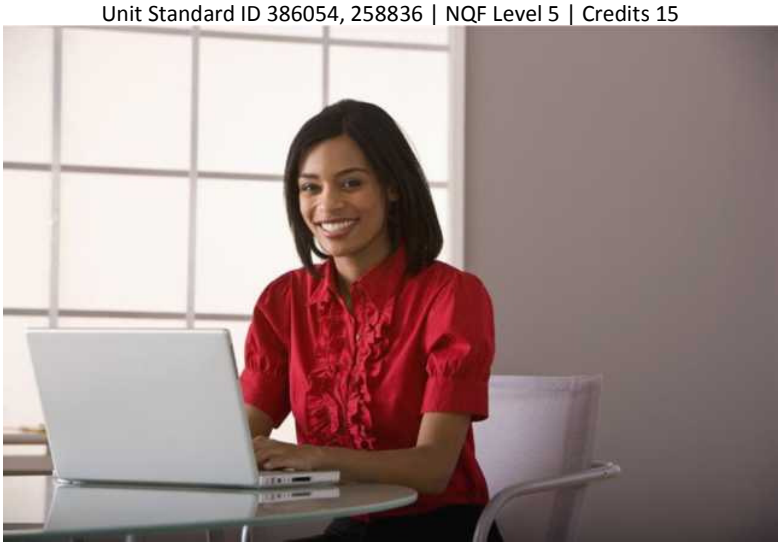
Qualification 83026: National Certificate: Information and Communications Technology (ICT) Software Testing

Test Delivery

Learner Guide

Unit Standard ID 386054, 258836 | NQF Level 5 | Credits 15



Contents

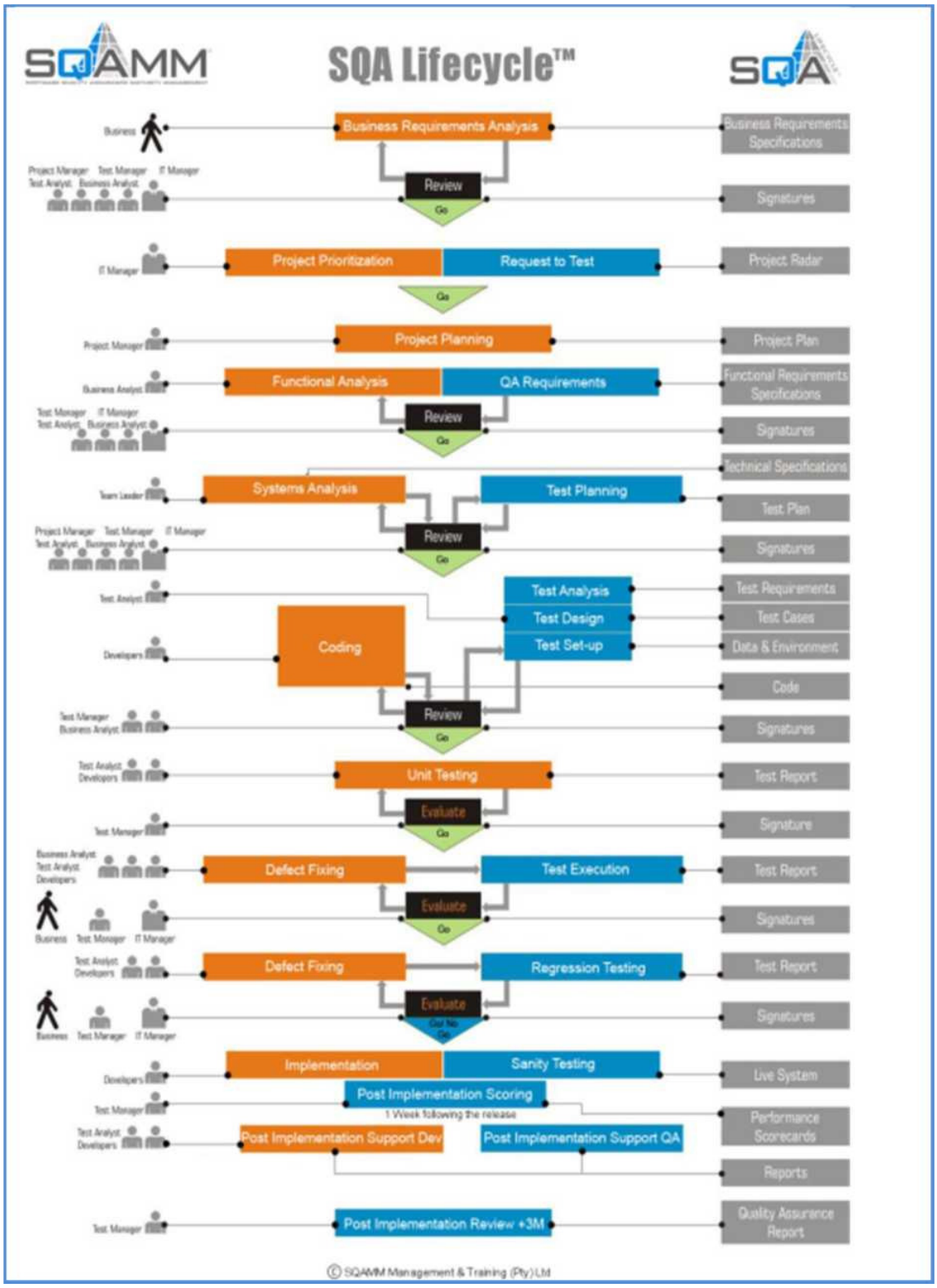
Overview

Glossary and Terms

Term	Definitions
Assessment	The process of collecting evidence of learners' work to measure and make judgements about the achievement or non-achievement of specified National Qualifications Framework (NQF) standards or qualifications.
Assessment Criteria	The assessment criteria for a course are the dimensions with which you will judge how well a learner has achieved the learning goals.
Assessor	A person who is registered by the relevant Education Training Quality Assurance (ETQA) body to measure the achievement of specified National Qualifications Framework standards or qualifications.
Credit	That value assigned by the authority to ten (10) notional hours of learning.
Critical Cross-Field Outcomes (CCFO)	Critical Cross-Field Outcomes refer to those generic outcomes that inform all teaching and learning. For example, CCFOs may include working effectively with others as a member of a team, and/or collecting, analysing, organising and critically evaluating information.
Essential Embedded Knowledge (EEK)	Essential embedded knowledge is explicit and resides within systematic routines. It relates to the relationships between roles, technologies, formal procedures and emergent routines within a complex system.
Formative Assessment	Refers to assessment that takes place during the process of learning and teaching. May also be integrative in nature.
Integrated Assessment	An assessment, which permits the learner to demonstrate applied competence and which uses a range of assessment methods.
Summative Assessment	It is the assessment for making a judgement about achievement. This is carried out when a learner is ready to be assessed at the end of a programme of learning.
Moderation	The process which ensures that assessment of the outcomes described in the NQF standards and qualifications is fair, reliable, and valid.
Outcome	Contextually demonstrated end product of the learning process.

Term	Definitions
Recognition for Prior Learning (RPL)	The comparison of the previous learning and experience of a learner, howsoever obtained, against the learning outcomes required for a specified qualification, and the acceptance for purposes of qualification of that which meets the requirements.
SETA	A body responsible for the organisation of education and training programmes within a specific sector.
Specific Outcome	Knowledge, skills and values (demonstrated in context) which support one or more critical outcomes.
Unit Standard	Registered statement of desired education and training outcomes and its associated assessment criteria together with administrative and other information as specified in the regulations.

SQA Lifecycle



About this Course

SQA Management & Training -

What is SQA?

SQA is an acronym for Software Quality Assurance Maturity Management. It is an independent research and analysis company that specializes in Software Quality Assurance.

The SQA framework is made up of:

- 3 certifications
- 7 products

SQA™ is an "all-in-one" package that delivers end-to-end QA across the development lifecycle ensuring high quality output, reduced costs of rework and an optimized delivery cycle.

In addition, it has tried and tested solutions which enable organisations to validate 3rd party implementations further reducing time to market and costly on-going maintenance.

SQA™ is supported by two key frameworks geared to ensuring that testing services are of high quality and that there is a constant drive to improve testing services and increase SQA maturity levels towards a world class standard. These two quality initiatives are:

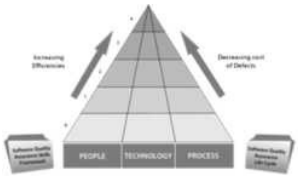
- SQA Health-check™
- SQA Toolkit™

SQA™ and its supporting frameworks, SQA Health-check™ and SQA Toolkit™, are all based on industry best practices that have been extensive research and are an adaptation of the following standards:

- IEEE 829: 1998 Standard for Software Test Documentation
- IEEE 1044-1: 1993 Standard Classification for Software Anomalies
- IEEE 1061 Standard for software quality metrics and methodology

SQA - SQA Maturity Model™

The SQA Maturity Model is the foundation on which SQA Maturity Management is based.



It is a benchmark against which you can measure the efficiencies of your PEOPLE, PROCESS & TECHNOLOGY

Skills development and performance management framework for testing professionals. Process framework for the application of quality assurance and testing to the Product / Systems Development Life Cycle

SQA™ Skills Framework

The SQA skills framework deals with the People side of the SQA Maturity Model™ and provides the delegate with skills required to perform the core competencies of a Professional Tester

Key differentiating factors:

- The SQA Vision is to become the SKILLS PROVIDER OF CHOICE by gaining the reputation within the IT Industry of providing skilled resources that are head and shoulders above everyone else
- Expert Trainers based on a combination of +/- 40 years of testing experience, expert knowledge and subject matter experts.
- All our training is presented by our training partner STS Skills & Training Academy
- Our training material is outcome based, and we encourage all our delegates to become solution driven.
- Our training covers all the skills, on the core competencies of a Professional Tester, that are required for international certification

SQA - Certification

A SQA certification is awarded on the competency of individual to execute quality control within a development lifecycle.

It is divided into three parts:

- SQA Certified Test Analyst
- SQA Certified Testing Professional
- SQA Certified Test Management Professional

The Road to SQA Tester Certification

The SQA™ certification process for the Testing Professional has been divided into 2 levels.

The first level, SQA™ Certified Test Analyst contains 4 core modules, Tester, Requirements Management, Test Analysis and Test Design.

Level 2, SQA™ Certified Testing Professional will require the trainee to complete the remaining 3 modules dealing with Testing in Agile, Functional Automation and Performance Testing.

All trainees are at liberty to apply for a credit on any one of the modules if they feel that they have reached the required competency levels. Apply for a credit will require the trainee to take the course assessment without attending the course.

In keeping with international certification standards, the delegate is expected to attain a minimum mark of 65% on all module assessments. In the event of a delegate failing to reach the required certification mark, delegates will be allowed one "rewrite" of the course assessment at a later date, at no extra cost.
Additional "re-writes" will be available at the ruling price.

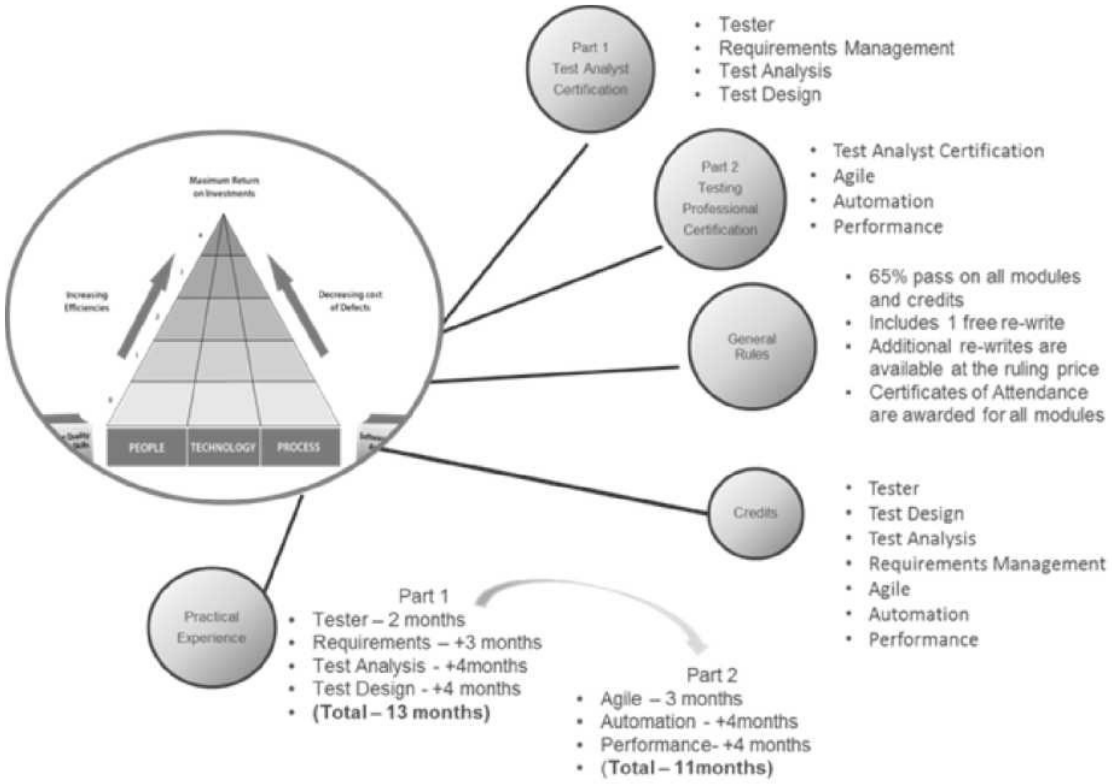
All delegates will be awarded a 'SQA™ Certificate of Attendance' for each course attended irrespective of whether the required mark is reached.

Finally, the delegate will be required to gain the prescribed practical experience in all modules as follows:

Role		Duration
Tester		2 months
Requirements Management	+	3 months
Test Analysis	+	4 months
Test Design	+	4 months
Test Analyst Certification	+/-	13 months
Agile	+	3 months
Automation	+	4 months
Performance	+	4 months
Total		24 months

When the delegate has passed all the required modules, for Levels 1 or 2, they will be required to write a final exam covering all the modules for that level. Once again the pass mark required is 65% to be awarded the SQA™ Certification

SQA™ certification requires a real commitment from both the Trainee and the Organisation. It will also assist with retention of skills.



Skills Programme Framework

Learning Programme Details

The learning material for this programme has been designed and developed according to:

Unit Standard (US) Title	US NO	NQF Level	Credits
Manage the software testing process	386054	5	7
Analyse and apply different Information and Communication Technology (ICT) Systems Development Lifecycle (SDLC) models for a given scenario	258836	5	8

Study Unit 1.1: Understand the Test Environment

US 386054

Manage the software testing process Specific Outcome 1

Demonstrate an understanding of the test environment.

Assessment Criteria

1. An explanation is given of the importance of independent testing.

2. The benefits and drawbacks of independent testing are explained with examples.
3. Team members are identified in terms of skill level.
4. The tasks of a test leader and a tester are listed to differentiate between the roles.

Introduction: The Fundamentals of Software Testing

What is Testing?

Introduction

Contrary to popular belief, testing doesn't consist only of running test. Running test is a very small part of a much larger picture. Test activities exist before and after test execution. Activities such as planning and control, choosing test conditions, identifying test requirements, designing test cases, defining and managing test data, executing test, checking results and many more are all part of the process.

Testing also includes reviewing of documents, including source code as well as static analysis. The thought process of designing tests early in the SDLC can help prevent defects from being introduced into the code.

Definition

Testing is the means by which people, methods, measurements, tools and equipment are integrated together in order to verify that a system satisfies its specified requirement. It identifies the differences between expected and actual results.

Testing is more than test execution.

Organization Activities:

- Policies and strategies
- Process Improvements
- Test Improvements
- Test Closure

Test Management Activities

- Test Planning
- Test Control
- Reviews
- Check Exit Criteria

Dynamic Testing

- Test Analysis
- Design test solution
- Implement the test solution
- Execute tests
- Manage results

Testing and Debugging - Not the same!

Is there a difference between testing and debugging?

Certainly!

- Testing highlights the failures that are caused by defects
- Debugging identifies the cause of the defect

The responsibility for each activity is very different

- Testers do the testing
- Developers do the debugging

Testing

Generally performed by the testing specialist but developers can be involved in component testing

Generates failures caused by defects Writes incident reports about the problem Run confirmation tests to ensure the correct fix Regression tests

Debugging

Performed by developers irrespective of the level of testing that generates failures

Identify the defect that caused the failure

Repair the defect in the code

Checks that the defect has been repaired correctly

Main Purpose of Testing The main purpose of testing is to find the bugs ... and get them fixed before implementation

How is this done?

Verification:

Is the process of evaluating a system or component to determine whether the products of the given development phase satisfy the conditions imposed at the start of the phase?

Validation:

Is the determination of the correctness of the products of the software development with respect to the user needs and requirements

Governance:

The ability of an organisation to measure and build its capability to repeatedly produce high quality software and deliver value to business.

The Role of Testing

Software Development

Any defects found by running tests in development can be corrected immediately so fewer defects are released Maintenance

Must ensure that enhancements and defect fixes do not affect the working system Operations

Ensure that systems continue to perform its intended function Contractual or Legal requirements
Some industry sectors, like safety-critical or pharmaceuticals, have their own requirements for testing

Testing Techniques

There are 2 terms to describe how software is tested

- Static - refers to testing something that is not running
- Dynamic - is something we would normally think of as testing, i.e. running and using the software

The easy way to describe the difference between these two types of techniques is to use the analogy of buying a second hand car: -

- Checking the condition of the tyre treads, paint work and engine are all static testing techniques
- Starting the engine and driving down the road are dynamic testing techniques

Both these techniques have similar objectives

- Provide information about the software or system
- Identify defects which will improve the system and give confidence
- Measure the aspects of the system (e.g. complexity or estimated defect density)
- Find ways to improve processes:

- Development process
- Testing process

What are the differences?

Static

- Software under test is not executed (run)
- Manual methods:
  - Reviews of code
  - Reviews of documents
- Automated methods:
  - Static analysis of code
  - Static analysis of documents
- Doesn't run test cases
- Detectable defects
-
  - Finds all occurrences
  - Finds defects early

Dynamic

- Software under test is executed (tests are run)
- Could be manual or automated
- Needs test cases, inputs, expected results, pre and post conditions
- Test design can help find defects early
- Sampling of all possible tests

Common objectives of these techniques

- To generate failures (dynamic testing)
- To find defects (static testing)
- Gain confidence
  - On the level of quality
  - To provide information
- Prevent future defects
- Manage Risk
  - Assess Risk
  - Mitigate risk

What are the benefits?

- Early detection and correction of errors
- Improvements in development productivity
- Reduction of testing time and costs
- Fewer defects in production
- Lifetime costs per function deduction
- Improved communication between parties

The Pareto Principle

Also known as the 80:20 rule, the funny law of nature

- 20% of clouds will produce 80% of rain or
- 20% of people do 80% of the work
- Just like any other activity in business, the 80:20 rule applies in testing
- The obvious 80% (the easy to see) will only find 20% of the bugs
- While difficult 20% will account for 80% of the defects
- It is very important to pay attention to detail
- One must be a lateral thinker
- Testing separates the "men from the boys"

Why is testing necessary?

What does software do?

Software accepts input, stores and manipulates data or performs calculations and sends the output to a screen or report

Why does software contain errors?

- Programs are written by people -

- Who know something, but not everything
- Who have skills, but aren't perfect
- Who are under increasing pressure to deliver to strict deadlines
- Who make mistakes
    - People who are under increasing pressure to deliver to strict deadlines
        - No time to check but assumptions may be wrong
        - Systems may be incomplete
    - The probability of changing code correctly on the first try is a 50% per 10 changes
    - 1 new defect will be introduced for every 5 fixed
    - There are 3 errors per 100 lines of code for every completed program
    - There is still 1 error per 100 lines of code after unit testing
    - Assumptions may be wrong
    - An application of +/- 400,000 lines of code will contain between 4000 & 12000 defects
    - There are no free rides -

Time not spent testing correctly in development Time spent debugging in production

Time Changes Everything!

- People have been writing software for the past 60 years
- The Lines of Code per application has increased substantially over the years
- They are more complex than before and getting worse
- They are a lot more inter-connected
- New versions of the software are released more frequently

The existence of system software

Where would we normally find Software?

- Everywhere (Banking, automotive systems, medical devices, household appliances, etc.)

Does it always work correctly?

- We tend to assume that it does, but is this always the case

Defects are present in Code, Systems and Documents.

What can be affected?

- Organizations (loss of money, time and reputation)
- Environment (tank overflow, radiation leak)
- People (medical devices)

Software errors

What is a software error?

- A software error is a mismatch between the program and its specification
- This is true IF, and only IF, the specification exists and is correct.

What is a Defect or "BUG"?

Let's look at a couple of definitions.

- ERROR (mistake): a human action that produces an incorrect result
- DEFECT(fault /bug): a flaw in the code or document caused by the error, which results application failure

One error may cause several faults or vice versa If executed, a fault may cause a failure

- FAILURE: deviation of the software from its expected delivery or service

The difference between these three items is the point at which it occurs in the development life cycle

The Error-Defect-Failure cycle

A person makes the mistake or error

That causes a defect in the software

Which causes a failure in the operation?

Therefore, a Failure is an event and a Defect is the state of the software caused by the Error

Causes of Software Errors

A distinction has to be made between the cause of a defect and the root cause

- The cause is the immediate reason why someone made an error, i.e. A misunderstanding of some kind, carelessness, etc.
- The root cause is an organizational issue that causes the employee to make this particular mistake, i.e. No training on the application.

Testing should be integrated into a full quality approach along with development standards, defect management processes, training, etc.

Causes and Root Causes

Causes of Defects (Product-specific)

- Misunderstandings, wrong assumptions, complex systems
- Error-prone tasks, carelessness, lack of experience
- Interactions between components or systems

Root Causes (Process issues)

- Ineffective communications
- Increasingly aggressive deadlines leading to excessive time pressures
- Lack of training due to budget restraints
- Race conditions

Reliability vs. Fault

Reliability is the probability that software will not cause the failure of the system for a specified time under specified conditions

- Can a system be fault-free? (zero faults, right first time)
- Can a software system be reliable but still have faults?
- Is a "fault-free" software application always reliable?

How much testing is enough?

We will always be in the position that we have too much to test and not enough time to run all the tests. It is impractical to

exercise all combinations of inputs and pre-conditions

How much is enough?

- it's never enough
- when you have done what you planned
- when your customer/user is happy
- when you have proved that the system works correctly
- when you are confident that the system works correctly

It will depend on the project constraints

- Time
- Budget
- Team Dynamics

You must be able to provide information to all stakeholders

- So that they can make informed decisions about moving to the next stage
- So that they can make informed decisions about releases to production

It will depend on the risk of application failure to your organisation

- Technical and Business (product/project)
- Implementation Complexity

RISK of application failure can manifest itself in many forms, e.g.

- risk of missing important faults
- risk of incurring failure costs
- risk of releasing untested or under-tested software
- risk of losing credibility and market share
- risk of missing a market window
- risk of over-testing, ineffective testing

Use RISK to determine ...

- The prioritization of your tests
- Where to place the emphasis
- What to test first
- What to test most
  - How thoroughly to test each item
  - What not to test (this time!)

There are many different criteria that can be used to prioritize tests so they can be used in combinations.

Possible ranking criteria include the following: -

- Tests that will find the most severe failures
- Tests that would find the most visible failures
- Tests that would find the most likely failures
- Ask the end-user to prioritize the requirements and test those first
- Test the areas of the software that contained the most defects in the past
- Test the areas of software that are the most complex or critical

There may be contractual obligations linked to custom-made systems, i.e. A contract between a software supplier and a customer may require the supplier to achieve 100% statement coverage.

Legal requirements may impose a degree of thoroughness in testing.

Safety critical industries, like railroad switching and air traffic control have their own standards defined with the intention of ensuring rigorous testing

So little time and so much to test ...

Test time will always be limited. So, what then, is the most important principle?

Allocate the time available for testing by prioritizing the tests so that whenever you stop testing, you have done the best testing in the time available

Who does the testing?

Programmer (with the exception of unit and component testing)?
Business Analyst?
Customer or the Business?

None of the above is correct. It's the Testing Specialist that should be doing the testing

What is a Test?

A test is a documented procedure that verifies a requirement of a system. Every test should contain the following:-

- Stated objective/purpose
- Description of the actions to take to navigate through the system
- Sample data to be used
- Detailed description of what to expect in the event of a successful test

What constitutes a successful test?

The primary objective of testing is to find faults.
Its secondary objective is to prove that the system works.
The strange thing about this is that testing on any project is usually done in the reverse order!

Characteristics of a good test

It has a reasonable probability of catching a defect

It is the best of its breed

It is neither too simple nor too complex

What is a test case?

A test case is a collection of Tests which a tester will use to determine whether an application or software system is working correctly or not.

In order to fully test that all the requirements of an application are met, there must be at least two test cases for each requirement: one to prove that the function works (positive) and one to find the errors (negative).

Written test cases should include a description of the functionality to be tested, and the preparation required to ensure that the test can be conducted.

Characteristics of a good test case

- Effective - It finds faults
- Representative - Cover more than one test condition
- Evolvable - It's easy to maintain
- Economical - It's cheap to use

What is a test scenario?

A test scenario is a collection of test cases and the sequence in which they should be executed. A test scenario can also be described as a collection of test cases that will ensure that all business process flows are tested end- to-end.

Test scenarios can consist of a set of independent tests or a series of tests that follow each other. Test scenarios are prepared by reviewing the functional requirements and preparing logical groups of functions. They are designed to represent both typical and unusual situations that can occur in an application

So ... Why is testing necessary?

- Because software is likely to have defects
- To learn about the reliability of the software
- Because failures can be very expensive
- To avoid being sued by customers
- To stay in business

Quality

Introduction

A system is a combination of functions and attributes. Functions that drive the business of an organization and attributes which include characteristics such as reliability, usability, maintainability, etc.

To know when you have it, you have to test for it. To test it effectively, you have to measure it. How do you measure the "immeasurable"? Well, according to Tom Gilb in his book 'Fundamentals of Software Engineering Management' anything can be made measurable in a way which is superior to not measuring it at all

Definition

The quality of a product depends on:

- The features that make the customer want to use the program
- The flaws that make the customer wish he'd bought something else (Kaner et al, 1990)

More definitions of Quality

- The Project Management Institute (PMI) defines quality as conformance to requirements and fitness of use.
- Quality is the degree of excellence that a thing possesses. The degree of conformance to a standard (Hutcheson, 2003)

What are the basics of Quality?

There are three key elements to quality (based on the Fundamentals of Six Sigma)

- Customer Satisfaction
- SQA Governance
- Employee commitment

What is Six Sigma?

Six Sigma in many organizations is simply means a measure of quality that strives for near perfection. It is a highly disciplined data-driven process that helps focus on the development and delivery of near perfect products and services. It was originally developed by Motorola in the U.S.A in 1986

The central idea behind the concept is that if you can measure the number of defects in an application you can systematically figure out how to eliminate them and get as close as possible to 'zero defects'. Six Sigma must become 'this is the way we work' - in everything we do and in every product, we design

Customer Satisfaction

- The customer defines quality
- They expect performance, reliability, on-time delivery, service, no defects and many more
- They demand proficiency and efficiency
- In short, exciting our customers has become a necessity for our existence.

SQA Governance

Governance deals with a series of processes dealing with the way we do things in the organization. Quality requires that we see our activities from the customers perspective and not our own. This means that we should look at our processes from the outside-in (If I were the customer ...) and create our processes accordingly. By understanding the customers' needs we can identify areas where we can add significant value and improvement.

Quality cannot be added on at the end. It must be built into the process

Employee Commitment

- Quality becomes the responsibility of every employee in the organization
- It should not be compromised in any way
- It's the people who create the results
- The business drives technology, who in turn should enable business. People make this happen

The Role of Quality

Quality Assurance

- Planned with systematic action to provide confidence in the product or process
- Quality Plan, Standards and Procedures
  - Training in development and testing
  - Defect analysis process (from test to reviews including root cause analysis)

Quality Control

- Products are monitored for conformance to standards
- Processes are monitored for conformance to procedures
- Checking that we have 'done the job right'
- Continuous monitoring

Testing and Quality

What does testing test?

- System function and correctness of operation
- Non-functional qualities like reliability, usability, maintainability, re-usability, etc.

Testing measures software quality (functional and non-functional).

Testing can find faults and when they are removed, software quality and reliability are improved.

Provides confidence in the application.

Reduces risk.

Testing proves quality.
Effective Quality Assurance is Expensive!

Compared to what? Not testing properly = faults in production (This is guaranteed). Not testing properly can also be expensive

What do Software faults cost?

Large sums of money

- Arianne 5 launch crash (development $7billion + the destroyed rocket with its cargo was worth $500m)
- Mariner1 space probe to Venus was destroyed 5 minutes after launch because it veered off course ($250m)
- American Airlines Flight 587 ($150m - it took 6 years to complete the investigation)

(Customer satisfaction says it all!!!!)

How do we implement Quality?

Very little or nothing at all

- Minor inconvenience
- No visible or physical detrimental impact Software errors are not "linear":
- Small input may have very large effect - (A period in the wrong place caused a R20m loss to a mixing process at SASOL)

Software faults can also cause death or injury

- radiation treatment kills patients (Therac-25)
- train driver killed
- aircraft crashes (Airbus & Korean Airlines)
- bank system overdraft letters cause suicide

Defect escalation

What happens when a defect is found?

- Reported
- Fixed
- Fix gets distributed
- Deployed

How is Quality measured?

Features that make the customer want to use the product

- Flaws that make the customer wish he had chosen something else
- Ability to carry out Business Rules

- Proper requirement extraction forms the foundation on which the entire testing effort is based (measure twice ... cut once ...)
- Testing is a team effort that ....
  - Requires a Quality Culture.
  - Requires a Quality Maturity.
- Quality vs. Time vs. Resources.
- Quality does costs money.
- Poor quality also costs money.

- People make quality happen by..
  - Quality Assurance.
  - Good management.
  - On-going processes.
  - Quality Control.
  - Effective Testing.
  - Validation
  - Verification.

One last word on Quality

Quality cannot be added on at the end of the process, it must be built into it.

What do you think will happen if we add the eggs to the cake after we have finished baking it??? Remember:

Quality is never an accident

It's the result of .....

High intentions

Sincere Effort

Intelligent direction ..... and

Skillful execution *It represents the wise choice of many alternatives*

General Testing Principles **Introduction**

Effective testing is a very complex activity, which can be very difficult to master. What follows is a set of general testing principles, that have been developed over the years from a variety of sources, to help testers understand what is required to perform the activity effectively. They are not all obvious, but their purpose is to guide testers, and prevent problems.

Principle 1 - Testing shows the presence of defects

- Testing can show that defects are present, but cannot prove that there are no defects.
- It reduces the probability of undiscovered defects remaining in the software
- No matter how much testing we do, we can't prove there are no defects
- Even if no defects are found, it is not a proof of correctness

Principle 2 - Exhaustive testing is impossible

Exhaustive testing is a test approach in which all possible data combinations (all combinations of inputs and preconditions) are used.

How much time will exhaustive testing take?

- infinite time
- not much time
- impractical amount of time

Testing everything is not feasible except maybe for trivial cases. Suppose our application contains the following specs:-

- Contains 20 Business Functions
- There are an average of 4 Options per Function
- There are an average of 10 fields per Option
- Each field has 100 possible values

How many tests will we need to perform exhaustive testing on this system?

20 x 4 x 10 x 100 = 80,000

How long will it take to run all these tests?

Suppose we allow an average of 10 minutes per test

80,000	**Tests**		
	(* 10)	**800000**	**mins**
	(/ 60) (/	**13333**	**hours**
	5.5 h) (/	**2424**	**days**
	22 d)	110	**months**
		9	**years**

Calculated in 'man' hours, days, months and years

It's important to note that testing time on any project is limited (we do not have the luxury of having all the time that we need). This means that we will have to make a choice as to which requirements need the most attention

How we make this decision will depend on the risk of function failure to the business. In some cases it may just be a minor inconvenience but in others it could result on millions of rand. Instead of exhaustive testing, risk analysis and priorities should be used to focus testing efforts.

Principle 3 - Test activities should start as early as possible

Many problems in software systems can be traced back to missing or incorrect requirements.
In early testing we are trying to find errors and defects before they are passed to the next stage of the development process.

Testing activities should start as early as possible in the software or system development life cycle. Testing should focus on:-

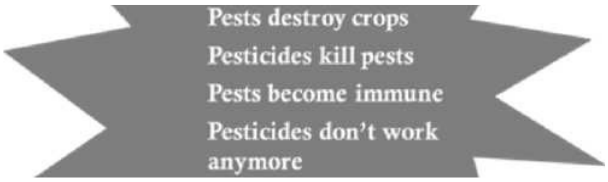
- Meeting deadlines
- Defined objectives
- Increasing quality

Principle 4 - Defects tend to stick together

Defects are social creatures and tend to hang around together. The spread of defects is not uniform. In a large application, it is often a small number of modules that exhibit the majority of the problems.

This is the classic application of the Pareto principle to software testing: (80% of the problems are found in 20% of the modules

Principle 5 - Pesticide Paradox



True or False?

- Tests should find defects
- Tests should be repeatable
- Same test, same software - find new defects?

If the same tests are repeated over and over again, eventually they will no longer find any new defects. To overcome this "pesticide paradox", the test cases need to be regularly reviewed and revised. New and different tests need to be written to exercise different parts of the software or system to potentially find more defects.

Principle 6 - Testing is context dependent

- Testing is done differently in different contexts e.g. safety-critical software is tested differently from an e-commerce site.
- Testing should not be the same everywhere
- Risk can be a large factor in determining the type of testing that is needed.
- The higher the possibility of losses, the more we need to invest in testing the software before it is implemented.
- Good testing does not just happen. It has to be planned and systematically executed

Principle 7 - The absence of errors fallacy

Finding and fixing defects does not help if the application is unusable and does not fulfil the users' needs or expectations. Software with no known errors is not necessarily ready to be deployed. Testing can show the presence of defects but cannot guarantee 'zero-defects'

The Basic Test Process

Introduction

The basic test process is made up of five levels:

- Test Planning and Control
- Analysis and Design
- Test Execution
- Defect Management
- Test Closure Activities

These activities can be sequential or concurrent

The process breaks down even further into four steps, dealing primarily with Analysis, Design, Implementation and Execution

- Analyse requirements
- Identify test conditions
- Develop and specify test cases and test data
- Develop test procedures

Test Planning Activities

There are a number of different levels of Test Planning

- Company level deals with the Testing Policy and Testing Strategy of the organization
- Project Level deals with the High Level Test plan. (one for each project)
- Test Phase Level involve individual plans for the different phases in the SQA Lifecycle (Test Execution, Implementation, Pilot Testing, etc.)

Test planning include some of the following tasks:- Determines the scope of testing.

Defines the business processes to be tested.

Defines the functional & non-functional attributes to be tested and not to be tested. Identifies approach to testing.

Identifies test techniques to be used.

Determines the entry and exit criteria.

Determines the test environment.

Test Control Activities

Test control can be thought of as the test management tasks that is required throughout the test process in order to keep the testing aligned with the software development process, the needs of the project, and the needs of the organization

Control of the test plan include the following tasks:

- Compare actual progress against the plan
- Report the status, including deviations from the plan
- Take corrective or pre-emptive action in order to meet the objectives

- Control involves monitoring throughout the project

- Progress
- Coverage
- Exit criteria
  - Make decision
    - Who, what, when, where, etc.

Test Analysis and Design

This phase can be broken down into four distinct tasks:

Identify: Determine 'what' is to be tested (identify test conditions) and prioritise Design: Determine 'how' the 'what' is to be tested, (i.e. design test cases)

Build: Construct the test repository Implement: Define test processes

Identify Test Conditions

Determine 'what' is to be tested and prioritise. List the conditions that we would like to test:

- Use the test design techniques specified in the test plan
- There may be many conditions for each system function or attribute e.g.
  - "Life assurance for a racing drivers"
  - "Discounts based on volume"
  - "Special dates, e.g. 29-feb-2012"
- Prioritise the test conditions
  - Must ensure most important conditions are covered

Design Test Cases

Determine 'how' the 'what' is to be tested

- Design test input and test data
- Each test exercises one or more test conditions

Ij

Determine expected results

- Predict the outcome of each test case:
  - What is output
  - What is changed
  - What is not changed

Design sets of tests

- Different test sets for different objectives such as regression, building confidence, and finding faults

Build

Construct the test repository

- Prepare test scripts
  - The less system knowledge the tester has the more detailed the scripts will have to be
  - Scripts for Test Tools have to specify every detail
- Prepare test data
  - What are the input values?
  - What are the execution pre-conditions?
  - What are the expected results?
  - What are the execution post-conditions?

Data that must exist in files and databases at the start of the tests

Implement

Define test processes

- Define the sequence of execution
- Identify test dependencies
  - Identify if the predecessor should only be run
  - Identify if the predecessor should have run and have a status of 'passed'
- Identify exit criteria per Test cycle
  - A test cycle will be considered to be complete when all tests, both manual and automatic, have been run and all have passed their respective check criteria

Set up the test environment

Identify test implementation tasks

- Create test data
- Identify 'Risk' indicators in order to prioritize testing activities

Test Execution

Actual execution of the identified tests

- Most important ones first
- Would not execute all test cases if:
  - Testing only fault fixes
  - Too many faults found by early test cases
  - Time pressure
- Identification and execution of the test that have been scheduled.

It is dependent on the release of software and what is contained within it.

- Can be performed manually or automated

Defect Management

The defect management process is based on the following general principles:

- The primary goal is to prevent defects. The goals are to both find the defect as quickly as possible and minimize the impact of the defect.
- Defect measurement should be integrated into the software development process and be used by the project team to improve the process.
- Defect information should be used to improve the process. This, in fact, is the primary reason for gathering defect information.
- Most defects are caused by imperfect or flawed processes. So to prevent defects, the processes must be improved.

Test Closure Activities

- Ensuring that your systems meet their functional requirements and perform as expected is the primary reason for testing
- Completion or exit criteria applies to all levels of testing to determine when to stop
- Assess test execution against the defined test objective for
  - Coverage, using a measurement technique, e.g.
  - Branch coverage for unit testing
  - User requirements
  - Most frequently used transactions
  - Actual vs. expected results
  - Cost or time
- Test completion tasks
  - Check the test logs against exit criteria specified in the test plan
  - Assess if more tests are needed, or exit criteria should be changed
  - Write a test summary report for stakeholders
- Consolidate experience, test ware, facts, etc.
  - Based on data from completed activities
  - Releases
  - Test project
  - Milestones
- Test closure activities
  - Check planned deliverables, close incidents, raise change notices, document system acceptance, etc.
  - Finalize and archive the test-ware, environment and infrastructure for later reuse
  - Handover the test-ware to the maintenance team
  - Analyse lessons learned to improve test maturity

The Psychology of Testing Different mindset

The point of view of a developer is different to that of a tester. When a developer is working with code, their focus is on what the code should do and writing programs that will do what is required. It is possible that they may encounter technical restraints within which they must operate, but their

main aim is to get the program to work.

The tester's point of view is also concerned with understanding the requirements, but their approach is to try and break the function or consider what could go wrong. Another way of describing the testers point of view is to look at the function from a negative perspective

So the BIG questions that needs to be answered is

- Can or should developers test their own work? Or
- Is a developer capable of testing their own work?

The answer to both these questions in 'Yes', but ...It's going to be extremely difficult for a developer to 'change gear' and start thinking testing when his mind is firmly in developer mode. In addition, some developers are very good at testing their own work, other aren't and will not be able to see the problems.

Most of us are 'blind' to some extent to our own mistakes. It is because of this that best practice dictates that it is far better to use someone independent of the developer as they are likely to find more defects.

Aspects of Independence

People are driven by objectives

- 'Find defects' vs. 'Confirm that the software works'
- Clear objectives are very important

Independent testing can be done at any level

- Component, integration, system, acceptance, etc.
- Reviews are a form of independent testing too

Independence does not replace familiarity

- Developers can test their own work ..... but
- Independent testers will find more or different defects

Who wants to be a Tester?

It is a destructive process, as you are the bearer of bad news *("your baby is ugly").* They work under worst time pressure (at the end), and need to adopt a different view, a different mindset

- "What if it isn't?"
- "What could go wrong?"

How should fault information be communicated (to authors and managers?)

- Constructively, correctly, objectively, fact-based, neutral, etc.
- Focus on the common goal of better quality systems
- Confirm that both sides have understood correctly

- Understand how the other person feels

Testers Bill of Rights

A Tester has the right to:

- Accurate information about:
  - progress of software to be tested
  - changes to the software that may affect your tests
- Insight from developers about areas of the software that:
  - are difficult to test
  - contain a high number of faults
  - or hard to reproduce faults
- Delivered code tested to an agreed standard (testable state)
- Regarded as a professional
- Find faults
- Challenge any specification for omissions, inconsistencies, ambiguity, etc.
- Comment on areas that are not under test
- Be involved in reviewing specification documents on which the tests are based
- Suggest any additional tests not in the plans
- Have faults report taken seriously even if they are "soft" such as user interface problems
- Some effort directed at faults that cannot be reproduced
- Information about faults missed in testing
- Make predictions about future fault levels
- Improve their own testing process

A Tester has a responsibility to:

- Report faults objectively and factually (no abuse!)
- Check that tests/data were correct before reporting an incident as a software fault

- Not over-step the mark :
  - Report on the quality of the software and risks
  - Don't make the release decision
- Remember it is the software, not the programmer, that is being tested
- Follow the test plans, scripts etc. As documented
- Be careful when testing outside planned scope
- Prioritise what is reported
- Communicate the truth

Destructive or Constructive?

Testing is both

Destructive

'Breaks' the software (much like a toy's safety test) Constructive

Manages, and therefore reduces, product and project risk Defect information can help improve development skills Defects found early will save time and money

Developers are not the enemy

When testing is done by an independent group, a level of animosity can build between testers and developers. This can be very damaging to the organization

Ways to improve communication and build relationships include

- Remember that you both have a common goal
- Start with co-operation rather than confrontation
- Communication about defects should be factual, objective and neutral
- Be sensitive to the other persons feelings
- Use 'active listening' to confirm that communications have been correctly understood (echo back by both parties)

The Importance of Testing Skills

Which is more important?

- Extensive business knowledge
- Sound knowledge of structured testing techniques

Why?

Where good testing skills are lacking, the tester has to rely on extensive business knowledge to derive the tests - Such tests are highly subjective and lead to gaps in the test coverage A tester that lacks the business knowledge is more objective and analytical in the test process

Code of Ethics Introduction

Involvement in software testing enables individuals to sometimes come across confidential and privileged information. A Code of Ethics is necessary to ensure that the information is not put to inappropriate use What follows are the minimum requirements recognized internationally and each organization will probably add additional codes to this standard

Contents of the Code

Public:

- Act in a manner that is consistent with the public interest

Client & Employer

- Act in a manner that is in the best interests of both the client and the employer, consistent with public interest

Judgment

- Maintain integrity and independence in your professional judgment

Management

- Test managers and leaders shall subscribe to, and promote, an ethical approach to the management of software testing

Profession

- Certified software testers shall advance the integrity and reputation of the profession consistent with public interest

Colleagues

- Always be fair to and supportive of your colleagues and promote co-operation with software developers

Self

- Always participate in lifelong learning regarding the practice of your profession
- Always promote an ethical approach to the practice of the profession

1.1.1 The Importance of Independent Testing

- It is important to realise that organisations will have different requirements when it comes to testing.
- Different stages of testing will be performed within these organisations with varying degrees of independence using different approaches
- Independence is important for effective testing
- Greater independence gives a more objective view of the item under review
- If we plot the no. of defects over time, it would probably rise steeply at first and then begin to decline. (the hump of pain)
- Releasing the product at this point usually results in the no. of defects starting to rise again
- The reasons for this is that users have a different 'view' of the product as they start to use the application in a live operation
- If our aim is to find as many defects as possible, we need to have as many of these different 'views' as possible and as early as possible
- Our challenge is to ensure that the defects are found during testing and not the end user
- There are advantages of both familiarity and independence and one should not replace one with the other - both are needed
- Independent testers provide a more objective assessment of the software and can find defects that people with detailed knowledge of the application would miss

Test Organisation and Independence

Benefits

- Developers test their own code (Not independent)
- Developers test each other's code (buddy)
- Tester within the development team (Prevent defects)
- Tester team reporting to Project Manager or higher (diminished authority)
- Testers from business and IT (objective testing)

1.1.2 Benefits and Drawbacks of Independent Testing

- Test specialist e.g. certification, security (subjective testing)
- Tester from an outside organisation (3rd party/outsourced testers)

Examples of Independence

- Large complex or safety critical
  - Multiple levels of testing
  - Independent testers at some (or all) levels
- May define test processes and rules
  - Testing by developers at lower levels
- Lack of objectivity could limit effectiveness
- Internal application
  - Informal component test by developers
  - Independent acceptance test by internal users
  -

Independent testers see other and different defects and are generally unbiased Verify assumptions people made during specification and implementation of a system Developer concentrates on delivering the project, the tester concentrates on using the project, so the difference in mindsets will result in a better product

Drawbacks

Can be too isolated from development team
Independent testers can prove to be a bottle neck to getting the application released, resulting in the schedule not reaching deadline
Because of delayed deliverables, cost may go up
Developers may lose a sense of responsibility for quality and feel others should pick it up

1.1.3 Identify Team Members in terms of Skill Level.

Definition of Management

Management is the act of coordinating the efforts of people to accomplish desired goals and objectives, using available resources efficiently and effectively.

Effective management involves five basic functions Planning Control
Organize (including delegation)
Leading/Directing Staffing (incorporating motivation)

Two of these functions have a direct effect in Test Analysis, Planning and Control

The Ideal Tester

One of the requirements of a successful testing policy is to ensure that the testing team is made up of trained skilled people. This means that each member of the team should meet the all the characteristics of the ideal tester as well as fit the recommended capability profile.

The capability profile is measured using the following criteria: -

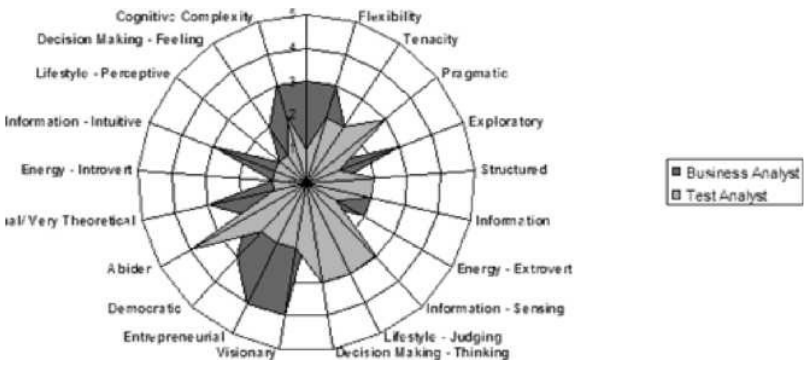
- Capability refers to the "nature" of a person and cannot be taught or easily changed.
- Matching the nature of the person with the nature of the job ensures sustainability.
- Competency is comprised of Knowledge, Skills and Attitude.
- It refers to the aspects of the position that can be taught.

The ideal characteristics of a tester are measured by the following criteria

- Understands the application (BA)
- Understands the computer (CS)
- Lateral thinker
- Good communicator
- Pays attention to detail
- Committed to Quality

Business and Test Analyst Preferred Capability

Profile



Testing Roles on a Project

Testing on a development project typically require people fulfilling the following roles

- QA Manager
- Test Manager
- Test Analyst
- Test Designer
- Tester
- Automated Tools Expert
- Test Scripter
- Test System Administrator

Project Sponsor or Champion

The role of the project sponsor/champion is to ensure that the projects support the business strategies. He is the link between the business and the project.

Project Manager

The main responsibility of the project manager is to deliver the project on time and within budget and scope through normal best/common practices aligned to this role. Convene regular project progress meetings. Report overall progres. He/she must also manage all relationships: internal as well as external.

Test Manager

The test manager is responsible for creating a test plan relating to every task and deliverable of the test team. It extremely important that the test plan compliments the project plan

Business owner/Product owner

The business/product owner is responsible for managing all activities relating to the business community. They are also responsible for managing requirements and the resulting scope creep

Business Analyst

The business analyst is responsible for a number of activities, some of which include:-

- Manage the business need
- Analyse documents and organize functional requirements
- Facilitate JAD sessions
- Produce various models and diagrams
- Review and approve all testing documentation
- Take part in acceptance testing

Systems Analyst

The system analyst is responsible for the technical design and program specifications of the product as well and quality assure all code and components

Developers

The developer is responsible to produce program code and conduct unit tests to validate what has been produced. They are responsible for test cases, test data and test execution of all unit tests

Quality Assurance Manager

The quality assurance manager is responsible for the following: -

- Manages and co-ordinates the testing effort
- Responsible for the master test plan:
- Project schedule
- Requirements
- Gatekeeper
- Manages company standards and procedures
- Manages automated testing tool selection

Test Manager

This role and tasks will be discussed in Study Unit 4.1.4.

Test Analyst

Like the Business Analyst, the TA is responsible for a whole lot of different tasks like:

- Mine requirements using documentation available as well as interviewing subject matter experts
- Produce the test requirements from the sources above
- Develops the test scenarios (thus identifying test cases)
- Identifies test data
- Analyse all documentation produced on the project
- Where there is no or very little documentation available and in case of legacy systems, the source code can also be scrutinized in driving out requirements

The Test Analyst role is also responsible for:-

- Effectively manage test requirements
- Identify and define all required test scenarios
- Monitor test coverage and evaluate the overall quality experienced
- Evaluating the outcome of each test cycle
- Gathering and managing the Test Data

Each of these responsibilities is detailed as you proceed through the course guide.

In addition to responsibilities mentioned before, the test analyst's role also includes the following appropriate skills and

knowledge:

- Good analytical skills
- A challenging and enquiring mind
- Attention to detail and tenacity
- Understanding of common software failures and faults
- Knowledge of the domain (highly desirable)
- Knowledge of the system or application-under-test (highly desirable)
- Experience in a variety of testing efforts (desirable)

Everything that the test analyst needs to know and do are all bundled together and referred to as the Core Competences of the Test Analyst

Core Competences of the Test Analyst

The core competencies of a Test Analyst are as follows:

- Produce test requirements
- Know and understand different Test Techniques
- Develop test scenarios
- Identify test cases
- Identify test data and database requirements
- Identify the test cases that are candidates for automation
- Review defect reports by tester
- Conducts review sessions of test scenarios, cases
    - Take part in reviews session, where documentation (Functional specs, user manuals) is review
- Sign off documentation (as above)
- Take part in peer reviews
    - Defect management: All aspics, including reporting to project or line management
- Conduct JAD sessions, in order to drive out requirements
- Document legacy systems (using context diagrams, flowcharts)
    - Make recommendations to Test Manager in terms of processes, defect management

It must be noted that the core competencies of a Test Analyst are built over time (anything between 3 and 5 years) as the individual advances their career as a Testing Professional

The level of proficiency in these tasks would also differ for Junior to Senior Test Analysts from "Limited Practical Experience" to "Expert, well-versed".

Test Designer

The test designer is responsible for creating test cases that will validate as many requirements using the minimum amount of test cases. This is achieved using the different test design techniques available in the IT industry: -

- Creating test that will provide 100% requirement coverage
- Design tests that will find all the errors
- Make use of the different test design techniques to obtain maximum coverage with the minimum amount of tests
- Ensure that there are sufficient tests to compliment all test scenarios identified by the test analyst

Tester

This role and tasks will be discussed in Study Unit 4.1.4.

Automated Tools Expert

- Acts as a reference for all problems relating to automated tools
- Trains other users
- Creates special custom functions
- Tunes the test environment
- Interfaces between QA department and the automated tool supplier
- Assists with Automated Testing Tool selection

Test Scripter

- Performs the same role as the tester but users an automated testing tool
- Does need to have programming capability

Test System Administrator

- Manages the automated test system tools and the test-ware repository
- Administers the test environment: -
  - Testing software installation
  - Maintenance and customization
  - Back-up and archiving
  - Database refresh procedures
- May generate utilities

Stakeholders on a Project

A stakeholder is defined as anyone who has a vested interest in the project This includes marketing, training, users, technical experts, etc. In addition to testing roles, a development project also consists of the following stakeholders:-

- Project Sponsor/ Champion

- Project Manager
- Business Representative or owner/Product Owner
- Business Analyst
- Systems Analyst
- Developer

1.1.4 The Tasks of a Test Leader and a Tester

In smaller companies the test leader is usually the test manager, but in large organisations there may by several test leaders, doing the task of the test manager, but reporting to the overall test manager.

Test Manager

The test manager on a project is responsible for the following: -

- Produce test plan per project.
- Manage the production of the test cases.
- Define the test infrastructure requirements and manages the creation of the necessary test environment where this does not exist.
- Provide the project manager with the cost implications of creating this environment.
- Define any test tool requirements and ensure that the test resources are trained in the use of them.
- Define the error reporting process.
- Manage the execution of tests and tracking of defects through all planned test phases

The Role of the Tester

Introduction

Testers are 'special' - in a good way! As tester you

- act as an auditor would, having the ability to focus on detail
- have the staying power which enables to concentrate at consistent levels for long period of times
- be innovative in finding defects
- be self-assured when presenting new ideas
- be able to interact with project team members on all levels
- be able to work under tremendous pressure

'Special' people, indeed.

You are the Headlights

A project is like an endurance rally (like the Paris to Dakar motor rally).

- Sometimes you drive under good conditions in broad daylight (simple, routine projects)
- Sometimes (more often than not) you are driving at night over treacherous terrain. These are the complex

projects in need of good headlights. You are the headlights.

- Testers light the way - developers and managers, while bickering over the map, see where they are because of the information you give them
- Testing is done to find information. Critical decisions about the project or project are made on the basis of that information

The Organizations' QA policy

The company's QA policy drives everything you do. You may need to

- follow standards, *and/or*
- test until all known defects have been fixed, *or*
- find the important defects before going live and fix as you go afterwards, *or*
- provide only a general assessment of the quality of the project

The clients you serve

- Project Manager
- Reporting test status on demand, in standard way
- Developers
- Business Analysts (functional documents need to be QA'ed)
- Production Support
- Marketing (they need to know about known defects)
- Training (documentation)

Working closely with developers

As tester you

- Support developers. Your feedback helps them to work more efficiently.
- Should aim for the quickest feedback loop
- Remember - developers are not the enemy

Question everything

As a tester you need to question everything, but not necessarily out loud. If you do not question, your testing will be mechanical and will get boring. Even though you will probably test using test cases compiled by the Test Designer, you need to question these too if you have to. People are not perfect, they make mistakes.

Remember the saying 'you catch more flies with honey'. The way in which you question will make life easier for the person being questioned as well as for you. Small doses as well as gentle if the situation calls for it are good rules of thumb.

Being a tester ... and thinking like one

Testers have to offer evidence as to why software does not work. In order to do this, they have to test - and report on the results - in a meticulous fashion.

Testing may seem semi-automatic, while you just follow test cases on a day-to day basis. However, it is possible to start thinking like an exceptional tester, even if you have to follow test cases. By applying the information in this chapter, you might be able to share your insights with the analysts and test manager, maybe devising new ways of testing or identifying gaps in the current test strategy or test cases.

How to get a jump-start as a tester

Learn more about the industry you are going to work in (e.g. Banking or Insurance). Learn more about the type of product you are going to test (e.g. new credit card, new internet site, insurance product)

Being a better than average tester

More tips on getting better than the rest:

- 'Tour' the system, not only the bits you must test. Get a feel for the whole, as opposed to your section. Do this by using the "Help" function.
- Ask the Test Analyst or Test Manager for documentation on the system. This could include the functional and technical specifications as well as training and user manuals. Read all of it.
- Think creatively. Imagine what could possibly go wrong, what possible problems there may be. Share this with the Test Analyst.
- Think critically: Evaluate your ideas and make inferences.
- Read up on competing products, related products, comments from customers on sites like www.hellopeter.com

Study Unit 1.2: Conduct Test Planning and Estimation

US 386054

Manage the software testing process Specific Outcome 1

Conduct test planning and estimation

Assessment Criteria

The purpose and content of documentation.

1. the test plan are summarised according to the standard for software testing

Test approaches are

2. differentiated between in order to plan successfully.

A test execution schedule is

3. developed for a given set of test cases.

4. Preparation and execution activities are identified, and explained for a test case.
5. Typical factors that may influence the test effort are listed and explained with examples.
6.

Adequate test entry and exit criteria are described for specific test levels and groups of test cases.

ASSESSMENT CRITERION RANGE

Levels may include but are not limited to: Component testing, Integration testing, Acceptance testing and System

1.2.1 The Purpose and Content of the Test Plan

The Basic Test Process

The basic test process is made up of five levels:

- Test Planning and Control
- Analysis and Design
- Test Execution
- Defect Management
- Test Closure Activities

Test Planning

- What is the purpose of a test plan?
  - Who does it communicate to?
  - Why is it a good idea to have one?
- What information should be in a test plan?
  - What is your standard for contents of a test plan?
  - Have you ever forgotten something important?
  - What is not included in a test plan?
- There are a number of different levels of Test Planning on a project
  - Company level deals with the Testing Policy and Testing Strategy of the organisation
  - Project Level deals with the High Level Test plan. (one for each project)
  - Test Phase Level involves individual plans for the different phases in the SQA Lifecycle (Test Execution, Implementation, Pilot Testing, etc.)
- As a Test Analyst you will probably be involved in Test Phase level
- Apply company testing approach or strategy
  - Test levels, entry and exit criteria
- Document any exceptions
- Reasons for non-compliance must be documented
  - Consistent approach to testing is encouraged
  - Regular feedback on the test strategy
- Provides opportunity for keeping strategy up to date and relevant
  - Source: ANSI/IEEE Std. 829 "Standards for Software Test Documentation

A software project test plan is a document that describes the objectives, scope, approach, and focus of a software testing effort. The process of preparing a test plan is a useful way to think through the efforts needed to validate the acceptability of a software product. The completed document will help people outside the test group understand the 'why' and 'how' of product validation.

Why is Test Planning so important?

- Determines the scope of testing.
- Defines the business processes to be tested.
- Defines the functional & non-functional attributes to be tested and not to be tested.
- Identifies approach to testing.
- Identifies test techniques to be used.
- Determines the entry and exit criteria.
- Determines the test environment.

Factors that influence Test Planning

There are many factors that can influence test planning some of which include: -

- Test Policy
- Constraints
- Scope of Testing
- Test Objectives
- Resource Availability
- Risk
- Project
- Application
- Criticality
- Development Methodology

Contents of the Test Plan

The contents of a test plan have been defined in the IEEE 829 standard. There are rather a lot of items that need to be included in a test plan. To help remember, some fundi in the industry devised acronym, SPACE DIRT that maps all the headings that have to be included in the test plan.

The contents of the test plan are discussed in detail in the Test Manager module

Test items, Features to be, and not to be tested

Scope

People

Staffing and training needs, schedule responsibilities

Approach

Approach, Test Strategy

Criteria

Acceptance, Suspension and Resumption criteria

Environment

Environment

Testing deliverables, schedules

Deliverables

Test Plan Identifier, Introduction, Approvals

Incidentals

Assumptions and Dependencies

Risks

Risk and Contingencies Testing Tasks

Tasks

1.2.2 Differentiate between Test Approaches

Test Strategies (Approach)

- How do we know what to do and when
- Two main approaches defining when test design starts
  - Preventive- as early as possible
  - Reactive- responding when the application is delivered
- Examples
  - Pharmaceutical: analytical risk based), industry standards, methodical (check list)
  - Startup package: consultative (potential users), dynamic, heuristic (exploratory)
  - Legacy system: regression averse, model- based (operational profiles)

Typical Test Approaches to Consider

There are many different approaches to consider: -

- Consultative
- Depends on the technology or expert
- Regression
- Re-use and automation
- Analytical
- Focus on risky areas
- Model based
- Statistical information

- Methodical
- Failures and Checklists
- Dynamic and Heuristic
- Exploratory
- Process or standard
- Industry specific

These approaches can also be combined

Which Test Approach?

