

# CSTE CBOK, redefined

Certified Software Test Engineer

Examination Study Guide

&

Common Body of Knowledge (CBOK)

Copyright © Quality Assurance Institute® (QAI)

Compiled by Four Quality Assurance Analysts at HP/Compaq in Houston

Author's Note:

The original CSTE CBOK, 1999 edition, published by QAI is redefined and recompiled with accurate and concise information based on the skills and experience of four QA analysts working at HP / Compaq in Houston, Texas. Data is also accumulated from distinguished web sites for the purpose of attempting the CSTE test in June 2004.

This material is by no means adequate for CSTE preparation. Rather it is designed to supplement (a) disciplined knowledge of the software development lifecycle (b) comprehensive authority on the latest industry standards on software testing and (c) several years of working experience that are pre-requisites for CSTE certification.

We have added various acronyms in an attempt to easily remember the long list of components that each chapter and topic headers branch out to. References to the page numbers are based on the 1999 edition only and are invalid for other publications.

While all attempts have been made to ensure accuracy and completeness, there is always room for improvement but we believe we have made a significant contribution in a very limited time, in spite of our professional commitments.

All recognition is credited to QAI and CBOK. This recompiled version is distributed under no obligation and for no monetary value. We hope this will help other testers score better grades and eventually add value to their testing efforts.

We wish you the best of luck on your CSTE.

D, E, N and R

Testing Myths:

Page 4-5 & 4-12

Testing identifies errors, and therefore appears to place the blame of those errors on the development team. This approach can lead to "us versus them" attitude among the entire team.

- Anyone can test and no particular skill set or training is required.
- Testers can test for quality at the end of the development cycle.
- Testers hold up implementation.
- Giving testers less time to test will reduce the chance of them finding defects.
- Letting the testers find the bugs is the appropriate way to debug the system.
- Defects found in the production environment are the testers fault.
- Testers do not need any training, only programmers do.
- Testing is easy
- Anyone can perform testing
- No training or prior experience required

Testers need to:

- Thoroughly understand the system
- Thoroughly understand the technology the system is being developed on
- Possess creativity
- Understand development methodology and resulting artifacts

Important:

William Perry and Randall Rice authors of the book, "*Surviving the top ten challenges of software testing. A people oriented approach*", identified ten people challenges

The continuous improvement cycle:

Page 4-6

Plan: Device a Plan, define your objective and identify a strategy with supporting methods.

Do: Execute the plan, create the conditions and perform necessary training.

Check: Check the results, performance and changes in conditions & abnormalities.

Act: If the work is progressing according to the plan, take corrective actions to evade deviations from plan.

Remember: PDCA

System Proposal

Page 5-6

The purpose of the proposal from the producer's point of view is to get agreement on more work. Three steps; Preparing, Presenting and Closing

## Effective Criticism

Page 5-4 & 5-5

1. Do it privately.
2. Have the facts
3. Be prepared to help
4. Be specific on expectations
5. Follow a specific process
  - State the positive first
  - Criticize the work, not the individual
  - Get an agreement that there is a problem
  - Seek advice on how he thinks he can improve his performance
  - If individual is unable, suggest a course of action
  - Make a specific contract and define expectations

## Effective Listening

Page 5-10 thru 5-13

- Hear the speaker
- Attend to the speaker
- Understand the speaker

*Remember: U-HAUL, You Hear, Attend and Understand. They are all parts of Listening*

## Hearing the speaker

- Information Channel (the subject or content)
- Verbal Channel (the selection of words)
- Vocal Channel (the tone of voice used)
- Body Channel (body language and gestures)
- Graphics Channel (pictures and presentations)

*Remember: IVVVG*

## Attending to the speaker

- Maintain eye contact
- Provide continuous feedback
- Periodically restate what you heard the speaker say
- Free your mind of all other thoughts
- Periodically ensure that the information has been adequately understood

