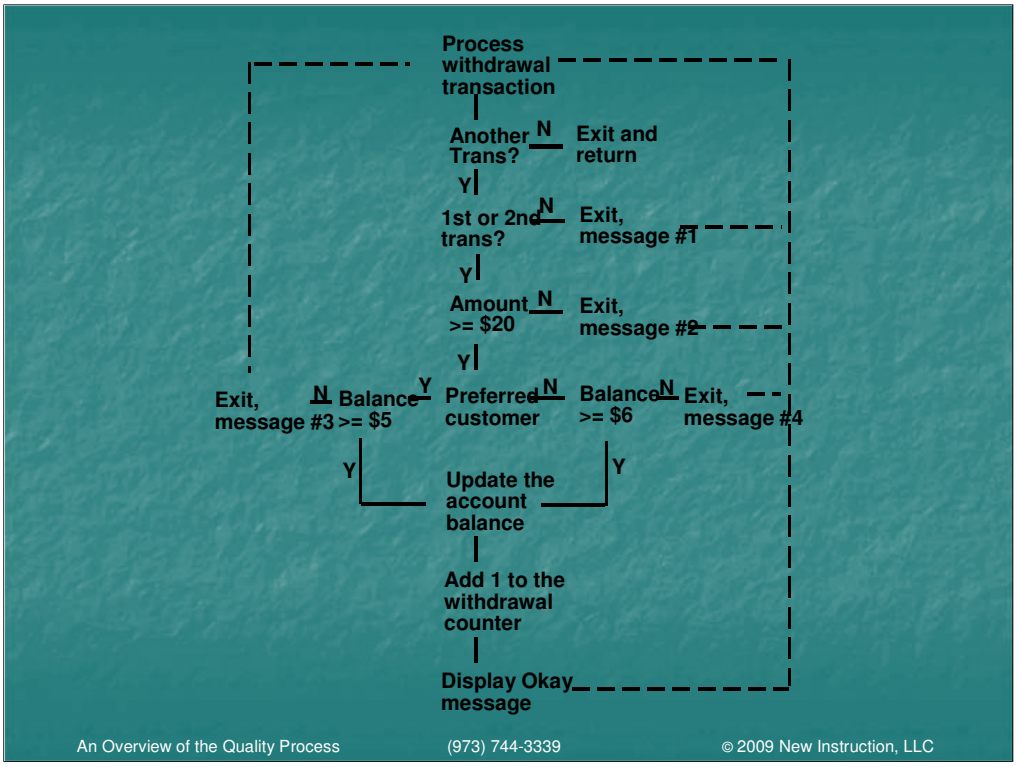
The following specification identifies account balance requirements for making an ATM withdrawal. The account balance must satisfy the following requirements before an ATM withdrawal will be approved:

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.

An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC


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
N
Balance >= $6
N
Exit, message #4
Y
Update the account balance
Add 1 to the withdrawal counter
Display Okay message
Balance >= $5
Y
N
Exit, message #3
Y
Update the account balance
Add 1 to the withdrawal counter
Display Okay message
An Overview of the Quality Process
(973) 744-3339
© 2009 New Instruction, LLC


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
Exit, message #3
N
Balance >= $5
Y
Preferred customer
N
Balance >= $6
N
Exit, message #4
Y
Y
Update the account balance
Add 1 to the withdrawal counter
Display Okay message
An Overview of the Quality Process
(973) 744-3339
© 2009 New Instruction, LLC


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
N
Balance >= $6
N
Exit, message #4
Y
Balance >= $5
N
Exit, message #3
Y
Update the account balance
Add 1 to the withdrawal counter
Display Okay message
An Overview of the Quality Process
(973) 744-3339
© 2009 New Instruction, LLC

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?

An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC

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.