Selection of a test approach based on context

- Risk of failure of project
- Impact to people and the organisation
- Skills and experience of staff in the various techniques, tools and methods
- The objective of the testing mission
- Regulatory considerations
- Nature of product and business

Testing Budget

- The Testing Budget represents the total number of hours in a testing cycle.
- It is calculated by adding together the total number of hours each team member will be available to the project during the testing cycle.
- Testers must remember to make provision for times that are required for non-project activities
- The progress of the project is monitored by measuring the time available against the amount of work still to be completed

Test Types

Introduction

Some types of testing you may come across:

- Functional (this is where you do the bulk of your work when you start off as a tester)
  - Acceptance
  - Regression
- Non Functional
  - Performance Testing
- Load.
- Stress
- Spike
- Soak
- Capacity
  - Usability Testing
  - Configuration Testing
  - Security Testing

- Automation
- Re-Testing
- Maintenance Testing

Functional testing

The tester will be involved in two kinds of functional testing and is where the bulk of the work is done when starting off as a tester

- Acceptance Testing
- Regression Testing

In this type of testing you test the functions of a new system/feature. It is performed on the elements that the tester can see (inputs, outputs). The tester is not able to see what is happening with the process (he cannot see the code)

Acceptance Testing

- Acceptance testing is a final stage of testing that is performed on a system prior to it being delivered to a live environment This kind of testing is known by a variety of names, e.g. User Acceptance, Customer Acceptance, Formal Acceptance and Final Acceptance to name a few)
- During this type of testing the specific focus is the exact real world usage of the application.
- The Testing is done in an environment that simulates the production environment.
- The Test cases are written using real world scenarios for the application

Regression Testing

- Focuses on the areas that did not change
- Looks for any unexpected side effects
- Test all the functions in an application
- Regression tests should be carried out in 2 phases
  - The first phase should concentrate on the areas that, even though they did not change, they are linked to the changed area via a common program or screen. This testing must form part of the testing scope of a project.
  - The second phase of regression testing must test the whole application concentrating on the unrelated areas of the application. This testing is usually carried out in an independent phase in the test plan and should be the final test before deployment to production

Non-Functional

The focus here is on the response time to a request from the user under varying load conditions. The following lists the different kinds of

Performance testing

- Load testing is done to determine the capability of the system to handle the anticipated number of concurrent users without much degradation in system performance.
- Stress testing is conducted to identify the peak load handled by the system.
- Spike testing is to determine behaviour of the application when subjected to sudden high activity
- Soak testing is to subject the application to running prolonged peak over time
- Capacity testing is to determine the amount of load the system can handle before breaking down

Usability testing

Usability is not (only) the QA department's responsibility although QA has to offer opinions. Usability has to do with the user-friendliness of a system. Attention is paid to:

- Error messages
- Feedback ('wait' messages)
- Readability
- Help functions (precise, constructive, available)
- Navigation
- Error recovery
- Support for human memory (short-term memory load should be manageable)

Configuration testing

Configuration testing verifies how the product works on different hardware and when combined with different third party software. There are typically many combinations that need to be tried; therefore configuration testing is not cheap. But, it is worth it when you discover that your standard in-house platform which "entirely conforms to industry standards" actually behaves differently from most of the machines on the market.

Security Testing

In this type of testing the following is tested:

- Passwords
- Encryption
- Levels of access to information Learner Guide
- Authorization
- Firewalls

Automation

Introduction

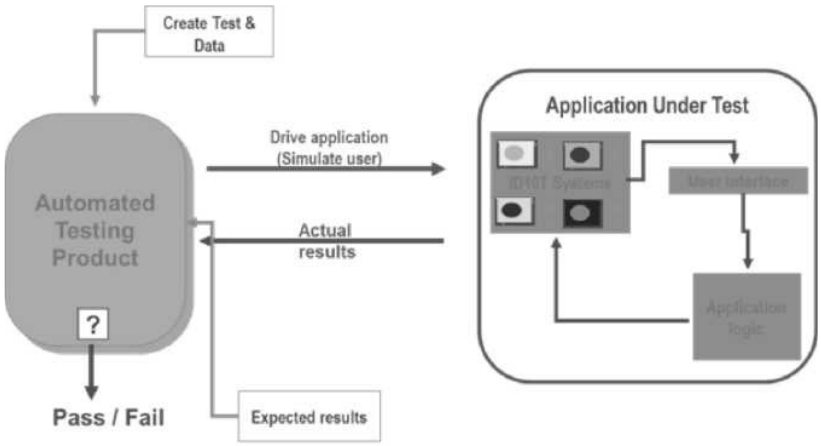
What is Test Automation?

- It is software testing software
- Software that can do much of the boring and repetitive testing work

The automation testing product must perform three major functions:

- Automatically drive the application. This is usually achieved by simulating the user actions using the User Interface
- Test the current operation of the application at a given moment
- Compare the current operation of the application to a documented expected result to verify if the current operation is correct.

When the test has ended, a designated person must be informed of the status of the run. In the event of a failure, the test must also cater for exception handling. All of this must be done without human intervention.



Why Automate?

Automation is not a fix-all. There is a myth that testing happens and the push of a button, automatically reporting on every defect found. This is unfortunately is all it is, a myth:

Technical testers are needed to automate test cases and analyse failures before they are reported. However, extensive and thorough the test suites are, automation is helpful in certain situations:

If, for contractual or legal reasons, you have to prove that the final product underwent an extended battery of tests

- Regression testing
  - Informal surveys reveal that the percentage of defects found by automated tests is surprisingly low. (Kaner, 1999)
- Automating some of your testing might or might not be useful.
  - Pros
    - It saves time
    - Extend your reach (more tests in the same amount of time
    - Automated test scripts can be used to create large files and lots of test inputs
  -
    - Cons

Can distract you ▪ and waste resources

A word of warning

Don't try and equate manual testing to automated testing. When testing is performed by hand, an entire range of human capabilities are brought to bear. A tester can improvise new tests or notice things that he did not or could not have

anticipated.

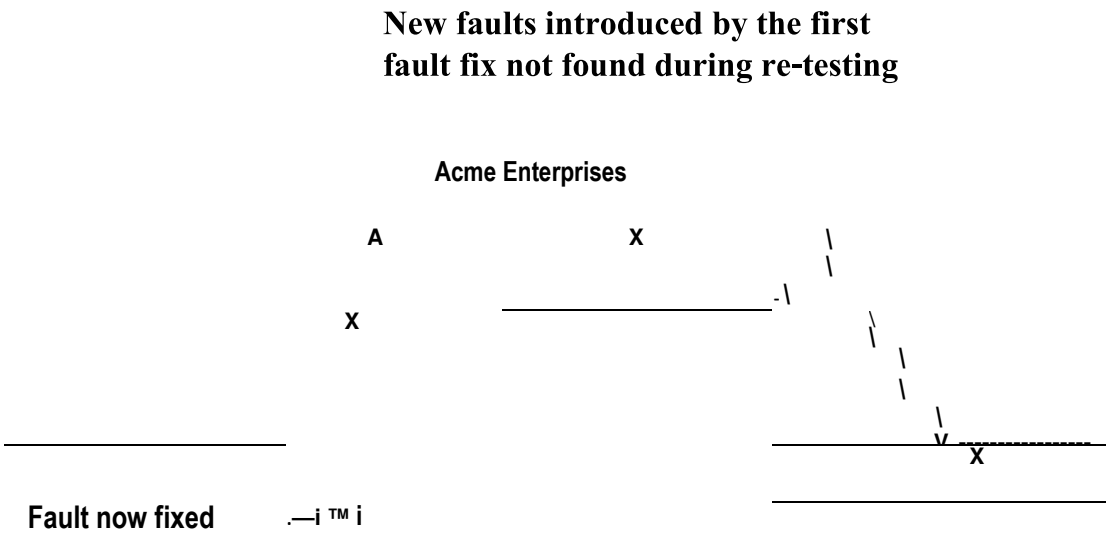
The human tester can immediately investigate anomalies that occur, preserving the context in which they occur. They can also recognise and pass over a wide range of false alarms that would confuse automation.

Re-Testing

Re-running tests usually follows this pattern: -

- Run a test, it fails, fault reported
- New version of software with fault "fixed"
- Re-run the same test (i.e. re-test)
  - Must be exactly repeatable
  - Same environment, versions (except for the software which has been intentionally changed!)
  - Same inputs and preconditions

  If test now passes, it is assumed that fault has been fixed correctly or has it?



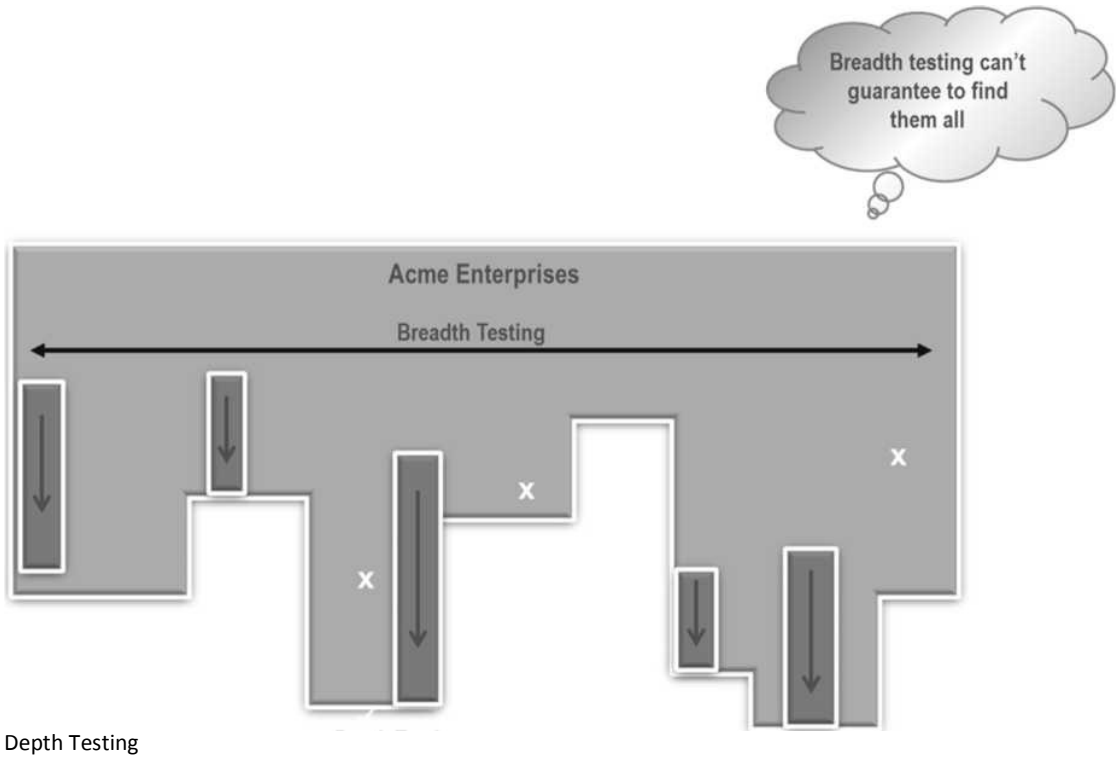
Maintenance

Software maintenance is defined as:-

- The "modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a modified environment".

Implementing software packages has become very popular in to-days modern IT world. It is important to understand that there are differences when testing these applications.

- Breadth tests need to be carried out to establish overall confidence
- Depth tests are carried out to investigate changes and critical areas
- Predominantly regression type testing



Depth Testing

What should be tested in maintenance testing?

- Test any new or changed code
- Impact analysis is crucial
  - What could this change have an impact on?
  - How important is a fault in the impacted area?
  - Test not only what has been affected, but also test how much
  - Most important affected areas
  - Areas most likely to be affected
  - Whole system

The answer: "it depends". What should you do if there are poor or missing specifications?

- Consider what the system should do
  - Talk with users
- Document your assumptions
  - Ensure other people have the opportunity to review them
- Improve the current situation
  - Document what you do know and find out
- Track cost of working with poor specifications
  - To make business case for better specifications

Alternatively, establish 'What should the system do?'

- the way the system works now must be right (except for the specific change) - use existing system as the baseline for regression tests
- look in user manuals or guides (if they exist)
- ask the experts - the current users

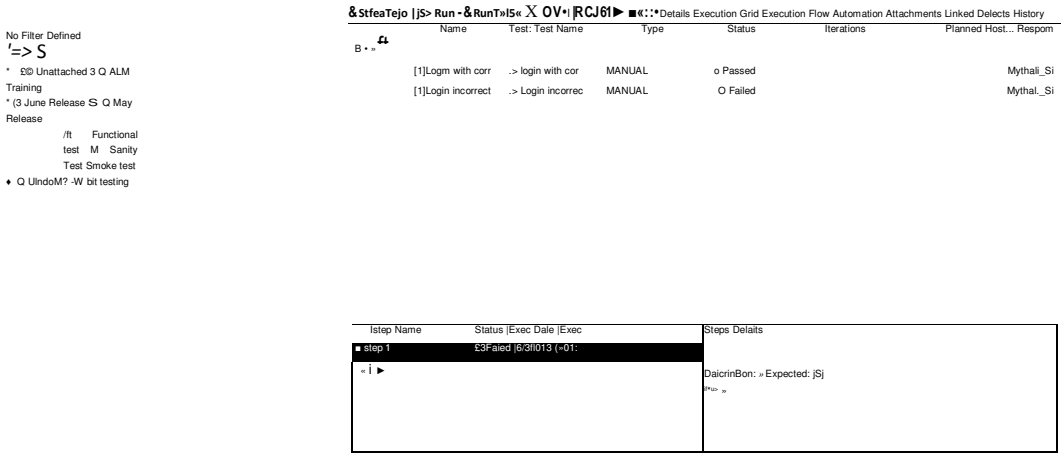
Without a specification, you cannot really test, only explore. You can validate, but not verify.

1.2.3 Develop a Test Execution Schedule **Test Planning Tasks**

Test planning includes the following major tasks:

- Implementing the test policy and/or strategy that is applicable
- Determining the scope, risk and objectives of testing
- Determining the test approach
- Integrating the test activities into the SDLC
- Deciding what to test and which role will undertake each task
- Determine the availability of all resources
- Schedule all test activities
- Define the test documentation requirements
- Select the metrics for monitoring and control of the test activities

The following is an example of a test execution schedule:



) www.SoftwareTestingHelp.com

1.2.4 Identify and Explain Preparation and Execution Activities **Test**

Control Activities

Test control can be defined as the test management tasks that is required throughout the test process in order to keep the testing aligned with the software development process, the needs of the project, and the needs of the organisation

- Compare actual progress against the plan
- Report the status, including deviations from the plan
- Take corrective or pre-emptive action in order to meet the objectives
- Control involves monitoring throughout the project
- Progress
- Coverage
- Exit criteria
- Make decision - Who, what, when, where, etc.

1.2.5 Typical Factors that may Influence the Test Effort

When test plans are created and the testing effort and schedule are estimated, the factors discussed below must be kept in mind otherwise plans and estimates can be misleading at the beginning of the project and can betray you at the middle or end.

The test strategies or approaches that are picked will have a huge influence on the testing effort and it is important to look at factors related to the product, the process and the results of testing.

In Product factors it is important that there is sufficient project documentation so that the testers be clear on what the system is, how it is supposed to work and what correct behaviour looks like. This will assist them in doing the job more efficiently.

The factors which affect the test effort are:

- Good project documentation is a positive factor, but having to produce detailed documentation, such as meticulously specified test cases, can cause delays. During test execution, it takes a lot of effort to maintain such detailed documentation, as well as working with fragile test data that must be maintained or restored often during testing.
- When the size of the product is increased, the size of the project and the project team is also increased. Increases in the project and project team makes it more difficult to predict and manage them. This leads to the disproportionate rate of collapse of large projects.
- The life cycle itself is an influential process factor, as the V-model is usually more fragile in the face of late change while incremental models usually have high regression testing costs.
- Another factor is process maturity, including test process maturity, especially the implication that mature processes involve that change in the middle and end of the project must be managed carefully. This reduces test execution

cost.

- Time pressure is another important factor. Pressure should not be an excuse to take unwarranted risks. However, it is a reason to make careful, considered decisions and to plan and re-plan carefully and intelligently throughout the process.
- Because people execute the process, people factors are as important or more important than any other. Important people factors include the skills of the individual team members as well as the team as a whole, and how those skills are aligned with the project's needs. While there are many troubling things about a project, an excellent team can often make good things happen on the project and in testing.
- A project team is a team, therefore solid relationships, reliable execution of agreed-upon commitments and responsibilities and a determination to work together towards a common goal are important. This is especially important for testing, where so much of what is tested, used, and produced either comes from, relies upon or goes to people outside the testing group. The stability of the project team is an important people factor because of the importance of trusting relationships and the lengthy learning curves involved in software and system engineering.
- The test results themselves are important in the total amount of test effort during test execution. The delivery of good-quality software at the start of test execution and quick, solid defect fixes during test execution prevents delays in the test execution process. Once a defect is identified, it should not have to go through multiple cycles of fix/retest/re-open, at least not if the initial estimate is going to be held to.

1.2.6 Test Entry and Exit Criteria

Different Test Types

There are a number of different test types that form part of SQA: Acceptance Ad-hoc

Black Box Testing White Box Testing

Configuration/Compatibility

Conformance Exploratory testing

Functional Integration Load testing

Maintenance Performance

Recovery Regression Security Sanity

testing Stress testing System Unit

Usability

Volume

Test Types, which can be similar, should not be confused with Test Cycle, some of which are:

- Unit testing
- Integration testing
- Acceptance testing
- Alpha testing
- Beta testing

Testing to verify a product meets customer specified requirements. A customer usually does this type of testing on a product that is developed externally. This kind of testing is known by a variety of names, e.g. User Acceptance, Customer Acceptance, Formal Acceptance and Final Acceptance to name a few) are described with an example of each.

Ad Hoc Testing

Similar to exploratory testing, but often taken to mean that the testers have signific ant understanding of the; software before testing it.

Black Box Testing or Functional Testing

The application is treated like a Black Box where the contents of the box (the code) cannot be seen and is not taken into consideration during testing. The test effort in this regard is focused on passing various inputs through the black box and verifying their outputs.

the box (the code) can be

White Box Testing or Context Driven Testing

In this type of testing the application is treated like a white box where the contents of 1 seen and is taken into consideration during testing. The test effort in this regard is focused on the optimisation and the quality of the code. Examples of white box testing include unit testing and performance testing.

Compatibility Testing

Testing to ensure compatibility of an application or Web site with different browsers, e.g. Internet Explorer, Mozilla Firefox, different Operating Systems together with their various versions and hardware platforms. Compatibility testing can be performed manually or can be driven by an automated functional or regression test suite.

This type of testing is also known as Configuration testing.

Verifying implementation conformance to industry standards.

Producing tests for the behaviour of an implementation to be sure it provides the portability, inter-operability, and/or compatibility a standard defines.

Exploratory Testing

Creative, informal software test that is not based on format test plans or test cases; testers may be learning the software as they test it

Functional Testing

Validating than an application or Web site conforms to its specifications and correctly performs all its required functions. This

entails a series of tests which perform a feature by feature validation of behaviour, using a wide range of normal and erroneous input data.

It can involve testing of the product's user interface, APIs, database management, security, installation, networking, etc. Testing can be performed on an automated or manual basis using black box or white box methodologies

Integration Testing

Testing in which modules are combined and tested as a group. Modules are typically code modules, individual applications, client and server applications on a network, etc. Integration Testing follows unit testing and precedes system testing.

This type of testing is done to determine the capability of the system to handle the anticipated number of concurrent users without much degradation in system performance.

Maintenance Testing

This type of testing should concentrate on the areas that, even though they did not change, they are linked to the changed area via a common program or screen i.e. the impact of the change.

This testing must form part of the testing scope of a project. Maintenance testing can be more difficult than development testing:

- There are no specifications
- Documentation is out of date
- Lack of regression test scripts
- Knowledge base is limited due to age of the system

Performance Testing

This type of testing is performed to determine how a system performs in terms of time responses and stability under varied workloads. There are various types of performance tests like load, spike, soak or capacity or volume.

Performance testing generally involves an automated test suite as this allows easy simulation of a variety of normal, peak, and exceptional load conditions. Special expertise is required in the tool and in performance testing techniques to design tests and interpret results

Recovery Testing

This type of testing measures how well a system recovers from crashes.

Regression Testing

This type of testing allows a consistent, repeatable validation of each new release of the product. It focuses on the areas that

did not change and looks for any unexpected side effects. It test all the functions in an application.
This testing is usually carried out in an independent phase in the test plan and should be the final test before deployment to production. At least 95% 'positive' testing'.

Regression test cases are good candidates for automation, since they are repeated often

Security Testing

This testing is concerned with two different kinds:

- Application perspective (passwords, encryption and access to certain materials)
- Environment level which deals with firewall, user authentications, etc.

Once again, special expertise is required in the tool and in security testing techniques to design tests and interpret results, especially in the environment level

Sanity Testing

Sanity testing is a cursory test to confirm if particular software produces the desired results or not. This implies that the software has already passed other kinds of testing before they can undergo sanity testing. It is important to note that sanity test is not as in-depth as other kinds of testing.

Sanity testing can be categorised under regression testing because the procedures for both these tests are basically the same. In reality, sanity test is a superficial type of test as it just gives a cursory glance on the software. It must be noted that this type of testing has its limitations as well therefore sanity testing must be done along with other testing methods to make the software better and smarter

Soak Testing

Soak Testing is a type of performance test that verifies a system's stability and performance characteristics over an extended period of time. It is typical in the type of performance test to maintain a certain level of user concurrency for an extended period of time.

Spike Testing

Spike testing is a type of load test. The object of this type of performance test is to verify a system's stability during bursts of concurrent user and or system activity to varying degrees of load over varying time periods. It should also verify that an application recovers between periods of spike activity.

Testing conducted to evaluate a system or component at or beyond the limits of its specified requirements to determine the load under which it fails and how. A graceful degradation under load leading to non-catastrophic failure is the desired result.

Often Stress Testing is performed using the same process as Load Testing but employing a very high level of simulated load

System Testing

Testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

Unit Testing

Functional and reliability testing in development environment. Producing tests for the behaviour of components of a product to ensure their correct behaviour prior to system integration

Usability Testing

This testing has to do with 'user friendliness' of the software. Special attention should be paid to things like error messages, feedback ('wait' messages), readability, help functions (precise, constructive, available), navigation, error recovery and support for human memory (short-term memory load should be manageable)

Volume Testing

Volume testing is done against the efficiency of the application. Huge amounts of data are processed through the application (which is being tested) in order to check the extreme limitations.

This type of testing purposely subjects a system (both hardware and software) to a series of tests where the volume of data being processed is the subject of the test. It also seek to verify the physical and logical limits to a system's capacity to ascertain whether such limits are acceptable to meet the projected capacity of the organisation's business processing
Volume testing is sometimes referred to as Capacity Testing

Entry Criteria

The entry criteria of each phase is dependent on the exit criteria of the previous phase. It will depend on the business case, FRS, testable requirements, test plan, test scenarios, test preparation, functional testing that was produced, as discussed in Life Cycles in Study Units 5.1 - 5.3.

Exit Criteria

Purpose

- Define when to stop testing (at a particular test level)

Example

- Thoroughness measures

- Coverage of code, functionality or risk
- Estimates of defect density or reliability measures
- Cost or schedules
- Effort spent, time elapsed
- Residual risks
- Defects not fixed, low coverage in some areas

This will also be discussed in more detail in Life Cycles in Study Units 5.1 - 5.3.

Study Unit 1.3: Monitor and Control Test Progress

US 386054

Manage the software testing process

Specific Outcome 1

Monitor and control test progress

Assessment Criteria

1. Common metrics are described and compared for use for monitoring test preparation and execution.

   ASSESSMENT CRITERION RANGE

   Common metrics may include but are not limited percentage of work done in test case preparation, percentage of work done in test environment preparation, test case execution, defect information, test coverage of requirements, subjective confidence of testers in the product, dates of test milestones and testing costs.

2. The purpose and content of the test summary report are explained according to test requirements.
3. The testing process is controlled according to the test plan.
4. Monitoring and control mechanisms are applied in terms of meeting the requirements of the test plan.
5. An explanation is given of how configuration management supports testing.

1.3.1 Describe and Compare Common Metrics

Common metrics include all of the following:

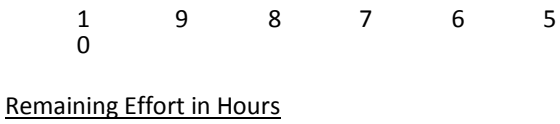
- Percentages/ comparison
  - Percentage of work done in test case preparation
  - Percentage of work done in test environment preparation
  - Test cases written vs. plan
  - Test case execution (run/not run, passed/failed
  - Preparation of environment, test cases etc.
  - Actual against milestone dates
  - Costs against budget
- Defect information
  - Found/fixed, failure rate, density, re-test
- Coverage of code or application
- Coverage of requirements
- Subjective confidence of testers in the product
- Confidence reporting in application
- Dates of test milestones
- Testing costs

When you have to compare the common metrics, a Sprint Backlog spreadsheet and a BurnDown chart are handy tools to use. Below is an example of a Sprint Backlog spreadsheet, and a BurnDown chart that indicates the differences in common metrix between expected delivery and actual delivery.

Sprint Backlog

Team:	8		
Days:	10	Sprint Start Date:	2006/09/25
Hours:	45	Sprint End Date:	2006/10/06
Max Hrs:	358		
		Working Days Left:	

Ge	P	P #	mini Item	Task	Status	1	2	2006/09/25	2006/09/26	2006/09/27	2006/09/28	2006/09/29	2006/10/02	20 10
								344	317	248	226	176	176	1
1		Complete XXX Page (continued)											16	
2		licensing (design only)			Complete	CJ	CC	40	37	32	16	16		
3		Field visibility (tech design)			In Progress	SB	CJ	20	20	20	20	20	20	2


10
9
8
7
6
5
Remaining Effort in Hours

	Remove side panel & implement tabs		Complete	GB	AM	24	20								
	Lookups (dropdowns, lookups, datepicker)														
		Classification Code implementation	Complete	SB	GB	32	27	23	8						
		Lookup control build	Complete	SB	GB				32	24	24	8			
		UI Testing	In Progress	CA									8	6	
		Documentation	In Progress	GB									8	8	
	Web partification		Complete	GB		8	5	8							
	AAA Use Cases														
		ABC Management	Complete	RG	AT	4	4	4	4	4	4	4	4		
		Personal Information Management	Complete	RG	AT	4	4	4	4	4	4	4	4		
		Address Management	Complete	RG	AT	36	36	32	24	16	16	12	6		
		DEF Management (scenario 1)	Complete	RG	AT	8	8	5	5	4	4	4	4		

		Individual Hours			
Colours	Not Started	CJ	60	57	52	36	36	36	28	21	4
		SB	52	47	43	60	44	44	28	20	4
	In Progress	CC	40	37	32	16	16	16	8	1	

Complete

Estimate to completion

d

Testing

Task
s

Summary of Item

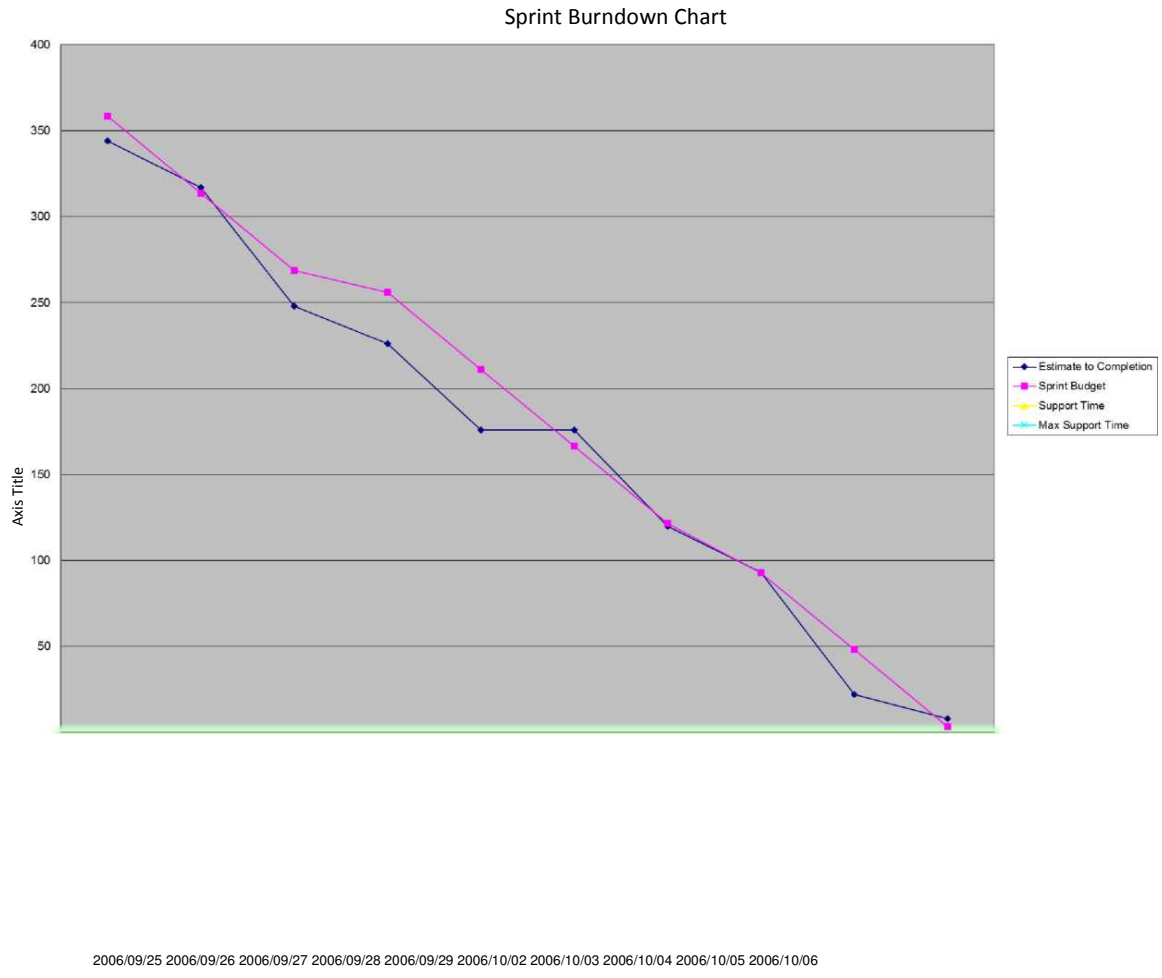
Single Line Item Support Item

Day
Sprint
Budget
Adjustments
Support
Used
Support
Available

Sum of Estimate to completion

HWH>OT3<H>ODDO^>

	1	2	3	4	5	6	7	8	9	10
	24	20								
	64	52	31	40	24	24	8	8	8	2
								8	6	4
	52	52	45	37	28	28	24	18		
	52	52	45	37	28	28	24	18		
	358	314	269	256	211	166	122	93	48	3
				32				16		
	344	317	248	226	176	176	120	93	22	8


Sprint Burndown Chart
400
350
300
250
200
150
100
50
Axis Title
Estimate to Completion
Sprint Budget
Support Time
Max Support Time
2006/09/25 2006/09/26 2006/09/27 2006/09/28 2006/09/29 2006/10/02 2006/10/03 2006/10/04 2006/10/05 2006/10/06

1.3.2 The Purpose and Content of the Test Summary Report

Test reporting:

Provide a summary of the status of testing at any given time

What happened during testing?

Were dates and exit criteria met?

Analyse metrics and make appropriate recommendations Is it worth carrying on with testing?

What's the situation with risks?

Are we all confident the application will work

There are two types of reports

- Phase specific
- Interim

Typical reports required on a project are:

- Test Strategy/Plan for the Project
- Daily Test Status
- Test Closure
- Project Close-out
- Post Implementation

Contents of a Test Report

A test report should include a summary which describes the evaluation of what was tested with detailed identification (version, sub-version and all applicable numbers), as well as references to the different forms of test documentation, like the test plan, test design document, test case and procedures, defect or anomaly reports and test logs, where available.

It is vital to identify and describe the variations that were identified between the components that should have been tested, and those that were actually tested. It is also important to identify and include any differences and variations that were identified with regards to the test documentation, and what the reasons are for these differences of variation. Furthermore, there should be a comprehensive assessment that highlights the level of coverage of the actions that were planned in the test plan, and here it is important to identify all incidents and defects that were resolved, what was done to resolve these defects or incidents, and any anomalies that are still unresolved.

The report should also include a global evaluation of each component that was tested, and an identification of any limitations identified based on the test results and the pass/fail criterial. It should also include a summary of the main test activities and all notable events, and should include a synthesis of the usage of resource.

Defect Reporting

Introduction

In the SQA LIFECYCLE™, the Defect Management process is shown as starting from Analysis and Design up to and including Support and Maintenance. During Analysis and Design phase, defects in the Functional Specification should be logged and managed in the same way as during testing. This is to ensure the quality of the Functional Specification.

Defects in test requirements and test cases during the test analysis and test design phases as well as the implementation phases should also be managed in a structured and efficient way. Your main task is to find defects and log them. Learn how to log your defects well, so that developers act on them.

Purpose of a defect report

Defect reports list defects but that is not all. They are also put to use in the following manner:

- Provide feedback on a problem to assist in identification, isolation and correction
- Allow the test leader to track the progress of the application under test
- Allow the test leader to track the quality of the application under test
- Business Analysts use them to update Functional documentation
- Systems Analysts use them to update Technical documentation/design
- Defect Reports provide background information for writers who are developing training material, troubleshooting sections as part of the Help or on the company's website
- Defect Reports provide key information for after-sales support (known defects)
- Defect Reports provide starting-point suggestions for improvement at the start of the next release of the product
- Management uses Defect Reports to assess the status and quality of the project

Warning:

- Your defect reports are your primary work output and will form the audience's impression of you.
- Your audience is not only the developers but include all levels of management.
- Every person you criticize will see the report.

If you are not sure how the report will be read, have a colleague/the Test Manager read it and listen carefully to their comments.

Recording Defects

Defect Reports are one of the critical deliverables from a tester. Find out about the organizations standards and procedures for reporting defects (Defect Management Policy). They are viewed by the development team, business analysts, project managers, technical managers and the quality assurance engineers along with the testers. Hence, the defect reports must carry enough information about the problem, e.g. Steps to reproduce the defect, the test you were

running, the step you were running, the data you were using, expected result and actual result.

Defects also carry other vital information, such as Severity, Priority, Assigned To, Status and Test Environment details, without which, the defect report is considered as incomplete

What to do:

- Log defects immediately. Do not wait, you may have forgotten the detail by tomorrow
- Log all defects (minor ones, non-producible ones etc.) The management team will decide whether the minor defects will be fixed, and when
- Pay close attention to the structure of a logged defect (a remarkable number of defects are fixed late in the project, when people are exhausted.) Number all steps
- Use white space
- Use short, simple sentences
- Indicate what was supposed to have happened, not only what happened
- Include additional comments if they will make it easier for the developer to recognize the problem
- When the developer reports the defect as fixed, test it yourself before you accept it as fixed
- Make sure the summary line of the defect makes absolute sense. This is the reader's introduction to the defect you are reporting and should state clearly what the defect is about.
- Keep tracks of 'deferred' defects
- Pick your battles

What not to do

Do not do the following on your defect reports (note that these are STS guidelines):

- Use sms-type language (e.g. gr8, u, u2, c'ya, et cetera) - use full sentences and grammatically correct language to create reports which everyone can understand (even 'old stick in the mud' developers and managers)
- Use CAPS when typing - it comes down to screaming at the developers
- Use humour to liven up the reports or because you like it or because you think it will soften the blow of the defect. Humour will be misunderstood
- Do not edit other peoples' logged defects, add comments to them instead
- Offer a solution for a defect; you can't think for other people, and they will not appreciate it. Do not make it easy to reject your report
- Exaggerate the facts

There is still one more facet of defect reporting that is very important: What is the difference between Severity and Priority?

- Severity refers to the impact of the defect. How severe is it?
- Priority indicates when your company wants it fixed.

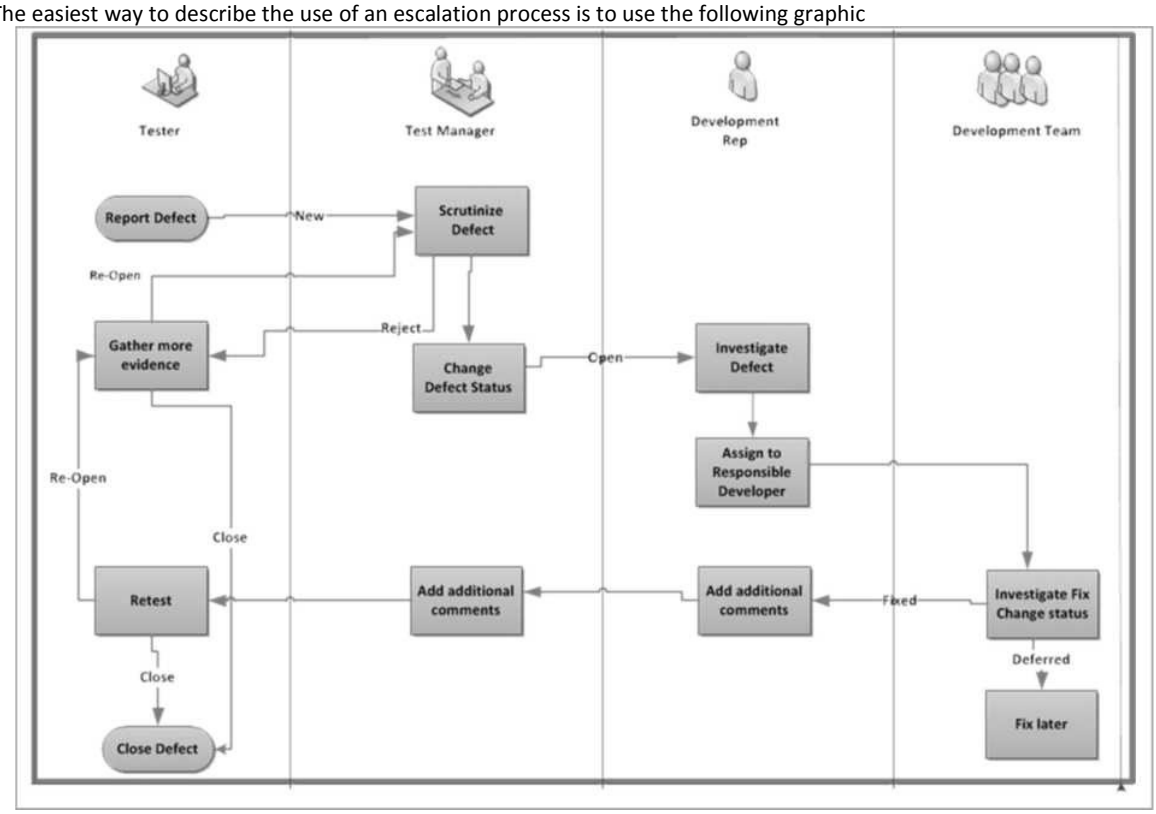
Severe problems may not always be worth fixing. A high priority defect is not necessarily a severe one (e.g. product name is misspelt)

Defect Escalation Process

The objective of testing software is to identify defects. The quality of the software is no better if defects are not fixed. Defect Escalation is the process to efficiently manage defects to ensure that they are resolved with minimal impact to the system, thereby increasing the quality of the system under test.

How is it used?

The easiest way to describe the use of an escalation process is to use the following graphic



The illustration, although somewhat simplified, is merely an example of what to expect. As the defect passed through the process each person to whom the defect will be assigned will be responsible for updating certain fields.

Take note of what you have been taught in the section on defect reporting. Description layouts as well as individual fields will be different in every Defect Management System. Make sure that you know and understand the DMS at your organization

Defect Categories

All defects are broadly categorized into the following areas

- Errors of commission: something wrong is done
- Errors of omission: something left out by accident
- Errors of clarity and ambiguity: different interpretations
- Errors of speed and capacity

As this type of categorization is extremely broad, it does not contribute to the quality process in its present format. There are many listed categories (+/- 18) and what follows are the ones most commonly found when performing functional testing.

Defect Categories are context dependent so each organization will define them to suit their type of business

Commonly used categories

- Functionality
  - If a program doesn't do something that it's supposed to do, or perhaps does not complete the process, this is known as a functional defect
- Usability
  - Usability has to do with the user-friendliness of a system. (how easy /difficult is the software to use and does it look nice
- Requirements
  - We know through experience, that anything that causes user dissatisfaction is regarded as a defect
  - In most cases, the product is the business solution to a problem
  - If the problem hasn't been resolved, the product does not work
- The number of defect that can be directly linked to bad requirements can be as high as 46%Test Cases
  - Mistakes often appear in test cases due to a number of reasons
  - It is important that these kinds of errors are reported so that they can match the Application functionality at all times
- Configuration
  - When a function behaves differently when running on different environments (Windows XP vs. Windows

Specialised Categories

- Performance
  - When program responses do not meet the specified response times there are categorized as performance issues
- Load
  - The main focus of load testing is a performance test which subjects the target-of-test to varying workloads to measure and evaluate the performance behaviours
  - It will also measure the ability of the target-of-test to continue to function properly under these different workloads
- Stress
  - The main focus of this type of test is to subject the target-of-test to large amounts of data to determine which limits are reached that cause the software to fail
  - It will also identify the continuous maximum load or stress the target-of-test can handle for a given period of time.
- Security
  - This testing is usually done by experts. Password testing is a typical example

1.3.3 Control of the Testing Process according to the Test Plan

Measuring Test Execution Progress

Use the Agile principle of a daily Burn-down Chart:

- It's a simple yet powerful tool to help track the progress of the project
- Imagine lighting a candle at the start of testing which will burn out when the allocated time on the project reaches 0 hours.
- During the testing phase the only part of the candle that is visible is the amount of candle that still has to burn (the estimated time to completion)
- The idea behind the burn-down chart is to simply plot progress the project visually on a graph
- The original estimates are show on one line showing the testing budget moving towards 0 hours
- The Estimate To Completion (E.T.C.) are plotted on a different line
- When the actual line drops below the original estimation line this indicates that the project is running ahead of time
- When the actual line climbs above the original line this indicates that the project is behind schedule and corrective action is necessary
- Using E.T.C instead of completed work has the advantage of highlighting the impact of scope creep on the project as soon as it is known
- Each individual is responsible to update their E.T.C. on the functions they are working on every morning
- The team goal is quite simply to be able to deliver working software when the project reaches h 0 hours

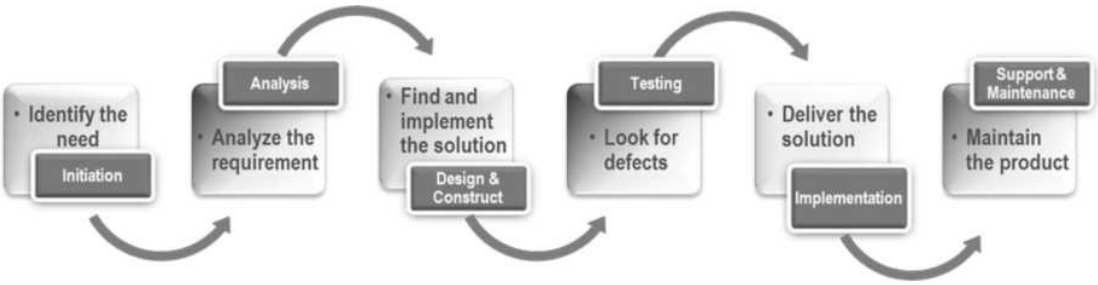
1.3.4 Apply Monitoring and Control Mechanisms

Reasons for Test Progress Monitoring

- To provide feedback and visibility of testing activities
- Show how we are doing against the plan
  - What metrics do you keep and why
- Adequacy of test objectives for this test level
- Adequacy of test approaches taken
- Effectiveness of the testing with respect to its objectives

Support and Maintenance

This is the process of monitoring a systems' performance once released into production. Many organisations embark on Contact/Call Centres which handle client queries. These queries range from training issues through to more severe issues where actual functionality may not be working in the production environment. Support also initiates changes such as enhancements and bug fixes. These initiatives follow the SDLC from the beginning, and so the cycle of development continues.



1.3.5 How Configuration Management Supports Testing

In large companies there will be a configuration laboratory that will support the different hardware versions.

In these laboratories non-functional tests will be done. That means that the function has already been proved to be working, but now the function is tested on different hardware configurations to ensure that it is working on all the different configurations possible.

Configuration management determines clearly about the items that make up the software or system. These items include source code, test scripts, third-party software, hardware, data and both development and test documentation.

Configuration management is also about making sure that these items are managed carefully, thoroughly and attentively during the entire project and product life cycle.

In configuration management there are a number of important implications for testing, for example configuration management allows the testers to manage their testware and test results using the same configuration management mechanisms.

Configuration management also supports the build process, which is important for the delivery of a test release into the test environment. It will not be sufficient to simply send Zip archives by e-mail, because there are too many opportunities for such archives to become polluted with undesirable contents or to harbour left-over previous versions of items. Especially in later phases of testing, it is critical to have a solid, reliable way of delivering test items that work and are the proper version.

Last but not least, configuration management allows us to keep the record of what is being tested to the underlying files and components out of which it is made up. This is very important. An example is when defects are reported, they must be reported against something, something which is version controlled. If it is not clear what we found the defect in, the programmers will have a very tough time of finding the defect in order to fix it. For the kind of test reports discussed earlier to have any meaning, it is important to be able to trace the test results back to what exactly was tested.

Ideally, when testers receive an organised, version-controlled test release from a change-managed source code repository, it is along with a test item transmittal report or release notes. [IEEE 829] provides a useful guideline for what goes into such a report. Release notes are not always so formal and do not always contain all the information shown.

IEEE 829 STANDARD: TEST ITEM TRANSMITTAL REPORT TEMPLATE

Transmittal report identifier
Transmitted items
Location
Status
Approvals

Configuration management is very complex, therefore advanced planning is very important to make this work. During the project planning stage - and perhaps as part of the test plan - it is important to include configuration management procedures and tools. As the project proceeds, the configuration process and mechanisms must be implemented, and the key interfaces to the rest of the development process should be documented.

Study Unit 1.4: Apply Risk and Testing Processes

US 386054

Manage the software testing process Specific

Outcome 1

Apply risk and testing processes.

Assessment Criteria

1. Risks as possible problems are described in accordance with project objectives.
2. The level of risk is determined in terms of their likelihood and impact on the testing process.
3. Potential solutions to risks are identified to serve as a back-up plan.

1.4.1 Risks as Possible Problems

Risk management is a critical activity in the planning and tracking of software testing. It includes the following processes:

- Identification
- Prioritisation/analysis
- Treatment of risks faced by the business

Risk management is performed at different levels, namely project level, program level, organisation level, industry level and even national or international level.

Risks arise from many different perspectives like project failure, safety, security, legal liabilities and non-compliances with regulations. It is important to understand that risks are potential problems that have not yet occurred. A problem that has already occurred is an issue and is treated in a different way in software test planning.

Risk Identification

Risks are identified within the scope of the project. Risks can be identified by using a number of different resources like project objectives, risk lists of past projects, prior system knowledge, understanding of system usage, understanding of system architecture/ design, prior customer bug reports/ complaints, project stakeholders and industry practices. For example, if specific areas of the system are unstable and the development of those areas are continued in the current project, it should be listed as a risk.

Good risk management is to document the identified risks in detail so that it remains in the project memory and can be clearly communicated to project stakeholders. Usually risk identification is an iterative process. It is important to re-visit the risk list whenever there is a change in the project objectives or new business scenarios are identified. As the project proceeds, some new risks appear and some old risks disappear.

1.4.2 Determine the Level of Risk in terms of their Likelihood and Impact on the Testing Process.

It is much easier to prioritise a risk if the risk is understood accurately. The understand risk, two measures, Risk Impact and Risk Probability, are applied to each risk. Risk Impact is estimated in tangible terms (e.g. rand value) or on a scale (e.g. 10 to 1 or High to Low). Risk Probability is estimated somewhere between 0 (no probability of occurrence) and 1 (certain to occur) or on a scale (10 to 1 or High to Low). For each risk, the product of Risk Impact and Risk Probability gives the Risk Magnitude. If the Risk Magnitude is sorted in descending order, it provides a list in which the risks at the top are the more serious risks and need to be managed closely.

When all the Risk Magnitudes is added, it gives an overall Risk Index of the project. If the same Risk Prioritisation scale is used for all projects, it is possible to identify the riskier projects by comparing the Risk Magnitudes

1.4.3 Potential Solutions to Risks to Serve as a Back-Up Plan

Each risk in the risk list should receive the following Risk Treatments.

a. Risk Avoidance: For example, if there is a risk related to a new component, it is possible to postpone this component to a later release. Risk Avoidance is seldom done because it has an impact on the project objectives e.g. delivery of new features.

b. Risk Transfer: For example, if the risk is that the security testing of the system is insufficient, it may be possible to hire specialists to do the security testing. Risk Transfer takes place when this vendor is held accountable for sufficient security testing of the system. Risk Transfer increases the project cost.

c. Risk Mitigation: This is a common risk treatment. The objective of Risk Mitigation is to reduce the Risk Impact or Risk Probability or both. For example, if the testing team is new and does not have prior system knowledge, a risk mitigation treatment may involve a knowledgeable team member join the team to train others while the process is ongoing. Risk Mitigation also increases the project cost.

d. Risk Acceptance: If a risk is not treated by any prior treatments, it has to be accepted. This happens when there is no viable mitigation available because of reasons such as cost. For example, if the test environment has only one server, risk acceptance means that another server is not built. If the existing server crashes, there will be down-time and it will be a real issue in the project.

A few other points are:

1. Risk management brings clarity and focus to the team and other stakeholders. Though the team should be careful not to burn more time on risk management if it is not providing more value.
2. The risk list should be a live working document, consisting of current risks, how they are prioritised and their treatment plans. The test approach and test plan should be adopted according to the risk list whenever the latter is updated.
3. Bigger projects usually need more stakeholders and the bigger the project, the more formal the risk management process should be.

Study Unit 1.5: Report on Software Testing

US 386054

Manage the software testing process

Specific Outcome 1

Report on software testing.

Assessment Criteria

1. Types of reports are explained with examples of when each is used.
2. The content of a report is described according to organisational requirements. ASSESSMENT CRITERION RANGE
   Types may include but are not limited to light reporting and comprehensive reporting.
3. Issues for improving the testing process are explained in terms of the test process.
   ASSESSMENT CRITERION RANGE
   Issues for improving the testing process may include but are not limited to data, measures, metrics, knowledge and improvement loop.
4. An incident report is written covering the observation of the testing.

1.5.1 Types of Reports

Reporting is often referred to as either light or comprehensive reporting.

Light reporting: Daily reporting on what happened the previous day

Medium reporting: Weekly reporting that combines everything that happened on a daily basis, and what is still outstanding by the end of the week.

Comprehensive reporting: This happens at project closeout, will include what went well, what did not go well, if timelines and costs were met, etc.

There are two types of reports

- Phase specific
- Interim

Typical reports required on a project

- Test Strategy/Plan for the Project
- Daily Test Status
- Test Closure
- Project Close-out
- Post Implementation

1.5.2 The Content of a Report

The report will usually consist of the following two sections, with the information needed in each section indicated:

Provide a summary of the status of testing at any given time

- What happened during testing?
- Were dates and exit criteria met?

Analyse metrics and make appropriate recommendations

- Is it worth carrying on with testing?
- What's the situation with risks?
- Are we all confident the application will work?

1.5.3 Issues for Improving the Testing Process

The following should be included to address the issues and what should be done to improve the testing process:

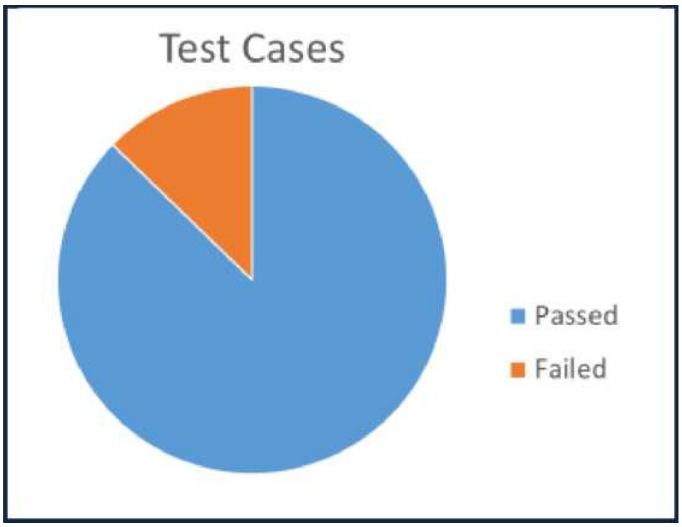
Metrics

Metrics will help to understand the results of the test execution, the status of test cases and defects etc. It is also possible to add Required Metrics if necessary. Charts or graphs can be attached for better visual representation, and to add more clarity.

For example:

a) No. of test cases planned vs executed
b) No. of test cases passed/failed

Test cases planned	Test cases executed	Test cases pass	Test cases failed
60	55	48	7



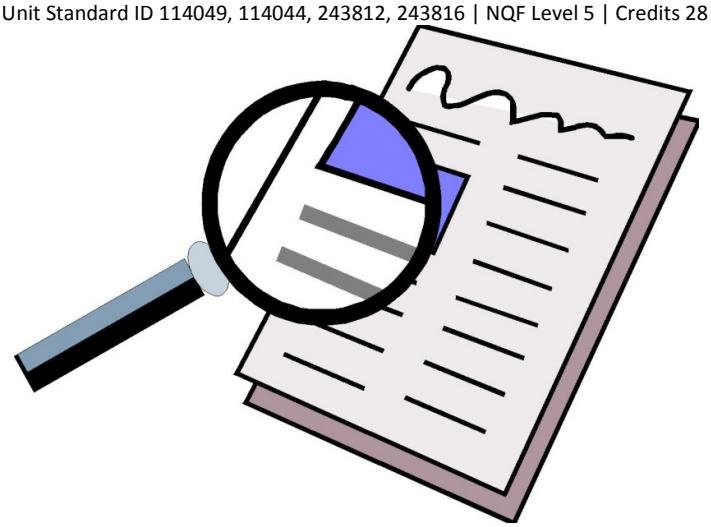
```
ERROR: stackunderflow
OFFENDING COMMAND: ~

STACK:
```

Qualification 83026: National Certificate: Information and Communications Technology (ICT) Software Testing

Test Manager

Learner Guide

Unit Standard ID 114049, 114044, 243812, 243816 | NQF Level 5 | Credits 28



Contents

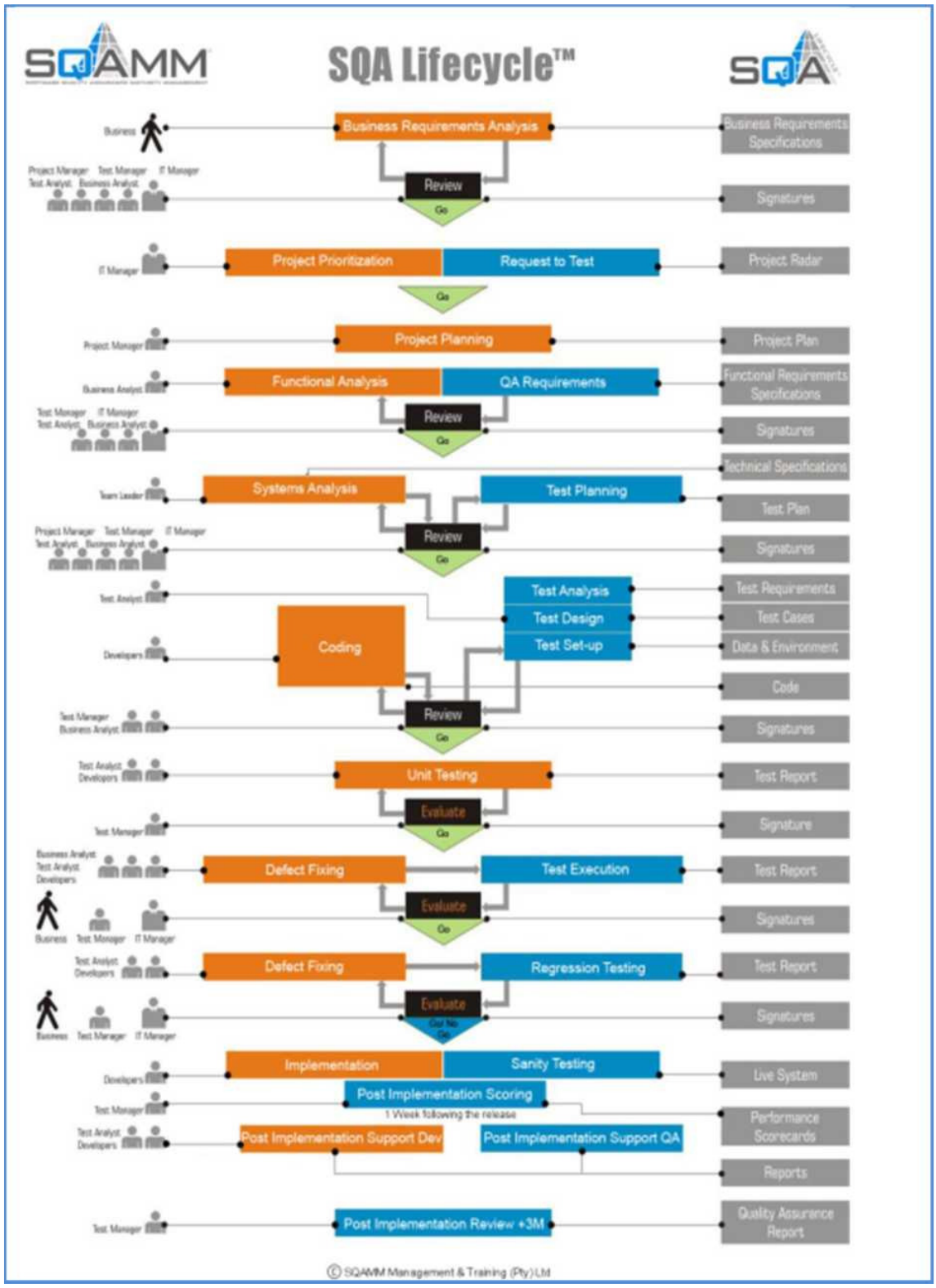
Overview

Glossary and Terms

Term	Definitions
Assessment	The process of collecting evidence of learners' work to measure and make judgements about the achievement or non-achievement of specified National Qualifications Framework (NQF) standards or qualifications.
Assessment Criteria	The assessment criteria for a course are the dimensions with which you will judge how well a learner has achieved the learning goals.
Assessor	A person who is registered by the relevant Education Training Quality Assurance (ETQA) body to measure the achievement of specified National Qualifications Framework standards or qualifications.
Credit	That value assigned by the authority to ten (10) notional hours of learning.
Critical Cross-Field Outcomes (CCFO)	Critical Cross-Field Outcomes refer to those generic outcomes that inform all teaching and learning. For example, CCFOs may include working effectively with others as a member of a team, and/or collecting, analysing, organising and critically evaluating information.
Essential Embedded Knowledge (EEK)	Essential embedded knowledge is explicit and resides within systematic routines. It relates to the relationships between roles, technologies, formal procedures and emergent routines within a complex system.
Formative Assessment	Refers to assessment that takes place during the process of learning and teaching. May also be integrative in nature.
Integrated Assessment	An assessment, which permits the learner to demonstrate applied competence and which uses a range of assessment methods.
Summative Assessment	It is the assessment for making a judgement about achievement. This is carried out when a learner is ready to be assessed at the end of a programme of learning.
Moderation	The process which ensures that assessment of the outcomes described in the NQF standards and qualifications is fair, reliable, and valid.
Outcome	Contextually demonstrated end product of the learning process.

Term	Definitions
Recognition for Prior Learning (RPL)	The comparison of the previous learning and experience of a learner, howsoever obtained, against the learning outcomes required for a specified qualification, and the acceptance for purposes of qualification of that which meets the requirements.
SETA	A body responsible for the organisation of education and training programmes within a specific sector.
Specific Outcome	Knowledge, skills and values (demonstrated in context) which support one or more critical outcomes.
Unit Standard	Registered statement of desired education and training outcomes and its associated assessment criteria together with administrative and other information as specified in the regulations.

SQA Lifecycle



About this Course

SQA Management & Training -

What is SQA?

SQA is an acronym for Software Quality Assurance Maturity Management. It is an independent research and analysis company that specializes in Software Quality Assurance.

The SQA framework is made up of:

- 3 certifications
- 7 products

SQA™ is an "all-in-one" package that delivers end-to-end QA across the development lifecycle ensuring high quality output, reduced costs of rework and an optimized delivery cycle.

In addition, it has tried and tested solutions which enable organisations to validate 3rd party implementations further reducing time to market and costly on-going maintenance.

