

- As part of Continuing Education, I taken practical training course (64-credit hrs) of Network Specialist to gain further knowledge and working exposure of computer networking that was completed on 15 Feb 2004.

In Computer Security I studied and gained knowledge about historical Review of Computer Security, Threats and attacks to Data Security, A Generic Model of Network Security, Introduction to Cryptology, Cryptographic and Cryptanalysis Techniques, Symmetric and Asymmetric Encryption Algorithms, Data Encryption Standard, Triple Data Encryption Standard, Advanced Encryption Standard, Simplified AES, Key Exchange Problem in Symmetric key cryptography, Public Key Cryptography, Key Exchange Approaches using Symmetric Key algorithm and Need for Public Key Cryptography, Introduction to Number Theory, Diffie-Hellman Key Exchange Algorithm, Rivest-Shamir-Adleman (RSA) Algorithm, Digital Signatures, Authentication Requirements, Message Authentication Code, Hash Functions, RSA Approach, Digital Signature Standard (DSS), Email Security, PGP Algorithms, IP Security, Virtual Private Networks, Web and WAP Security, Transport Layer Security, IDS, Firewall and Viruses, Architecture and Policies in Intrusion Detection Systems, Operating Systems Security.

This subject has helped me to practically understand and analyze various encryption at program level for example in passwords, encrypting webconfig details and license key generations, transaction files encryption etc

7) Mathematical foundations of computing

In this knowledge area, I studied following subject

- Finite Automata Theory

In **Finite Automata Theory**, I studied about theory of Computers, concepts of Sets, Logic, Functions, Relations, Languages, Regular Languages, Language Concepts, Strings, Distinguishing one string from another Unions, Intersections and Complements , Recursive Definitions Regular Languages, Regular Expressions, Finite Automata, formal definition of FA, The Memory Required to recognize a Language Transition Graphs, Kleens Theorem, Proofs of Kleens Theorem, Concept of Non-Determinism, proliferation of states, formal, definition of NFA, conversion of NFA to DFA, NFA with Moves (Conversions b/w NFA- to NFA and then FA, Finite Automata with Output, Moore Machines, Mealy machines, conversions of Moore and Mealy machines, sequential circuits, Language Grammar, Turing Machines, Definition and Examples, Combining Turing Machines, Variations of Turing Machines.

Certificate of Completion - Networking Specialist- KISWA

BS(Computer Engineering) Transcript

Course, CE302

openzka02.com

8) Operating System Knowledge Areas:

In these knowledge area, I studied following subjects

- System Programming-I and II
- Advance Principals of Operating Systems

BS(Computer Engineering) Transcript

Course, CE303 & CE306 & CE403

In **system programming** I studied about general principal of operating system like Operating Systems, Compiler, Assembler, Linker Loader, Operating Systems History, Goals and Evolution of Operating systems, Process and Thread management, Multithreading, Kernel and User Modes, CPU Scheduling, Problems of cooperative processes, Concurrency Control and Synchronization, Semaphores, Deadlocks, Memory management and virtual memory, Relocation, Fragmentation, Paging and Demand Paging, Secondary storage, Disk Structure and Scheduling, File systems, I/O management, Operating System Security.

In **Advance Principal of Operating system**, I studied about Architecture of Distributed System Communication: Remote Procedural Calls, Synchronization: Mutual Exclusion, Synchronization: Distributed Mutual Exclusion, Synchronization: Deadlock detection and Algorithms, Distributed Scheduling, File System, Distributed File System, Fault Tolerance and Recovery, Case Study of Distributed Operating System, Managing Multiprocessing

systems, Real-Time Operating System, Operating System Security.

9) Software Domain Knowledge Areas:

In these knowledge area, I studied following subjects

1. Relational Database Management System
2. Numerical Methods
3. Simulation and Modelling

BS(Computer Engineering) Transcript

Course, CE301 & MS302 & CE407

Relational Data base Management System

In this subject I studied about Introduction to Relational Database Management Systems and databases and its benefits and essentials. I studied about drawbacks of File Systems, Database management system, Data Dictionary, Relational Database, Properties of Relational Databases, E. F. Codd Twelve rules, Database users, Database Systems Architectures (Three level architecture), Mappings, Centralized Systems, Client-Server Systems, Parallel Systems, Distributed Systems, Data Modelling using Entity-Relationship Modelling, Conceptual Modelling with E-R Approach, Entities and Entity Types, Attribute Types, Entity Sets, Relationship Types, Sets and Instances, Relationship Degree, Role names, and Recursive Relationship, Constraints on Relationship Types, Weak Entity Types, ER diagrams, Naming Conventions and Design issues, Proper naming of Schema Constructs. I also studied a Case Study of a Relational Database ER to Relational Mapping, Entity Conversion, Attribute Conversion, Multi-valued Attributes, Composite Attributes, Derived Attributes, Relationship Conversion, Data Modelling using Extended/Enhanced Relationship Modelling, Relational Database design, Functional Dependencies, Trivial and Nontrivial Dependencies, closure of a set of Dependencies, Armstrong's Axioms, Closure of a set of Attributes, Irreducible set of Dependencies. I also studied Relational Database, Normalization, An Introduction to Relational Data Integrity and its need, Candidate Keys, Primary Keys and Alternate Keys, Candidate keys and Nulls, Foreign Keys, Foreign key Rules, Foreign key and Nulls. I also studied Case Studies of some Relational Databases. I studied Relational Operators (Relational Algebra), PL/SQL, Data Protection, Recovery, Transaction Concept, Transaction Recovery, ACID Properties, System Recovery, Two-Phase Commit, SQL Support, Concurrency, Storage and File structure, Data Warehousing, Data Mining, Database Security and System Administration.

- As part of Continuing Education, I taken one month training of Essentials of Microsoft SQL Server 2008 from 20th April 2013 to 30th May 2013 from Rahimabad Ismaili Students Association (RISA).

Certificate of Participation in Workshop from RISA.

This course has provided me hands on practical experience to work with Microsoft SQL Server 2008 and latest advancement in database systems.

Practical Exposure:

I also learned the foundation of relational database management system as part of software engineering during bachelor program, in which I learned the structure and relationships of data base tables and relationship between stored items of information and learned to work with database software applications to capture and analyze data. I learned to work with Microsoft access database during my academic period. This course has helped me to understand and learned to create data base definitions, data base updating, forms, reports and administration of database and its implementation with various application as back end data sources .

In this course I also learned how DBMS serves as an interface between data base and end users application programs, and this knowledge helped me to develop skills to perform various roles to review the database schema, data base structure, SQL queries, stored procedures, views used in applications and testing software applications with various data types.

During my professional career, I learned to implement and analyze databases with various application technologies like Java and Dot net. I worked in Microsoft SQL 2005 while working in Mobile Complete as a backend database of Mobile testing solution (Java based application), work to review data base schema and table structures, and actively participated in normalization of database and its implementation. In various software development houses, I played key role in designing and reviewing data base structure. I worked with Microsoft SQL Server 2008R2 during Home Remittance System implementation (Episode-3). I worked with IBM DB2 in Bank Al habib while working with Core Banking System.

In **Numerical Methods**, I studied about Error estimation in numerical computations, Errors, absolute error, relative error, percentage error, sources of errors, significant digits, error in function evaluation, and error in arithmetic operations. Solution of non-linear equations, Bisection, Newton-Raphson and fixed point iteration methods. Interpolation and extrapolation, Newton divided difference formula, Lagrange interpolating polynomials. Solution of linear system of equations, LU decomposition and Gauss-Siedal iteration method. Numerical differentiation and integration, Various methods of differentiation and integration. Solution of ordinary and partial differential equations, Numerical methods of solving first and second order ordinary differential equations and first order partial differential equations. Statistical inference, Estimation of parameters such as mean and variance, classical and Bayesian methods of estimation. Hypothesis testing, z-test, t-test and goodness of fit test. Simulation, Random numbers and their generation, generation of random deviates from different distributions, special and general methods of simulation. Simulation of probability models and test of goodness of fit.

In **Simulation and Modelling**, I studied about the basic aspects of modelling and simulation. It includes knowledge about statistical models, queuing theory, random variable generation, discrete system simulation, simulation languages, graphic output with animation, and validation of simulation models.

Fundamental Engineering Course Studied During Masters Program,

In **Data Mining**, I studied about the Concept of Data mining and data warehousing, Data Pre-processing, Descriptive data mining that include characterization and comparison, Classification and prediction, Cluster analysis, Mining Complex types of data, Application and trends in data mining.