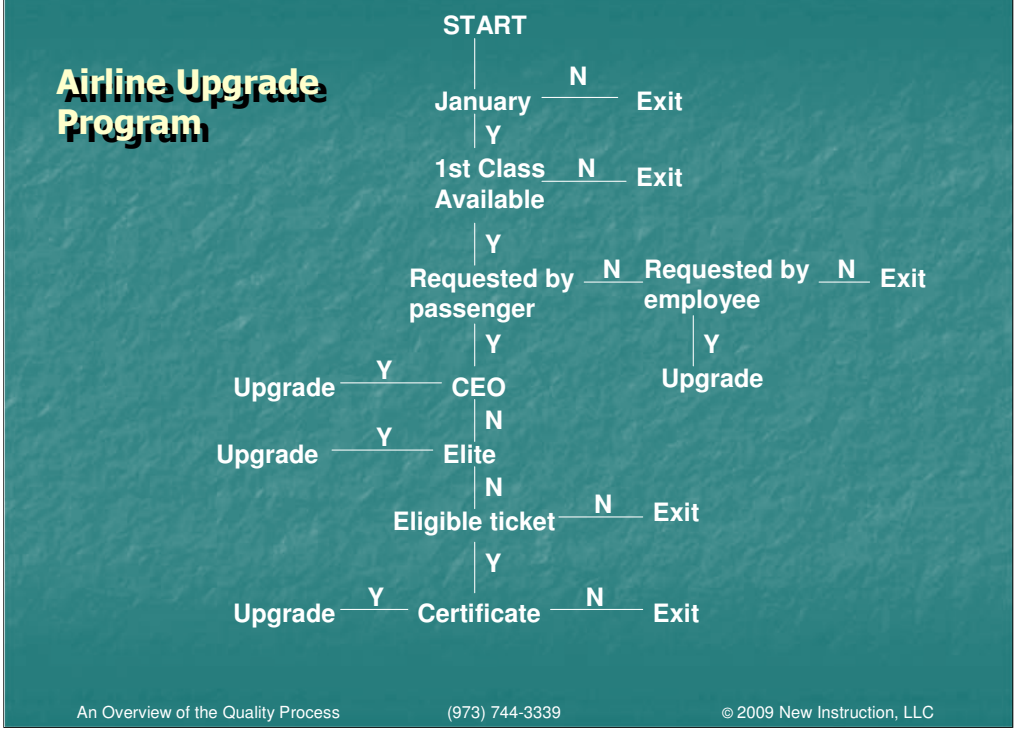
An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC

Airline Upgrade Program

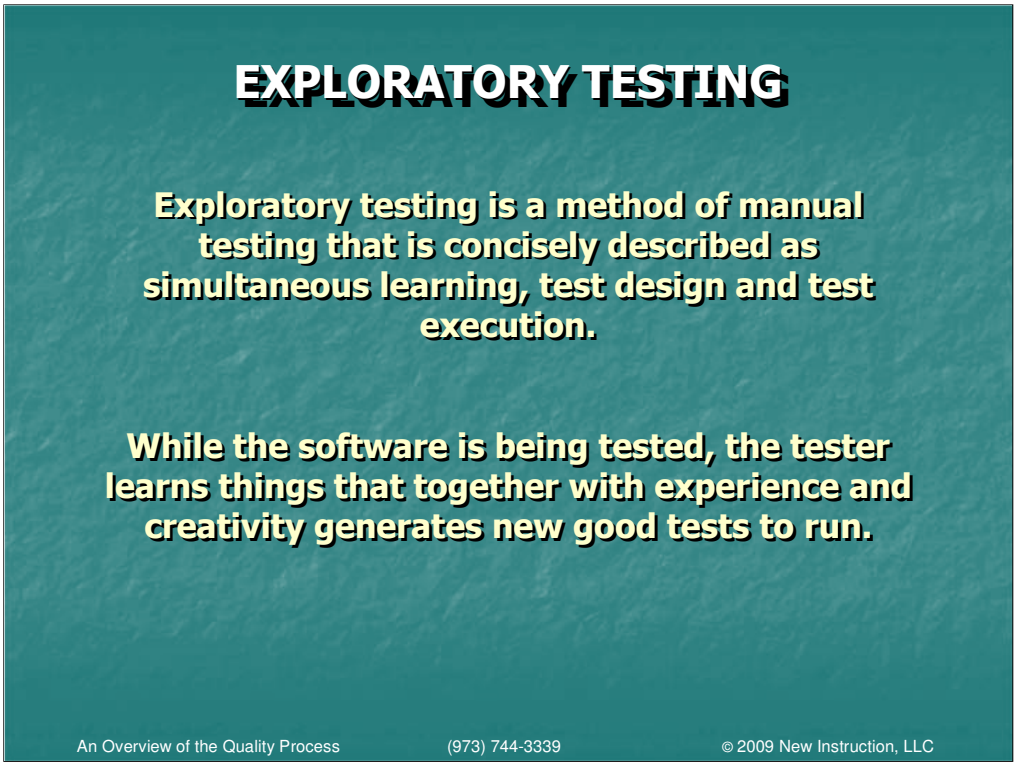
	1	2	3	4	5	6	7	8	9
Decisions:									
January	N	Y	Y	Y	Y	Y	Y	Y	Y
1st Class available	--	N	Y	Y	Y	Y	Y	Y	Y
Requested by passenger	--	--	Y	N	N	Y	Y	Y	Y
Requested by employee	--	--	--	Y	N	--	--	--	--
CEO	--	--	Y	--	--	N	N	N	N
Elite status	--	--	--	--	--	Y	N	N	N
Eligible ticket	--	--	--	--	--	--	N	Y	Y
Certificate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