SQA™ is supported by two key frameworks geared to ensuring that testing services are of high quality and that there is a constant drive to improve testing services and increase SQA maturity levels towards a world class standard. These two quality initiatives are:

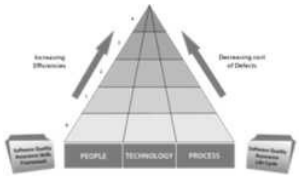
- SQA Health-check™
- SQA Toolkit™

SQA™ and its supporting frameworks, SQA Health-check™ and SQA Toolkit™, are all based on industry best practices that have been extensive research and are an adaptation of the following standards:

- IEEE 829: 1998 Standard for Software Test Documentation
- IEEE 1044-1: 1993 Standard Classification for Software Anomalies
- IEEE 1061 Standard for software quality metrics and methodology

SQA - SQA Maturity Model™

The SQA Maturity Model is the foundation on which SQA Maturity Management is based.



It is a benchmark against which you can measure the efficiencies of your PEOPLE, PROCESS & TECHNOLOGY

Skills development and performance management framework for testing professionals. Process framework for the application of quality assurance and testing to the Product / Systems Development Life Cycle

SQA™ Skills Framework

The SQA skills framework deals with the People side of the SQA Maturity Model™ and provides the delegate with skills required to perform the core competencies of a Professional Tester

Key differentiating factors:

- The SQA Vision is to become the SKILLS PROVIDER OF CHOICE by gaining the reputation within the IT Industry of providing skilled resources that are head and shoulders above everyone else
- Expert Trainers based on a combination of +/- 40 years of testing experience, expert knowledge and subject matter experts.
- All our training is presented by our training partner STS Skills & Training Academy
- Our training material is outcome based, and we encourage all our delegates to become solution driven.
- Our training covers all the skills, on the core competencies of a Professional Tester, that are required for international certification

SQA - Certification

A SQA certification is awarded on the competency of individual to execute quality control within a development lifecycle.

it is divided into three parts:

- SQA Certified Test Analyst
- SQA Certified Testing Professional
- SQA Certified Test Management Professional

The Road to SQA Tester Certification

The SQA™ certification process for the Testing Professional has been divided into 2 levels.

The first level, SQA™ Certified Test Analyst contains 4 core modules, Tester, Requirements Management, Test Analysis and Test Design.

Level 2, SQA™ Certified Testing Professional will require the trainee to complete the remaining 3 modules dealing with Testing in Agile, Functional Automation and Performance Testing.

All trainees are at liberty to apply for a credit on any one of the modules if they feel that they have reached the required competency levels. Apply for a credit will require the trainee to take the course assessment without attending the course.

In keeping with international certification standards, the delegate is expected to attain a minimum mark of 65% on all module assessments. In the event of a delegate failing to reach the required certification mark, delegates will be allowed one "rewrite" of the course assessment at a later date, at no extra cost.
Additional "re-writes" will be available at the ruling price.

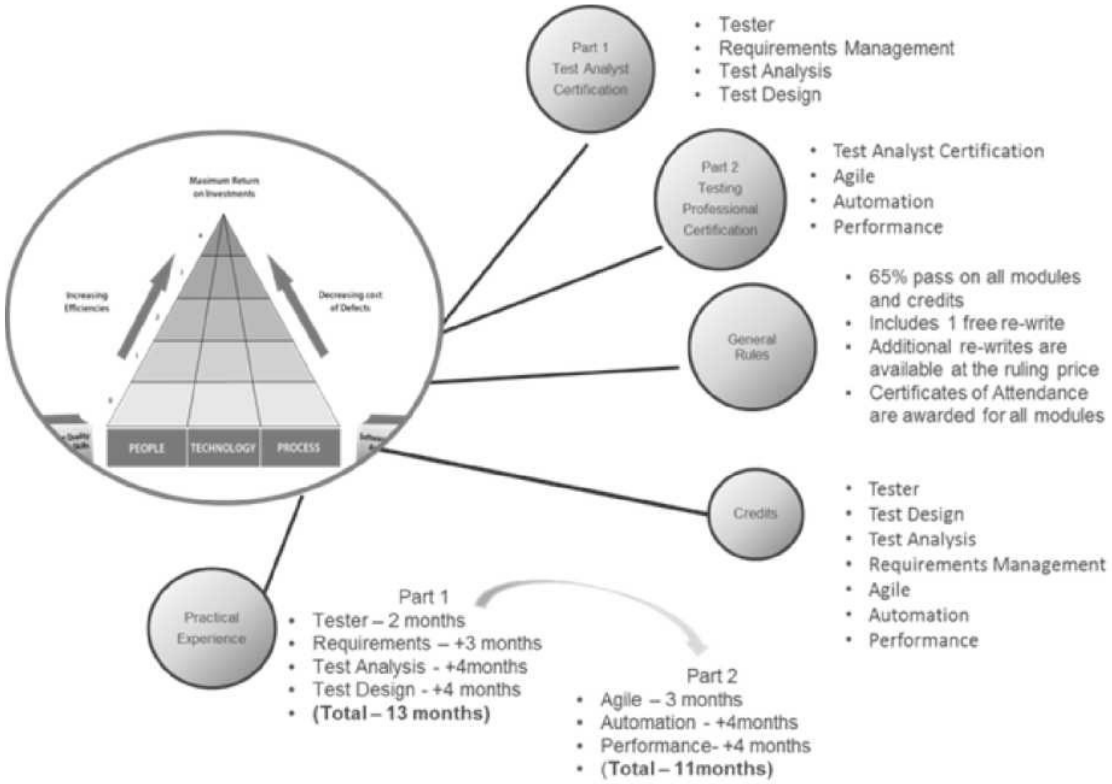
All delegates will be awarded a 'SQA™ Certificate of Attendance' for each course attended irrespective of whether the required mark is reached.

Finally, the delegate will be required to gain the prescribed practical experience in all modules as follows:

Role		Duration
Tester		2 months
Requirements Management	+	3 months
Test Analysis	+	4 months
Test Design	+	4 months
Test Analyst Certification	+/-	13 months
Agile	+	3 months
Automation	+	4 months
Performance	+	4 months
Total		24 months

When the delegate has passed all the required modules, for Levels 1 or 2, they will be required to write a final exam covering all the modules for that level. Once again the pass mark required is 65% to be awarded the SQA™ Certification

SQA™ certification requires a real commitment from both the Trainee and the Organisation. It will also assist with retention of skills.



Skills Programme Framework **Learning Programme Details**

The learning material for this programme has been designed and developed according to:

Unit Standard (US) Title	US NO	NQF Level	Credits
Demonstrate an understanding of Computer Database Management Systems	114049	5	7
Demonstrate an understanding of change management for computer systems	114044	6	3
Monitor and control the execution of the project management plan for a simple to moderately complex project	243812	5	12
Develop a project quality management plan for a simple to moderately complex project	243816	5	6

These unit standards also form part of certain qualifications and skills programmes.

Purpose of the Training Programme

The course prepares a person for a position as Test Analyst in any company that follows a structured approach to software development and testing. It also ensures that the delegate gains an understanding of the key concepts in testing

Furthermore, the purpose is to ensure that the Learner gains the knowledge and skills to :

- Demonstrate an understanding of Computer Database Management Systems
- Demonstrate an understanding of change management for computer systems
- Monitor and control the execution of the project management plan for a simple to moderately complex project
- Develop a project quality management plan for a simple to moderately complex project

During the Skills programme, you will be guided through an interactive learning process that will provide you with the necessary skills to prove your competence against the following Specific Outcomes:

Unit Standard	Specific Outcomes
114049	Describe data management issues and how it is addressed by a DBMS. Describe commonly implemented features of commercial database management systems. Describe different type of DBMSs. Review DBMS end-user tools.
114044	Describe a change control policy for an organisation. Produce guidelines for the implementation of changes to computer systems.

Unit Standard	Specific Outcomes
243812	Monitor and control the performance of the project. Manage project baselines through integrated change control. Manage and resolve issues and new risks. Manage the development of the products of the project.
243816	Demonstrate an understanding of project quality management principles. Gather and analyse quality requirements for a project. Compile the project quality management plan. Obtain approval for the project quality management plan

Credit Value

The credit weighting is 28 credits; this adds up to 280 notional hours, these hours are made up of 88 hours theoretical, 96 hours practical and 96 hours' work place experience

After you completed the learning experience you will be required to complete a Portfolio of Evidence (PoE) based on the specific outcomes and assessment criteria as indicated in this learner guide.

A registered assessor will assess your portfolios and measure your outcomes. If you are declared competent, you will receive the credits and a statement of results from the relevant SETA. If you are declared not yet competent, you will have the opportunity to address the gaps that are outlined in the assessor report.

Notional hours include time spent in instruction, individual learning as well as structured learning and assessment.

- A eleven (11) day contact session (including theory and practical application.
- Workplace Experience
- Assessment Preparation
- Workplace Assessment
- Final Summative Assessment

Target Audience

This qualification is aimed at those people who wish to conduct a practical investigation to provide stakeholders with information about the quality of the software product or service under test.

Previous disadvantaged persons who have been assessed without the required qualification will be able to be assessed against this standard through RPL assessment to gain National recognition for their competence. This will enhance their employability and career advancement and contribute to the socio-economic transformation of the country.

Assessment

In order to be declared competent against the unit standard comprising the Module, you will need to complete both formative and summative assessment activities.

Formative assessment refers to assessment that takes place during the process of learning and this will be the completion of

the activities in the Portfolio of Evidence. Your facilitator will assess these activities and give continuous feedback on your progress. You will be required to do the necessary remediation.

Summative assessment is assessment for making a judgement about achievement. This includes a Knowledge Questionnaire on the completion of the unit standard in the module. Where you need to be re-assessed, the following conditions will apply:

- Specific feedback will be given so that learners can concentrate on only those areas in which you were deemed not yet competent.
- Re-assessment will take place in the same situation or context and under the same conditions as the original assessment.
- Only the specific outcomes that were not achieved will be re-assessed.

Always work in black ink and remediate in blue ink. Never work in pencil. Ensure that you include all formative and summative assessments in your Portfolio of Evidence (PoE). Ensure that you include all formative and summative assessments in your Portfolio of Evidence (PoE)

The quality of evidence will be measured in terms of the VARCS principles.

	Term	Definition
V	Valid	It should relate to what is being assessed.
A	Authentic	You must generate your own evidence at the workplace and may not use the evidence/work of other people in any form.
R	Reliable	Reliability relates to consistency.
C	Cu rrent	It should be as recent as possible.
S	Sufficient	There should be enough evidence, but not too much.

Delivery Mechanism and Learner Roles and Responsibilities

The programme will be presented to you as a workshop. During the workshop you have the opportunity to learn in groups by completing various activities and taking part in group discussions as well as individual learning by doing activities on your own.

The principles of adult learning apply. This means that you are expected to accept responsibility for your own learning. Show up in class and be prepared to learn. Obey rules to ensure that a safe and effective learning environment is created. Show the necessary respect for property and equipment as well as your facilitator and fellow learners. You show respect by treating others, as you would like to be treated.

Group Discussion	This icon indicates when a group discussion should take place.
Activity	This icon indicates an activity that takes place in class.

Introduction: The Role and Responsibilities of a Test Manager

Not Unit Standard Based - Additional Client Requested Information	
n/a	
Specific Outcome	
	n/a
Assessment Criteria	
	n/a

Introduction 1: The Role of the Test Manager

Why do we need managers?

- Co-ordination of People, Processes and Technology.
- Governs QA processes, to achieve project deliverables.
    - Buffer between project stakeholders and test team. Test Analysts & Testers can focus on deliverables.
- Single point of contact & control.
- The Test Manager is the Quality Gatekeeper on a project!!!

Core Focus Areas

Test Team Management

- Leading and directing
- Identify resource requirements (competency levels and training)
- Manage team performance
- Delegation of duties

Define and manage the test plan

- Identify the critical path
- Identify strategic milestones
- Monitor project progress
- Take corrective action where necessary

Strategic Improvements

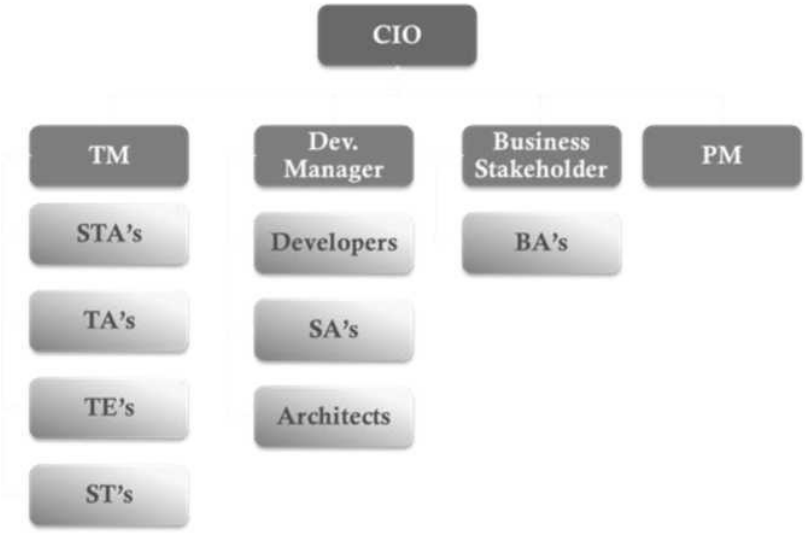
- Keep in touch with industry trends
- Make changes to processes when necessary
- Make use of available testing tools

Reporting and Managing Stakeholders

- This is a key component in the role of Test Manager

- Produce phase specific or interim reports when necessary
- Ensure reports are of high standard and are easy to read
- Identify and meet stakeholders' expectations

The Test Manager position in the Team



Attributes of a Good Test Manager

- Thorough understanding and belief in the value that Quality Assurance adds to the SDLC
- Team player and people person.
- A firm and charismatic leader.
- Delivery centric and focused on success.
- Diplomatic, has a good temperament with excellent negotiation skills.
- Exceptional effective communication skills
- Be a lateral thinker
- Embody the values of the organisation (e.g. STS values EXPERTS.QUALITY. DELIVERED)

Roles and Responsibilities

Roles and Responsibilities of the Test Manager are:

- Define the TEST STRATEGY for the project or department. This includes operational as well as strategic objectives.
- Estimate TESTING EFFORT by analysing the requirements of a project. Define and obtain management support for the time, resources and budget required to perform the testing on a project.
- Develop the TEST PLAN including scope, strategy, timelines, roles, team structure, risks, technical requirements, environmental requirements and budget. Obtain stakeholder support for this plan.
- Build a TESTING TEAM of professionals with appropriate and varied skills, experience and attitude. Organise the hierarchy and reporting lines of the test team.
- PERFORMANCE MANAGEMENT of the test team including reviews and development plans. Regular informal performance management should take place at least once a week
- Take part in motivating the team, ensuring EMPLOYEE WELLNESS. Staff retention is the responsibility of the Test

Manager.

- Organize and manage the testing MEETINGS required during the test lifecycle.
- Arrange the hardware and software requirements for the setup of lab pcs, TEST TOOLS and TEST ENVIRONMENTS.
- ASSIGN TASKS to all testing team members and ensure that all team members have an appropriate workload on the project.
- MANAGE THE DELIVERY of all tasks defined in the Test Plan. Utilise and leverage off other senior resources within the team.
- Ensure content and structure of all testing documents and artefacts are DOCUMENTED CORRECTLY and maintained by the team
- Document, implement, monitor, and enforce all PROCESSES for testing as per standards defined by the organisation. SQAMM™ PROCESSES should be followed as far as possible and customised if required.
- Identify TRAINING requirements and assist HR in driving training plan (Technical and Soft skills).
- Attend all project meetings, REPRESENTING TESTING within those meetings.
- REVIEW samples of all team member's test cases, reports, defects and testing documents.
- Keep track of the new REQUIREMENTS and changes in scope of the Project
- Manages PROJECT REQUIREMENTS (Software, Hardware, Resources) with the Project Manager.
- Organize test status meetings and send STATUS REPORTS (Daily/Weekly) to all stakeholders.
- Act as the single POINT OF CONTACT between Development and Testers.
- Coordinate and supervise the DEFECT MANAGEMENT process. Utilise and leverage off other senior resources within the team.

- Design, prepare and conduct PRESENTATIONS which are well structured and informative.
- All documents and outputs from the SQA LIFECYCLE must be produced e.g. test plans, test execution reports, defect reports, sign off, post implementation.
- COMMUNICATE EFFECTIVELY at all levels, elicit and probe for information by asking the right questions.
- LISTEN openly to ideas and needs from team members. Provide constructive feedback.
- Work both INDEPENDENTLY as well as WITHIN THE TEAM.
- Resolve potential DISPUTE/CONFLICT situations with tact and diplomacy.
- Show assertiveness and exude CONFIDENCE at all levels- from test team to project stakeholders.
- Manage handover to sales and support teams. Ensure they understand the product well and provide good after sale service
- Resolve potential DISPUTE/CONFLICT situations with tact and diplomacy.
- Show assertiveness and exude CONFIDENCE at all levels- from test team to project stakeholders.
- Manage handover to sales and support teams. Ensure they understand the product well and provide good after sale service.
- Perform and manage the POST IMPLEMENTATION review process.
- Live by your organisations values.
- If you are an independent resource ensure that you represent your company values within the client environment

Building Relationships

- Build relationships with everyone and at every level.
- A well-liked test manager will get more buy in than someone disliked.
- No one is unimportant. The CEO to the tea lady should know and respect you.
- LISTEN openly to ideas and needs from team members.

  Provide constructive feedback.

  - Work both INDEPENDENTLY as well as WITHIN THE TEAM.
  - Resolve potential DISPUTE/CONFLICT situations with tact and diplomacy.
  - Show assertiveness and exude CONFIDENCE at all levels- from test team to project stakeholders.
  - Manage handover to sales and support teams. Ensure they understand the product well and provide good after sale service

Communication Fundamentals

- Show genuine interest
  - Conversation links (Name, Home, Family, Work, Travel or Hobby)
- Focus on the other person
  - To do this stop, look, listen and focus
- Be a strong conversationalist
  - Ask - Who, What, When, Why and How?

Meetings

- Schedule meetings in advance and try to stick to 15-30 min meetings.
- Invite only the required decision makers. Stakeholders can be invited as optional.
- Each meeting should have an agenda distributed with the invite.
- As the coordinator of the meeting, ensure people stick to the points at hand and reign them in if required.
- Take longer discussions offline for further discussion afterwards.
- Each discussion item needs to be concluded with an action to be performed, an assignee and a due date.
- Take minutes and send out after the meeting. The person who takes the minutes of the meeting controls the actions afterwards.
- Cater important meetings or those during lunch with coffee/muffins/snacks if possible.
- There's a fine line between having valuable group sessions vs. too many and too long meetings. Stick to the SQA Certification ™ communication plan. (to be discussed in detail later)

Introduction 2: Managing Change

Change is Constant

Change is Constant:

- It's a cliche, but change has always been the only constant.
- In recent times, the pace of change has accelerated greatly, and we all need to find ways to deal creatively with this fact of modern life.
- Everyone is resistant to change YET it's the only thing that has brought progress
- Anyone who wants to be successful - in career, in relationships, in life - must learn how to see and manage change the way that successful 'change leaders' do
- Test Managers are not exempt from this and must be adaptable, creative and responsive to change
- Change is also inherently ambiguous, and those who deal creatively with change will have a high tolerance for uncertainty and 'shades of grey.'
- Skillful managers of change see themselves as inherently powerful and having the ability to control elements of the situation in which they find themselves
- Some circumstances cannot be changed, but the way we respond to them is always a choice
- We will always have a sphere of influence, however small.

- By focusing on this sphere of influence, and not expending energy bemoaning the area outside it, the circle will start to expand and give us progressively more control.
- Some of the biggest challenges faced by test managers today is making effective process changes within an organisation.
- There is the belief that 'we have always done things this way and it works for us so why should be change'
- Most organisations do note keep up with modern best practices and the way they are used to working may no longer be relevant.
- Solutions to problems always exist, and the 'internal self' should reflect the desire to find them together with the certainty that they can be successfully implemented.
- There are many ways of dealing with change and some people are better at it than others
- Solutions to problems always exist, and the 'internal self' should reflect the desire to find them together with the certainty that they can be successfully implemented

Seven Key Factors for Managing Change

There are many ways of dealing with change and some people are better at it than others

i. Spend time reflecting on your own values

A sense of purpose is essential to success and effectiveness, and those without a clear idea of what they are doing and why they are doing it will not have the foundation to keep going in the face of change.

ii. Be persistent

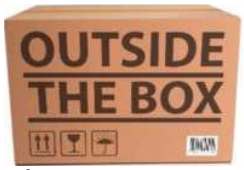
Success is usually more to do with tenacity that genius. Persistence is only possible when you have clarified your values and when you are able to build on the bedrock of purpose. Successful people keep going in the face of change, finding new and creative ways to achieve a positive outcome.

iii. Be flexible and creative

Persistence does not mean pushing through by force. If you are unable to achieve success one way, try another, and then another. Keep looking for more creative solutions and innovative responses to problems.

iv. Think outside the box

Read widely, and don't confine yourself to your own area of 'expertise.' Try to see links between apparently separate and diverse elements in your life and experience.



v. Accept uncertainty and be optimistic

Life is inherently uncertain, so don't waste your energy trying to predict the future. Of all the possible outcomes, focus on the most positive one. Accept that if you respond well and work to the best of your ability, a good outcome is as likely as any other. Don't waste your energy being negative

vi. Keep fit and healthy

Eat well, get enough sleep, exercise regularly. Meditation can help, too. This will keep up your energy levels and allow you to keep going in tough times. Not taking care of yourself physically, mentally and spiritually is foolish and short sighted.

vii. See the big picture

Change is inevitable, but if you take a bird's-eye-view of the landscape, the change won't be so disorientating, and you will keep perspective at all times.

Change Management and Control will be discussed in more detail in Study Unit 2.1 and 2.2

Introduction 3: Leadership

Only three things can happen in an office without leadership:

i. Friction

ii. Confusion

iii. Underperformance

Everything else requires leadership

Leadership is the art of motivating a group of people to act towards achieving a common goal. The process of encouraging and helping others to work enthusiastically towards objectives.

Leadership is ultimately about creating a way for people to contribute to making something extraordinary happen.

Leadership and Management

There are three mistakes commonly made by people regarding leadership and management

- Mistake No 1 - People use the terms "Leadership" and "Management" inter-changeably. This shows that they don't see the crucial difference between the two and the vital functions that each role plays
- Mistake No 2 - They use the team 'leadership" to refer to the people at the top of the hierarchy. They refer to the people in the middle layers as "management" and the rest of the people as specialists or individual contributors
- Mistake No 3 - People often think of leadership in terms of personality characteristics usually as something call 'charisma'. Since few people have great charisma this leads to the conclusion that few people can provide leadership which gets us into increasing trouble.

Importance of Leadership

Leadership transforms potential into reality. It is not mere using people and their potential for realising an organisation's goals. It has the ultimate aim of raising the level of human conduct and ethical aspiration of both the leader and the led.

The leader should elevate, inspire, and evangelise his followers to higher things in life. Leadership and power are closely linked. People tend to follow those who are powerful and because others follow, the person with power leads.

There are basically two forms of power:

Negative: When a leader has the power to award bonuses, give raises or has the ability to fire staff and then uses this power to get the job done

Positive: Deals with people who are experts in their fields or when they enjoy the admiration of their team

Formal and Informal leadership

Formal Leadership

- Occurs when a manager leads by exercising formal authority.
- The exercise of formal authority through assigning duties derives, from the

manager's official position within the organisation's hierarchy of authority.

Any employee who is assigned a managerial position has the opportunity and responsibility to exercise formal leadership

Informal Leadership

- Arises when a person without formal authority is influential in directing the behaviours of others.
- Although not formally appointed or elected he becomes a leader through his actions or personal attractions. Ultimate aim of raising the level of human conduct and ethical aspiration of both the leader and the led.

Core Leadership Theories

Trait Theories

- Effective leaders share a number of common personality characteristics.
- These theories are helpful to identify traits and qualities like integrity, empathy, assertiveness, good decision-making skills and likeability

Behavioural Theories

- This kind of theory focuses on what people do
- Developed in the 1930's by Kurt Lewin

There are three major styles of leadership

- Autocratic or authoritarian
- Democratic or Participative
- Laissez-Faire or Free Reign

Contingency Theory

This theory focuses on the situational influence. They try to predict which style is best in which circumstance Popular based contingency-based models include

- Hersey-Blanchard Situational Leadership Theory
- House's Path-Goal Theory
- Fiedler's Contingency Model

Power and Influence Theory

Based on the different ways that leaders use their power and influence to get things done. The best known theories in this group are:

- French and Ravens' 5 forms of power
- Transactional Leadership (reward's influence)
- Lead by example

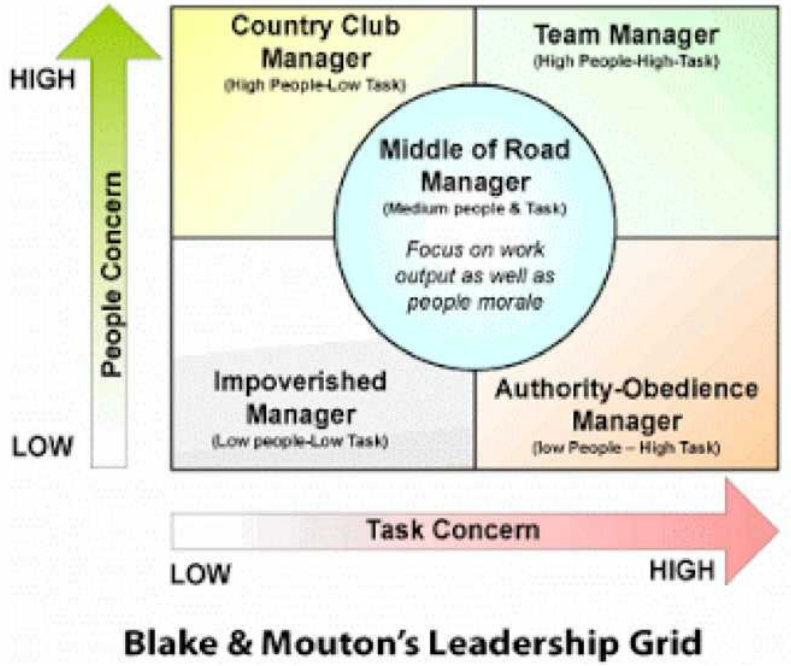
Transformational Leadership

- Effective Leadership is not simply based on a set of attributes, behaviours and influences
- A wide range of abilities and approaches are needed
- This is why this kind of leadership is the most accurate.
- This kind of leader shows integrity and they know how to develop a robust and inspiring vision of the future
- They are able to motivate people to achieve this vision, manage its delivery and are able to build ever stronger and more successful teams

Blake and Mouton's Leadership Grid

This leadership grid is a two-dimensional grid based on

- Concern for people - This is the degree to which the leader considers the needs of team members, their interests and areas of personal development when deciding how best to accomplish a task
- Concern for task - This is the degree to which the leader emphasized concrete objectives, organisational efficiency and high productivity when deciding how best to accomplish a task



http://neilinbahrain.blogspot.co.za/2015/09/mouton-and-blakes-leadership-grid.html

The Grid identifies five different styles of leadership:

Impoverished Leadership:

i. Low on Results and low on People
ii. Very ineffective leadership style
iii. This kind of leader has neither a high regard for creating processes for getting the job done, nor for creating a work environment that is both satisfying and motivating.
iv. Very ineffective leadership style
v. The result is disorganisation, dissatisfaction and disharmony.

Country Club Leadership:

i. Low on Results and High on People

ii. This style of leader is most concerned about the needs and feelings of members of his/her team.

iii. These leaders operate under the assumption that as long as team members are happy and secure then they will work hard.

iv. This result is a work environment that is very relaxed and full of fun so production suffers due to lack of direction and control.

Produce or Perish Leadership (Authority-Obedience):

i. High on Results and Low on People

ii. This style of leader believes that employees are simply a means to an end.

iii. Employee needs are always secondary to the need for efficient and productive workplaces.

iv. This type of leader is very autocratic, has strict work rules, policies, and procedures, and views punishment as the most effective means to motivate employees.

Middle of the Road Leadership:

i. Medium on Results and Medium on People

ii. This style of leadership seems to be a balance of the two competing concerns and may at first appear to be an ideal compromise.

iii. BUT .... Therein lies the problem.

iv. When you compromise, you could be giving away a bit of each concern, so neither production nor people needs are fully met.

v. Leaders who use this style settle for average performance and often believe that this is the most anyone can expect.

Team Leadership:

i. High on Results and High on People

ii. This style of leader believes that employees and production needs are equally high.

iii. Although ideal, it is not that practical as leaders are faced with all kinds of situations, on a daily basis, that can and do sabotage their best intentions

Conclusion

There are many different theories of Leadership. There are also many different styles of Leadership. We have only just scratched the surface when it comes to these Leadership Theories and Leadership Styles. As this is a Test Management Course and not a Leadership Course, it is not possible for us to delve any deeper into the subject.

The most effective leader is one that can master a minimum of four different styles and can make use of the different styles to suit different situations when the need arises.

Introduction 4: Level 5 Leadership

Good to Great <u>What is Good to Great?</u>

Is a book written by Jim Collins and is based on the results of a six year research project in which 18 well known and established companies in the USA were compared to counterpart companies in the same area of business. The research was able to answer the question "What enabled some companies to last so long while others in the same market either struggle to get by or simply fade away after a short period of time?"

Two sets of comparison companies were selected:

- Companies in the same industry with the same opportunities and resources
- Companies that made the short-term shift from good to great, but failed to maintain the trajectory. What made the difference?

A Different Kind of Leadership

Different kinds of leaders are the leaders who focus equally on what to do, what not to do and what to stop doing. They are leaders who are willing to get the right people on the bus and the wrong people off the bus before deciding where the bus is going; leaders who are willing to confront the brutal facts and debate vigorously in search of the right answers yet unify behind decisions irrespective of parochial interest. They are also leaders who see the company as a simple, extremely clear concept of what it is, (the hedgehog concept).

What is the hedgehog concept?

In a famous essay, philosopher Isaiah Berlin described two approaches to thought and life using the following parable:

A fox knows a little about many things but the hedgehog knows only one very big thing very well. The fox is complex, and the hedgehog simple yet the hedgehog wins every time.

Breakthrough from good to great requires a simple hedgehog understanding of seeing the company as three intersecting circles that show:

- What they can make money at
- What they can be passionate about
- What they can be the best in the world at

The following are good leaders:

- Leaders who are willing to build a culture full of well-disciplined people who take disciplined decisions that are fanatically consistent with the three circles.
- Leaders who understand that technology is an accelerator, not an agent of change
- Leaders who avoid technology fads and bandwagons through carefully selected technology
- Leaders who understand to move forward requires a constant steady work rather than quick acceleration (The Flywheel affect)
- Leaders who avoid the vicious circle of rushing first in one direction and then another in the hope of creating a sudden break with the past that will propel them to success (the Doom-loop affect)

Developing Level-5 Leadership

A level-5 leader sits on top of a hierarchy of capabilities and is a necessary requirement to transform an organisation from good to great. But what lies beneath?

There are four other layers, each one appropriate in their own right, but none with the power of level-5. To become a level-5 leader it is not necessary to move sequentially through each level of the hierarchy to reach the top.

BUT ..... to be a fully-fledged level-5 leader you will require all the capabilities of the lower levels plus the special characteristic of level-5.

The Level-5 Hierarchy

1. Highly Capable Individual: Makes productive contributions through talent, knowledge, skill and good work habits
2. Contributing Team Member: Contributes to the achievement of group objectives Works efficiently with others in a group setting
3. Competent Leader: Organizes people and resources towards the effective and efficient pursuit of predetermined objectives
4. Effective Leader: Catalyses commitment to and vigorous pursuit of a clear and compelling vision Stimulates the group to high performance standards
5. Level 5 Executive: Builds enduring greatness through a paradoxical combination of personal humility and professional will

Growing to Level-5

Grow level-5 seeds by practicing by practicing the following disciplines:-

- People first, strategy second. Great vision without great people is irrelevant
- Deal with the brutal facts of your current reality while at the same time maintaining absolute faith that you will prevail
- Keep pushing toward the eventual breakthrough from good to great
- Practice the hedgehog concept
- Use technology as an accelerator and not an agent of change
- Develop and embrace a culture of discipline

Humility and Will = Level-5

Level 5 leaders blend the paradoxical combination of deep personal humility with intense professional will How do

level-5 leaders manifest humility?

They routinely credit other, external factors and good luck for their company's/departments success.
BUT ... when results are poor they blame themselves.

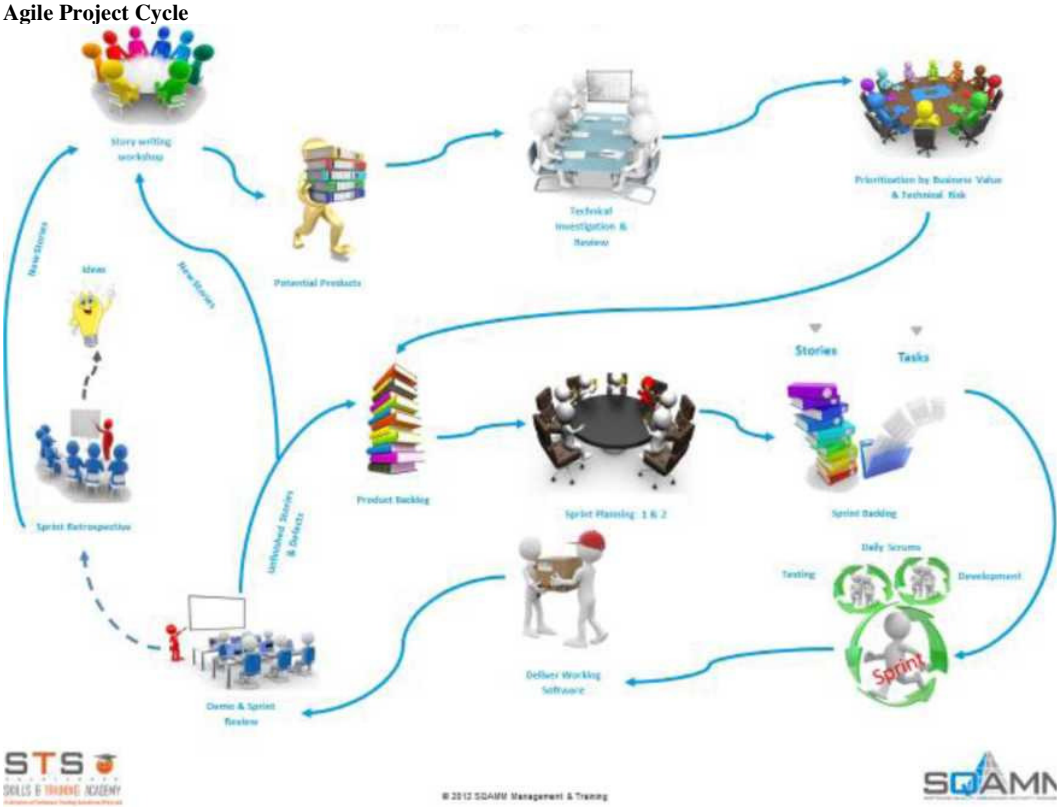
They also act quietly, calmly and determinedly relying on inspired standards rather than charisma to motivate Inspired standards demonstrate the level- 5 leader's unwavering will. The do not tolerate mediocrity and are stoic in their resolve to produce great results

Introduction 5: Developing Standards and Procedures

A summary of level-5 traits

Personal Humility	Professional Will
Is never boastful and demonstrates a compelling modesty and shun adulation	Creates superb results, a clear catalyst in the transition from good to great
Acts with quiet determination and relies in inspired standards and not charisma to motivate	Demonstrates an unwavering resolve to do whatever must be done to produce the best long term results no matter how difficult
Challenge ambition into company and not self. Sets up successors for even more greatness in the next generation	Set the standard of building a great company and will settle for nothing else
Looks in the mirror and not out the window to apportion responsibility for poor results never blaming other people, external factors or bad luck	Looks out the window and not in the mirror to apportion credit for the success of the company - other people, external factors and good luck

Agile Project Cycle



Test Strategy

The test strategy is the approach taken for testing within a particular project, team or department. It is also referred to as the master test plan, test framework, test approach or test policy. A typical testing strategy in any organisation is made up of a number of standards and procedures.

Examples of these standards and procedures include some of the following: -

- Test Approach
- Risk-based Requirements Management
- The Release Process
- The Test Plan
- Estimating Time and Effort
- Managing and Controlling the Test Process
- Defect Management
- Data Management
- Reporting

If you are using a Testing tool you can add another three:

- Project customization
- Housekeeping procedures
- Naming standards

The following is important:

- Appoint an owner of your testing process (Team or Individual)
- Document any exceptions. Reasons for non-compliance must also be documented
- Consistent approach to testing is encouraged
- Regular feedback on the test strategy provides opportunity for keeping strategy up to date and relevant. Source:

ANSI/IEEE Std. 829 "Standards for Software Test Documentation

Release Management

Week-4	Post Implementation	Prod Support.	Prod Support. Post Implementation Review. Incorporate lessons learnt from PIR into Test Strategy.	Post Implementation Meeting.	Post Implementation Report. Revised Test Strategy.
	Request to Test	Development of code	Initial request to test received. Provisional planningand agreement of scope. Evaluate Test Strategy for project suitability.	Project Planning Session.	Test Strategy. Test Scope.
Week-3	Test Planning	Development of code	Test Planning. Resource Allocation. Product Backlog/Scope Finalisation for the Release. Initial BR5 walkthrough	BRSreview with BA. Test Plan Review with Project Manager. Test Plan Review with Test Team. Daily TestTeam Scrum.	Test Plan. Test Schedule. Weekly Test Status Update.
Week-2	Test Analysis	Development of code	Review, QA and Sign Off BR5/FRS. Develop Test Requirements, Scenarios in QC. Test Req uireme nts/Scenarios Peer Review.	Daily Test Team Scrum. BR5 Review and sign off. Test Scenario Peer Review Feedback Session. Test Scenario JAD session with Bas, SMEs. Data Requirements JAD	Te st Req ui reme nts Completed in QC. Data Requirements doc compiled by each team member. Daily Test 5tatus Update.

Week-1	Test Design	Development of code	Develop Test Cases. Test Case Peer Review. Acquire]NT,QAand Prod Data. Create Exploratory Test Set. Sign off Test Cases.	Daily TestTeam Scrum. Test Case Peer Review Feedback Session. Exploratory Test Set JAD. Test Case Sign Off session.	Test Cases Completed in QC. Reference Numbers for each data item added to Data Requirments Doc. Daily Test Design Report from QC.
Week 1	Exploratory testi ng	Defect Management. Envi ronment Management. Unit Testing.	Unit Testing Sign Off. Exploratory Testing in INT. Integration testing.	Daily TestTeam Scrum. Defect Meetings.	Daily Test Ex ecuti o n+Def e ct+5tatus Report. Cycle close out report.
Week 2	Cycle 1	Defect Management. Envi ronment Management. Unit Testing.	Cycle 1 INT. Sign off to QA Envi ronment. Integration testing.	Daily TestTeam Scrum. Defect Meetings.	Daily Test Ex ecuti o n+Def e ct+Status Report. Cycle close out report.
Week 3	Cycle 2	Defect Management. Envi ronment Management. Unit Testing.	Cycle 2 QA. Prod Data Prep. Regression Testing(Manual and	Daily TestTeam Scrum. Defect Meetings.	Daily Test Ex ecuti o n+Def e ct+Status Report. Cycle close out report.
Week4	Cycle 3	Defect Management. Release into Production.	Cycle 3 QA. Regression Testing{Manual and Automated). Business UAT. Sign Off.	Daily TestTeam Scrum. Defect Meetings. Sign Off Meeting. Prod Test Scenario JAD.	Daily Test Ex ecuti o n+Def e ct+Status Report. Test5ign Off Report. Prod Test Scenarios.

Test Planning

What is the purpose of a test plan?

It is the portion of the project plan that details the testing work to be done

What information should be in a test plan?

- Use the IEEE 829 standard as a guideline?
- Helps to ensure that nothing important is omitted?
- It is not a test design specification, a collection of test cases or a set of test procedures.

Whom does it communicate to?

- The plan itself serve as the means of communication with other members of the project team, testers, peers, managers and other stakeholders.

What information should be in a test plan?

- Writing a test plan forces, us to confront the challenges that await us and focus our thinking on important topics.

Factors that Influence Test Planning

i. The company's testing policy
ii. Project constraints
iii. Scope of Testing
iv. Resource availability
v. Project criticality
vi. Project and product risk
vii. Test objectives

All this information should be included in the Master Project / Test Plan, System Test Plan and the Actual Test Plan

Contents of the Test Plan

1. Test plan identifier
   - Some unique reference for this document
2. Introduction
   - Management summary
   - References to other documents (e.g. project plan, configuration management, etc.)
3. Test items
   - Test items; what is to be tested (e.g. software or documents) including version/revision
   - References to software documentation
4. In Scope
   - Identify test design specifications/ techniques

5. Out of Scope

- Reason for exclusion

6. Testing approach (strategy)

Test Methodology Types of Testing Test Prioritization Test Metrics

Test Deliverables

7. Item pass/fail (for what is tested)

- Exit criteria for project or phase

a. Coverage (code or functionality)

b. Residual risk- defects not fixed or lack of coverage

c. Defect density or reliability measure

d. Target date reached

e. Budget spent

8. Suspension criteria and resumption criteria

- For all or parts of testing activities
- Which activities must be repeated on resumption

9. Item pass/fail (for what is tested)

- Exit criteria for project or phase

1. Coverage (code or functionality)

2. Residual risk- defects not fixed or lack of coverage

3. Defect density or reliability measure

4. Target date reached

5. Budget spent

9. Suspension criteria and resumption criteria For all or parts of testing activities Which activities must be repeated on resumption

10. Test deliverables Test plan

Test design specification (Test conditions)

Test case specification (Test Cases)

Test procedure specifications (Steps for executing a set of tests) Test item transmittal reports Test logs (Release Notes)

- Test incident reports
- Test summary reports (What happened)

11. Testing tasks

- Including inter task dependencies and special skills
- Environment
- Physical, hardware, software tools
- Responsibilities

12. To manage, design, prepare, execute, witness, check, resolve issues, providing environment, providing the software to test

13. Staffing and training needs

- This includes training on the Application Under Test, testing techniques and testing tools

14. Schedule

- Test milestones in project schedule
- Item transmittal milestones
- Additional test milestones

- What resources are needed when

15. Risks and Contingencies
    - Contingency plan for each identified risk (Good idea to include assumptions)
16. Approvals
    - Names and when approved

Sprint Schedule

Week 1

	Analysis	Dev	Testing
Mon	JADS Sprint Retrospective (M)	Start Coding C1 Defect Fixes Sprint Retrospective (M)	Test Design (Team) Sprint Retrospective (M)
Tue	Modelling Test Case Review	Coding Cycle 1 Dev QA	Test Design Test Case Review (M)
Wed	Modelling	Coding Cycle 2 Deploy Cycle 1 Dev QA	Test Execution Cycle 1 Defect Management (TL)
Thu	Stories	Coding Cycle 2 Defects Cycle 1 Dev QA	Test Execution Cycle 1 Defect Management (TL)
Fri	Stories Demo (M)	Coding Cycle 2 Dev QA Demo (M)	Test Execution Cycle 1 Defect Management (TL) Demo (M)

The contents of a test plan have been defined in the IEEE 829 standard.

There are rather a lot of items that need to be included in a test plan. To help remember, some fundi in the industry devised acronym, SPACE DIRT that maps all the headings that have to be included in the test

Scope	Test items, features to be, and not to be tested
People	Staffing and training needs, schedule responsibilities
Approach	Approach, Test Strategy
Criteria	Acceptance, Suspension and Resumption criteria
Environment	Environment
Deliverables	Testing deliverables, schedules
Incidentals	Test Plan Identifier, Introduction, Approvals Assumptions and Dependencies
Risks	Risk and Contingencies
Tasks	Testing Tasks

plan

Test Planning Tasks

Test planning includes the following major tasks:

- Implementing the test policy and/or strategy that is applicable
- Determining the scope, risk and objectives of testing
- Determining the test approach
- Integrating the test activities into the SDLC
- Deciding what to test and which role will undertake each task
- Determine the availability of all resources
- Schedule all test activities
- Define the test documentation requirements
- Select the metrics for monitoring and control of the test activities

Risk-based Quality Management

When planning testing activities, we generally do not have unlimited resources at our disposal. This means that we will have to make a choice as to which items are going to need the most attention. e.g. Requirements that have low criticality to the business or those that have only a minor risk associated with their implementation. Using risk-based quality management enables one to calculate at which level to test each requirement, based on the nature of the requirement and the resources that are available. Planning your testing process can be based on these recommendations.

In order to effectively manage requirements using risk of failure (product risk), requirements need to be broken down

into the following sub-policies. All these sub-policies are closely interlinked and none can be eliminated when defining the organisations Risk-based Quality Management policy

i. Requirement Types
ii. Requirement coverage analysis
iii. Requirement Risk
iv. Testing Level
v. Testing Times

Requirement Types

In order to effectively determine the scope of the different functions that make up an application, the recommended practice is to break down each function into a four or 5-tiered hierarchy. Each level of the hierarchy has a different, but dependent, role to play in the requirement management process. The naming standards of the different levels will differ from company to company.

What follows is an example of a typical requirement hierarchy

Application Under Test:

- 1st level of the hierarchy and used to document a brief summary of the application under test

Application Function:

- 2nd level of the hierarchy and used to list the business functions and features of the application

Functional Elements (User Stories)

- 3rd level of the hierarchy and used to list the different functional elements or user stories of the

Application Function. The relationship between Functional Elements and Functions to be Tested could be 'one to one' or 'one to many'

Testable Conditions (Acceptance Criteria):

- Bottom level of the hierarchy and used to list the different 'rules' that govern each Functional Element
- This is probably the most important level as it documents the requirements that are required to be validated by running one or more tests against them
- Once again the relationship between the Testable Condition and Functional Element could be 'one to one' or 'one to many'

Note: All of the previous Requirements Types are mandatory for an organisation using Risk-based Quality Management

- Function Group: This level of the hierarchy is optional and, if in use, will be inserted between the Application Under Test and the Application Function levels. It has been added to the hierarchy to enable an organisation to place the Application Functions into logical groups. If the need arises a Sub-Group can also be included at this level

In practical terms it is important to understand that Application Functions, Functional Elements and Testable

Conditions are the most important levels of the hierarchy. Experience has shown that, providing there is a clear understanding of the roles that these levels play in requirements management, the recommendation provided is sufficient when extracting requirements.

Requirement Coverage Analysis

Coverage Analysis is a means of identifying how requirements will be managed in terms of the following criteria:

- Must the functionality be validated by running one or more tests?
- Must a Risk Assessment be performed on the requirement?
- Will the requirement be used to analyse the testing times and testing levels?

It is closely linked to the Requirement Type hierarchy as it used to define the role that the requirement type will play in determining application readiness. To do this, when defining requirement types, they should be placed into one the following categories

- Analysis: Items used to establish testing times and ultimately the Testing Level
- Assessment: Items used in Risk Categorization
- Test: Items that need to be validated

Using both requirement type and coverage categories will result in each Requirement Type being defined as follows:

Application under Test:

- Does not require validation
- Will be used for Requirement Analysis

Function Group:

- Does not require validation
- Will be used for Requirement Analysis

Application Functions:

- Does not require validation
- Will be used for Requirement Analysis

Functional Element: (User Story)

- Does not require validation
- Requirement needs to be assessed for risk

Testable Condition (Acceptance Criteria)

- Requires validation (Needs one or more tests)
- No risk assessment or requirement analysis required

This process completes the definition of each documented requirement type.

Measuring Requirement Risk

How much testing should we be doing?"

- It will depend on the project constraints
    - Time
    - Budget
    - Team Dynamics
- You must still be able to provide information to all stakeholders
    - So that they can make informed decisions about moving to the next stage
    - So that they can make informed decisions about releases to production
- It will depend on the risk of application failure to your organisation
    - Technical and Business (product/project)
    - Implementation Complexity
- We will use risk to determine
    - The prioritization of your tests
    - Where to place the emphasis
- What to test first
- What to test most
    - How thoroughly to test each item
    - What not to test (this time!)
- Where to start?
    - To assist with answering this question, requirements should be divided into the following classes:
- Must test
- Should test
- Can test
- Will test if time permits

The next step in the process is to apply some sort of Risk Indicator/Index to the Assessment Class requirements. The indicator is usually made up of two sub-indicators, one that looks at the business value and the other determines the technical risk of the requirement. The way these sub-indicators are derived is to use a simple method of evaluating a standard set of risk factors to a standard set of values. Each value is provided with a weighted factor. The sum of the weighted factors will place the requirement into one of the organisations documented risk categories

Each risk factor can be measured by a number of risk condition, but it is always best to try and limit the number of weighted values to three. The first subindicator is called the Business Priority, is made up of two types of Risk Influences

- Business Value
- Cost of Failure
- The Business Value of a requirement is a measure of how important the requirement is to the business

- Cost of Failure is a measure of what it will cost and organisation in the event of failure.
- As a rule of thumb, keep the categorization as simple as possible and restrict it to three possible groups, for example:
    - A -Business Critical
    - B - Important but not Critical

- C - Nice to Have

- The second sub-indicator is called the Technical Risk, is also made up of two types of Risk Influences
  - Technical Complexity
  - Functional Density
- The Technical Complexity of a requirement is a measure of the number of independent paths in the context flow
- Functional Density is a measure of the number of elements involved and how this impacts its implementation.
- Once more, keep it simple and also restrict it to three possible groups, for example:
  - 1 - High
  - 2 - Medium
  - 3 - Low

The combination of the Business Priority and Technical Risk sub-indicators make up the requirements Risk Indicator. The Risk Indicator is determined by simply concatenating the two sub-indicators (A1 to C3) or if your organisation follows the IEEE829 standard for Test Planning, it would be determined using a matrix as follows: -

Business Priority	Technical Risk		
	1-High	2 - Medium	3 - Low
A - Business Critical	Cat 1	Cat 2	Cat 3
B - Important but not critical	Cat 4	Cat 5	Cat 6
C - Nice to have	Cat 7	Cat 8	Cat 9

The final step in defining the Risk Policy is to calculate the group boundaries. This is done by defining the maximum, middle and minimum boundaries as follows: -

- The maximum boundary is defined by adding the highest number of weighted factors for each risk factor together
- In contrast, the minimum boundary is defined by adding the lowest number weighted factors for each risk factor together
- The middle boundaries, the lowest value for the 1st and 2nd groups, are not calculated, but are determined per individual organisation and based on agreed values.
- If I had 5 Risk factors with a weighted maximum value of 40 and a minimum value of 10 the boundaries would be:

Testing Levels

In line with deciding how much testing we should be doing, the recommended best practice of Defining Testing Levels in an organisation is to use the following four levels: -

- Full (100%) (must test) - a complete end-to-end depth test is conducted on requirements

- Partial (should test) - expressed as a percentage of full test and consists of a reduced end-to-end depth test on these requirements
- Basic (can test) - also expressed as a percentage of full test and consists of breadth testing only on these requirements (positive tests only)
- None (0%) (will test if time permits)

Assuming that you will use a percentage of full testing, the following is an example defining the testing levels in an organisation

Testing Levels	
Partial	70%
Basic	30%

Testing Times

Best practice dictates that when using Risk-based Quality Management that the organisation assigns a default testing time per risk category. Either one of the Risk sub-indicator can be used as the basis for the policy. The following table is an example of setting up the testing times in an organisation using the method of calculating the times as a percentage of full testing

Testing Time per Technical Risk	
1 - High	18 Hours
2 - Medium	15 Hours
3 - Low	12 Hours

Putting it all together

The process of defining the default Testing Time per Risk Category of an organisation is made up of five components as follows: -

- Units used to measure testing effort can be set in Hours, Days, Weeks or Months
- Testing Time per Risk Sub-indicator. Either one of the sub-indicators can be used. The time values in these fields will depend on the individual organisation's requirements
- Testing Levels. These values indicate a proportional percentage of the testing time for each level
- Default Testing Policy. The values in this grid display the amount of testing time and the testing level to be allocated to each requirement based on its Risk Indicator
- It must be noted that using the percentage of full testing is not the only way of defining default testing times per risk category.
- Organisations are at liberty to allocate a default time directly to each of the nine risk categories
- The Testing Time per Risk Category is displayed using a matrix
- The following table is an example of setting up the calculated testing times in an organisation

Testing Level	Technical Risk		
	1 - High	2 - Medium	3 - Low
Full (100%)	18 hours	15 hours	12 hours
Partial (70%)	12 hours	10 hours	8 hours
Basic (40%)	6 hours	5 hours	4 hours
None (0%)	0	0	0

The following table is an example of setting up the default testing policy in an organisation

Business Risk	Technical Risk		
	1 - High	2 - Medium	3 - Low
A - Business Critical	Full (18)	Full (15)	Full (12)
B - Important but not critical	Partial (12)	Partial (10)	Partial (8)
C - Nice to have	Basic (6)	Basic (5)	Basic (4)

This matrix displays the Testing Policy of an organisation using the Risk Indicator, Testing Times and Testing Levels as should be used as a starting point when estimating testing times. Though not ideal, these times can be manipulated to a point where required testing time will match available testing time in the event of time constraints.

Introduction 6: Managing the Test Process

Estimating Time and Effort

In general, estimating any job involves a number of tasks:

- Identify tasks
- How long for each task
- Who should perform the task
- When should the task start and finish
- What resources are available
- What skills level
- Predictable dependencies
- Task precedence
- Technical precedence

In addition to the estimating testing times and effort previously mentioned tasks, there are additional destabilizing dependencies that exist t in Testing: -

- Testing is an independent activity
- Delivery schedules for testable items are missed
- Fixed target dates
- Test environments are critical

When estimating time and effort take not of the following: -

Managing Up

- Under commit
- Over deliver

Managing Down

- Over commit
- Under deliver

Both can have an effect on the stake holder's perception on the abilities of the Test Manager

Estimating Methods

Whenundertaking the estimation, you need to consider a number of elements. Some or all may apply in each case:

- Quality of specifications
- Size and complexity of application
- Requirements for non- functional testing
- Stability and maturity of development process
- Type and location of test environments
- Use of test tools
- Skills of people involved
- Time available
- Amount of re-work require

Methods that can be used include:

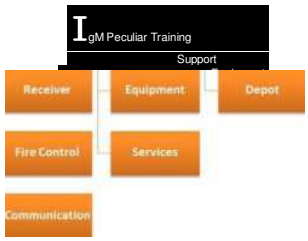
- Risk Based: Involves the use a risk indicator (9 categories - IEEE 829). Default testing times and testing levels are allocated to each risk category

- Team Velocity
    - This is the method used mainly on Agile projects
    - It uses 'Story Points' to determine the Agile Teams delivery capability
- Metrics based
    - Measures of previous or similar projects
- If we have historical or typical value
    - Expert based
    - Assessment by experts or task owner
- Dependent on expertise
- Other approaches

FIA (finger in the air)

Work breakdown structure

Level 1



- Test point analysis
- Estimation model

• The estimation is based on number of test cases and approximate/average time to execute

- Perform a sampling exercise on a small piece of the new functionality

Test Environment

Check environment requirements

- Confirm the testable state of the function/application under test
- Ensure that the testers have the correct access permissions to:
    - Databases
    - Host systems
    - Input / Output data
- Test cycles
- Deployment standards
- Set the ground rules
- Balance between flexibility and maintaining a reasonable base, e.g. use of environment by other departments

Exit criteria <u>Purpose</u>

- Define when to stop testing (at a particular test level)
- Example
    - Thoroughness measures
- Coverage of code, functionality or risk
    - Estimates of defect density or reliability measures
    - Cost or schedules
- Effort spent, time elapsed
    - Residual risks
- Defects not fixed, low coverage in some areas

Defect Management

- Strategy defined in the Test Plan
- Roles and responsibilities clearly defined
- Tool customized
- Graphs filtered and saved
- Standards for capture, qualify and escalation process is clearly defined and has been communicated to all

Test Schedule

- Project Gantt that Indicates tasks, status and planned dates
- Is a living document that must be referenced daily
- It can be created in some sort of planning software like MS Project
- You can also create your plan using MS Excel

Data Plan

- Ensures that each team member has data prepared for each environment.
- Governs and provides evidence.
- Data must include: -
    - Static Data - This is data that does not change and should represent data conditions in the 'live' environment
    - Dynamic Data- This is the data that changes during testing and represents the data that is used during test execution

Execution Planner

- Simple calculator that defines the test execution targets per day, per tester and module.
- Must be updated daily to re-plan and shuffle the test execution.
- Remember to use the 'Estimate to Completion' of outstanding work and not 'Work Done' to measure progress
- Mitigate test execution risks from day 1.
- Proactively determine if the execution will be on schedule.
- The team knows their targets per day. Defined goals.
- In Agile this task is referred to as the 'Burn-down Chart'

Reporting

- Key component of the role of the Test Manager
- The quality of your reports define you
- Communication mechanism
- Must be easy to understand and follow
- Must meet your stakeholder's expectations
- 2 Types of Reports

- Phase specific
- Interim

Interim Reports

- Can be requested any time during the project
- Recommended best practice that these kind of reports be produced on a daily/weekly basis
- They should take the form of an Executive Summary
    - needs to be informative enough to give readers the summary of where the testing is at. What took place today, risks and challenges and the plan for the next day
    - Risk table- risk description, status and mitigating actions.
    - Test Design or Test execution stats
    - QC (or test tool) reports- Test execution summary, defect summaries.
- Sample Report - Daily Status Report

Phase Specific Reports

- These reports are produced at the end of a phase, including the end of the project
- They include some of the following examples
    - Phase/Project signoff
    - Post Implementation Reports
    - Post Implementation Performance Scoring

Introduction 7: Controlling the Testing

Process **Test Control**

Test control can be defined as the test management tasks that is required throughout the test process in order to keep the testing aligned with the software development process, the needs of the project, and the needs of the organisation

- Compare actual progress against the plan
- Report the status, including deviations from the plan
- Take corrective or pre-emptive action in order to meet the objectives

Control involves monitoring throughout the project

- Progress
- Coverage
- Exit criteria

Make decision

- Who, what, when, where, etc.

Testing Budget

The Testing Budget represents the total number of hours in a testing cycle. It is calculated by

adding together the total number of hours each team member will be available to the project during the testing cycle.

Testers must remember to make provision for times that are required for nonproject activities (non-project meetings, lunch breaks, tea breaks, smoke breaks, ad-hoc leave (sick, lame and lazy). The progress of the project is monitored by measuring the time available against the amount of work still to be completed.

Measuring Test Execution Progress

- Use the Agile principle of a daily Burn-down Chart
- It's a simple yet powerful tool to help track the progress of the project
- Imagine lighting a candle at the start of testing which will burn out when the allocated time on the project reaches 0 hours.
- During the testing phase the only part of the candle that is visible is the amount of candle that still has to burn (the estimated time to completion)
- The idea behind the burn-down chart is to simply plot progress the project visually on a graph
- The original estimates are show on one line showing the testing budget moving towards 0 hours
- The 'Estimate to Completion (E.T.C.)' is plotted on a different line
- When the actual line drops below the original estimation line this indicates that the project is running ahead of time
- When the actual line climbs above the original line this indicates that the project is behind schedule and corrective action is necessary
- Using E.T.C instead of completed work has the advantage of high-lighting the impact of scope creep on the project as soon as it is known
- Each individual is responsible to update their E.T.C. on the functions they are working on every morning
- The team goal is quite simply to be able to deliver working software when the project reaches h 0 hours

Reasons for Test Progress Monitoring

- To provide feedback and visibility of testing activities
- Show how we are doing against the plan: What metrics do you keep and why?
- Adequacy of test objectives for this test level
- Adequacy of test approaches taken
- Effectiveness of the testing with respect to its objectives

Common Metrics

Percentages/ comparison

- Test cases written vs. plan
- Test execution (run/not run, passed/failed
- Preparation of environment, test cases etc.
- Actual against milestone dates
- Costs against budget Defect information
- Found/fixed, failure rate, density, re-test Coverage of code or application

Confidence reporting in application

Introduction 8: Managing Risk in Your Project

The Essentials of Managing Risk in Your Project

Managing Risk is a significant aspect of project management, irrespective of whether you are managing the whole project or just the Quality Assurance of the end product

According to the Project Management Institute's (PMBOK), Risk management is one of the ten knowledge areas in which a proj'ect manager must be competent.