*Remember: MPPFP*

## Understanding the speaker

- Discriminative Listening
- Comprehensive Listening
- Critical Listening
- Appreciative Listening
- Therapeutic Listening

*Remember: DCCAT*

## Reasons people don't listen

Page 5-10

1. They are impatient

2. Too busy rehearsing what to say next
3. Self conscious
4. External stimuli
5. Lack of motivation
6. Lack of interest

Classification as per Chapter 2 of CBOK

Page 2-2

- A. Communications
  - Giving Information
  - Audience Evaluation
  - Effective Presentation
  - Written Correspondence
  - Oral Delivery
- B. Receiving Information
  - Effective Listening
  - Interviewing
  - Analyzing
- C. Personal Effectiveness
  - Negotiation
  - Conflict Resolution
  - Influence and Motivation
  - Judgment
  - Facilitation
- D. Continuing Professional Education
  - Identification of Training Needs
  - Behavior Change Techniques
- E. Leadership
  - Meeting Chairing
  - Facilitation
  - Team Building
  - Process Definition
- F. Recognition
- G. Networking
- H. Code of Conduct

Conflict Resolution

Page 5-14 & 5-15

Dr. Leonard Zunin, a human relations consultant, in his book called *Contact: the First Four Minutes* states that unless complaints are solved within the first four minutes, they will:

- a) Give up on you,
- b) Sense that you have not accepted the urgency,

- c) Not accepted their problem as your problem and
- d) Feel that you are not the right person to address or resolve their problem.

Recommended actions:

- a) Get on your customer's wavelength
- b) Get the facts
- c) Establish and initiate an action program
- d) Follow up with the customer.

Image and Influence

Page 5-16

Whether you are promoted or your recommendations are accepted is partially dependent on the image you project.

Image = Visual intelligence about you.

According to [James R. Baehler](#), a management consultant and author of the book, *The New Manager's Guide to Success*; the six basic attributes of executive image are;

1. Purposeful
2. Competent
3. Analytical
4. Decisive
5. Confident
6. Appearance

Task Force

Page 5-17 & 5-18

Task forces are suited for decision making and not problem solving.

*Management principles of task force*

1. The leader should be an expert
2. Include individuals from all areas
3. Select the best people available
4. Should be organized for a single purpose
5. Should have a clear description of the problem
6. The output is a recommended course of action

*Organizational principles of task force*

1. Should be between 3 to 8 members
2. Should begin when the need is recognized
3. Should finish the business quickly
4. Should meet on neutral ground
5. Appoint someone to record all significant information
6. Proceedings should not be discussed publicly

*Remember: Associate the 12 rules to the 12 member jury. Most functions are similar to the responsibilities of the jury.*

## Behavior change techniques

Page 5-19 thru 5-21

Awareness involves two steps of five tasks each.

## Prepare for awareness training

1. Select awareness topic
2. Identify topic customers
3. Define awareness training objectives
4. Define customer benefits
5. Develop administrative training plan

## Conduct awareness training

1. Attendee's needs
2. Awareness topic / product
3. Identify objections to product/problem
4. Overcome objections
5. Recommend course of action

## Group Think

Page 5-22

According to [Irving Janis](#), who conducted a study on policy making groups, *Group Think* is the production of faulty decisions.

## Problems of Group Think

- The illusion that the group is invulnerable
- They rationalize and or discount negative information
- Ignore ethical or moral consequences
- Stereotyped views
- One-track mind

## Prevention techniques

- Assign a person to be the devil's advocate
- Use external expert resources
- Break into sub groups for discussion
- Take extra time
- Prevent members to divulge personal opinions

## Cost of Quality

Page 5-40

Cost of Quality is everything, other than the true cost of building the product. It is the sum of;

- Cost of Prevention
- Cost of Appraisal
- Cost of Failure

*Remember: Pacific Air Force*

## Definition of Quality

Page 5-35

- Quality is defined as meeting the customer’s requirements, the first time and every time.
- Quality is much more than the absence of defects.
- Quality requires controlled and continuous process improvement.
- Quality saves, it does not cost
- Quality is not a problem but the solution to the problem
- Improving quality is the most direct way to increase profits
- Quality is the single most important factor that affecting the organization’s long term performance.