A Decision table illustrates decisions to be made and actions that will results-within the grid, test cases required are identified. The table can summarize the testing that needs to be done – identifies test scripts, not necessarily the number of test cases required to test a given set of conditions.

Airline Upgrade Program

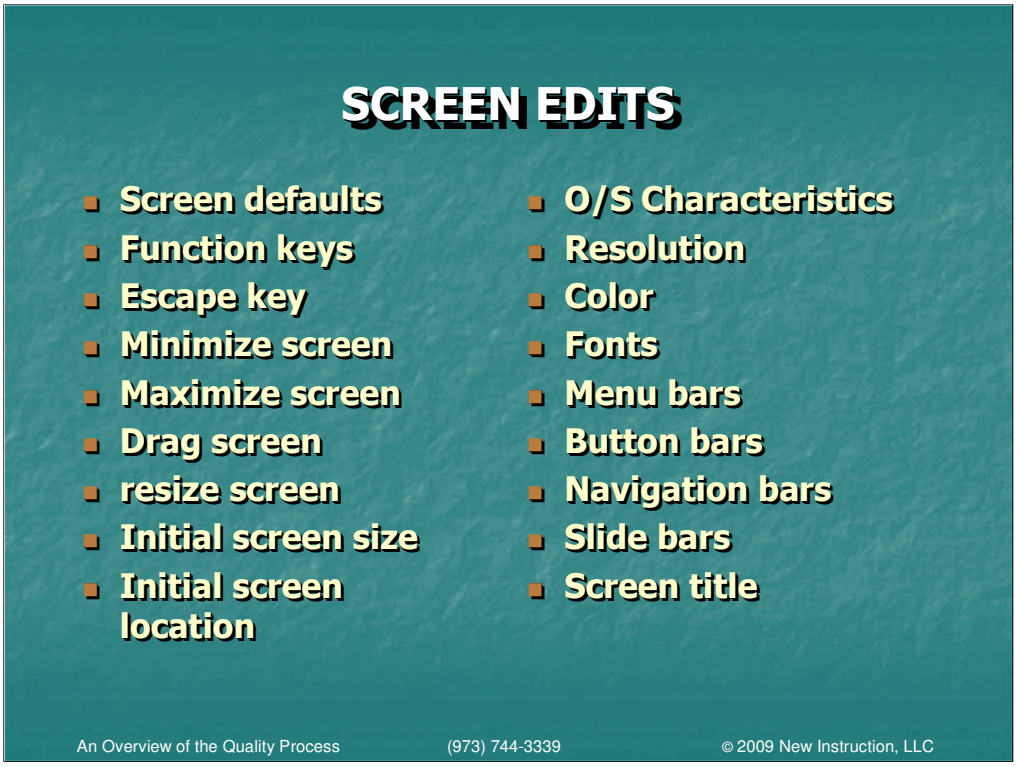


An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC



Starts with basic testing and moves to unscripted testing based on tester knowledge. (I wonder what happens if I . . . . .)