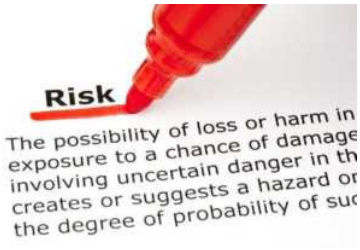
As a test manager, you need to identify risks early, as possible, so you can determine how you will manage them. However, you have to know more than just what risks might exist. You also have to know how best to discuss them with the rest of the test team members and even make decisions about which risks you will try to manage:

Understanding and Managing Risk in Your Project

Risk is the possibility that you may not achieve your product, schedule, or resource targets because something unexpected occurs or something planned doesn't occur. All projects have some degree of risk because predicting the future with certainty is impossible.

Project risk increases under the following circumstances: -

- Very long projects
- The longer the time is between preparing your project plan and starting the work
- The less experience you, your organisation, or your team members have with similar projects
- The newer your project's technology is



In other words, anything that can cause you either to fall short of or to exceed your established project targets, if it occurs, is considered a risk. While some approaches for analyzing and responding to both types of risks are similar. In most cases, the term *risk* always refers to a negative risk or threat, unless otherwise noted.

Risk management is defined as the process of

i. Identifying possible risk.
ii. Assessing their potential consequences.
iii. Developing and implementing plans for minimizing any negative effects.

Risk management can't eliminate risks, but it offers the best chance for successfully accomplishing your project despite the uncertainties of a changing environment.

Use the following process to determine, evaluate, and manage the risks that may affect your project:

i. Identify risks. - Determine which aspects of your plan or project environment may change.
ii. Assess the potential effects: - Consider what can happen if those aspects don't work out the way you

envision.

iii. Develop plans for mitigating the effects of the risks. - Decide how you can protect your project from the consequences of risks.

iv.

v. Monitor the status of your project's risks throughout performance. - Determine whether existing risks are still present, whether the likelihood of these risks is increasing or decreasing, and whether new risks are arising.

vi. Inform all stakeholders of all risks involved with your project. - Explain the status and potential effect of all project risks — from the initial concept to the project's completion.

Project Risk manifests itself into two distinct categories: -

i. Project Risk

ii. Product Risk

In most cases, even though the Test Manager has to concern him/herself with some of the known project risks, Quality Assurance concerns itself with Product Risk.

Product risk, also sometimes referred to as Quality risk, is defined as the possibility that the system or software might fail to satisfy, or fulfil, some reasonable expectation of the customer, user, or stakeholder.

The simple differences between Product Risks and Project Risks

The first step is to understand the differences between these two entities:

Project Risks

These are situations that may or may not happen and if they materialize they usually cause delays in the project's timelines, and the source of these risks may be internal or external. The responsibility for Project Risks are the Project Managers, who are also in charge of the project's schedule.
QA, as part of the overall team, is responsible for the Project Risks within the testing work areas.
There are many ways manage all the risks for the project and project managers will use their preferred method of achieving this task

The most usual project risks related to the QA & Testing work are:

i. Delays in the delivery of the AUT for testing

ii. Lack of technical knowledge on specific areas of the product

Lack of testing environments and/or data that effectively simulate real customer usage etc.

iii. Unreasonable time lines.

iv. Unrealistic estimation of development time and effort

Product Risks

These are the areas in the AUT where there is a high risk of finding defects, usually due to changes or other internal

factors.

As testers one of our tasks is to manage Product Risks. Our role in the QA process is to be aware of all Product Risks, to make sure that all stakeholders are also aware of them, and to coordinate this information with Project Management in order to ensure that our schedules are taking these risks into account. In addition, we are expected to plan our testing strategy based on these risk, scheduling more tests (and earlier tests) on areas with higher risks in order to find these issues faster.

The preferred method for handling Product Risks is to document the Risk Management Policy in the Test Plan. The policy is typically made up of five sub-policies namely:

i. Requirement Types (Analysing testing time and effort, assessing for risk, or conducting one or more test to validate the requirement)
ii. Requirement Coverage
iii. Requirement Risk
iv. Default Testing Times
v. Default Testing Levels

Product Risk Analysis

Product risk analysis involves analysing the product to be tested with the aim of achieving a joint view for the test manager, as well as all the stakeholders, of the properties of the product to be tested. This view should represent all risk levels (high, medium and low), as well the thoroughness of testing that can be related to this view.

Product Risk Analysis of a requirement is determined by adopting a very simple method of assigning a risk indicator to the requirement. (Also known as a Risk Index). The indicator made up of two sub-indicators, one that looks at the business risk and the other determines the technical risk of the requirement. Both sub-indicators are derived by measuring a standard set of risk factors to a standard set of values. Each value is provided with a weight factor and sum of the weighted factors will place the requirement into one of the organisations documented risk categories.

Each Risk Factor can have many risk values, but it is always best to try and limit the number of weighted values to three. The first sub-indicator is called the Business Risk, is made up of two types of Risk Influences

- Business Value
- Cost of Failure

The Business Value of a requirement is a measure of how important the requirement is to the business. As a rule of thumb, keep the categorization as simple as possible and restrict it to three possible values, following the 'MoSCoW pattern, for example:

i. *Must have:* Cannot be missed.
ii. *Should have:* Almost essential; nice to have this feature if at all possible.
iii. *Could have:* Not essential but we could have this if it does not affect anything else.

iv. *Would have:* Not essential now but we would like to have it in the future.

- A - Critical (Must have - cannot be missed)
- B - Important (Should have - Almost essential)
- C - Nice to Have (Could have or Would have - Not essential - something we would like in the future)

The Cost of Failure is a measure of what it will cost and organisation in the event of failure. It measures the chance that the product will fail in relation to the expected damage a failure will cause.

Product risk = Chance of Failure * Damage
Where Chance of failure = Chance of faults * Use frequency

The chance of failure is the chance that a product (component) will fail during operational use because it contains a fault. The presence of a fault in the product, however, does not mean that this fault will actually manifest as a failure in production. The chance that the product will fail increases with the frequency of its use.

Damage relates to the negative impact resulting from the failure of the product. Product failure may result in damage for multiple stakeholders.

Examples of damage that may occur are:

i. Loss of market share
ii. Compensation claims
iii. Damage to image and reputation, for example through corrective measures
iv. Increase in aftercare and higher maintainability costs
v. Higher than planned costs of internal provisions
vi. Extra staffing needs

The second sub-indicator is called the Technical Risk, is also made up of two types of Risk Factors

- Technical Complexity
- Functional Density

The Technical Complexity of a requirement is a measure of how complex it will be to deliver. It used to measure the number of linearly-independent paths through a program module, i.e. the complexity of a program source code.

An easy method of calculating the complexity of a program would be to: -

i. Count the number of decisions (diamond shaped boxes) on a control flow graph and add 1 .... or
ii. Count the number of IF's and any loop constructs (DO, FOR, WHILE, REPEAT) and add 1 if working from code

Functional Density is a measure of the number of components (screens, reports, etc.) involved and how this impacts its implementation.

Once more, keep it simple and also restrict it to three possible values, for example:

- 1 - High (Complex)
- 2 - Medium (Moderate)
- 3 - Low (Simple)

The outcome of the PRA will have a direct effect of the Testing Times and Testing levels of the requirement. Once again the decision on the testing levels will be based on the MoSCoW acronym

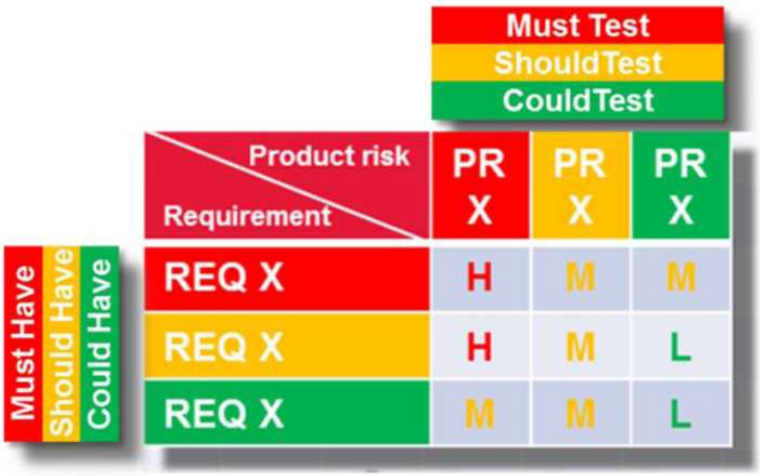
i. *Must test:* Full (100%) must be tested, otherwise no acceptance. (a complete end-to-end depth test is conducted on requirements (100%)

ii. *Should test:* It's important to test this functionality (usually expressed as a percentage of the unit assigned to a full test (66%) and consists of a reduced end-to-end depth test on these requirements.

iii. *Can test:* Conduct basic testing (also usually expressed as a percentage of the unit assigned to a full test (34%). Consists of breadth testing only on these requirements (positive tests) Tests can be skipped after consulting the stakeholders.

iv. *Won't test:* We will not test this. If we have time, we will test this

v.



Examples of Product Risks are:

Complex features affecting multiple areas of the existing product, like an upgrade/migration of the system.

New Technologies used in the product; for example, a new DB server, a new programming language, a new integration, etc.

New Developers or Development Teams, who may lack experience and thus pose a higher risk to the existing product.

iv. Tight Schedules, that make people work in a rush and commit more mistakes'

v. There's always a chance that users will reject your product even if the product that has been delivered matches requirements

Product Risk Reporting:

The following image is an example of a product risk report.

lllkq					
■ No Imminent Problems ■ Points of Concern ■ Urgent Resolution Required					
This report serves as a weekly status report					
Risk Description	**Date**	**Raised By**	**Responsibl**	**PROJECT Warning Flag**	**Comments / Actions**
				■	
				■	

Introduction 9: People Development

Empowering the Testing Professional

In order for testing to be effective, the tester must be provided with

- Means - a suitable environment within which to work. This includes appropriate hardware and proper standards and procedures
- Ability - Adequate skills development
- Accountability - Regular 'Performance Appraisals' measured against organisational standards

Skills Development

The SQA skills framework deals with the People side of the SQA Maturity Model™ and provides the delegate with skills required to perform the core competencies of a Professional Tester.

Why do we recommend the SQA Standard

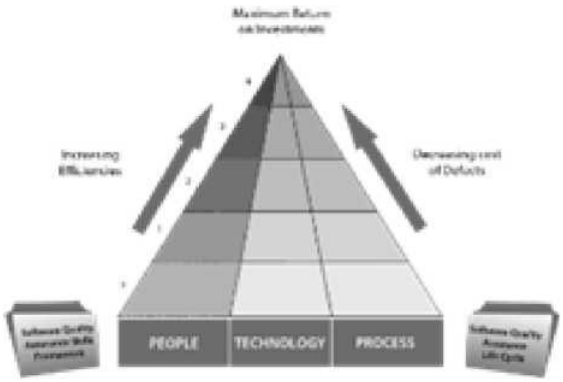
- The SQA Vision is to become the SKILLS PROVIDER OF CHOICE by gaining the reputation within the IT Industry of providing skilled resources that are head and shoulders above everyone else
- Expert Trainers based on a combination of +/- 60 years of testing experience, expert knowledge and subject matter experts.
- All training material is outcome based, and we encourage all our delegates to become solution driven.
- The training covers all the skills, on the core competencies of a Professional Tester, that are required for international certification

SQA People Certification

This certification is awarded on the competency of individual to execute quality control within a development lifecycle.

It is divided into three levels:

- SQA Certified Test Analyst
- SQA Certified Testing Professional
- SQA Certified Test Management Professional



Team Building

- Have you ever wondered how some work groups exhibit effective team work and others remain dysfunctional for the life of the team?
- Effective team work is both profoundly simple and difficult at the same time.
- The factors that affect success in team work occur both within the team itself and in the work environment in which the team must function.
- People in every workplace talk about building the team, working as a team and my team
- Few understand how to create the experience of team work or how to develop an effective team.
- Belonging to a team, in the broadest sense, is a result of feeling part of something larger than yourself.
- It has a lot to do with your understanding of the mission or objectives of your organisation.
- In a team-oriented environment, one contributes to the overall success of the organisation.
- You work with fellow members of the organisation to produce these results.
- Even though you have a specific job function and you belong to a specific department, you are unified with other organisation members to accomplish the overall objectives.
- The bigger picture drives your actions; your function exists to serve the bigger picture.
- Fostering teamwork is creating a work culture that values collaboration.
- In a teamwork environment, people understand and believe that thinking, planning, decisions and actions are better when done cooperatively.
- People recognize, and even assimilate, the belief that "none of us is as good as all of us."

Successful Teamwork

- In order for successful team work to take place, they need to be operating in the right environment
- Building the right environment requires the following issues to be in place
- The team understands the goals and is committed to attaining them.
- People are comfortable taking reasonable risks in communicating, advocating positions and taking action
- Communication is open, honest, and respectful. (Team members feel free to express their thoughts, opinions and potential solutions to problems)
- Team members have a strong sense of belonging to the group. (They experience a deep commitment to the group's decisions and actions)
- Team members are viewed as unique people with irreplaceable experiences, points of view, knowledge and opinions.
- Creativity, innovation and different viewpoints are expected and encouraged.
- The team is able to constantly examine itself and continuously improve its processes, practices and the interaction with each other
- The team has agreed upon procedures for diagnosing, analysing, and resolving team work problems and conflicts
- Participative leadership is practiced in leading meetings, assigning tasks, recording decisions and commitments, assessing progress, holding team members accountable, and providing direction for the team.
- Members of the team make high quality decisions together and have the support and commitment of the group to carry out the decisions made.

Successful team work is the cornerstone for creating functioning, contributing teams.

Building the Team

- Many work environments tend to focus on individuals and personal goals, with reward & recognition singling out the achievements of individual employees.
- Team building usually refers to the process of selecting or creating a team from scratch.
- Team building refers to a wide range of activities, designed for improving team performance.
- When building a team one needs to differentiate between
    - the overall sense of teamwork from
    - the task of developing an effective intact team that is formed to accomplish a specific goal.
- Unfortunately, people tend confuse these two team building objectives which is the primary reason why team building initiatives fail
- Team building is pursued via a variety of practices that range from simple bonding exercises to complex simulations and multi-day team building retreats designed to develop a team
- These include things like group assessment and group-dynamic games or something somewhere in between
    - Team building is also an important factor in any environment as its focus is to specialize in bringing out the best in a team to ensure self-development, positive communication, leadership skills and the ability to work closely together as a team to problem solve.

- Team building can also be seen in day- to-day operations of an organisation
- Team dynamic can be improved through successful leadership
- Team building should also be seen as:
- Opportunity to socialize
- Must be enjoyable for "everyone

Team Dynamic

When assembling a team, it is very important to consider the overall dynamic of the team. Experts on the subject of Team Dynamics maintain that when building a team, five dynamics are fundamental to team success:-

- The team member: Successful teams are made up of a collection of effective individuals. These are people who are experienced, have problem solving ability, are open to addressing the problem, and are action oriented.
- Team relationships: For a team to be successful the members of the team must be able to give and receive feedback.
- Team problem solving: An effective team depends on how focused and clear the goal of the team is. A relaxed, comfortable and accepting environment and finally, open and honest communication are required.
- Team leadership: Effective team leadership depends on leadership competencies. A competent leader is: focused on the goal, ensures a collaborative climate, builds confidence of team members, sets priorities, demonstrates sufficient "know-how" and manages performance through feedback.
- Organisational environment: The climate and culture of the organisation must be conducive to team behaviour. Competitiveness should be discouraged and uniformity should be encouraged - this will eliminate conflict and discord among team members.

Managing Upwards

Upward management is one of those skills that some do very well, many never seem to master, and virtually all learn only through on-the-job lessons-learned. When done well, both the manager and employee work as a team to ensure each other is informed, address problems before they spin out of control, and be more effective at managing. When done poorly, both manager and employee are not only ineffective at getting the job done but are chronically frustrated due to missed deadlines and surprises.

Why is It necessary?

Line Managers

- Your line manager probably won't recommend you for a promotion, pay rise or bonus if they don't know what you've been up to, how you are progressing or where you'd like your career to go.
- Yes, it's their job to help you get ahead, but it's also your job to put them in a position to be able to do so.
- You need to make sure you communicate with them regularly, let them know when a project you work on has gone well and demonstrate how you personally have added value both to your client and to your firm.
- You also need to make sure your line manager understands your personal career goals.
- Don't rely solely on your formally arranged management meeting or annual / six monthly reviews.
- Take the initiative and arrange a fortnightly telephone call or send a regular email.
- Remember, assignments can be a long way from your home office so you might not be able to just catch them in the corridor.

Test/Team Manager

- The same rules apply for the test/team manager
- Communication is the key.
- Make sure that they know what you are working on
- Make sure that they know what your capacity is
- Make sure they know if and when you are likely to miss a deliverable
- Treat your manager as if they are your client
- If you make their jobs easier by managing upwards you will reap the reward further down the line when the project is successful
- Effective upward management will help you get ahead of the competition and assist you to climb the career ladder just that little bit quicker

Formulating a PDP

Introduction

- High achievers focus effort on achieving their goals.
- This very good, except that many times they neglect working on what's most important to their success.... i.e. themselves; their skills, abilities, knowledge etc.
- Goal setting takes care of what we want and not what we need

- Having a plan helps ensure that we are continually working on ourselves

Basic Foundation of a PDP

Identify where you are right now

- Perform a SWOT analysis
- Make use of your Performance Scorecards
- Use the results of the Skills Analysis that you have done
- Identify any Key Performance Indicators that you have

Identify where you want to be

- What needs to be worked on for you to get to the next level
- Which weaknesses need to be addressed
- What areas should be given priority
- Identify short and long term goals

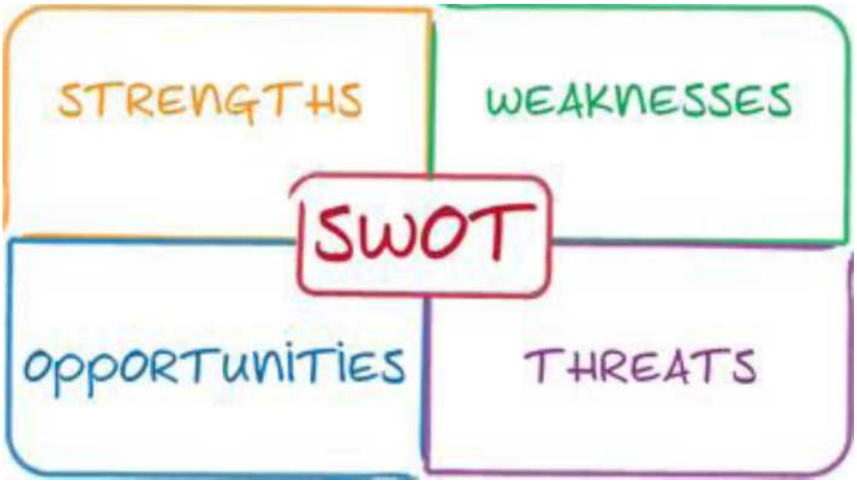
Create a plan to fill the gap

- Identify resources, scheduling etc.
- Set SMART goals (Specific, Measurable, Attainable/Action-oriented, Results-oriented/Relevant and

Time-frame limited

- Set up your time-line
- Action the plan (initiate kick-off)
- Work towards achieving your goals
- Monitor your progress

Preparing the SWOT Analysis



List your personal strengths

- These are your personal skills, credentials and other features that give you an advantage in the workforce.
- Examples could include your education, specialized knowledge and important business contacts.
- Do not write down every skill you have and every training course you have ever taken; only list the ones that separate you from your peers.

List your personal weaknesses

- Your weaknesses are the deficiencies that you have compared to others in your field of work.
- If you are a professor, for example, and you have a master's degree, this may be a weakness because most professors have doctorates.
- These weaknesses can indicate where you need to make improvements if you want to have a competitive chance of advancing in your career.

Write done a list of career opportunities

- These can include job opportunities in your own firm or other firms.
- Opportunities should also include educational opportunities that are available to you.
- Educational opportunities can range from university degrees and college diplomas, to professional development courses that can be taken in a few hours or on weekends.
- Be sure to think about all the opportunities available to you, even those you do not think you want to pursue.

List all the threats to your career security

Examples of threats might include

- New graduates with more credentials.
- An economic downturn that is causing layoffs in your sector
- Personal things such as poor performance reviews. You should include anything that may threaten the security of your job or advancement opportunities.

You should include anything that may threaten the security of your job or advancement opportunities.

Create a strategy

- Write down how you plan to change the status quo, based on your strengths, weaknesses, opportunities and threats.
- Use this strategy to plan what you will do next in your career

Performance Management

What is Performance Management?

Performance management is defined as an integrated and strategic approach, that is concerned with performance improvement and continuous development, which is necessary to achieve organisational, team and individual effectiveness.

Objectives of Performance Management

- Means of linking individual objectives to the organisation's goals, by measuring and monitoring performance against targets.
- Learning from areas of strength and taking corrective action on areas of weakness.
- Improved accountability - clarifying the output and outcomes to be achieved

Pre-requisites to Effective Performance Management

- Clear strategy
- Clearly defined role and responsibilities
- Continuous review and feedback
- Lead by example
- Joint accountability
- Focus on development and not judgment
- Climate
- Consistency

Key Diagnostic Measures

- Quality
- Delivery
- Competency
- Attitude

Criteria for Measuring Quality

Quality of Deliverable

- Delivers high quality work
- Assists in improving the quality of other team member's work Risk Management

Identifies and raises any risks on a project which might result in late delivery, system unreliability, or not meeting the stated requirements

Focus

Is constantly aware of and able to focus on goals and objectives and achieve targets within suitable and agreed upon time frames

Ownership and Initiative

Takes ownership of problems and tasks and shows initiative in solving them.

Self-Management

Plans and executes own work effectively and efficiently within available time.

Drive and Perseverance

Continuously pursues solutions to problems until conclusion. Puts in a focused effort to resolve problems. Ability to make things happen.

Criteria for Measuring Competency

Test analysis

- Is able to analyse and interpret specifications (Use Cases and UML) and identify possible gaps.
- Contributes to the requirement gathering process.

Test planning

- Is able to develop and document a test plan.
- Identifies systems involved in testing and arranges access to the systems and a test environment.
- Ensures that all scope change requests are incorporated in the test pack.

Test scripting/design

- Is able to produce thorough, well organized, complete test scripts incorporating all requirements as per the specification, using Quality Center.

Demonstrates an understanding and use of all test design techniques to identify test conditions

Test execution

- Is able to execute tests in a thorough and concise manner.
- Ensures that all defects are logged and valid and that defects are brought to the test manager and developer's attention for correction.
- Monitors that turn-around time for correction is within the agreed-upon period.
- Generates the appropriate test documentation.

Systems and Tools Knowledge

- Understands the systems impacted within the specific projects.
- Assesses and applies the appropriate tools to complete the necessary testing.

Problem Solving

- Understands the process of analytical thinking
- Demonstrates a logical and systematic approach to solving problems

Criteria for Measuring Attitude

- Teamwork and Relationships: Assists other staff willingly and effectively. Maintains good working relationships with co-workers and customers.
- Listening: Ability to listen, understand and respond effectively.
- Feedback and Communication: Provides regular, accurate and relevant feedback on progress to managers and team members.
- Flexibility: Is able to listen to, analyse and accept other opinions and views. Also, is flexible in the type of work the person is willing to do.

Introduction 10: SQA Maturity Model and Associated Templates **SQA**

Maturity Model™

What is the Software Quality Assurance Maturity Model™?

- Benchmark against which you can measure the efficiencies of your People, Technology, and Disciplines and Processes
- Skills development and performance management framework for testing professionals
- Process framework for the application of quality assurance and testing to the Product / Systems Development Life Cycle

The benefits of implementing model

Efficiencies

- Reduction in development cost - Process
- Staff retention - People
- Quick to market products - Process
- Optimised availability of systems - Technology
- Measure and monitor
- Increased Return on Investment
- Increased Return on Intellectual Capital
- Flexible and customizable
- Optimize strategic management capacity
- Tailor made for the financial and telecommunications sectors
- Industry approved standards
- Certified Staff
- Capacity to focus on implementation

Level 0

People	Technology and Disciplines	Process
No defined roles and responsibilities Developers, business analysts or support staff perform systems testing Poorly skilled testing staff	Unstructured testing No testing tools	Ad-hoc testing as no defined processes are in place

Level 1:

People	Technology and Disciplines	Process
At least level one competencies for testers, test analysts and test manager No people measures or monitors in place Clearly defined roles and responsibilities	Functional testing repository created Manual regression testing Practiced Test management tool implemented	Quality assurance and testing processes implemented, i.e. the Software Quality Assurance Lifecycle

Level 2:

People	Technology and Disciplines	Process
Average of level two competencies for Test Analysts and Test Manager Basic Performance Management measures in place	Standards and guidelines utilised Test cases designed for automation Basic Performance, Configuration, Security and Usability Testing practiced	Defect Management process implemented Basic Test Management Reports generated Test environment Management process implemented Release management process Implemented

Level 3:

People	Technology and Disciplines	Process
Average of level three competencies for all testing professionals Comprehensive performance management	Automated Regression implemented Application Security Testing implemented Usability Test Lab in place Configuration Test Lab in place Comprehensive Performance Testing utilising appropriate tools	Test Data Management process implemented Performance, Configuration, Security and Usability Testing processes implemented Comprehensive Test Management reports generated

Level 4:

People	Technology and Disciplines	Process
Level three competencies for all testing professionals	Optimum utilisation of tools and all testing disciplines implemented	Comprehensive Test Management reports generated across disciplines

Realising the Benefits

Associated Templates

There are a number of templates available to the Test Manager when using SQA Certification standards

- Roles and Responsibilities
- Balance Scorecards
- Test Repository Standards
- Release Process
- Test Schedule
- Test Plan
- Data Plan
- Reporting Templates
- Execution Planner
- Defect Management Process
- Cycle Report
- Sign-off Report
- Post Implementation Report

Time Management

Goal - increased return on time management

- Effective use of the test strategy
- Stakeholder reporting
- Business admin vs. getting involved in operational matters
- Make use of management tools (MS Project, Outlook calendars and reminders, test management tools)

Lead vs. Do-It-Yourself

- Create an environment where staff do not rely on management to resolve issues.
- Do not get involved in test execution
- Empower the team members through training, process and team organisation

When do you stop?

- Don't take your work home with you
- Effective use of your time at work
- Stop working +/- 30 mins prior to leaving to enable you to recap and plan your activities for the next day

Relaxation Techniques

- Eat meals away from your desk.
- Gather your thoughts
- Chat to you team about their areas of interest

Practice what you preach

- Use your time effectively
- No procrastinating
- Be on time (arriving at work, meetings, etc.)

Study Unit 1.1: Data Management Issues and a DBMS

US 114049

Demonstrate an understanding of Computer Database Management Systems

Specific Outcome 1	
IBS"	Describe data management issues and how it is addressed by a DBMS
Assessment Criteria	
	1. The description identifies the problem they represent and includes examples. 2. The description outlines ways which database management systems address the issues.

Introduction to Database Management Systems: (DBMS):

Database management systems (DBMSs) are computer software applications that interact with the user, other applications, and the database itself to capture and analyse data.

There are also many different types of database management systems, ranging from small systems that run on personal computers to huge systems that run on mainframes.

Some DBMS examples include MySQL, PostgreSQL, Microsoft Access, SQL Server, FileMaker, Oracle, RDBMS, dBase, Clipper, and FoxPro. Since there are so many database management systems available, it is important for there to be a way for them to communicate with each other. For this reason, most database software comes with an Open Database Connectivity (ODBC) driver (explained in detail later) that allows the database to integrate with other databases. For example, common SQL statements such as SELECT and INSERT are translated from a program's proprietary syntax into a syntax other databases can understand.

A general-purpose DBMS is designed to allow the definition, creation, querying, update, and administration of databases.

Basic Terminology:

i. Database: - A collection of data, modelling a real-world enterprise

ii. Database Management System: Computer software applications that interact with the user, other applications, and the database itself to capture and analyse data.

iii. Applications: The software to access and process the data (Enters the business logic. e.g. enter exam grades or purchase something on-line)

iv. Information System: Groups everything together (Database + DBMS + Applications). Other types of information systems, based on information-retrieval or knowledge-base, are also available

DBMS Organisation

From a technical standpoint, database management systems differ widely. The terms relational, network, flat, and hierarchical all refer to the way a DBMS organizes information internally. The internal organisation can affect how quickly and flexibly you can extract information.

DBMS Connectivity

In computing terms, Open Database Connectivity (ODBC) is a standard application programming interface (API) for accessing database management systems (DBMS) which has been designed specifically to make it

independent of database systems and operating systems. An application written using ODBC can be ported to other platforms, both on the client and server side, with few changes to the data access code.

ODBC accomplishes DBMS independence by using an ODBC driver as a translation layer between the application and the DBMS. The application uses ODBC functions through an ODBC driver manager with which it is linked, and the driver passes the query to the DBMS. An ODBC driver works similarly to that of a printer driver or other driver, providing a standard set of functions for the

application to use, and implementing DBMS-specific functionality.

An application that can use ODBC is referred to as "ODBC-compliant". Any ODBC-compliant application can access any DBMS for which a driver is installed. Drivers exist for all major DBMSs, many other data sources like address book systems and Microsoft Excel, and even for text or comma-separated values (CSV) files.

ODBC was originally developed by Microsoft during the early 1990s, and became the basis for the Call Level Interface (CLI) standardized by SQL, used for access group in the Unix and mainframe space. However, full ODBC retained several features that were removed as part of the CLI effort and was later ported back to those platforms, and became a de facto standard considerably better known than CLI. The CLI remains similar to ODBC, and applications can be ported from one platform to the other with relatively few changes.

Requests for information from a database are made in the form of a query, which is a stylised question e.g. the query would be styled as follows: -

```
SELECT ALL WHERE NAME = "SMITH" AND AGE > 35
```
requests all records in which the NAME field is SMITH and the AGE field is greater than 35.

The set of rules for constructing queries is known as a query language and different DBMSs support different query languages. There is a semi-standardized query language called SQL (structured query language), which is used in most cases.
There are also some sophisticated languages available for managing database systems that are called fourth- generation languages (or 4GLs for short).

A fourth-generation programming language (4GL) is a computer programming language that attempts to get closer than 3GLs, to human language form of thinking and conceptualisation. They are designed to reduce the overall time, effort and cost of software development. SQL is an example of a 4GL language. Wikipedia lists more than 45 known 4th generation computer languages

The information from a database can also be presented in a variety of formats. Most DBMSs include a report writer program that enables you to output data in the form of a report. Many DBMSs also include a graphics component that enables you to output information in the form of graphs and charts.

1.1.1 Problems Represented by DBMS and how they are Addressed

The following are some briefly described problems that might arise in the management of research, financial, or administrative data.

Access File System

Typically, data is stored in files.

1. Flat files usually contain one table at a time.
2. Flat files contain values at each row and the rows are separated with a special symbol. For you to be able to reach the data you have to parse each row and obtain an array of values before you can query the data.
3. To control the data in a file, you have to read it line by line and parse it.
4. There is no control mechanism in files.
5. DBMS systems make use of special language called SQL to modify and reach the data easily.
6. DBMS systems make use of indexes to reach the data, so that it is not necessary to read it line by line.
7. There are much more control mechanisms that approves the correctness of data.
8. You can easily and safely reach the data across networks by using a DBMS system.

Need for Database Management Systems

Data access through file systems suffers from the following problems.

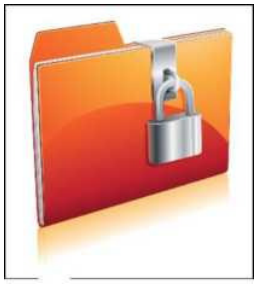
- Minimal data redundancy
- Difficulty in accessing data
- Inconsistency
- Concurrent access
- Data isolation
- Integrity
- Atomicity
- Security

A good database management system solves all the above-mentioned problems which a user generally faces in a file system.

Shared Use

Data is protected and maintained much better when it can be shared using a DBMS instead of creating new iterations of the same data stored in new files for every new application. The DBMS provides a central store of data that can be accessed by multiple users in a controlled manner.

Security and Risk



Database security deals with all various aspects of protecting risk regarding the database content, its owners, and its users. It ranges from protection from intentional unauthorised database uses to unintentional database accesses by unauthorised persons or computer programs.

Database access control deals with controlling who (a person or a certain computer program) is allowed to access what information in the database. The information may consist of specific database objects (e.g., record types, specific records, data structures), certain computations over certain objects (e.g., query types, or specific queries), or the use of specific access paths to such objects (e.g., using specific indexes or other data structures to access information). Database access controls are set by personnel that were specially authorised by the database owner, and that are dedicated protected security DBMS interfaces.

Data security prevents unauthorised users from viewing or updating the database. Users are allowed access to the entire database or subsets of it called "subschemas" by using passwords. For example, an employee database can contain all the data about an individual employee, but one group of users may be authorised to view only payroll data, while others are allowed access to only work history and medical data.

Data security in general deals with protecting specific parts of data, both physically (i.e., from corruption, or destruction, or removal), or the interpretation of them, or parts of them to meaningful information.

Integrity

Data integrity refers to the overall completeness, accuracy and consistency of data. This can be indicated by the absence of alteration between two instances or between two updates of a data record, meaning data is intact and unchanged. Standard procedures and rules are usually imposed during the database design phase to ensure data integrity. Data integrity can be maintained through the use of different error checking methods and validation procedures.

Data integrity is enforced in both hierarchical and relational database models. The following three integrity constraints are used in a relational database structure to achieve data integrity:

- Entity Integrity: This deals with the concept of primary keys. The rule states that every table must have its own primary key and that each has to be unique and not zero.
- Referential Integrity: This is the concept of foreign keys. The rule states that the foreign key value can be in two states. The first state is that the foreign key value would refer to a primary key value of another table, or it can be zero. Being zero could simply mean that there are no relationships, or that the relationship is unknown.
- Domain Integrity: This states that all columns in a relational database are in a defined domain.

The concept of data integrity makes it possible to trace and connect all data in a database to other data. In this way everything is recoverable and searchable. Having a single, well-defined and well-controlled data integrity system increases stability, performance, reusability and maintainability. If one of these features cannot be implemented in the database, it must be implemented through the software

Privacy

How Database Information is Leaked

SQL Injection - One of the best-known database exploits around, SQL injection happens when unexpected data is sent to a webserver which interacts with a database. The data may be sent through a number of methods, but is always specially formed, containing SQL statements which can compromise any unprotected data source. Data is usually passed through either the address or through form variables.

Browser Method: When data is passed through the address to a page which interacts with a database, the data is passed in the form of variables appended to the end of the address. For example "www.cnn.com?id=1234" would pass 1234 as data to cnn.com If this data is directly used in a database command, one could change 1234 in a specific way as to introduce a SQL command to be executed by the database. This ability could give the attacker a powerful tool to compromise data and accounts of an unprotected database. The user is able to change the address variables at will, which makes this an extremely easy database leak anyone can exploit

Form Method: Forms take data on one page and pass that data directly to another page for processing, usually with a backend database involved. In this case the user is prevented from modifying address variables as the form data, if passed properly, is hidden from the user. Because the transfer is hidden, sites will believe they have satisfactorily sanitised user input by enforcing rules (digits only for certain fields, length limits) with client side JavaScript. This is supposed to prevent users from entering SQL statements disguised as bad input, but a clever user could download the source of the file and modify the data subverting any JavaScript. Entering erroneous SQL statements such as '; drop table users --could compromise an entire database and its users.

Preventing SQL Injection

Syntax Checking: The trick of any SQL injection attack is that the user is able to insert malicious statements into invalidated user input. To protect against this problem users must sanitise input that is collected from the user on the server side. The types of sanitation that must be performed is to remove any semicolons or back tick marks, as these are the stronghold of the syntax required for a proper SQL injection.

While checking syntax is good, a prepared statement is the best prevention of SQL injection. When using prepared statements, also known as stored procedures, every interaction with a database is prewritten, which allows only enough rights as are required for any given command. Thus when an unexpected, powerful command is run there are not enough rights allocated for the malicious command under the ruse of a stored procedure to be properly executed even if it is properly inserted.

Reliability

Reliability is the measure of the success with which the system conforms to some authoritative specification of its behaviour. When the behaviour deviates from that which specified is, this is called failure.

Levels of reliability:

- application dependent
- application independent

The application dependent part refers mainly to the integrity constraints. The application independent part refers mainly to the ensuring of the ACID concepts.

Main aspects of reliability are:

- Correctness: The system works in accordance with the corresponding specifications. Some kind of software testing methods can be used. There is no appropriate tool to proof the correctness of a complex software system.
- Availability: Fraction of the time that the system meets its specification.

Usually the requirements of the correctness and of the availability cannot be met at the same time. When correctness in increased, availability is decreased; when availability in increased, correctness is decreased

Main Problem Areas of Reliability

- To provide an appropriate protocol for transaction commitment: A method is needed which could provide a good trade-off in the case of failures too.
- To provide an appropriate protocol for data replication: A data element may be replicated at different sites. How can the different instances managed in the case of failures?
- Determining the state of the network: How could we ensure that every participant has the correct. information regarding the operational states of the other partners?
- Detection and resolution of inconsistencies: How can we detect and resolute the inconsistencies caused by a not fully correct decision of the co-ordinator?
- Checkpoints and cold restarts: How can the global consistency of the distributed database ensured in the case of a cold restart?
- Commission errors: How can we detect and correct actions that seem formally well but semantically are incorrect?

Performance

Several key performance factors influence database performance. Understanding these factors helps to identify performance opportunities and avoid problems:

- System Resources
- Workload
- Throughput
- Contention
- Optimisation

System Resources

Database performance relies heavily on disk I/O and memory usage. To accurately set performance expectations, it is important to know what the baseline performance of the hardware is on which the DBMS is deployed. Performance of hardware components such as CPUs, hard disks, disk controllers, RAM, and network interfaces will have a huge influence on how fast the database performs.

Workload

The workload equals the total demand from the DBMS, and it varies over time. The total workload is a combination of user queries, applications, batch jobs, transactions, and system commands directed through the DBMS at any given time. For example, it can increase when month-end reports are run or decrease over weekends when most users are not in the office. Workload strongly influences database performance. If you know your workload and peak demand times, it will help in the

planning of the most efficient use of the system resources and will enable the processing of the largest possible workload.

Throughput

A system's throughput defines its overall capability to process data. DBMS throughput is measured in queries per second, transactions per second, or average response times. DBMS throughput is closely related to the processing capacity of the underlying systems (disk I/O, CPU speed, memory bandwidth, and so on), so it is important to know the throughput capacity of the hardware when setting DBMS throughput goals.

Contention

Contention is the condition in which two or more components of the workload try to use the system in a conflicting way — for example, multiple queries that try to update the same piece of data at the same time or multiple large workloads that compete for system resources. As contention increases, throughput decreases.

Optimisation

DBMS optimisations can affect the overall system performance. SQL formulation, database configuration parameters, table design, data distribution, and so on help the database query planner and optimiser to create the most efficient access plans.

Integration

The most overlooked data integration issues and challenges include:

- Using a data profiling tool to analyse and measure data quality of both source or target environment data before the development of the integration logic.
- Establishing data acceptance criteria for an individual application system. Perfect data isn't practical; the focus should be "good enough." And the business stakeholders involved in an integration project should be able to identify usage scenarios that reflect what "good enough" looks like.
- Including a data quality/accuracy review step as part of the design review step in an overall development project.
- Establishing standard load and export interfaces for systems that provide data and take it on board on a regular basis. Most integration development follows a one-off approach even when it's common for some systems to on-board data from new systems on a regular basis.

Data Administration

Some organisations define separate roles for the business aspects and the technical aspects of data. The business aspects of data are aligned with data administration, whereas the more technical aspects are handled by database administration. Not every organisation has a data administration function. Indeed, many organisations combine data administration into the database administration role.

Sometimes organisations also split up the technical aspects of data management, with the DBA that is responsible to use the DBMS and a system administrator or systems programmer responsible for installing and upgrading the DBMS.

Data Administration

Data administration separates the business aspects of data resource management from the technology used to manage data; it is more closely aligned with the actual business users of data. The task of the data administrator (DA) is to understand the business lexicon and translate it into a logical data model. The DA would be involved in the requirements gathering, analysis, and design phase, the Data Base Administrator (DBA) in the design, development, testing, and operational phases.

Another difference between a DA and a DBA is the focus of effort. The DA is responsible for the following tasks:

- Identifying and cataloguing the data required by business users
- Producing conceptual and logical data models to accurately depict the relationship among data elements for business processes
- Creating an enterprise data model that incorporates all of the data used by all of the organisation's business processes
- Setting data policies for the organisation
- Identifying data owners and stewards
- Setting standards for control and usage of data

Issues

At a basic level, master data management seeks to ensure that an organisation does not use multiple (potentially inconsistent) versions of the same master data in different parts of its operations, which can happen in large organisations.

Other problems include (for example) issues with the quality of data, consistent classification and identification of data, and data-reconciliation issues. Master data management of vastly different data systems requires data transformations as the data extracted from the different source data system is transformed and loaded into the master data management hub. To synchronise the disparate source master data, the managed master data extracted from the master data management hub is again transformed and loaded into the disparate source data system as the master data is updated.

Solutions

Processes commonly seen in master data management include source identification, data collection, data transformation, normalisation, rule administration, error detection and correction, data consolidation, data storage, data distribution, data classification, taxonomy services, item master creation, schema mapping, product codification, data enrichment and data governance.

*

The tools include data networks, file systems, a data warehouse, data marts, an operational data store, data mining, data analysis, data visualization, data federation and data virtualization. One of the newest tools, virtual master data management makes use of data virtualisation and a persistent metadata server to implement a multi-level automated master data management hierarchy.

Transmission of master data

There are several ways in which master data may be collated and distributed to other systems. This includes:

- Data consolidation - This process captures master data from multiple sources and integrating into a single hub (operational data store) for replication to other destination systems.
- Data federation - This process provides a single virtual view of master data from one or more sources to one or more destination systems.
- Data propagation - This process of copies master data from one system to another, usually through point-to-point interfaces in legacy systems.

Study Unit 1.2: Commonly Implemented Features of Commercial Database Management Systems

US 114049

Demonstrate an understanding of Computer Database Management Systems

Specific Outcome 1	
IBS"	Describe commonly implemented features of commercial database management systems..
Assessment Criteria	
	1. The description identifies the purpose of each feature. 2. The description identifies the way in which each feature contributes to the solution of data management issues.

1.2.1 The Purpose of Each Feature and Solution of Data Management Issues

Data Access Tools

Data access is the process where a database is entered to store or retrieve data. Data Access Tools are end user oriented tools through which users build Structured Query Language (SQL) queries by pointing and clicking on the list of table and fields in the data warehouse.

Through computing history, there have been different methods and languages already that were used for data access and these varied depending on the type of data warehouse. The data warehouse contains a rich repository of data with regards to organisational business rules, policies, events and histories and these warehouses store data in different and incompatible formats. Because of this several data access tools have been developed to overcome problems of data incompatibilities. Some of these more

popular standards include SQL, OBDC, ADO.NET, JDBC, XML, XPath, XQuery and Web Services.

Structured Query Language (SQL) is a computer language used in Relational Database Management Systems (RDBMS) to retrieve and manage data. Although SQL has been developed to be a declarative query and data manipulation language, many vendors have created SQL DBMS and added their own procedural constructs, data types and other propriety features. SQL is standardised both by ANSI and ISO.

ODBC, which stands for Open Database Connectivity is a standard software application programming interface that is used for data management systems. Different computer languages can access data into different types and implementation of RDBMS using the ODBC.

JDBC which stands for Java Database Connectivity is very similar to ODBC, but is used for the Java programming language.

ADO.NET stands for ActiveX Data Object, and is a Microsoft proprietary software component to access data and data services. This is part of the Microsoft .Net framework.

XML stands for Extensible Markup Language is basically a general purpose markup language. It is used to tag data so that sharing of structure data can be done through disparate systems across the internet or any network. This makes data of any format portable among different computer systems, which makes XML one of the most used technologies in data warehousing.

XML data can be queried using XQuery. Semantically this is almost the same with SQL. XML Path Language is used to address parts of an XML document or other computing values like strings, Booleans, number and others based on any XML document.

Web services are software components that make the interoperability of machine to machine interaction over the internet possible. They are known as Web API that are accessed over the internet and execute on another remote system.

Many software vendors develop applications that have Graphical User Interface (GUI) tools so that even nonprogrammers or non-database administrators can build queries by just clicking the mouse. This GUI data access tools give users access via data access designer and data access viewer. With the data access designer, an end user can create complex databases even if he/she does not have good background.

Readymade templates are available that are complete with design framework and sample data. With the data access viewer, the user can run and enter data and make changes and modifications and graphically see the commands without having to be involved with the complex process happening in the background.

Data access tools makes the tasks of database administrators a lot easier especially if the database that is being managed is a large data warehouse. Having a graphical interface for data access gives the administrator a clearer status of the database because most programmatic query languages may look cryptic on the command line interface.

Backup and Recovery

Backup and recovery are methods that allow you to protect your data from loss. The database system provides a separate process, from that of a network backup, that can be used to back up and recover data. The only way to recover the database If a hard drive fails and the database stored on the hard drive is not accessible, is from a backup.

If a computer system fails in the middle of a complex update process, the recovery subsystem makes sure that the database is restored to its original state. These are two more benefits of a database management system

Backup

As long as new data are being created and changes are being made, it will be necessary to do backups frequently. Individuals and organisations with anything from one computer to thousands of computer systems all need to protect their data. The scales may be very different, but the objectives and limitations are basically the same. Those who perform backups need to know how successful the backups are, regardless of scale.

Objectives

Recovery point objective (RPO)

The point in time that the restarted infrastructure will reflect. Basically, this is the roll-back that will be experienced as a result of the recovery. The most desirable RPO would be the point just before the data loss event. To make a more recent recovery point achievable, the frequency of synchronisation between the source data and the backup repository must be increased.

Recovery time objective (RTO)

The amount of time that it took between disaster and restoration of business functions.

Data security

Apart from preserving access to data for its owners, data must also be restricted from unauthorised access. Backups must be performed in such a way that it does not compromise the original owner's undertaking. This can be done with data encryption and proper media handling policies.

Data retention period

Regulations and policy may require backups to be retained for a specific period, but not any further. Retaining backups after this period can lead to unwanted liability and sub-optimal use of storage media.

Limitations

An effective backup scheme will take into consideration the limitations of the situation.

Backup window

The period of time when backups are allowed to run on a system is called the backup window. This is usually the time when the system sees the least usage and the backup process will have the least amount of interference with normal operations. The backup window is usually planned with users' convenience in mind. If a backup extends past the defined backup window, a decision is made whether it is better to abort the backup or to lengthen the backup window.

Performance impact

All backup schemes have some performance impact on the system being backed up. For example, for the period of time that a computer system is being backed up, the hard drive is busy reading files for the purpose of backing up, and its full bandwidth is no longer available for other tasks. It is important that such impacts are analysed.

Costs of hardware, software, labour

All types of storage media have a finite capacity with a real cost. An important part of the design of a backup scheme is to match the correct amount of storage capacity (over time) with the backup needs. Any backup scheme has some labour requirement, but complicated schemes have considerably higher labour requirements. The cost of commercial backup software can also be very high.

Network bandwidth

Distributed backup systems can be affected by limited network bandwidth.

Implementation

To meeting the defined objectives with regards to the above limitations can be a difficult task. The tools and concepts below can make that task more achievable.

Scheduling

The use of a job scheduler can improve the reliability and consistency of backups by removing part of the human element to a great extent. Many backup software packages include this functionality.

Authentication

Over the course of normal operations, the user accounts and/or system agents that perform the backups need to be authenticated at some level. The power to copy all data off of or onto a system asks for unrestricted access. Using an authentication mechanism is a good way to prevent the backup scheme from being used for purposes that are not authorised.

Chain of trust

Removable storage media are physical items and must only be handled by trusted individuals. Establishing a chain of trusted individuals (and vendors) is extremely important to defining the security of the data.

Measuring the process

To ensure that the backup scheme is working as expected, it is important to monitor key factors and maintain historical data.

Backup validation

This provides information about the backup, and proves compliance to regulatory bodies outside the organisation. Many organisations rely on third-party or "independent" solutions to test, validate, and optimize their backup operations (backup reporting).

Reporting

In larger configurations, reports are useful to monitor media usage, device status, errors, vault co-ordination and other information about the backup process.

Logging

In addition to the history of computer generated reports, activity and change logs are useful to monitor backup system events.

Validation

Many backup programs make use of checksums or hashes to validate that the data was accurately copied. There are several advantages to this. First, data integrity can be verified without reference to the original file: if the file as stored on the backup medium has the same checksum as the saved value, then it is very probably correct. Second, some backup programs can use checksums to avoid making redundant copies of files, and that will improve the backup speed. This is especially useful for the de-duplication process.

Recovery

Crash Recovery

DBMS is a highly complex system with hundreds of transactions that are executed every second. The durability and robustness of a DBMS depends on its complex architecture and its underlying hardware and system software. If it fails or crashes while doing transactions, it is expected that the system would follow some sort of algorithm or techniques to recover lost data.

Failure Classification

To see where the problem has occurred, a failure is generalised into the following categories:

Transaction failure

A transaction has to abort when it fails to execute or when it reaches a point from where it can't go any further. This is called transaction failure where only a few transactions or processes are hurt.

Possible reasons for a transaction failure are:

- Logical errors - Where a transaction cannot complete because there is some code error or any internal error condition.
- System errors - Where the database system itself terminates an active transaction because the DBMS cannot execute it, or it has to stop because of some condition in the system. For example, in case of deadlock or resource unavailability, the system aborts an active transaction.

System Crash

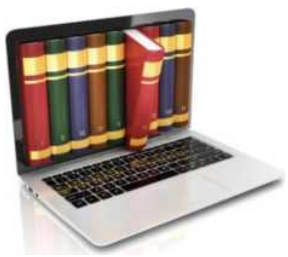
There are problems - external to the system - that may cause the system to stop suddenly and cause the system to crash. For example, interruptions in power supply may cause the failure of underlying hardware or software failure. Examples can also include operating system errors.

Disk Failure

In early days of technology evolution, it was a common problem where hard-disk drives or storage drives used to fail quite often. Disk failures include formation of bad sectors, unreachability to the disk, disk head crash or any other failure, which destroys all or a part of disk storage.

Storage Structure

In short, the storage structure can be divided into two categories:



- Volatile storage: As the name suggests, it is not possible for a volatile storage to survive system crashes. Volatile storage devices are placed very close to the CPU; normally they are embedded onto the chipset itself. Main memory and cache memory are examples of volatile storage. They are fast but can store only a small amount of information.
- Non-volatile storage: These memories are made to survive system crashes. They can store a huge amount of data, but is slower in accessibility. Examples are hard-disks, magnetic tapes, flash memory, and non-volatile (battery backed up) RAM.

Recovery and Atomicity

When a system crashes, it may have several transactions that are being executed and different files that are open for them to modify the data items. Transactions are made of different operations, which are atomic in nature. But according to ACID properties of DBMS, atomicity of transactions as a whole must be maintained, that is, either all the operations are executed or none.

When a DBMS recovers from a crash, the following should be maintained:

- It should check the states of all the transactions, which were being executed.
- A transaction may be in the middle of some operation; the DBMS must ensure the atomicity of the transaction in this case.
- It should check whether the transaction can be completed now or whether it needs to be rolled back.
- No transactions would be allowed to leave the DBMS in an inconsistent state.

There are two types of techniques, which can help a DBMS to recover as well as maintain the atomicity of a transaction:

- Maintaining the logs of each transaction, and writing them onto some stable storage before actually modifying the database.
- Maintaining shadow paging, where the changes are done on a volatile memory, and later, the actual database is updated.

Recovery with Concurrent Transactions

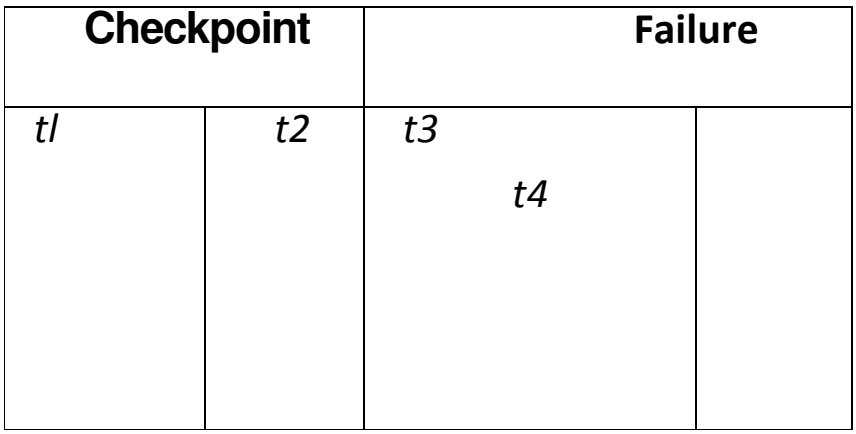
When more than one transaction are being executed in parallel, the logs are interleaved. At the time of recovery, it would become hard for the recovery system to backtrack all logs, and then start recovering. To assist with this situation, most modern DBMS use the concept of 'checkpoints'.

Checkpoint

Keeping and maintaining logs in real time and in real environment may mean that all the memory space available in the system is filled out. As time passes, the log file may grow too big to be handled at all. Checkpoint is a mechanism where all the previous logs are removed from the system and stored permanently in a storage disk. Checkpoint declares a point before which the DBMS was in consistent state, and all the transactions were committed.

Recovery

When a system with concurrent transactions crashes and recovers, it behaves in the following way:



Time

- The recovery system reads the logs backwards from the end to the last checkpoint.
- It maintains two lists, an undo-list and a redo-list.
- If the recovery system sees a log with <Tn, Start> and <Tn, Commit> or just <Tn, Commit>, it puts the transaction in the redo-list.
- If the recovery system sees a log with <Tn, Start> but no commit or abort log found, it puts the transaction in undo-list.

All the transactions in the undo-list are then undone and their logs are removed. All the transactions in the redo- list and their previous logs are removed and then redone before saving their logs.

Audit

Database auditing is basically a facility to track the use of database resources and authority. When auditing is enabled, each audited database operation produces an audit trail of information, which includes information such as what database object was impacted, who performed the operation and when. The comprehensive audit trail of database operations that were produced can be maintained over time so that DBAs and auditors, as well as any authorised personnel, can perform in-depth analysis of access and modification patterns against data in the DBMS.

Database auditing helps to answer questions like

- Who accessed or changed data?
- When was it actually changed?
- What was the old content prior to the change?

The ability to answer such questions can make or break a compliance audit. Sometimes it may be necessary to review certain audit data in greater detail to determine how, when and who changed the data.

Tracking who does what to which piece of data is important because there are many threats to the security of and organisation's data. External agents trying to compromise security and access the organisation's data are definitely a threat. But many industry studies show that most security threats are internal - within an organisation. The most typical security threat comes from a disgruntled or malevolent current or ex-employee that has valid access to the DBMS. Auditing is vital because it may be necessary to find an unauthorised access emanating from an authorised user.

A typical auditing facility allows auditing at different levels within the DBMS, for example, at the database, database object level, program level and user levels. One of the biggest problems with existing internal DBMS audit facilities is performance degradation. The audit trails that are produced must be detailed enough to capture before- and after-images of database changes. But capturing so much information, especially in a busy system, can cause poor performance. Furthermore, this audit trail must be stored somewhere and this can cause a problem when a huge number of changes occur. Therefore, a good auditing facility must allow for the selective creation of audit records to minimise performance and storage problems.

Database Access Auditing Techniques

There are several popular techniques that can be used to audit database structures. Let's briefly discuss three of them and highlight their pros and cons.

Trace-based auditing

This technique is usually built directly into the native capabilities of the DBMS. Commands or parameters are set to turn on auditing and the DBMS begins to cut trace records when activity occurs against audited objects. Each DBMS offers different auditing capabilities, and some common items that can be audited by DBMS audit facilities are:

- Login and logoff attempts (both successful and unsuccessful attempts)
- Database server restarts
- Commands issued by users with system administrator privileges
- Attempted integrity violations (where changed or inserted data does not match a referential, unique, or check constraint)
- Select, insert, update, and delete operations
- Stored procedure executions
- Unsuccessful attempts to access a database or a table (authorization failures)
- Changes to system catalogue tables
- Row level operations

The problems with this technique are:

- A high potential for performance degradation when audit tracing is enabled
- A high probability that the database schema will need to be modified
- Insufficient granularity of audit control, especially for reads

Scan and Parse Database Transaction Logs

Every DBMS uses transaction logs to capture every database modification for recovery purposes. Software interprets these logs and identifies what data was changed and by which users. The drawbacks to this technique are that reads are not captured on the logs. There are ways to disable logging that will cause modifications to be lost, performance issues scanning volumes and volumes of log files looking for only specific information to audit and it is difficult to retain logs over long periods for auditing when they were designed for short-term retention for database recovery.

Proactive Monitoring of Database Operations at the Server

This technique captures all SQL requests as they are made. It is important that all SQL access is audited, not just network calls, because not every SQL request goes over the network. This is especially important for mainframe platforms where much of the activity is centralised and the most important business transactions never venture over an IP network (e.g., a CICS transaction accessing DB2).

Proactive audit monitoring does not need transaction logs, does not need database schema modification, should be highly granular in terms of specifying what to audit, and should incur only minimal overhead.

Distributed Data Management

A distributed database is a database in which portions of the database are stored on different computers within a network. Users have access to the portion of the database at their location so that they can access the data that is relevant to their tasks without interfering with the work of others. A centralised distributed database management system (DDBMS) manages the database as if it were all stored on the same computer. The DDBMS synchronises all the data from time to time and, in cases where a number of users must access the same data, ensures that updates and deletes performed on the data at one location will be automatically reflected in the data stored elsewhere.

Transaction Processing

The aim of transaction processing is to maintain a system's Integrity (typically a database or some modern filesystems) in a known, consistent state, by ensuring that interdependent operations on the system are either all completed successfully or all cancelled successfully.

Transaction processing links a number of individual operations in a single, indivisible transaction, and ensures that either all operations in a transaction are completed without error, or none of them are. If some of the operations are completed but errors occur when the others are attempted, the transaction-processing system "rolls back" all of the operations of the transaction (including the successful ones), in this way erasing all traces of the transaction and restoring the system to the consistent, known state that it was in before processing of the transaction began. If all operations of a transaction have been completed successfully, the transaction is committed by the system, and all changes to the database are made permanent. Once this is done the transaction cannot be rolled back.

Transaction processing makes sure that there are no hardware and software errors that might leave a transaction partially completed. If the computer system crashes in the middle of a transaction, the transaction processing system makes sure that all operations in any uncommitted transactions are cancelled.

Usually, transactions are issued concurrently. If they need to touch the same portion of the database (overlap), this can create conflicts. However, forcing transactions to be processed in sequence is not efficient. Therefore, concurrent implementations of transaction processing is programmed to guarantee that the end result reflects a conflict-free outcome, the same as could be reached if the transactions were executed sequentially in any order.

Methodology

The basic principles of all transaction-processing systems are the same. However, the terminology may be different from one transaction-processing system to another.

Rollback

Transaction-processing systems ensure the integrity of the database by recording intermediate states of the database as it is modified, then using these records to restore the database to a known state if a transaction cannot be committed. For example, copies of information on the database before its modification by a transaction are set aside by the system before

the transaction can make any modifications. If any part of the transaction fails before it is committed, these copies are used to restore the database to the state it was in before the start of the transaction.

Rollforward

It is also possible to keep a separate journal of all modifications to a database management system. This is not necessary for rollback of failed transactions but it is useful to update the database management system when there is a database failure, so some transaction-processing systems provide it. If the database management system fails completely, it must be restored from the most recent back-up. The back-up will not show any transactions that were committed since the back-up was made. However, once the database management system is restored, the journal of after images can be applied to the database (rollforward) to bring the database management system up to date. Any transactions in progress at the time of the failure can then be rolled back. This results in a database that is in a consistent, known state that includes the results of all transactions committed up to the moment of failure.

Deadlocks

In some cases, two transactions may, while they are being processed, attempt to access the same portion of a database at the same time, in a way that stops them from proceeding. For example, transaction A may access portion X of the database, and transaction B may access portion Y of the database. If, at that point, transaction A then tries to access portion Y of the database while transaction B tries to access portion X, a deadlock occurs, and neither transaction can move forward. Transaction-processing systems are designed to detect these deadlocks as they happen. Both transactions will now be cancelled and rolled back, and then they will be started again in a different order, automatically, so that the deadlock doesn't occur again. Or sometimes, just one of the deadlocked transactions will be cancelled, rolled back, and automatically restarted after a short delay.

Deadlocks can also occur among three or more transactions. The more transactions involved, the more difficult it is to detect them, to the point that transaction processing systems find there is a practical limit to the deadlocks they can detect.

