Knowledge Assessment SELF REVIEW (FORM KA02)

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of Applicant:** | **SASHI KANTH** | **Membership number  or date of birth:** | **11th AUGUST 1987** |

**Section One**

# Important Instructions and Guidance

**Carefully read the following instructions and guidance. They are designed to assist you in providing in providing a portfolio of evidence that best demonstrates the comprehension and application of your engineering knowledge to Washington Accord equivalence.**

**Section One – Instructions and Guidance**

* Familiarise yourself with the definition of ‘*complex engineering problems’* (Appendix One) as you are required to demonstrate you can apply your engineering knowledge to solve complex engineering problems.
* Identify the ‘*engineering discipline and field’* (Appendix Two) you will provide evidence of your comprehension and application of engineering knowledge in.
* The knowledge assessment is based on Washington Accord knowledge profile. This form is designed to capture information to assist the evaluation of your evidence

**Section Two – Knowledge Profile**

* As you do not have a formal engineering qualification that formally benchmarks to a Washington Accord accredited degree, it is essential that you demonstrate that you have acquired an equivalent level of knowledge.
* The Context and performance indicators provide guidance on the evidence to be provided
* Consider each element of the knowledge profile, including the context statements and performance indicators. Summarise key aspects of your knowledge under each element and how this has been developed through academic study, on-job learning and/or continuing professional development. It is important you use the performance indicators and complexity definitions to enable you to describe your knowledge and how it has been developed.
* When describing how your educational program contributed to your development, focus on the more advanced pieces of work you did, the knowledge you needed in order to perform that work, and the abilities you needed in order to apply your knowledge in an engineering context.
* The word document is formatted to allow you expand a text box if required.
* Write your material in the first-person using ‘I’ or ‘me’ instead of ‘we’ or ‘us’. This makes it easy for the assessors to see what your personal contribution was.

**Section Three – Evidence of Application of Knowledge**

* Describe 3-4 engineering projects or activities (Work/Study Episodes) that you have been involved with, which demonstrate your ability to apply your engineering knowledge to solve complex engineering problems. Think of activities where you have had to apply a high level of engineering knowledge – such as some analysis that you have done, work you have done in scoping a problem and then developing a solution or design. What engineering models did you use? What assumptions were made in the development of the model and how did you test the model was relevant in the way you used it?
* For engineers with limited practical experience post-graduation, project work undertaken during your study is likely to be one of the best ways of illustrating the application of your knowledge. As well as projects conducted within university or college, you may be able to draw on any industry experience required as part of the educational program.
* You are required to include actual samples of your work – calculations, analyses or reports that you have personally undertaken - to substantiate your work/study episodes.
* Write your material in the first-person using ‘I’ or ‘me’ instead of ‘we’ or ‘us. This makes it easy for the assessors to see what your personal contribution was.
* The word document is formatted to allow you expand a text box if required.

**Section Four – Supplementary Evidence**

* You are required to submit a certified copy of your academic transcript(s) (formal record of papers taken and grades received) if you have not submitted to IPENZ already.
* Summarise your work history but include a representative sample of specific engineering projects or activities that evidence the development or application of the knowledge profile.
* Rather than listing all your CPD activities, provide details of those activities that have extended your professional engineering knowledge in your discipline and field and have assisted you to develop the knowledge profile of a professional engineer. A summary of all relevant activities – including those going beyond the most recent 6 years - will assist knowledge assessors in assessing your engineering knowledge. Assessors will be looking for how any gap between your qualification and a Washington Accord qualification has been bridged by your CPD.
* The word document is formatted to allow you expand a text box if required.

**Section Five – Payment**

* The fee for a knowledge assessment is NZ$1,351.25 GST incl. Please complete your credit card details.
* Send all documentation to address advised

**What happens next?**

The knowledge assessor will review your portfolio of evidence to determine the need for further challenge tests. This will involve an interactive assessment, that you will need to make yourself available for, either via tele or video conference and may also involve a series of challenge tests that may include one or a combination of:

* an oral and/or written examination
* a work simulation
* a case study

Your knowledge assessor will be in touch with you to discuss the next steps.

Section Two – Knowledge Profile

|  |  |
| --- | --- |
| Element One  **A systematic, theory-based understanding of the natural sciences applicable to your discipline (e.g. calculus-based physics)** | |
| **Context**  All engineering fields are rooted in one or more of the natural sciences. In a broad context, natural science is separated into physical and biological sciences. Physical sciences include chemistry, calculus-based physics, astronomy, geology, geomorphology, and hydrology. Biological sciences involve living systems and include biology, physiology, microbiology, and ecology.  Washington Accord graduates are expected to be able to apply this knowledge of the natural sciences to solve complex engineering problems in their discipline. | |
| Performance Indicators   * Fundamental quantitative knowledge underpinning nature and its phenomena. * Knowledge of the physical world including physics, chemistry and other areas of physical or biological science relevant to your discipline * Knowledge of key concepts of the scientific method and other inquiry and problem-solving processes; * Application of knowledge from one or more of the natural sciences to the solution of complex engineering problems relevant to your discipline. | |
| Summarise your knowledge of the natural sciences relevant to your discipline 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.  In this section, I will go through the knowledge of physical sciences, that I learned through my Engineering discipline and I will explain, how I was able to develop my knowledge and skills to understand the application and use of these sciences. The sciences that I studied were the building blocks for what came next in my career. I will co-relate the courses with the applications of sciences as well as the systems, which are the basic of the current systems in my field of Engineering.  I studied 4 courses of physics and 4 courses of mathematics in the first year of my Engineering study period in the University. In mathematics, I studied Algebra, Geometry, Matrices, Trigonometry, Logarithms, Exponentials, Integral & Differential calculus and Differential equations. This helped me solve, analyse and understand the problems in my career in software application building, creating statistics of the software applications, software testing (Boundary value analysis) etc..  While developing my mathematical knowledge and skills, I was also learning Physics. I studied Thermodynamics, Engineering Mechanics, Mechanics of solids, Fluid Mechanics, Dynamics including laws of motion in my Engineering courses.  By studying physical sciences, it helped me in implementing following tasks in my career. Some of the tasks co-relating physical sciences are -   * Comparison of the performance of various algorithms and the complexity of various problems using O notation. O notation relies on the idea of the limits of ratios of functions as a variable tends to infinity. * **Numerical analysis for scientific software.** * **Modelling software for engineering applications.** * **Data analysis and prediction of business applications.** * **I have written programs that use “Intermediate Value Theorem” to find the roots of a polynomial, as part of a larger program.** * **I have written programs to draw functions of graphs that used calculus concepts to determine which values of x to plot points.**   **Examples of Application of Natural Sciences:**    **Example 1:**  **Situation –**  Build automation solution for 50 different partners which uses credit card application of Barclaycard US for Credit cards domain in Barclays. The solution needs to be build using Visual Basic (VBS) programming language and HP Quick Test Professional automation tool (now known as HP Unified function Tester tool).  **Problem –**  Assess and arrange highly configured system architecture that can handle the execution of all flows at a time without any latency and performance issues.  **Description –**  Barclaycard US is the 2nd largest credit card provider in United States providing cards of more than 50 partners which includes Apple, William Sonama, Bank of Hawaii, Travelocity etc.  Barclaycard US has also started providing bank cards for individual customers in US. US Cards and Marketing vertical handles all the functional & automation testing of customer facing Acquisition websites (aka Apply) and customer servicing websites (aka EMWE) in Agile model. Barclaycard US internet websites are developed as per requirement of partners to cater the customer requirements for online access. Automation testing of this project involves end to end testing of enhancements involving integration with multiple applications viz., TS2 Mainframe Backend, Oracle Database, and MQ Series.  There were 26 flows that need to be automated for 50 partners, which in turn sum up to 1300 flows (26\*50).  In order to build automation testing solution, accommodating such a huge number of flows and applications in very short duration, I needed robust system/computer architecture because the tools running on machines directly accesses the memory of the system and I needed such system to support and handle the memory utilization by the automation tools to execute all the flows.  After building the automation solution, I started with a 32 bit operating system, 4 GB RAM (Random Access Memory), 20 GB HDD virtual machine to test the automation flows. After rigorous execution of flows, I observed that the CPU usage of the machine was increasing drastically and the performance of the applications running on the system was very bad. So, I decided to go for a bit higher configuration machine and finally reached a stage where I was able to execute all 1300 flows in a span of 4 hours.  Steps for Resolution :-  Following configurations were used finally, in order to achieve the stability between systems and tools.   * Used x64 bit windows operating system, hosted on VMWare (Virtual machine) * Used 2TB HDD hard disk space * Used 16 GB RAM to speed up the response times * Used latest versions of all the automation tools * Used latest drivers to support the issues that were occurring due to technology upgrade     In this way I used natural science such as interaction, concurrency and causality, synchrony and asynchrony, compositional modelling and reasoning qualitative versus quantitative reasoning, operational methodologies, continuous versus discrete, etc. as models from physics.  **Example 2:**  **Situation –**  Build an automation solution for accessibility testing of HSBC credit card applications/websites using JAWS tool to meet WCAG guidelines and be compliant with the American Disability Act laws when I was employed in Capgemini.  **Problem –**  Synchronization between automation tool, application and JAWS tool was inconsistent.  **Description –**   * *HSBC credit card application* – HSBC credit cards is broadly divided into 2 categories CSeCare and RSeCare.   CSeCare internet websites was developed for the Card Services Business Unit of a Leading North American Bank to cater the customer requirements for online access of their credit card accounts.  RSeCare internet websites, on the other hand, was developed for the retail services business unit to cater the customer requirements for online access of their retail card accounts provided by Over 200 Leading Retail Stores in America like Best Buy, GAP, CVS, Menards etc.   * *American Disability Act (ADA)* - In the year 2009, US government amended the Americans with Disabilities Act of 1990 (ADA) and other disability non-discrimination laws at the Federal level of the United States.   According to this law, all websites owned by the banks or organizations/firms in US should meet and be compliant with the WCAG guidelines, in order to prohibit discrimination against individuals with disabilities in all areas of public life, including jobs, schools, transportation, and all public and private places that are open to the general public.  Since all the HSBC north America websites, applications and infrastructure was maintained by Capgemini. We got a project to test all the HSBC credit card websites to be compliant with WCAG guidelines. The websites were made compliant and released into production.  After successful implementation I started working on preparing regression suites to validate that the changes that are being made in subsequent releases are not breaking any of the accessibility amendments that were made in the websites.  I used HP Quick test professional automation tool (now known as HP Unified functional Tester) to create the automation suites for accessibility testing.  I started with integration of automation tool, JAWS tool (tool to assess accessibility of websites) and HSBC credit card websites. During development of automation suites, I observed that synchronization between these tools were not consistent.  For Example, automation tool triggers an event (keystroke) without waiting for JAWS tool to complete the actions that it is required to perform and due to this the entire performance of the system was also poor, resulting in a faulty automation package.  Hence I started working on synchronizing these tools by adding wait times and optimizing the hardware of the systems to achieve expected outcome and better results. Finally, I completed the development of the suite with a proper hardware and synchrony between these systems.  JAWS overview (look and feel) –    List of key strokes used for accessbility testing using JAWS tool –        In this way I used natural science such as interaction, concurrency, synchrony and asynchrony, compositional modelling, operational methodologies, continuous versus discrete, etc. as models from physics. | **Provide annotations to your supplementary evidence (document and page number)** |