The key to this testing is that it won't do everything, but it allows greater flexibility in our testing. The difficulty with this type of testing is to be able to define exactly what happened to generate the error so the error can be recreated. First phase – not documented, not scripted. Second phase – should be documented and scripted to become part of the test suite.



CHECKLISTS – Like a grocery list, reminding us of what needs to be tested, types of testing that need to be done. Slides include examples of checklists for testing screen edits, button edits, character entry.

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

An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC

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

An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC

Section 3

Test Planning

An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC

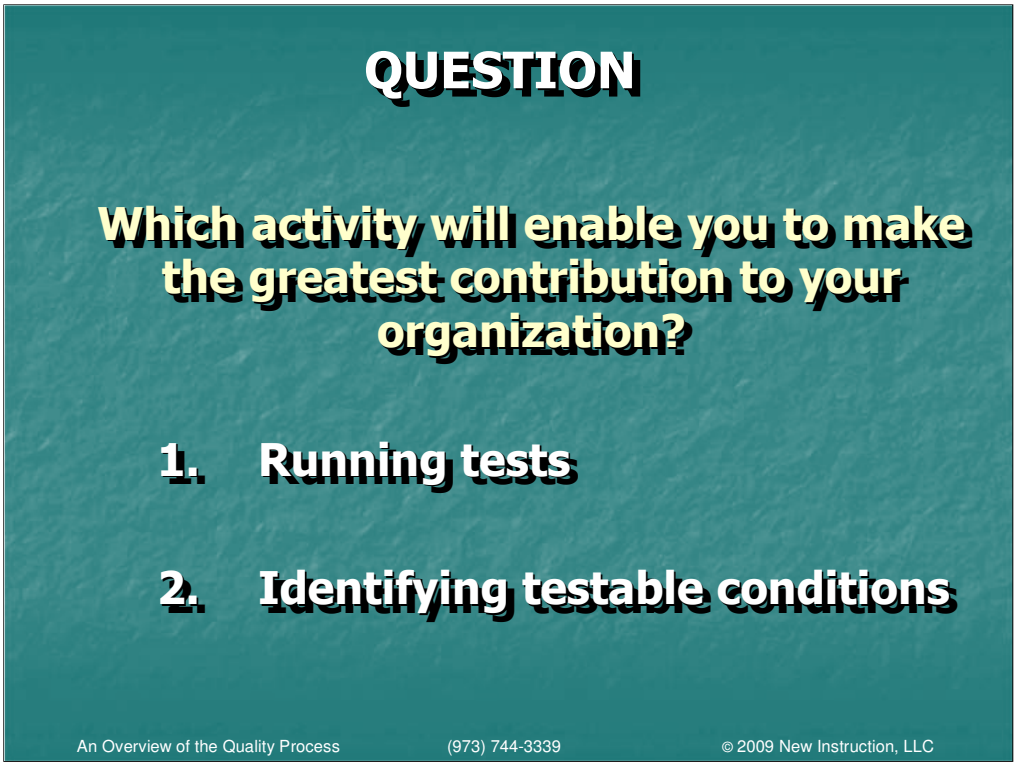
SETTING TEST OBJECTIVES

How will you know when to stop testing?