Compensating transaction

In systems where commit and rollback mechanisms are not available or cannot be used, a compensating transaction is often used to undo failed transactions and restore the system to a previous state.

Benefits

Transaction processing has the following benefits:

- Computer resources can be shared among many users
- The time of job processing can be shifted to when the computing resources are less busy
- It avoids idling the computing resources without minute-by-minute human interaction and supervision
- It is used on expensive classes of computers to help amortise the cost by keeping high rates of utilisation of those expensive resources

Study Unit 1.3: Different Type of DBMS's.

US 114049

Demonstrate an understanding of Computer Database Management Systems

Specific Outcome 1	
A	Describe different type of DBMS's..
Assessment Criteria	
ni	1. The description describes characteristics of the DBMS-type. 2. The description gives examples of the use of the DBMS-type.

1.3.1 Characteristics and Examples of Use of DBMS

There are different types of Database Management Systems, of which the following are examples:

- Hierarchical
- Relational
- Network
- Object

Hierarchical

A hierarchical database is a design that uses a one-to-many relationship for data elements. Hierarchical database models use a tree structure that links a number of disparate elements to one "owner," or "parent," primary record.

The idea behind hierarchical database models is useful for a specific type of data storage, but it is not very versatile. Its limitations mean that it has very specific uses. For example, where each employee in a company may report to a given department, the department can be used as a parent record and the individual employees will represent secondary records, each of which links back to that one parent record in a hierarchical structure.

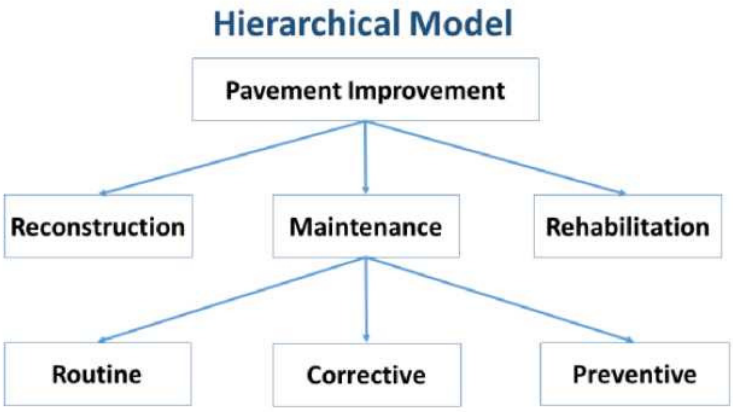
Hierarchical databases were popular in early database design, in the era of mainframe computers. While some IBM and Microsoft models are still in use, many other types of business databases use more flexible models in order to use more sophisticated types of data management. Hierarchical models make the most sense where the primary focus of information gathering is on a concrete hierarchy such as a list of business departments, assets or people that will all be associated with specific higher-level primary data elements.

Example of a Hierarchical Model

A record in the hierarchical database model corresponds to a row (or tuple) in the relational database model and an entity

type corresponds to a table (or relation).

In the hierarchical database model each child record has only one parent, whereas each parent record can have one or more child records. In order to retrieve data from a hierarchical database the whole tree needs to be travelled through, starting from the root node.



Examples of Hierarchical Data represented as Relational Tables

An organisation could store employee information in a table that contains columns such as employee number, first name, last name, and department number. The organisation provides each employee with computer hardware as needed, but only the employee to which it is assigned may use the computer equipment. The organisation could store the computer hardware information in a separate table that includes each part's serial number, type, and the employee that uses it. The tables might look like this:

Employee Table				Computer Table		
Emp No	First Name	Last Name	Dept. Num	Serial Num	Type	User Emp No
100	Peter	Smith	10-L	3009734-4	Computer	100
101	Collin	Tsabe	10-L	3-23-283742	Monitor	100
102	Tabisa	Ndlovu	20B	2-22-726426	Printer	100
103	Jane	Downey	20B	232342	Monitor	100

In this model, the employee data table represents the "parent" part of the hierarchy, while the computer table represents the "child" part of the hierarchy. In contrast to tree structures usually found in computer software algorithms, in this model the children point to the parents. As shown, each employee may possess several pieces of computer equipment, but each individual piece of computer equipment may have only one employee owner.

Look at the following structure

EmpNo	Designation	ReportsTo
10	Director	
20	Senior Manager	10
30	Typist	20
40	Programmer	20

In this, the "child" is the same type as the "parent". The hierarchy stating EmpNo 10 is boss of 20, and 30 and 40 each report to 20 is represented by the "ReportsTo" column. In Relational database terms, the ReportsTo column is a foreign key referencing the EmpNo column. If the "child" data type were different, it would be in a different table, but there would still be a foreign key referencing the EmpNo column of the employees table

Relational

Relational data model is the most used data model around the world for data storage and processing. This model is simple and it has all the properties and capabilities that is needed to process data with storage efficiency.

Concepts

- Tables - In relational data model, relations are saved in the format of Tables. This format stores the relation among entities. A table has rows and columns, where rows represents records and columns represent the attributes.
- Tuple - A single row of a table, which contains a single record for that relation is called a tuple.
- Relation instance - A finite set of tuples in the relational database system represents relation instance. Relation instances do not have duplicate tuples.
- Relation schema - A relation schema describes the relation name (table name), attributes, and their names.
- Relation key - Each row has one or more attributes, known as relation key, which can identify the row in the relation (table) uniquely.
- Attribute domain - Every attribute has some pre-defined value scope, known as attribute domain.

Constraints

Every relation has some conditions that must be true for it to be a valid relation. These conditions are called Relational Integrity Constraints. There are three main integrity constraints:

- Key constraints
- Domain constraints
- Referential integrity constraints

Key Constraints

There must be at least one minimal subset of attributes in the relation, through which can uniquely be identified. This minimal subset of attributes is called key for that relation. If there are more than one such minimal subsets, these are called candidate keys.

Key constraints mean that

- in a relation with a key attribute, no two tuples can have identical values for key attributes

- a key attribute cannot have NULL values

Key constraints are also referred to as Entity Constraints.
Domain Constraints

Attributes have specific values in real-world scenario. For example, age can only be a positive integer. The same constraints have been tried to employ on the attributes of a relation. Every attribute must have a specific range of values. For example, age cannot be less than zero and telephone numbers cannot contain a digit outside 0-9.

Referential integrity Constraints

Referential integrity constraints work on the concept of Foreign Keys. A foreign key is a key attribute of a relation that can also be referred in other relation.

Referential integrity constraint states that if a relation refers to a key attribute of a different or same relation, then that key element must exist.

Examples of the use of Relational data models

A very simple example of a description of some relation variables (relvars) and their attributes:

- Customer (Customer ID, Tax ID, Name, Address, City, State, Zip, Phone, Email,Sex)
- Order (Order No, Customer ID, Invoice No, Date Placed, Date Promised, Terms, Status)
- Order Line (Order No, Order Line No, Product Code, Qty)
- Invoice (Invoice No, Customer ID, Order No, Date, Status)
- Invoice Line (Invoice No, Invoice Line No, Product Code, Qty Shipped)
- Product (Product Code, Product Description)

In this design we have six relation variables: Customer, Order, Order Line, Invoice, Invoice Line and Product. The bold, underlined attributes are candidate keys. The non-bold, underlined attributes are foreign keys.

Usually one candidate key is chosen to be called the primary key and used in preference over the other candidate keys, which are then called alternate keys.

A candidate key is a unique identifier to make sure that no tuple will be duplicated; this would make the relation into something else, namely a bag, by violating the basic definition of a set. Both foreign keys and superkeys (that includes candidate keys) can be composed of several attributes. Below is a table that shows a relation of the example Customer relvar; a relation can be thought of as a value that can be attributed to a reliver.
Customer relation

Customer ID	Tax ID	Name	Address	[More fields...]
1234567890	555-5512222	Holdsworth, L	32 Main Street	
2223344556	555-5523232	Metete, VY	120 Vuyani Road	
3334445563	555-5533323	Meier, J	87 10 th Street	
4232342432	555-5325523	Pillay, D	123 Grace Way	

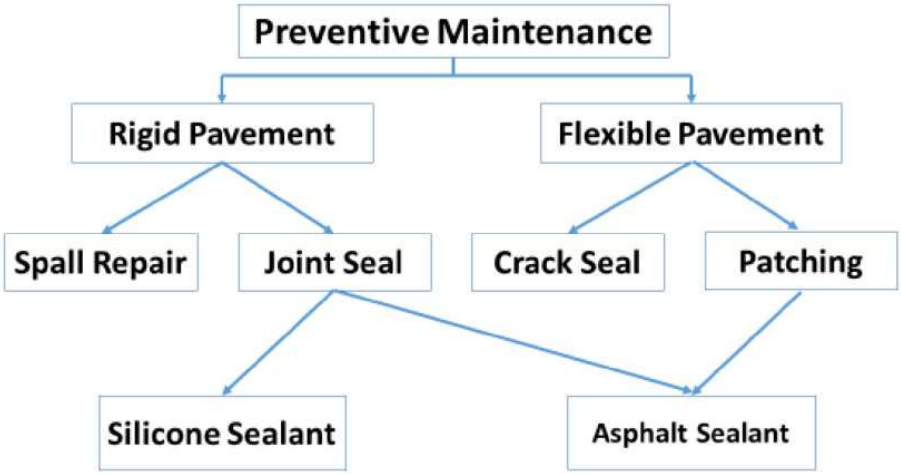
If we would try to insert a new customer with the ID 1234567890, this would violate the design of the relation variable since

Customer ID is a primary key and we already have a customer 1234567890. The DBMS must reject a transaction such as this that would render the database inconsistent by a violation of an integrity constraint.

Network

The network model is a database model that is regarded as a flexible way of representing objects and their relationships. Its distinguishing feature is that the schema, that is seen as a graph in which object types are nodes and relationship types are arcs, is not restricted to being a hierarchy or lattice.

The network model replaces the hierarchical model with a graph, which allows for more general connections among the nodes. The main difference of the network model from the hierarchical model is its ability to handle many to many relationships. In other words it allow a record to have more than one parent.



The basic data modelling construct in the network model is the set construct. A set consists of an owner record type, a set name, and a member record type. A member record type can have that role in more than one set, therefore the multiparent concept is supported. An owner record type can also be a member or owner in another set. The data model is a simple network, and link and intersection record types (called junction records by IDMS) may exist, as well as sets between them.

The most important argument in favour of the network model, in comparison to the hierarchic model, is that it allowed a more natural modelling of relationships between entities.

Advantages of Network Model-

The major advantages of the network model are-

1.) Conceptual simplicity: Just like the hierarchical model, the network model is also conceptually simple and easy to design.
2.) Capability to handle more relationship types: The network model can handle the one to many and many to many relationships, which is beneficial in modelling the real life situations.
3.) Ease of data access: The data access is easier and more flexible than the hierarchical model.
4.) Data integrity: In the network model no member can exist without an owner.
5.) Data independence: The network model is better than the hierarchical model in isolating the programs from the complex physical storage details.

6.) Database standards

Disadvantages of Network Model-

1.) System complexity: All the records are maintained using pointers, therefore the whole database structure becomes very complex.
2.) Operational Anomalies: The insertion, deletion and updating operations of any record need a large number of pointers adjustments.
3.) Absence of structural independence: Structural changes to the database is very difficult.

Object

Object DBMS add database functionality to object programming languages. They bring much more than persistent storage of programming language objects. Object DBMS extend the semantics of the C++, Smalltalk and Java object programming languages to provide full-featured database programming capability, while at the same time they retain native language compatibility. A major benefit of this approach is the unification of the application and database development into a seamless data model and language environment. As a result, applications require less code, use more natural data modelling, and code bases are easier to maintain. Object developers can write complete database applications with a small amount of additional effort.

In contrast to a relational DBMS where a complex data structure must be flattened out to fit into tables or joined together from those tables to form the in-memory structure, object DBMS have no performance overhead to store or retrieve a web or hierarchy of interrelated objects. This one-to-one mapping of object programming language objects to database objects has two benefits over other storage approaches:

- It provides higher performance management of objects
- It enables better management of the complex interrelationships between objects

This makes object DBMS much more efficient to support applications such as financial portfolio risk analysis systems, telecommunications service applications, world wide web document structures, design and manufacturing systems, and hospital patient record systems, which have complex relationships between data.

Object-Oriented Model

Object li Maintenance Report Object 1 Instance

Date	**01.12-01**
Activity Code	***24***
RDUte No.	**1-35**
Daily Production	***2.5***
Equipment Hours	6.0
Labcr Hours	6.0

Object *2:* Maintenance Activity

Activity Code	
Activity Name	
Production Unit	
Average Daily Production Rate	

Study Unit 1.4: Review DBMS End-User Tools

US 114049	
Demonstrate an understanding of Computer Database Management Systems	
Specific Outcome 1	
A	Review DBMS end-user tools...
Assessment Criteria	
	1. The review identifies the features and limitations of the tools. 2. The review outlines the interaction between the tools and the database. 3. The review is based upon use of the tools.

1.4.1 Features, Limitations of Tools and Interaction between Tools and Database

DBMS at STS

Even though STS's core operations are providing professional services in the SQA space, we do have some development of websites for which we use the Mendix application. Development is done through 'smart' models, something like BPMN, instead of a standard computer language. The Database that is used is something called PostgreSQL.

PostgreSQL is the world's most advanced open source database. Developed over 25 years by a vibrant and independent open source community,

Management of the data is performed by the service provider, in this case it's Mendix, who are responsible for finding any anomalies in the data.

This is, of course, a way that STS can have a look at the contents of the database but this involves downloading the back-up copies of the data and signing into the software with an admin user. This is the same as any other DBMS on the market.

Which Database to Choose

Databases should be chosen to match a company's strategy. Databases are the hidden workhorses of many organisations' IT systems, holding critical business data and carrying out hundreds of thousands of transactions each day.

There is a huge choice of database management systems (DBMS), which includes packaged and open source database suites. The

main suppliers include Fujitsu, Hewlett-Packard, Hitachi, IBM, Microsoft, NCR Teradata, Oracle, Progress, SAS Institute and Sybase.

Microsoft SQL Server

One of the biggest selling database is Microsoft's SQL Server. Its growing popularity is partly due to its native integration with the Windows Server software stack, and also because of the technologies it uses, specifically in development, security and business intelligence.

Oracle

The main Oracle database versions in use include Oracle 7, Oracle 8, Oracle 8i, Oracle 9i, and Oracle 10g, with Oracle Database 11g in beta. For Oracle, the database is a key part of its Fusion applications platform, although it is possible to use other databases with Oracle's business software.

Several versions of the Oracle database are available, with different pricing and features to reflect how it may be used. The Standard Edition contains basic database functionality and is usually used on servers running between one and four processors.

IBM DB2

IBM's DB2 is also a very popular DBMS. IBM now refers to its DB2 database as a "data server" and, like the Oracle database, there are many version of the suite designed for a range of computers, from mainframes to handheld devices.

DB2 version 9, codenamed Viper, is the latest incarnation of IBM's DBMS. IBM offers several licensing arrangements that can allow users to avoid paying for database -features they do not need.

DB2 versions include Workgroup, Workgroup Unlimited, and Enterprise Server Edition. The most sophisticated edition for Linux, Unix and Windows is DB2 Datawarehouse Enterprise Edition (DB2 DWE). This edition is designed for a mixed workload, such as online transaction processing with datawarehousing or business intelligence implementations.

DB2 DWE has several excellent features for business intelligence, such as extraction, transforming or loading, data mining, online analytical processing acceleration, and inline analytics.

Open Source Alternatives

Alongside the big three database products, systems from smaller suppliers have also grown in popularity, differentiating themselves by focusing on niche markets, such as embedded or mobile DBMS.

The Linux operating system is also getting more and more popular among suppliers and users as a database platform, and many users are choosing it over Windows, Unix and mainframe database platforms. Of the open source databases, Ingres, PostgreSQL and MySQL come out the best in Forrester Research's product evaluation, according to senior analyst Noel Yuhanna.

Open source databases from Derby, Firebird and Oracle are also strong performers, while Oracle and MySQL offer strong support for

embedded database platforms.

Another open source database that is becoming increasingly popular is EnterpriseDB, which claims to be able to run Oracle-compatible applications at a lower cost. EnterpriseDB users include Sony Entertainment and Vonage, which have switched from commercial databases.

Measuring Performance

Database performance is measured in several ways, and this can be useful in choosing the right product. The main benchmark is the TPC-C from non-profit body the Transaction Performance Processing Council (TPC), which measures online transaction processing performance.

The aim of the TPC benchmark is to simulate real-world usage. A typical transaction would include updating a database system for such things as inventory control (goods), airline reservations (services), or banking (money).

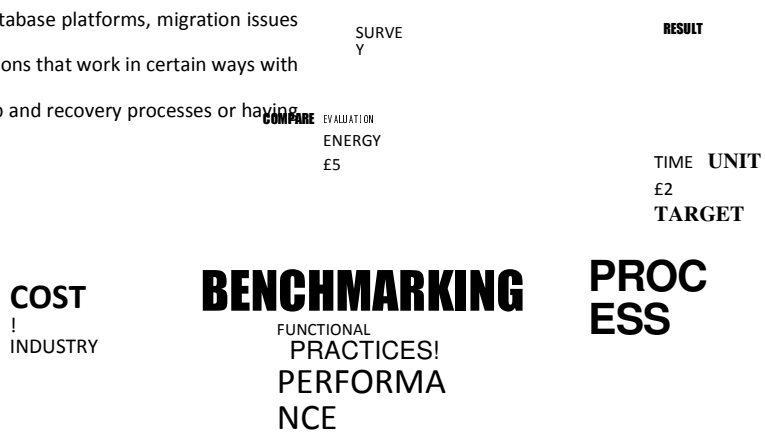
In these environments, a number of customers or service representatives are able to input and manage their transactions via a terminal or desktop computer connected to a database. The TPC would typically produce benchmarks that measure transaction processing and database performance in terms of how many transactions a particular system and database can perform per unit of time.

The current top 10 TPC-C performance results include servers running IBM DB2 9, Oracle Database 10g and Microsoft SQL Server 2005 databases. At the time of writing, the top score was an IBM System p5 595 server running DB2 9 producing 4,033,378 transactions per minute.

Other benchmarks include TPC-R and TPC-H for datawarehouses and decision support systems, and TPC-W for web-based database systems.

Supplier-based benchmarks are also available, and there are several open source benchmarks, such as the Open Source Database Benchmark and RoleDemo, an open source Java framework for benchmarking databases.

Benchmarks can help users choose the best database for their needs, and this is essential because, once chosen, few organisations will ever move off their platform. For users who do switch database platforms, migration issues can come from individual business applications that work in certain ways with specific databases that need certain backup and recovery processes or having different ways of failing over.



The way forward for databases is to have a virtualised master data management layer, which can feed requests into a pool of databases. This means it is not so important where the data resides, and it can be kept on several smaller databases that are arranged in a grid, and which have in-built redundancy and failover capabilities.

Adding another layer of software will have a performance hit, but it could give smaller organisations a more cost-effective

alternative to running a larger, more expensive database and having to back up their data at a datacentre.

Relationship Between Visual Database Tools and Databases

To work with the Visual Database Tools, you have to connect to a database. You can connect to several databases at once. The database management system (DBMS) affects the available tool features

The type of database you connect to can affect the behaviour of the tools, because different commercial DBMS products offer different features. The Visual Database Tools are sensitive to the features that exist in the DBMS you connect to. Thus, some dialogue boxes and property pages will be different, depending on the underlying DBMS. This variance might be as simple as different lists of values in a drop-down list, or it might be as important as the presence or absence of some dialogue-box controls. In addition, the Query and View Designer will construct SQL in the particular dialect of the underlying DBMS to which you are connected.

The tools retain your work in memory

The Visual Database Tools retain local copies of your work before you save it to the database. This means that you can make changes as much as you like without immediately affecting the underlying database. Thus, you can experiment with "what if" scenarios.

After having finished your modifications to any database object, you have three choices:

- You can save the changes, propagating them to the database. You can save database objects such as tables, diagrams, triggers, indexes, stored procedures, user-defined functions.
- You can create a script file that contains the SQL code that was generated by your changes. Later, you or another user can run this script file to modify the database accordingly.
- You can discard your changes. In Query and View Designer, discarding modifications is straightforward; simply close the window without saving your work. But in Database Designer and Table Designer, there are special considerations.

The Visual Database Tools use several strategies to retain local copies of your work. One strategy applies to the work you do with the Database Designer and Table Designer. Another strategy applies to the work you do with the Query and View Designer. The tools support multiuser environments.

You can work with Visual Database Tools in a multiuser environment — an environment in which more than one user can simultaneously connect to and change the database. When you save a modified database object, the Visual Database Tools verify that the object has not been modified since you last saved it. You might need to reconcile your changes with changes that were already made by another user.

Study Unit 2.1: Change Control Policy

US 114044

Demonstrate an understanding of change management for computer systems

Specific Outcome 1	
IBS"	Describe a change control policy for an organisation.
Assessment Criteria	
	1. The description outlines the procedures for change management. 2. The description outlines the documentation required for change management. 3. The description outlines the roles and responsibilities for change management.

2.1.1 Procedures for Change Management

Change Management Program (CMP), often also known as Change Control Process or Change Control Management Process, is a formal process that is used by organisations to ensure that changes to a product or system are introduced in a controlled and co-ordinated manner (as defined by ISO 20000). CMP is not the same as Organisational Change Management (OCM), as OCM manages the impacts of new business processes, which include those stemming from system roll outs and IT initiatives, changes in organisational structure, or cultural changes within an organisation. In short, OCM manages the people side of change.

The purpose of the CMP is to ensure that the negative impact of changes to an organisation's Information Technology system is minimised with the use of a process of governance that is standardised. Some changes are not optional. If, for example, the bar code standard is changing, the organisation must adapt; if a tax withholding structure changes, the organisation must have a change. Nevertheless, all changes of this kind are still subject to governance.

Objectives

The primary objectives of change control are to: -

- manage each change request from initiation through to closure;
- process change requests based upon direction from the appropriate authority;
- communicate the impact of changes to appropriate personnel; and
- allow small changes to be managed with a minimum of overhead

It should never happen that ad-hoc changes are made to the system or to procedures without some oversight. This idea must be initiated by senior management and be passed down, with no exceptions, to everyone in the company. Without backing at the highest level, the CMP is a useless waste of time and money. With proper backing, this program will save the organisation from some very expensive errors.

Process

The following is a general guideline for the change management process. Most changes will require a subset of the steps listed: -

i. Identify (identify and document the required change)

ii. Validate (verify the change is valid and requires management)

iii. Analyse (analyse and record schedule, cost and effort impact of change)

iv. Control (decide whether to execute the change)

v. Action (execute decision, including revision to project plans if necessary)

vi. Close (verify that action is complete and close change request)

Identify Change Request:

Action	*Responsibility*
1. Identify and record the issue (in [location]).	Project Manager or Team Lead

Validate Change Request

Action	*Responsibility*
2. Identify member of the management team as the issue owner. 3. Validate change request with project team members as appropriate. 4. Assess and evaluate change for necessity to project 5.. Update change request with target date for completion of analysis	Project Manager

. Analyse Impact *Action*	*Responsibility*
6. Triage w/ consultation of the Project Sponsor 7. Meet contract requirements for responding to Change Requests. 8. Assign resources to review the impact of the change request. 9. Direct activity to assess the scope, cost and schedule impact of the change. 10. Update change request with impact analysis and estimates in terms of scope, cost, schedule and effort impacts. 11. Update change request with target date for decision.	Project Manager

Control Change Request

Action	*Responsibility*
12. Meet Sigma contract requirements for responding to Change Requests.	Project Manager
13. Determine required approvals and assign priority to the change request. 14. If changes impact scope, budget or schedule place request on agenda for next Steering Committee meeting. 15. If changes do not impact scope, budget or schedule decide whether to proceed with the change.	
16. Review and discuss analysis of change request. 17. Develop recommendation for the Project Sponsor.	Steering Committee
18. Decide whether to proceed with the change. 19. If impact of change cannot be absorbed by [functional owner], schedule discussion with [advisor names].	Project Sponsor
20. Review and discuss analysis of change request. 21. If change request should be escalated to Steering Committee, place request on agenda for next meeting (or email if request is urgent). 22. Develop recommendation for the Steering Committee.	Project Sponsor, [advisor names]
23. Review and discuss analysis of change request. 24. Decide whether to proceed with the change.	Steering Committee
25. Generate approval signature sheets for each outstanding change request. 26. Update status of change request with control decision.	Project Sponsor

Standard Change Request Template:

Change Request

Project: Date:

Change Requestor: Change No:

Change Category (Check all that apply):

□ Schedule □ Cost □ Scope □ Requirements/Deliverables

□ Testing/Quality □ Resources

Does this Change Affect (Check all that apply)?

□ Corrective Action

□ Preventative Action

□ Defect Repair □ Updates

□ Other

Describe the Change Being Requested:
Describe the Reason for the Change:
Describe all Alternatives Considered:
Describe any Technical Changes Required to Implement this Change:
Describe Risks to be Considered for this Change:
Estimate Resources and Costs Needed to Implement this Change:
Describe the Implications to Quality:

Disposition:

□ Approve □ Reject □ Defer

Justification of Approval, Rejection, or Deferral:

Change Board Approval:		
Name	Signature	Date

Action Change Request

Action	*Responsibility*
27. Negotiate contract changes. 28. Execute contract changes.	Project Manager, Project Sponsor, Technology Acquisition Manager
29. Incorporate change request into appropriate plans and work plan. 30. Update work plan baseline for agreed changes.	Project Manager

Close Change Request

Action	*Responsibility*
31. Close change request. 32. Communicate work plan change to project team. 33. Monitor and report progress against project plan. 34. Confirm all updates have been recorded and file all Change Request Documents	

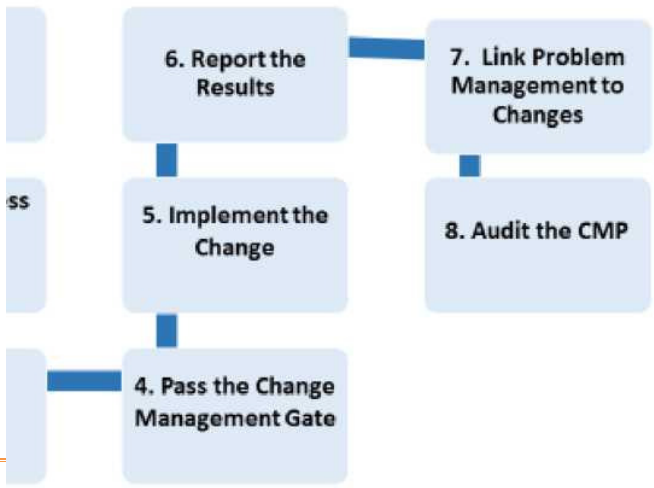
Example of a Change Control Log

MANAGEMENT LOG

	<optional>
	<required>
r Name:	<required>
	<required>

	Current Status	Priority	Change Request Description	Assigned To Owner	Expected Resolution Date	Escalation Required (Y/N)?	Action Steps
	Open	Critical	**EXAMPLE:** Request for product functionality increase			Yes	**EXAMPLE:** Analyze impact of requested cl then meet with the change control board (present findings for final decision on the re change
	Work In Progress	High	**EXAMPLE:** The schedule slipped due to unexpected weather related events			No	**EXAMPLE:** Adjust the schedule to account weather related events
	Closed	Medium					
		Low					

Steps for a Change Management Programme



1: Develop a Request for Change (RFC):

I; Obtain Busine change **Acceptance**

3. initiate the Development Project

1. Develop a Request for Change (RFC): This may be necessary because of problem management where an issue, or a series of related issues, is identified and a mitigating change is necessary to prevent (or minimise) future effects of such issues. The RFC may also come a result of a business decision that will need some modification (add, delete, change) to the supporting technology. An RFC may also be necessary because of outside influences (i.e. governmental regulations or changes made by business partners).

2. Obtain Business Change Acceptance: The decision to make a change is usually a business decision where costs vs. benefits are carefully weighed. Even in situations where the change is necessary strictly to infrastructure (component or system failure) the decision to spend money must be made by the business, not with the IT department. There are instances when procedures are developed in advance to preauthorise changes such as emergency system maintenance, but regardless of the timing of the authorisation, the decision still rests with the business management.

3. Initiate the Development Project: Development of the change (including testing) is an IT-guided function. Where an emergency change must be done (server is down), those functions are usually predetermined. When a new system must be developed, there is a collaborative effort between the business users and the IT team. The process is:

- The systems are designed by IT
- The design is approved by the business partners (users),
- The design is developed by IT
- It is tested by a combination of IT and the users
- The final product is approved by both

Careful attention must be given to ancillary effects the new change may have on existing systems

4. Pass the Change Management Gate: The Change Advisory Board (CAB) reviews all changes before they can be put into production. The CAB will usually consist of a group of people with different perspectives, backgrounds and areas of expertise. Their function is to review the change from a process and governance point of view to ensure that all foreseeable risks have been identified and mitigated, and that there are compensatory techniques in place for any elements of exposure (things that could go wrong). The development team and the change sponsor will present the change to the CAB. The focus will be to evaluation of risk. Elements on which the CAB is must focus are:

- Implementation strategies
- Communication to affected stakeholders
- Backup plans
- Post-implementation monitoring

The CAB is not responsible to determine if the change is appropriate - that decision has already been made. The CAB is also not responsible to determine if the change is cost effective. Again, that is strictly a business decision.

5. Implement the Change: If the CAB does not approve the change, the reasons are listed. Such reasons are that certain risks have not been mitigated or communications have not been planned. The development team will then be given time to fix those issues and reschedule a meeting before the CAB. If the change is approved, the implementation can now be scheduled. The CAB is not usually represented at implementation, although it is possible that some members of the CAB have expertise that is necessary

during the implementation. In such a case they will not be present as official CAB representatives, but rather as subject matter experts (SME). How the change is implemented, the checklist and steps, are predefined and were presented to and approved by the CAB. The entire process must be thoroughly documented and the approved process must be precisely followed.

6. Report the Results: The following are possible results:

- The change was implemented successfully with no issues
- The change was implemented with issues that were corrected during implementation
- The change was implemented with issues that were deemed acceptable
- Issues arose that were unacceptable and the change was rolled back
- In the worst case the change was implemented with unacceptable issues and could not be rolled back.

Whatever the result, it must be documented and returned to the CAB. The CAB must then distribute that information to the stakeholders and store and maintain the results in the Change Management system (that may either be an automated database or a paper filing system, but the documents must be maintained for audit purposes).

7. Link Problem Management to Changes: Issues that arise should be compared to the CAB documentation of changes in order to isolate any unanticipated adverse effects of a change. It is often happens that undesirable effects of a change are not noticed immediately, but are identified when problems emerge in ancillary systems. For example, the addition of several fields to a database might not have a direct negative effect on the users but could impact network performance that would be apparent to other users who are not directly involved with the modified system.

8. Audit the CMP: At least once each year an audit of the CMP should be conducted to assure that all change documentation is maintained and available. Every change approval document should be examined to assure that the proper signatures are in place and that the results of the implementation are properly documented.

2.1.2 Documentation Required for Change Management.

Change Management Plan <u>Purpose</u>

The Change Management Plan establishes how changes will be proposed, accepted, monitored, and controlled. The change control procedures that are identified in the Change Management Plan will govern changes to the baseline project scope. This includes changes to the work breakdown structure and requirements from project inception through to completion. The change control procedures will also govern changes to the baseline schedule and cost. This Change Management Plan addresses the following activities:

- Identification and inventory of change requests
- Analysis and documentation of the complete impact of requested changes
- Approval or rejection of change requests
- Tracking changes and updating of project documentation to account for approved changes

Procedures for Change Identification

In this section the procedures to be used to identify and document change requests to project baselines (e.g., approved scope baseline, cost baseline, and schedule baseline) must be described. It must also describe who is authorised to submit change requests.

Any project team member can submit a change request to the Project Manager. When he/she identifies the need for a change to the approved baseline, the change will be clearly defined using the Change Request Form.

- The Requestor completes Section 1 of the Change Request Form and submits it to the Project Manager for review.
- The Project Manager records the request in the Change Control Log and assigns a change request number to the change request.

Procedures for Change Analysis

In this section the procedures to be used to analyse and assess the impact of the proposed change are described. The change should be assessed with regards to the project's baseline scope, schedule and total project costs.

Also, describe the process that will be used by the project team to evaluate and approve a proposed change to be forwarded to the Project Sponsor, Executive Sponsor, and/or the Change Control Board (CCB) for final review and approval. A CCB is a group of individuals assigned to control identified project changes, review impacts, and grant approvals or rejections of proposed changes. The CCB is made up of project stakeholders or their representatives. In many projects, the Steering Committee functions as the CCB.

To fully evaluate and accept or reject the change request, the project team should make sure that the approver(s) are aware of the impact that the change will have on the project. The main interest of the Project Manager and the approver(s) will be in the impact of change to the project scope, schedule, cost, quality, and risk.

- The Project Manager will assign a project team member to complete Section 2 of the Change Request Form, which will provide comprehensive details about the work to complete the change and the impact of the change to the project and deliverables.
- The Project Manager will determine if the request is viable and decide whether it is necessary for the request to be considered by the Project Sponsor, Executive Sponsor, and/or CCB.

Change Request Approval Process

This section documents the process to be used to evaluate, approve, and communicate changes to the project scope, schedule, and cost baseline. Within this section, it is important to indicate clearly what members of the project team are needed to assess changes, maintain change records, present proposed changes to management, and review and approve or reject changes.

- When the impact of the change has been recorded, the Project Manager must send the Change Request Form to the Project Sponsor, Executive Sponsor, and/or CCB for acceptance or rejection.
- The Project Sponsor, Executive Sponsor, and/or CCB review the change request and indicate the decisions that were made by completing Section 3 of the Change Request Form and returning it to the Project Manager.
- If approved, the Project Manager will update the appropriate project documentation to reflect the change. For example, if the scope is changed, the Project Scope Statement should reflect the updated scope. Corresponding contract modifications may also be necessary if the approved change impacts the contractual scope, schedule, costs, or other terms.
- If rejected, the Project Manager will update the Change Control Log.

Change Tracking

Project Change Request Form

Project Information

This section describes the process to maintain a master log of all changes that were submitted, approved, and rejected for the project, so that clear traceability of all proposed changes is evident.

- The Project Manager must maintain a master log of all change requests and the resolution of each request. All requests must be maintained in a Change Control Log. A sample Change Control Log is included.
- For approved changes, the Project Manager must complete Section 4 of the Change Request Form to indicate that project document updates have been completed and must file the form with other project artefacts.

The following are templates that can be used:

Project Title: Project Number:

Project Manager:

Section 1: Change Request

Requestor Name: Requestor Phone:	Date of Request:	Change Request Number: *Supplied by (PM)*
Item to be Changed:		Priority:

Description of Change:

Estimated Cost & Time:

Section 2: Change Evaluation	
Evaluated by:	Work Required:
What is Affected:	
Impact to Cost, Schedule, Scope, Quality, and Risk:	

Section 3: Change Resolution			
Accepted Rejected	Approved by (Print):	Signature:	Date:

Comments:

Section 4: Change Tracking			
Completion Date	Completed by (Print):	Signature:	Date:

My signature above indicates that the project documentation has been updated to accurately and comprehensively reflect the approved changes.

Problem Report Form

Project / System	Give System Name	Problem Reference Number
Location of Problem	Give Module, Screen or Report name	System Version Number
Environment Details	Give details such as Operating System / Machine / Office / City / Country	
Problem Details Include indication of importance & any Business Deadlines Critical □ High □ Medium □ Low □	Give details such as : Specification that defines how the system should operate, Step-by-step description of what happened and how to reproduce the problem, Location of any supporting evidence (eg report, screen print, log file). Tick if Continued Overleaf □^	
Problem Date	Give Date Problem Raised	
Person Raising Problem	Give Name, Organisation, and Contact Number	
Investigator of Problem	Print Name	
Investigation Outcome	Suggested Action and details of other items affected	
Suggested Priority Critical / High / Medium / Low	Give Schedule for Resolution	
Release Number		
Signoff	Signature	Date
Reviewer		
Project Manager		

Release Note

Section A	
Project / System	System Name
Problem Reference Numbers	List all problem reference numbers resolved by this Release
Changes in Release	Describe changes in this release
Items Being Changed	List ALL items being changed Tick if Continued Overleaf □^
Installation Instructions Give as much detail as possible or cross reference relevant documents Tick if Continued	
Section B	
Attachments □ Source Code □ Build Scripts □ Executables □ Data □ Documentation Updates	
Release Reference	Give reference number and / or date for release to customer
Signoff	Signature
Releaser	To indicate release is complete
Project Manager	To indicate release has been fully reviewed

Change Control Form

Section A		
Project		Change Number
Controlled Item		Item Version
Identification of Aspect to be Change	For Document give section number / page number For Software give Module, Screen or Report name	
Change Details Include indication of importance and urgency	Tick if Continued Overleaf □^	
Requester of Change Print Name		Date Raised
Section B		
Investigator of Change		
Impact, give details of other items affected		
Investigation Outcome Reject / Action at No Cost / Action at Cost	Suggested Priority High / Medium / Low	Date Investigated
Section C		
Implementer		Date Scheduled
Section D		
Change Implemented	Signature	Date
Implementation		
Project Manager		

Document Control

Title:	Change Control Procedure
Issue:	Issue 1
Date:	17 January 2001
Author:	Mark Pillatt
Distribution:	EC DG Enterprise - Gavino Murgia Project Team
Reference:	IDA-MS-CCP
Filename:	IDA-MS-CCP-i1
Control:	Reissue as complete document only

Document Signoff

Nature of Signoff	Person	Signature	Date	Role
Author	Mark Pillatt			Senior Consultant
Reviewer	John Brinkworth			Project Controller

Document Change Record

Date	Version	Author	Change Details
08 December 2000	Issue 1 Draft 1	Mark Pillatt	Initial Draft
14 December 2000	Issue 1 Draft 2	Mark Pillatt	Update following John Brinkworth review comments
15 January 2001	Issue 1 Draft 3	Sue Turner	Re-formatting
17 January 2001	Issue 1	Mark Pillatt	Add Introduction and issue

Change Control Log

Project Information	
Project Title:	Project Number:
Project Manager:	

Change Number	Description of Change	Priority	Date Requested	Requested By	Status (Evaluating, Pending, Approved, Rejected)	Date Resolved	Resolution/Comments

2.1.3 Roles and Responsibilities for Change Management **Change**

Review Board

The Change Review Board (CRB) is a team of people made up of IT management and subject matter experts, members of the user community, vendors and outside consultants. The chairperson is the Change Co-ordinator. The task of the Change Review Board is to plan and monitor all changes that are introduced into the IT environment.

For every change, the CRB must ensure that:

- The Change Request has been submitted together with the required information
- There is a business reason for the change
- A user has been identified who is totally accountable for the change
- There is a viable Change Plan
- There is a viable Recovery Plan
- There is a viable User Acceptance Plan
- A Technical Impact Assessment has been done
- A Business Impact Assessment has been done
- There is a developed Communications Plan to ensure that communication is sent to the IT and user communities of the intended change and the potential impact to them in case of failure of the implementation
- Appropriate management approval has been obtained based on the risk assessment category
- There will be a post installation review of the completed change to ensure proper and successful implementation

Change Co-Ordinator

The Change Co-ordinator is the user's primary interface into the change process and represents the interests and the IT requirement of the user to meet service commitments.

The Change Co-Ordinator must:

- Own, maintain and ensure accuracy and timeliness of the consolidated change schedule
- Make sure that all changes are planned and communicated appropriately
- Review change requests for procedural compliance, information quality and completeness
- Ensure that accurate priority and impact are assigned to change requests
- Co-ordinate and own the change approval and rejection process; this incorporates routing to reviewers, receiving reviewer responses and relaying the correct information to requesters, and also includes negotiation with both parties and final ruling
- Undertake post change reviews and is responsible for the sign off mechanism
- Schedule and attend all meetings regarding the Change Management process
- Ensure that asset and configuration updates are only permanently applied after successful and signed- off changes
- Participate in project meetings with applications development teams and the business where change impact analysis, implementation and scheduling of large changes are to be discussed
- Take responsibility for owning efficient and quality problem resolution, where Change Management service expertise is required
- Take responsibility for the escalation and exception report to the Change Management Process Owner
- Comply with Change Management service standards, process and procedures as required
- Provide input to Change Management service improvement
- Provide and take responsibility for the accuracy of appropriate Change Management service input to knowledge bases as a result of changes and problem resolution
- Take responsibility for contract management and day to day maintenance management of third party technology suppliers that result from Change Management issues with the service
- Initiate and be a technical resource to the IMAC service where Change Management service expertise is required
- Provide a technical resource to Project Requests where Change Management service expertise requirement forms an element of the project
- Capture and report correct Change Management service measurement data
- Attend appropriate problem escalation / resolution, project development and service support reviews, where Change Management service expertise is required

Change Requester

The Change Requester is responsible to initiate and complete the Change Management process and has overall responsibility to accept changes. Standards dictate who can act as a change requester.

The Change Requester is responsible to

- Own individual requested changes and is ultimately responsible for the success of the change
- Review change requests with the department manager for approval
- Assess the change for risk/impact and make sure that the appropriate risk category has been applied
- Ensure that the change request is complete, and that accurate information at a sufficient level of detail is provided to implement the change without intervention
- Ensure that proper lead-time for changes is allowed
- Conduct User Acceptance testing of change
- Convene the Level 1 Expedite meeting as required
- Communicate intention of change to all parties that may potentially be affected
- Notify the Change Co-ordinator of any change in date or time of implementation before the

target date is reached.

- Initiate the Change Control service by completing and distributing a completed change request
- Take responsibility to obtain appropriate resource for all change tasks requiring completion for change success
- Co-ordinate task documentation within a change request with other participating staff as appropriate
- Ensure that pre and co-requisites for the change have been considered and are completed
- Ensure that the reviewer distribution list is as complete as possible
- Attend change assessment, scheduling and review meetings, as necessary
- Take responsibility for accurate representation of priority, impact and change window / time requirements
- Ensure that all owned rejected changes are placed into a 'fit state' for re-submission
- Co-ordinate all implementation tasks where necessary and is responsible for any backout decisions for owned changes in conjunction with Change Control
- Update owned change requests as required and requested
- Take responsibility to undertake change tasks within the change implementation as required
- Participate in change task co-ordination meetings as part of change implementation planning

Change Implementer

The Change Management Implementer has overall responsibility to understand the requested change; to document its implications on both the business and the IT environments; to create, with other resources as required, the coding, system, procedure and/or process modifications necessary to implement the change; to represent the change to the Change Review Board together with the Change Requester; to monitor and/or test the code before the final promotion into production and requesting change closure by the Review Board.

Responsibilities of the Change Implementer include to:

- Monitor the Change Assignment process for assigned Changes
- Meet with the Change Requester to understand the requested change and to complete the Change Request documentation
- Assist the Change Requester in the preparation of a business cases for the change as necessary
- Present the Change to the Change Review Board together with the Change Requester, from a technology and IT architecture perspective for approval to proceed
- Develop technology and operations impact statements
- Develop backup and/or back out plans
- Identify and assemble the team required to create the change
- Design and create the code, procedures or process modifications required to effect the change
- Design and develop the tests that are required to demonstrate the quality and usefulness of the change
- Present the completed change, together with the Change Requester, to the Change Review Board for approval to promote into production
- Monitor the promotion (on site or remotely) of the change into production
- Resolve issues associated with promoting the change into production, where possible
- Participate in the review of the promotion of the change together with the Change Requester
- Request closure of the successful change from the Board

Change Management Process Owner

The Change Management Process Owner has overall responsibility to ensure the quality of the Change Management process.

Responsibilities of the Change Management Owner are:

- Take responsibility for and own the Change Management service
- Take responsibility for the development and implementation of Change Management mission and strategy, in line with the organisation and IT strategies
- Escalate exceptions to senior management as appropriate
- Take ultimate responsibility for resolving Change Management service dissatisfaction
- Ensure compliance with Change Management process standards and procedures
- Have a nominated deputy to cover for service owner absence
- Sponsor and/or manage internal improvement projects to implement new technology and process improvement
- Communicate Change Management service procedures and working practices and changes to internal standards, processes, procedures and technology
- Co-ordinates and set annual service requirements, objectives and targets for the Change Management service in conjunction with other IT Process Owners
- Finalise annual service requirements, objectives and targets with the Service Request, Call Center and Problem Management Process Owners
- Attend appropriate senior management level service support and development reviews
- Be involved in development of and subsequent agreement on service level targets and target improvements related to the Change Management service
- Develop requirements for Change Management standards, procedures, measurements, tools and technology in conjunction with other IT Process Owners
- Approve and sponsor Change Management improvement ideas

Service Center Administrator

The Service Center Administrator has overall responsibility to monitor and configure the Change and Problem Management processes.

Responsibilities of the Change Management are to:

- Develop requirements for Change Management standards, procedures, measurements, tools and technology in conjunction with other IT Process Owners
- Manage the tools and procedures that support Change Management
- Identify process, procedure and tool improvements that may benefit the organisation to the Change Review Board
- Make changes to the processes, procedures and tools that support the Change Management process as directed by the Change Review Board
- Escalate to senior management issues as appropriate
- Ensure compliance with Change Management process standards and procedures
- Communicate changes to Change Management internal standards, processes, procedures and technology
- Attend appropriate senior management level service support and development reviews

Change Controller

The Change Controller is responsible to schedule and oversee all changes that are introduced into the IT production environments and service delivery infrastructure to ensure that there is minimised risk of impacting the availability of services.

Responsibilities are to

- Monitor Change Categories and Service Risks
- Ensure that all Information Technology customer and Company interests are protected
- Build and maintain the Consolidated Change Schedule:
- Integrate new changes into the existing change schedule
- Identify conflicts in the schedule, and negotiate adjustments with the relevant parties
- Notify affected parties that changes have been scheduled and are ready for implementation
- Monitor the implementation of changes
- Handle schedule slips and escalate the appropriate parties to recover the schedule
- Identify and resolve change assignment issues
- Manage change approval
- Facilitate the Change Review Board meetings
- Manage exceptions of rejected records
- Resolve day-to-day change co-ordination actions
- Accept and manage external change input
- Monitor regular change control measurements
- Create, co-ordinate, consolidate, and monitor the change schedule.

Change Scheduler

The Change Scheduler is a delegate of the Change Controller, and is responsible to create, co-ordinate, consolidate, and monitor change schedules. He/she

- Performs duties delegated by the Change Controller
- Represents the Change Controller on initial issues relating to the Change Schedule
- Is responsible to create, co-ordinate, consolidate, and monitor change schedules

Study Unit 2.2: The Implementation of Changes to Computer Systems

US 114044

Demonstrate an understanding of change management for computer systems Specific Outcome 1

Produce guidelines for the implementation of changes to computer systems.

Assessment Criteria

The guidelines are
1. consistent with the change control policy for the organisation.

The guidelines include
2. documentation of client change requirements.

The guidelines outline a
3. procedure for managing the different versions of systems.

ASSESSMENT CRITERION RANGE Development, implementation, operation.

4. The guidelines outline procedures for managing changes.

ASSESSMENT CRITERION RANGE Logging, follow-up, etc.

5.

The guidelines outline
6. methods of maintaining accuracy and consistency in the change process.

The guidelines ensure the
7. integrity of the client/user database.

The guidelines indicate
8. others who need to be on the document circulation list. The guidelines include procedures for

education client/user on the change management process.

2.2.1 Change Control Policy for the Organisation

As seen in Study Unit 2.1, change management refers to a formal process for making changes to IT services. The goal of change management is to increase awareness and understanding of proposed changes across an organisation and ensure that all changes are made after careful consideration and in a way that minimise negative impact to services and customers.

Change management usually includes the following steps:

- Planning: Plan the change, including the implementation design, scheduling, communication plan, testing plan and roll-back plan.
- Evaluation: Evaluate the change, including determining the priority level of the service and the risk of the proposed change; determine the change type and the change process to use.
- Review: Review Change Plan with peers and/or Change Advisory Board as appropriate to the change type.
- Approval: Obtain approval of the Change Plan by management as needed.
- Communication: Communicate about changes with the appropriate parties.
- Implementation: Implement the change.
- Documentation: Document the change and any review and approval information.
- Post-change review: Review the change with an eye to future improvements.

Change Management Policy

Each organisation must have a Change Management Policy. The purpose of a Change Management Policy is to establish management direction and high-level objectives for change management and control. This policy will ensure that the change management and control strategies are implemented in a way that it mitigates associated risks such as:

- Information being corrupted and/or destroyed
- Computer performance being disrupted and/or degraded
- Productivity losses being incurred
- Exposure to reputational risk

According to ISO 27001, the following sections should form part of the Change Policy:

1 Introduction

2 Scope

3 Purpose

4 References and definitions

4.1 Normative references

4.2 Definitions and abbreviations

4.2.1 Audit trail

4.2.2 Information resources

4.2.3 Abbreviations

5.1 Preamble

5.1.2 Operational Procedures

5.1.3 Documented Change

5.1.4 Risk Management

5.1.5 Change Classification

5.1.6 Testing

5.1.7 Changes affecting SLA's

5.1.8 Version control

5.1.9 Approval

5.1.10 Communicating changes

5.1.11 Implementation

5.1.12 Fall back

5.1.13 Documentation

5.1.14 Business Continuity Plans (BCP)

5.1.15 Emergency Changes

5.1.16 Change Monitoring

6 Roles and Responsibilities

7 Compliance

8 IT Governance Value statement

9 Policy Access Considerations

Each organisation will define its own policy for managing change. A good change control policy should include the following topics: -

i. Will it be a single or multi-tiered approach for change approval?
ii. Set out the responsibilities of the Project Manager to accept or reject a change.
iii. Set out the escalation process for changes that impact scope, budget, project risk or schedule.
iv. Set out the responsibilities of the Project Sponsor on any decision requests that may result in a significant change in scope, schedule, and budget, i.e. the impact of the change cannot be covered by resources. This group will advise the Steering Committee.
v. Who is responsible for making the final decision?

2.2.2 Documentation of Client Change Requirements

The client initiates change by making a formal request for something to be changed. The change control team then records and categorises that request. This categorisation would include estimates of importance, impact, and complexity.

Change Control for Validated Systems

Organisations need to very clearly define their processes for evaluating changes to validated systems. There should be a well-defined, multidisciplinary approach to considering the effects from proposed changes. Some changes, such as adding a data field to a form or report may be a small change; other changes, such as changing how a program stores and organises data can be quite extensive. Before changes are implemented, it is important for organisations to document the expected outcomes of the changes and have an established plan to implement and test the change and update any existing validation documentation. Part of defining the process for evaluating change control should include the requirements for implementing minor, major and critical changes. In this way the organisation is able to focus proportionate validation resources to the change effort.

An excellent way to determine the extent of revalidation is Risk Assessment. By reviewing the original validation requirements, and evaluating the new risks that are introduced through the changes to the system, the Risk Assessment process can help determine which sections of the system will need re-testing. If the risk assessment determines that the change is small or does not affect the requirements of the system, only limited testing, focused on the affected system object would be necessary to demonstrate that the system has maintained its validated state. Major changes will ask for additional re-validation and critical changes could trigger and entire re-validation of a system.

Documentation

The following documents should be assessed for potential impact due to the change, and updates should be planned, where required:

• Validation package including user requirements specification (URS), technical requirements specification (TRS), TM, design qualification (DQ), installation qualification (IQ), operational qualification (OQ), performance qualification (PQ), and validation plan and report
- Design documentation
- Procedures for using and maintaining the system.

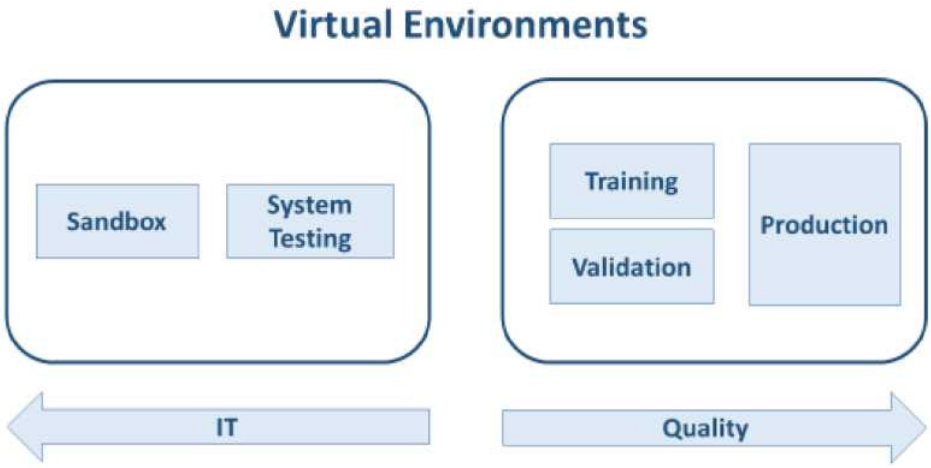
In some cases (e.g., for large or complex changes, or due to cumulative change over time), it may be necessary for a complete rewrite of certain affected documents because of addenda or point revisions.

2.2.3 Procedure for Managing Different Versions of Systems

Executing the Change

In the execution phase, the change is actually made in a staging environment so it can be tested before the implementation of the production. The change (and other aspects of the system that may have been affected) is tested to ensure that the system is accurate, reliable, and consistent to the intended performance. The testing must be documented, and the results should either lead to corrections and additional testing, or confirm that the end result after the change is what was intended. All documentation regarding the change should also be completed.

Changes should initially be implemented separate from the production environment of the validated system. This will ensure that no changes are made to the production environment until you are sure that they are fully qualified and functioning as expected. In computer systems, it is preferred to have more than one virtual environments defined in the architecture landscape. Typical environments are as follows:



- Development environment (sometimes referred to as "Sandbox")—a virtual environment where experimental coding/configuration takes place, as the developer/configurator is trying different solutions, doing preliminary unit testing, etc.
- System testing—a virtual environment that is used for preliminary systems testing conducted by IT
- Validation—a virtual environment that is frozen and represents production, set up for validation testing, and controlled as unchangeable throughout validation testing
- Training (not always used by all companies for all systems)—a virtual environment used for hands-on training on the new or revised system
- Production—the live business environment or "instance" of the system.

Testing should verify that:

- The new system performs as expected after the changes were made
- The new system's original functionality continues to work after the changes were made
- New changes do not introduce errors that keep the system away from performing as intended.

Version Control

ISO 27001 states that *Any software change and/or update shall be controlled with version control. Older versions shall be retained in accordance with corporate retention and storage management policies.*

A version control system (also known as a Revision Control System) is a repository of files, often the files for the source code of computer programs, with access that is monitored. Every change that is made to the source is tracked, along with who made the change, why they made it, and references to problems fixed, or enhancements introduced, by the change.

Version control systems are vital for any form of distributed, collaborative development. Whether it is the history of a wiki page or large software development project the difference between a well-managed and controlled process and an uncontrolled 'first come, first served' system is made by the ability to track each change as it was made, and to reverse changes when necessary. It can also serve as a mechanism for due diligence for software projects.

Version Tracking

It may be important for developers to compare today's version of some software with yesterday's version or last year's version. Since version control systems keep track of every version of the software, this is then an easy, straightforward task. Knowing the what, who, and when of changes will help to compare the performance of specific versions, working out when bugs were introduced (or fixed), and so on. Any problems that arose from a change can then be followed up by finding out who made the change and the reasons they gave for making the change.

Co-ordinating Teams

Resource development is usually carried out by teams, either co-located or distributed. Version control is important to co-ordinate teams of contributors. It allows one contributor to work on a copy of the resources and then release their changes back to the common core when ready. Other contributors work on their own copies of the same resources at the same time, and they are unaffected by each other's changes until they decide to merge or commit their changes back to the project. Any conflicts that arise - when two contributors independently change the same part of a resource - are automatically flagged when the changes are merged. Such conflicts can then be managed by the contributors.

Typically, in open source projects, version control systems allow anyone to read and copy the project resources, but only authenticated users, known as committers, may update source code in the repository.

2.2.4 Procedures for Managing Changes

The following is a general guideline for the change management process. Most changes will need to have a subset of the steps listed: -

i. Identify (identify and document the change that is required)
ii. Validate (verify the change is valid and requires management)
iii. Analyse (analyse and record schedule, cost and effort impact of change)
iv. Control (decide whether to execute the change)
v. Action (execute decision, including revision to project plans if necessary)
vii. Close (verify that action is complete and close change request)

Ad-hoc changes should never be made to the system or to procedures without some oversight. This idea must start with senior management and be passed down, with no exceptions, to everyone in the company. The CMP is a useless waste of time and money without it being backedat the highest level. With proper backing, this program will save an organisation from some very costly errors.

Best practice indicates that the following procedure should be used when managing change:-

1. Develop a Request for Change (RFC): The RFC could originate from problem management, where an issue, or a series of related issues, are identified and it is necessary to make a mitigating change to prevent (or minimise) future effects.

   The RFC could also originate because of a business decision that will need some modification (add, delete, change) to the supporting technology or some outside influences (i.e. governmental regulations or changes made by business partners).

2. Obtain Business Change Acceptance: The decision to make a change is usually a business decision where costs are weighed against benefits. It it important to note that even in situations where the change is strictly aimed at infrastructure (component or system failure) the decision to spend money must be made by the business, not by the IT department.

   There are occasions when procedures are developed in advance to preauthorise changes such as emergency system maintenance, but regardless of the timing of the authorisation, it must be remembered that the decision still rests with the business management.

3. Log the RFC: In most organisations RFC's are managed by using either a Change Management Log (Paper filing system) or Change Management Database. Information that is usually needed would be: -
   a. Change details (unique number, application, date, change requestor)
   b. Change category (scheduled, emergency, cost, scope, requirement deliverables, testing quality and required resources)
   c. Impact (corrective action, preventive action, defect, repair or update)
   d. Change description
   e. Reason for the change
   f. Alternatives considered
   g. Technical requirements
   h. Description of the perceived risks
   i. Required resources
   j. Disposition (accept, reject or defer)
   k. Justification of acceptance, rejection or deferral

4. Initiate the Development Project: Development of the change (including testing) is an IT-guided function. In the event of an emergency change (server is down) those functions are typically predetermined.

   When it is necessary to develop a new system, there must be collaborativon between the business users and the IT team. The systems are designed by IT, the design is approved by the business partners (users), developed by IT, tested by a combination of IT and the users.

   Finally, the final product is approved by both. Careful attention must be given to ancillary effects that the new change may have on existing systems.

5. Pass the Change Management Gate: It is the task of the Change Advisory Board (CAB) to review all changes before they can be considered for production.

   The CAB will usually consist of a group of people with different perspectives, backgrounds and areas of expertise. Their function is to review the change from a process and governance point of view to assure that they have identified and mitigated all foreseeable risks, and that there are compensatory techniques in place for any elements of exposure (things that could go wrong). The development team and the change sponsor will present the change to the CAB, where the focus will be on the evaluation of risk.

   Elements on which the CAB is required to focus are implementation strategies, communication to affected stakeholders, backup plans and post-implementation monitoring.

   It is important to remember that the CAB is not responsible to determine if the change is appropriate - that decision should have already been made. The CAB is also not responsible to determine if the change is cost effective. Again, this is strictly a business decision.

6. Implement the Change: If the CAB does not approve the change, the reasons are listed. This happens when certain risks have not been mitigated or communications have not been planned. In such a case the development team will be given time to fix those issues and reschedule a meeting before the CAB.

   If the change is approved, the implementation is scheduled. The CAB is not normally represented at implementation although it is possible that some members of the CAB have expertise that is necessary during the implementation. In such cases they will be present as subject matter experts (SME), rahter than official CAB representatives.

   How the change is implemented (e.g. checklist and steps) have been predefined and were presented to, and approved by, the CAB.

   The entire process must be thoroughly documented and the approved process must be precisely followed

7. Report the Results: There can only be two outcomes of the implementation process. Either the change was implemented:
   a. Successfully with no issues.
   b. The change was implemented with issues that were corrected during implementation.
   c. The change was implemented with issues that were deemed acceptable.

      Or

   a. Issues arose that were unacceptable and the change was rolled back.
   b. In the worst case the change was implemented with unacceptable issues and could not be rolled back.

   Whatever the result, it must be documented and returned to the CAB, who is then responsible to distribute that information to the stakeholders. The CAB must also store and maintain those results in the Change Management system (that may either be an automated database or a paper filing system). It is important to maintain the documents for audit purposes.

8. Link Problem Management to Changes: Issues that arise should be compared to the CAB documentation of changes so that any unanticipated adverse effects of a change can be isolated.