|  |  |
| --- | --- |
| Element Two  **Conceptually-based mathematics, numerical analysis, statistics and formal aspects of computer and information science to support analysis and modelling applicable to your discipline** | |
| **Context**  Branches of mathematics applied in engineering include arithmetic, algebra, geometry, trigonometry, calculus, differential equations, numerical analysis, optimization, probability and statistics, simulation, and matrix theory. Engineers apply mathematics in a wide variety of functions typically carried out in engineering organisations such as planning, design, manufacturing, construction, operations, finance, budgeting, and accounting.  Washington Accord graduates are expected to be able to apply this mathematical knowledge to solve complex engineering problems in their discipline. | |
| Performance Indicators   * Knowledge of mathematics, statistics and numerical methods that supports the development or application of models that replicate ‘real world’ behaviours * An understanding of the assumptions behind theoretical models and their impacts in the development and use of those models * Ability to organise and analyse a data set to determine its statistical variability; * Knowledge of trigonometry, probability and statistics, differential and integral calculus, and multivariate calculus that supports the solving of complex engineering problems * Ability to apply differential equations to characterize time-dependent physical processes | |
| Summarise your mathematical knowledge relevant to your discipline 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.  Mathematics is one of the natural sciences connected to the programs written in building computer software’s. I have studied Engineering mathematics I, II, III, IV subjects related to mathematics and statistics throughout my Engineering discipline. Though the above titled subjects are specifically for mathematics, I have also used mathematics and statistics in all the subjects that I have studied in my Engineering discipline.  In addition to above mentioned subjects, I have done Industrial training and also studied subjects like Operations research & Production planning and control which are relevant to financial accounting.  In this section, I will elaborate on the knowledge of numerical methods and statistics that I learned through my Engineering discipline and I will explain, how I was able to develop my knowledge and skills to understand the application. Throughout my work experience, I have been applying arithmetic, algebra, trigonometry, differential equations, numerical analysis, optimization, probability and statistics, simulation, and matrix theory in building automation suites. For example -   * I have used algebra in calculating the limit bounds of an array or a list of variables, while testing the algorithms. * **Simulation of production systems in QA environment to synchronize the actual behaviour of applications.** * **Data analytics in all phases of SDLC (Software Development Life Cycle) & STLC (Software Testing Life Cycle) in one form or the other to optimize the solutions, preparing matrices, Software testing delivery etc..** * **I have used arithmetic calculations and trigonometry to build/prepare end to end tools, documents, reports in SDLC & STLC.** * **I have implemented differential equations in order to stream line the processes.** * **Optimization in different phases of SDLC & STLC. Here are the few optimization techniques I have worked upon –** * Software code optimization * Design optimization * Model optimization * Algorithmic optimization * Profiling * Loops tuning * CPU/platform specific features * Test scenarios optimization * **I have prepared metrics to track the project deliverables with milestones in SDLC & STLC. Here are the few matrices that I have worked upon -** * Requirement traceability matrices * Forward Traceability Matrix - Mapping of requirements to test cases. * Backward Traceability Matrix - Mapping of test cases to requirements. * Bi-Directional Traceability – Mapping from test cases to basis documentation and vice versa. * Test Plan matrices * Mapping of test scenarios to test resources * Defect traceability matrices * Mapping of defects to test scenarios to requirements * Test Result Matrices * Matrices showing test coverage, bugs, etc.. in the form of graphs and charts   Sample Test Execution metrics (following S-Curve methodology) –    Sample Defects metrics –    **Examples of Application of numerical methods and statistics:**  **Example 1:**  **Situation –**  Testing of HSBC credit card applications when I was employed in Capgemini in the year 2008.  **Problem –**  Testing of highly complex application, which is used to calculate the FICO score of the customers located in USA based on their transaction history and assets.  **Description –**  FICO tool is used by the HSBC bank to calculate the FICO score of the customers in USA, who are willing to apply for a credit card of HSBC.  FICO Scores are calculated based on the five categories listed below. For some groups, the importance of these categories may vary; for example, people who have not been using credit long will be factored differently than those with a longer credit history.   * Payment history * Amounts owed (Credit Utilization) * Length of credit history * Credit mix in use * New credit   The importance of any one factor in credit score calculation depends on the overall information in credit report. For some people, one factor may have a larger impact that it would for someone with a much different credit history. In addition, as the information in credit report changes, so does the importance of any factor in determining FICO Scores.  Therefore, it’s impossible to measure the exact impact of a single factor in how your credit score is calculated without looking at entire report. Even the levels of importance shown in the FICO Scores chart are for the general population, and will be different for different credit profiles.  Image result    Source of fico score chart/image : Internet  In order to calculate the FICO score based on the data of customers, I have to use the algorithm of the application. The logic of the application was build using physical sciences like calculus, probability, numerical analysis and formulas. Hence, in order to test the application, I have to build stubs using same calculus and formulas in order to mock the behaviour of the application in QA environment (Quality Assurance).  Apart from calculating scores, the tool also suggests customers on points to improve their credit scores, if their score is less than 600. So I wrote logic in Java to test the application by using formulas and calculus.  **Example 2:**  **Situation –**  Build automation solution for the trading application using Java and C# programming languages for Equities domain in Citibank.  **Problem –**  Automation solution for Citi trading application (CitiSmart) from scratch.  **Description –**  Citismart application is the trading engine which connects with the stock exchanges and gets the fills for the orders placed by client.  When a client places an order from a customer facing UI application, the orders are routed to the CitiSmart application and based on the strategy and the quantity of the order it connects with different exchanges and internal dark pool (inventory maintained by Citi to fulfil the client orders internally) to check for the fills and depending upon the availability of stock, it can either respond back with the partially filled orders or fully filled order.  Citismart algorithm basically works upon the strategy (pre-defined by the client) of the order. The strategies in algo trading are broadly classified as –   * VWAP (Volume Weighted Average Price) * TWAP (Time Weighted Average Price) * DAGGER * ICEBERG * DMA (Direct Market Access) * PEGGED * CLOSE   Based on the strategies, algorithms are defined in the CitiSmart application to pick the order from the client in the form of fix messages and create splits of the order (child orders) and send parent level & child level fix messages through Tibco EMS topics to internal systems (KDB services, UI’s, Databases etc.) and exchanges.  As per the logic defined in exchange simulators, fills are sent back to the client through CitiSmart application.  End to End applications involved in building automation suite –   1. CFORE (Client facing UI application) 2. CitiSmart (Algo trading engine) 3. Exchange Simulators (Stubs of actual stock exchanges)   TRADE FLOW :-    Based on the behaviour and applications mentioned above, I started working on preparing a hybrid framework (Data driven and keyword driven) which collates the data from XML and excel files and pumps data into CFORE application (mocking client actions) and reads fix messages from tibco ems and validates the values of the fix tags as per the requirement.  In order to validate the values of tags I used concepts of calculus, mathematics and formulas. Some of the sample tag names which requires calculation are Client order ID, Order quantity, Fills quantity, Trade aggression level, Child price, Child quantity and custom tags defined by Citi standards.  Sample formulas used –   * To calculate square root –      * To generate Fibonacci series –      * Mod of a number      * Generate random alphanumeric string in C# | **Provide annotations to your supplementary evidence (document and page number)** |

|  |  |
| --- | --- |
| Element Three  **A systematic, theory-based formulation of engineering fundamentals required in the engineering discipline** | |
| **Context**  Engineering fundamentals provide the knowledge base for engineering specialisations and represent a systematic formulation of engineering concepts and principles based on mathematical and natural sciences to support applications.  The core areas of engineering fundamentals knowledge include fluid mechanics, statics and dynamics, electric circuits, solid mechanics, thermodynamics, heat transfer, mass transfer, and properties of materials.  Washington Accord graduates are expected to be able to apply this knowledge of engineering fundamentals to solve complex engineering problems. | |
| Performance Indicators   * Ability to define key factual information in core areas of fundamental engineering knowledge relevant to your engineering discipline * Evidence of sufficient depth of knowledge of engineering fundamentals to demonstrate an ability to think rationally and independently within and outside a chosen field of specialisation * Evidence of sufficient breadth of knowledge of engineering concepts and principles to allow subsequent professional development across a broad spectrum of engineering * Ability to apply knowledge of engineering fundamentals to solve complex engineering problems relevant to your discipline | |
| Summarise your knowledge of the core engineering fundamentals (as listed above) and how they have 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.  The courses that I studied during my study to obtain my Engineering degree can be classified based on their relevance to my discipline as follows.   * Foundation courses: These are the basics of the following courses – Mathematics, Physics, Fundamentals of computer engineering (I & II) and Fundamentals of computer engineering lab (I & II). * Supporting courses: These courses provide the tools for Engineering practice, such as Computer programming, C/C++, Data Structures, Engineering drawing and Computer aided drafting, CAD/CAM theory, CAD/CAM lab and Database management systems.   I have already covered the courses and examples related to natural science in element#1, so I will focus much on the fundamental knowledge of Information technology concepts like SDLC (Software Development Life Cycle)/STLC (Software Testing Life Cycle) in this section.  Software development is the process of computer programming, documenting, testing, and bug fixing involved in creating and maintaining applications and frameworks resulting in a software product by following all the engineering principles and minimum development standards like Design & code quality, Source code management, Build & release management, Unit testing and code reviews by peers.  Software development is a process of writing and maintaining the source code, but in a broader sense it includes all that is involved between the conception of the desired software through to the final manifestation of the software, sometimes in a planned and structured process. Therefore, software development may include research, new development, prototyping, modification, reuse, re-engineering, maintenance, or any other activities that result in software products.  I have been implementing and working upon following engineering fundamentals in SDLC & STLC throughout my experience -   * Software development fundamentals * Modularity      * Data Abstraction      * Polymorphism      * Encapsulation      * Inheritance * *Inheritance allows child classes to inherit the characteristics of existing parent class* * *Attributes (fields and properties)* * *Operations (methods)* * *Child class can extend the parent class* * *Add new fields and methods* * *Refine methods (modify existing behaviour)* * *A class in any OOP’s language can implement an interface by providing implementation for all its methods*      * Documentation * Cost * Reusability * Performance   Implemented following engineering fundamentals while developing a search engine tool using PHP programming language –   * Model Construction * Structural design * Implementation model * Software testing model   **Examples of Application following Engineering fundamentals:**  **Example 1:**  **Situation –**  Build automation solution for the declined payment scenario on Ryanair website using Java programming language.  **Problem –**  Capgemini was trying to buy in a project from Ryanair and I was asked to create a POC (Poof of concepts) to demonstrate the automation approach using tools like “Selenium web driver” to higher management of Ryanair.  **Description –**  *Source – Wiki*  Ryanair established in 1984, has grown from a small airline flying the short journey from Waterford to London into one of Europe's largest carriers. Ryanair now has over 9,500 members of staff working for the company, most of whom are employed and contracted by multiple agencies to fly on Ryanair aircraft, or as is the case for pilots, the vast majority are either agency employed or self-employed and their services are contracted to Ryanair. As of 2013 including over 1,200 pilots. After the rapidly growing airline went public in 1997, the money raised was used to expand the airline into a pan-European carrier. Revenues have risen from €231 million in 1998, to €1,843 million in 2003 and €3,013 million in 2010. Similarly net profits have increased from €48 million to €339 million over the same period  Ryanair Ltd. (ISEQ: RYA, LSE: RYA, NASDAQ: RYAAY) is an Irish low-cost airline headquartered in Swords, Dublin, Ireland, with its primary operational bases at Dublin and London Stansted Airports. Ryanair was both the largest European airline by scheduled passengers carried, and the busiest international airline by passenger numbers.  Ryanair operates over 350 Boeing 737-800 aircraft, with a single 737-700 maintained on the roster as a backup plane and for pilot training.[6] The airline has been characterised by its rapid expansion, a result of the deregulation of the aviation industry in Europe in 1997 and the success of its low-cost business model. Ryanair's route network serves 32 countries in Europe, Africa (Morocco), and the Middle East (Israel).  **Automation solution** -  I prepared an automation solution for Ryanair website using Java programming language and “Selenium web driver” automation tool. Approach & Technology relating engineering fundamentals: -  * Understood the requirements and assessed the feasibility of creating the automation suite. * Used selenium webdriver drivers with Page object model framework (POM) framework and Java as programming language. * The requirements have been broken into modules/pages following POM framework and modularity, encapsulation, inheritance and polymorphism engineering fundamentals & principles. * Test data driven approach by taking Microsoft excel as input file, using Apache POI approach to integrate excel with Java following data abstraction engineering fundamental & principle. * Handle errors using exception handling techniques. * Advanced reporting using ExtentReports and documentations. HTML file format with rich UI experience (time stamp, title, validation steps, user friendly look) and screenshots. This follows the documentation and reusability concepts of software fundamentals & principles.  |  |  | | --- | --- | | Tools & Technologies | Details | | Operating system | Windows | | Programming Language | Java | | Automation tools | Selenium webdriver & Maven | | Automation framework | Page Object Model (POM) with Page Factory framework | | TDD approach | Apache POI API | | Reporting methodology | Advanced selenium reporting using ExtentReports | | IDE | IntelliJ IDEA | | Browser | Mozilla Firefox | | Continuous Integration (CI) tool | Git |   **Code Overview** –    Modularization  Data abstraction  Polymorphism  Encapsulation  Inheritance  **Example 2:**  **Situation –**  Build automation solution for the trading application using Java and C# programming languages for Equities domain in Citibank.  **Problem –**  Automation solution for Citi trading application (CitiSmart) from scratch.  **Description –**  Citismart application is the trading engine which connects with the stock exchanges and gets the fills for the orders placed by client.  When a client places an order from a customer facing UI application, the orders are routed to the CitiSmart application and based on the strategy and the quantity of the order it connects with different exchanges and internal dark pool (inventory maintained by Citi to fulfil the client orders internally) to check for the fills and depending upon the availability of stock, it can either respond back with the partially filled orders or fully filled order.  I have explained the entire Trading flow in element 2, so I will focus more on the automation solution and approach and explain how engineering fundamentals were followed in building the automation solution.  **Automation solution** -  I prepared an automation solution for Citismart application using Java programming language and “Selenium web driver” automation tool. I also used C# to interact with the Tibco servers. Approach & Technology relating engineering fundamentals: -  * Understood the requirements and assessed the feasibility of creating the automation suite. * Created modules (classes) for each components of citismart application using Java programming language. * Used selenium webdriver driver jars with TestNG framework. * The requirements have been broken into modules (classes) following TestNG framework and modularity, encapsulation, inheritance and polymorphism engineering fundamentals & principles. * Test data driven approach by taking XML’s , property files as input and integrate them with Java following data abstraction engineering fundamental & principle. * Handle errors using exception handling techniques. * Advanced reporting using TestNG and documentations. HTML file format with rich UI experience (time stamp, title, validation steps, user friendly look) and screenshots. This follows the documentation and reusability concepts of software fundamentals & principles.  |  |  | | --- | --- | | Tools & Technologies | Details | | Operating system | Windows | | Programming Language | Java | | Automation tools | Selenium webdriver & Maven | | Automation framework | TestNG framework with Cucumber | | Reporting methodology | Advanced selenium reporting using TestNG | | IDE | IntelliJ IDEA | | Browser | Mozilla Firefox/IE/Chrome | | Continuous Integration (CI) tool | Git |   Modularization  **Code Overview** –    Polymorphism  Encapsulation  Inheritance    Data abstraction and implementation  Data abstraction and utilization | **Provide annotations to your supplementary evidence (document and page number)** |