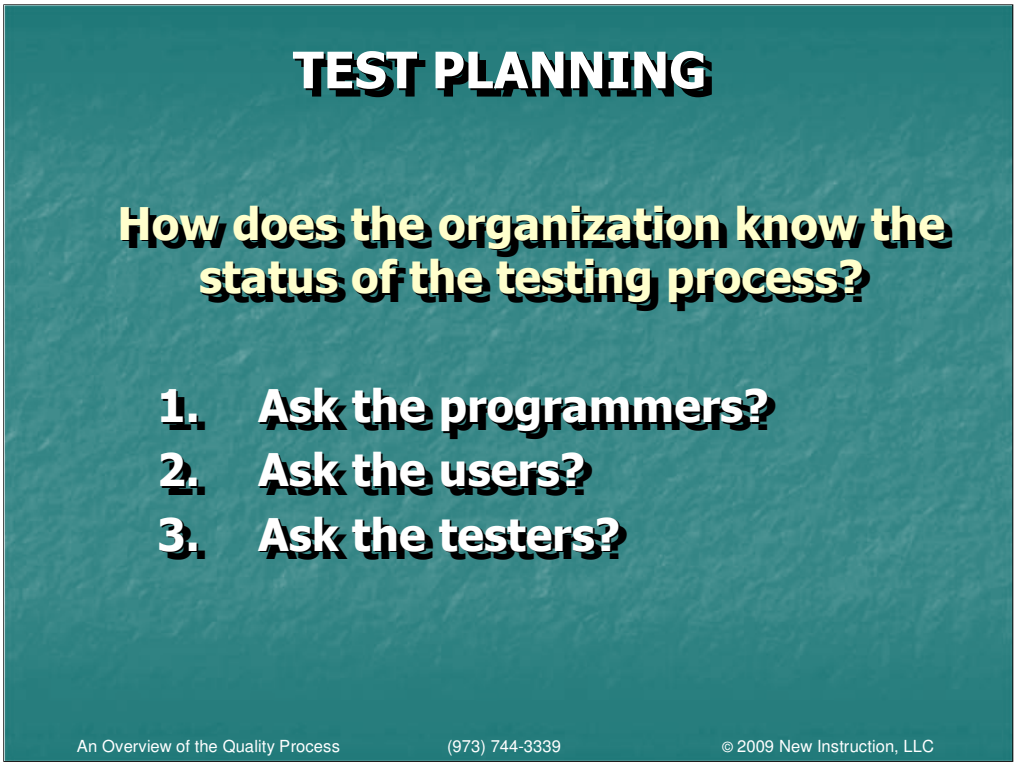
1. **Programmers say so.**

2. **Time is up.**

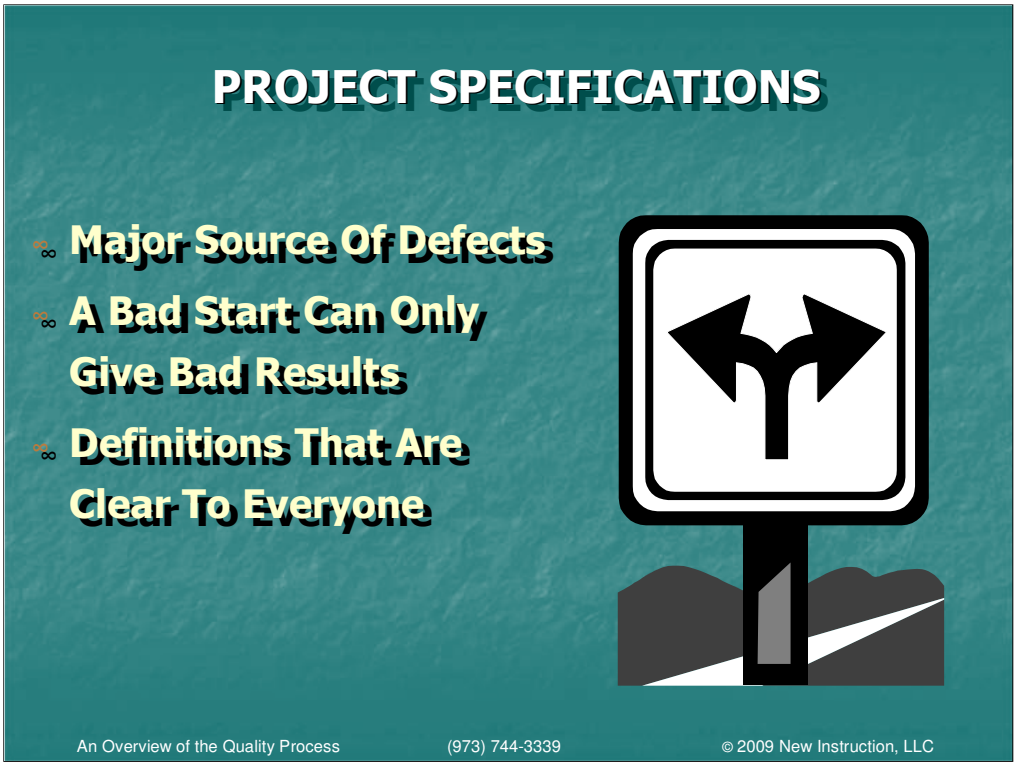
An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC



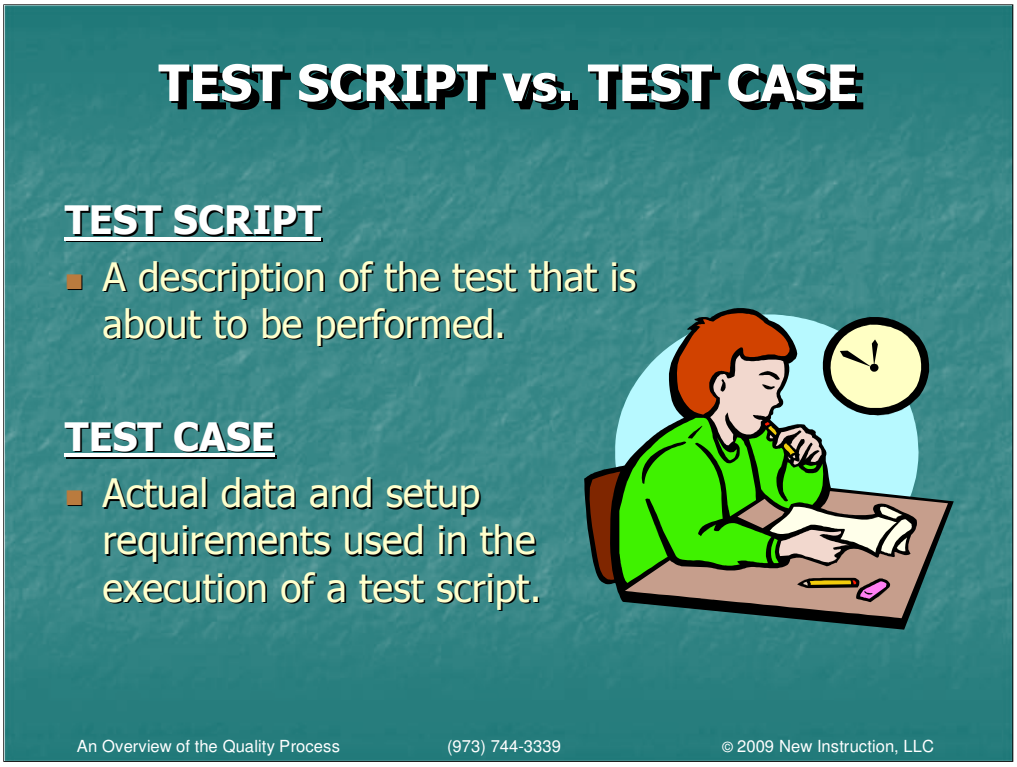
Identifying the RIGHT tests to run is the most important aspect of test planning.



Status of testing should be collected from programmers, users, and testers – they have different perspectives on this topic.



Bad specifications can only give a BAD result – specs must include definitions that are clear for all stakeholders.



Test Script vs. Test Case:

Script is higher level, a description of the test about to be performed. There is a one to many relationship between test scripts and test cases. However, if there are more than 20 – 25 test cases in one test script, probably should break into multiple scripts.

Test case includes the actual data to be tested.

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

An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC

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.

An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC

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.

An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC

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.