In this subject, I also studied about Data Preparation Techniques, Outlier and missing data analysis, Data Reduction Techniques, learning methods in Data Mining, Statistical Methods in Data Mining, hierachal, agglomerative and Naive Bayesian methods, Decision trees and decision rules, Association Analysis and Rules, Other Soft Computing Approaches in Data Mining, Artificial Neural Networks, Fuzzy Logic and Fuzzy set Theory, Genetic Algorithm, Evolutionary algorithms.

- As part of Continuing Education, I participated in two days workshop named 'Search Engine Optimization - SEO' that was held from 2nd March to 3rd March, 2013 organized by Rahimabad Ismaili Students Association (RISA).

This workshop had broadened my understanding to know how SEO Optimization can use Data Mining Approach. As we all know that huge amount of data is digitally available on the world wide web (internet). It may be WebPages, images, videos and other information. This pool of information is changing dynamically and due to this reason, it is becoming difficult to find useful information on the internet. Here web crawler is useful to us for automatic data and web mining. For most of the users who are trying to explore the information based on web pages, then search engine provides the gateway to them.

MS (Computing)
Transcript

Course, SE5224

Certificate of
Participation -
Workshop-SEO
from RISA

Example 3-2: Setup and Configuration of Disaster Recovery Site for Home Remittance System

After complete implementation of Home Remittance System (third party) in Bank Al Habib Lt, all technical support queries were responsibility of the implementation team comprised of Project Manager and senior technical person.

Please refer to episode-3 for complete details about my role and project.

Problem Background: The system was working at Production (Live Environment) for several weeks before it crashed one day, when business operations had just started in the morning. Due to this incident, whole business was completely down on that day.

Previously, IT Infrastructure team used to take backups of entire system on Tape Drive on daily basis so they got the system up and running next day but one whole day the business suffered a major downtime.

Although, my job role was not directly related to IT Infrastructure, but because we had implemented the new system (Home Remittance) and I had to constantly communicate with vendor for system related issues, I decided to discuss with IT Management for a solution.

After developing an initial understanding with the management, I recommended to setup a disaster recovery site (secondary site) also called DR Site for this system, so that if primary servers go down, the business operation could be redirect to secondary server. The reason was that my project manager and I, both were responsible to provide technical support to system users.

Implementation: I worked on this project under close coordination with IT infrastructure for hardware and installation related tasks. The implementation was comprised of following steps:

1. Arrange Server Hardware for DR Site having same configurations that Primary Servers have. There were two servers, one for Application and Other for Database.
2. Arrange availability of free IP for Secondary Servers (Application and Database)
3. Deployment and configuration of Application on Secondary Server
4. Configuration for database synchronization, between Primary and Secondary servers with replication mechanism.
5. Shadowing of an Archive folder (backend folder) that contained Exchange Company Process Files, between Primary Server and Secondary Server.
6. Test the Disaster Recover Site by switching from Primary Server to Secondary Server on real time basis and verify that its working fine ensuring business operation remain operating without any problem.

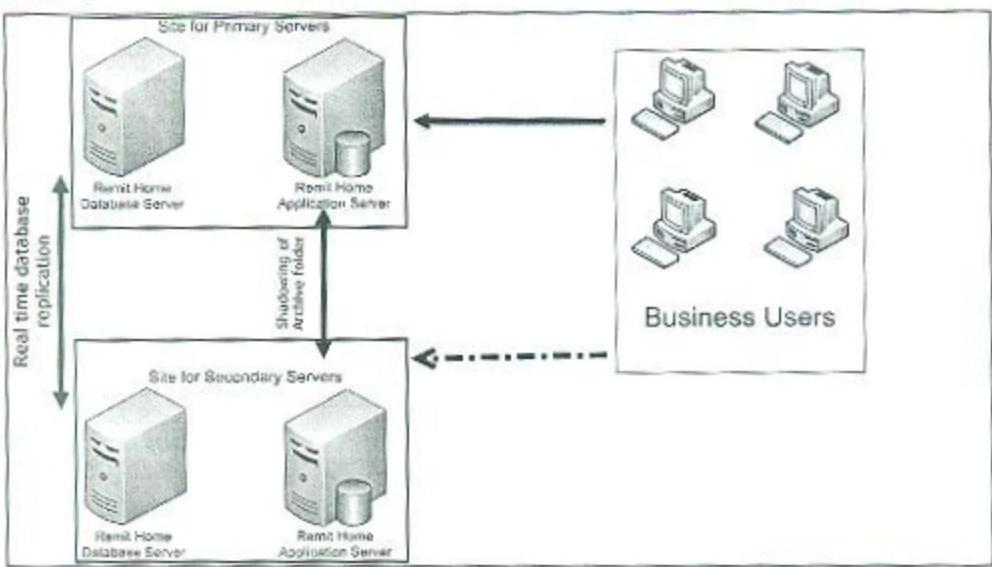
In this task, I worked in close coordination with IT infrastructure team to arrange hardware and free IPs, to assign to Secondary Servers. In this case, my knowledge of computer hardware, operating systems and networking has helped me to understand and apply that knowledge to configure system and setup the environment.

I deployed the same version of application for primary server to secondary server. In order to synchronize the database from primary database server to secondary database server, I first restored the backup of primary server to secondary server and configured peer-to-peer replication. Peer-to-peer replication propagates transaction level consistent changes in real-time from primary server to secondary server and vice versa. It provides replication of data redundancy on both nodes on real time basis.

There was an another folder in application server under application website that contained the archive files of exchange companies, that folder was used to stored decrypted files of exchange company. These files were uploaded by the users from application interface. The data size of this folder was increasing rapidly as there were hundreds of exchange company files were being loaded on daily basis.

In order to secure that folder, I configured shadowing on that folder to get the real time copy to secondary server application folder under same path and vice versa if applicable.

Following is the diagrammatic representation of Home Remittance System's Primary and Secondary Servers



DR (Disaster Recovery) Site Testing:

Following were the steps of DR Site Testing

1. Intimate users to stop working on Live system
2. Switch the IPs of Secondary Application server with Primary Application Server

For Example:

Before Switching

Primary Application Server IP is 10.200.36.41 (Live Application Server)

Secondary Application Server IP is 10.200.36.141 (Replica Application Server)

Note: In Secondary Application Server, the application settings contained same settings of Primary Server i.e. WebConfig and other Settings.

After Switching IPs

Primary Application Server IP is 10.200.36.141

(After Switching, This will be Replica Application Server)

Secondary Primary Application Server IP is 10.200.36.41

(After Switching, This will be Live Application Server)

3. Similarly switch the IPs of Secondary Database server with Primary database Server

For Example:

Before Switching

Primary Database Server IP is 10.200.36.51 (Live Database Server)

Secondary Database Server IP is 10.200.36.151 (Replica Database Server)

Note: The instance name of Database Server is same as of Primary Server Database instance.

After Switching IPs

Primary Database Server IP is 10.200.36.151

(After Switching, This will be Replica Database Server)

Secondary Database Server IP is 10.200.36.51

(After Switching, This will be Live Database Server)

4. Archive folder is already in synched between primary and secondary server
5. After switching, verify that user is able to login to system and execute business operations as usual.

Note: The DR Site Testing was also done for individual Application Server or Database Server and both.



Element Four

Engineering specialist knowledge that provides theoretical frameworks and bodies of knowledge for the accepted practice areas in the engineering discipline, much is at the forefront of the discipline

Context

In addition to a broad understanding of fundamental engineering principles, professional engineers are required to develop specialised engineering knowledge to support their practice. This may be aligned with traditionally defined fields of specialisation such as structural, industrial or geotechnical engineering, coherent combinations of such traditional areas, or more recently emerging fields such as software, biomedical or mechatronics engineering.

Advancing technological knowledge and complexity means that technical specialisation is increasingly necessary for an engineer to remain abreast of technological development throughout their career.

Washington Accord graduates are expected to be able to apply this engineering specialist knowledge to solve complex engineering problems.

Performance Indicators

Evidence of sufficient depth of knowledge to support practice within one or more recognised field of engineering

Evidence of a systematic understanding of the coherent body of knowledge related to a particular field of engineering, its underlying principles and concepts, its usage and applications, and analytical and problem solving techniques

Ability to apply specialist engineering knowledge to solve complex engineering problems

Summarise your specialist engineering knowledge and how it has been developed through formal study, on-job learning and/or continuing professional development.

Note, please cross-reference to your academic transcript(s) and continuing professional development records, as appropriate.

Software Test engineering and Software Quality Engineering are major knowledge areas related to software engineering process disciplines. In these discipline, I have worked for various software life cycle processes in different organization and have gained thorough understanding and knowledge with relevant skills and practices.