|  |  |
| --- | --- |
| 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.  In Jan 2016, When I joined CitiCorp services (Citibank) and I started working on complex problems related to Citi trading applications. The algorithms written in Citi trading engine is too complex and in order to verify/validate all the algorithms, I have to write complex java programs which can interpret the algorithms and give the desired results as output.  Since, I have been working on such technologies from last 8 years, I have core depth knowledge of software engineering concepts like C, C++, Data Structure and algorithms, code development, Database management systems, Java technologies, Mainframe systems, Software testing, Business analysis & optimization, Requirement analysis, Test planning, project delivery, reporting methodologies, end to end development frameworks etc..  Being expertise in these technologies and applications helped me in solving the complex problems in my current organization (Citibank).  **Example 1:**  **Situation –**  Create unit test cases using JUnit for OD (Order distributor) & TE (Trading Engine) components and validate the methods which are configured as per the compliance rules.  **Problem –**  Interpret and build unit test cases which can cover all the aspects/flows of the complex algorithms written in OD & TE components of Citismart application.  **Description –**  Algorithms written in OD & TE components are very complex and in order to interpret and cover all the aspects/flows of algorithms using JUnit, I used Java and JUnit to achieve the results.  **I started with adding dependencies in POM.xml, since I was using Maven framework.**  <dependency>     <groupId>junit*</*groupId>     <artifactId>junit</artifactId>     <version>4.11</version>  </dependency>  **I used following JUnit annotations to indicate the tag the code with the tests** –  ***@Test:*** The Test annotation tells JUnit that the public void method to which it is attached can be run as a test case. To run the method, JUnit first constructs a fresh instance of the class then invokes the annotated method. Any exceptions thrown by the test will be reported by JUnit as a failure. If no exceptions are thrown, the test is assumed to have succeeded.  ***@Test (expected = Exception.class):*** Sometimes we need to test the exception to be thrown by the test. @Test annotation provides a parameter called 'expected', declares that a test method should throw an exception. If it doesn't throw an exception or if it throws a different exception than the one declared, the test fails.  ***@Test(timeout=xxx):*** Sometimes we need to measure the performance in terms of time. The @Test annotations provides an optional parameter called 'timeout', which causes a test to fail if it takes longer than a specified amount of clock time (measured in milliseconds).  ***@Before:*** When writing tests, it is common to find that several tests need similar objects created before they can run. Annotating a public void method with @Before causes that method to be run before the Test method. The @Before methods of super classes will be run before those of the current class.  ***@After:*** If you allocate external resources in a Before method you need to release them after the test runs. Annotating a public void method with @After causes that method to be run after the Test method. All @After methods are guaranteed to run even if a Before or Test method throws an exception. The @After methods declared in superclasses will be run after those of the current class.  ***@BeforeClass:*** Sometimes several tests need to share computationally expensive setup (like logging into a database). While this can compromise the independence of tests, sometimes it is a necessary optimization. Annotating a public static void no-arg method with @BeforeClass causes it to be run once before any of the test methods in the class. The @BeforeClass methods of superclasses will be run before those the current class.  The annotations @BeforeClass and @Before are same in functionality. The only difference is the method annotated with @BeforeClass will be called once per test class based, and the method annotated with @Before will be called once per test based.  ***@AfterClass:*** If you allocate expensive external resources in a BeforeClass method you need to release them after all the tests in the class have run. Annotating a public static void method with @AfterClass causes that method to be run after all the tests in the class have been run. All @AfterClass methods are guaranteed to run even if a BeforeClass method throws an exception. The @AfterClass methods declared in superclasses will be run after those of the current class.  The annotations @AfterClass and @After are same in functionality. The only difference is the method annotated with @AfterClass will be called once per test class based, and the method annotated with @After will be called once per test based.  ***@Ignore:*** Sometimes you want to temporarily disable a test or a group of tests. Methods annotated with Test that are also annotated with @Ignore will not be executed as tests. Also, you can annotate a class containing test methods with @Ignore and none of the containing tests will be executed. Native JUnit 4 test runners should report the number of ignored tests along with the number of tests that ran and the number of tests that failed.  You can also use @Ignore annotation at class level.  **In addition to above mentioned dependencies and annotations, I also focused on below mentioned techniques** –   * Created nested classes to use output of a method as input in child classes. * Used queues, stacks & Linked Lists data structures to handle the flow of data between classes. * Used assertions like assertTrue(), assertFalse(),assertEquals(), assertNull(), assertNotNull() etc.. to validate the output of classes with the expected results. * Used constructors, abstract classes etc.. to initialize the global variables and use them across the Junit test cases.   I couldn’t attach the snapshot of these Unit test cases built in Citi because it’s confidential information and related to trading algorithms which cannot be accessed from public network and are also against the policies of the organization. However, an overview of how a JUnit test case looks can be seen in the snapshot below.  img_55537df53fe7c  I am a knowledgeable Software engineer and expertise in critical designing, research and operations of software’s. I developed these skills through on-job learning and continuing professional development.  Using above mentioned dependencies, annotations and assertions, exception handling techniques, I was able to cover all the aspects and flows of OD & TE components and solve the complex problems. I am an expert in the computer programming using Java language where I create computer programs that were executable and used it to solve problems in computing.  **Example 2:**  **Situation –**  Build automation solution for the declined payment scenario on Ryanair website using Java programming language.  **Problem –**  Capgemini was trying to buy in a project from Ryanair and I was asked to create a POC (Poof of concepts) to demonstrate the automation approach using tools like “Selenium web driver” to higher management of Ryanair.  **Description –**  *Source – Wiki*  Ryanair established in 1984, has grown from a small airline flying the short journey from Waterford to London into one of Europe's largest carriers. Ryanair now has over 9,500 members of staff working for the company, most of whom are employed and contracted by multiple agencies to fly on Ryanair aircraft, or as is the case for pilots, the vast majority are either agency employed or self-employed and their services are contracted to Ryanair. As of 2013 including over 1,200 pilots. After the rapidly growing airline went public in 1997, the money raised was used to expand the airline into a pan-European carrier. Revenues have risen from €231 million in 1998, to €1,843 million in 2003 and €3,013 million in 2010. Similarly net profits have increased from €48 million to €339 million over the same period  Ryanair Ltd. (ISEQ: RYA, LSE: RYA, NASDAQ: RYAAY) is an Irish low-cost airline headquartered in Swords, Dublin, Ireland, with its primary operational bases at Dublin and London Stansted Airports. Ryanair was both the largest European airline by scheduled passengers carried, and the busiest international airline by passenger numbers.  Ryanair operates over 350 Boeing 737-800 aircraft, with a single 737-700 maintained on the roster as a backup plane and for pilot training.[6] The airline has been characterised by its rapid expansion, a result of the deregulation of the aviation industry in Europe in 1997 and the success of its low-cost business model. Ryanair's route network serves 32 countries in Europe, Africa (Morocco), and the Middle East (Israel).  **Automation solution** -  I prepared an automation solution for Ryanair website using Java programming language and “Selenium web driver” automation tool. Approach & Technology relating engineering fundamentals: -  * Understood the requirements and assessed the feasibility of creating the automation suite. * Used selenium webdriver drivers with Page object model framework (POM) framework and Java as programming language. * The requirements have been broken into modules/pages following POM framework and modularity, encapsulation, inheritance and polymorphism engineering fundamentals & principles. * Test data driven approach by taking Microsoft excel as input file, using Apache POI approach to integrate excel with Java following data abstraction engineering fundamental & principle. * Handle errors using exception handling techniques. * Advanced reporting using ExtentReports and documentations. HTML file format with rich UI experience (time stamp, title, validation steps, user friendly look) and screenshots. This follows the documentation and reusability concepts of software fundamentals & principles.  |  |  | | --- | --- | | Tools & Technologies | Details | | Operating system | Windows | | Programming Language | Java | | Automation tools | Selenium webdriver & Maven | | Automation framework | Page Object Model (POM) with Page Factory framework | | TDD approach | Apache POI API | | Reporting methodology | Advanced selenium reporting using ExtentReports | | IDE | IntelliJ IDEA | | Browser | Mozilla Firefox | | Continuous Integration (CI) tool | Git |   **Code Overview** –    During development of this automation solution, I had & still possess the knowledge of software development/engineering concepts. By following these princples, I implemented the techniques in the solution. As outlined and detailed above, I used specialist engineering knowledge to solve complex software engineering problems. | **Provide annotations to your supplementary evidence (document and page number)** |