Developers’ view: Meeting customers’ requirements

Customers’ view: Meeting customers’ needs

Page 5-41

Five Quality perspectives

- Development based; conforms to requirements
- Product based; possesses the desired features
- User based; fit for use
- Transcendent; I know it when I see it
- Value based; developed at an acceptable cost

Remember: Development PUT Value

Five critical factors for success

1. Portability
2. Ease of use
3. Accuracy
4. Service Level
5. File Integrity

Remember: Filthy PEAS

QA vs. QC

Page 5-41 & 5-42

Quality Assurance	Quality Control
A planned and systematic set of activities necessary to provide adequate confidence that requirements are properly established and products or services conform to specified requirements.	The process by which product quality is compared with applicable standards; and the action taken when nonconformance is detected.
An activity that establishes and evaluates the processes to produce the products.	An activity which verifies if the product meets pre-defined standards.
Helps establish processes.	Implements the process.
Sets up measurements programs to evaluate processes.	Verifies if specific attribute(s) are in a specific product or service
Identifies weaknesses in processes and improves them.	Identifies defects for the primary purpose of correcting defects.



QA is the responsibility of the entire team.	QC is the responsibility of the tester.
Prevents the introduction of issues or defects	Detects, reports and corrects defects
QA evaluates whether or not quality control is working for the primary purpose of determining whether or not there is a weakness in the process.	QC evaluates if the application is working for the primary purpose of determining if there is a flaw / defect in the functionalities.
QA improves the process that is applied to multiple products that will ever be produced by a process.	QC improves the development of a specific product or service.
QA personnel should not perform quality control unless doing it to validate quality control is working.	QC personnel may perform quality assurance tasks if and when required.

Examples:

#### Quality Assurance

- A task force selects and installs a system development methodology.
- On a random basis application systems are audited after implementation to determine whether or not the system met standards.
- A measurement system is established to measure the effectiveness of system and unit testing.
- QA is sometimes called quality control over quality control, because it evaluates whether quality control is working

#### Quality Control

- The conduct of an inspection of source code is a Quality control activity
- A computer operator verifies that the jobs to be run that day have been run.
- A computer programmer conducts unit tests to validate that the program works.

#### Quality Improvement

Page 5-38

[Dr. Joseph Juran](#) (*Pareto Principle*) identified ten steps to Quality Improvement.

1. Build awareness
2. Set goals
3. Organize
4. Train
5. Report progress
6. Recognize
7. Communicate
8. Keep score
9. Strive for break through
10. Institutionalize

#### Pareto Principle

Page 5-43

[Vilfredo Pareto](#), an Italian economist, identified that a large percentage of wealth was concentrated in a small portion of the population.

[Dr. Joseph Juran](#) used the *Pareto Principle* to describe any bad distribution as found in quality. Dr. Juran refers to this principle as the separation of the *vital few* from the *trivial few*” also know as the 80/20 rule.

Interpretation of 80-20 rule /Pareto analysis

20% of organization’s Customers account for 80% of its Revenue  
 20% of organization’s Products generate 80% of the Profits  
 80% of the Complaints are due to 20% of Problems

Remember: CPC RPP

Steps involved in using the Pareto Chart

- Name the events that will be analyzed
- Count (Quantify) the named incidences
- Rank the counts of frequency by a bar chart
- Validate reasonableness of the Pareto analysis

Remember: Captain Pareto would make you walk the plank, so it's all about RANK.

Remember: Captain Pareto sailed in a ship, Never drove a Honda CRV

Pareto

- Identify the events
- Quantify the events
- Rank them in order, as in a *Histogram*
- Validate reasonableness of analysis by checking if 20% of the defects account for 80% of the total frequency

Checklist

Page 5-45

One of the *most powerful quality control tools* is a checklist.