   It often happens that undesirable effects of a change are not noticed immediately, but are identified by the emergence of

problems in ancillary systems. For example, when several fields are added to a database, it might not have a direct negative effect on the users but could have an impact ont network performance apparent to other users who are not directly involved with the modified system.

9. Periodically Audit the CMP: No process in an organisation should be considered to be cast in iron. It is good practice to, at least once each year, conduct an audit of the CMP. This is done to ensure that:

a. All change documentation is maintained and available.

Every change approval document must be examined to assure that the correct signatures are in place and that the results of the implementation are documented properly.

b. The steps in the CMP are still appropriate. In many cases, processes in their original form have become redundant because of changes in technology or best practice standards.

Tips:

- Make sure that standard periodic maintenance is preapproved. If it is a normal process to reboot a server on Sunday morning at 2:00 AM, it is not necessary to submit an RFC each time, but that process must be approved in advance.
- Ad-hoc maintenance must adhere to the CMP. It is good to include items like testing the fire suppression systems, cleaning sub-flooring in the data centre, HVAC inspection and testing and even pest control maintenance in maintenance plans. Some organisations require an RFC if a light bulb is changed in the data centre (the ladder fell and damaged the network).
- Procedures should also be subject to Change Management. If there is a change in system backup scheduling, that must go through Change Management. Every change of any kind (system or procedure) should be analysed to determine if there is any possible risk.

Warnings:

- Rotate CAB members regularly. To always have the same members can lead to favouritism, and it can lead to burnout. It is important for the CAB to be fresh, pay attention, and not be subject to outside political influences. It may be a good idea to create a 'squad' of CAB members that can be scheduled to attend meetings at different times.
- Politics can often get in the way of the CAB. "This change is required" may be true, but it could also be a personal agenda from one of the executives. The CAB must have ultimate authority to make decisions on implementation.

2.2.5 Methods to Maintain Accuracy and Consistency

One of the key responsibilities in project management is to ensure that, when implementing maintenance changes, the integrity of both the application files and the databases remains intact.

Even though application back-up and recovery forms part of disaster recovery strategies, the process of performing application backups (database and application files) must form part of the project plan.

The ability to restore from valid backups is a vital part of ensuring business continuity.

One of the key responsibilities in project management is to prepare for the possibility of media, hardware and software failure. Should any of these failures occur, the major objective is to ensure that the applications are available to users within an acceptable time period, while at the same time ensuring that there is no loss of data. Project Managers should evaluate

their preparedness to respond effectively to such situations by answering the following questions:

- How confident are they that the data on which the company business depends, has been backed up successfully and that the data can be recovered from these backups within SLA time limits?

  Have measures been taken to draft and test the procedures in order to protect, as well as recover, the application from numerous types of failures?

2.2.6 The Integrity of the Client/User Database

Data integrity refers to maintaining and assuring the accuracy and consistency of data during the change process. It is also a critical aspect to the design, implementation and usage of any system which stores, processes, or retrieves data.

The term data integrity is broad in scope and may have widely different meanings depending on the specific context

Data integrity is the opposite of data corruption, which is a form of data loss. The overall intent of any data integrity technique is the same:

i. Ensure data is recorded exactly as intended (such as a database correctly rejecting mutually exclusive possibilities,) and upon later retrieval, ensure the data is the same as it was when it was originally recorded.

ii. In short, data integrity aims to prevent unintentional changes to information.

Data integrity is not to be confused with data security which is the discipline of protecting data from unauthorized parties.

If the changes are the result of unauthorized access, it may also be a failure of data security.

Any unintended changes to data as the result of storage, retrieval or processing operation is failure of data integrity. This includes malicious intent, unexpected hardware failure and human error.

Backing-up the data:

The following is a checklist for application backup procedures: -

i. Develop a comprehensive backup plan.
ii. Perform effective backup management.
iii. Perform periodic application restore testing.
iv. Have backup and recovery SLA been drafted and communicated to all stakeholders.
v. Has the recovery plan been drafted and documented?

Comprehensive Backup Plan

i. Decide what needs to be backed up. All information relating to the back-up requirements needs to be communicated to whoever is responsible for performing the procedure.
ii. Determine the appropriate backup type to use for your data.

Establish a strategy for handling backups:

It is common practice for an organization to have a global policy for maintaining the integrity of their data. The project manager is also responsible for developing a project related strategy using project risk as the main contributor to identify which tasks need to be performed

Establish an appropriate backup schedule and window:

The backup schedule will form part of the implementation plan created for the project.

Decide where to store backups:

Back-up data can be stored on a variety of storage medium, e.g. disk, cd or tape.

Develop a backup retention policy:

Project related back-up will probably be kept for about 24 hours. The retention period for the global policy will depend on the nature of the business e.g. Financial sector will have a 30-day retention where SME companies will range from 3-days to 10-fays

Restoring the data:

Backups are of no use if the IT team cannot restore the data to the system at the time of need.

The following is a checklist for application recovery procedures: -

1. Databases restore testing—There should be a requirement to test application restores irrespective of the store medium
2. Validating restores where possible. Facilities are available to validate and verify backups without doing actual restores. These utilities will do everything except actually restore the database.

   This is the best method to determine if the backup is good and usable before being in a situation in which it becomes critical.
3. Refreshing non-production data from production backups. It is good practice to periodically build nonproduction data from production backups using appropriate backup/restore utility commands as a restore practice.
4. Performing annual/biannual restore testing as part of audit. The restore process will have to be explained through a narrative, preserve logs and screenshots to show this type of restore testing.
5. Actual restores: When testing actual restores, data also needs to be backed-up prior to commencing the restore. Each restore scenario will be different so, depending on the type of loss and backups available, decisions will have to be taken on the type of restore (complete (point-in-time) or incomplete recovery). An incomplete recovery can be time- based, cancel-based or change-based.

   Finally, the backup and recovery strategy should be able to maintain data security while at the same time minimize storage needs, backup times and recovery times.

One needs to consider these seven steps when building a sound backup and recovery strategy:

i. Never back up databases to local disk
ii. After you back up the databases to a file share, back up the share to tape
iii. Justify the cost of the network share by lower licensing costs and simpler backups
iv. Do regular practices on rebuild and restores.
vi. Keep management informed on restore time estimates
vii. Trust no one. Make sure that IT Governance is always adhered to.
viii. Have solid backup and disaster recovery plans in place before you move to the cloud

2.2.7 The Document Circulation List

All users, significantly affected by a change, must be notified of the change and must form part of the document circulation list. It will include at least the following role players:

- Change Review Board
- Change Co-ordinator
- Change Requester
- Change Implementer
- Change Management Process Owner
- Service Center Administrator
- Change Controller
- Change Scheduler
- Client/user and anybody else that has a direct interest in the documents

2.2.8 Educate Client/User on the Change Management Process

When the change has been completed, the results must be reported for evaluation to those responsible for managing changes, and then presented as a completed change for stakeholder agreement. This must include the closing of related incidents, problems, or known error.

This review must also include any incidents that took place as a result of the change (if they are known at this stage). If the change is part of a service provided by an external organisation, details of any contractual service targets will be necessary (for example, no priority 1 incidents during the first week following implementation).

A post-implementation review (PIR) should be carried out to confirm that the change has met its objectives, that the initiator and stakeholders are happy with the results, and that there have been no unexpected problems or side effects. Lessons learned should be included into future changes. Small organisations may choose to spot- check changes rather than conduct a large-scale PIR; in larger organisations, sampling will have a value when there are many similar changes taking place.

Change management must review new or changed services after a specified period has elapsed. This process will involve CAB members, because change reviews are a standard CAB agenda item. The purpose of such reviews is to establish that:

- The change has met its objectives and had the desired effect
- Users, clients, and other stakeholders are satisfied with the results (if not, the review should identify all the identified shortcomings)
- There are no unexpected or undesirable problems or side effects to functionality, service levels, or warranties, such as availability, capacity, security, performance, and costs
- The resources that were used to implement the change were as planned
- The release and deployment plan worked correctly (the review should include comments from the implementers)
- The change was implemented on time and to cost
- The remediation plan functioned correctly, if needed

Where a change has not achieved its objectives, change management (or the CAB) should decide what follow-up action is required, which could involve raising a revised RFC. If the review is satisfactory or the original change is abandoned (for example, when the circumstances that required the change are no longer current and the requirement disappears) the RFC should be formally closed in the logging system.

Study Unit 3.1: Monitor and Control the Performance of the Project

US 243812

Monitor and control the execution of the project management plan for a simple to moderately complex project

Specific Outcome 1

Monitor and control the performance of the project

Assessment Criteria

3.

4.

1. Performance is monitored and measured in accordance with project procedures.
2. Performance results are evaluated against the project management plan in accordance with project procedures.
   ASSESSMENT CRITERION RANGE
   Performance results include but are not limited to schedule, costs and quality of project deliverables or work elements.
   Preventative and/or corrective actions are developed to ensure compliance to the project management plan.
   Opportunities are identified and actions are developed to benefit from such opportunities in support of project objectives.
   ASSESSMENT CRITERION RANGE
   Opportunities may be early start, early finish, cost savings, other results other than outcome.
5. Developed actions are validated to ensure that it will achieve the desired outcomes.
5. Developed actions are implemented in accordance with project procedures. Performance is reported to stakeholders in required format and time frame.
7
.

3.1.1 Monitor and Measure Performance

Measuring performance is a critical factor in optimising performance. To achieve optimal performance it means that you have to sustainably achieve multiple, often conflicting, objectives under changing conditions.

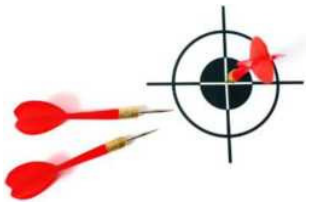
Project performance, on the surface, seems easy to measure; just track time, cost and scope and it's done. But when looking more deeply, it becomes clear that there is much more to it.

First there might be confusion about what must actually be measured. Is it the success of a single project, the success of project performance across many projects, or the success of project management and a project office? If the objective is to measure the success of a single project, are we looking at the project during its life or at the project after it has been completed?

Once it is clear what must be evaluated, it is important to identify the key performance indicators (KPIs) that will determine whether the project has been or is being successful or not and to what degree. KPIs give managers the most important performance information to enable them to assess the performance of a project or process. Usually they look for from two to five or so indicators. One is not enough for any complex process and project performance is a complex process. Too many KPIs make it difficult to see the big picture clearly. Too few make it difficult to diagnose and correct the issues that prevent the project from performing optimally.

Project and Business Objectives

Keep in mind that KPIs are not objectives. They are readings that enable a manager to assess performance towards the achievement of objectives. As an example, and on a personal level, KPIs such as weight, blood pressure, and cholesterol level are used to assess general health. The objective is general health, the measures are indicators that can be used to determine if the person is tracking towards the objective, or not.



Project objectives are to deliver goods and services, within time, cost, quality and other constraints, satisfying stakeholder expectations. Project deliverables are used to deliver benefits to satisfy the needs of sponsors and clients.

It is important to realise that there is a difference between project and business objectives. Projects deliver goods and services that are most often used after the project has ended to achieve business objectives such as reducing costs and risks, increasing revenues, etc. Projects are initiated to achieve business objectives. Project objectives are a means to that end.

When measuring project success, especially when the project is on-going, it is necessary to focus on the project objectives and performance against schedule and budget estimates. In most cases, it is not possible to measure a project's success in achieving business objectives until well after the project has ended. Apart from that, there are also factors that are outside of the control of the project that influence benefits realisation. For example, in product development the performance of sales and marketing, market conditions, and other factors affect the degree to which expected benefits are achieved.

Project Performance - Schedule and Budget

Performance measurement during a project is firstly to know how things are going so that there can be an early warning of problems that might get in the way of achieving project objectives, and so that expectations can be managed. A second benefit is that information gathered can be used to improve the planning and performance of future projects.

Typical Key Performance Indicators (KPIs) include:

- Schedule and budget compliance
- Number of scope changes
- Number of issues and defects
- Stakeholder satisfaction

There is broad agreement that schedule and budget compliance during the course of the project are essential indicators. Projects must end and completion time is often closely linked to the business objectives that drove the initiation of the project. Tracking to a budget in rands and/or resource time is a key indicator because it provides sense of whether the project is performing as it is expected to perform. In most cases, project sponsors and clients are cost conscious. They want to know how much they will spend on the project and they want to know it before the project is over.

Assessing the degree to which the project is staying on schedule and budget provides an indication of whether the team is going to meet stakeholder expectations (a critical objective in any project). If the project is not on track, it also helps to look into the causes of variance. Causes may be poor estimating, loss of resources, price changes, underperforming resources, too many changes, errors, omissions and defects, etc. Once it is clear what the causes are, a course of action can be decided, either accepting things as they

are or making changes to remediate any problem that has been identified.

It is important to combine schedule and budget perspectives to get a true sense of overall project health. It is never good to look at these measures independent of one another, as a project can be under budget because it is behind schedule, because prices have gone down, because performers have used clever means to get their work done less expensively, or for other reasons. A stakeholder who is budget oriented can easily get the wrong idea of project health if he/she only looks at standalone budget data.

Quality Deliverables

To effectively measure performance, it is necessary to plan so that tasks are defined in terms of specific deliverables with clearly stated requirements. The availability of an accepted deliverable is the only measure of a task that is completed successfully. It is all too easy to deliver something that doesn't work or meet requirements.

Effort and Cost Tracking

It is always necessary to account for effort and the cost of other resources to monitor budget compliance and to estimate to project completion based on current performance.

It is possible to measure project performance without tracking effort and cost. It is done all the time. However, without a sense of the effort/cost being expended, any assessment of schedule compliance is only guesswork. Capturing and using effort and cost data is often difficult, rand it needs appropriate tools and cultural change, but it is worth the effort and cost if you want to manage your project effectively.

Project Performance - Other Indicators

Other indicators, aside from schedule and budget performance, monitor the following:

- Number and types of issues
- Changes, and/or defects
- The degree to which they are quickly addressed

Issues are questions, disputes or problems that come up during a project and that must be addressed to satisfy stakeholders and ensure that the project is keeping on track. Issues vary in priority, complexity and the amount of time and effort they need to be solved. Issues are inevitable, plan for them and track the effort required to address them against your estimate.

Scope changes result from requests for change in requirements. Changes will need careful analysis, decision making and execution. Scope changes, especially those that occur late during the project life, can disrupt the project largely. When planning the project it is best to estimate time, effort and cost to create a fund for expected changes and then to monitor actual changes against this fund.

Defects are discovered when testing is performed to validate deliverables. Defects need effort to determine their cause and correct or accept them. Again, it is best to estimate defects and the impact that they have on duration and cost and to track against this estimate.

Each of these is a KPI. High instances of each indicate that there will be project schedule and budget slippage. Issues, changes and defects should be tracked and aged to give management a sense of what is happening outside of the schedule and budget.

One of the most important and often ignored project performance indicators is stakeholder satisfaction. During the course of a project assess the degree to which clients, sponsors and performers are satisfied with how you respond to their issues, the sense that there is enough progress being made, the degree to which they are involved, the health of relationships and their general feeling and satisfaction with the project's performance.

3.1.2 Evaluate Performance Results

Performance results must be evaluated to determine the success or failure of the project, whether the project is still on track in the case of an ongoing project, whether the budget is still within the agreed amounts, and to determine the quality of project deliverables or work elements.

All these aspects have already been discussed in detail in Study Unit 1.1.1.

Evaluation Methods

The approach used to evaluate a project will depend on the nature of the project and its objectives. There are no absolute rules here: the methods, broadly speaking, should be in line with what must be determined by the evaluation. Measuring impact will typically be achieved using quantitative questions; assessing the strengths and weaknesses of an intervention will usually involve gathering open-ended, more qualitative information. Often it is necessary to use both.

Qualitative Methods

Qualitative approaches help you understand 'why' rather than 'how many'. As such, they are used to understand a complex situation or develop an initial understanding of an area before developing a structured questionnaire.

Qualitative methods, being open-ended, also have the value of making your audience feel involved and consulted

Methods include:

- open-ended questions on questionnaires
- personal interviews
- focus group discussions
- observation
- comment boards, diaries, visitor books, etc.

The most commonly used qualitative method is the open-ended question on a questionnaire which allows respondents to talk about their thoughts, opinions and feelings in their own words

e. g. what did you enjoy about the event? These are simple to use and often extremely helpful. You may also want to consider more extended qualitative approaches such as one-to-one interviews or focus groups; these are good to use for in-depth or exploratory studies.

Quantitative Methods

Quantitative methods are concerned with measuring 'how many' or 'to what extent', and rely on statistical theory to judge the importance of results or generalise findings to a broader population.

Simple self-completion questionnaires form the backbone of most project evaluation. However, self-completion methods can result in very low proportions of responses, or biases in the response (e.g. only those who were very happy or very unhappy respond).

If the project warrants it, you may want to consider using personal interview methods (face-to-face or by telephone).

3.1.3 Preventative and/or Corrective Actions

Corrective Action, Preventive Action, and Defect Repair are the most commonly used key elements in the quality management system of project management. It is important to understand these terms in order to have a better command over the quality management processes in the PMBOK Guide (Project Management Body of Knowledge).

The PMBOK® Guide is the standard for managing most projects most of the time across many types of industries. This standard describes project management processes, tools, and techniques for managing scope, schedule, quality, and cost, as well as any project environment aspects that influence the project's outcome.

Defect Repair is an easy concept—repair the defects. However, the difference between corrective action and preventive action is not so obvious, and can be quite confusing.

Corrective action is aimed at taking action to correct the non-conformance event that happened in the past. Whereas, preventive action takes action to avoid or mitigate any potential non-conformance event that may occur in the future.

Let's look at it in more detail.

Defect Repair

Defect Repair is a process of repairing the defective part or replacing it, as needed. For example, let us say that you are manufacturing some component. Suddenly you see that a component is in bad shape or has any kind of discrepancy or non-conformity. You will physically inspect the material and you will see if this defect can be corrected. If this defect can be corrected, you will correct it. And if this defect cannot be corrected, then you will simply replace it.

Defect Repair is also known as the correction and it is performed when the product does not meet the quality requirements.

Corrective Action

Corrective Action is a further response to the defect repair process or the correction, so that the cause of error or non-conformity will not occur again. For example let us say that during the inspection you find some defective component, and you corrected the defective component. Now, you don't want this defect to happen again.

Therefore, you will look into root cause of the problem, find a solution, and apply it to your operations so that the defects do not occur again.

In other words, you can say that the Corrective Actions are the steps taken to correct the root problem and stop the recurrence of deviation.

Corrective Action is a reactive process and it is performed to bring the deviation under control.

Preventive Action

Preventive Action is an action that is taken to avoid any anticipated future defects that may appear in the component. For example, let us say that you are going to start production of some component. Before starting the production process, you think that some defects may appear on the component. Therefore, you review your processes and procedures and make some changes (if needed) so that the cause of anticipated defects could be prevented.

Preventive Action is a Proactive Process

Please note that there is a difference between Corrective Action and Preventive Action. In corrective action, a problem has occurred, and you try to make sure that this problem should not recur. On the other hand, in preventive action, a problem has not yet occurred. You simply take some measures so that any problem should not occur.

In other words, you can say that the preventive action is a process of identification of the most likely cause of any potential non-conformity in order to prevent it from initially occurring.

Preventive Actions are performed to ensure that there should not be any deviation from the baselines.

3.1.4 Identify Opportunities and Develop Actions

The Importance of Controlling Changes to the Duration of Activities

Effective project change control allows you to maintain the proper focus needed to complete projects on time and on budget. Changes to a project are inevitable; the trick is to exercise good judgment and handle change in a streamlined, non-bureaucratic way. Change Control is the process used to properly integrate or postpone requests for changes to the project's scope into the project budget and completion schedule. This plan is

designed to help you implement an efficient and effective method of change control within your project management framework.

Progress against a planned schedule should always be monitored and updated correctly on a continuous basis in order to inform a correct action, and to identify opportunities and develop actions to rectify any problems. These problems may include time variances, for instance early start or early delivery of a project, or cost implications, where a project is running at higher cost than was budgeted for. In order to obtain this data, information pertaining to the progress of the schedule should be collected and distributed timeously.

Once the information has been collected and updated correctly the next step would be to compare the actual versus planned progress. In order to do this many different types of techniques can be used to assist project managers.

Once you have collected the information, you can compare this with the actuals, and make a judgment on whether you're on track or not. This provides you with variance information which can be useful for future estimating.

For instance, a task was planned to start last week and the planned effort/duration was 10 days. It should therefore finish at the end of this week. It is now Monday of week 2. The task actually started last Wednesday and, having reviewed the task, you now estimate is that it will take another 8 days to complete it.

Planned Start Monday Week 1	Actual Start Wednesday Week 1
Planned Finish Friday Week 2	Scheduled Finish Wednesday Week 3
Planned Duration 10 days	Revised Duration 11 days

Hence on current estimates the task will take an extra day to complete and will be three days late.

This is a fairly basic way of assessing how well you're doing. If you add in a value for progress (percentage complete) a better measure is to use Earned Value. If you record the actual hours spent on the task, you may use software may be able to calculate the percentage complete value.

Once a variance has been identified, the correct action step should be taken to address the variance. These action or corrective steps should be communicated with higher management according to agreed company policy and procedure.

Lessons Learned

During the project schedule process, many lessons can be learned to avoid repeating the same mistake in the future for similar projects. The project team could have identified gaps, where if they had done things differently it would have yielded better results. Such lessons learned must be documented and communicated to relevant stakeholders according to agreed company policies and procedures.

Schedule Changes

Corrective action plan could lead to schedule changes due to identified variances. Such schedule changes must be documented and approved, and the associated schedules must be updated according to company policies and procedures. The next step would be to communicate all relevant changes to stakeholders according to company procedure.

Opportunity Management

Opportunity management determines the payback of the project within the initiation stage. By establishing the payback period within the opportunity management process, project managers may continually assess the project expenditures and re-evaluate the payback period on an ongoing basis.

Action Plan

Action Plans are simple lists of all of the tasks that you need to finish to meet an objective, whether it is to start early, finish early or to save some costs.
They differ from To-Do Lists Ad in that they focus on the achievement of a single goal.

Action Plans are useful, because they give you a framework for thinking about how you'll complete a project efficiently. They help you finish activities in a logical order, and they help you ensure that you don't miss any of the important steps. Also, because you can see each task laid out, you can quickly decide which tasks must be prioritise, which you can delegate or outsource, and which tasks you may be able to ignore.

Using Action Plans

Use an Action Plan whenever you need to plan a project.

To draw one up, simply list the tasks that you need to carry out to achieve your objective, in the order that you need to complete them. (This is very simple, but it is still very useful!)

Use the three-step process below to help you:

Step 1: Identify Tasks

Start by brainstorming all of the tasks that you need to complete to accomplish your objective.

It's helpful to start this process at the very beginning. What's the very first action you'll need to take? Once that task is complete, what comes next? Are there any steps that should be prioritised to meet specific deadlines, or because of limits on other people's availability?

Step 2: Analyse and Delegate Tasks

Now that you can see the entire project from beginning to end, look at each task in greater detail.

Are there any steps that you could drop, but still meet your objective? Which tasks could you delegate to someone else on your team, or could be dealt with by a freelancer? Are there any deadlines for specific steps? Do you need to arrange additional resources?

Step 3: Double-Check With SCHEMES

Use the SCHEMES mnemonic to check that your plan is comprehensive.

SCHEMES stands for:

You may not need to think about all of these to complete your project. For instance, for a small internal project to streamline the format of your team's reports, you might only need to think about "Helpers/People," "Expertise," and "Systems."

Note:

Once you've completed your plan, keep it by you as you carry out the work, and update it with additional activities if required.

Learning from Your Action Plan

If you think you'll be trying to achieve a similar goal again, revise your plan after the work is complete, by making a note of anything that you could have done better.

For instance, perhaps you could have avoided a last-minute panic if you'd alerted a supplier in advance about the size of order you'd be placing. Or maybe you didn't allow enough time to do certain tasks.

3.1.5 Validate Developed Actions

During the project life cycle, it is important that the project activities are controlled and validated. The controlling can be mainly done by adhering to the initial protocols such as project plan, quality assurance

Validation is a supporting activity that runs from first day to the last day of a project. Each and every developed action, and delivery should have its own validation criteria in order to verify the successful outcome or the successful completion.

When it comes to project deliveries and requirements, a separate team called 'quality assurance team' will assist the project team for validation and verification functions.

Validation of the developed actions confirms that the actions that were decided on match the details in the WBS, project scope plan and project plan. This is done using reviews or audits and user trials. It differs from quality control because it is about the acceptance of the definition of the deliverables while quality control deals with whether or not the deliverables meet their quality requirements.

3.1.6 Implement Developed Actions

After all paperwork is done, in this phase, the project management implements the developed actions of the project in order to achieve project objectives.

When it comes to implementation, each member of the team carries out his or her own assignments within the given deadline for each activity.
The detailed project schedule will be used to track the progress of the project

During the project implementation and execution, there are many reporting activities to be done. The senior management of the company will require daily or weekly status updates on the project progress. This is discussed in more detail in Study Unit 1.1.7.

In addition to that, the client may also want to track the progress of the project. During the project implementation and execution, it is a must to track the effort and cost of the project in order to determine whether the project is progressing in the right direction or not.

In addition to reporting, there are multiple deliveries to be made during the project execution. Usually, project deliveries are not once off deliveries made at the end of the project. Instead, the deliveries are spread out throughout the project execution period and delivered upon agreed timelines.

3.1.7 Report Performance to Stakeholders

The Performance Reports are the reports given to project stakeholders to make them aware of the current status and the forecasted progress of the project.

Performance Reports have a variety of information pertaining to project parameters, and updates on the project's progress. They show stakeholders how the project is going, the forecast analysis of what they should expect if the project is allowed to keep going in the same way, or what additional funds or resources may be required to complete the project if there is any deviation from any baselines (e.g. cost and schedule baselines).

The fifth edition of the PMBOK Guide defines the Performance Reports as follows:

"The physical or electronic representation of work performance information compiled in project documents, intended to generate decision or raise issues, actions or awareness."

In other words, it can be said that the Performance Reports organise and summarise the information collected through work performance data and work performance information, and represent it to the stakeholders in such a way that they can understand the progress made in the project.

Performance Reports show the stakeholders the current status of the project and its performance against the planned baselines.

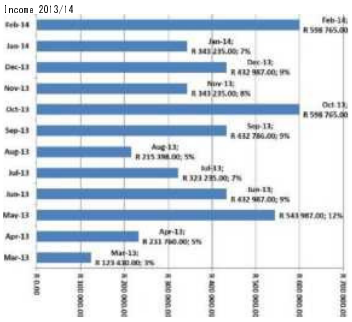
If the stakeholders see that the project is not making progress as was planned, they may decide to take corrective action. For example, if extra funds or resources are needed, or any time extension is needed to complete the project, etc., the stakeholders are made aware of it timeously.

Performance Reports can be given to stakeholders in a format desired by them and decided in the communication management plan. They can be a detailed report or just a summary.

Contents of Performance Reports

The content of the Performance Reports includes, but is not limited to:

- Percentage of the work completed during the reporting period
- Balance of the work to be completed
- Cost incurred during the reporting period
- Balance of funds available
- Balance of time available
- Major risks that have occurred, or passed without occurring
- Major remaining identified risks
- Results of variance analysis: e.g. schedule variance and cost variance
- Performance indexes: e.g. schedule performance index and cost performance index
- Forecasted fund required to complete the remaining work (if the project cost is overrun or underrun)



- Forecasted time required to complete the remaining work (if the project is delayed or ahead of schedule)
- Summary of major approved change requests during the reporting period etc.

The format of the Performance Reports may be any combination of these formats:

- Burndown Chart
- S-Curve
- Bar Charts
- Histograms
- Tables
- Run Charts.

With this information on hand, stakeholders are very aware of the project health, and can make a decision based on these objective data and analysis.

Study Unit 3.2: Manage Project Baselines through Integrated Change Control

US 243812

Monitor and control the execution of the project management plan for a simple to moderately complex project

Specific Outcome 2

Manage project baselines through integrated change control.

Assessment Criteria

1. Change needs are identified and recorded in accordance with project change control procedures.
2. The impact of the change needs are assessed and evaluated in terms of the relevant baselines.
3. Recommendations to address the change needs are compiled in required format and approval obtained in accordance with project change control procedures.
4. Approved changes are implemented in accordance with project change control procedures.
5. Change information is communicated to relevant stakeholders in accordance with project change control procedures.

Schedule and scope changes are part of any project as it progresses from start to finish. Some scope changes are critical and may reflect a complete shift in the direction of the project. Others may be unnecessary to the primary functionality of your end solution but they may be necessary to make future risks less severe. In the light of the impact that these types of adjustments have on the project schedule and requirements, it is very important to have a process in place to help project managers decide whether or not a change should be adopted.

3.2.1 Identify and Record Change Needs

A request for change is not only restricted to a business user asking for fancier functionality. Instead, it can fall into one of the following four categories:

- Corrective Action: If your project schedule has fallen behind, or if quality is not what it should be, a request for corrective action may be done. This type of request sets out specific directions for future project work to bring performance in line with the original project management plan. It may include contingency plans and workarounds for risks that were not identified previously.
- Preventative Action: As with corrective action change requests, the role of project managers in forecasting risks and issues is very important. A request for preventative action is documented direction to include in the project plan activities that can make the chances of risks developing into issues down the road less severe.
- Defect Repair: This type of change request serves as documented identification of a defect in a project component with a recommendation to either repair the defect or completely replace the component.
- Updates: These are scope changes to formally controlled documentation and plans, including requirements, project plans, design specifications, etc.

Part of the process of monitoring and controlling project performance is regular comparison between the planned project progress and the actual results that are achieved. Comparing planned vs. actual technical, schedule, and cost performance may be the reason for one or more of the above change requests.

Notice too that for each of the request types there must be documentation of updates or actions that need to be taken. If you need to take a holistic view of project needs and documenting scope changes for the good of the end solution, it means there must be proactive thinking and the forecasting of modifications to protect the project plan. Documentation is important to ensure that everyone is on the same page, whether the change is asked for by business stakeholders or by the project development team.

Typically change request forms include:

- Description of the change.
- The impact of not approving the change.
- The impact of the change on project scope/schedule/cost.
- And defined signature paths for review as well as a tracking a log number.

An abridged version of change request log for a construction project is presented below. These logs are used to monitor change requests. They typically summarise the status of all outstanding change requests and include such as useful information as source and date of the change, document codes for related information, cost estimates, and the current status of the request.

Every approved change must be identified and integrated into the plan of record through changes in the project WBS (work break down Structure) and baseline schedule. The plan of record is the current official plan for the project in terms of scope, budget, and schedule. The plan of record serves as a change management benchmark for future change requests as well as the baseline for

evaluating project progress.

If the change control system is not integrated with WBS and baseline, project plans and control will soon self- destruct. Thus, one of the keys to a successful change control process is document, document, document.

The following is an example of a change request:

Project name: _____ Project Sponsor: _____

Request number: _____ Date: _____

Originator:_____ Change request by: _____

Description of requested change

1. Request river dancers to replace small Irish dance group.
2. Request one combination dance with river dancers and China ballet group.

Reason for change

River dancers will enhance stature of event. The group is well known and loved by Chinese people. Areas of impact of proposed change-describe each on separate sheet

Scope Cost other

Schedule Risk		
Disposition	Priority	Funding Source
Approve	Emergency	Mngmt. Reserve
Approve as amended	Urgent	Budget reserve
Disapprove	Low	Customer
Deferred		Other
Sign-off-Approvals		
Project Manager		Date:
Project sponsor		Date:
Project Customer		Date:
Other		

3.2.2 Assess and Evaluate the Impact of the Change Needs

The Project Management Book of Knowledge (PMBOK) calls the review and approval of documented change requests the process of 'integrated change control'. Controlling project changes is a process that begins right at the beginning of a project and doesn't end until implementation has been completed. It is necessary to institute a formal review and approval process for the suggested changes to project activities, timelines, and scope in order to make sure that the impacts to the project budget are being considered for every change proposal and that the integrity of project baselines is being maintained.

When assessing a change request, the following main categories of dependencies should be considered:

- Cost Estimates: When formulating a change request or when looking at one that has been submitted, it is important to spend time to evaluate projected changes to the original project budget based on the extent of the scope change or on the corrective actions that are being requested. Identify whether or not the cost of implementing the change matches up with the benefits proposed.
- Activity Sequences and Dates: Does the proposed change have an effect the completion of project activities and the order with which they've been planned on the project schedule? If so, analyse how these sequence changes will affect later deliverables. There may be project teams or business departments that are dependent on work that has been scheduled previously, and that now need to adjust their own plans of action. Remember: Keep the bigger picture in mind to avoid unnecessary work disruptions.
- Resource Requirements: As with activity sequences and dates, assessing change requests also involves thinking about what new resources may be necessary to put the project changes into effect. Or on the other hand, what resources will no longer be necessary. Will it be necessary to review work contracts? What about work packages down the road that were dependent on resources that may now be tied up for longer than was originally planned? All these aspects will have an effect on cost and schedule and will need serious consideration before a request is approved.
- Risk Responses: Finally, before approving a change request, consider any new risks that arise, even if the request proposed is meant to mitigate other risk factors. Keep in mind that a project's "triple constraint" is named as it is for a reason - mitigating risks in one area may cause issues in another, like scheduling, resources, etc. Always gather input from all affected work groups before final approval is given.

Review and analysis of change requests should be done as quickly as possible, since slow decisions may influence risk, schedule, and budget factors that the request was looking to mitigate in the first place. The longer you wait, the less feasible other project changes may become, especially if development changes are involved that affect future design work packages.

3.2.3 Compile Recommendations to Address the Change Needs and Obtain Approval

Throughout this whole process, managing stakeholder expectations is a very important factor to a successful deployment and may not always be easy, especially when there may be unexpected project changes. It is important to find a way to support business lead, end users, and others in this new or modified approach and address concerns they may have. The project manager can lead the way by recommending actions to take in preparing for a change in scope. There are three main groups in particular that will need guidance:

The Development Team

First priority should be to lead the development team through an update in their work packages or through a modification in scheduling or resource allocation. While most developers appreciate a challenge, a negative team spirit can develop if project changes are not properly introduced or if a change is misunderstood by the technical team. They are the implementers of any changes in scope, so it is important to make sure that they understand the new requirements 100% and that priorities are clearly set.

Business Stakeholders and End Users

What will be the main concerns for end users? Consider the importance of any process changes that end users will have to manage once the new application is deployed. Will their role or any of their requirements change even slightly? If so, address training needs with managers to make sure users will be comfortable with what is required of them and productivity remains as expected following deployment.

Project Sponsors

A ask the same sort of questions than in the case of end users when considering how best to guide sponsors through project changes: What will be their main concerns going forward? Primarily, they will be concerned with budget updates, new schedules, and changes to Return On Investment (ROI). Communicate the dependencies considered when assessing change impacts to sponsors to make sure everyone is on the same page. These include:

- New cost estimates
- Project phase scheduling and dates
- New resourcing requirements
- Any issue/risk mitigations

Managing sponsors will be less difficult as long as emphasis is placed on communication and open dialogue to mitigate any surprises that come about because of project changes.

Obtain Approval

There are three possible courses for the approval of the change:

- Minor changes within scope can be approved by the Project Manager.
- Any change affecting an external sub-contractor would need to be reviewed with that contractor who would agree any necessary contract revisions or payments etc.
- Changes of scope and contract revisions would require the approval of the Steering Committee (or it might have been a Change Control Board).

In all instances, the company policies and procedures document must be consulted and one must ensure the parameters outlined in the document are complied with.

In making the decision, the Project Manager, Change Control Board or Steering Committee would be guided by the pre-established principles for making change decisions.

3.2.4 Implement Approved Changes

Scope changes are performed much like other project work package items and are managed end-to-end in a similar way. The two main points to consider (that become more critical when the work is a result of a change request) are:

- Making sure that all documentation us kept updated and the work is in line with customer expectations
- Closely managing testing of the new or modified scope items to make sure that no negative impacts have been made on already developed functionality

The change management plan that was created when the project first began will now come in handy. This plan is what had documented how changes should be managed, monitored, and controlled. It should contain which project documents should be revisited if a project change is approved. If this was not done it will be necessary to double-check the list of standard documents below to cover your bases:

- Business Requirements Document
- Functional Design Document
- Technical Design Document
- Test Cases/Scripts

If applicable, it may also be necessary to take a second look at any mock-ups separate and apart from the Functional Design Document to ensure they are consistent with interface changes.

Once documentation is up to date, it is now necessary to test the new or modified scope items. That begins with making sure that the appropriate scope has been properly captured. Once that is complete, the traceability of the requirements will lead the way.

But here's a point to consider: What if a project change has been requested shortly before go-live, meaning testing of work packets has already been performed by developers and users? Is a rigorous testing cycle still needed after a new scope change is developed? YES! Even after a regular project's testing cycle is complete, performing due diligence on all project changes in the way of unit, integration, and regression testing will ensure there are no backslides in performance and reliability of your end result. While it may be tempting, don't be so confident that you lose sight of the potential for even the smallest scope change to throw your whole project off schedule if the change is not properly managed.

3.2.5 Communicate Change Information to Relevant Stakeholders

It is important that all change information is communicated on a regular basis to all the stakeholders involved. Below are some information on how to communicate with stakeholders.

The most important element in stakeholder communications is identifying the target audience. Always obtain input from all known groups to find the unknown groups. It can be tough when too late in the project a critical person or group is identified that has not received any of the communication through course of project and has valuable links that need to be addressed. So make sure you avoid this scenario and take all the steps early to create a document with all stakeholders with whom there should be communication. Once that has been done, the ways below can help you to keep communication active, frequent and ongoing

collaboration so there is strong support for the project.

Create formal methods for communicating, and make occasions when info should be presented.

1. Meetings - One of the most common ways to communicate. It is important to maximise every minute of the time spent to discuss important matters. Make sure it is a dialogue (two-way discussions) and not a monologue (one way speeches). It is the best way as you have the verbal and non-verbal cues that enhance the communication and avoid misinterpretation.

2. Conference Calls - These days this is the most common as it does not require the time and expense of travel. The dialogue can take place

though its dependant on voice intonation and clarity of the verbal message. They only require cost of phone call and there are many paid and free services that will facilitate use of a conference call line for many people to dial into.

3. Newsletters/ Email/ Posters - This strategy is one way communication and makes use of e-mailed updates, hard copy brochures, posters, newsletters mailed or emailed. One of the weaknesses is that messages are delivered and you have no idea whether they were read and understood, or simply deleted, as sometimes there is no feedback. That immediate feedback is valuable for strengthening the message and making sure impacts and feedback are quickly received.

Informal Methods - It is important to not only rely on formal channels but to also make use of informal communication. The impromptu channels are often more information rich and critical for relationship building.

4. Hallway Conversations, Bathroom conversations - These meetings are great for one on one communication, but also be clear and do not establish false expectations with casual comments dropped.

5. Lunch Meetings, Drink at the bar after work - These casual environments can be great for connecting, getting feedback, ideas, and work to build support

6. Sporting events - tennis, golf, etc. are an easy forum to get the input on what support exists, feedback on ideas, brainstorming to strengthen your communication and build stakeholder support

7. Voice mail - this is often underutilized since email is so common but still shown to be more often listened to than an email will be read. By using voice intonation for excitement, urgency, etc. it can be more compelling. This can be a solo voice mail, a voice mail broadcast to large team or you could pursue use of automated calling to get the word out depending on the size of audience.

Project Communication Plans

It's not enough to just have a plan. It is critical to make sure that you understand what your stakeholders want, both spoken and unspoken. The expectations must be carefully managed from beginning to end. Every team and project varies in its rate of change, so pick the best communication channel, frequency and make sure its effective. Just as having the plan is important, monitoring its effectiveness, adding and cancelling supplemental ways of communicating will be required.

Communication is a constant, error on the side of over communicating as there are always people that didn't hear, understand or make connection when they heard it the first time.

Study Unit 3.3: Manage and Resolve Issues and New Risks

US 243812

Monitor and control the execution of the project management plan for a simple to moderately complex project

Specific Outcome 3

Manage and resolve issues and new risks.

Assessment Criteria

1. Issues are identified and addressed in accordance with project procedures to achieve project objectives.
2. New risks are identified and addressed in accordance with project procedures to achieve project objectives. ASSESSMENT CRITERION RANGE Risks include threats and opportunities.
3. Issues and new risks that may have a change impact on the project, are dealt with in accordance with project change control procedures.

The Risk Management Process

The figure below represents the risk management process.

Step 1: Risk Identification

Analyse Che project to Identify Che source of risk **Step 2: Risk Assessments**

Assess risk in terms of;

Severlcy of Impact Likelihood of occurring Controllability

Step 3: Risk Response Development

D eve lap a strategy to reduce possible damage Develop contingency plan

Step 4: Risk Response Control

Implemenc risk strategy Monitor and adjust plan for new risks Change management

3.3.1 Plan, Identify and Address Issues

In the life cycle of any project, there will almost always be unexpected problems and questions that crop up. When these issues arise, it is important to be ready to deal with them - or they can potentially affect the project's outcome.

Most issues are, by their nature, unexpected, so it is important to make sure that you'll be able to deal with them quickly and effectively. Ideally, an issue resolution process should be in place before the start of the project - to make sure that that project stays on schedule, and meets its objectives.

Issue management is the process of identifying and resolving issues. Issues like problems with staff or suppliers, technical failures, material shortages might all have a negative impact on a project. If the issue goes unresolved, unnecessary conflicts, delays, or even failure to produce the deliverable are created.

Issues versus Risks

Issues and risks are not quite the same thing. However, the exact nature of both is largely unknown before the start of a project. With risks, you usually have a general idea in advance that there's a cause for concern. An issue is usually not as predictable; it can arise without any warning. For example, being unable to find qualified staff is an identifiable risk. However, when one of your staff is in a car accident, and ends up in hospital for three weeks, that becomes an issue!

It's important to identify risks before the start of the project. A Risk/Impact Probability Chart is a useful framework to help you prioritise your risks. You can then develop a plan to manage those risks proactively with solutions that were already thought through and prearranged. However, issues have to be dealt with as they happen. Issue management, therefore, is a planned process for dealing with an unexpected issue - whatever that issue may be - if and when one arises.

Tip: When risks are not identified and reduced at the beginning of a project, they can often become issues later on. Make sure you understand your risks early. Learn from previous projects, and benefit from the team's past experiences. This way, you'll have fewer issues to manage as you move forward.

Issues Log

Issues - also known as problems, gaps, inconsistencies, or conflicts must be recorded when they happen. An issues log provides a tool for reporting and communicating what's happening with the project. This makes sure that issues are indeed raised, and then investigated and resolved quickly and effectively. Without a defined process, issues are ignored or not taken seriously enough - until it's too late to deal with them successfully.

An issues log allows you to do the following:

- Have a safe and reliable method for the team to raise issues.
- Track and assign responsibility to specific people for each issue.
- Analyse and prioritise issues more easily.
- Record issue resolution for future reference and project learning.
- Monitor overall project health and status.

An issues log can be created by hand, by using a spreadsheet or database, or buy issue management software from a wide variety of vendors. However, do bear in mind that the success of your issue management process doesn't necessarily depend on which tracking mechanism you use, but rather on the type of information you track.

The following information should be included in an issues log:

- Issue type - define the categories of issues that could probably be encountered. This helps to track issues and assign the right people to resolve them. You could have broad descriptions like these:
    - Technical - relating to a technological problem in the project.

- Business process - relating to the project's design.
- Change management - relating to business, customer, or environmental changes.
- Resource - relating to equipment, material, or people problems.
- Third party - relating to issues with vendors, suppliers, or another outside party.

Identifier - record who discovered the issue.

Timing - indicate when the issue was identified.

Description - provide details about what happened, and the potential impact. If the issue remains unresolved, identify which parts of the project will be affected.

Priority - assign a priority rating to the issue, for example:

- High priority - a critical issue that will have a high impact on project success, and has the potential to stop the project completely.
- Medium priority - an issue that will have a noticeable impact, but won't stop the project from proceeding.
- Low priority - an issue that doesn't affect activities on the critical path, and probably won't have much impact if it's resolved at some point.

Assignment/owner - determine who is responsible to resolve the issue. This person may or may not actually implement a solution. However, he or she is responsible for tracking it, and ensuring that it's dealt with in line with its priority.

Target resolution date - determine the deadline for resolving the issue. If a date for resolution changes, keep both the old date and the new date visible. This helps to identify issues that have been on the log for a long time. Then they can either receive extra attention, or be taken off the list if they're no longer important.

Status - track the progress of the resolution with a clear label identifying the issue's overall status. For example:

- Open - the issue has been identified, but no action has yet been taken.
- Investigating - the issue, and possible solutions, are being investigated.
- Implementing - the issue resolution is in process.
- Escalated - the issue has been raised to management or the project sponsor/steering committee, and directions or approval of a solution is pending.
- Resolved - the resolution has been implemented, and the issue is closed. Use 'traffic lights' when reporting issues. This provides an easy-to-see indication of whether issues are under control. Traffic lights could be used as follows:

Red - cannot proceed before issue is resolved.

Yellow - resolution in process, and you'll be able to proceed soon. Green - resolution implemented, and issue no longer exists.

- Action/resolution description - describe the status of the issue, and what has been done to find and implement a resolution. Include the dates of each action. For example:
  - January 5 - assigned issue to Samantha.
  - January 7 - testing started to identify origin of problem.
  - January 8 - solution suggested, and sent to steering committee for approval.
  - January 10 - approval received. Assigned implementation to Gregory.
  - January 14 - solution successful. Issue resolved.
- Final resolution - include a brief description of what was done to address the issue.

Issues Management Framework

Supplement your issues log with a framework, or process, for dealing with those issues. This framework helps the project team understand what to do with issues once they've been identified and logged. Developing the framework answers questions like these:

- How will you assign responsibility for resolving the issue? For example, is there one person who handles all technical issues? Who would handle a vendor issue?
- How will you know when to escalate an issue to management or the steering committee? It may be necessary to create a matrix of potential business impact versus issue complexity to help with the decision on which issues should be taken to higher levels of management.
- Which criteria will determine the priority status of an issue?
- Who will set the target resolution date?
- How will issues be communicated within the team? Will there be regular meetings, log checks, status update emails, and so on?
- How will different issues be identified if several occur during one project? It is helpful to number them so that it easy to identify issues when discussing them in progress meetings.
- If change orders are needed, how will those be handled?
- When the resolution affects the budget or schedule, what will the update process be, and who will be responsible?

One of the most important challenges of issues management is to resolve the problem quickly and then move on, with as little impact to the project as possible. The framework provides a structure for making decisions when issues arise. Remember to also consider the needs of the team when developing the framework.

3.3.2 Plan, Identify and Address New Risks

Risk Management Planning

Risk Management Planning is the process of deciding how to approach and conduct the risk management activities for a project. Planning of risk management processes is important to ensure that the level, type, and visibility of risk management are appropriate with both the risk and importance of the project to the organisation, to provide sufficient resources and time for risk management activities, and to establish an agreed-upon basis for evaluating risks.

The Risk Management Planning process should be completed early during project planning; the result of Risk Management Planning is a Risk Management Plan. The risk management plan identifies and establishes the activities of risk management for the project in the project plan (RMP).

Risk Identification

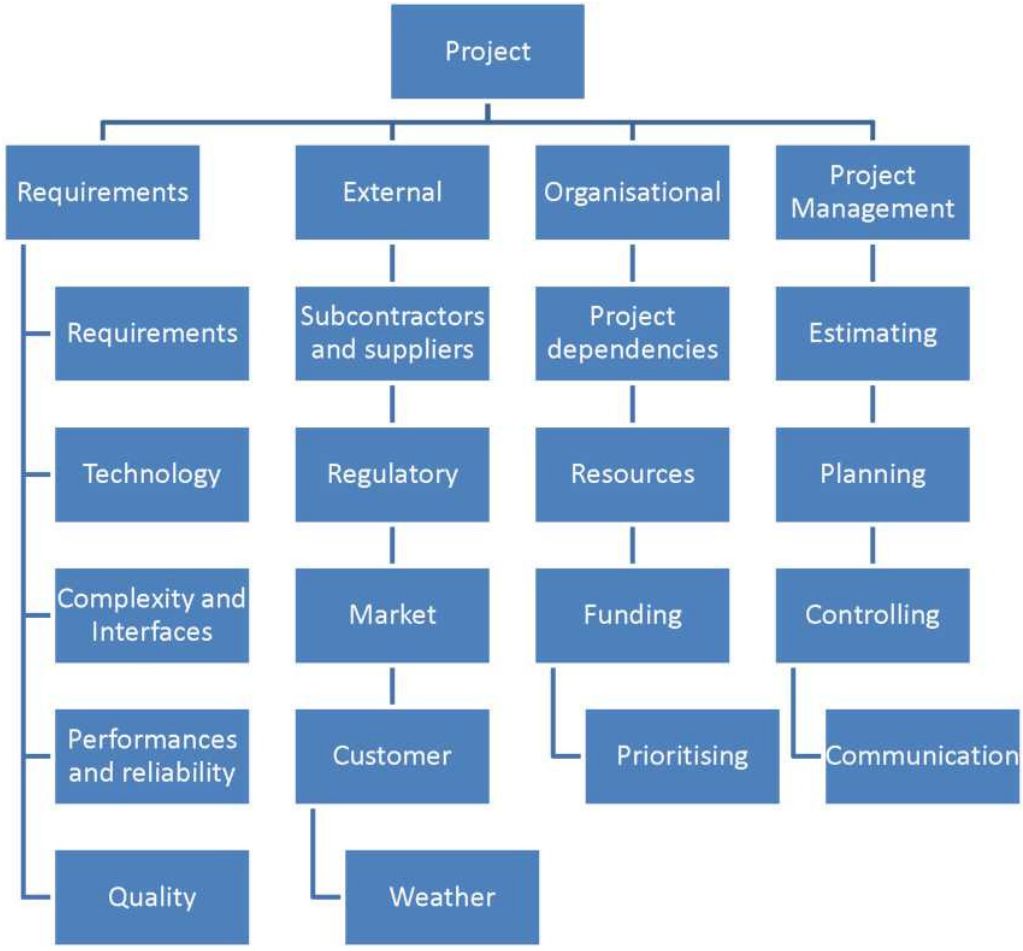
Risk identification involves identifying potential project risks. The project manager pulls together, during the planning phase, a risk management team consisting of:

- Project manager
- Project team members
- Risk management team (if assigned)
- Subject matter experts both from the project and from outside the project team
- Customers, end users, other project managers, stakeholders, and risk management experts

The team uses brainstorming and other problem identifying techniques to identify potential problems. Participants are encouraged to keep an open mind and generate as many probable risks as possible.

The whole objective of the risk identification step is to produce a risk register. A risk register is a document used to document identified risks and their characteristics that might affect the project's ability to achieve its objectives. Risk Identification documents which risks might affect the project and documents their characteristics.
The assigned team members identify the potential risks (threats and opportunities), using:

- The risk breakdown structure- the risk breakdown structure focuses on risks that affect the whole project as opposed to a specific section of the project or network. The figure below shows an example of a Risk Breakdown Structure (RBS)



The Work Breakdown Structure (WBS)

- Another tool that can be used to identify risks within a project is a work breakdown structure, the Use of a WBS reduces the chance a risk event will be missed. On large projects multiple risk teams are organised around specific deliverables and submit their risk management reports to the project manager.

Other tools that can be used include:

- The risk team's own knowledge of the project or similar projects.
- Consultation with others who have significant knowledge of the project or its environment. Consultation with others who have significant knowledge of similar projects.

When identifying a risk, it is important to state the risk correctly. You need to state what the cause of the risk is and if it occurs what impact it will have on the projective objective. For instance, a risk has a cause and, if it occurs, an impact on a project objective.

The risk statement structure that you should use to identify a risk should be as follows:

"Because of the ...(cause and condition that is true), (a risk) may occur, leading to an impact on XX objective (where XX is cost, time, scope and or quality)

This structure helps specify the risk correctly. Example: The use

of the risk statement structure

The fact that the bridge is built over water is not a risk, it is a cause. The risk may be unknown sub-surface conditions, which if they occur may lead to re-design of the supports. Mitigation could involve coring at the support location and engineering analysis based on the findings, to reduce the probability of unknown conditions. In risk identification, sometimes there is a temptation to dismiss a risk because "we cannot do anything about it anyway".

The argument above does not change the risk into a non-risk, simply because there is no viable mitigation strategy. The risk that cannot be mitigated may have an effect on the project and can be calibrated in qualitative and quantitative analysis. It is just not possible to handle this risk, but it is still a risk. Also, some of these risks may be affected by risk handling if people think carefully about it, for instance, political risk may be influenced by public outreach and information campaigns.

In identifying the risk, the team needs to consider:

- Threats — a risk that will have a negative impact on a project objective if it occurs (what might happen to jeopardize the project's ability to achieve its objectives)
- Opportunities — a risk that will have a positive impact on a project objective if it occurs (what might happen to improve the project's ability to achieve its objectives)
- Triggers — symptoms and warning signs that indicate whether a risk is becoming a near-certain event and a contingency plan/response plan should be implemented.

The team should also consider:

- Residual risks - Risks that remain even after developing responses to the project's original risks. Example: You identify delays caused by hazardous waste issues as one of your primary risks. If you are able to develop a response that mitigates only problems caused by underground fuel tanks, you may still have other hazardous waste risks. Your goal is to reduce residual risks to an acceptable level.
- Secondary risks - Secondary risks are caused by responses to the project's original risks. For example, if you decide to hire outside help as a way of mitigating a project risk, you now have additional concerns that arise as a result of using the external vendor. The timeliness of their work and potential contractual disputes are risks you did not have before you decided to use their services.
- Risk interaction - The combined effect of two or more risks occurring simultaneously is greater than the sum of the individual effects of each free standing risk. For instance, Federal budget cuts may increase delays in Federal Highway Administration permits at the same time federal programming dollars become scarcer.

The risk identification process should not be limited to just the core team. Input from customers, sponsors, subcontractors, vendors and other stakeholders should be solicited. Relevant stakeholders can be formally interviewed or included on the risk management team. Not only do these players have a valuable perspective but by involving them in the risk management process they also become more committed to project success.

The goal of risk identification is to find potential problems, to document them before they happen. It is important that project managers set the right tone and complete the risk management process so members gain confidence in themselves and the project.

Risk Assessment

Step 1 produces a list of potential risks. Not all of these risks deserve attention. Some are trivial and can be ignored, while others pose serious threats to the welfare of the project. Management needs to develop methods of sifting through the lists of risks, eliminating inconsequential or redundant ones and stratifying worthy ones in terms of important and need of attention.

Scenario analysis

A scenario analysis is the easiest most commonly use technique used to analyse risks. Team member's asses the significance of each risk event in terms of:

- Probability of the event
- Impact of the event

When assessing the risk the risk is evaluated in terms of the likelihood or probability that the event is going to occur and the impact or consequence that it will have if it occurs.

For Example

The risk of a project manager being struck by lightning at a work site would have major negative impact on the project, but the likelihood is so low it is not worthy of consideration. Conversely, people do change jobs, so an event like the loss of key project personnel would have not only adverse impact but also a high likelihood of occurring in some organisation

In order to mitigate or respond to the risk identified above, it will be wise for that organisation to be proactive and mitigate this risk by developing incentive schemes for retaining specialists and or engaging in cross-training to reduce the impact of the turnover.

In order to identify the probability or likelihood of a risk occurring different levels of risk probabilities can be used, examples of such levels include:

- Scale ranging from very unlikely to almost certainly or
- Numerical probabilities such as (0, 1, 2, 3, 4, 5).

Impact scales are a bit more complicated since adverse risks affect project objectives differently.

For Example

A component failure may cause only a slight delay in project schedule but a major increase in project cost. If controlling cost is a high priority, then the impact would be severe. If, on the other hand, time is more critical than cost, then the impact would be minor.

Because of this complication impact needs to be assessed in terms of project priorities, different kinds of impact scales can be used.

- Some scales may simply use rank-order descriptors, such as "low", "moderate", "high", and "very high".
- Others use numeric weighs e.g. (1-10).
- Others may focus on specific project objectives.

The risk management team needs to establish up front what distinguishes a 1 from a 3 or "moderate" impact from "sever" impact. The figure below provides an example of how impact scales could be defined given the project objectives of cost, time, scope, and quality.

Relative or Numerical Scale					
Project Objective	1 Very low	2 Low	3 Moderate	4 High	5 Very high
Cost	Insignificant cost increase	<10% cost increase	10-20% cost increase	20-40% cost increase	>40% cost increase
Time	Insignificant time increase	<5% time increase	5-10% time increase	10-20% time increase	>20% time increase
Scope	Scope decrease barely noticeable	Minor areas of scope affected	Major areas of scope affected	Scope reduction unacceptable to sponsor	Project end item is effectively useless
Quality	Quality degradation barely noticeable	Only very demanding applications are affected	Quality reduction requires sponsor approval	Quality reduction unacceptable to sponsor	Project end item is effectively useless

Documentation of scenario analyses can be done in risk assessment forms. Below is an example of a risk assessment from used on an IS project involving the upgrade form Windows XP to Windows Vista.

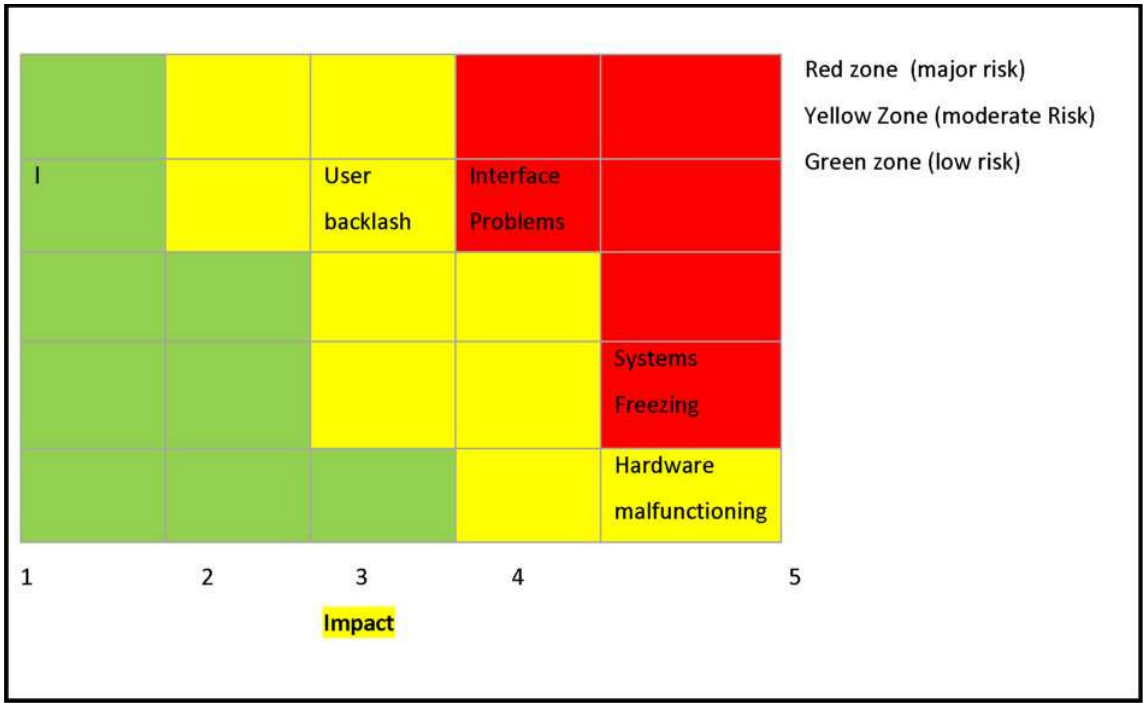
Risk Event	Likelihood	Impact	Detection Difficulty	When
Interface problems	4	4	4	Conversion
Systems freezing	2	5	5	Start up
User backlash	4	3	3	Post installation
Hardware malfunctioning	1	5	5	Installation

In addition to evaluating the likelihood and probability of risk events the team should also asses when the event might occur and its detection difficulty. Detection difficulty is a measure of how easy it would be to detect that the event was going to occur in time to take a mitigation action, that is, how much warning we would have. The organisation should also examine the severity of the risks.

In order to categorise the severity of different risks, organisations often use a risk assessment matrix.

Risk Assessment Matrix

The risk assessment matrix is typically structured around the impact and likelihood of the risk event. The figure below is an example of a risk assessment matrix.



This severity matrix consists of a 5x5 array of elements representing a different set of impact and likelihood values. The matrix is divided into red, yellow and green zones representing major, moderate and minor risks, respectively. The red zone is centered on the top right corner of the matrix (high impact/high likelihood), while the green zone is centered on the bottom left corner (low impact/low likelihood). The moderate risk, yellow zone extends down the middle of the matrix. Since impact is generally considered more important that likelihood (a 10% chance of losing R1000, 00 is usually considered a more severe risk than a 90% chance of losing R1000), the red zone (major risk) extends farther down the high impact column.