An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC

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

An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC

SYSTEM / ACCEPTANCE TEST CASES

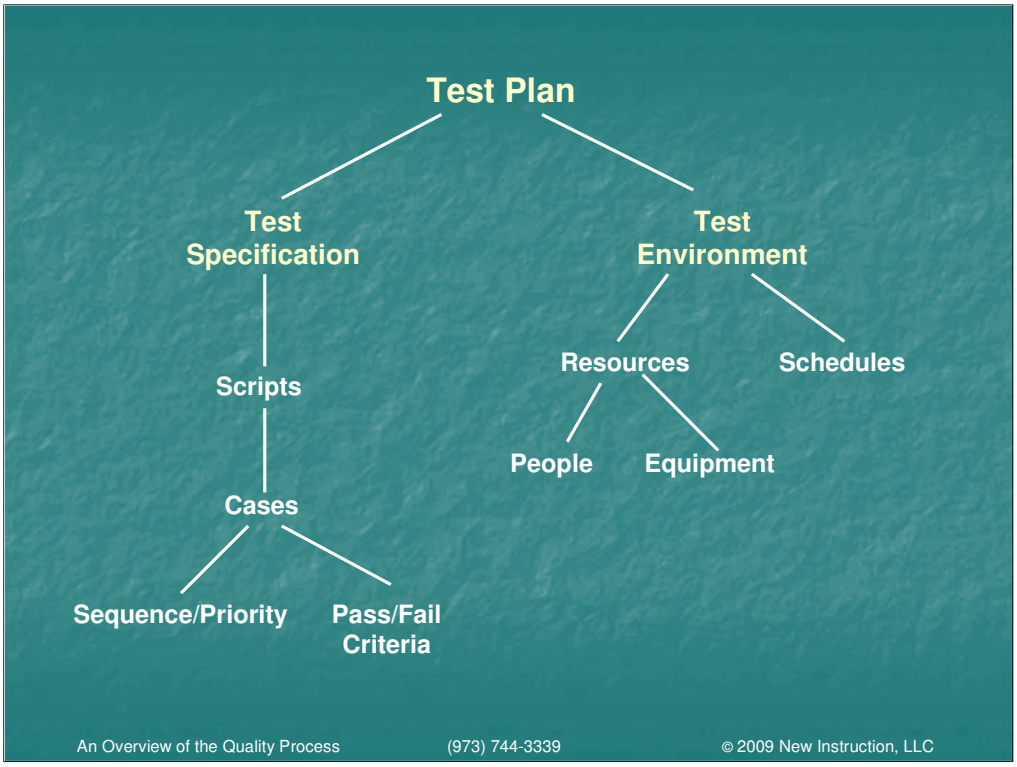
Script:

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

An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC


Test Plan
Test Specification
Test Environment
Scripts
Resources
Schedules
Cases
People
Equipment
Sequence/Priority
Pass/Fail Criteria
An Overview of the Quality Process
(973) 744-3339
© 2009 New Instruction, LLC

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

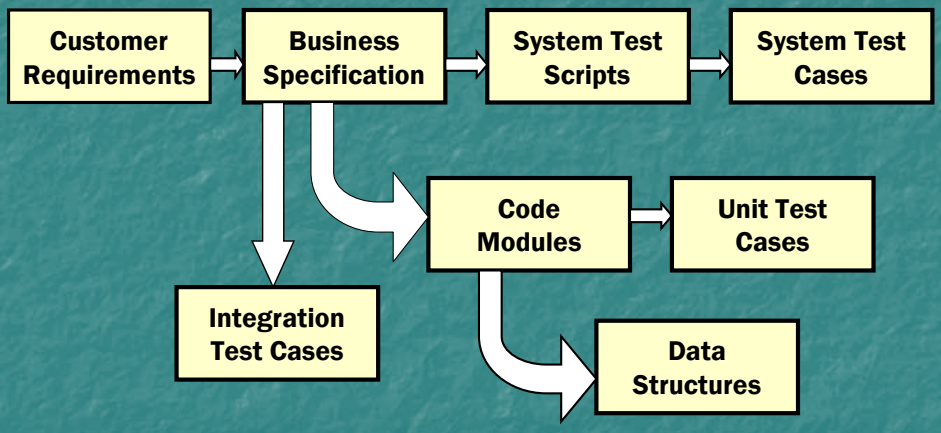
An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC

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?

An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC

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.

An Overview of the Quality Process (973) 744-3339 © 2009 New Instruction, LLC