Two types of checklists: Factual and Investigative.

Testing Techniques

Page 5-50

1) White Box Techniques:

- Statement coverage; executes all statements at least once
- Decision coverage; execute each decision direction at least once
- Condition coverage; execute each decision with all possible outcome at least once
- Decision Condition coverage; execute each decision with all possible outcome in each decision
- Multiple Condition coverage; invoke each point of entry at least once.

2) Black Box Techniques:

- Equivalence partitioning
- Boundary value analysis
- Error guessing

3) Incremental Testing Techniques:

- Top down
- Bottom up

4) Thread Testing

Reviews

Page 5-57

Three types of reviews

- Informal Review or Peer checklist
- Semi Formal Review or Walk-through
- Formal Review or Inspection

*Remember: Real Cold Water Intake for Reviews: Checklist, Walk through, Inspections.*

Phase-end reviews

- Requirement Review
- Design Review
- Test Readiness Reviews

Review Rules

- The product is reviewed, not the producer.
- Defects and issues are identified, not corrected.
- All members of the reviewing team are responsible for the results of the review.

Advantages

- Reviews are an efficient method of educating a large number of people on a specific product in a relatively short time.
- Team members are exposed to a variety of approaches to technical issues.
- Reviews provide training and enforce standards.

Validation and Verification process

Page 5-58

Validation	Verification
The process of determining that the system does the right things	The process of determining that the system does things right
Determining if the system complies with the requirements and performs functions for which it is intended and meets the organization's goals and user needs. It is traditional and is performed at the end of the project.	The review of interim work steps and interim deliverables during a project to ensure they are acceptable. To determine if the system is consistent, adheres to standards, uses reliable techniques and prudent practices, and performs the selected functions in the correct manner.
Determines if the system accesses the	Determines if the system accesses the correct

data	data
High level activity	Low level activity
Performed during development on key artifacts, like walkthroughs, reviews and inspections, mentor feedback, training, checklists and standards	Performed after a work product is produced against established criteria ensuring that the product integrates correctly into the environment
Determination of correctness of the final software product by a development project with respect to the user needs and requirements	Demonstration of consistency, completeness, and correctness of the software at each stage and between each stage of the development life cycle.

Test Objectives

Page 5-63

- Defining the test objectives is the first step in the test planning process.
- It is the testing goal that the testers are expected to accomplish.
- It enables the manager to gauge the testing progress and success.

Tool acquisition

Critical path method; doesn't allow the purchase of tools through the side door.

Tool Development and / or Acquisition

Page 5-64

Formal Approach	Informal Approach
Technical requirements	Selection Criteria
User Review	Identify candidate
Generate request for purchase	User review
Issue request for purchase	Score candidate
Proposal evaluation	Select tool
Select source	

Entrance and Exit Criteria

Page 5-73

Entrance Criteria: defines the required conditions and standards for product quality that must be present or met for entry into the next stage of the development stage.

Exit Criteria: defines standards for product quality, which block the promotion of defects to subsequent stages of the development process.

Examples:

System test entrance criteria:

- Successful execution of the integration test plan
- No open severity 1 or severity 2 defects
- 75-80% of total system functionality and 90% of major functionality delivered
- System stability for 48-72 hours prior to start of test

System test exit criteria:

- Successful execution of the system test plan
- System meets pre-defined quality goals
- 100% of total system functionality delivered

Risks:

Page 4-9

We cannot eliminate risks but can reduce their occurrences and or impact or loss.  
Too little testing is a crime; too much testing is a sin.

Risk Identification

Page 5-80

Risk is identified as the product of the probability of the negative event occurring and the impact or potential loss associated with the negative event.

Testing is driven by the impact of the risk. Lame projects with minimum functionalities require the least amount of testing while high risk projects like Airport air traffic control, nuclear plants and defense systems require rigorous and thorough testing.