An Example of the Windows Vista

Using the Windows Vista project again as an example, interface problems and system freezing would be placed in the red zone (major risk), while user backlash and hardware malfunctioning would be placed in the yellow zone (moderate risk)

The risk severity matrix provides a basis for prioritising which risks to address. Red zone risks receive first priority followed by the yellow zone risks. Green zone risks are typically considered inconsequential and ignored unless their status changes.

Risk Response Development

When a risk event is identified and assessed, a decision must be made concerning which response is appropriate for the specific event. Responses to risk can be classified as mitigating, avoiding, transferring, sharing, or retaining.

Avoid

Risk avoidance involves changing the project plan to eliminate the risk or to protect the project objectives (time, cost, scope, quality) from its impact. Although it is impossible to eliminate all risk events, some specific risks may be avoided before you

launch a project e.g. choosing an Australian supplier as opposed to an Indonesian supplier would virtually eliminate the chance that political unrest would disrupt the supply of critical materials.

Transfer

Risk transference requires shifting the negative impact of a threat, along with ownership of the response, to a third party. An example would be the team transfers the financial impact of risk by contracting out some aspect of the work. Another more obvious way of transferring risk is insurance. Transference reduces the risk only if the contractor is more capable of taking steps to reduce the risk and does so. Risk transference nearly always involves payment of a risk premium to the party taking on the risk.

Transference tools can be quite diverse and include, but are not limited to the use of: insurance, performance bonds, warranties, guarantees, incentive/disincentive clauses, A+B Contracts, etc.

Mitigate

Risk mitigation implies a reduction in the probability and/or impact of an adverse risk event to an acceptable threshold. Taking early action to reduce the probability and/or impact of a risk is often more effective than trying to repair the damage after the risk has occurred.

Risk mitigation may take resources or time and hence may represent a trade-off of one objective for another. However, it may still be preferable to going forward with an unmitigated risk.

Exploit

The organisation wishes to ensure that the opportunity is realised. This strategy seeks to eliminate the uncertainty associated with a particular upside risk by making the opportunity definitely happen. Examples include securing talented resources that may become available for the project.

Share

Sharing involves allocating ownership to a third party who is best able to capture the opportunity for the benefit of the project. Examples include: forming risk-sharing partnerships, teams, working with elected officials, special- purpose companies, joint ventures, etc.

Acceptance

A strategy that is adopted because it is either not possible to eliminate that risk from a project or the cost in time or money of the response is not warranted by the importance of the risk. When the project manager and the project team decide to accept a certain risk(s), they do not need to change the project plan to deal with that certain risk, or identify any response strategy other than agreeing to address the risk if and when it occurs. A workaround plan may be developed for that eventuality.

3.3.3 Issues and New Risks that may have a Change Impact on the Project

The following are possible issues and risk factors that may arise during your project, and that may have a change impact on the project:

Life Cycle Stage	Possible Risk Factors
All	You or your team spends insufficient time on one or more stages.
	Key information isn't in writing.
	You or your team moves to a subsequent stage without completing one or more of the earlier stages.
Starting the project	Some background information and/or plans aren't in writing.
	No formal benefit-cost analysis has been done.
	No formal feasibility study has been done.
	You don't know who the originator of the project idea is.
Organizing and preparing	People unfamiliar with similar projects prepare your project plan.
	Your plan isn't in writing.
	Parts of the plan are missing.
	Some or all aspects of the plan aren't approved by all key audiences.
Carrying out the work	People on the project team didn't prepare the plan.
	Team members who didn't participate in the development of the project plan don't review it.
	You haven't made an effort to establish team identity and focus.
	You haven't developed any team procedures to resolve conflicts, reach decisions, or maintain communication.
	Needs of your primary clients change.
	You have incomplete or incorrect information regarding schedule performance and resource expenditures.
	Project-progress reporting is inconsistent.
	One or more key project supporters are reassigned.
	Team members are replaced.
	Marketplace characteristics or demands change.
	Changes are handled informally, with no consistent analysis of their effect on the overall project.

Closing the project	Project results aren’t formally approved by one or more project drivers.
	Project team members are assigned to new projects before the current project is completed.
Part of Project Plan	Possible Risk Factors
Project audiences	Your project has a new client.
	You've had prior problems with a client.
	Upper management or other key drivers show only mild interest in your project.
	Your project doesn't have a project champion.
Project background	Your project derived from a spontaneous decision rather than a well- thought-out assessment.
	You don't have conclusive proof that your project will eliminate the problem it addresses.
	You don't have conclusive proof that your project will eliminate the problem it addresses. Your project can't start until one or more other planned activities are completed.
Project scope	Your project is unusually large.
	Your project requires a variety of skills and knowledge.
	Your project involves different organisational units.
Project strategy	You have no declared strategy.
	Your project involves a new, untested technology or approach.
Project objectives and deliverables	One or more objectives or deliverables are missing.
	Some performance measures are unclear or missing.
	Some performance measures are difficult to quantify.
	One or more performance targets or specifications are missing.
	One or more objectives or deliverables haven't been approved by all drivers.
Constraints	Your constraints aren't written down.
	Your constraints are vague.
Assumptions	Assumptions aren't written down.
	Your assumptions are vague.
Work packages and activities	Work packages or activities are insufficiently detailed.
	Not all team members participated in preparing descriptions of their assigned work packages and activities.

Roles and responsibilities	Not all supporters were involved in developing their roles and responsibilities.
	You have an overdependence on one or more people.
	No primary responsibility is assigned for one or more activities.
	Two or more people have primary responsibility for the same activity.
	No one person has overall responsibility for the project.
Schedule (activity-duration estimates)	Time estimates are backed into from an established end date.
	You have no historical database of activity durations.
	Your project involves new procedures or technologies for some activities.
	Activities are performed by team members you haven't worked with before.
Schedule (activity interdependencies)	Interdependencies aren't specifically considered during schedule development.
	Partially related activities are scheduled simultaneously to save time.
	Your project plan uses no formal analytical approach to assess the effect of interdependencies on the schedule.
Personnel	Your project plan has no estimates for actual work effort required to perform activities.
	Your project plan doesn't formally consider availability or efficiency.
	Your project plan has no detailed work schedules for people working simultaneously on two or more tasks.
	Your team includes one or more new or inexperienced team members.
Other resources	You have no plans to identify the type, amount, or timing of required nonpersonnel resources.
Funds	You have no project budget.

Study Unit 3.4: Manage the Development of the Products of the Project

US 243812

Monitor and control the execution of the project management plan for a simple to moderately complex project

Specific Outcome 4

Manage the development of the products of the project.

Assessment Criteria

summative assessment^ po£ '

1. Opportunity is provided to clarify product requirements with relevant stakeholders.
2. Product acceptance criteria are agreed upon and documented in appropriate level of detail.
3. Proposed product solutions are agreed with relevant stakeholders in appropriate level of detail and in the required time frame.
4. Product performance evaluations are scheduled and conducted with relevant stakeholders to ensure compliance with product requirements.

   ASSESSMENT CRITERION RANGE

   Performance measures should include but are not limited to progress of production of deliverables in terms of time, cost, quality, meeting scope and resource usage.
5. Defect resolution and/or change needs are identified and dealt with in accordance with project change control procedure

3.4.1 **Clarify Product Requirements Project Charter**

Project Charters outline project goals, and give an overview of how projects will look and feel while the project manager works on them. The project manager can use Project Charters in discussions with project team members, governance groups, and other stakeholders - either individually or as part of workshops - as a way of ensuring that everyone understands the project's requirements.

Writing a Project Charter forces the project manager to think through the project as a whole. He/she must understand all the existing project documentation, consider how he/she wants to approach implementing certain parts of the project, and identify the key points that everyone involved in the project needs to understand.

The project manager can also use a Project Initiation Document (PID) instead of a Project Charter for these purposes as they are very similar documents. However, a Project Initiation Document is usually much more detailed. So a Project Charter is more suited to projects where the project manager doesn't have the resources to write a detailed Project Initiation Document, or where he/she wants to start work on the project quickly. Format of a Project Charter

The project manager can deliver the Project Charter as a report or a presentation.

- In report format - Use this if the Project Charter must be self-explanatory. For example, the project manager may want to use it as a team reference document, to provide a baseline of what he/she and the stakeholders require and expect from the project. The project manager can then submit the Project Charter as part of the project approval process.
- In presentation format - Use this if the project manager presents the Project Charter as part of a discussion. For example, the project manager can use the charter to give a project overview to the team at a project meeting; or use it to brief the governance group and stakeholders at their first project meeting.

Develop Project Charter is the process of developing a document that formally authorises a project or a phase and documenting initial requirements that satisfy the stakeholder's needs and expectations. The charter links the project to the on-going work of the organisation and authorises the project.

Projects are chartered and authorised external to the project by the organisation, a programme or portfolio management body. In multi-phase projects, this process is used to validate or refine the decisions made during the previous iteration of Develop Project Charter.

3.4.2 Agree Upon and Document Product Acceptance

Product acceptance means to obtain acceptance of deliverables and final sign-off from the stakeholders that the project has been completed.

Preparing for Acceptance

From Project Initiation onwards, focus the project processes on what it will take to get acceptance from the stakeholders. Without a clear understanding of the dynamics of acceptance, the project will never be completed satisfactorily, on schedule and within budget.

Make sure that the Project Management Plan that was prepared in Project Initiation specifically

defined the acceptance process and criteria for each deliverable.

Specify Deliverables and Acceptance Criteria

Estimate and specify the size of each deliverable. For paper deliverables, provide a table of contents and identify the expected page count for each section. For custom software, reference the system specifications and estimate the size in terms of function points or some other appropriate software metric.

Measuring Project Size and Complexity

For each deliverable with specified quality requirements, define or make reference to the customer's expectations for quality (quality factors), the internal process and product attributes that will indicate whether the quality factors are being satisfied (quality criteria), and the measures to be used to give visibility to the levels of quality being achieved (quality metrics).

Define Quality Requirements

Define the acceptance process and acceptance criteria for each deliverable, including the name of the representative of the customer who will meet with you to resolve of all issues concerning this deliverable, respond to the deliverable, and accept the deliverable on behalf of the customer.

On a small project with a limited number of deliverables, prepare a blank Personal Acceptance Document for each deliverable and review it with the customer as part of Project Initiation so that the customer is aware of his/her role and responsibility. When the customer knows that he or she will be required to sign a Personal Acceptance Document to accept the deliverable, he or she is more likely to take the responsibility seriously throughout the development process. On a large project, with a very large number of deliverables, the same purpose can be accomplished using a Document Deliverables Directory.

Get agreement with the customer ahead of time (included in the contract) that non-response by the customer within the agreed-upon time period will automatically trigger a "deemed accepted" response.

Managing System Acceptance

Establish the expectation that high quality systems are typically not perfect. It is not reasonable to expect zero problem reports for a large system.

Managing Final Acceptance

Make sure that signed Personal Acceptance Documents have been received for all project deliverables. Understand that final sign-off may still require some negotiation.

3.4.3 Agree Proposed Product Solutions with Relevant Stakeholders

Managing the Acceptance of Deliverables

Deliverable review and approval is an ongoing process that includes reviewing each deliverable and identifying problems, making sure that all identified problems have been addressed, and formally signing-off on each completed deliverable to

indicate that it is ready for release to the customer. On small projects, this is the responsibility of the Project Manager. On large projects, this process is usually the responsibility of the Quality Assurance Manager, who provides the primary interface with the customer for deliverables acceptance. See also:

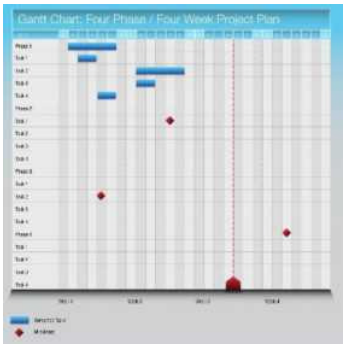
Deliverable Review and Approval Process

Schedule a deliverable review meeting with the customer for each deliverable, to discuss the customer's comments and agree on the changes to be made. Once agreed-upon customer changes have been incorporated, have the customer sign-off on the Personal Acceptance Document for each deliverable to formally document customer acceptance

3.4.4 Schedule and Conduct Product Performance Evaluations

Evaluation is an important tool to monitor a project. Although, final evaluations are often required by a sponsor, mid-term evaluations offer the advantage of answering many of the same questions asked during final evaluations, while providing the opportunity to make suggestions to improve the efficiency and impact of the project while the activities are still underway.

Common evaluation questions might include:



- Did the project succeed at accomplishing the outcomes, goals and impact desired?
- Was the project relevant, effective and efficient?
- Does the project have the potential to be sustainable in its operations and impact?
- Is the theory expressed in the logical framework upheld?

Ex-Post Evaluations

In addition, Ex-post evaluations examine project impact at a defined period of time after project completion, sometimes a year after the official close of the project. It is sometimes called a sustainable impact evaluation. An ex-post evaluation measures the extent to which project outcomes and impacts have been met. Ex-post evaluation findings can be an especially useful way of using evidence to improve the development approach.

Below you have a list of topics to consider for Ex-Post Evaluations:

- Stakeholders: who will be affected by, will be needed to support, or will be interested in the project outcome.
- Were specific roles and responsibilities assigned for staff members?
- Is anyone working on the project responsible for different areas? If so, is enough support provided so that they can complete their tasks?
- Do you need institutional and/or special approvals in order to start? If so, how will these approvals will affect your timeline/monitoring activities?
- Do you (any anyone else working in the project) understands any significant project risks and how they will be managed?
- Is everyone working on the project aware of the dates on which activities should be started and completed and milestones are reached?
- Do you have budgets for all required project resources (including, but not limited to, money)?
- Is management successfully controlling the performance of work, including:
    - Organizing, focusing, and continually motivating project personnel
    - Tracking and comparing project work and results against the project plan
    - Considering and making changes to plans when tracking suggests a change is called for

- Keeping everyone informed of project accomplishments, issues, and changes
- Continuously tracking and dealing with evolving project risk
- Organisation information systems can be used to support project planning and control, including the maintenance of records of:

■ The amount of work effort expended by people on project activities

■ The funds expended on project activities

3.4.5 Identify and Deal with Defect Resolution and/or Change Needs

This has been discussed in detail in Study Units 3.1 and 3.2

Study Unit 4.1: Project Quality Management Principles

US 243816

Develop a project quality management plan for a simple to moderately complex project

Specific Outcome 1

Demonstrate an understanding of project quality management principles.

Assessment Criteria

1. The purpose of quality management on a project is explained with examples illustrating the consequences of good and poor quality management on the end product.
2. The processes required to manage quality on a project are explained, with examples, in accordance with organisational standards and practices or recognised industry practice.
3. The components of a project quality management plan are explained with examples.
4. Quality assurance and its application on a project are explained with examples.
5. Quality control and its application on a project are explained with examples.
6. Continuous process improvement and its application on a project are explained with examples.
7. Cost-benefits analyses and their application in quality management are explained with examples.
8. Methods, tools and techniques for analysing and measuring quality needs and performance are described with examples of application to a project. ASSESSMENT CRITERION RANGE
   Methods, tools and techniques may include but is not limited to cause and effect diagrams, Taguchi, Pareto chart or diagram, statistical sampling, trend analysis, Six Sigma, control charts.

summative assessment

4.1.1 The Purpose of Quality Management on a Project

Quality Management is the management of the processes in an organisation that determine quality policies, objectives, and responsibilities so that project will satisfy the needs for which it is undertaken.

The main purpose of project quality management is to ensure that the project will meet or, even better, exceed stakeholder's needs and expectations. The project team must develop a good relationship with key stakeholders, especially the owner and the beneficiaries of the project, to understand what quality means to them. One of the causes for poor project evaluations is that the focus of the project is to only meet the written requirements for the main outputs and ignores other stakeholder needs and expectations for the project.

Quality must be treated on the same level of importance as with scope, schedule and budget. If a project owner is not satisfied with the quality of how the project is delivering the outcomes, the project team will have to make adjustments to scope, schedule and budget to satisfy the owner's needs and expectations. To deliver the project scope on time and on budget is not enough. It is very important to develop a good working relationship with all stakeholders and understand their stated or implied needs so that stakeholder satisfaction on the project can be met.

Consequences of Good and Poor Quality Management on the End Product

Poor quality management can cause the following:

Delay in the delivery

The output has been rejected by the quality control department and now the work must be reprocessed to meet the quality standard. To rework the process will take time and the delivery is going to be delayed. It is therefore important for a project manager to avoid the schedule slippage by maintaining the optimum quality.

Customer Dissatisfaction

After screening the customer noticed some quality problem - it is a huge problem for the customer as well as for your organisation. The customer lose faith in your organisation will not give you any new work. Business is lost as the customer is not satisfied with the delivered quality. Note that a project manager can't compromise on the quality, as it is ethically and totally unacceptable. Quality should be kept in higher priority and it is a project manager's responsibility to keep the customer happy on the quality. Customer satisfaction is required for a successful project

Increase in the cost

When a customer rejected the output you and your team are working again on that process - it means that the cost for the project will increase. It is the responsibility of the project manager to prevent this additional cost due to unacceptable quality. No organisation will simply accept increase in cost because of poor quality. As quality is a project constraint, it should be balanced throughout the life cycle of the project. It is everyone in the organisation's responsibility to maintain good quality is and a project manager should take the responsibility to create and update the best practices pertaining to the good quality.

Low morality

If poor quality is delivered to a customer, the morality will be very down. And it goes both ways: both the team and the customer will be in the low morality. And in that way there is always possibility of losing the work from that customer. So to maintain the high morality throughout, it is important to maintain the good quality every time.

More consequences of good and poor quality management are:

Cost of Conformance	Cost of Non-conformance
Prevention Costs • Training • Document Processes • Equipment • Time To Do It Right	Internal Failure Costs • Rework • Scrap
Appraisal Costs • Testing • Destructive Testing Loss • Inspections	External Failure Costs • Liabilities • Warranty Work • Lost Business

4.1.2 The Processes Required to Manage Quality on a Project

Project management consists of four main processes:

- Quality Planning
- Quality Assurance
- Quality Control
- Quality Improvements

Inputs	Process	Outputs
WBS	Plan: Define the standards for quality	Quality Standards
Scope Statement	Do: Assurance that quality is followed	Quality Plan
Policies	Check: Quality control	
	Adapt: Quality improvements	

The first component, quality planning, involves the preparation of a quality management plan that describes the processes and metrics that will be used. The relevant stakeholders must agree on the quality management plan to ensure that their expectations for quality are correctly identified. The processes that are described in the quality management plan should conform to the processes, culture and values of the host organisation.

Quality assurance provides confidence to the host organisation that its projects, programmes and portfolios are being managed well. It validates the consistent use of procedures and standards, and ensures that all the employees working on the contract have the necessary knowledge, skills and attitudes to fulfil their project roles and responsibilities competently. Quality assurance must be independent of the project, programme or portfolio to which it applies.

The next component, quality control, consists of inspection, testing and measurement. It verifies that the deliverables meet

stakeholder expectations, conform to specification, and are fit for purpose. Quality control activities determine whether acceptance criteria have, or have not, been met. For this to be effective, there must be strict configuration control over the specifications. It is possible that, once agreed, the specification may have to be modified. Usually this is to accommodate change requests or issues, while maintaining acceptable time and cost constraints. Any consequent changes to acceptance criteria should be approved and communicated.

The last component, continual improvement, is the term used by organisations to describe how information that is provided by quality assurance and quality control processes is used to drive improvements in efficiency and effectiveness. A P3 maturity model provides a framework against which continual improvement can be initiated and embedded in the organisation.

4.1.3 The Project Quality Management Plan

The Purpose of a Project Quality Plan

Quality can be defined as meeting the customer's expectations or exceeding the customer expectations achieved by way of deliverables and/or activities performed to produce those deliverables.

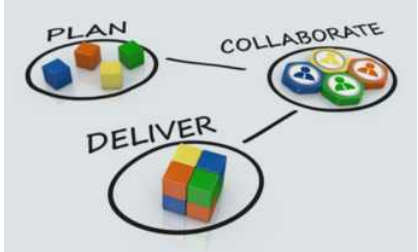
A Project Quality Plan is a set of activities that are planned at the beginning of the project that helps achieve quality in the project being executed. The purpose of the Project Quality Plan is to define the activities / tasks that are necessary to deliver products, while at the same time focussing on achieving customer's quality expectations. These activities / tasks are defined on the basis of the quality standards set by the organisation that delivers the product.

The Project Quality Plan identifies which Quality Standards apply to the project and determines how can they be satisfied. It includes the implementation of Quality Events (peer reviews, checklist execution) as well as the use of different Quality Materials (templates, standards, checklists) available within the organisation. The holding of the Quality Event is also called Quality Control. As an output of the different activities, Quality Metrics or Measurements are captured which assist in continuous improvement of quality, and that adds to the inventory of Lessons Learned. Quality Assurance deals in preparation of the Quality Plan and formation of standards in the organisation.

The Quality Management Plan helps the project manager determine if deliverables are being produced to an acceptable quality level and if the project processes used to manage and create the deliverables are effective and properly applied

Guidelines to write the Project Quality Management Plan

Project Quality Plan should be written with the objective to provide easy access to the project management to the quality requirements. It also should have the procedures and standards ready available.



The following is a list of the different Quality Elements that should be included in a detailed Project Quality Plan:

- Management Responsibility: This section describes the quality responsibilities of all stakeholders.
- Documented Quality Management System: This refers to the existing

standardised Quality Procedures that have been used within the organisation.

- Design Control: This section specifies the procedures for Design Review, Sign-Off, Design Changes and Design Waivers of requirements.
- Document Control: This defines the process to control the various Project Documents at each Project Phase.
- Purchasing: This section states the Quality Control and Quality Requirements for sub-contracting any part / whole part of the project.
- Inspection Testing: This details the plans for Acceptance Testing and Integration Testing.
- Non-conformance: This defines what procedures will be followed to handle any type of nonconformance work. The procedures include defining the responsibilities, the conditions and the availability of required documentation in such cases.
- Corrective Actions: This describes the procedures when taking Corrective Actions for the problems that were encountered during project execution.
- Quality Records: This describes the procedures that must be followed to maintain the Quality Records (metrices, variance reports, executed checklists etc.) during project execution as well as after completion of the project.
- Quality Audits: During each phase of the project an internal audit should be planned and implemented.
- Training: This should specify any training requirements for the project team.

People involved in writing the Project Quality Management Plan

- Project Manager
- Project Team
- Customer
- Project Sponsor

4.1.4 Quality Assurance and its Application on a Project

Assurance is the activity of providing the evidence that will create confidence among all stakeholders that effective quality-related activities are being performed; and that all planned actions are being done to ensure that there is adequate confidence that a product or service will satisfy the stated requirements for quality.

Quality Assurance is a process that provides confirmation based on evidence to ensure to the donor, beneficiaries, organisation management and other stakeholders that the product will meet needs, expectations, and other requirements. It assures that there are effective processes and procedures, and that safeguards are in place to make sure that the expected levels of quality will be reached to produce quality outputs.

Quality assurance takes place during the implementation phase of the project and includes the evaluation of the overall performance of the project on a regular basis to provide confidence that the project will meet the quality standards as defined by the project.
One of the purposes of quality management is to find errors and defects as early in the project as possible. For that reason, a good quality management process will take more effort hours and cost upfront. The goal is to minimise the chances that products or services will be of poor quality after the project has been completed.

Quality assurance is not only done to the products and services delivered by the project but also to the process and procedures used to manage the project. That includes the way in which the project uses the tools, techniques and methodologies to manage scope, schedule, budget and quality. Quality assurance must also make sure that the project

meets any legal or regulatory standards.

Quality Audits

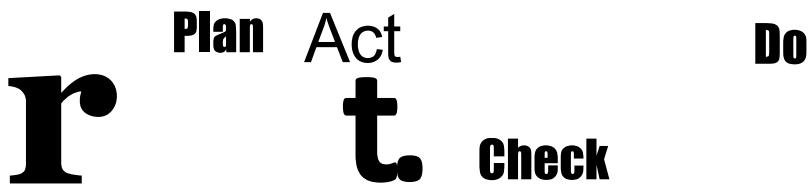
Quality audits are structured reviews of the quality management activities that help to identify lessons learned that can improve the performance on current or future project activities. Audits are done by project staff or consultants with expertise in specific areas. The purpose of quality audit is to review how the project uses its internal processes to produce the products and services it will deliver to the beneficiaries. Its goal is to find ways to improve the tools, techniques and processes that create the products and services.

If problems are detected during the quality audits, corrective action must be done to the tools, processes and procedures that are used to ensure that quality is re-established. Part of the audit may include a review of the project staff understanding of the quality parameters or metrics, and skills expertise and knowledge of the people in charge of producing or delivering the products or services. If corrective actions are necessary, they must be approved through the change control processes.

The PDCA Cycle

The most popular tool used to determine quality assurance is the Shewhart Cycle. This cycle for quality assurance consists of four steps: Plan, Do, Check, and Act. These steps are usually abbreviated as PDCA. The four quality assurance steps within the PDCA model stand for:

- Plan: Establish objectives and processes that are needed to deliver the desired results.
- Do: Implement the process developed.
- Check: Monitor and evaluate the implemented process by testing the results against the predetermined objectives.
- Act: Apply actions that are necessary for improvement if the results require changes.



The PDCA is an effective method to monitor quality assurance because it analyses existing conditions and methods that are used to provide the product or service to beneficiaries. The goal is to ensure that excellence is part of every component of the process. Quality assurance also helps to determine whether the steps that are used to provide the product or service is appropriate for the time and conditions. In addition, if the PDCA cycle is repeated throughout the lifetime of the project it helps to improve internal efficiency.

Quality assurance demands a degree of detail in order to be fully implemented at every step. Planning, for example, could include investigation into the quality of the raw materials that are used in manufacturing, the actual assembly, or the inspection processes used.

The Checking step could include beneficiary feedback or surveys to determine if all the beneficiary needs are being met or exceeded and why they are or are not. Acting could mean a total revision in the delivery process in order to correct any flaws. The goal to exceed stakeholder expectations in a measurable and accountable process is provided by quality assurance.

Assurance vs. Control

Quality assurance is often confused with quality control. Quality control is done at the end of a process or activity to verify that quality standards have been met. Quality control by itself does not provide quality, although it may identify problems and suggest ways to improve the quality. On the other hand, quality assurance is a systematic approach to ensure that quality standards are obtained.

Quality assurance is something that must be planned for from the earliest stages of a project, with corrective measures taken at every stage.

4.1.5 Quality Control and its Application on a Project

Quality control is the use of techniques and activities that compare the actual quality performance with the goals that were set, and then define appropriate action in response to a shortfall. It is the process that monitors specific project results to determine if they comply with specific standards and then identifies different approaches to eliminate the causes for the unsatisfactory or poor performance.

The goal of quality control is to improve quality. It involves the monitoring of the project outputs to determine whether they meet the quality standards or definitions based on the project stakeholder's expectations. Quality control also includes how the project performs in its efforts to manage scope, budget and schedule

- Acceptance: The beneficiaries, the donor or other key project stakeholders accept or reject the product or service that was delivered. Acceptance occurs after the beneficiaries or donor has had a change to evaluate the product or service
- Rework: The action that is taken to bring the rejected product or service into compliance with the requirements, quality specifications or stakeholder expectations. Rework is expensive, that is why it is so important to do a good job in quality planning and quality assurance to avoid the need for rework. Rework and all the costs associated with it is usually not refundable by the donor and the organisation may end up covering those costs.
- Adjustments: Correct or take the necessary steps to prevent further quality problems or defects based on quality control measurements. Adjustments are identified to the processes that produce the outputs and the decisions that were taken that lead to the defects and errors. Changes are taken to the Change Control processes of the project

4.1.6 Continuous Process Improvement and its Application

Quality Improvement

It is the systematic approach to the processes of work with the aim to remove waste, loss, rework, frustration, etc. in order to make the processes of work more effective, efficient, and appropriate.

Quality improvement refers to the use of methods and tools to close the gap between current and expected levels of quality by understanding and addressing system deficiencies and strengths to improve, or in some cases, redesign project processes.

There are a variety of quality improvement approaches exists, which range from individual performance improvement to redesign of entire project processes. These approaches are different with regards to time, resources, and complexity, but share the same four steps in quality improvement:

- Identify what you want to improve: The project uses the data found in the quality control process to identify the areas that need improvement.
- Analyse the problem or system: The team then investigates the causes for the problem and its implications to the project, the causes may be internal or external to the project.
- Develop potential solutions or changes that appear likely to improve the problem or system: The team brainstorms ideas and potential solutions to the problem, taking in consideration its impact to the project schedule and budget. After careful considerations the team decides and chooses the best alternative.
- Test and implement the solutions: The team may decide to test the solution on a small scale to verify that it can in fact fix the problem, it testes for the initial assumptions made about the problem and once it confirms that the solution is a viable alternative, it then implements the solution in a full scale.

Continuous Improvement

Quality is a continuous process to ensure quality is performed in all aspects of the project, therefore it is not something that is done at the end of a phase or at the end of the project. The goal is to continuously improve based on the lessons learned and new insights provided by the project. To be effective it should happen during all activities of the project.

With regards to project quality, continuous improvement always focuses on improving stakeholder satisfaction through continuous and incremental improvements to processes. This includes to remove any unnecessary activities. By applying a process that continuously improves every element of the project, it can achieve better results than trying to wait until the end of a phase or a midterm evaluation to start making changes and improvements to the work. It requires little effort and by doing small incremental improvements the project can reach significant levels of quality.

To implement continuous improvements, it is necessary to have a culture of reflection that allows the project team to learn from mistakes and apply the lesson on the next phase or cycle and not spend time and effort trying to put blame, If this is not the case, the team will be too scared to report any problems with quality and it will be too late to do anything once the donor or the beneficiaries find out.

4.1.7 Cost-Benefits Analyses

A cost-benefit analysis is a process use to analyse business decisions. The benefits of a given situation or business- related action are calculated and then the costs associated with taking that action are subtracted. Performing one is critical to any project. When performing a cost-benefit analysis, you make a comparative assessment of all the benefits that are anticipated from the project and all the costs to introduce the project, perform it, and support the changes resulting from it.

Cost-benefit analyses help you to

- Decide whether to undertake a project or decide which of several projects to undertake.
- Frame appropriate project objectives.
- Develop appropriate before and after measures of project success.
- Prepare estimates of the resources required to perform the project work.

Some anticipated benefits can be expressed in monetary equivalents (such as reduced operating costs or increased revenue). For other benefits, some, but not all, aspects can be calculated in numerical measures. If the project is to improve staff morale, for example, associated benefits are usually included, like reduced turnover, increased productivity, fewer absences, and fewer formal grievances. Whenever possible, it is good to express benefits and costs in monetary terms to facilitate the assessment of a project's net value.

Consider costs for all phases of the project. Such costs may be non-recurring (such as labour, capital investment, and certain operations and services) or recurring (such as changes in employees, supplies, and materials or maintenance and repair).

Furthermore, the following should be considered:

- Potential costs of not doing the project
- Potential costs if the project fails
- Opportunity costs (in other words, the potential benefits if funds were spent successfully performing a different project)

Cost-Benefit Analysis: Weighing Future Values Today

The farther into the future you look when performing the analysis, the more important it is to convert the estimates of benefits over costs into today's rands. Unfortunately, the farther you look, the less accurate the estimates can be. For example, you may expect to reap benefits for years from a new computer system, but changing technology may make your new system obsolete after only one year.

Thus, the following two key factors influence the results of a cost-benefit analysis:

- How far into the future you look to identify benefits

- On which assumptions the analysis is based

The net present value (NPV) is based on the following two premises:

- Inflation: The purchasing power of a rand will be less one year from now than it is today. If the rate of inflation is 3 percent for the next 12 months, R1 today will be worth 97 cents just 12 months from today. In other words, 12 months from now, you'll pay R1 to buy what you paid 97 cents for today.
- Lost return on investment: If money is spent to perform the project, you'll forego the future income you could earn by investing it conservatively today. For example, if you put R1 in a bank and receive simple interest at the rate of 3 percent compounded annually, 12 months from today you'll have R1.03 (assuming zero-percent inflation).

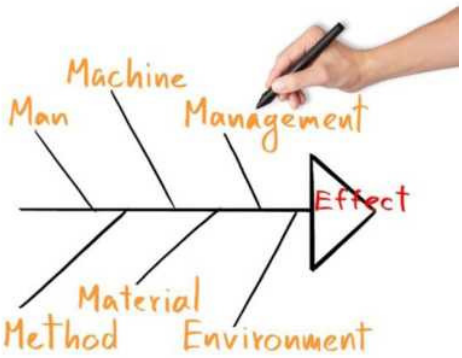
To address these considerations when determining the NPV, you specify the following numbers:

- Discount rate: The factor that reflects the future value of R1 in today's dollars, considering the effects of both inflation and lost return on investment
- Allowable payback period: The length of time for anticipated benefits and estimated costs

In addition to determining the NPV for different discount rates and payback periods, it is also important to calculate the project's internal rate of return for each payback period.

4.1.8 Methods, Tools and Techniques for Analysing and Measuring Quality Needs and Performance

Cause and Effect Diagrams



Cause and Effect Diagram is also known as fishbone diagrams or Ishikawa diagrams (named after Kaoru Ishikawa, a Japanese quality control statistician, who developed the concept in the 1960s), and is regarded as one of the seven basic tools of quality management. It is named fishbone diagram because of its fish-like appearance. It is an analysis tool that provides a systematic way of looking at effects and the causes that lead or contribute to those effects. The Cause and Effect Diagram is used by a problem-solving team as a tool to assemble all inputs, like what are the causes of the problem they're addressing, systematically and graphically. Such inputs usually come from a brainstorming session. It helps the team to focus on the reason why the problem occurs, and not on the history or symptoms of the problem, or other topics that digress from the reason for the session. It also displays a real-time 'snap-shot' of the collective inputs of the team as it is updated. The possible causes are written down at various levels of detail in connected branches, with the level of detail increasing as the branch goes outward, i.e., an outer branch is a cause of the inner branch it is attached to. Thus, the outermost branches usually indicate the root causes of the problem.

Taguchi

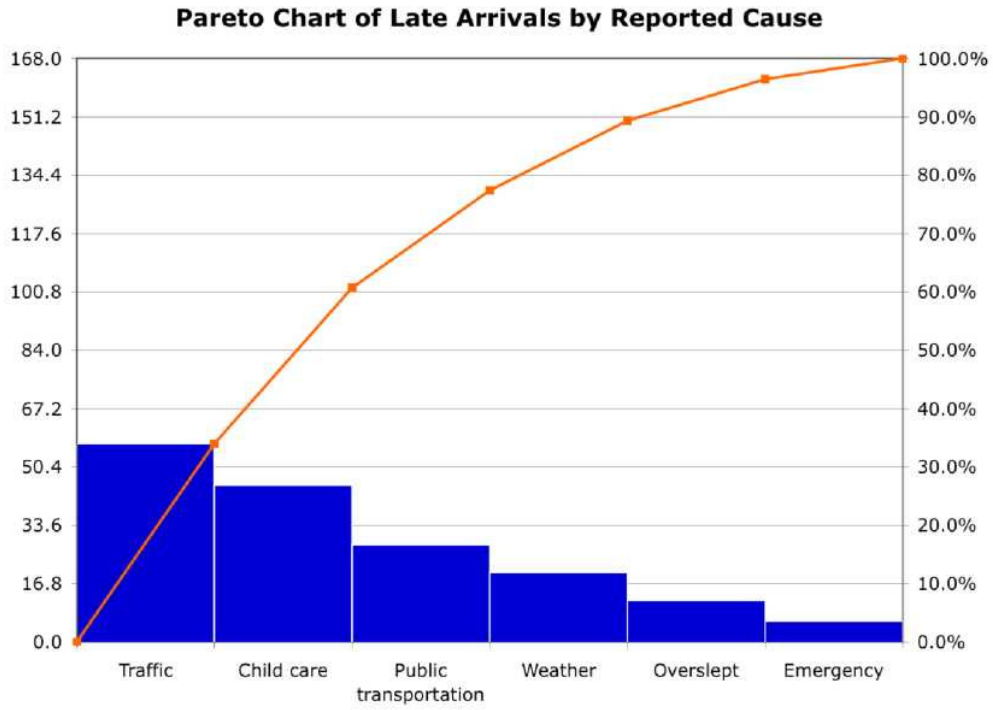
The Taguchi method of quality control is an approach that emphasises the roles of research and development, product design and product development in reducing the occurrence of defects and failures in products. According to the Taguchi

method design is more important than the manufacturing process in quality control and it tries to eliminate variances in production before they can occur.

Pareto Chart or Diagram

This is based on Pareto's rule, which states that 80 percent of the problems are often due to 20 percent of the causes. I assumes that most of the results in any situation are determined by a small number of causes and helps to identify the important few contributors that account for most quality problems. The chart is a form of

histogram that arranges the data by frequency of occurrence; it shows how many defects were generated by a type of category of identified cause. For example to determine the errors in the collection of beneficiary data the project team identified five causes and for each cause the frequency they contained errors, the data is plotted, the bars represent each category and the line the cumulative percentage of the errors, the chart allows to identify that 80% of the errors could be reduced just by improving the collection of data in two categories instead of focusing efforts to correct all categories.



en.wikipedia.org

Statistical Sampling

To determine whether a group of deliverables meets the quality requirements, it is not necessary to test every single one. In many instances, a sample of them can be tested and then determine the viability of the entire group based on the outcome of the sample. Say the Project Management Office Director wanted to determine how well all the project managers are following the change control policies.

If 15 project managers were managing 42 projects, the director could randomly select five change control logs from different project managers for different projects and infer the performance of the rest of the projects based on the results of those five change control logs.

When sampling, the following can be used:

- Attribute sampling: Indicates whether the result conforms.
- Variable sampling: Indicates the degree to which the result complies as rated on a continuous scale

Trend Analysis

During Project Management, it is sometimes necessary to identify trends and this is called Trend Analysis. This is the part of Project Management where historical data is used, given a set of mathematical parameters, and then processed in order to determine any possible variance from an established baseline of given constraints such as Budget, Cost, Schedule, or Scope. These parameters are modelled after data from reporting over certain periods, and is used during the Project Management process to project how much variance a project manager can expect if the project is not adjusted in some way during an estimated period of time.

The aim of this to anticipate a specific trend and make the necessary adjustments during the Project Management process in order to maximise the full potential of desired outcomes. Using historical data is a proven method to be able to identify these trends. Trend Analysis is a mathematical scientific approach and eliminates potential error by using precise calculations in order to provide the utmost accuracy. It is the most dependable and efficient method to anticipate possible future behaviour and desired outcome of a specific project.

This term is defined in the 3rd and the 4th edition of the PMBOK.

Six Sigma

Six Sigma is a business management tool developed by the Motorola Organisation in the mid 80's. It seeks to improve the quality output and efficiency of organisations by identifying the probable defects in a given process and minimising the variability in output. This process uses a set of methods including statistical methods to create an infrastructure of people within the organisation. Each Six Sigma project process in an organisation follows a specific sequence with the aim of reducing cost and maximising profit.

Why "Sigma"? The word is a statistical term that measures how far a given process deviates from perfection. The central idea behind Six Sigma is that if you can measure how many "defects" you have in a process, you can systematically figure out how to eliminate them and get as close to "zero defects" as possible. To achieve Six Sigma Quality, a process must produce no more than 3.4 defects per million opportunities. An "opportunity" is defined as a chance for non-conformance, or not meeting the required specifications. This means we need to be nearly flawless in executing our key processes.

Lean Six Sigma

The root of both Lean and Six Sigma reach back to the time when the greatest pressure for quality and speed were on manufacturing. Lean rose as a method for optimising automotive manufacturing; Six Sigma evolved as a quality initiative to eliminate defects by reducing variation in processes in the semiconductor industry. It is not surprising that the earliest adopters of Lean Six Sigma arose in the service support functions of manufacturing organisations like GE Capital, Caterpillar Finance, and Lockheed Martin.

Key Concept

In short, what sets Lean Six Sigma apart from its individual components is the recognition that you cannot do "just quality" or "just speed," you need the balanced process that can help an organisation to focus on improving service quality, as defined by the customer within a set time limit.

Lean Six Sigma for services is a business improvement methodology that maximises shareholder value by achieving the fastest rate of improvement in customer satisfaction, cost, quality, process speed, and invested capital. The fusion of Lean and Six Sigma improvement methods is required because:

- Lean cannot bring a process under statistical control
- Six Sigma alone cannot dramatically improve process speed or reduce invested capital
- Both enable the reduction of the cost of complexity

Ironically, Six Sigma and Lean have often been regarded as rival initiatives. Lean enthusiasts note that Six Sigma pays little attention to anything related to speed and flow, while Six Sigma supporters point out that Lean fails does not address key concepts like customer needs and variation. Both sides are right. Yet these arguments are more often used to advocate choosing one over the other, rather than to support the more logical conclusion that we blend Lean and Six Sigma.

Six Sigma Principles

- Emphasises the need to recognise opportunities and eliminate defects as defined by customers
- Recognises that variation hinders our ability to reliably deliver high quality services
- Requires data driven decisions and incorporates a comprehensive set of quality tools under a powerful framework for effective problem solving
- Provides a highly prescriptive cultural infrastructure effective in obtaining sustainable results
- When implemented correctly, promises and delivers

Lean:

- Focuses on maximising process velocity
- Provides tools for analysing process flow and delay times at each activity in a process
- Centers on the separation of "value-added" from "non-value-added" work with tools to eliminate the root causes of non-valued activities and their cost

Control Charts

This is a graphical display of data that illustrates the results of a process over time. The purpose of a control chart is to prevent defects, rather than detect them or reject them. The chart helps to determine whether a process is in control or out of control over specified length of time. Control charts are often used to monitor the production of large quantities of products, but can also be used to monitor the volume and frequency of errors in documents, cost and schedule variances and other items related to project quality management. T

Control charts can also be used to the project management areas, such as schedule and budget control, to determine whether the costs variances or schedule variances are outside the acceptable limits set by the donor.

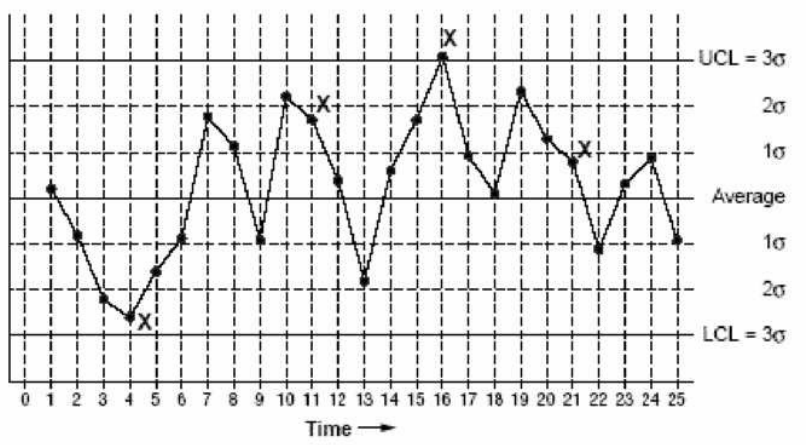
Control Chart Basic Procedure

1. Choose the correct control chart for your data.
2. Determine the correct time period to collect and plot data.
3. Collect data, construct the chart and analyse the data.
4. Look for "out-of-control signals" on the control chart. When one is identified, mark it on the chart and investigate the cause. Document how you investigated, what you learned, the cause and how it was corrected.

Out-of-control signals

- A single point outside the control limits. In the graph below, point sixteen is above the UCL (upper control limit).
- Two out of three successive points are on the same side of the centerline and farther than 2 a from it. Point 4 sends that signal.

- Four out of five successive points are on the same side of the centerline and farther than 1 a from it. Point 11 sends that signal.
- A run of eight in a row are on the same side of the centerline. Or 10 out of 11, 12 out of 14 or 16 out of 20. Point 21 is eighth in a row above the centerline.
- Obvious consistent or persistent patterns that suggest something unusual about your data and your process.



- Continue to plot data as they are generated. As each new data point is plotted, check for new out-ofcontrol signals.
- When a new control chart is started, the process may be out of control. If so, the control limits calculated from the first 20 points are conditional limits. When you have at least 20 sequential points from a period when the process is operating in control, recalculate control limits

Study Unit 4.2: Quality Requirements for a Project

US 243816

Develop a project quality management plan for a simple to moderately complex project Specific Outcome 1

Gather and analyse quality requirements for a project.

Assessment Criteria

5.

1. Participants in defining quality requirements for the project are identified and involved in the project quality planning process.
2. Project quality objectives are determined with input from the relevant stakeholders and documented in the required format.
   Organisational quality policies, procedures and standards are identified and reviewed for applicability.
3. Established quality management methods, tools and techniques are identified and reviewed for applicability.
   Requirements for a project quality management system are determined in support of the project quality objectives.
4. ASSESSMENT CRITERION RANGE
   The project quality system may include but is not limited to: organisation structure, responsibilities, procedures, processes, resources, tools, check lists, operational definitions.
6. Lessons learned from previous project are identified and reviewed for applicability.
7. Product quality requirements and criteria are identified with input from the relevant stakeholders and documented in the required format.

4.2.1 Participants in Defining Quality Requirements for the Project

The following are participants in a project, and are also responsible to define the quality requirements for a project:

- Projects' Board
- Steering Committee
- Project owner
- Project sponsor
- Project manager
- Project administrator
- Systems Developer
- Systems Administrator

Each of these will be discussed in more detail below:

Projects' Board

1. Identifies and initiates projects promoting the strategies of the University
2. Monitors the progress of the project
3. Prioritises funds and other resource allocations
4. Solves disputes and addresses problems that have escalated to this level

Steering Committee

Consists of:

1. Project owner
2. Key stakeholders
3. Project manager (reporting)

Responsibilities

1. Ensures that projects are aligned with corporate objectives.
2. Indicates strategic direction.
3. Keeps the focus on the client and business needs.
4. Ensures that the environmental impact is taken into account (internal and external)
5. Solves disputes that have escalated to this level.
6. Provides the decision-making forum for problems and issues in the project.
7. Ensures that the project has sufficient resources to successfully achieve its objectives.
8. Approves changes.
9. Maintains the commitment to the project objectives.

Project Owner

The project owner is the person who:

1. Stands to win or lose the most in terms of the outcome of the project
2. Accepts full authority for the project
3. Accepts accountability for the performance of the project (and who wants to do the project)
4. Provides resources.

Responsibilities

1. Gives guidance on the overarching vision and strategy in terms of the project.
2. Ensures that project objectives remain aligned with corporate and business needs.
3. Identifies the project manager.
4. Supervises the processes, procedures, budget and monitoring.
5. Reacts pro-actively on potential risks.
6. Reacts on issues (i.e. risks that materialised).
7. Ensures that the project has the required resources.
8. Helps to eliminate conflict of resources.
9. Manages the benefits of the project (after completion of the project)

Project Sponsor

1. Acts as champion of the project.
2. Is accountable for the delivery of planned benefits associated with the project.
3. Ensures resolution of issues escalated by the Project Manager or the Project Board.
4. Sponsors the communications programme; communicates the programme's goals to the organisation as a whole.
5. Makes key organisation/commercial decisions for the project.

6. Assures availability of essential project resources.
7. Approves the budget and decides tolerances.
8. Leads the Project Board.
9. Ultimate authority and responsibility for the project.

Project Manager

1. Identify the project's stakeholders

a) Obtain the buy in of key stakeholders

b) Manage the relationship with the stakeholder or client (in collaboration with the project owner)

2. Guide and manage the project team

a) Select the core team in collaboration with the project owner

b) Determine the role and responsibilities of each team member and obtain agreement

c) Work directly with the project administrator (PA)

d) Provide regular, informal feedback to each team member

e) Formally review each team member's performance, as agreed with the team member

f) Arrange team building sessions

g) Acknowledge and reward performance and expectations exceeded.

3. Manage the project

a) Define the project

b) Identify and manage the risks

c) Manage the communication plan

d) Allocate and obtain resources

e) Monitor project progress

f) Report on project progress

g) Solve problems that interfere with project progress

h) Identify changes to the scope of the project (change management)

i) Control costs

j) Deliver project deliverables achieving the expected quality standards

Project Administrator

General administration

1. Maintaining the distribution lists for the project concerned

2. Setting up and maintaining the project network space (provide for project-specific information, e.g. history, project management information)

3. Maintaining electronic copies of project files

4. Co-ordinating the logistical needs and needs of new staff members

5. Scheduling meetings and workshops on request

6. Consolidating information for project meetings, as requested by the project manager
7. Distributing information to nominated persons
8. Arranging social functions
9. Taking telephone messages
10. Negotiating with other project managers regarding requests for information
11. Liaising with the project support office regarding the standardisation of information, guidelines, etc.

Maintenance of the project plan

1. Receiving project plan from project manager
2. Maintaining project plan - NB! Any changes to the project plan may only take place on the basis of an approved change request.
3. Drawing up reports, e.g. status report and report of overdue work
4. Distributing status and progress report to the approved distribution list

Financial administration

1. Keeping the project budget updated
2. Providing financial reports (budget vs actual expenditure)

Administration of change requests

1. The team member sends the change request to the project administrator
2. Recording the change request in the change request statement
3. Allocating a number to the request
4. Arranging a meeting to discuss the request or add a permanent item to the agenda of the regular meetings for dealing with change requests.
5. Preparing the documentation for the meeting
6. Updating the change request statement with the outcome of the meeting (e.g. approved, or first do an impact analysis)
7. Updating the project plan after the request has been approve

Systems Developer

1. Working with the Project Manager on definition of development requirements and priorities.
2. Data Migration.
3. Interfaces with other systems.
4. Reporting configuration and deployment.
5. Set up and maintenance of security rights and access permissions.
6. Contributing to technical strategy, policy and procedure.
7. Development and operation of technical testing programmes.
8. Production of technical documentation to agreed quality standards.
9. Reporting on progress/issues to management and users

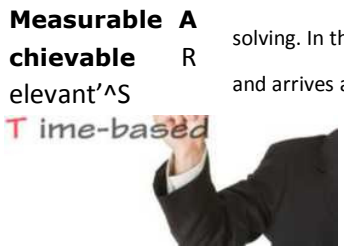
Systems Administrator

1. Management and support of the various environments.
2. Network operating systems management and support.
3. Database management and support.
4. Back-up and disaster recovery measures.
5. Contributing to technical strategy, policy and procedure.
6. Development and operation of technical testing programmes.
7. Production of technical documentation to agreed quality standards.

4.2.2 Project Quality Objectives

When gathering requirements for a project, a manager should not only specify what is being developed (scope) and when it will be delivered (time).It is important to also plan quality measures for each deliverable, which contributes towards the end product or service.

Quality management can be seen as answering the "how" part of problem solving. In the planning process, a project manager assesses product/service specifications and arrives at S.M.A.R.T (Specific,



Measurable, Attainable, Realistic, and Time Based) quality criteria for each deliverable. These plans must then be executed throughout the project lifecycle (via testing, inspections, walkthroughs etc.). As the project manager controls and monitors the project, it may be necessary to modify and correct product/service specifications and plans to achieve better quality.

Lastly, the project manager conducts an audit of product/service quality as the project reaches closure. A key concern of the project manager at this stage is to have stakeholders formally accept the final product/service through achieving a sign-off document. If quality planning and execution are done properly within a project, it ensures a better end-product to stakeholders

ISO 9000

According to the ISO 9000:2000 Standard [Clause 5.4.1]:

> "Top Management shall ensure that Quality Objectives, including those needed to meet requirements for product, are established at relevant levels within the organisation. The quality objectives shall be measurable and consistent with the quality policy."

IEEE ISO 9000 indicates that a quality management system and continual improvement of the system must be supported by the collection of appropriate data, analysis of the data and reporting the results.

In ISO 9000:2000, the definition of a Quality Objective is: *"Something sought, or aimed for, related to quality."*

The key concept here is that if you aim for something, you need to know if you achieved it, therefore, the objectives should be measurable.

Quality Objectives should be associated with key elements of quality, such as:

- Fitness for use
- Performance
- Safety
- Reliability
- Conformance to standards

Examples of quality objectives

Quality objectives can be very specific or generalised - but they should always be measurable!

Specific Objective: (sometimes known as the quality criteria) To achieve a 95 percent confidence level that data displayed online is an accurate representation of the current stock levels.

General Objective: To upgrade the technology knowledge of all personnel through continuous improvement training

It is common for quality objectives to be a general statement - but related to a specific area.

4.2.3 Organisational Quality Policies, Procedures and Standards

In strategic planning of project quality the development of a policy that could state high-level goals and general direction the project quality management process is very important. A project quality policy is an effective tool of quality management to identify the right way for an organisation to implement a project which follows expected quality levels and performance.

Project Quality Policy Definition

Project Quality Policy is a statement of a high-level strategic governance model used to identify project quality goals and constitute the general direction of the quality management process. It is a strategic vision of an organisation with regards to formal expectations and requirements of project quality that must be achieved during the process implementation. The project quality policy statement gives a common description of all low-level (tactical) elements of the governance model, such as standards, plans, guidelines, approaches, activities and procedures.

The project quality policy statement is often a document that is never changed. The development of the project quality policy document is the most important process within project quality planning and management that is undertaken at the Project Planning Phase when the plans to manage a project are to be designed and approved. The quality policy statement is a basis for developing a project quality management plan which is a series of tasks and activities to put the policy in practice and produce operational results.

During the development of your quality policy statement it is important to take into account direction and expectations of senior management regarding desired quality of project products and processes. Several approaches to the implementation of your policy should also be considered. These approaches are reviewed below.

Project Quality Policy Approaches

To achieve successful implementation of the project quality policy statement, it is important to adapt and apply a series of approaches during the course of the process for project quality planning and management. Quality policy approaches can be divided into four groups, such as Human Resources, Quality Reviews, Process and Performance.

Human Resources

Human resources are the greatest asset of the project. Organisations need to consider employees a resource that is worth investments because by having qualified and experienced employees the organisation can follow the best project quality management practices and implement competent solutions in the field of quality control and assurance. This approach should be a priority in the project quality policy statement and must serve as a

motivation for employees to use their full potential and achieve the highest possible quality level. Project quality management software will be required for designing human resource investment estimates.

Employee feedback ensures that employees are actively encouraged to send their feedback to the project quality management team and report current level of performance. The employee feedback should be used to implement the stated project quality policy objectives and continual improvement of the organisation's system of project quality management and planning. In this way employees can be actively involved in the process of project quality policy planning and can help the organisation achieve strategic objectives. Software for project quality management can be used for receiving and processing employee feedback.

Quality Reviews

This approach is targeted at developing and maintaining a programme of employee quality and performance appraisal to facilitate the project quality management and planning team with measuring employee performance, identifying productivity gaps, and describing reasons for reduced employee quality. Project quality management software will help conduct reviews of employee equality and develop the programme.

Process

- Certification: Organisations should follow some quality certification(s) that is presented in the project quality policy statement example. For example, it could be ISO 9001:2010 certification. The choice will depend on the project nature and the product to be produced. The effort for project quality policy management should focus on maintaining existing quality management systems and ensuring continued certification.
- Strategic Alignment: Objectives of the project quality planning and management process should be aligned with business objectives of your organisation - this will help make your project quality policy statement sample strategically aligned and meeting project goals. Project quality management software will help the organisation create dependencies and relations between the objectives.

- Audit: Analysis of quality policy planning and management can be done with help of an internal audit. Organisation need to establish a surveillance process to track all activities associated with quality assurance and control. Results of auditing should be communicated to senior management of the project to enable them to make further decisions.

Performance

- Performance Reviews: This approach of project quality policy management requires conducing regular reviews of performance to ensure that the statement remains effective and meets strategic objectives and needs of the organisation.
- KPIs: Key performance indicators (KPIs) should be identified and used to assess efficiency of project quality management and planning activities.
- Excellence Model: Such a model is used to monitor project performance and drive continual improvement of quality policy planning and management across the organisation. Project quality management software will help create models of excellence and build charts and diagrams for tracking and monitoring project performance.

4.2.4 Established Quality Management Methods, Tools and Techniques

According to the PMBOK the following are tools and techniques that should be used in quality management: **Cost-Benefit**

Analysis

As discussed in Study Unit 4.1.8, it is important to consider cost-benefits trade-offs during the quality planning process. The most important benefit of meeting sufficient quality requirements is that there will be less rework, higher productivity, lower costs, and greater satisfaction from the stakeholder. The main cost of achieving such quality requirements is the expense that comes with activities relating to Project Quality Management.

Benchmarking

Benchmarking is a process where planned or existing project practices are compared to the practices set in place for other projects so that ideas as to which areas of the project could be improved upon can be generated. Furthermore, it is also used to provide a basis to measure overall performance. The projects used for comparison can be from within the performing organisation or from a source outside of it, and do not necessarily have to be from within the same application area to be used.

Design of Experiments (DOE)

Design of Experiments (DOE) is a method in which factors are identified which may influence certain aspects of a product or process during the time it is under development or in production. It also holds a key role in the process of optimising products/processes. DOE would be used by organisations to reduce the sensitivity of product performance to factors caused by differences in manufacturing or the environment. The main benefit of DOE is that it provides the organisation with a framework for systematic change to all the important factors regarding a project, rather than changing them one at a time. When the data obtained is analysed, an organisation can find the optimal conditions for their product/process, with a focus on factors influencing the results, and showing the existence of correlations and interactions within the factors.

Cost of Quality (COQ)

Quality costs are the total of all costs that were incurred in order to prevent non-conformance to established project/process requirements, appraising the product for conformance to requirements, and any rework was necessary because of a failure to meet requirements. Failure costs are divided into internal and external costs, and are also known as cost of poor quality.

Additional Quality Planning Tools

Additional quality planning tools are often used to better define the situation and assist in planning effective and efficient quality management activities. These include

Brainstorming

Brainstorming with a group of people is a powerful technique to create new ideas, solve problems, motivate and develops teams. Brainstorming motivates because members of a team get involved in bigger management issues, and it gets a team working together.

However, brainstorming is not simply a random activity and it needs to be structured and follow brainstorming rules. You will need a flip-chart or alternative, and it is important to follow the process described below as brainstorming needs to involve the team, which means that everyone must be able to see what's happening. The facilitator must be able to manage the process, people's involvement and sensitivities, and then manage the follow up actions.

Brainstorming Process

- Define and agree the objective
- Brainstorm ideas and suggestions having agreed a time limit
- Categorise/condense/combine/refine
- Assess/analyse effects or results
- Prioritise options/rank list as appropriate
- Agree action and timescale
- Control and monitor follow-up

Affinity Diagrams

After a brainstorming session, the pool of ideas needs to be analysed and prioritised before they can be implemented. A smaller set of ideas are easy to sift through and evaluate without applying any formal technique. Affinity diagramming is an effective technique to handle a large number of ideas. It is typically used for

- Traversing large data set, like ideas generated from brainstorming and to sieve them for prioritisation
- Simplifying complexity due to diverse views and opinions.
- Group involvement and consensus

Association, kinship, likeness are synonymous to affinity, and they are the underlying principle to be followed while using this technique. During affinity diagramming the team must categorize the ideas based on their subject knowledge thereby making it easy to sift and prioritise ideas.

How to conduct an Affinity Session

- Use a venue free from interruptions and distractions
- Start the session by explaining the purpose, preferably already written and highlighted on the board.
- Place the Post-its randomly on the board/flat surface, making sure that all participants can see them
- The first objective is to sort ideas, opinions and issues based on the most natural relationship amongst them
- Invite participants to get involved in the exercise of creating similar groups from the idea pool.
- Participants should look for ideas that are related in some way, move the post-its for the similar ideas to be together. The process should be followed till all ideas have been grouped.
- When there is conflict of ideas falling in more than one group. resolve it by placing another post-it with the same idea in both groups.
- Participants now can discuss on the patterns identified and focus on the controversial ideas. A consensus can be reached by discussion, making the changes in grouping.
- When ideas are grouped, select a suitable heading to categorise each group. Now move groups under their respective headings to view the Affinity Diagram.
- At the end of session the categorised ideas are now ready for further shortlisting, if necessary.

Nominal Group Techniques

Nominal group technique (NGT) is a structured method for group brainstorming that encourages contributions from everyone.

Nominal Group Technique Procedure

Materials needed: paper and pen or pencil for each individual, flipchart, marking pens, tape.