I have studied following specialized subjects during Masters Programme.

Software Quality Engineering:

This is a specialized course that has added additional value to my over all experience and knowledge while working as Software Quality Assurance Professional in various organization. This course has helped me to develop critical awareness and understanding about its importance in software engineering. I learned to develop various methods for quality assurance and quality control and learned to evaluate quality issues in the implementation of software systems using various design methodologies. I gained in-depth understanding and important of Regression Testing procedures and learned to develop effective testing strategies. This course has taught me how to measure quality of software with various metrics. I gained thorough understanding and knowledge about the Configuration Management techniques and its role in achieving software quality.

In this course, I studied about quality and reliability, quality assurance and quality control procedures throughout the software development life cycle, reviews and walkthroughs.

Provide annotations to your supplementary evidence (document and page number)

MS (Computing) Transcript
Course: SE5202

Implementation of software from design methodologies such as Data Flow Diagrams and Object-oriented Design, suitability of design and implementation for different categories of software systems, coding defensively, exception handling and coding for quality, error message design. In this subject, I studied about testing procedures, black box, white box, stress, thread and unit testing methods, statistical testing, test planning and its role in the software development life cycle. I also studied about verification and validation procedures, introduction to formal verification and proving programs as correct. Metrics for quality and reliability, complexity metrics. Configuration management and the management of software maintenance. In this course, I went through group discussions, study of various research papers, case studies and course work assignments.

Software Requirement Engineering (SRE):

This specialized course has given me in-depth knowledge and understanding of Software requirement Engineering and its application in software development industry. In this course, I gained exposure toward SRE processes, methodologies and techniques as per the industry standards. The course has gave me detailed understanding of Software Requirement Engineering Life cycle. In this course, I went through group discussions, study of various research papers, case studies and course work assignments.

In this course, I studied about Issues and terminology of software requirement, System and Software Engineering Concepts and Overview, The Concept of Operations Document as per IEEE Guideline, Software Requirement Analysis and Specifications and Techniques. I also studied about IEEE recommended practices and guide for system development, Identifying and Measuring Quality in Software Requirement Specifications, Software Requirements Methodologies and Tools, Requirements and Quality Management, Verification and Validation Processes, Software System Engineering Process Models.

Software System Design and Architecture:

In this course, I gained in-depth understanding about how Software Architecture holds the key to achieve system quality. I studied that Software architecture is a reusable asset that can be applied to subsequent systems, and is crucial to a software organization's business strategy.

In this specialized course, I gained knowledge and understanding about latest development in the field, concepts and best practices that are being followed in industry related to Software Architecture, which mean how software is constructed and the system elements are meant to interact. In this course, I studied in detail about the implementation, algorithm, and data representation. I also studied designing, specifying, and validating a system emphasizing the importance of the business context in which large systems are designed, Software architecture in a real world setting, reflecting both the opportunities and constraints that companies encounter.

In this course, I went through group discussions, study of various research papers, case studies and course work assignments.

In this subject, the summary of the course topics I studied are **Introduction**: Its contained Basic Concept and Motivation, Software Paradigms, Software Architecture and its need. **Software Architecture Patterns**, **Introduction to Architecture Patterns**: It contained Components, Connector, Attributes, Interface, and Configuration. AP1 (Data Flow Systems), AP2 (Call and Returns systems), AP3 (Independent Components), AP4 (Virtual Machines), AP5 (Data Centered Systems), Other Architectural Patterns. **Design Patterns**: It contained Design pattern introduction, Creational Patterns, Structural Patterns, Behavioural Patterns, Sample Design Pattern description and usage. **Software Architecture Global Analysis**: It contained Introduction to Architectural Analysis, Global Analysis details, Case Studies. **Software Architecture Conceptual View**: It contained Introduction to conceptual View, Case Studies. **Software Architecture Module View**: that contained Introduction to module View, It contained Module sub system, layer and interface, Case studies. **Software Architecture Code View**: It contained introduction, source component, and Intermediate

MS (Computing)
Transcript

Course:
SE5101

MS (Computing)
Transcript

Course:
SE5102

component. Deployment component: It contained build procedure, configuration management and Case Studies. Software Architecture RUP View: It contained The 4-View Architecture model, 4+1 views architecture, Use Case View, Logical View, Process view, Component View, Deployment View, and Implementation View. Architecture Evaluation Techniques: It contained Architecture Tradeoffs Analysis Method (ATAM), Cost Benefit Analysis Method (CBAM) Components, Framework and Product Lines: It contained Components, Framework and its examples, Software Product Lines, Service Oriented Architecture.

The two major life cycle processes in which I have extensively worked are

1-Engineering Processes:

This process directly specify, implement or maintain the software product, its relation to the system and its documentation. In engineering Process I have worked in the capacity of

- a) Software requirement analysis
- b) Software Testing
- c) Functional and System Integration Testing

2-Customer Supplier Processes (Software Supporting Life Cycle Processes):

This process directly affects the customer. It support development and transition of software to the customer and provides support for the correct operation and use of the software. In this process I have worked in the capacity of

- a) Software configuration management
- b) Software quality assurance
- c) Verification and Validations
- d) Software Support
- e) Documentation and Audits.

Software Requirement Analysis:

While working in various software development organizations, I have been involved in developing thorough understanding of system requirement, identify potential risks, flaw and incompleteness within requirement as per the standards and procedures. I have worked to reviewing system architecture for product road map and recommended technical and non-technical directions for system upgrades and its usability.

During my master program, I have studied following courses that directly benefit in my overall knowledge and learning in terms of advance practices of software engineering processes and its application mentioned in detailed below.

- Software Requirement engineering (Course detail mentioned above)
- Software System design and architecture (Course detail mentioned above)

It was very essential for me to be involved in software requirement analysis phase because I was responsible for pointing out the un-testable requirements, un-clear requirement, and was able to understand the system requirement in early stages. In this way, I was able to develop system understanding thoroughly and started to think aggressively about the test specifications by the time the code arrives. This practice has always helped me to not accept inadequate requirement documentations and system design documents upon which the actual testing effort is based; involving myself in requirement phase always helped me to create suitable tests before the code arrive and review practice has always helped me to raises several critical issues before development started. This way I was not only responsible for quality checks of the system but assure that the quality of the system is as per user need and actual requirement. Understanding requirement and system architecture in early stages has always helped me to determine the appropriate estimation of testing effort that I had to

Please refer to
Experience
Letters from
Companies

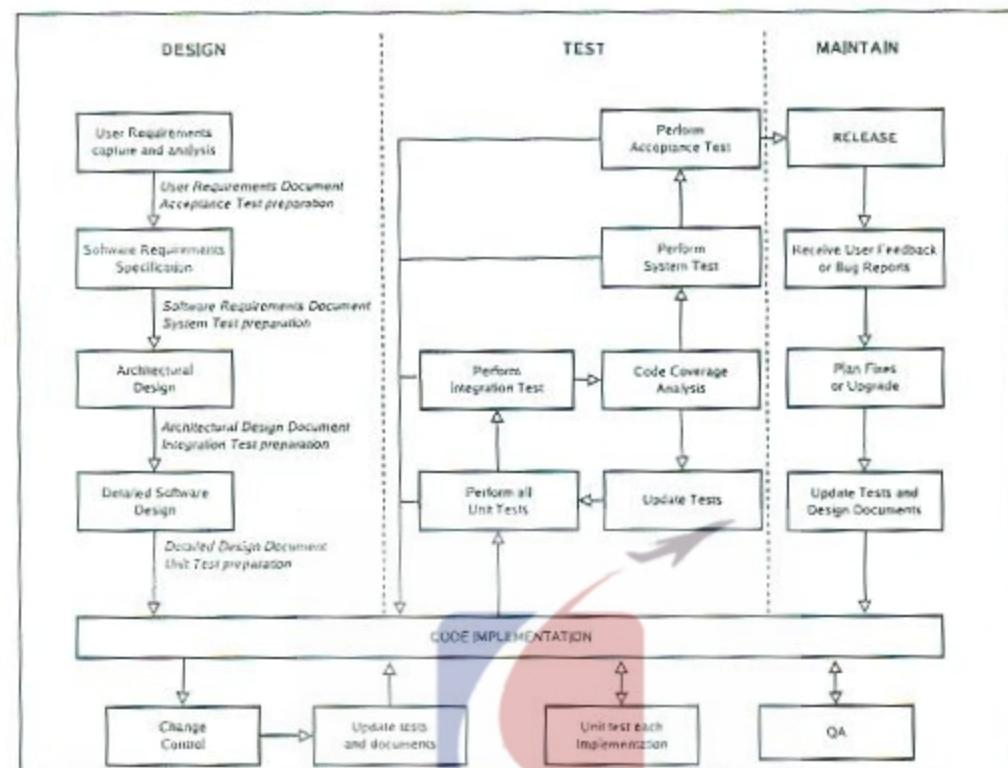
Mobile
Complete,

Wavetec
(Pvt)Ltd.