*Remember: Magnum PI was in a Risky business. PI = Probability and Impact*

FIVE dimensions of risk

1. Technology integration
2. Size and complexity
3. System environment and stability
4. Criticality / mission impact
5. Reliability and integrity

William Perry proposed FOUR methods for conducting Risk Analysis:

1. Judgment and Instinct
2. Estimate on the cost of failure
3. Identification and weighing the risk attribute
4. Software risk assessment package

Special Types of Testing

Page 5-82

Usability Testing determines how well the user will be able to understand and interact with the system

Vendor Validation Testing verifies if the functionalities of the third party software meet the organization's requirements

Conversion Testing is specifically designed to validate the effectiveness of the conversion process

Stress testing is one which deliberately stresses a system by pushing it to and beyond its specified limits. It is testing the system at and beyond its maximum planned output and determining the point where the system breaks. The intent is not to 'obtain a free ride' but to determine if the system can recover as specified.

Load Testing is where additional load is simulated either manually or using tools to determine how the system will performance in real life situations.

Performance Testing is comparing and measuring the performance of the application against a pre-determined or expected level that is pre-defined by the business users.

Recovery Testing evaluates the contingency features built into the application. Tests the backup, restoration and restart capabilities

Configuration Testing validates if the application can be configured to run in different operating systems and or different versions of internet browsers including Netscape, and modem speeds etc.

Benefit Realization Testing is conducted after the application is moved to production to determine if it delivers the original projected benefits. The analysis is usually conducted by the business users and results are delivered to executive management.

## Test Planning

### Page 5-84

According to [Glenford J. Myers](#), (author of *The Art of Software Testing*), Testing is the process of executing a program with the intent of finding errors.

## Developing the test plan

The purpose of the test plan is to communicate the intent of the testing activities. It is critical that this document be created as early as possible. It may be desirable to develop the test plan iteratively, adding sections, as the information is available with constant updates as changes occur. Test plan is a document that describes the overall approach to testing and its objectives and contains the following headers:

- Test Scope
- Test Objectives
- Assumptions
- Risk Analysis
- Test Design
- Roles and Responsibilities
- Schedules and Resources
- Test Data Management
- Test Environment
- Communication Approach
- Test Tools

*Remember: Vultures SOAR over Dirty Rail Roads, in Soaking Rain for Dead MEAT.*

## Test Coverage

### Page 5-91

The Test Plan and Strategy utilizes a balance of testing techniques to cover a representative sample of the system. The test planning process is a critical step in testing process.

Since exhaustive testing is impractical and testing each and every scenario with each variable is time consuming and expensive, the project team establishes a test coverage goal, based on the risks and criticality associated with the application under test.

Projects where the application supports critical functions like air traffic control or the military defense systems, or the nuclear power plants or space technology etc., the coverage goal may be 100% at all stages of testing.

One way to leverage a dynamic analyzer during system test is to begin by generating test cases based on functional or black box test techniques. When functional testing provides a diminishing rate of additional coverage for the effort expended, use coverage results to conduct additional white box or structural testing on the remaining parts of the application until the coverage goal is achieved.

Tools for tests coverage analysis: *McCabe* and *Battlemap*.

#### Requirement Attributes

- Specific
- Measurable
- Attainable
- Realizable
- Traceable

Remember: SMART

#### Requirement Tracing

##### Page 5-88

A requirement is defined as the description of a condition or capability of a system.

Each requirement must be logical and testable. To ensure that all requirements have been implemented and tested, they must be traceable. Each requirement must be mapped to a test cases and test steps and defects.

*Mercury Interactive's* Test Director does a good job of mapping the entire history and then updating the status of the requirement as a defect is detected, re-tested or corrected. Other tools like DOORS, or Caliber and *IBM's* Rational Requisite Pro can also be used to log, track and map requirements.