|  |  |
| --- | --- |
| Element Five  **Knowledge that supports engineering design.** | |
| **Context**  The design process – the root of engineering – is the process of devising a system, component or process to meet desired needs. Engineering design is a systematic process that involves problem definition and scoping, research, analysis, option development and selection, modelling to predict future performance, detailed design and testing. Importantly, it also involves communication of the outcome in a way that enables the design solution to be realised.  Washington Accord graduates are expected to be able to apply this knowledge of the design process to solve complex engineering problems. | |
| Performance Indicators   * Ability to undertake research and analysis to support the design process * Ability to investigate a situation or the behaviour of a system and identify relevant causes and effects * Ability to develop from first principles and construct mathematical, physical and conceptual models of situations, systems and devices, with a clear understanding of the assumptions made in development of such models * Application of technical knowledge, design methods and appropriate tools and resources to design components, systems or processes to meet specified criteria * Ability to analyse the pros and cons of alternative design options to support the development of an optimised design alternative * Ability to analyse the constructability or manufacturing feasibility of a project or product * Experience of personally conducting a significant design exercise, providing evidence of the consideration of various realistic constraints, such as safety, reliability, ethics, economic factors, aesthetics and social impact. * Ability to apply appropriate design methods in solving complex engineering problems | |
| **Summarise your knowledge that supports engineering design relevant to your discipline and how it has been developed and applied 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.**  In software engineering and considering my area of experience being in software automation development, the software design part is very critical comes under my roles and responsibilities. As a software automation developer I have been implementing the software designing throughout my career.  Let me give you an overview of software designing first and then I will go through the examples, where I will cover, how I have implemented software designing concepts in my development career.  ***Software Designing*** :-  Software design is a process to transform user requirements into some suitable form, which helps the programmer in software coding and implementation.  For assessing user requirements, an SRS (Software Requirement Specification) document is created whereas for coding and implementation, there is a need of more specific and detailed requirements in software terms. The output of this process can directly be used into implementation in programming languages.  Software design is the first step in SDLC (Software Design Life Cycle), which moves the concentration from problem domain to solution domain. It tries to specify how to fulfil the requirements mentioned in SRS.  **Software Design Levels**  Software design yields three levels of results:   * **Architectural Design** - The architectural design is the highest abstract version of the system. It identifies the software as a system with many components interacting with each other. At this level, the designers get the idea of proposed solution domain. * **High-level Design** - The high-level design breaks the ‘single entity-multiple component’ concept of architectural design into less-abstracted view of sub-systems and modules and depicts their interaction with each other. High-level design focuses on how the system along with all of its components can be implemented in forms of modules. It recognizes modular structure of each sub-system and their relation and interaction among each other. * **Detailed Design** - Detailed design deals with the implementation part of what is seen as a system and its sub-systems in the previous two designs. It is more detailed towards modules and their implementations. It defines logical structure of each module and their interfaces to communicate with other modules. * **Modularization**   Modularization is a technique to divide a software system into multiple discrete and independent modules, which are expected to be capable of carrying out task(s) independently. These modules may work as basic constructs for the entire software. Designers tend to design modules such that they can be executed and/or compiled separately and independently.  Modular design unintentionally follows the rules of ‘divide and conquer’ problem-solving strategy this is because there are many other benefits attached with the modular design of a software.  Advantage of modularization:   * Smaller components are easier to maintain * Program can be divided based on functional aspects * Desired level of abstraction can be brought in the program * Components with high cohesion can be re-used again * Concurrent execution can be made possible * Desired from security aspect * Concurrency   Back in time, all software are meant to be executed sequentially. By sequential execution we mean that the coded instruction will be executed one after another implying only one portion of program being activated at any given time. Say, a software has multiple modules, then only one of all the modules can be found active at any time of execution.  In software design, concurrency is implemented by splitting the software into multiple independent units of execution, like modules and executing them in parallel. In other words, concurrency provides capability to the software to execute more than one part of code in parallel to each other. It is necessary for the programmers and designers to recognize those modules, which can be made parallel execution.   * **Coupling and Cohesion**   When a software program is modularized, its tasks are divided into several modules based on some characteristics. As we know, modules are set of instructions put together in order to achieve some tasks. They are though, considered as single entity but may refer to each other to work together. There are measures by which the quality of a design of modules and their interaction among them can be measured. These measures are called coupling and cohesion.   * **Cohesion**   Cohesion is a measure that defines the degree of intra-dependability within elements of a module. The greater the cohesion, the better is the program design.  There are seven types of cohesion, namely –   * **Co-incidental cohesion** - It is unplanned and random cohesion, which might be the result of breaking the program into smaller modules for the sake of modularization. Because it is unplanned, it may serve confusion to the programmers and is generally not-accepted. * **Logical cohesion** - When logically categorized elements are put together into a module, it is called logical cohesion. * **Temporal Cohesion** - When elements of module are organized such that they are processed at a similar point in time, it is called temporal cohesion. * **Procedural cohesion** - When elements of module are grouped together, which are executed sequentially in order to perform a task, it is called procedural cohesion. * **Communicational cohesion** - When elements of module are grouped together, which are executed sequentially and work on same data (information), it is called communicational cohesion. * **Sequential cohesion** - When elements of module are grouped because the output of one element serves as input to another and so on, it is called sequential cohesion. * **Functional cohesion** - It is considered to be the highest degree of cohesion, and it is highly expected. Elements of module in functional cohesion are grouped because they all contribute to a single well-defined function. It can also be reused. * **Coupling**   Coupling is a measure that defines the level of inter-dependability among modules of a program. It tells at what level the modules interfere and interact with each other. The lower the coupling, the better the program.  There are five levels of coupling, namely -   * **Content coupling** - When a module can directly access or modify or refer to the content of another module, it is called content level coupling. * **Common coupling** - When multiple modules have read and write access to some global data, it is called common or global coupling. * **Control coupling** - Two modules are called control-coupled if one of them decides the function of the other module or changes its flow of execution. * **Stamp coupling** - When multiple modules share common data structure and work on different part of it, it is called stamp coupling. * **Data coupling** - Data coupling is when two modules interact with each other by means of passing data (as parameter). If a module passes data structure as parameter, then the receiving module should use all its components. * **Design Verification**   The output of software design process is design documentation, pseudo codes, detailed logic diagrams, process diagrams, and detailed description of all functional or non-functional requirements.  The next phase, which is the implementation of software, depends on all outputs mentioned above. It is then becomes necessary to verify the output before proceeding to the next phase. The early any mistake is detected, the better it is or it might not be detected until testing of the product. If the outputs of design phase are in formal notation form, then their associated tools for verification should be used otherwise a thorough design review can be used for verification and validation.  By structured verification approach, reviewers can detect defects that might be caused by overlooking some conditions. A good design review is important for good software design, accuracy and quality.  **Examples of Application using Engineering design:**  **Example 1:**  **Situation –**  Build automation framework from scratch for Barclaycard US EMWE websites when I was employed in Barclays.  **Problem –**  Barclaycard US team was a new team and I was appointed to develop the automation framework and solution for Barclaycard US EMWE (customer servicing websites) from scratch.  **Description –**  Barclaycard US is the 2nd largest credit card provider in United States providing cards of more than 50 partners which includes Apple, William Sonama, Bank of Hawaii, Travelocity etc.  Barclaycard US has also started providing bank cards for individual customers in US. US Cards and Marketing vertical handles all the functional & automation testing of customer facing Acquisition websites (aka Apply) and customer servicing websites (aka EMWE) in Agile model. Barclaycard US internet websites are developed as per requirement of partners to cater the customer requirements for online access. Automation testing of this project involves end to end testing of enhancements involving integration with multiple applications viz., TS2 Mainframe Backend, Oracle Database, and MQ Series.  Since it was a new team, the environment was not setup for automation projects and given an opportunity, I started working on setting up a new robust QA automation environment/framework to accommodate the functionalities involved in EMWE.  I started preparing the proof of concepts for the framework by gathering the details like -   * Automation Tools & Technologies required. * Methodology for interacting between multiple systems. * Return on Investment (ROI) * Development standards * Effort savings (Cost savings)   I defined the development standards for automation developers so that I and the new team members can use it later in preparing the automation suites, which is in adherence to the hybrid framework.  The hybrid framework included –   * AUTOMATION COMPONENTS * HPQC FOLDER STRUCTURE * CODING STANDARDS AND NAMING CONVENTIONS * Common External Actions * Project External Actions * COMMON FUNCTIONS * GENERIC GUIDELINES     Let me go through the each components of framework in detail. Automation Components  * 1. **Test Scripts**   Test Scripts are QTP scripts which is a combination of External Actions and the Library functions. These scripts are the test scenarios which are executed on the Test Lab in HPQC.  The Test scripts are saved under the Folder “05 Projects” in the Test Plan.     * 1. **External Action**   External Actions are the reusable components which also is a QTP script. These scripts are developed for any common function which is to be implemented using the Object Repository.  The External Actions are saved under the Folder “04 External Actions” in the Test Plan.     * 1. **Object Repository**   Object Repository is the collection of Test Objects which are identified by QTP. It acts as interface between the Test Script and the AUT in order to identify the objects during the execution.  The Object Repositories are saved under the Folder “02 Object Repositories” in the Test Resources.       * 1. **Test Data**   Test Data is stored in the spread sheets. The test scripts, external actions access the test data files during the execution of the test scripts.  The Test Data spread sheets are saved as attachments to the “03 Test Data” folder in Test Plan.     * 1. **Function Library**   Function library is Quick Test Document containing VBscript functions and subroutines. The Function Libraries are saved in the “01 Functions” folder in Test Resources.   HPQC Folder Structure The Test Plan and the Test Resources in HPQC are structured based on the different Automation components used for the development of the automation packages.  The Test Plan and the Test Resources contains all the automation components. A detailed classification of the Test Plan and the Test Resources structure are as explained below:  **2.1)** **Test Plan:**  The Test Plan module contains the Test Data, Save Value sheets, External Actions and the Test Scripts. Each of these components is saved onto their respective folders as shown in the figure.    Under the Subject Folder, “03 Test Data”, “04 External Actions” and “05 Projects” are created. “Save Values” folder is created under the “03 Test Data” folder.  In the “03 Test Data” folder all the test data spread sheet would be saved as an attachment and the all the Save Values sheets would be saved as an attachment to the “Save Values” folder.  The Reusable External Actions are saved under the “04 External Actions” folder onto their respective Application Folder.  The Test Scripts are saved under the “05 Projects” folder onto their respective Application Folder.  **2.2) Test Resources:**  The Test Resources module contains the Function Library and the Object Repositories. Each of these components is saved onto their respective folders as shown in the figure.    Under the “Resources” folder a “Subject” folder is created which contains two folders namely “01 Functions” and “02 Object Repositories”.  The function library file is saved under the “01 Functions” and the Object Repositories are saved under the “02 Object Repositories” folder. Coding Standards and Naming Conventions  * 1. **Object Repositories**   **Note.** Most of the Object Repositories are created already. Before start to create the new Object Repositories you need to be sure that this Object Repository does not exist.   * + 1. **Object Repository Name**   The Name of the Object Repository must reproduce name of the group of screens which it belongs to. It should contain the Application Name with the Application Mnemonic for which it has been created.  The naming format would be **“<App Mnemonic>\_<App Name>”** e.g. The Object Repository for the TLS application would be “UO\_TLS”   * + 1. **Object Naming conventions**   **Browser Object:** The Browser Object should be the name of the application along with the Application Mnemonic.  The naming format would be **“<App Mnemonic>\_<App Name>”** or **“<App Name>” (only if applicable)** e.g. The Browser Object name for the IP application should be UE\_IP  **Page Object:** The Page Object should be the name of the page displayed in the application screen. It should contain the Page Title.  The naming format would be **“<Page Title>” (exclude the server details, if applicable)** e.g. “CS4 - Customer Products”  The title property is mandatory for a page. Use regular expression for the title property in order to use the page object for both English and French screens. Also use regular expression if there are any server details included in the title property value.      **Other Test Objects:** For the rest of the object, the object name should be a logical name which has is relevant to the page or the application.  The Object properties should contain Regular Expression in order to use the same object for both English and French screens and the name of the object should be an English equivalent name. There should not be a separate test object created for the French screen.      For the Table objects it is mandatory to use **“column names”** as a mandatory property. Use **“Index”** property, only if applicable.    **Enable Smart Identification:** This should always be **False** for any Test Object. Make sure that the Smart Identification is disabled before adding any new objects. Follow the below steps to disable the Smart Identification.  In the QuickTest Professional, under the Tools menu, select “Object Identification…”    The **“Enable Smart Identification”** should be Unchecked for the required Test Objects.     * 1. **External Actions**   External Actions are the Reusable Actions; these actions are saved under “04 External Actions” folder.    There are two types of External Actions with respect to Test Scenario, Common External Actions and the Application specific External Actions.  The External Actions should be saved and called from its respective application folder. e.g. If we need to use a Print Document action, then it has to be called from the “Print Actions” folder. Likewise if we need to use Customer actions like “CS3\_Customers\_Found” we need to call it from “Customer” folder.  There should be a separate External Action for the Data Entry and for the Verification while writing External Actions for the same page.    **3.2.1 External Actions Naming Convention**    External Actions are named using a standard format. This naming convention will give the information about the purpose of the external action and which application it belongs to.  The naming convention is of the following format:  **“<App Mnemonic>\_<Page Mnemonic><*space*><Page Name> <*space*>** - **<*space*> <Purpose>”**  e.g**.** “UE\_IP415 PAC Setup – Enter data”    **3.2.2 External Actions Properties**  The External Action property name should be same as the External Action name.    The Input parameters define the number of parameter(s) that is passed to an External Action. The Default value for an Input parameter should always be “NULL”.    **3.2.3 Other Conventions**    While writing the External Actions, the first check is to be made to the NULL parameters e.g. as shown below  If <Parameter Id> = “NULL” Then  ‘Reporter Statements  ‘Exit the Test  End If  After the first Null Check the External Action should mandatorily have a Sync statement of the page for which the External Action belongs to. e.g. a UE\_IP416 PAC Inquiry action should have a Sync statement for the PAC Inquiry page. (Use whichever is appropriate)  At the End of the External Action there should be a Reporter Statements that reports the successful execution of the External Action.  Avoid using the button click or any other click operation (which submits a transactions) if it has to be used at the end of the action. Use these click operations outside the external action (use it in the test scripts).   * 1. **Test Scripts**   Test Scripts are the Test Scenarios, which is a combination of Automation components. These Test Scripts should be placed in “**05 Projects”** folder under the **Subject** folder.  Each Application has an Application folder under the “05 Projects” folder.    **3.3.1 Naming Convention**    The Test Scripts have a standard naming convention. The test script name gives the following information: Application name, Language Type, Module name and the purpose.  The naming convention is of the following format:  **“<Application Mnemonic>\_<Language>\_<Serial No>\_<Module Name (if applicable)>\_<Purpose: Logical Name>”**  **3.3.2 Action Properties**  The Action property name should not be same as the Test Script name. The Action Property name should only be the <Logical Name> part of the Test Script name. The test scripts are not reusable components and hence the “Reusable action” check box should be unchecked.    The Test Scripts do not have any Input Parameters.    **3.3.3 Other Conventions**    While developing the scripts, comments must be included in the Scripts. Scripts should include sectional comments and comments for individual actions or step if necessary. For the French test scripts, objects used have an English name and hence we should mention the corresponding French word in the comments.  If there are any Pre Conditions like “Print Yes No” or “Entering the UAT cycle date” it should written as a separate test script.   * 1. **Function Library**   Function library is Quick Test Document containing VBscript functions and subroutines.  The Function Libraries are saved in the “01 Functions” folder in Test Resources.    The function library is a .vbs format file. All the web functions are saved in the “CommonFunctions.vbs”    The Common Function library contains the test data paths; excel file connections, error handling, Click function and some of the other important functions which are called during the execution of the Test Scripts.   * 1. **Test Data**   Test Data is stored in the spread sheets. The test scripts, external actions access the test data files during the execution of the test scripts. The Test Data spread sheets are saved as attachments to the “03 Test Data” folder in Test Plan.    Each sheet in the Test Data should represent the data for a particular page; screen or a tab.  The Test Save Values sheets hold the data which are captured from the application during the test execution. The Save Values sheets are saved under the “03 Test Data” folder with the name “Save Values”.    **3.5.1 Naming Conventions**  The Test Data name should represent the application mnemonic and the application name. The format for naming the Test Data is: **“<Application Mnemonic>\_<Application Name>”**  The name of the sheet in the Test Data should represent the page mnemonic and the page name. The format for naming a sheet in the test data is: **“<Page/Screen/Tab Mnemonic (if applicable)>\_<Page/Screen/Tab Name>”**    The sheet name of the Save Values sheet should be same as the file name excluding “\_SaveValues” e.g. If the Save Value file name is **“IX\_BranchEmailUpgrade\_SaveValues.xls”** then the sheet name would be **“IX\_BranchEmailUpgrade”.**   1. **Actions**  Common External Actions The external actions which can be reused and called across multiple test scripts/actions in a project are called common external actions. Project External Actions Project External Actions are a specific project actions but it may be used from another Project. You can find all Project External Actions under HP QC Test Plan (04 External Actions/’Project’) folder.   1. **Common Functions**   All Common Functions must be placed in Quality Center under: **Resources/Subject/01 Functions/ (Test Resources)**    5.1) Function: fnSetVariable **Purpose:** To set the value into SaveValues.xls  fnSetVariable (strSheetName, strVariableName, strVariableValue)  5.2) Function: fnGetVariable **Purpose:** To get the value from SaveValues.xls  fnGetVariable(strSheetName, strVariableName)  5.3) Function: fnClick **Purpose:** Click on Button, Link, Image, WebElement  fnClick (obj)  Example :-  *fnClick(Browser("Customer").Page("CS5 – Customer Information").WebButton("Done"))*  5.4) Function: DropDownListVerification **Purpose:** To verify DropDownList  DropDownListVerification(pData,pSheetOrFileName)   1. **Generic Guidelines**  * Do not use fnClick. RegisterUserFunc is now been used to redirect the Click method to call the fnClick internally. Earlier we use to use fnClick(<<Object to be clicked>>),   Eg: - fnClick(Browser("LogIn").Page("IntraLink Portal").link("My Profile"))  Now we can directly use the Click method on the object.  Eg: - Browser("LogIn").Page("IntraLink Portal").link("My Profile").Click  By following the above mentioned standards, methods and techniques I was able to successfully build a robust hybrid automation framework.  By building the automation framework successfully, I implemented the concepts of software designing, which in turn resulted in implementing the principles & concepts of engineering design.  **Example 2:**  **Situation –**  Build automation framework for OPTIMUS application in Citi using Java, Spring, Maven and Cucumber.  **Problem –**  Citi has built a new application called OPTIMUS which provides trading recommendations to clients based on the current market conditions. Since it was new application a framework needs to be built to carry out the automation testing of the application.  **Description –**  Optimus application provides a platform which generates recommendations to achieve best client order objectives. The platform evaluates multiple models, making use of analytics and signals and generate recommendations within acceptable timeframe. It is real time and based on the market data and corrective measures are implemented as required. It also provides transparency to client in every phase of trading life cycle (pre-trade, in-trade and post-trade).  It also enables clients to visualize order state and execution performance of trade in realtime.  **Hi-Level Logical Architecture Design**  -    **Following are the components of Optimus application –**   * Optimus Order Event Generator * Optimus Engine * Optimus Performance Monitor * Optimus Algo * Optimus Analytics Engine * Optimus Signal Engine * Optimus Client alerts Engine * Optimus Content Engine * Optimus Dashboard App   This is the high level overview of what Optimus application is. Now, let me walk you through the automation framework that I have built to accommodate the functionalities of components involved in Optimus application.  I started building the automation framework by using Spring, Maven, Cucumber and Java as programming language.  I followed the Modularity technique of software designing and created modules for services, databases (KDB services) and Kafka topics (topics used for sending and receiving fix messages in Investment banking domain) that are being used to integrate the components of Optimus application.  Modularity :-    Modularity: Created classes for services involved in Optimus  Cohesion :-  I followed the Cohesion technique of software designing and created step definitions which contains methods to initialize the data, generate models, create cache to read and store client messages, evaluate and validate the instructions and messages etc..    Coupling :-  I followed Coupling technique of software designing and created feature files which contains different create scenarios to validate/verify the functionalities by coupling/integrating the methods that are written in step definitions.  I used Cucumber’s Gherkins syntax of Given, When & Then to initialize the pre-requisites, specify the conditions and then validate the results against benchmark values.    Apart from above mentioned techniques, I also used internal procedures to specify the parameters required by each system (in .properties files), data required for executing the scenarios and many other forms of input data/parameters to setup the framework.    In this way, I was able to follow the software desiging techniques, research and analysis to support the design process, investigate the behaviour of Optimus system, develop first principles and construct mathematical, physical and conceptual models of situations, systems with a clear understanding and made in development of such models and apply appropriate design methods in solving complex engineering problems. | **Provide annotations to your supplementary evidence (document and page number)** |