LN
Technologies

communicate to project manager with proper justification.

Working as software quality assurance professional and involving in every stages of software development life cycle, I have gained thorough experience and knowledge of how requirement actually elicit-ate and defined. I have actively worked with business analyst and contributed significant roles in finalizing system requirement and identifying testing effort estimations. In short I have always involved myself with specific role in each stage of design phase depicted in below diagram



Example 4-1: Reviewing User Story of CIF Mechanism for Integration Testing between Old Core Banking and New Core Banking System

CIF(Customer Information Folio), is the mechanism that was being introduced in Al Habib Core Banking System in order to integrate with new core banking system (Orbit). As a QA Professional, I was involved in thoroughly reviewing of user story of CIF Mechanism

There were various issues with different levels of severities in requirements and user scenarios mentioned in user story, which I identified while reviewing it. Since two system had communicate with each other on the basis of unique CIF Numbers generated for each unique document IDs, the document IDs for Natural Person may be CNIC, Passport, CRC, Birth Certificate and document IDs for Entity may be NTN or Registration Numbers. There were various terminologies that were present in new system but did not exist in old system for which table structure from both system had to mapped with each other. I identified various issues related to them and finalized related design changes working with business analyst.

It was a very challenging project and took several weeks to fully test it. The Scope of Integration Testing is explained later in this element under Integration testing onward.

Software Design Review Process:

Software design review process is a systematic way of inspecting software design to verify that specified requirements of design are adequate and design meet the specified requirement. I have been involved in software design review processes in various

Attached is the user Story.

organizations. Following is the checklist of software design review process I have created while working in Mobile Complete

Structure

- Has the design been decomposed to the point at which the next phase of design can begin?
- Has each component been completely and testably specified?
- Is the modular decomposition consistent with local standards for modular strength and coupling?
- Does the architecture allow for implementation of all of the requirements?
- Has the architecture been adequately decomposed?
- Have the system functions been appropriately allocated to components?
- Does the architecture provide an adequate base for subsequent design work?
- Is the architecture feasible for implementation?
- Have maintainability issues been adequately addressed?
- Can the program set be integrated and tested in an incremental fashion?

Data

- Are global constants and passed data parameterized?
- Has all the data been properly defined?
- Are data structures and element names meaningful and compliant with naming conventions?

Correctness

- Does the architecture avoid unnecessary redundancy?
- Have all reliability and performance requirements been addressed?
- Have all security considerations been addressed?
- Does the architecture consider all existing constraints?
- Are all necessary, and only the necessary, data structures defined?
- Will the proposed architecture satisfy all specified quality attributes and performance goals?

Standards and Traceability

- Have all architecture standards been followed?
- Can all parts of the architecture be traced back to requirements?
- Are standard, not proprietary, components being used wherever possible?

Logic

- Is there any missing or incomplete logic?
- Are all possible states or cases considered?

Interfaces

- Are all interfaces clear and well-defined?
- Is the minimum data passed at each interface?
- Is minimum global system data added or impacted?

Software Testing:

During my master program, I have studied following course that directly benefit in my overall knowledge and experience in terms of advance practices of software engineering processes with focus on Software Testing and its various applications.

- Software Quality Engineering (Course details mentioned above)

I worked as Software Quality Engineer and worked for the Software Product Quality from May 2006 to date. I worked as different positions and acted as Primary point of contact about the overall quality of the Software Product.

I have been extensively involved in executing Software Testing by conducting investigation and provided system stakeholders with information about the quality of the product under test. I have worked to provide independent view of the software to business and make them able to understand the risk of software implementation.

In Software Testing I have worked to evaluate one or more properties of the system to indicate that

1. System meets the requirements and its design; and to ensure that development is according to requirement.
2. System responds to all kind of acceptable inputs appropriately.
3. System and its functions perform within appropriate time.

4. System is sufficiently usable by the users of the system.
5. System can be installed and run in Intended environments.
6. System produce results that stakeholder's desire.

Test Planning:

In order to achieve above objectives, I have also worked to develop comprehensive test plans by incorporating following items.

1. Scope,
2. Target Test Items
3. Outlines of Planned Tests
4. Test Approaches
5. Entry and Exit Criteria
6. Deliverables
7. Test Environment setup
8. Task responsibility, technical and non-technical staffing and their training needs.
9. Identify risks, dependencies, assumptions and constraints

Test Case Creation, Execution and Management:

In order to execute testing, I have been extensively involved in designing and reviewing of test cases according to internally developed standards keeping following key points in mind.

- Testing Pre and Post condition of use cases.
- Testing the normal and alternate flow of the use cases
- Testing business rules
- Negative testing for all use cases.
- Identifying business patterns specific to the feature and design test cases to exercise them.
- Design related test cases, which check integration of the feature with other Web/Desktop Applications features.

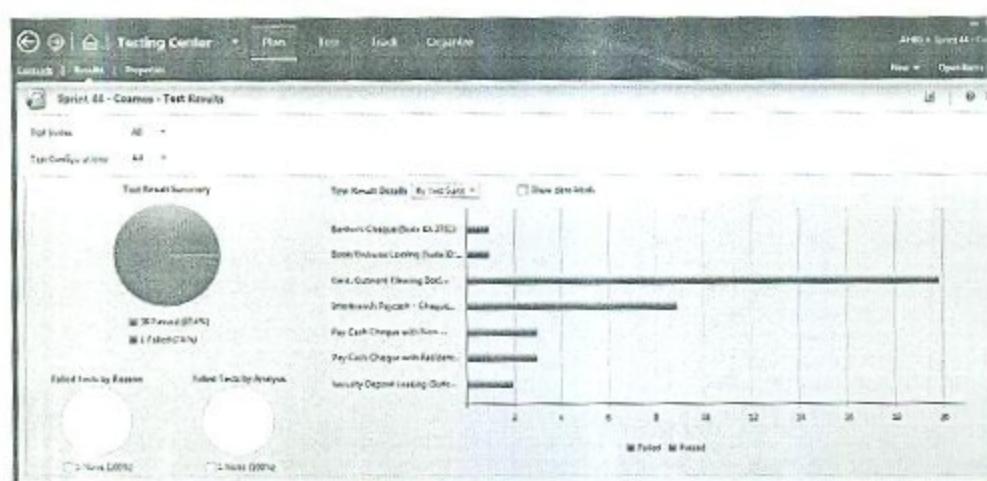
Example 4-2: Test Management system in Bank Al Habib Ltd (IT Innovation Dept):

I have also extensively work with commercially available test case management system like Mercury Quality Centre and Microsoft Test Manager and have gained hands on experience and knowledge about their administration, management and reporting's.

Following is the sample of MTM used in Bank Al habib Ltd (IT Innovation) that is integrated with Microsoft Team Foundation Server.

The screenshot shows the Microsoft Test Manager interface. The main window displays a list of test cases grouped under 'Test Plan - Core, Outward Clearing Bank (Build ID: 2754)'. The columns for the test cases are: ID, Title, Priority, Conf., Tested, and Area Path. The test cases listed are:

ID	Title	Priority	Conf.	Tested	Area Path
1	TC01 Add Setup Of Main Branch	2	1	Siddharth Chaudhary	Al-BD/Core Banking1
2	TC02 Add Setup Of Sub Branch	2	1	Siddharth Chaudhary	Al-BD/Core Banking1
3	TC03 Update Information File	2	1	Siddharth Chaudhary	Al-BD/Core Banking1
4	TC04 Set Up Hub Branch	2	1	Siddharth Chaudhary	Al-BD/Core Banking2
5	TC05 Set Up - Reverse Hub Branch	2	1	Siddharth Chaudhary	Al-BD/Core Banking2
6	TC06 Set Up - Inquire Hub Branch	2	1	Siddharth Chaudhary	Al-BD/Core Banking2
7	TC07 Core Outward Clearing - Set Up Department	2	1	Siddharth Chaudhary	Al-BD/Core Banking2
8	TC08 Add New User Account	2	1	Siddharth Chaudhary	Al-BD/Core Banking2
9	TC09 Setup Modify Map Branch	2	1	Siddharth Chaudhary	Al-BD/Core Banking2
10	TC10 Setup Reverse Map Branch	2	1	Siddharth Chaudhary	Al-BD/Core Banking2
11	TC11 Setup - Inquire Map Branch	2	1	Siddharth Chaudhary	Al-BD/Core Banking2
12	TC12 Gen Outward Map Branch Reg & Release Normal	2	1	Siddharth Chaudhary	Al-BD/Core Banking2
13	TC13 Update File - Settle Investment Interest	2	1	Siddharth Chaudhary	Al-BD/Core Banking2
14	TC14 Outward Clearing Normal Logon	2	1	Siddharth Chaudhary	Al-BD/Core Banking2



Test Approaches:

In order to execute system testing, I have adapted following test approaches determining the test cases and type of testing required to execute them.