#### Post Planning Activities

##### Page 5-89

- Versioning
- Change Control
- Change Management
- Configuration Management

Large projects have a full time configuration manager who is required to keep the environments configured appropriately. He ensures that the test environment must mirror the production environment. He is also responsible for managing, documenting and tracking the following:

- Source Code
- Requirements
- Analysis Model
- Design Model
- Test Cases and Procedures

- Automated Test Scripts
- User Documentation / Manuals and Help Files

#### Examples of CM Tools

ChangeMan, Production Version Control System PVCS, Clear Case, Visual Source Safe VSS

#### Test Execution:

##### Page 5-92

The test plan should be updated throughout the project to ensure that the true expected results have been documented for each planned test. The test plan should contain the procedure, environment and tools necessary to implement an orderly controlled process for test execution, defect tracking, coordination or rework and configuration and change control.

Executing the integration test plan should begin once unit testing for the components to be integrated is completed. The objective is to validate the application design, and process that the application components can successfully be integrated to perform one or more application functions.

Depending on the sequence and design of the integration test build, the application may be ready for System Test once the pre-defined Exit Criteria has been met.

Executing the system test should begin as soon as a minimal set of components have been integrated and successfully passed integration testing. Testing for standard backup and recovery operations are conducted and the application is subjected to performance load and stress tests for measuring the response time under peak load with multiple users applying multiple functionalities at the same time.

System test ends when the test team measures the application capabilities and each reported defect is thoroughly regression tested to generate enough confidence that the application will operate successfully in the production environment.

#### How do you know when testing is complete?

##### Page 5-94

When the test manager is confident that the application will perform as expected in the production environment. This confidence is derived by analysis conducted on:

- Meantime between failures
- Percentage of coverage achieved
- Number of open defects and their severity
- Risk associated with move to production
- Risks associated with continuing to test.

*Remember: Mummy is Polite and Nanny is Really Rude*

#### Defects:



Page 4-7

Two types of Defects:

Variance from product specifications (Developer's point of view):

The product varies from the specifications. Example; when a added to b is to produce c and the product is producing d.

Variance from customer expectations (Customer's Point of view):

Something that the user wanted but was (a) not specified in the requirements docs, (b) the requirement was incorrectly interpreted, (c) requirement was omitted or (d) the manner or method by which the requirement was implemented was unsatisfactory.

Three Attributes of defects

Page 5-103

Defect Naming:	Requirement defect Design defect or Documentation defect
Severity of Defect:	Critical; Show stoppers Major or Incorrect output is delivered Minor Spelling mistakes, etc.
Type of Defect:	Wrong; specs were implemented incorrectly Missing; specs were not implemented at all Extra; specs implemented correctly but were not requested

Important:

Dr. W. Edwards Deming, author of the book; *Out of Crisis* said "90% of all defects are caused by process problems".

Defect Tracking

Page 5-95

It is critical that defects identified at each stage are tracked to resolution.

Defects are recorded for four major purposes:

1. To correct defects
2. To report the status of the application
3. To gather statistics used to develop defect expectation in future applications
4. To improve the software development process

Attributes of defect

A defect log should include;

- Defect identification #
- Description of the defect
- Detection date
- Detected by
- Environment found
- Test case #
- Pre requisite conditions
- Detailed steps that lead to the defect
- Severity level

- Defect priority
- Assigned to
- Defect status
- Snapshots, additional info

### Severity vs. Priority

#### Page 5-96

Defect severity is directly related to the impact it causes on the application. A severity status of *critical* is assigned to defects that are show stoppers or cause data corruption, system crash or security violations while a severity status of *minor* would be a minor misspelling. This attribute is added to the defect log by the test team to classify and quantify the total defects so an overall status of the application can be determined.

Defect Priority is related to the level of defect-fixed urgency. It is usually more subjective and based upon user input. Defects that cause several other unrelated requirements to malfunction, or when several other functionalities cannot be tested unless a particular defect is fixed, is usually assigned a top priority.