| Element Six  **Knowledge of engineering practice in the engineering discipline** | |
| --- | --- |
| **Context**  Engineers require knowledge of a broad range of tools and techniques relating to technical (measurement, modelling, drawing, design), business (financial management, project management) and interpersonal (communications, teamwork) aspects of modern engineering practice.  Washington Accord graduates are expected to be able to:   * Create, select and apply appropriate techniques, resources, and modern engineering and IT tools, including prediction and modelling, to complex engineering problems, with an understanding of the limitations. * Apply knowledge of management principles and economic decision making as part of the management of engineering projects * Function effectively as an individual and as a member or leader in diverse teams * Communicate effectively with both technical and non-technical audiences | |
| Performance Indicators  Tools and technologies:   * Awareness of critical issues affecting current technical and professional practice * Awareness of current tools of analysis, simulation, visualisation, synthesis and design, particularly computer-based models and packages, and competence in the use of a representative selection of these * Appreciation of the accuracy and limitations of such tools and the assumptions inherent in their use * Knowledge of materials and resources relevant to the discipline and their main properties and ability to select appropriate materials and techniques for particular objectives * Knowledge of a wide range of laboratory procedures relevant to the discipline and a clear understanding of the principles and practices of laboratory safety * knowledge of current types of systems, equipment, information technology, and specifications that accomplish specific design objectives   Communication:   * write correspondence that clearly and concisely communicates facts and circumstances related to a project, product or process * plan, prepare and deliver an oral presentation, with appropriate visual aids and other supporting materials * communicate effectively with both technical and non-technical individuals and audiences   Engineering management principles and economic decision making:   * apply appropriate tools and techniques to monitor project schedules and costs   Team work:   * Operate as an effective team member or leader of a multidisciplinary team | |
| **Summarise your knowledge in each of these core areas underpinning engineering practice and how it was 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.  **Tools & Technologies**: I gained knowledge of automation tools from the fresher’s learning program which was a part of our learning program, when I started my career in Capgemini and during my professional experience as an automation analyst, automation developer and automation architect. During the program and in my career, I have worked in preparing automation suites using various kinds of automation tools like HP Quick test professional (which is now known as HP Unified Functional Tester), Test Complete, Selenium webdriver and many freeware automation tools.  During my course of work, I gradually realized that the tools and technologies are getting upgraded very rapidly and in order to be up to date with latest tools and technologies I have to keep myself updated with the advancements. Hence, I started exploring and learning new technologies to increase the knowledge about latest tools and technologies and be sync, whichever seemed relevant to my job profile.    ***Critical issues affecting current technical and professional practice*** :  Software testing is basically the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements. I have been working in software testing for past 8 years now and during my course of work, I have been experiencing different issues that have impact on the methodologies & technologies that I have been following in delivering quality products to clients.  Some of the issues that I have been experiencing in my practice are –  **Development & Testing Methodology**: Testing methodologies are approaches to testing, from unit testing through system testing, functional testing, non-functional testing and beyond till the product is live in production for customers. Some of the basic and common development methodologies that are being followed are Waterfall model, V model, Incremental model, Iterative model and agile model. Choosing right model for developing of the software product or application is very important. Based on the model the development and testing processes are carried out. Each methods listed above has their own advantages and disadvantages. So, it’s pretty much difficult for the organization to decide on which methodology to be followed. When compared to the methodologies from past 8 years, most of the organizations have shifted from waterfall model, V model to Agile model, quite obviously because of its short release cycles, quality of implementations etc… So, to conclude, deciding on the development & testing methodologies has always been a challenge and issue currently in most of the organizations.  **Test Coverage**: Test coverage measures the amount of testing performed by a set of tests in STLC. Due to frequent changes and amendments during release cycles, it is very hard to achieve 100% test coverage and that’s the reason we get production issues. I have been through such situations where I almost get to see 10% of production tickets (which is usually higher than expected) on the features that we deliver and this has been a challenge in most of the QA teams that I have been in and all actions are taken in advance to compensate the production issues.  **Performance Issues**: I have been monitoring the performance of applications very closely during my experience. Performance of an application depends upon the server responses, connectivity, interfaces, storage capacity and infrastructure. Due to rapid advancement in technologies, it is very hard for organizations to cope up with the latest infrastructure, which makes the performance of the applications poor. Despite of improvements, organizations lack in maintaining the best infrastructure.  There are several tools like Load runner, JMeter etc... that I used for measuring the performance of the applications/systems, but each one of them has their own limitations. So, maintaining consistent performance of application is always an issue.  **Communication:**  During my course of experience I have performed different roles like Software tester, automation developer, test lead, automation test analyst and automation test architect. In each of these roles, I have regularly interacted and coordinated with my peers, stake holders, on-site coordinators, business analysts, project managers and developers on technical discussions and get updates on preparing presentations for POC’s (Proof of Concepts), projects, requirements, deliverables, resolutions, statuses, risks & mitigation plans etc…  Being expertise in automation technologies, I was also involved in taking technical interviews of lateral entry candidates in our team for fresher level and mid-level positions. All these roles and jobs required me to communicate effectively and appropriately.  While working in various technical and team lead positions in organizations, I have also interacted with team members who are located in different countries like U.S, Italy, UK and Singapore, which helped me in improving my verbal communication skills. I have ability to interact with diverse groups across the globe.  Apart from the above mentioned points, I have also attended the class room and online trainings related to technology, delivery and practices in all the organizations that I have worked till now. Though these trainings are more focused the subject, it also increases the listening skills (1 of the component of communication skills).  **Team work**:  Team work is secret behind most of the successful projects. It is the most essential & crucial factor, when multiple people are working on product/project delivery.  I have always been a part of the team in all the organizations that I have worked till now. Though, sometimes tasks were confined only to me, I have coordinated, trained & helped my team members in terms of resolving complex problems, sharing the work load etc…I have contributed in so many projects and delivered them as a team.  I have also leaded a team size of 30, when I was employed in Capgemini. I was given this opportunity, because I was working in that team for past 3 years and became the subject matter expert (SME) for RSeCare application. Being able to resolve queries on the application, I was given extra responsibility to handle the team and deliver a compliance project called ADA Implementation (American Disability Act).    **Examples of practical implementation of Tools & Technologies, Communication and Team Work**  **Example 1:**  While working as an automation test architect in Citibank (Citicorp services), my role was to focus more on the architectural parts like designing automation frameworks using different languages, which gave me an exposure of designing models, applications and solutions which can interact with different systems and give the desired output with minimal manual intervention.  I started working on a project from scratch, where in the requirement was to build an automation model/solution that can interact with the citi trading systems internally in the form on fix messages and validate the tags that flows between the systems starting from client’s input to the algo engine to the execution engines (Stock Exchange) of the equity trades. Since the requirement was to interact between multiple systems (built in different technologies), I was pretty much challenging task to interpret the different forms of inputs and outputs and transform them into the language that is understood by the automation tool (TDM – Test Data Manager).  For example, the interaction between tibco servers, which gives the output in the form of fix messages and citismart application, which take the inputs in the form of Java objects, I have to create an interpreter that can covert the fix messages into Java packages and push it into citismart application.  Moving forward, I started creating an end to end model which can read data from client systems and validate each and every fix tag that flows from one system to another and finally create a user friendly report, which can portray the results in well-mannered way. An overview of the application and the model can be seen in the snapshot below.  Following steps were performed in building the model for Citismart application –   * Identify the systems involved in the end to end of the trade flow * Identify the common values that can be stored in “Environment Variables” for parameterization * Identify all the tags that flows as a part of fix messages between the systems * Build a mechanism to interact and then read the fix messages that flows in the tibco severs * Build a mechanism to interact and then read/update the records in KDB services (database) * Validate the values of each and every tag that flows between the systems. There are about 30 tags that flow between each system. * Build a mechanism to report the validations/assertions in a tabular format on a HTML file   **Tools & Technologies used:**   * Used TDM (Test Data manager) automation tool to collate the actions/functions created for each system, data that flows between the systems, Environment variables, executable scripts (.vbs) together. This tool is also used for execution of scenarios created by collating the actions together. * Used C# (C Sharp), VBS (Visual Basic Scripting), VBA (Visual Basic for Applications) programming languages to build the dynamic expressions and executable scripts. * Used HP ALM (HP Application Lifecycle Management) test management tool to maintain the test scenarios, execution cycles, dashboards and reports. * Used excel macros with VBA programming language to pump the data automatically in TDM * Used HTML 5, CSS & Javascript languages to create the interactive reports     Overview of report :    In order to build such a huge model and solution, I needed inputs from the functional team members like the functional behaviour of the applications, architectural structure of the applications, Data fornats required for execution, customized fix tags & their values, expected results from a particular flow etc.. In order to get these inputs I organized and attended the knowledge transfer sessions with the functional team on regular basis, which helped me in increasing my communication and coordination between the team menbers. Also, upon successful completion of automation model and solution, I have also given a presentation on the working model, usage, total cost saving in terms of effort and advantages of the solution to the entire functional team and management.  Working in a team gave me an exposure on challenges & mitigation techniques, learn new technologies functionally, got innovative ideas from other team members, got to learn and improve coordination skills and exchange ideas technically and non-technically.  In this way, I was able to communicate both technically and non-techincally and deliver a presentation, user guide, effective cost savings on the model built for citismart application. Also, this helped me in understanding the critical issues that affects the current technical projects, got exposure of the tools & technologies that needs to implemented as a part of this model/package, designed the automation solution to accommodate all the functional requirements, rewarded with start award for creating an outstanding and robust model, assessed and implemented the resources/inputs required for the project.  **Example 2:**  While working as a test lead in Capgemini in the year 2011, since I was a subject matter expert (SME) in the eCare team, I was given an opportunity to lead a team of 30 and deliver a critical compliance project for HSBC (client) credit card applications called ADA – Phase 2.  In ADA – Phase 2, my role was to handle a team of 30 and deliver RS eCare part of the HSBC credit card applications. My core roles and responsibilities were –   * Preparing daily delivery plan and allocating work to team members * Assisting team on queries related to functionalities. * Project planning and requirement understanding * Preparing estimates according to the requirements and scenario * Running client management calls during SIT & UAT phase of testing * Monitoring and maintaining deliverables in agreed time lines with client * Running defect management calls * Dealing with project management and resource management issues * Preparing project closure reports and metrics * Monitoring and managing end to end delivery of projects * *HSBC credit card application* – HSBC credit cards is broadly divided into 2 categories CSeCare and RSeCare.   CSeCare internet websites was developed for the Card Services Business Unit of a Leading North American Bank to cater the customer requirements for online access of their credit card accounts.  RSeCare internet websites, on the other hand, was developed for the retail services business unit to cater the customer requirements for online access of their retail card accounts provided by Over 200 Leading Retail Stores in America like Best Buy, GAP, CVS, Menards etc.   * *American Disability Act (ADA)* - In the year 2009, US government amended the Americans with Disabilities Act of 1990 (ADA) and other disability non-discrimination laws at the Federal level of the United States.   According to this law, all websites owned by the banks or organizations/firms in US should meet and be compliant with the WCAG guidelines, in order to prohibit discrimination against individuals with disabilities in all areas of public life, including jobs, schools, transportation, and all public and private places that are open to the general public.  Since all the HSBC north America websites, applications and infrastructure was maintained by Capgemini. We got a project to test all the HSBC credit card websites to be compliant with WCAG guidelines. The websites were made compliant and released into production.  This project covered functional testing, regression testing and accessibility testing using JAWS tool. I started working on identifying the flows of RSeCare application and that can be validated against WCAG guidelines and ask team members to work on accessibility testing and assigning tasks to team on daily basis and make sure that all functionalities are being covered and accessible using JAWS tool.  JAWS tool –  JAWS, Job Access With Speech, is the world's most popular screen reader, developed for computer users whose vision loss prevents them from seeing screen content or navigating with a mouse. JAWS provides speech and Braille output for the most popular computer applications on your PC.  JAWS users around the world sent us videos about the impact JAWS has made on their lives. We compiled these testimonials and more in a video to celebrate the 20th anniversary of JAWS for Windows.  Features -   * Two multi-lingual synthesizers: Eloquence and Vocalizer Expressive * Talking installation * Built-in free DAISY Player and full set of DAISY-formatted basic training books * Works with Microsoft Office, Internet Explorer, Firefox, and much more * Supports Windows® 8.1 and Windows 10, including touch screens and gestures * Support for MathML content presented in Internet Explorer that is rendered with MathJax * Fast information look-up at your fingertips with Research It * Convenient OCR feature provides access to the text of PDF documents, even those with scanned images that are reported as empty documents by screen readers * Save time with Skim Reading * The only Windows screen reader to provide contracted Braille input from your Braille keyboard * Fully compatible with MAGic, screen magnification software, and OpenBook, scanning and reading program   JAWS overview (look and feel) –    List of key strokes used for accessbility testing using JAWS tool –        In this way, I was able to communicate both technically and non-techincally with team members and understand the issues that affects the current technical projects. I also got exposure of new accessibility testing tools & technologies that was implemented. On successful delivery of the project, the entire team was awarded with the best team award of the year. | **Provide annotations to your supplementary evidence (document and page number)** |