I have been extensively involved in considering test approaches that are actually testing techniques outlining how each technique will be implemented by manually and/or automated wherever required during testing cycle. Following are the major testing approaches that I have followed while executing system testing.

- a) Data and Database Integrity Testing
- b) Functional Testing
- c) Business Cycle Testing
- d) User Interface Testing
- e) Performance Profiling
- f) System Integration Testing
- g) System User Acceptance Testing

Data Integrity Testing and Data Migration Testing:

I have been involved in Data Base and Data Integrity testing to verify that data is stored by the system in way that it is not being compromised by updating or during restoration or retrieval process. I use to follow this type of testing to identify any flaw in design that may cause data corruption, un-authorize access of data, lack of data integrity across multiple tables, un adequate performance of transaction. The best way to perform data integrity testing is that databases, data files, and the related processes should be tested as a subsystem within the application. In this type of testing, test data generation tools are used to generate test data in order to be able to check the validity of applications.

Example 4-3: Database Migration testing:

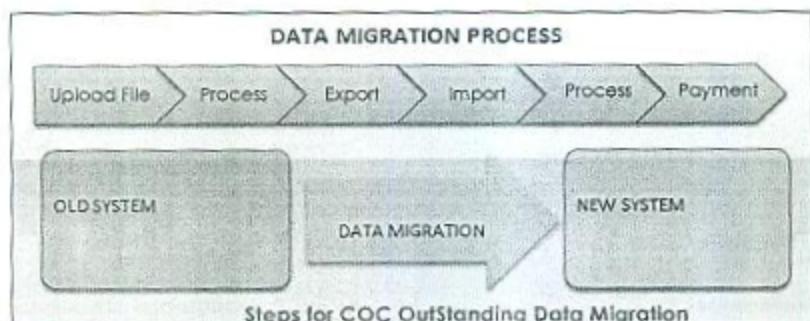
While implementing Home remittance system (third party) in Bank Al habib, it was big challenge to migrate the outstanding transaction of Cash over the counter (COC) from old system to new system.

Cash Over the counter are transactions that are belong to those walk-in customers who must not have account in bank. They visit to branch and validate the reference number they got from remitter with branch representative and take the cash. These visitors visit branch at any time.

While moving from old Remittance System to new System, there were numerous

outstanding COC transactions pending at old system and they were required to migrate to new system because business team could only use one system at a time.

In order to test the complete migration, I adapted following strategy described below



Previously all the transactions were in different state in old system like file uploading, bulk processed, AML, Verification, Authorization state. After careful analysis, I found that all transactions must be moved to a common state for smooth re-conciliation.

For this purpose, I processed all the transactions to common outstanding state and then I reconciled with current COC outstanding status. After successful reconciliation that was very challenging to verify for all exchange companies and thousand of transactions.

I exported the transaction using database queries and import them in new system. I again reconcile all the transactions with old system. After this, I executed testing to process them in new system to verify that transactions from old system are also able to process in new system and then finally I executed payment test cases in new system to verify that transactions are processed successfully.

Functional Testing:

I have been involved in Functional testing in almost all the organization while working as Software Quality Assurance Professional. During functional testing, I focused on software requirements that can directly trace to actual business requirement and relevant business rules. Here the courses of Software Requirement Engineering and Software Quality Engineering mentioned above have helped me to apply relevant knowledge.

The goal of Functional Testing is to verify that system accept proper data, it is processed and retrieved smoothly, and verify that all the business rules are properly implemented. This type of testing is normally called black box techniques, in which it is verified that the application and its internal processes are working as per the requirement, by interacting with application user interface (GUI) and analyzed the results. In this type of testing functional testing tools may be used to automate functional test cases.

I have also been executing Regression testing to verify the impacted areas to check, if they are not broken again due to fixes of bugs found during functional testing.

I have been leading Product Functional Quality Assurance in following organization

1. DeviceAnywhere Enterprise Suite in Mobile Complete Inc
2. Information Display System in Wavetec Pvt Ltd
3. ERP system of Traffic Program in LN Technologies

Please refer to
Experience
Letters from
companies:

Mobile
Complete,
Wavetec Pvt Ltd
LN
Technologies

Example 4-4, Example of Automated Functional Testing:

While working in Mobile Complete Inc, I used Quality Test Professional for Functional testing of Web based system.

In order to reduced repeated testing effort and to ensure stability of existing functionalities, I implemented Quality First Test (QFT) in Bank al habib Ltd to automated Functional Testing and Regression testing of Core Banking system (Refer to Element-5)

Following is the sample snap shot of Functional Test Scripts Developed in QFT.



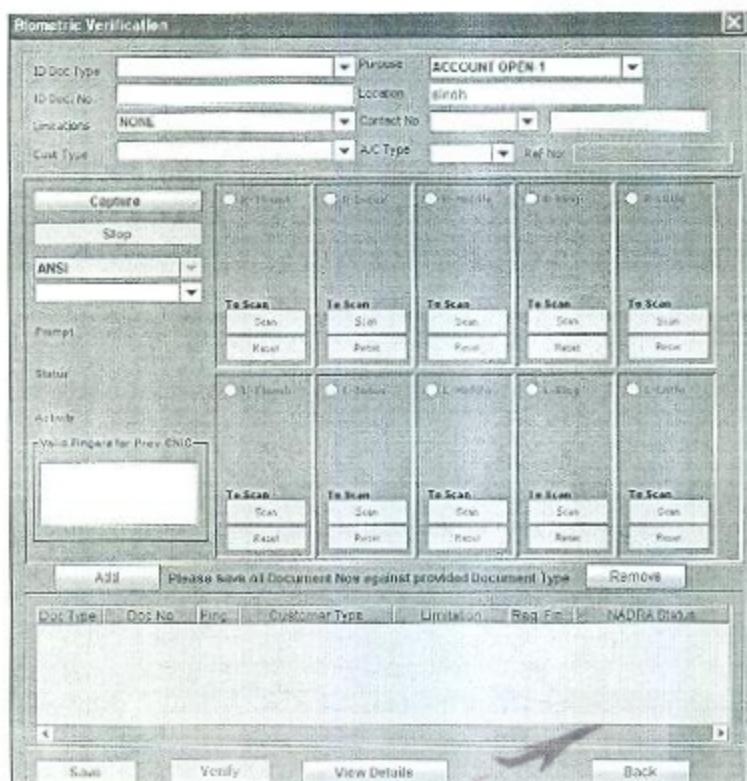
Example 4-5. Functional Testing of Biometric Feature in Account Opening Process:

Bio Metric Feature was introduced in AHBS (Core Banking System) at Account opening level due to regularity requirement enforced by State Bank of Pakistan. This system was comprises of consuming NADRA Service that was in house developed to capture Bio Metric of Pakistani Nationals having valid Computerized National Identification Card (CNIC). The systems take bio metric of customer and save in AHBS database and process the account using biometric information captured. The Account opening module was already in system, but due to newly introduced bio Metric feature the entire code of this module was re-engineered to add this Mechanism. This feature was highly complex in nature and required thorough functional and integration testing with subsequent regression testing due to bug fixes.

Peer
Recognition
Certificate by
Development
Team Lead
Signed by GM-
IT Innovation

I was the owner of Functional, Regression and Integration testing of this entire feature. I used my analytical skills to understand the requirement, developed test strategy, designed complex test cases and Identified and analyze the defects found in system. There were various complex checks in system for validations that I tested and systematically identified various issues having different levels of severity (Normal, Major, Critical, Showstoppers). The objective of entire test execution was to highlight issues that required me to apply critical thinking and strong analytical skills with appropriate testing approach and related test strategies, so that high priority issues highlight and resolved early in development life cycle.