### Regression Testing

#### Page 5-97

Regression testing is conducted during all stages of testing after changes have been made or when defects are fixed to ensure that no new defects are introduced. A full regression test is usually conducted when multiple changes have been made to critical components of the application.

### Metrics

A metric is a mathematical number that shows the relationship between two variables. It is a quantitative measure of the degree to which a system, component or process, possesses a given attribute.

#### Process Metric

A metric used to measure characteristics of the methods, techniques and tools.

#### Product Metric

A metric used to measure characteristics of documentation and code.

#### Software Quality Metric

SQM is a function whose inputs are software data and whose output is a single numerical value that can be interpreted as the degree to which the software possesses a given attribute that affects its quality.

### Test Data

#### Page 5-98

Testers are typically responsible for reporting their test status at regular intervals. The following measurements are usually reported:

- Total number of tests
- Total number of tests executed
- Total number of tests executed successfully

#### Defect Tracking Data

- Total number of defects detected in each activity
- Total number of defects corrected in each activity
- Average duration between defects detected and defect correction
- Average effort to correct a defect
- Total number of known defects that remain uncorrected at delivery

#### Software Performance Data

- Average CPU utilization
- Average memory utilization
- Measured input / output transaction rate

#### Test Reports

##### Page 5-102

Test reports are a key mechanism available to the test manager to communicate the value of testing to the project team and the management.

#### Interim Reporting

##### Page 5-104

- Functional testing status report
- Functions working timeline
- Expected vs. Actual defects timeline
- Defects detected vs. Corrected gap timeline
- Average age uncorrected defect by type
- Defect distribution report
- Relative defect distribution report
- Testing action report
- Final test reports
- Unit test report
- Integration test report
- System test report
- Acceptance test reports

#### Capability Maturity Model

##### Five steps to CMM

1. Optimize
2. Manage
3. Define
4. Repeatable
5. Initial

*Remember: See My MOM DRI clothes. CMM O M D R I*

#### Difference between QA and Internal Audit

Quality Assurance is that activity within information systems charged with implementing the quality policy through developing and improving processes. QAI has recommended that quality assurance be a leadership position, and emphasizes the strong interpersonal activities involved in making improvement occur.

Internal Auditing is an independent appraisal activity within an organization for the review of operations and is a service to management. It is a managerial control which functions by measuring and evaluating the effectiveness of other controls.

QA and IA professions generally have the following five accepted criteria

1. Code of ethics
2. Statement of responsibilities
3. Program of continuing education
4. Common body of knowledge
5. Certification program

Cause Effect Diagram is a problem analysis tool. It's a systematic approach to identify the all the associated causes that contribute to an effect (generally used to analyze the negative effect). Once all the causes are identified, these are categorized in to direct, contributory and root causes.

This technique is used with other brainstorming tools for identifying the cause categories as well as the causes. Pareto analysis, for example is used to prioritize the actions for implementation etc. To better understand the problem, the effect and it's associated causes are put as a diagram and because of it's shape, it's also know as the Fishbone Diagram. Dr. Kaoru Ishikawa, a Japanese quality control statistician invented the fishbone diagram and this is also referred to as the Ishikawa Diagram.

Whatever name is used, remember the value of the fishbone diagram is to assist teams in categorizing the many potential causes of problems or issues in an orderly manner and identifying the root causes.

#### Test Effectiveness vs. Test Efficiency

To ensure Test Effectiveness, make sure that the application is "doing the right thing". This enables a prediction of future test effort and corrective maintenance to be made. Monitoring changes to the testing process will requires metrics about errors found during and after User Acceptance Testing. This can give an on-going measure of test quality.

Test Efficiency: Software Testing "does the thing right".

Testing is always limited in time and budget. Good testing must first of all be effective but must also be as efficient as possible.

There are two measures of efficiency: defect-based and confidence-based.