|  |  |
| --- | --- |
| Element Seven  **Comprehension of the role of engineering in society and identified issues in engineering practice in the discipline: ethics and the professional responsibility of an engineer to public safety; the impacts of engineering activity: economic, social, cultural, environmental and sustainability** | |
| **Context**  Engineers design artefacts (facilities, structures, systems, products and processes) that are intended to meet a societal need, but which typically impact on individuals or groups in different ways. As a result, design and decision making processes must take account of often conflicting stakeholder needs. An understanding of this societal context and the ethical obligations that the engineer has in service of society are critical components of engineering practice.  Washington Accord graduates are expected to be able to:   * Apply ethical principles and commit to professional ethics and responsibilities and norms of engineering practice * Understand and evaluate the sustainability and impact of professional engineering work in the solution of complex engineering problems in societal and environmental contexts. | |
| Performance Indicators   * Demonstration of ethical behaviour in accordance with ethical codes of conduct and established norms of professional conduct * Evidence of making ethical decisions and regulating one’s own professional conduct in accordance with a relevant code of ethical conduct * Implementation of appropriate health and safety practices * Application of safe practices in laboratory, test and experimental procedures * Awareness of the social and environmental effects of their engineering activities * Awareness of sustainable technologies and sustainable development methodologies * Ability to identify risks as a consequence of engineering compromises made as a result of project or business constraints, and understanding of techniques to mitigate, eliminate or minimise risk * Knowledge of appropriate risk management techniques used to assess the accuracy, reliability and authenticity of information * Understanding of the role of quality management systems tools and processes | |
| **Summarise your knowledge of the role of engineering in society 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.  The role of Engineers & Engineering in society very much important and it is a key that drives to success and prosperity, software engineering is no exception to that.  Being an engineer and as a part of its community I have been following ethics during my entire course of professional experience.  To elaborate –   * I perform my roles and responsibilities with an objective, honestly and with complete diligence. * I accept full responsibility of my work. Built and work with software which have a well-founded belief that it is safe, meets specifications, passes appropriate tests, and does not diminish quality of life, diminish privacy or harm the environment. The ultimate effect of the work should be to the good. * I always restrict myself in getting engaged in all unfair activities and the activities that conflicts with the laws & policies of the organization or which causes harm to organization objectives. * I disclose matters to appropriate persons or authorities and always follow code of conduct. * I demonstrate everything relevantly and loyally in all aspects with all the organizations I have worked till now and will be continuing to do the same. * Cooperate in efforts to address matters or concerns caused by software, its installation, maintenance, support or documentation. * I always help in developing an organizational environment favourable to acting ethically. * I do not engage in any activities that are dishonourable to my profession or field of profession. * I do not accept any kind of gifts, bribes etc… from my colleagues or any employee, clients, customers, or business associated workers of the organization which impacts the integrity of the organization * I perform and take responsibilities of tasks that are reasonable and expect my skills for completing the job. I always aim to bring improvement in the organization by performing and delivering the tasks with quality. * I always be compliant with rules & policies that are defined in the organization and never use the confidential information for personal gains in any manner. * I practice my profession with quality, dedication, high standards of competency, morality and dignity. I always improve my skills and be updated with latest technologies by researching, attending trainings and by studying materials which are present on internet or in books. * I always believe in sharing knowledge and do the same by conducting trainings for colleagues in the organization. I also give external trainings in private organizations. * Temper all technical judgments by the need to support and maintain human values.   **Examples of ethical behaviour in accordance with ethical codes of conduct**  Code of conduct is a set of regulations that defines the social norms and rules and responsibilities of, or proper practices for, an individual, party or organization. Related concepts include ethical, honour, moral codes and religious laws.  In every organization that I have worked till now, I have gone through, signed and agreed to the code of conduct outlined by the organization. I have always adhered to them strictly and also ensured that team members, colleagues and everyone in the organization is following the same.  For Example, one of the points in code of conduct of Citibank (Citicorp) states that, all employees should take prior approvals before buying/selling the trades or shares, because I work in Citibank’s Investment banking domain and as per the norms, I should not use any citi’s confidential idea/practice in doing trading for self, hence the organization make sure and checks that I am not using their techniques in doing personal trading. I have never traded without taking approvals from authorities of Citibank.  **Examples of making ethical decisions**  In every organization that I have worked till now, downloading and installing unauthorized software’s for carrying out the testing activities is restricted, though the organization doesn’t block the systems to download the software’s, it is unethical to do such activities.  When I was employed in Vodafone shared services India, the security systems or firewalls were not strong enough to block the downloading of unethical software setups. Since I was working as an automation test analyst, I needed so many different automation tools to check the feasibility of creating automation suites and to carry out a proof of concepts (POC’s).  I needed Selenium web drivers automation tools to carry out the POC. My manager suggested me to download the software drivers on my own since it is a freeware, which violates the software’s policies of Vodafone. I suggested my manager that I will not violate the policy and I will follow the ethical process of getting the software. Though the processes take more time, I followed the process and raised an incident/ticket and assigned it to IT helpdesk for downloading/installation instead of downloading it directly.  **Examples of Implementation of appropriate health and safety practices**  As an automation test analyst/architect, my job is not only to prepare automation framework and suites; It’s also my responsibility to maintain the repository of these suites in order to res use the code in other projects or OPCO’s as per the requirement.  I use the official version controlling tools like SVN, Git etc…to maintain the code base and share it with the teams, managers & stakeholders.  While maintaining these repositories, I always follow the good practices of version controlling –   * Always check out the latest version before working on the new changes * Commit often because it is easier to read and debug * Make a note of changes before committing the code * Avoid using locks on files * Follow structured hierarchy and braches in maintaining the code of modules * Always put comments and tags in each commit * Always use versioning in documents & user guides   ***Autmation developers working model*** -    ***Version controlling struture*** –    **Examples of awareness of the social and environmental effects in software engineering**  Being in software industry and working in large operational development centres (ODC’s) it is very necessary to understand the environmental effects caused by the equipments and resources used in the offices. Also, it is very important to maintain the healthy surroundings.  For Example, most of the ODC’s have smoking zones in and around the premises, which has adverse effects on the environment and the people in the ODC’s. As a intiative to protect environment and being a volunteer in operations, I have raised requests to infrastruture management to put No-Smoking boards with in 100 meters radius of ODC’s, so that we can maintain healthy and good environment in the offices.  As an another initiative to go green, I as a volunteer in operations keep on sending reminder emails to colleagues in Citibank and make them aware of usage of papers and effects of using papers in large amounts. I always keep on putting suggestions for avoiding papers and increase the usage ebooks, soft copies etc..  **Examples of sustainable technologies and sustainable development methodologies**  Due to increased adoption of IT, the usage of hardwares like computers, servers, monitors, printers, storage devices and networking and commujnicatios systems is also increasing. Due to this, there is a high rise in the energy consumptions. All forms of hardware systems are controlled by software components, although software systems don’t consume energy directly, they affect hardware utilization, leading to indirect energy consumption. Therefore, it’s important for software engineer to optimize the energy consumption. Most of the organizations has started adopting the initiatives like “Green IT”, which is more focused on product and process efficiency, in terms of environmental sustainability, as well as applying IT to create energy-efficient, environmentally sustainable business processes and practices.  Sustainable development in software engineering is a principle and set of practices that enable us to achieve and maintain an optimal development pace indefinitely. It’s about efficiency and balancing the needs of the short and long term requirements of the customers or stakeholders (business) while reducing environmental impacts. Some of such sustainable development methodologies that I have been working on are Agile software development, applications hosted on cloud, Virtual desktops, Open-source technologies etc.. These technologies are designed by considering environmental, ethical, social, political, cultural and economic aspects of targeted community. These technologies are harness the power of distributed peer review and transparency of IT processes.  **Examples of risks and mitigation plans in the project**  During my course of experience as automation test analyst/architect, all the projects that I have worked till now have risks in one or the other forms. While identifying the risks, I also make sure that I propose an appropriate mitigation plan against each risk in the same context.  Some of the common risks in the projects that I work are –   * Software applications unavailability * Servers going down * QA environment unavailability * Resources unavailability * Tools unavailability * Unavailability of skilled resources * Cost/budget for building automation suites * Applications instability   When I was employed in Capgemini and playing a role of test lead. I have to manage a team of 30 resources and make sure that the delivery of the project is not getting impacted despite of issues or risks both technically and non-technically.  Being a test lead, it was my responsibility to highlight the risks to stakeholders in advance and also give a set of mitigation plans against each risk, making sure that there is very minimal effect on business.  Let me list down some of the risks that I raised and mitigation plans I proposed while leading a team of 30 and delivering the ADA compliance project –   |  |  | | --- | --- | | **Risks** | **Mitigation Plans** | | **Level 0 estimates for the project was proposed considering 5% of downtime. Increase in downtime may affect the project delivery timelines and there was a huge risk, impacting the production implementation.** | In case the downtime exceeds 5% (as per the level 0 estimate), I may buy in extra resources or go for risk based testing methodology to incorporate the timelines. | | **Since, it is a compliance project; I had very tight timelines to deliver the project. If the testing is delayed due to unavailability of application changes in QA environment, as per the scheduled availability date, the test cannot be extended beyond the UAT scheduled end date** | In case the application is not available as per the schedule, we may either put extra efforts by extending the working hours or reduce the regression cycle to incorporate the testing of new implementations. | | **Not enough resources or resources going on unplanned leaves.** | In case of unplanned resources unavailability, we may borrow resources from other departments (CSeCare etc..) and try to complete the QA cycle as per the schedule. | | **Defects are found at a late stage of the cycle or at a late cycle; defects discovered late are most likely be due to unclear specifications and are time consuming to resolve.** | In case we find low impacting defects late in the QA cycle, we have to move those defects to next cycle or test it in pre-production environment, assuming development teams will fix them in the pre-production environment. | | **Scope not defined completely** | In case the requirement or scope changes in middle of QA cycle, we will have to go for a retrospective testing. | | **Natural disasters** | Business continuity plans (BCP) will be invoked and we will follow the guidelines outlined in it. | | **Non-availability of Independent Test environment and accessibility** | In case any of the QA environments is not available during QA cycle, we will start testing in development environment, assuming that the QA environment will be in sync with the dev environment. | | **Delayed testing due to new issues** | In case there are unexpected issues in the middle of QA cycle, we will go for Risk based testing methodology. |   **Examples of risk management techniques**  Risk is an uncertainty. It is an unpredicted loss or a problem that may or may not occur while managing a project. It is generally caused due to insufficient information, lack of control over activities or timelines. The possibility of suffering a loss and handling those using defined procedures in software testing process is called Risk management.  Risk management process consists of 6 basic steps –   1. Identify: Search for the risks before they create a major problem 2. Analyze: understand the nature, kind of risk and gather information about the risk. 3. Plan: convert them into actions and implement them. 4. Track: we need to monitor the necessary actions. 5. Control: Correct the deviation and make any necessary amendments. 6. Communicate: Discuss about the emerging risks and the current risks and the plans to be undertaken.   pic1  Snapshot of some of the practical risks & challenges that I have identified/encountered while working as an individual contributor and delivering an automation project in Citicorp are –      **Examples of quality management systems tools and processes**  A Quality management system (QMS) is a formalized way of organizing resources like processes, procedures, duties and responsibilities in order to achieve policies and objectives of the projects with quality. It helps the team and the organization in satisfying the needs and requirements of the clients/customers. It also improves the efficiency of the projects because the entire team follows a same set of the rules/procedures/processes in the organization, leaving no chances of errors and achieving quality work.  QMS tools are used to achieve the objectives outlined in the above paragraph. It integrates the systems in an organization to achieve the quality and make sure’s that corrective & preventive actions are being taken, there are no deviations, no non-compliant actions are being performed, incidents are being raised against the production tickets or issues in QA cycle, time management, cross functional trainings are being attended etc.. It also ensures delivery of quality products by following the set of procedures and accelerates the time taken to deliver the product in market for customers.  QMS basically focuses on customer needs, customer satisfaction, procedures, processes, adherence and organizational model, which in-turn helps the organization to focus on the activities and its goals in more formalized way. The techniques followed in QMS leads to high level of customer satisfaction because the organizations also makes sure that the product is being delivered to customers consistently in small cycles and covers all the requirements of clients.    Some of the quality management tools that I have used in the organizations I have worked till now like were hybrid web hosted set of procedures with multiple quality checks, which use to ensures that the product is meeting all the quality standards. Most of the organizations that I have worked with are six sigma certified or they achieved some of the ISO levels by proving the quality of products delivered till now.  I being an automation test analyst, I have a different set of rules to ensure the quality of code we deliver like peer reviews, code coverage analysis, matrices to trace the requirements with the code written. Some of the tools that I have worked (like HP Application Lifecycle management, JIRA, LISA etc..) maintains these records or evidences for audits and compliances. | **Provide annotations to your supplementary evidence (document and page number)** |