Following is the Sample screen of Bio Metric Interface in AHBS (Core Banking)



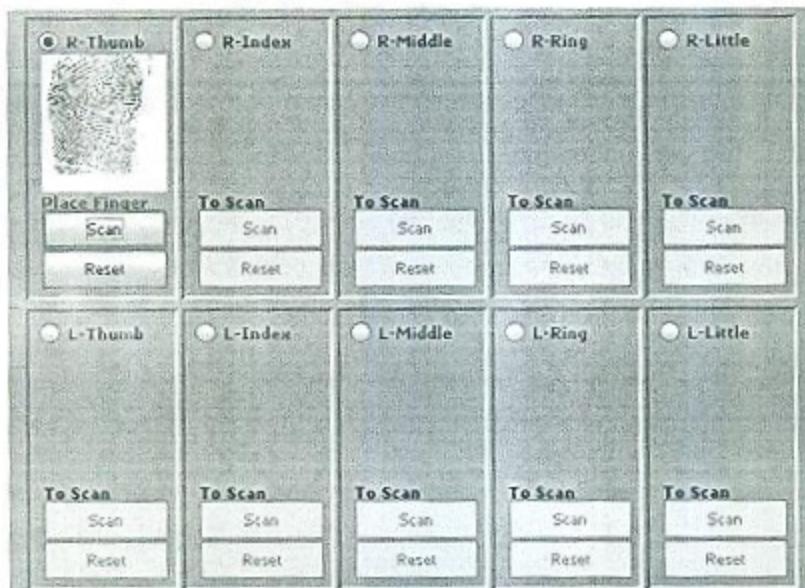
There were two type of biometric devices used at bank branch. Users were able to capture fingers impressions using these bio devices.

- a) OneTouch-Digital Persona
- b) SecuGen

User select device and then press Capture button.



Example of thumb impression taken



Business Test Cycle:

Business testing cycle are executed when end to end testing of real life business scenario has to verify in order to ensure that complete business scenario is intact and working. In this, testing, specific periods are identified and specific transactions and activities are executed during a year. This includes all daily, weekly, and monthly cycles, and events that are date-sensitive.

I have been testing business test cycle in Core Banking system to ensure that each business rule is properly applied.

Example 4-6: Example of Business Test Cycle:

For example in Core Banking System in Bank Al Habib, the Monthly Profit in saving account products i.e. Monthly Saver Account is accrued at the EOD (End of Day) process in last working day of every month. In order to ensure that business cycle is working, I execute End of Day Process in Branch dummy database until month end and verify that profit is accrued correctly.

Monthly Saver Product						
Online Customer Profile - Statement of Account - Display						Transactions
Customer Information			From 01/02/2016 To 02/03/2016			6
Branch	S.I.T. E-1004	From	01/02/2016	To	02/03/2016	
A/C No.	XXXXXXXXXXXXX-8	A/C Type	ALHABB MONTH SAVER - D078			
AC Name	XXXXXX(X)(X)(X)(X)(X)(X)(X)(X)	Currency	PAKISTANI RUPEES - 586			
Post Date	Value Date	Doc Ref No.	Transaction Description	Credit	Debit	Balance
24/02/2016			Opening Balance			0.00
24/02/2016	5563848		GASH DEPOSIT	240,000.00		240,000.00
27/02/2016			CASH BOOK CH-3	140.00		239,859.00
28/02/2016			PRO/TMARS/001		327.43	239,531.57
29/02/2016			WATAXSURCHARGE	17.75		240,019.73
29/02/2016	6109533405		BANK CHARGES DR	200.00		239,819.73
29/02/2016	6109533405		FEDERAL TAX	32.00		239,787.73
29/02/2016			Closing Balance			239,767.73
		Total		309.75	240,127.48	

User interface Testing:

I have been developing and executing testing strategy for user interface in almost all the organization. In user Interface (UI) testing, I verified that when user interact with the software, the interface provides the user with appropriate access and navigation through the desired functions. In addition, I ensured that object within the UI functions are working as expected and conform to UI standards.

Performance Profiling:

Performance profiling is a type of performance testing in which time sensitive requirements are measured like response times, transaction rates etc. I have been involved in performance profile testing to verify that performance requirements have been achieved. In performance, profiling it is investigated and analyzed how the system behaves during execution. Based on the data used and results, it is analyze how the performance of the system can be further optimized.

This testing strategy is followed by performance testing tool that is normally not possible to execute manually and create related scenario. For such type of testing, I have used various performance testing tools like rational performance tester and JMeter.

System Integration Testing:

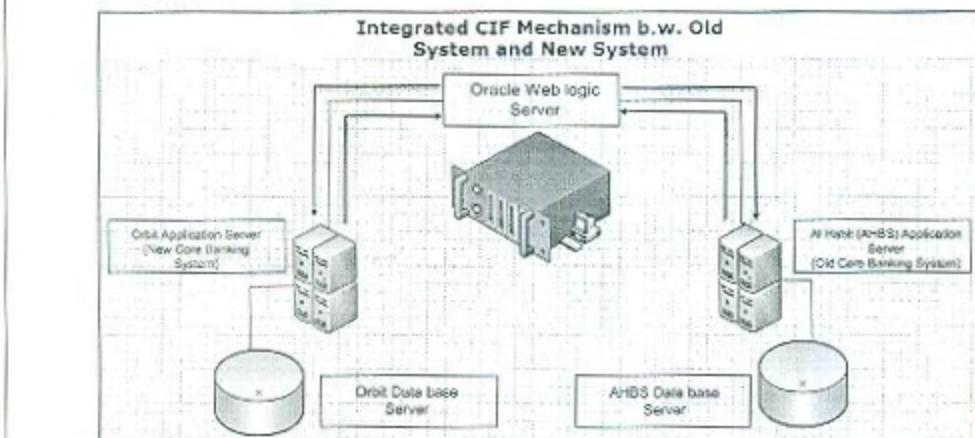
Integration testing is the software testing phase in which individual software modules are combined and tested as a group. I have been extensively involved in system integration testing in which I have configured different modules of the system in separate environment and grouped them in larger aggregates to ensure that all modules work together to achieve the system functionalities as specified in design.

Example 4-7. Integration Testing of CIF Mechanism between old Core Banking system and New Banking System

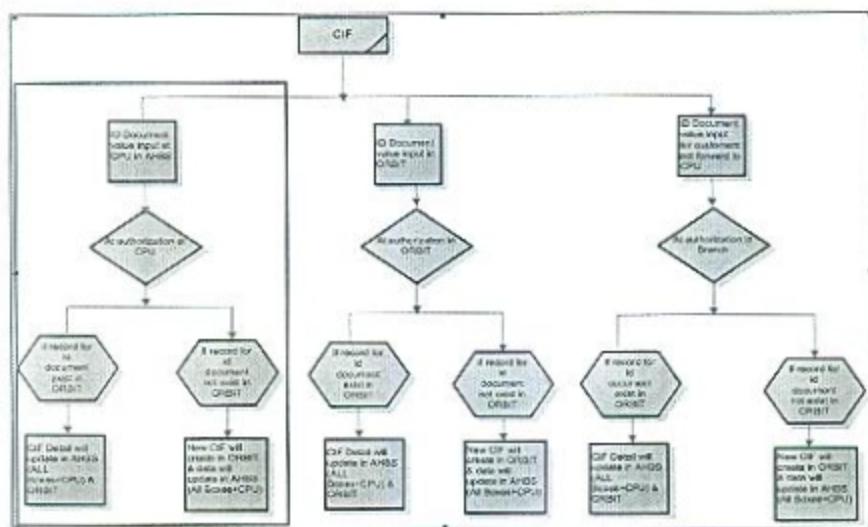
While working in Bank Al Habib, I performed integration testing of CIF mechanism. CIF stands for Customer Information folio and it is consider as heart of ORBIT Core Banking System (New System). This concept is not in old AHBS Core banking system where customer data is maintains on branch level and CPU level with the combination of branch code and the customer number in order to uniquely identify a customer. In New Core Banking system, CIF maintained the entire customer information in centralized repository. In order to work with both core banking as Coexistence, there was an essential need to introduce the concept of CIF in Old Core Banking System i.e. AHBS, so that the centralized CIFs could be used in Old Core Banking as well as New Core banking simultaneously allowing both system to work in co-existence.

Peer
Recognition
Certificate by
Development
team lead
signed by GM-
IT-Innovation

The below diagram shows the architecture of Integrated CIF Mechanism



Following is the process flow of CIF Generation or CIF Update due to Account Creation and Account related update at AHBS end. The Scope of Integrated testing I performed is highlighted in below flow diagram.



Scope of Integration Testing.

The entire CIF project was highly complex in nature that required thorough functional testing and integration testing with hundreds of combination of data input at AHBS end.

In Summary, I had verified following key points in Integration Testing.