#### Test Strategy

Test Strategy is the guide or steps to define objective to deliver a quality product. It is very brief consolidated and well defined. Test strategy answers the; What, When, Who, and Where type of questions whereas the test plan is focused around How, i.e. how to perform to meet the guidelines in strategy and it's well defined steps or procedure to perform.

Test Strategy is the subset of test plan. It gives full insight to the counterpart on how testing is to be carried out. Objective is to reduce the risks focus on addressing those risks is not diverted. A strategy for the testing portion of a project describes the general approach and objectives of the test activities. It includes how the stages of testing (unit, integration and system) are to be addressed and which kinds of testing (function, performance, load, stress, etc.) are to be performed.

Test strategy also defines testing techniques and tools to be employed and what test completion and success criteria are to be used. For example, the criteria might allow the software to progress to acceptance testing when 95 percent of the test cases are successfully executed.

#### Three techniques for test bed creation

Resource Constraints - The physical limits of a lab, the ability to segregate a test bed from a production-level cluster and budget limitations affect the decisions on the size, content, and completeness of a test bed.

Culture - A test bed represents a reasoned approach toward managing risk. The deployment of updated applications, new hardware, and patches all entail the risk of disrupting a production-level cluster. Management must support the costs involved in managing the risk.

Support - A test bed, as with any other system of hardware and software, requires support. Individuals must be charged with responsibility for maintaining an up-to-date test bed.

#### Definitions

##### Data Migration

Data migration is defined as the process of translating data from one format to another. This is basically required when new software / DBMS is purchased which is not compatible with the existing system.

One of the many ways to test data Migration is to test the source and the target/destination information i.e.: to validate if the source and target are the same by validating the row count on both of these data sources. Data migration does you no good if the data is not usable after migration.

The source file format can be in .txt, .csv etc. using any of the delimiters like fixed width, | (pipe delimiter), comma separated (,) etc. while the target can be the any database that the new software would be using.

Audience Evaluation: Audience evaluation means ability to evaluate audience needs and develop appropriate presentation material.

Interviewing: Developing and asking questions for the purpose of collecting oral data to be used in an analysis or evaluation is called Interviewing.

Facilitation: The purpose of helping the progress of some event or activity is known as Facilitation.

Team Building: The process of aiding a group to define a common goal and work together towards is known as Team building.

## Bibliography

Testing is the process of executing a program with the intent of finding errors.  
Glenford J. Myers  
*The Art of Software Testing*  
[Page 5-84](#)

A large amount of wealth is concentrated in a small portion of the population.  
Vilfredo Pareto, an Italian economist  
[Page 5-43](#)

Peter R. Scholtes, defines a contrast between *Effectiveness* and *Efficiency* while  
Patrick Townsend, redefined the quality fact and perception.  
[Page 5-35](#)

Dr. W. Edwards Deming developed 14 points management philosophy and identified 7  
deadly management diseases. Michael P. Tveite regrouped and classified them.  
[Page 5-37 & 5-38](#)

Philip Crosby, stress on management commitment as the first step to quality improvement.  
[Page 5-38](#)

“Quality is free.” – Philip Crosby  
[Page 5-40](#)

Group Think is the production of faulty decisions.  
Irving Janis, (conducted a study on policy making groups)  
[Page 5-22](#)

Defines executive image and appearance  
James R. Baehler (a management consultant)  
*The New Manager's Guide to Success*  
[Page 5-16](#)

Complaints must be solved within 4 minutes.  
Dr. Leonard Zunin, (a human relations consultant)  
*Contact: The First Four Minutes*  
[Page 5-14](#)

Developed 14 points management philosophy  
Identified 7 deadly management diseases  
90% of all defects are caused by process problems.  
Dr. W. Edwards Deming  
*Out of Crisis*  
[Page 4-7](#)

Identifies ten people challenges  
William Perry and Randall Rice  
*Surviving the top ten challenges of software testing, a people oriented approach.*  
[Page 4-12](#)