|  |  |
| --- | --- |
| Element Eight  **Engagement with selected knowledge in the research literature of the discipline** | |
| **Context**  Research and broader lifelong learning capabilities are essential if the engineer is to remain up-to-date with rapidly evolving scientific knowledge, technology and engineering tools critical to engineering practice  Washington Accord graduates are expected to be able to use research-based knowledge and research methods as part of the investigation of complex problems in their discipline | |
| Performance Indicators   * Advanced knowledge in at least one area within your discipline, to a level that engages with current developments in that area * Understanding of how new developments relate to established theory and practice and to other disciplines with which they interact * Describe advancements in engineering research and technology and science in a particular area of engineering practice; * Review research articles pertaining to a project component typically encountered in a specific area of engineering design; * Choose topics most appropriate for continuing education to increase depth of technical knowledge pertinent to the specific area of engineering practice * Commitment to lifelong learning. | |
| **Summarise your research 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.  **Examples of advance knowledge of software automation tools and methodologies**  There is a huge shift in software development and testing methodologies in years. When I started my career in Capgemini in the year 2008, most of the companies use to Opt for the waterfall model to progress on development and testing activities and so do I used to work on traditional automation tools and methodologies where the outcome was expected in a longer time and the time duration was more.  8 years down the line, I used to work on traditional automation tools and methodologies like HP QTP/UFT tool and hybrid automation frameworks where the automation development activities needed larger window to accommodate the changes in the application that goes as a part of production release (following waterfall model).  As the years passed on, due to advancement in software technologies, ideas and due to high expectation from customers, the methodologies kept on changing to achieve best results in the market. Most of the organizations started opting for agile methodology because they want to deliver their product in short cycles and with quality. Due to this shift, the automation methodologies and techniques also demanded high output in a shorter cycle and in a better way.  Since there are so many open source automations tools like Selenium web driver, which are available in market for free now and due to new inventions in the programming languages and new ideas of implementing automation in a simpler way, many organizations started shifting from tools like HP QTP/UFT to tools like Selenium webriver etc.., so do I.  In all the automation proof of concepts (POC’s) that I have proposed in last 2 years are all based on using advanced automation tools, techniques and programming languages (like Java etc..)  Let me give an example of the latest POC that I have proposed using the advance knowledge of software tools and methodologies I possess, which was later accepted and I implemented it in building the automation suites –  Citi has built a new application called OPTIMUS which provides trading recommendations to clients based on the current market conditions. Since it was new application a framework needs to be built to carry out the automation testing of the application.  Optimus application provides a platform which generates recommendations to achieve best client order objectives. The platform evaluates multiple models, making use of analytics and signals and generates recommendations within acceptable timeframe. It is real time and based on the market data and corrective measures are implemented as required. It also provides transparency to client in every phase of trading life cycle (pre-trade, in-trade and post-trade).  **Hi-Level Logical Architecture Design**  -    **Following are the components of Optimus application –**   * Optimus Order Event Generator * Optimus Engine * Optimus Performance Monitor * Optimus Algo * Optimus Analytics Engine * Optimus Signal Engine * Optimus Client alerts Engine * Optimus Content Engine * Optimus Dashboard App   I started working on preparing the POC for building the automation framework by using Spring, Maven, Cucumber and Java as programming language.  I followed all the knowledge of advanced automation techniques, technologies and tools that I improved on my day to day job throughout my experience.  In order to cope up with the latest technologies the application was built in, I created the framework using the advanced technologies and tools. I followed these best practices to make it results in a robust & excellent automation suite –   * Determined detailed scope of Automation before the start of the project. * Selected the right automation tool: I selected Selenium automation tool with Gherkins which best fits to the automation requirements. * Opted for Cucumber framework * Followed scripting standards while creating the scripts. Some of them are -   + Create uniform scripts, comments and indentation of the code   + Adequate Exception handling - How error is handled on system failure or unexpected behaviour of the application.   + User defined messages should be coded or standardized for Error Logging for testers to understand. * Captured following metrics-   + Percent of defects found   + Time required for automation testing for each and every release cycle   + Minimal Time taken for release   + Customer satisfaction Index   + Productivity improvement     Code Samples -    In this way, I was able to implement the advanced knowledge that I had/have about automation testing/development in building the automation frameworks & suites in my current and past 2 organizations.  **Examples of New Development related to Established theory**  **And**  **Examples of advancement in tools and technologies**  Quality of the software is dependent upon the technology and methodology used in building it. “Waterfall” or “V-Model” is one of the development methodologies, in which the scope of requirement, development, testing, implementation and maintenance occurs sequentially. This effects the quality of the software because the testing is carried out after the development activities has been completed and since more defects are found in the later stage of the project, the developers have to revisit the code of the application again and again, which in turn increases the effort, time and cost of the project.  When a software is delivered using agile methodology, the quality of the software is more better than the product which is delivered using waterfall model, because agile is more convenient, flexible and changes are pushed continuously in short cycles/packages. When the testing is carried for a smaller piece of code and once the defects of that particular sprint are fixed, the small chunk is pushed into a stack and when the stack is completed, that particular feature is pushed into production. The collaboration between developers, business analysts, project managers and quality assurance team nurtures the product to be the best.  So the development in software development methodologies also increased the quality of the product.  Since all methodology has pros and cons, one of them with agile is carrying out automation in agile methodology. Being an automation test automation architect, it is a bit challenging because test automation development takes lot of time and carrying out test automation in the same pace is sometimes difficult is to me, but I handle these challenges by following the established frameworks that can be used in agile automation activities instead of reinventing the wheel like Cucumber, Maven etc…  In this way, I have understood how new developments in software development methodologies can be linked/related to test automation activities in software engineering.  This is how I can relate the advancement of technologies in software engineering.      Being an automation test architect, I have also seen how automation tools have been evolved. For example HP QTP tool, which was just used for automation of user interfaces in the year 2008, which has now evolved (renamed as HP UFT tool) and can be used to create automation suites for user interfaces, services, middleware, databases and many more.  **Examples of research in my career**  Being in engineering profession, I have always involved myself into research activities related to software automation. I took part in a research on new software automation technologies in the Indian Testing League organized by UNICOM in Bangalore city (India) in the year 2013.  *Certificate of participation in Indian Testing League at WORLD CONFERENCE NEXT GENRATION TESTING*  During this league, I was a part of the team constituting of 3 members and I was doing research on how software automation enhances and improves the delivery of the projects. I was given a sample requirement and I started evaluating on the basis of estimates, efforts, ROI (Return on Investment), cost, tools and technologies and presented the thesis of the research to jury in the semi-finals of the league.  I also prepared a presentation on the research I did on the CMMI levels of the HSBC projects in Capgemini. For which, I was awarded a certificate for implementing the CMMI level 5 regulations in the team for eBusiness Testing project in the year 2008.  Apart from above mentioned research, since I am involved in preparing automation frameworks and suites from scratch. My day to day job is to do research on applications, automation tools and technologies and then implement them in the projects.  I have also studied various research papers related to Software automation and I have gained knowledge and understanding of practices and advancement made in methodologies. I got a lot of benefits in terms of new ideas for implementing automation in projects and experience in practicing the automation techniques and in designing the frameworks from scratch.  Some of the software automation research papers that helped in building automation frameworks and suites in my career are –   * Benefits and Limitations of Automated Software Testing: Systematic Literature Review and Practitioner Survey by Mika V. Mantyla * Software Test Automation - Approach on evaluating test automation tools by Tarik Sheth and Dr. Santosh Kumar Singh * Research on GUI-based Automation Test Technology Driven by Separated Definition Data by Zhenyu Liu, Qiang Chen and Lizhi Cai * An Agent-Based Framework for Automated Testing of Web-Based Systems by Samad Paydar and Mohsen Kahani * A Method for Automated Program Code Testing by Sigitas DRASUTIS, Vida MOTEKAITYT E and Algirdas NOREIKA * Implementation of Selenium with JUNIT and Test-Ng by Deepti Gaur and Dr. Rajender Singh Chhillar * Test Automation Using Cucumber and Selenium WebDriver by Yadvinder Singh Pannu * Agile Test Automation by James Bach * Selenium Keyword Driven Automation Testing Framework by Sherry Singla and Harpreet Kaur * Automating Test Automation by Suresh Thummalapenta, Saurabh Sinha, Nimit Singhania and Satish Chandra * A Methodology and Architecture for Automated Software testing by Ilan Gronau, Alan Hartman, Andrei Kirshin, Kenneth Nagin and Sergey Olvovsky.   And many more..  **Topics for continuing education to increase depth of technical knowledge**  Automated software testing is a process in which software tools execute pre-scripted tests on a software application before it is released into production. The objective of automated testing is to simplify as much of the testing effort as possible with a minimum set of scripts. If unit testing consumes a large percentage of a quality assurance (QA) team's resources, for example, then this process might be a good candidate for automation. Automated testing tools are capable of executing tests, reporting outcomes and comparing results with earlier test runs. Tests carried out with these tools can be run repeatedly, at any time of day.  The method or process being used to implement automation is called a test automation framework. Several frameworks have been implemented over the years by commercial vendors and testing organizations. Automating tests with commercial off-the-shelf (COTS) or open source software can be complicated, however, because they almost always require customization. In many organizations, automation is only implemented when it has been determined that the manual testing program is not meeting expectations and it is not possible to bring in more human testers.  The reason I have chosen software testing practices in automation is because I have interest in excelling in it and for my continuous learning & growth. I will always continue learning the automation techniques to gain in-depth knowledge in the area of my specialized engineering field. It will also help me in adapting the software automation culture and implementing new practices related to automation testing.  **Commitment to lifelong learning**  I have been and I am committed to learn the new technologies not only related to automation but all programming languages and techniques including my current practice. I will continue further study and research in software engineering fields to make sure that I grow with the advancement in technologies. I have been doing this since I have joined engineering discipline in college and from the start of my career and would like to pursue the same in future.  I have always attended class room and online trainings related to my field of work throughout my career. I have done the following trainings in my career –  ***TRAININGS & SEMINARS:***   |  |  | | --- | --- | | **Technical Programs -**  - Mercury Quick Test Professional  - Cloud testing  - Selenium Web Driver  - C, C#, C++ programming  - Java programming  - Automation tools | **Management and Leadership Programs -**  - Emerging Leaders Training  - Risk Based Project Management  - Project Management Professional – PMP  - Consulting with Clients  - Values Workshop | | **Industrial Programs -**  - Credit Cards 101  - Lending 101  - Insurance 101  - Banking 101  - Securities and Investments 101 |  |   As a result of the trainings and seminars I have attended, I possess and gained the following skills related to software engineering –   |  |  | | --- | --- | | **Domains** | Core Banking, Credit Cards, Lending & Mortgage | | **Operating Systems** | Windows, MACINTOSH OS, DOS, Ubuntu Linux | | **Integrated Development Environments** | Eclipse Juno, Intellij IDEA, Microsoft Visual Studio 2015 | | **Build & Version control tools** | GIT, JENKINS, SVN | | **Environment** | Web Based – Internet & Intranet Sites  Mainframe –Vision Plus Modules(CMS, CDM, FAS, LMS, TRAMS, LTS), TSYS | | **Database** | SQL, IBM DB2, MS SQL, Oracle Database, Toad, MySQL | | **Test Management Tools** | Quality Center 9.2, 10.0, SDLC Tool, HP ALM 11.5, JIRA, RALLY | | **Test Automation Tools** | HP Quick Test Professional (QTP 9.2,10.0, UFT 11.0, 11.5), Selenium IDE/RC/WEBDRIVER, Excel VBA macros, Quality Center OTA Development | | **Programming Languages** | C, C ++, OOPS concepts, VBS, VBA, PHP, PowerShell, TCL/Expect, Python, Shell Scripting, Perl, C#, Java (Maven, Cucumber – Gherkins), JUnit | | **Other** | Knowledge on Dispute Processing Systems – Chordiant, Usability Testing and Validation using Web Developer Plug-In, Testing Mobile Applications and Involved in middleware services testing using WSDL’s through SOAP UI tool. |   I am also planning to learn and pursue the following skills related to software engineering in future –   * Mobile Automation Testing * Cloud automation * Performance automation testing * Automation testing on applications build on different technologies * Core Java * C# .NET for building automation tools * UDD – UI Driven Development and more.. | **Provide annotations to your supplementary evidence (document and page number)** |