1. While opening new account in AHBS inputting ID Document value and process the record at CPU I verified that,
 - a. If record for ID document exist in Orbit DB then after authorization of new account in AHBS (CPU), CIF detail will be updated in AHBS (Branch DB and CPU DB) and Orbit DB as well.
 - b. If record for ID document not exists in Orbit DB then after authorization of new account in AHBS (CPU), New CIF will be created in Orbit DB and AHBS (Branch DB and CPU DB).
2. This testing was required to verify at the time of new account opening and maintenance of old accounts.
3. Account Categories as Natural Person involved (Singly, Jointly) and Entities involved (Proprietor, Partner, Company, Trust, Club and Association) with various combinations of data input and selections.
4. I verified all possible combination CIF input data from AHBS interface to Orbit DB as well verified that they are accurately displaying on Orbit UI interface.

Challenges:

The database table structure, terminologies were different in both core banking system for which there were various mappings available at both database levels. In this project I tested the mapping with in-depth understanding and analysis and identified various functional and technical issues and resolve them working with software developer.

For example, Singly and Jointly nature of Accounts in AHBS were considered as Natural Person category, While there was no concept of Natural person in AHBS.

In this testing, there was a lot of effort required to input data and verification of results in AHBS and Orbit interface for which I utilized the Quality First Test by making automated scripts and utilized the tool to generate test data.

System User Acceptance Testing

In this testing, I have been involved with end users to execute formal testing with respect to user needs, requirement and business process to ensure that system satisfies the acceptance criteria and enable the user to determine where system is acceptable or not.

In Bank Al habib – IT Innovation, the core banking system is already working at production environment. Whenever any change is system occur due to business requirement or regulatory bodies i.e. State Bank of Pakistan, I conduct beta testing of the new features or change in system, in which I worked with business users to verify business users requirement.

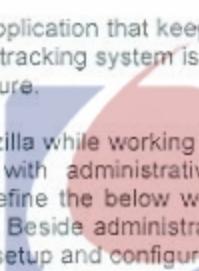
Other Specific Testing.

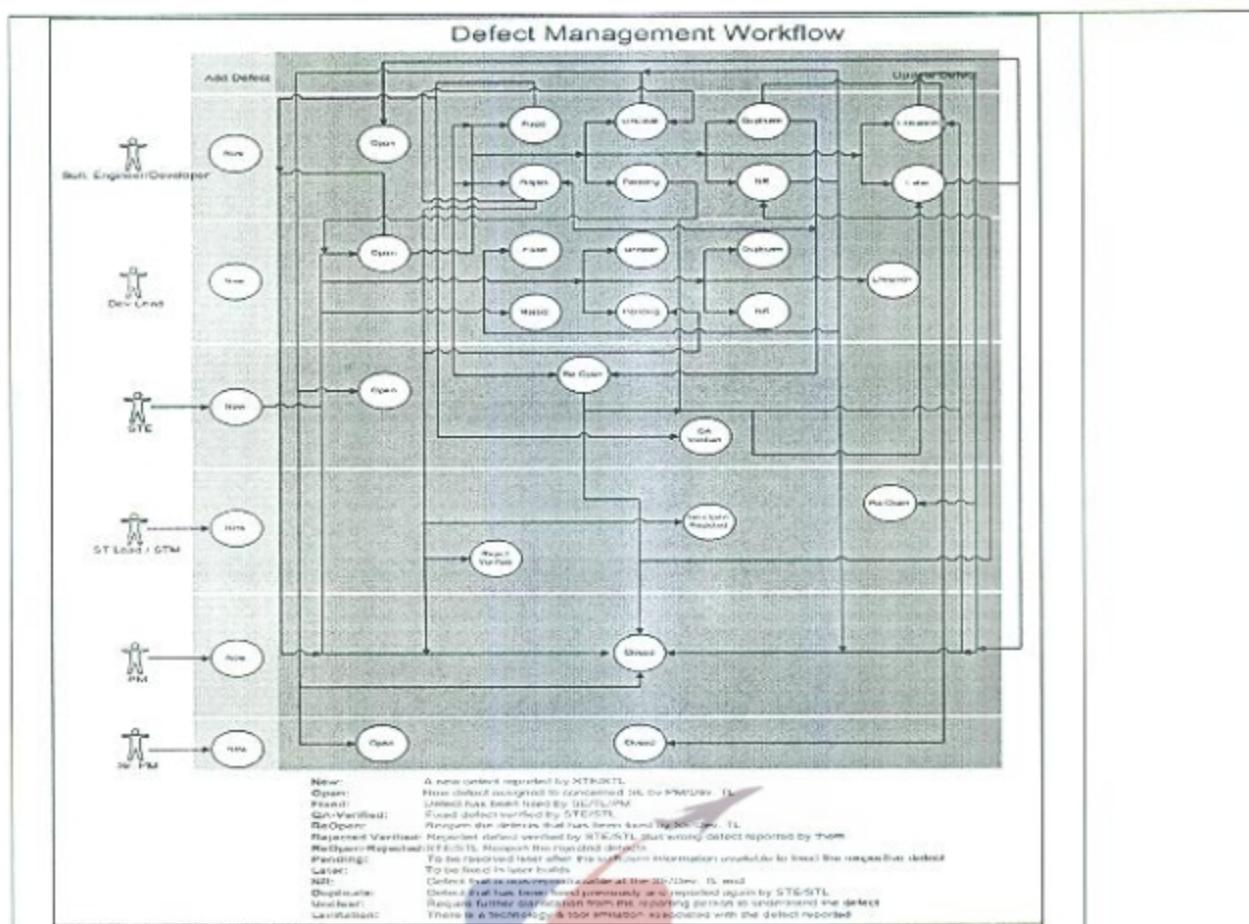
In order to confirm that web application is working fine in other Browsers, I have been executing browser certification testing. In order to achieve it, I create script and execute in every required browser to verify that application works as expected in different browsers. I have also been worked for other specialized testing like system configuration testing, operating system certification testing. I have also worked for other types of testing like sanity testing to execute final test round containing high level test case execution, Smoke Testing to execute test cases that may be impacted due to most recent change in software product.

Bug Tracking System

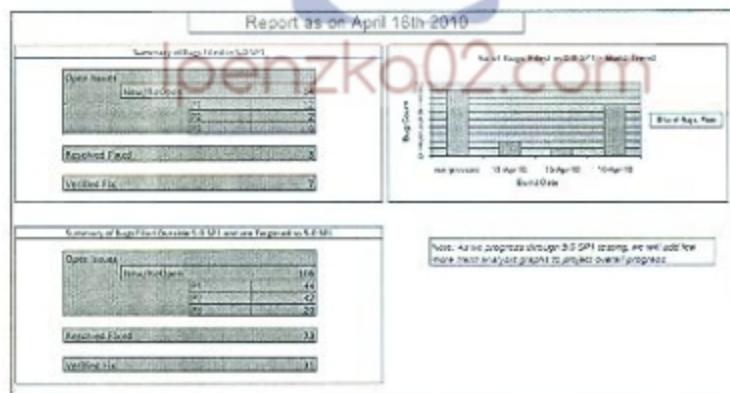
A bug tracking system is a software application that keeps track of reported software bugs in software development projects. A bug tracking system is usually a necessary component of a good software development infrastructure.

For bug reporting, I implemented Bugzilla while working in Mobile Complete Inc, that is open source software and it allow user with administrative rights to customize the Defect Management workflow. Therefore I define the below workflow in Bugzilla to be used while reporting bugs during testing process. Beside administrative and working knowledge, I have all the required skills and expertise to setup and configure Bugzilla for bug reporting system.

 lpenzka02.com



I have also worked to generate bug trend reports in excel sheets in order to escalate to management for decision making about the current state of product release. Following is the sample bug trend report I generated in Mobile Complete.



I have worked in various Defect Management Software's in different organization and have contributed to define bug cycles and relevant bug processes. For example Severity and Priority Matrix, Products Mappings, bug reporting standards, defect cycles etc.

Typically, bug-tracking systems are integrated with other software project management applications. Now a day's many large-scale application life cycle management systems have built-in bug tracking system e.g. Mercury Quality Centre and Microsoft Team Foundation Server. I have also gained extensive working knowledge of these systems while working with

various organization.