- State the subject of the brainstorming. Explain the subject if necessary until everyone understands it.
- Each team member silently thinks of and writes down as many ideas as possible in a set period of time (5 to 10 minutes).
- Each member in turn states one idea aloud, while the facilitator records it on the flipchart.
    - At this stage discussion is allowed, not even questions for clarification.
    - Ideas given do not need to be from the team member's written list. Indeed, as time goes on, many ideas will not be.
    - A member may "pass" his or her turn, and may then add an idea on a subsequent turn. Continue around the group until all members pass or for an agreed-upon length of time.
- Discuss each idea in turn. Only change wording when the idea's originator agrees. Ideas may be crossed out from the list only by unanimous agreement. Discussion may clarify meaning, explain logic or analysis, raise and answer questions, or state agreement or disagreement.
- Prioritise the ideas using multivoting or list reduction

Flowcharts

A flowchart is a visual representation of the sequence of steps and decisions needed to perform a process. Each step in the sequence is noted within a diagram shape. Steps are linked by connecting lines and directional arrows. This allows anyone to view the flowchart and logically follow the process from beginning to end. A flowchart is a powerful business tool. With proper design and construction, it communicates the steps in a process very effectively and efficiently

Prioritisation Matrices

A project prioritization matrix is used when there are competing priorities for team members in multiple existing projects create. The tool helps to determine the order in which projects should be completed, considering a number of factors. Team members rate each factor and assign a weight if needed. The sums will indicate the best opportunity, where the team should focus resources.

Benefits

- Assists with prioritising existing projects
- Evaluates various factors that have an impact on decisions

How to Use

Step 1. Determine the possible projects or criteria, and list them on one side of an empty matrix.

Step 2. Brainstorm important factors or criteria that will be used to evaluate the options and list them across the top of the matrix, along with a weight or importance multiplier for each.

Step 3. Fill in the matrix by ranking each project in each of the factor areas.

Step 4. Sum the totals of each factor.

4.2.5 Requirements for a Project Quality Management System

According to ISO 9001, the concept of a Quality Management System is:

A Quality Management System in its basic concept is quite simple. It seeks to:

- Recognise the external quality related requirements specified in Licenses to Trade, guidelines, specified customer requirements, and the chosen management system standard(s)
- Ensure that all requirements have been documented within the management system in the appropriate location in terms of defined specific system requirements
- Confirm that employees receive applicable training in the quality system requirements
- Outline performance processes, where applicable, to the quality system requirements
- Produce records or evidence that system requirements have been met
- Measure, monitor and report the extent of compliance with these performance procedures
- Continually monitor and analyse changes to the requirements and confirm that all changes are reflected in changes to the specific requirements when necessary
- Execute the audit and analyse the system processes and correct them when necessary
- Include processes that will help continually improve the quality system.

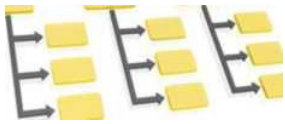
Included in the system should be the following:

Organisation Structure

This will be the organisational structure of the project. It must clearly indicate each role title, role description which would specify the responsibilities, goals, limits of authority, relationships, skills, knowledge and experience required of the role.

The project organisation structure would best be presented in a graphical or chart form, showing:

- The hierarchical dependency between the management group overseeing the project,
- The project manager and the different team leaders (when this level of organisation exists), and also
- The organisational environment of the project with entities external to the development (e.g. Expert group, technical committee, quality assurance, the client).



Specify the highest authoritative level of the project organisation which represents at managerial level the Business, User and Supplier interests of the project. This usually takes the form of either a Project Board or a Project Steering Committee. The composition of the Project Board or the Project Steering Committee should therefore comprise of at least -

- a Senior Executive who looks after the business interest of the project (e.g. a senior IDA representative),
- a Senior User who champions the desired outcome required by users and ensure that the project delivers it (e.g. senior Member State Representative, User Group representative, or Expert Group representative),
- a Senior Supplier member who has the authority to provide the necessary resources to deliver the contractual products.

Key Project Personnel and Representatives

List all key project personnel involved with the project. Each of the role title identified must be fully described as to why they are involved in the project and what their responsibilities, contributions and expectations are.

Responsibilities

These may relate to:

- Resources (personnel, premises, hardware, software and any other equipment) put at the project's disposal
- Co-ordination of activities involving expert and user groups, technical committees
- Providing the deliverables required for use by the supplier
- Providing all documentation and information necessary for the project within acceptable delays. This includes sufficient availability of the users and other involved persons
- Procedures and timetables for the acceptance of deliverables which have to be respected by the IDA project. Approval of a deliverable imply the approval of the users concerned with the content of the deliverable. A deliverable cannot be considered accepted until the IDA Project Manager has signed it off.
- A fast feedback from the IDA project to the Supplier. This is a necessary condition to diagnose quickly any possible misunderstandings between the partners. It is therefore important that the IDA Project comment on minutes of meetings and drafts of documents as soon as possible.

Obligations and Responsibilities of Other Involved Groups

The groups identified as having involvement in this project must: -

- Provide documentation and access to specialist information. Members of expert groups and technical committees have to provide all documentation and information necessary for the project within acceptable time-scale
- Ensure attendance at meetings to provide expert input. Members of expert groups and technical committees have to be available to attend those meetings that require their presence in furthering the project's progress.

Procedures and Processes

This should include the following:
Progress Measurement and Monitoring

The procedures and the types of information that would be needed and used to measure and monitor the progress of the project must be described here. The following list should be customised and used accordingly based on the size and complexity of the project:

- Information about work progress.
- The frequency and the format of the Project Progress Report.
- A first version of the project time plan
- Project progress meetings
- Technical and informal

Process Controls

The purpose of adopting controls is to ensure that the project: -

- Is producing the required products which meet the defined Acceptance Criteria
- is being carried out to schedule and
- in accordance with the resource and
- budget plans remains viable

The following list, which is not exhaustive, should be tailored and used according to the needs based on the size and complexity of the project:

- Quality reviews and approval process
- Risk Management
- Change Control
- Standards and protocols
- Project file
- Monitoring of subcontractors. To monitor the effectiveness of subcontractors the supplier, who is effectively, the prime contractor must consider addressing:
    - verification and checkpoints processes with an indication of:
    - the authority responsible for the action
    - a short description of what is going to be verified e.g. sub-system, documentation, etc.
    - when it will take place e.g. stated frequency or at the end of a phase (completion of a document, end of production, etc.)
        - the type of action to be taken e.g. inspection, walk-through, review, audit, etc.
    - the type of records that will be produced and kept (inspection report, test results acceptance sheet, audit report etc.).

Check Lists

A checklist is a structured tool, usually industry- or activity-specific, used to verify that a set of required steps has been performed. Checklists may be simple or complex. They are usually phrased as imperatives ("Do this!") or interrogatories ("Have you done this?"). Many organisations have standardized checklists available to ensure consistency in frequently performed activities. In some application areas, checklists are also available from professional associations or commercial service providers

The following is an example of a checklist:

CHECKLIST FOR MANAGING PROJECT QUALITY

	Yes	No	N/A	Remarks
Has the Quality Assurance role been identified and assigned an appropriate level of authority?				
Are project standards and procedures being defined for each area of the project to ensure consistent delivery in accordance with the quality requirements?				
Do the standards and procedures make reference to material in the knowledge				

	Yes	No	N/A	Remarks
base and simply note any project-specific additions, exceptions, or enhancements?				
Are the standards and procedures in place prior to the commencement of the work to which they apply?				
Are all team members adequately trained in the tools and standards and procedures for the project environment (or will additional start-up training be required)?				
Is each member of the team personally committed to abide by the standards and procedures?				
Is the team involved in the ongoing development and adoption of the standards (to promote a sense of ownership by the team)?				
Are formal structured walk-throughs and reviews being conducted in accordance with the standards and procedures (e.g., the proper people attend, minutes are taken and distributed, and action items are assigned and followed up)?				
Are appropriate test plans and procedures in place and being followed?				
Will the deliverables being produced meet the user's requirements?				
Are records being kept by the Quality Assurance role of quality problems, recommendations and actions?				
Are effective measurement techniques being used to identify the levels of quality and productivity being achieved and to provide a baseline for continuous improvement?				

	Yes	No	N/A	Remarks
Is project data being analyzed to determine trends and norms and to identify when quality threshold or target levels are not being achieved?				
Have quality improvement sessions been conducted between the Quality Assurance Manager and the project team to develop solutions to delivery problems and to explore opportunities for improvement?				
Are corrective actions being monitored for effectiveness in eliminating the cause of problems?				
In any cases where corrective action is not taken or is ineffective, has the problem been escalated to the Quality Management function outside the project team?				

Operational Definitions

An operational definition describes, in very specific terms, what something is, and how it is measured by the quality control process. For example, it is not enough to say that meeting the planned schedule dates is a measure of management quality; the project management team must also indicate whether every activity must start on time, or only finish on time; whether individual activities will be measured or only certain deliverables, and if so, which ones. Operational definitions are also called metrics in some application areas.

4.2.6 Lessons Learned from Previous Project

According to the Project Management Institute (PMI) and the Project Management Body of Knowledge (PMBOK), lessons learned is the learning that was gained from to process of performing previous projects. It is customary to hold a formal lessons learned session during project close-out, when the project is nearing completion. However, lessons learned can also be identified and documented at any time during the life cycle of a project. The purpose of documenting the lessons learned is to share and use knowledge that were gained from experience to:

- Promote recurrence of good outcomes
- Prevent the recurrence of poor outcomes

In practice, lessons learned includes all the processes that are needed to identify, document, validate and disseminate lessons learned. The use and incorporation of these processes are used to identify and document applicable lessons learned, distribute them to applicable employees, identify actions that should be taken as a result of the lessons learned, and follow up to ensure that the correct actions were taken.

4.2.7 Product Quality Requirements and Criteria

All material or services have characteristics according to which the identification of its quality is facilitated. The characteristics are part of the conditions of how the material, equipment and services are able to meet the requirements of the project and are fit for use by the beneficiaries. Quality characteristics are about the attributes, measures and methods attached to that particular product or service.

- Functionality is the degree, by which equipment performs its intended function, and for clinical equipment, this is especially important that the operation should be behave as expected.
- Performance, it's how well a product or service performs the beneficiaries intended use.
- Reliability, it's the ability of the service or product to perform as intended under normal conditions without unacceptable failures.
- Relevance, it's the characteristic of how a product or service meets the actual needs of the beneficiaries, it should be pertinent, applicable, and appropriate to its intended use or application
- Timeliness, how the product or service is delivered in time to solve the problems when its needed and not after, this is a crucial characteristic for health and emergency relief work
- Suitability, defines the fitness of its use, it appropriateness and correctness, the agriculture equipment must be designed to operate on the soul conditions the beneficiaries will use it on.
- Completeness, the quality that the service is complete and includes all the entire scope of services. Training sessions should be complete and include all the material needed to build a desired skill or knowledge
- Consistency, services are delivered in the same way for every beneficiary. Clinical tests need to be done using the same procedure for every patient.

Quality characteristics are not limited to the material, equipment or service delivered to the beneficiaries, but also applies to the material, equipment and services the project staff uses to deliver the project outputs. These include the vehicles, computers, various equipment and tools and consulting services the project purchases and uses to carry out its activities.

Quality characteristics must be included in all material, equipment and services the project will purchase, the procurement officers must have a complete description of what is required by the project, otherwise a procurement office may purchase the goods or services based on her or his information of the product.

Project Quality Management

Study Unit 4.3: Project Quality Management Plan

US 243816

Develop a project quality management plan for a simple to moderately complex project Specific Outcome 1

Compile the project quality management plan.

Assessment Criteria

1. Quality policies, procedures and standards applicable to the project are selected and documented in the required format.
2. Quality criteria are determined with input from the relevant stakeholders and documented in the required format.

   ASSESSMENT CRITERION RANGE

   Quality criteria include but are not limited to quality measurements and variances.

3. Quality management activities and resource requirements are determined and incorporated in the project schedule.
4. Roles and responsibilities for quality management activities are defined and agreed with relevant stakeholders.
5. The budget for quality management activities is determined and agreed with relevant stakeholders.
6. The project quality management system is defined to meet requirements and agreed with relevant stakeholders.
7. All project quality management planning components are gathered and consolidated into a project quality management plan in the required format.
8. The project quality management plan provides for quality assurance, quality control and process improvements.

The Project Quality Management Plan was already discussed in Study Unit 4.1.3, but we are now going to look in more detail at what should be included, and also at a template that can be used to draw up the Project Quality Management Plan

4.3.1 Quality Policies, Procedures and Standards

It is important to include all the details about quality policies, procedures and standards in the plan so that all stakeholders are aware of exactly what these entails. This was discussed in Study Unit 4.2.3.

4.3.2 Quality Criteria

Quality Criteria should be stated very clearly. The following should be included:

Quality Metrics and Measures

Quality metrics are ways in which the level of quality of a project is assessed, and also the processes that will be followed to carry out such measurements. Metrics outline the against which work will be measured, and is often typical and unique to each project and product. The quality metrics must be determined and defined during the planning phase of the project, ad is then measured throughout the project's life to ensure the projects conformity to the established quality baseline.

When the metrics by which quality of the project is measured against is determined, an established standard is identified and that standard is then used to establish a quality baseline for each defined quality metric. Such a baseline the acts as a barometer to measure overall project quality right through the life of the project. Sources of quality baseline information are:

- The quality plan of the organisation
- Similar projects completed within the last six months and lessons learned from such projects
- Industry standards

Acceptance Criteria

Acceptance criteria are minimum standards or requirements that a project or product must meet before the deliverables of the project can be accepted. These criteria are pre-established, and defined in the planning phase of the project. It is then tracked throughout the life of the project to ensure that the project conforms to the quality standards that were established. Acceptance criteria can include but is not limited to:

- Functionality requirements
- performance measures
- Essential conditions
- Regulatory compliance

4.3.3 Quality Management Activities and Resource Requirements

Quality management and control activities were discussed in detail in Study Unit 4.1.5.

It is important that all these activities, as well as the resources that will be needed to perform them is clearly stipulated in the Project Quality Management Plan.

4.3.4 Roles and Responsibilities for Quality Management Activities

Roles and responsibilities for quality management activities were discussed in detail in Study Unit 4.2.1.

It is necessary that all the role players are clearly indicated in the Project Quality Management Plan, as all stakeholders need to know who the responsible people are. This will ensure that communication regarding certain aspects of the quality management process is directed at the correct person, and also that all stakeholders know what their respective responsibilities are.

4.3.5 The Budget for Quality Management Activities

Project cost management is about the following processes involved in planning, namely estimating, budgeting, and controlling costs so that the project can be completed within the approved budget.

- Cost estimating: Developing an approximation of the costs of the resources that are necessary to complete project activities.
- Cost budgeting: Aggregating the estimated costs of individual activities or work packages to establish a cost baseline.
- Cost control: Influencing the factors that create cost variances and controlling changes to the project budget.

Project cost management is basically about the cost of the resources that are necessary to complete schedule activities. Project cost management should also keep in mind the effect that project decisions have on the cost of using, maintaining, and supporting the product, service, or result of the project. On some projects, especially ones of smaller scope, cost estimating and cost budgeting are so closely linked that they are seen as a single process that can be performed by a single person over a relatively short period of time.

The dependence between project cost management processes and project phases can be seen in the table below

Process	Project phase	Key deliverables
Estimate costs	Planning	Activity cost estimates, Basis of estimates
Determine budget	Planning	Cost performance baseline
Control costs	Monitoring and controlling	Work performance measurements

Project cost estimates are a key component of the planning process and provide a basis for important decisions. Cost estimate represents a prediction of quantities, cost, and/or price of resources that are needed by the scope of a project.

An activity cost estimate is a quantitative assessment of the expected costs of the resources necessary to complete schedule activities. This type of estimate can be presented in summary form or in detail. Costs are estimated for all resources that apply to the activity cost estimate. This includes, but is not limited to, labour, materials, equipment,

services, facilities, information technology, and special categories such as an inflation allowance or cost contingency reserve.

It is possible to assess the level of quality with costs. Organisations strive for high quality products and services, but the expenses associated with this may cause, in turn, its bankruptcy. Therefore, it is necessary to develop a budget to improve the quality and compare it with expected profit.

Estimating the cost of project is one of the most important tasks for project managers, and this should be done together with other stakeholders. The main factors that are usually estimated at the beginning of a development project are: cost, size, schedule, quality, people resources, effort, resources, maintenance costs, and complexity. Cost estimation tools, or model-based estimation techniques use data that was collected from past projects combined with mathematical formulae to estimate project cost. They usually need factors such as the system size as inputs into the model.

It is important to include this cost estimation budget in the Project Quality Management Plan, so that all stakeholders are aware of the costs and the budget, and can ensure that there is no overspending on the project.

The following is an example of a project budget template:

Expenses				
Salaries	Description	Quantity	Cost	Total
Staff/Faculty title	% of Time on Project			R0
Benefits	23% total staff/faculty salaries		0.23	R0
Consultants	Consultant fee (hourly or flat)			R0
Guest speakers	Honoraria rate			R0
Sub - total Salaries				$0
Travel	Description	Quantity	Cost	Total
Lodging	cost per person/per room			R0
Airfare/Train	cost per person/per fare			R0
Gas	.485 per mile		0.485	R0
Per Diem	meal stipend			R0
Other	(i.e. taxi, subway, light rail fare)			R0
Sub-total Travel				R0
Supplies	Description	Quantity	Cost	Total
Materials				R0
Other				R0
Sub-total Supplies				R0
Events	Description	Quantity	Cost	Total
Catering	event & # attendees			R0
Flowers	centerpieces, plants, etc.			R0
Gifts	give aways - cards, pens			R0
Awards	citations, trophies			R0
Other				R0
Sub-total Events				R0
Marketing/Communications	Description	Quantity	Cost	Total
Graphic Design/printing	posters, flyers, program			R0
Advertisements	print or television advertisements			R0
Mailing expenses	envelopes, postage			R0
Other				R0
Sub-total Marketing/Communications				R0
Other MICA Depts/Services	Description	Quantity	Cost	Total
Exhibitions	services needed			R0
Campus Tech	services needed			R0
FacMAN	services needed			R0
Security	# security guards needed			R0
Other				R0
Sub-total Other MICA Departs/Services				R0
Training	Description	Quantity	Cost	Total
Work Study	# work study students and tasks			R0

Other	space rental fee			R0
Sub-total Training				R0
MICA Indirect Costs		Quantity	Cost	Total
Indirect	15% of total expenses		0.15	R0
Sub-total Indirect				R0
Total Expenses				#REF!
Income				
Program Income	Description	Quantity	Cost	Total
Department Budget	Department Name			R0
Sales	ticket income			R0
In-kind contributions	goods/services donated			R0
Grants/Sponsorships/RFP's	donor/prospect name			R0
Other	revenue from partnership			R0
Total Income				R0
Net Income	Income-Expenses			#REF!

4.3.6 The Project Quality Management System

This was discussed in detail in Study Unit 4.2.5, and details should be included in the Project Quality Management Plan

4.3.7 Project Quality Management Plan

Appendix A is a template that can be used for the Project Quality Management Plan. This template is according to the prescriptions of PMBOK (Project Management Body of Knowledge), which provides guidelines for managing individual projects and defines project management related concepts. It also describes the project management life cycle and its related processes, as well as the project life cycle:

4.3.8 Quality Assurance, Quality Control and Process Improvement

Quality Assurance and Quality Control were discussed in detail in Study Units 4.1.4, 4.1.5 and 4.1.6, and details should be included in the Project Quality Management Plan

Study Unit 4.4: Approval for the Project Quality Management Plan

US 243816	
Develop a project quality management plan for a simple to moderately complex project	
Specific Outcome 1	
	Obtain approval for the project quality management plan.
Assessment Criteria	
	1. The project quality management plan is presented to relevant project authorities for approval. 2. The project quality management plan is incorporated into the integrated project management plan. 3. The project quality management plan is communicated to all relevant stakeholders.

4.4.1 Present Project Quality Management Plan for Approval

A Project Quality Management Plan is finalised when it is formally accepted and approved by the project sponsor and other designated stakeholders. Formal approval acknowledges that all the deliverables produced during the Plan Stage are complete, reviewed and accepted.

Signatures on the project plan document indicate final approval. This sign-off marks the plan as the go-forward agreement and can be viewed as a project management milestone. A Project Quality Management Plan represents a commitment to continue and to dedicate the required time and resources toward the project. It ushers the way into the Execute and Control Stage.

People Involved

- Project Manager
- Project Sponsor
- Project Stakeholders

4.4.2 Incorporate Project Quality Management Plan into Integrated Project Management Plan

Project management plan, as defined in the PMBOK Guide Third Edition, is a formal, approved document that defines how the project is executed, monitored and controlled. It may be summary or be detailed and may be composed of one or more subsidiary management plans and other planning documents.

The objective of a project management plan is to define the approach to be used by the project team to deliver the intended project management scope of the project.

The project manager creates the project management plan following input from the project team and key stakeholders. The plan should be agreed and approved by at least the project team and its key stakeholders. The project management

1. Scope management
2. Requirements management
3. Schedule management
4. Financial management
5. Quality management
6. Resource management

plan typically covers topics used in the project execution system and includes the following main aspects:

7. Communications management
8. Project change management
9. Risk management
10. Procurement management

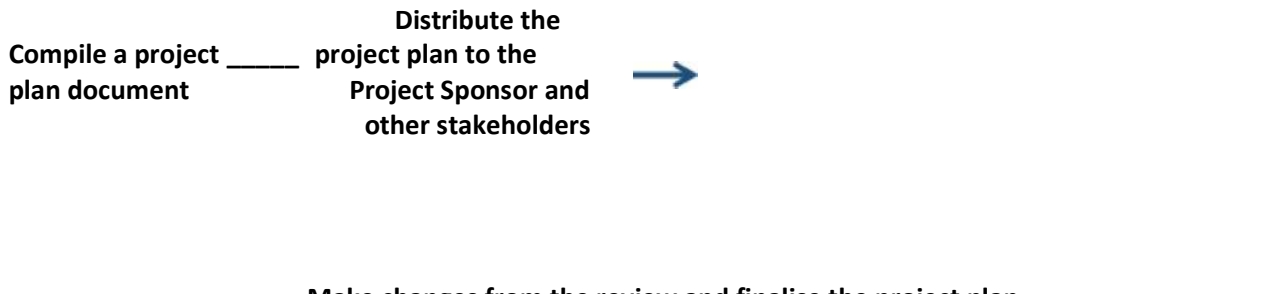
The Project Quality Management Plan will then form part of section 5: Quality Management.

It is good practice and mostly required by large consulting and professional project management firms, to have a formally agreed and version controlled project management plan approved in the early stages of the project, and applied throughout the project.

The project management plan is the key document that contains the overall planning, monitoring, and implementing activities to be done in a project.

4.4.3 Communicate Project Quality Management Plan to Stakeholders

The flow chart below describes the sequence of activities that results in a reviewed and approved plan:



Appendix A: Template for Project Quality Management Plan

[Company Address] Tel: [Company Phone]

Fax:

[Company E-mail]

[Ref. number] User

REVISIONS AND DISTRIBUTION

Revision	Release date	Distributed to*											
		t n i C l	Consultant	JV Main office(s)	All project mgmt. dept.	Sub-contractors	Suppliers						
Rev. 0 (draft)	29/10/2013												

*) Detailed distribution lists shall be prepared for each distribution event. Further details as per the project communication plan

AMENDMENTS

The from time to time may require updates. Any amendment to this plan shall be informed to the change control board by use of the change request form and approved by the project change control board prior to distribution. Only revised parts of the plan will be distributed along with the approval and shall be accompanied by instructions how to implement the changes.

The initial page numbering system (to be added upon initial approval) will be a normal continuous numbering displayed in the lower right corner of each page. In the event that pages have to be added, characters shall be added to the number. In case entire pages are deleted, the corresponding page shall be replaced by a blank page stating "page removed".

Each added/changed page shall have the revision number and date of approval displayed on the bottom of the page.

PROJECT SPONSOR APPROVAL

Prepared by:	Reviewed by:	Approved by Proj. Sponsor:
Place, dd/mm/yyyy	Place, dd/mm/yyyy	Place, dd/mm/yyyy
User Designation	Name Designation	Name Designation

Table of Contents

QUALITY MANAGEMENT APPROACH

Quality management under this document is defined as a system of planned activities that ensures that the deliverables of the xyz project meet or exceed the client requirements, specifications and expectations. The system includes inspections, verifications, audits and evaluations of materials and workmanship required to document the grade of quality of the deliverable.

This quality management plan in conjunction with the other various project plans also identifies the necessary personnel involved in the quality team, their responsibilities and inter-relationships.

OBJECTIVE

The objective of this project quality management plan is to ensure that the xyz project is delivered in accordance with the contractual specifications, to satisfy stakeholder requirements, to reduce cost of quality (namely cost of re-works, cost of non-conformity and deficiencies, unnecessary future maintenance and repair cost) and to complete the project within the contractual baselines.

BASIC PROCESSES OF QUALITY MANAGEMENT

- Plan quality management,
- Control quality,
- Perform quality assurance.

TOOLS AND TECHNIQUES TO BE APPLIED

- Field monitoring,
- Inspecting,
- Testing,
- Reporting,
- Reviewing technical and laboratory data,
- Identifying and documenting defects and necessary repairs,
- Compiling applicable data, log-files, as-built documents and photos.

GENERAL APPROACH

Quality management shall focus on both product quality and project quality

Product quality in this context basically is concerned about what is delivered, and therefore focuses on product specifications and requirements.

Project quality more is concerned about how the product is delivered, and focuses more on the production processes, planning etc.

The quality system in general shall follow the Plan - Do - Check (Study) - Act approach.

LAWS, REGULATIONS AND GUIDELINES

Strict adherence to this plan in no way absolves any party from any obligations or responsibilities under applicable laws and regulations.

The laws and regulations applicable to this project and relevant to quality management are defined in the contract and include:

- xyz law,
- xyz Construction Standard,
- Regulations by xyz bureaus and authorities.

OTHER PROJECT PLANS

This project quality management plan forms part of the overall project management plan. Further project plans to be read in conjunction to this project quality management plan are:

a) Project Management Plan,
b) Project Scope Management Plan,
c) Project Requirements Management Plan,
d) Project Schedule Management Plan,
e) Project Cost Management Plan,
f) *(this project quality management plan),*
g) Process Improvement Plan,
h) Project Human Resource Management Plan,
i) Project Communication Management Plan,
j) Project Risk Management Plan,
k) Project Procurement Management Plan,
l) Project Stakeholder Management Plan,
m) Project Financial Management Plan,
n) Project Health and Safety Management Plan,
o) Project Environmental Management Plan,
p) Project Claim Management Plan.

PROJECT SCOPE OF WORKS (BRIEF DESCRIPTION)

The overall project scope of works is defined in the scope of works documentation (SOW) and consists of:

- xyz
- xyz

The project and its site facilities is located at Address

A site layout plan depicting the current site conditions is provided as attachment 1.

Personnel, equipment and material enter and exit the project site by gated access located at Address

ORGANISATION AND STAFFING ORGANISATION CHART

The organisation chart is a supplemental chart to the overall project organisation chart as depicted in the project human resource management plan.

The quality department shall be organized in sub-divisions and their main roles shall be distributed as abstracted below:

<table>
<tr><td colspan="3">Quality Assurance & Quality Control Management</td></tr>
<tr><td colspan="3">QA & QC Manager
• Plans the project quality management
• Directs and manages the activities of the department
• Coordinates activities with other disciplines
• Reviews the outputs of the departments
• Proposes and implements process improvements where advisable</td></tr>
<tr><td>Quality Control</td><td>Quality Assurance</td><td>Document Control</td></tr>
<tr><td>QC staff
• Inspects materials and equipment
• Verifies compliance with methodologies and appropriate working conditions
• Inspects ongoing and results of activities (on and off site)
• Produces checklists, forms and records</td><td>QA staff
• Reviews material submittals and method statements
• Verifies compliance with quality criteria
• Evaluates inspection records
• Identifies training needs including supplier and sub-contr. personnel
• Evaluates project records
• Audits project procedures</td><td>DC staff
• Updates project records and registers
• Compiles the quality file
• Receives and registers relevant documents from other departments etc.
• Distributes documents to other departments</td></tr>
</table>

For more detailed descriptions of the roles and responsibilities refer to the corresponding section in the human resource management plan.

Please find the organisation chart of the QA & QC department attached to this plan (attachment 2).

QUALITY METRICS

The applicable quality metrics are defined in the contract, relevant laws and regulations and through requirements management.

ID	Item	Standard Value	UOM	Measurement method	Acceptable Tolerance	
					upper	lower
0001						

QUALITY ASSURANCE

Quality assurance (QA) is a set of activities for ensuring quality in the processes by which products are developed.

The focus of QA is the avoidance of problems and defects as well as the associated costs.

By auditing compliance to the approved project plans, processes, requirements and expectations processes continuously will be developed further with regards to workability and cost efficiency.

The key aim of QA is to continuously investigate issues and level-by-level develop a system and processes to avoid repetition of earlier issues and establish a management structure to facilitate process improvement at an organisational level.

The targets are:

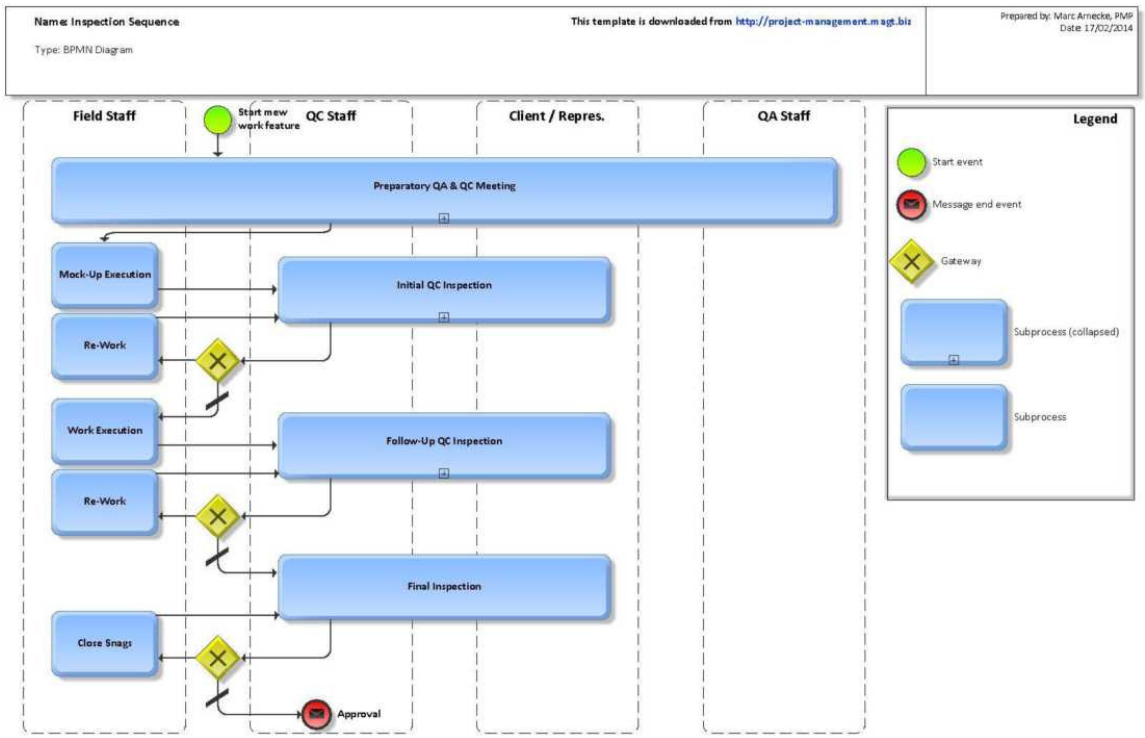
1) Increase awareness of processes that could enhance organisational competencies.
2) Identify processes that need organisational improvement.
3) Facilitate discussions to allow people sharing their experiences inside and outside their functional boundaries.
4) Prioritize processes to achieve continuous improvement.

It is the aim of this plan to determine potentially existing process errors and take corrective action accordingly wherever possible before the error effects in any way.

For further details read the "Process Improvement Plan".

GENERAL INSPECTION SEQUENCE

This sequence of Inspections is mainly concerned with workmanship.



Material and other inspections are not shown here and will be dealt with separately.

QUALITY CONTROL

Quality control (QC) is a set of activities for ensuring quality in products. The activities focus on identifying defects in the actual products produced.

QC Inspections shall be conducted as a three-phase control process such as

Prior mock-up	Preparatory meeting
Mock-up stage	1. Initial inspection
Production stage	2. Follow-up inspections
	3. Final inspections

Those inspections and meeting shall be carried out for each work item regardless of whether that particular item was or will be produced by the contractor or its sub-contractors.

Implementation of the three-phase control process vouches for compliance with approved plans, specifications and requirements. Each single control phase is important for achieving the required grade of quality, whereas the preparatory meeting and initial inspections are particular important to make provision against re-works and its effects such as failure, cost, delay and others.

PREPARATORY QC MEETING

Preparatory QC Meeting(s) shall be arranged by the QC staffs at the beginning of each new work item and all accompanying activities. QA staffs shall be notified and client representative shall be invited. All concerned field staffs (key persons) shall attend. in particular the preparatory meeting shall address

- Review of work plans,
- Review of operating procedures,
- Review of working conditions,
- Equipment check,
- Material check,
- Assessment of work sequences,
- Assessment of work methodology.

The purpose of this meeting is to verify that required plans, specifications and methodologies have been prepared, are approved and are available and known to the field staff before an activity or the production of a work item is started; equipment is appropriate for the intended use, fully functional and is calibrated if necessary and all appropriate arrangements h ave been made.

As part of the preparatory meeting the QC staff will verify that lessons learned during previous comparable activities have been implemented to avoid repetition of past problems.

Any identified discrepancy must be resolved with the approved plan and/or methodology and shall be resolved prior to the production of the work item and prior any other activity which makes the later rectification of the discrepancy virtually impossible.

The meeting shall be conducted by means of the "Preparatory Meeting Checklist" (checklist 1).

INITIAL QC INSPECTION

Initial QC Inspection(s) shall be carried out by the QC staffs after a work item has been produced for the first time. Thus the QC staff will validate that the work is in compliance with applicable specifications and procedures and

establishes an acceptable level of workmanship.

The inspection shall be conducted by means of the "Initial Inspection Checklist" (checklist 2). The results of the inspection shall be recorded in the "Inspection Register" (Register 1).

Any differences of opinion in the interpretation of project requirements, specifications and methodologies will be settled with a view to the production stage of the feature.

Once a product or intermediate product (whatever applicable) is ready "Inspection and Test Request" (form 1) shall be passed to the client or his representative and a copy of the Initial Inspection Checklist shall be attached thereto.

The initial inspection result shall be registered in the "Inspection Register" (register 1).

FOLLOW-UP QC INSPECTION

The purpose of the Follow-up QC Inspection is to ensure continuous compliance to applicable specifications, requirements and adherence to established methodologies as well as an acceptable level of workmanship.

Follow-up QC Inspection(s) shall be carried out by the QC staff regularly during the production phase of works.

The Follow-up Inspection shall be carried out at variable intervals during the execution of works and shall cover all different stages of the activity including

- activity preparation,
- activity execution and
- result control.

Any identified discrepancy must be resolved with the approved plan and/or methodology and shall be re-inspected prior to the commencement of the particular activity or any other activity which makes the later rectification of the discrepancy virtually impossible.

The inspection shall be conducted by means of the "Follow-up Inspection Checklist" (checklist 3).

Once a product or intermediate product (whatever applicable) is completed "Inspection and Test Request" (form 1) shall be passed to the client or his representative and a copy of the Initial Inspection Checklist shall be attached thereto.

Follow-up QC inspections shall be registered in the "Inspection Register" (register 1).

FINAL INSPECTION

Upon completion of the whole of works a pre-final inspection shall be carried out by the QC staffs and client or his representative.

The purpose of this inspection shall be to validate the final result of the whole of the works.

Outstanding and nonconforming items will be identified and documented in a punch-list. As each item is closed out it will be recorded. When all items recorded during the pre-final inspection have been corrected the client or his representative will be notified to schedule the final inspection.

The final inspection result shall be registered in the "Inspection Register" (register 1).

Further re-inspection is not foreseen, exceptional resumption of outstanding work and/or execution of instructed additional works.

CLIENT ABSENCE DURING INSPECTIONS

In the event that the client or his representative does not attend inspection(s), the inspection result(s) recorded by QC staffs shall be deemed binding and shall be passed on to the client or his representative "for information".

RECORD KEEPING

The QC staffs are responsible for the completion of forms and checklists. Checklists are to be filled during the ongoing inspection. All forms and checklists shall be passed to the Document Controller (DC) before the end of the workday.

The Document Controller registers the documents and distributes them to the concerned people respective departments as well as to the client or his representative (whatever applicable) promptly.

The QC staff furthermore shall maintain "Daily Activity Records" (record 1). Those records shall summarize all QC operations including activities, tests performed and inspections carried out for every single day during the project life cycle and shall be passed to QA staffs for review.

DOCUMENT SUBMISSION

In principle, all documents that are intended for the client or his representative shall be delivered to their respective office address by hand or postal service. The client or his representative may deliver documents to the contractor in a similar way.

All documents addressed to subcontractors or any other parties who have an office on site or nearby will be deposited for daily collection from the DC. The DC also receives, registers and distributes documents from sub-contractors or other parties.

For further details refer to the Communication Management Plan.

INSPECTION PROCEDURES

All final or intermediate products, whether produced by the contractor, sub-contractors or suppliers, shall be produced through the implementation of approved processes, appropriate use of equipment and control and management of materials and technical services.

Control procedures shall be implemented and serve in a manner that produces a cost effective and reliable product in accordance to the technical specifications and required grade of quality.

For process improvements refer to the Process Improvement Plan.

MATERIAL RECEIVING INSPECTION

QC staff as early as possible but latest one workday after delivery shall inspect all construction materials prior to use.

In order to plan and arrange timely material inspections, the Procurement In Charge will update the delivery schedule continuously.

48 hrs. prior to any material delivery the Procurement In Charge will prepare the "Inspection and Test Request" (form 1) with relevant documents attached thereto and pass it to the document controller. After registration the document controller will pass the document(s) to the material inspector promptly.

Inspection criteria include particularly

- Material identification and documentation (may include but not limited to),
  - "Submittal Review Sheet" (form 3, refer to Submittal Management below),
  - Delivery notes, Test results, Vendor certifications, Certificates of Origin,
  - ...
- Signs of damage,

- Reparable,
- Non-reparable,

- Completeness,
- Compliance with specifications.

Results of the material receiving inspections shall be documented in the "Material Receiving Register" (register 4).

Any identified discrepancy must be resolved prior to re-inspection; otherwise if rectification is not possible, the supplier shall remove the un-approved material immediately.

The Storekeeper will tag newly delivered material highly visible as "Uninspected - Don't Use". Only after the inspection is passed and inspection result is verified, the tag shall be removed and material can be used.

MATERIAL STORAGE INSPECTION

QC staff shall continuously watch over and inspect on a weekly basis the storage and handling of all construction materials. Inspection criteria particularly include storage and handling of material in accordance with manufacturer's recommendations as guidelines.

OFF-SITE INSPECTION

If required material sources and fabrication facilities shall be inspected by QC staff to control that specifications are followed and requirements are implemented. Thus the delivery of unacceptable materials or intermediate products at site shall be avoided.

WORKMANSHIP INSPECTION

QC staff will periodically inspect and verify workmanship to ensure that installation, fabrication or whatsoever is executed in line with the specifications and requirements. This inspection is particular important on work items which cannot be rectified later or reworks would become very difficult and cost intensive.

Workmanship Inspection includes both on-site and off-site.

EQUIPMENT AND PLANT INSPECTIONS

All devices, equipment, machines and plants requiring periodical calibration shall be inspected by QC staffs.

1) When made available at site it shall be inventoried in the "Inventory Register" (register 5).
2) Calibration and maintenance shall be monitored and controlled regularly.
3) All unsuitable items shall be tagged accordingly and not used anymore unless the item is repaired, maintained, calibrated or whatsoever and re-inspected.

MANAGEMENT AND SERVICES REVIEW

The QA manager will periodically review compliance with established processes and procedures. This inspection shall include both the contractor and sub-contractor(s).

(Please refer to the Process Improvement Plan)

SUBMITTAL PROCEDURE

Submittals shall be prepared for any type of material or intermediate product intended for permanent use, installation or whatsoever and thereby becoming part of the ready product, regardless whether the item is produced on-site or off-site and regardless whether it is produced by the contractor, sub-contractor, any supplier or

fabricator etc.

SUBMITTAL TYPES

"Approval" Submittals

"Approval" submittals are the type of submittal for conventional consideration. Examples of "approval" submittals include (but are not limited to) manufacturer's product data, catalogue cuts, shop drawings, samples, etc.. In accepting an approval submittal it is confirmed that the details of the ready product, finishes and materials are consistent with the design concept, specifications and requirements "Review" Submittals

"Review" submittals are the type of submittal for presenting procedures, methods, techniques or sequences prior to implementation. Examples include (but are not limited to) working drawings (i.e. scaffolding, shoring...), proposed equipment, production methods, safety precautions, etc.. In accepting such a submittal it is acknowledged that the proposed method etc. is sufficient to allow quality control and verification. Acceptance of this type of submittal does not relieve the contractor from the responsibility for insuring that the work is performed in accordance with the terms of the contract.

"Information" Submittals

"Informational" submittals demonstrate that the contractor has complied with some requirements and/or specifications. Examples include concrete batch records, daily reports, calibration certificates, test reports, etc.. Action is not required for the contractor to proceed with the works. Although these submittals typically are not approved or rejected, but they still need to be reviewed. If the provided information is determined not to comply with contractual documents, a non-conformance report maybe issued.

TYPICAL MATERIAL AND/OR VENDOR SUBMITTAL DOCUMENT

Complete material or vendor submittals generally may include (but are not limited to):

- "Submittal Cover Sheet" (form 2) with
  - General information (reference number, date, submitter, contact details etc.)
  - Date when material will be needed at site,
  - Short material description,
  - Area of application,
  - Material source,
- Table of content,
- Corresponding plans, specifications and requirements,
- Catalogues (or catalogue cuts),
- Data sheets,
- Shop drawings,
- Certifications,
- Work plans,
- QC plans and templates,
- Testing proposals,
- Diagrams, charts and curves,
- Reference letters,
- Operating manuals,
- Material samples,
- Company profile,
- Organisation charts,

Submittals shall be prepared by the supplier, manufacturer, distributor or whoever supplies the material or intermediate product to the project.

Relevant information shall be highlighted.

SUBMITTAL REGISTRATION

All submittals shall be submitted to the Document Controller for registration and further distribution to the concerned staffs or departments.

SUBMITTAL REVIEW

Submittals shall be reviewed by the designated QA staff. The Reviewer shall:

a) Upon receipt of a submittal prepare "Submittal Review Sheet" (form 3),

b) Record the submittal in the "Material Submission Register" (register 6),

c) Either

   a. If submittal is insufficient for review through DC return it to the submitter for revision or
   b. Provide the submittal with minor comments and/or recommendations (if any) to the QA Manager.

Upon re-submission the designated staff will

a) Update the "Submittal Review Sheet",

b) Provide the submittal with minor comments and/or recommendations (if any) to the QA Manager,

The QA Manager will than propose the material to the client or his representative (engineer) and recommend comment solution (if any). For this purpose he passes the submittal to the document controller for register update and submission.

Upon receipt of the client's or engineer's approval, comments, or rejection (whatever applicable)

a) The "Material Submission Register" shall be updated by the Document Controller accordingly,

b) A copy of the "Material Review Sheet" shall be returned to the submitter for his further action.

ideally a submittal shall be prepared and submitted as early as possible granting at least

- 10 workdays processing time to the contractor,
- 10 workdays processing time to the client or his representative.

In cases where specifications, plans and/or requirements (whatever applicable) were made available late

- The processing time by contractor shall be 3 workdays,
- The processing time by client or his representative shall be 3 workdays.

SUBMITTAL APPROVAL

A submittal approval generally does not extend to the means, methods, sequences, techniques, or construction procedures. Following the review, the conventional response is to approve, approve conditionally, or reject the submittal, but only insofar as the end-result conforms to the design concept and complies with the contract documents.

Possible submittal responses are:

- Approved,
- Approved as noted,
- Revise and Resubmit,
- Not Approved. APPROVAL STAMP

[Project Name]	
ACCEPTED/APPROVED	□
ACCEPTED/APPROVED AS NOTED	□
REVISE AND RESUBMIT	□
NOT APPROVED	□

Acceptance or approval of this document acknowledges only that the information being provided by the contractor conforms to the applicable requirements or specifications and to the design concept of the completed project as set

forth in the contract documents. Contractor solely is responsible for all matters related to fabrication, shipping, handling, storage, assembly, installation, construction (including all safety and environmental aspects of performing the works) and for coordinating the work and the means, methods, techniques, sequences, and procedures of construction to the extent that these items are not specifically addressed by the project specifications or requirements.

By: . Date:

REQUEST FOR INFORMATION

Any clarification of specifications and plans and/or requirements etc. is to be presented using the "Request For Information" (form 4). Requests have to be clear and precise. The referenced plan, specification or whatsoever shall be attached and highlighted.

Requests for information shall be prepared by the enquirer as early as possible. The targeted timeframe for RFI submission is 5 workdays after a certain information requiring clarification was made available for the first time.

RFIs shall be passed to the document controller. The DC registers the document in the "RFI Register" (register 7) and distributes it to the concerned staff respective department.

A copy of every RFI shall be passed to the QA department for the sake of process review and improvement.

RFIs must be answered as early as possible and the targeted response time generally shall not exceed 5 workdays. In cases where due to the severity a problem solution within this timeframe is not possible, the enquirer shall be notified through DC that the RFI is under review and further response will be given as early as possible.

RFI response shall be submitted back to the enquirer through DC. The DC updates the RFI register accordingly.

DEFICIENCY PROCEDURE

The primary goal of the quality program defined in this document is the prevention of non-conformances, reduction of reworks and continuous improvement of processes.

In the unfortunate event that non-conformance couldn't be prevented any identified deficiencies shall be resolved with the approved plan and/or method in a timely and cost-effective manner and re-occurrence shall be avoided to its maximum extend.

PREVENTIVE MEASURES

This Quality Management Plan is intended to be proactive, in order to reduce risks and avoid issues and deficiencies. The primer tools and techniques identified for this project in order to meet this target include (but are not limited to)

- Training and qualification,
- Inspections and verifications,
- Submittal management,
- Calibration and maintenance,

…

Overall quality shall be built in the product rather than to be inspected and rectified in the product.

CONTINUOUS IMPROVEMENT

The QC and QA manager(s) together with the project manager will review any instances where materials, components, assemblies, features of work, or completed products fail to meet the specified requirements, and will take appropriate action to prevent future occurrences.

All project team and workers are encouraged to suggest improvements.

For further details refer to the "Process Improvement Plan".

NON-CONFORMANCE REPORT (NCR)

Identified executed insufficient workmanship or used materials not conforming to the specifications and/or requirements or other non-conformities shall be documented by QC staff in a "Non-Conformance Report" (form 5) and signed by the QC manager.

The Non-Conformance Report will be passed to the document controller for registration in the "NCR Register" (register 8) and for submission to the QA manager and to the responsible/involved/executing department(s), sub-contractor(s), supplier(s) or whatever applicable.

As a deficiency is corrected, a Corrective Action Report (CAR, form 6) shall be filled out by the executor of the subject works and shall be passed to the document controller for registration and further distribution to the QC manager or his designee.

The NCR remains open until the non-conformance satisfactory is resolved, inspected and approved by the QC manager or his designee and client or his representative (if applicable).

For the avoidance of repetition each NCR also shall be included in the "Lessons Learned" (register 9) and shall be evaluated with regards to process improvement by the QA manager or his designee.

SITE OBSERVATION REPORT (SOR)

Significant deviations of any kind that can be corrected on the spot, but do not justify NCR at the discretion of the inspector are documented and communicated by Site Observation Report (form 7).

Such deviations are promptly resolved on the spot so that the SOR is only for documentation in terms of lessons learned and avoidance of future repetition.

In case of recurrence the SOR may be a precursor to the NCR.

On the other hand, above average positive findings may also be documented in the form of a SOR in order to be recorded as good practice in the lessons learned.

CORRECTIVE ACTION SYSTEM

Identified negative quality trends such as repeated NCRs, observations, defects or whatsoever shall be documented by QA staff in a "Corrective Action Report" (form 6) and registered in the "CAR Register" (register 10).

For further details refer to the "Process Improvement Plan".

PROJECT RECORDS AND PROJECT DOCUMENTATION

The QA&QC manager with the assistance of all QC and QA staffs establishes and maintains through document controller the quality file which is a part of the project documentation. The purpose of this file is to maintain a complete set of all relevant documents and records.

The quality file is a compilation of

- Plans,
- Reports,
- Registers and logs,
- Work orders,
- Change orders,
- Correspondences,
- As-built records,
- Certifications and
- Any other relevant records that provide information on the project.

Under no circumstances documents of whatsoever shall be removed from the quality file, even if superseded. In such case revisions shall be prepared and kept.

FILING SYSTEM

The QA &QC department will file its documents (here softcopies) within the projects filing scheme.

Hardcopies and electronic data shall be maintained simultaneously. Each data folder shall be represented by a corresponding box file and vice versa. The names of both box file and data folder shall be same; their contents consequently shall be same.

FILING SCHEME

Level 1 Document.docx	Level 2 Document.docx	Level 3 Document.docx	Level 4
05_Quality	051 Initiation	[empty]	
	052_Planning	0521_Quality Mgmt	
	053 Execution	0531_Quality Assurance	
	054 Monitor & Control	0541_Quality Control	
	055_Closure	[empty]	

It is undisputed that further folders may be required over time to sort files in a logic and manageable order.

However

1) No new folders shall be created on folder level 1 and 2, nor shall any existing folder be renamed and no data files such as e.g. word files, excel spreadsheets, drawings, PDF-files, images or whatsoever shall be saved at this level.

2) Folders on folder level 3 shall not be renamed and no data files such as e.g. word files, excel spreadsheets, drawings, PDF-files, images or whatsoever shall be saved at this level. On level 3 limited additional folders can be created upon approval by the responsible manager.

3) Preferably additional folders only shall be created on level 4. The responsible manager must be consulted and the filing scheme must be updated and communicated accordingly.

DATA BACKUP

All electronic data shall be backed up regularly.

Further details with regards to correspondences, data storage and data integrity etc. as well as the complete project filing scheme can be found in the Project Communication Management Plan.

TESTING

Testing shall be performed, recorded and reported and test results shall be verified under the responsibility of QC staffs to ensure that specifications and requirements are met.

Prior to the first execution of a new work item testing methods and frequency of testing shall be reviewed, communicated to concerned staffs and shall be in line with applicable standards, contractual requirements,

plans, approved method statements or whatever agreed upon.

GENERAL TESTING PROCEDURE

1) Not less than 24 hrs. prior to any testing client or his representative by means of "Inspection and Test Request" (form 1) shall be notified of the testing and may then at their own discretion decide to attend the testing or not.
2) Prior to any test the QC staff shall verify that
    a) Required equipment is available and calibrated,
    b) Testing criteria and procedure are known,
    c) Personnel are capable to perform test and operate equipment.
3) Upon verification of requirements the test may proceed and shall be witnessed and documented by QC staffs using a test specific form. Standard forms provided by a 3rd party testing organisation(if applicable) shall be acceptable.
4) All test results shall be
    a) Registered in the "Test Register" (register 11) and compiled in the quality file,
    b) Verified by QA staffs,
    c) Submitted to the client or his representative (if required).
5) Any failing test result shall be recorded in the "Deficiency Register" (register 12) and the test shall be repeated as early as possible (unless otherwise agreed). Failed tests shall be subject to review with regards to process improvement.

3RD PARTY TESTING

If testing by a third party laboratory is required, whether on-site or off-site, QA staffs shall be responsible to verify 3rd party's compliance to applicable standards and therefore shall review laboratory's historic data such as QA & QC procedures, calibration records, logs for similar testings etc.3rd party testing organisations not being able to provide such records shall be avoided if possible. In case of regular testing the review maybe carried out periodically.

A 3rd party testing organisation shall be capable to perform a required test within 2 workdays (if applicable) from receipt of samples.

The QC Manager at his discretion together with client or his representative may decide in general or case-by-case if witnessing the test by QC staff is required. Client or his representative may decide at their own discretion to attend and witness the testing.

In any case 3rd party test results shall be verified by their own senior laboratory personnel.

TEST RESULTS

Ideally the original test record shall be submitted to the QC department within 48 hrs from the test. Advance copies of successful tests sent by fax and/or email shall be deemed acceptable in order to proceed with the works.

Once test results are received, they shall be registered and distributed by document controller and shall be verified by QC staffs as to:

- Completeness of documents,
- Observance of the specified testing procedures,
- Acceptability of results.

TEST AND OTHER EQUIPMENT CALIBRATION AND MAINTENANCE

Test and measurement equipment shall be regularly maintained and calibrated according to the manufacturer's specifications and recommendations. The service provider shall provide calibration and maintenance records. (To be read in conjunction with equipment and plant operating inspection procedure.)

QA & QC MEETINGS

MUTUAL UNDERSTANDING MEETING

The QC and QA manager(s) will invite the client or his representative as well as key project team staffs and key field staffs in order to introduce the QC and QA systems and requirements after the PQMP is submitted and prior to the start of construction.

The purpose of this meeting is to establish a mutual understanding of QC and QA under the contract.

PREPARATORY QC MEETING

Unlike other meetings, this meeting is not usually held in the meeting room but at site.

(please refer to .Quality. Control above)

WEEKLY MEETING

After the start of construction, the QA & QC manager(s) will conduct weekly QC meetings at the work site with the QA & QC staff(s), and the responsible site engineer(s) who are performing the work on the current work packages. Further project staffs maybe asked to attend as required. The client or his representative may attend the meeting at his own discretion.

The following shall be accomplished at each weekly meeting (as applicable):

- Review the minutes of the previous meeting.
- Review the schedule and the status of work and rework.
- Review the status of submittals.
- Review the work to be accomplished in the next two weeks and identify the documentation required.
- Resolve production problems.
- Address items that may require revision of the Quality Management Plan.
- Review the status of training requirements, as applicable.

The QA & QC manager(s) or his designee prepares minutes of meeting (MOM) and provides a copy to all attending departments, sub-contractors, client (whatever applicable) within 2 workdays after the meeting ended. Comments on the minutes of meeting (if any) shall be presented in writing to the QA & QC manager(s) within 2 workdays after MOM being received.

APPROVAL OF DELIVERABLES xyz

DEFINITIONS

For the purpose of this plan and any further document developed through its use the following terms are used: Quality Control (QC) The primary objective of QC is to anticipate potential risks and issues with regards to the grade of quality. The QC measures are to be adequate to cover all operations both on-site and off-site.

Quality Assurance The primary objective of QA includes verification, audits and evaluations of (QA) implementation of the quality control system by the contractor, its sub-contractors and suppliers.

TS:

1

2

May	means an optional action
Shall	means a mandatory action

ACRONYMS AND ABBREVIATIONS

CAR ..... Corrective Action Report

COO ..... Certificate of Origin

DC ..... Document Control(ler)

Insp ..... Inspection

IR ..... Inspection Register

ITR..... Inspection and Test Request

NCR..... Non-Conformance Report

PQMP..... Project Quality Management Plan

QA..... Quality Assurance

QC..... Quality Control

QMP..... Quality Management Plan

RFI..... Request For Information

APPENDICES

Project Site Layout Plan Organisational Chart QA & QC dept.

Preparatory Meeting Checklist Initial Inspection Checklist Follow-up Inspection Checklist

Inspection and Test Request Submittal Cover Sheet Submittal Review Sheet Request For Information

Non-Conformance Report Corrective Action Report Site Observation Report

Inspection Register

Completion Inspection Register

Material Receiving Register

Inventory Register

Material Submission Register

RFI Register

NCR Register

Lessons Learned Register

CAR Register

Test Register

Deficiency Register

Daily Activity Records

References

https://ori.hhs.gov/education/products/rcradmin/topics/data/tutorial_12.shtml

https://en.wikibooks.org/wiki/DBMS

http://searchsqlserver.techtarget.com/definition/database-management-system

https://en.wikipedia.org/wiki/Database#Performance.2C_security.2C_and_availability

https://www.techopedia.com/definition/811/data-integrity-databases

https://ethics.csc.ncsu.edu/privacy/database/study.php

http://mazsola.iit.uni-miskolc.hu/DATA/research/tempus/discom/doc/db/tema11.pdf

http://gpdb.docs.pivotal.io/4340/admin_guide/perf_intro.html

http://searchdatamanagement.techtarget.com/answer/Four-common-data-integration-issues-and-challenges

http://www.informit.com/articles/article.aspx?p=29321&seqNum=4

http://www.learn.geekinterview.com/data-warehouse/dw-basics/what-are-data-access-tools.html

https://opentextbc.ca/dbdesign01/chapter/chapter-3-characteristics-and-benefits-of-a-database/

https://en.wikipedia.org/wiki/Backup#Managing_the_backup_process

http://www.tutorialspoint.com/dbms/dbms_data_recovery.htm

http://tdan.com/database-auditing-capabilities-for-compliance-and-security/8135

http://searchoracle.techtarget.com/definition/distributed-database

https://en.wikipedia.org/wiki/Transaction_processing

https://www.techopedia.com/definition/19782/hierarchical-database

https://en.wikipedia.org/wiki/Hierarchical_database_model

http://www.tutorialspoint.com/dbms/relational_data_model.htm

https://en.wikipedia.org/wiki/Relational_model

http://dbmsenotes.blogspot.co.za/2014/03/comparison-of-data-models-data-models.html

http://www.unixspace.com/context/databases.html

http://www.computerweekly.com/feature/Choosing-the-right-database-management-system

https://msdn.microsoft.com/en-us/library/aa290282(v=vs.71).aspx

http://www.wikihow.com/Develop-an-IT-Change-Management-Program

http://www.itsmcommunity.org/downloads/Sample_Process_Guide_-_Change_Management.pdf

http://is.oregonstate.edu/sites/is.oregonstate.edu/files/projects/change-management-policy.pdf

www.iso27001security.com/ISO27k_Model_policy_on_change_management

http://www.ofnisystems.com/services/validation/user-requirement-specifications/

http://www.ivtnetwork.com/sites/default/files/Computer.pdf

http://oss-watch.ac.uk/resources/versioncontrol

https://en.wikipedia.org/wiki/Change_management_(engineering)

https://en.wikipedia.org/wiki/Data_integrity

http://www.cisco.com/c/en/us/products/collateral/services/high-availability/white_paper_c11-458050.html

https://www.veracode.com/blog/2012/05/what-is-data-integrity

http://www.pma.doit.wisc.edu/plan/3-2/what.html

http://aspalliance.com/1137_Understanding_Project_Quality_Management.all

http://www.visitask.com/project-quality.asp

file:///C:/Users/User/Downloads/PM4DEV_Project_Quality_Management.pdf

http://knowledge.apm.org.uk/bok/quality-management

http://asq.org/service/body-of-knowledge/tools-cost-benefit-analysis

https://www.wyzant.com/resources/lessons/math/statistics_and_probability/introduction/sampling

http://project-management-knowledge.com/definitions/t/trend-analysis-technique/

http://www.dummies.com/how-to/content/performing-a-costbenefit-analysis.html

http://asq.org/learn-about-quality/data-collection-analysis-tools/overview/control-chart.html

http://stbweb01.stb.sun.ac.za/projektus/rolspelers/indexE.htm

https://en.wikibooks.org/wiki/Project_Management/PMBOK/Quality_Management

http://www.processimpact.com/articles/qualreqs.html

http://www.gla.ac.uk/media/media_26397_en.pdf

http://www.businessballs.com/brainstorming.htm

http://www.discover6sigma.org/post/2009/02/affinity-diagram/

http://asq.org/learn-about-quality/idea-creation-tools/overview/nominal-group.html

https://www. smartdraw.com/flowchart/

http://asq.org/service/body-of-knowledge/tools-project-prioritization-matrix

ec.europa.eu/idabc/servlets/Docc9f3.doc?id=18627

http://it.toolbox.com/blogs/enterprise-solutions/managing-project-quality-checklist-8499

http://www.softwareresearch.net/fileadmin/src/docs/teaching/SS06/PM/PMBOK8.pdf

http://www.mymanagementguide.com/stating-project-quality-policy/

http://www2a.cdc.gov/cdcup/library/pmg/implementation/ll_description.htm

http://www.intechopen.com/books/total-quality-management-and-six-sigma/project-costs-and-risks-

https://www.mica.edu/Documents/EFG/D_BudgetTemplateSTAFF.xls

http://www.pma.doit.wisc.edu/plan/4/what.html

https://en.wikipedia.org/wiki/Configuration_management