Sample of bug report generated from TFS (Team foundation Server)

Search results: Takmeel						
	Issue	Type	Description	Status	Assigned To	Tags
12	15100	Bug	Takmeel On Route Payment Only with Transaction recorded. Payment of Principal, Interest, Other Pre-encashment Miles, Transaction of Nominal account and No..	Verified	Nizar Mira Karim	
31407	51607	Bug	Takmeel On Coding. For every the Max with Age is not displaying in Front end.	Verified	Nizar Mira Karim	
32108	51201	Bug	Takmeel On Encashment. Takmeel is not displaying in Front end.	Verified	Nizar Mira Karim	
52234	51205	Bug	Interest Takmeel HC Nominally as Policy in Takmeel Insurance Inquiry	Verified	Nizar Mira Karim	
52235	51206	Bug	Takmeel After Second Renewal of the on pre-encashment mark is paid	Verified	Nizar Mira Karim	
52236	51207	Bug	Takmeel After Second Renewal of the on pre-encashment mark is paid	Verified	Nizar Mira Karim	
52237	51208	Bug	Interest Accrued on Old Fee is not added by the Policy of Previous Installment	Verified	Nizar Mira Karim	
52238	51209	Bug	Takmeel Discrepancy on SOD is Disclosed	Verified	Nizar Mira Karim	

Sample of bug reported in TFS (Team foundation Server)

Bug 51607*: Takmeel: After Pre-encashment todays Transaction of Takmeel account and Nominated Account dont display

Iteration: APR16V1Release 10/parent 27

STATUS

Assigned To: Nizar Mira Karim
State: Verified
Reason: QA has passed
Category: Core Banking
Issue Type: Bug
Feature: Account Closure
Reported by: Nizar Mira Karim
Reporting Date: 5/16/2016 2:30:59 PM

DETAILS

• Environment: 1 - Critical
• Severity: 1 - Critical
• Team: Core Banking 1
• Backlog Priority: 1
• Bug Category: Regression
• Resolution: Fixed
• Branch: Cine Live Banking

STEPS TO REPRODUCE: SYSTEM : TEST CASES : TASKS

1. Block Takmeel deal of 50000
2. Execute EOD
3. Pre-encash the deal

Actual Result:
Today's transactions of Takmeel account and nominated account didnt display transaction.

ACCEPTANCE CRITERIA: HISTORY : LINKS : ATTACHMENTS (1)

Transaction should be displayed.

Example 4-8, Example of Engineering problem and Solution by Implementing Bugzilla for Bug Tracking

I was working in Mobile Complete, Inc in 2006. The company was developing a product of mobile testing solution. At that time, the product road map was not clearly defined and there were very less clients who were using that product. After several month, product related changes rapidly start to appear and this was causing a lot of testing effort for software test engineers. Initially, during testing phase, all of the issues were being managed by individual software test engineers and gathers in Microsoft Excel and reported to management in emails.

At that time software engineering team were facing many issues in tracking of bugs on daily basis. As a team lead, I raised this concern to management to resolve this problem as it was causing various issues in capturing, recording, accountability, prioritization, resolving and reviewing of the reported issues. Ultimately, it caused increased cost of bug fixing tracked later.

After several discussions and meetings with management and considering the challenges of product road map, management decided to adapt a proper system that could solve this problem but management was reluctant to invest in such tools. I was given this assignment

to solve this problem with minimum cost. I went through various bug tracking software's and found that all the software were highly costly. Then I decided to research open source software's, and found bugzilla i.e. an open source bug tracking software. After configuration, setup and details exploration of this tool, I found it suitable for implementation and gave presentation to management. In that presentation I explained the actual problem being faced by engineering team while reporting the issues and how the software provide solution. I also explained how the system can be utilized by the organization considering long term benefit.

Following were the problems that software engineer team (comprised of software developers and software tester) was facing and their solutions:

1) Issue Capturing:

Software testers were not cleared who they should report the bugs and got worried that if the issue is not captured timely, it will not resolved.

Solution:

Bugzilla provided a central place to record and address bugs, queries, assignments, ideas, and more, hence it caused less chance to slip through the bugs. After implementation, every user of the bugzilla found it easy to log the bug without any self-memorization.

2) Issue History:

Software engineering team were unable to review what actually decided with the bug and time when it was decided because there was no any history maintained before.

Solution:

After implementation of Bugzilla, bugs were reported and maintained with proper history by bugzilla itself, the relevant stakeholders were able to update the comment with the final decision and engineering team were able to view the history of the bugs later on.

3) Accountability:

It was not possible to track the responsible person who will be resolving the bug. It was also difficult to determine the responsible person, the decision maker of resolution, when the resolution was recommended.

Solution:

After implementation of bugzilla, it was easy to track the reporter and the person whom the bug is assigned for resolution. This way it helped encouraged forward progress and reduced communication.

lpenzka02.com

4) Prioritization:

It was difficult to track the priority and severity of the bug that how much it is important and when it should be resolved. It was then impossible to track when bug count exceeded 100 and no person was able to know which problem need attention first.

Solution:

It provided a central place to prioritize the bugs against any new feature or requirement to make sure things are being resolved in the right order and on time.

5) Resolving:

It was difficult to know what solution was decided with the bug and how it was resolved

Solution:

Bugzilla had helped developer and other stakeholders to focus on forward progress, by providing history of the bugs to view the decision taken if any. That way it helped engineering team to save time and reduced chances to do wrong fixation.

6) Reviewing:

It was difficult to review the bug to know how it was identified with reproducible steps and how it was fixed.

Solution:

This is the layer of quality assurance and it enables how the software bug review process was followed to ensure that the outcome matches the expectation. Using this system, it was easy to identify the steps to reproduce the bug along with actual and expected outcomes.

Beside solution to above problems, the tool has also opened further gates of opportunity to use other features provided by the system.

For Users	For Administrators
✓ Advanced Search Capabilities	✓ Save and Share Searches
✓ Email Notifications Controlled By User Preferences	✓ Excellent Security
✓ Bug Lists in Multiple Formats (Atom, iCal, etc.)	✓ Extension Mechanism for Highly Customizable Installations
✓ Scheduled Reports (Daily, Weekly, Hourly etc.) by Email	✓ Custom Fields
✓ Reports and Charts	✓ Custom Workflow
✓ Automatic Duplicate Bug Detection	✓ Full Unicode Support
✓ File/Modify Bugs By Email	✓ Localization
✓ Time Tracking	✓ mod_perl Support for Excellent Performance
✓ Request System	✓ Webservices (XML-RPC) Interface
✓ Private Attachments and Comments	✓ Control Bug Visibility/Editing with Groups
✓ Automatic Username Completion or Drop-Down User Lists	✓ Impersonate Users
✓ Patch Viewer	✓ Multiple Authentication Methods
✓ "Watch" Other Users	✓ Support for Multiple Database Engines
✓ Move Bugs Between Installs	✓ Sanity Check

Benefit realization by the organization:

- Communication between all stake holders was drastically improved.
- Product Quality was increased as all the issues once reported in system were being addressed by proper prioritization and escalations.
- Customer satisfaction was highly improved, because all the issues faced by the customer were also logged in bugzilla and those issues were given with high priority to address them timely.
- It created greater accountability among all stakeholders. Because it was very easy to find out the owner of the bug. Either the issue is belong to Business Analyst, Developer, Software Architect, Software Tester etc
- Due to above key points, overall productivity of all stake holders increased.

Software Configuration Management:

I started my career in Kalsoft (Pvt) Ltd as Software Configuration Engineer in May 2005.

I was responsible to provide support to organization for the activities related to software process engineering and provides guidance for SPEG (Software Engineering Process Group). In this organization, I learned to plan and undertook quality assurance activities and was responsible for providing related support to different projects in compliance with software quality standards.

At that time, I also worked to develop processes and procedures for further improvement in configuration management process area from the CMMI level II guidance and trained organization and project stake holders with this process.

Refer to Kalsoft (Pvt) Experience Letter.

Following are the best practices of Configuration Management, I was responsible for

- Configuration identification
- Configuration control
- Configuration status accounting
- Configuration Audit.

Other than this, I was also responsible for preparing shipment package as per the requirement for a particular project, perform backups of configuration management system as per schedule in Tape Drives, I was also responsible to test the recovery and restore the system in test environment as per schedule and when disaster appear.

I was also administrating and maintaining the configuration management servers and was responsible to evaluate the current environment and propose recommended CM solutions.

Configuration Management Tools and technology:

The entire Configuration management system was implemented using Microsoft Visual Source Safe 2005, in which all the project and organization wide artefacts and their versioning were maintained.

Beside above tool, I also have all the required skills, expertise and knowledge about the implementation of configuration management tools like

- a) CVS-Concurrent Version system
- b) Rational Clear Case

Most of the application life cycle management systems have built-in software configuration management system like Microsoft Team foundation server, for which I have working knowledge.