**Section Three - Evidence of Application of Knowledge**

In this section you are required to provide evidence of the application of your engineering knowledge using 3-4 engineering projects or activities (Work/Study Episodes) that you have been involved with.

Provide a general overview of the scope or parameters of each project or activity, your role in it and the particular challenges or complexities involved. Then describe, in narrative form, how it provides evidence of the application of different aspects of your engineering knowledge. Cross reference to the relevant elements of the knowledge profile in the right hand column.

You are also required to complete the Knowledge Matrix to summarise the contribution to knowledge demonstration made by each project. The work/study episodes are expected to provide at least 2 examples of the application of each knowledge element.

|  |  |
| --- | --- |
| **Work/Study Episode 1**  **(April 2009 – May 2010)**  **Overview of the project**  HSBC credit cards services are broadly divided into 2 categories CSeCare and RSeCare.  Credit Card Services Internet Websites have been developed for the Card Services Business Unit of a Leading North American Bank to cater the customer requirements for online access of their credit card accounts. User Acceptance Testing Phase of this project involves end to end functional testing and validation of all enhancements including marketing content validation, compliance validation and regression testing.  Retail Services Internet Websites have been developed for the Retail Services Business Unit of a Leading North American Bank to cater the customer requirements for online access of their Retail Card Accounts provided by Over 200 Leading Retail Stores in America like Best Buy, GAP, CVS, Menards etc., System Integration Testing Phase of this project involves end to end functional testing of enhancements involving integration with multiple applications viz., Mainframe Backend, IBM DB2 Database, and MQ Series.  The project was to create automation suites for CSeCare and RSeCare functionalities which included following systems and technologies  **Automation tool** –  HP QTP 9.2 with Terminal Emulator, Web and ActiveX addins in integration with HP Quality Centre 9.2  **Systems** –  Web based application  Oracle Database  Vision plus mainframe systems with modules-   * CMS (Credit management system) * CDM (Credit decision management) * LMS (Loyalty management system)   **Your role and responsibilities**   * Understand the requirement and feasibility of automating the flows * Prepare estimates for the automation project, based on complexity of the flows * Prepare plan of automation code delivery * Analysis of functional flows for creating common functions * Coding using VB scripting, C# and QTP automation tool and self-designed hybrid framework (keyword driven & data driven) * Dry run, Integration testing and Unit testing of the code * Delivering code as per the plan * Code fixing as per the defects logged by manual functional team * Preparing closure reports showing the savings in terms of time and cost   **Complexities (using the complexity definitions) and challenges of the project**  This was my first automation project in my career. Being a fresher and individual contributor in automation testing, it was very much challenging for me to implement the niche practices in automation. I had to learn new automation techniques & technologies on my own without any mentor.  When I released the 1st draft version of the automation suite, there were so many bugs and the code was not so efficient enough to handle the complex scenarios. I struggled in fixing the issues in this project. Eventually, I gained knowledge about latest automation techniques by exploring e-books on internet, by attending global learning programs in the organization and by practical hurdles.  **How does this project demonstrate application of your engineering knowledge?**  As a fresher in automation testing it was very much challenging to follow the engineering fundamentals and implement the engineering principles, design and architecture in this project. The plan & overall delivery of the project demanded core implementation by meeting the engineering standards.  Following steps were taken in implementing the automation solution –   1. **Understanding the requirements, systems and technologies the system were built in, to validate the feasibility of automation** –   Customer database  CSeCare home screen  WebServices  Authentication  Customers Profile screen  Add/Remove Auth Users  Servicing database  Update Personal Info  Credit Line Increase  Manage Profile  Payment Services  Report Lost/Stolen  Dispute Transactions  Balance Transfer  Statements    Special Offers  LMS (Loyalty Management System)  Mainframe Systems  CMS (Credit Management System)  CDM (Credit Decision Management)  Web   * CSecare system was built using 3-Tier architecture. * The entire CSeCare system of HSBC was hosted using a web application. * These web pages were integrated with web services for transmission of data from other systems. * Webservices were built for integrating the flows and calling appropriate API’s * Oracle databases were used to store the customer and servicing data.   Technologies Used – Java, Web (HTML, CSS, Javascript), ActiveX components, Terminal Emulators, Databases  I used my functional knowledge of Data communication and software communication to understand the connectivity between systems in the architecture and assess the feasibility of automation**.**   1. **Preparing estimates & automation plan based on complexity of functionalities** -   I use to prepare estimates based on the historical data and analysis. The functionalities were bifurcated and classified based on their complexity (Complex, Medium & Simple). Fixed man days were decided for each complexity level and the estimates were rolled out in three stages -     1. Level 0 2. Level 1 3. Level 2   Functional Classification –    Factors that affects test automation estimation –    Sample estimates –    In this way I was able to implement engineering principles by selecting appropriate resourcing techniques in modern engineering and applied knowledge of management principles and economic decision.   1. **Analysis of functional flows for creating common functions** –   Based on the functional flows, I used to identify the systems which are common among most of the functionalities and create function files (.vbs or .qfl) to store the common functions. These common functions can be called across different functionalities based on the flow, by doing this I used to reduce the redundancy of automation code, making the automation suite more robust in terms of design and architecture.  For Example –  In most of the flows, the Login component is common because all the manage services functionalities can be accessed only if the customer logs into the Customer service website. Hence, I will create “Login.vbs” which will be called in all the manage services flows mentioned below –   * Add/Remove Authorized users * Update Personal Info * Credit Line Increase * Payment Services * Report Lost/Stolen * Dispute transactions * Balance Transfer * Statements * Special Offers   I used the concepts of software designing (engineering design) like Modularity and Coupling by implementing the concepts of common functions in automation suite development. Hence I applied appropriate design methods in solving complex engineering problems in the automation suite**.**   1. **Automation development using VB scripting, C# and QTP automation tool and self-designed hybrid framework** –   For building an automation suite, the primary thing that a developer needs is framework. Hence before starting any automation suite, I build a hybrid framework (data driven and keyword driven) making sure that it can accommodate all the requirements of system, business and clients. By building the framework, I also have an end to end understanding of schema of the systems, behaviour and functionalities.  While building automation suite for HSBC credit cards, I made sure that I followed these characteristics for making it an ideal one –   * Portable, extendable and reusable across and within projects. * Ease of functionality Plug-ins/outs based on application version changes. * Loosely coupled with the test tool wherever possible. * Extended recovery system and exception handling to capture the unhandled errors and to run smoothly. * Step, Log and Error Information provide easier debugging and customized reporting facilities of scripts. * Ease of test data driven to the scripts and they need to be loosely coupled. * Easily controllable and configurable test scope for every test run. * Simple and easy integration of test automation components with test management, defect tracking and configuration management tools.   I built automation suite using VB scripting and by following concepts of software development to achieve stability, polymorphism, encapsulation, inheritance, modularity and coupling of appropriate resources etc... By following these development techniques I was able to build a stable automation pack.  Some of the functionalities of CSeCare like Credit Line Increase, Payment Services etc.. and code logics to implement wait times for synchronizing automation tool with application needs mathematical and algebra calculations.  For Example –   1. If a customer needs increase in Credit Limit on their credit card, the customer has to login into the HSBC credit cards website and use Credit Line Increase functionality to request the same. Since the functionality was built to give instant decisions (approved, declined & pending). The decisions were made logically on the basis of mathematical calculations done by considering factors like current credit limit, duration of credit card usage, cibil scores etc.. Hence, In order to simulate such scenarios in QA environment, I used to do calculations by implementing logics in automation suite. 2. In order to synchronize the buffering speed of browsers/applications with automation tool, I used the mathematical calculations to loop the code and wait until the object appears on the application. For..Next Loops are generally used in VB scripting to synchronize the automation tool with application under test.   In this way, I used the concepts of software development like stability, polymorphism, encapsulation, inheritance, modularity and coupling to build the automation suite and evident that I have specialist engineering knowledge of automation development and understanding of systems involved in it to solve complex engineering problems.  AND  I used concepts of physical sciences in judging the hardware components required for building automation frameworks and suites.  AND  I used mathematical and algebra calculation in automation suite to handle the part of logics and the simulate the behaviour of the functionalities using automation.  In delivering this automation suite I have proactively taken responsibilities of completing the task and delivered all the modules of CSeCare and RSeCare honestly and with complete diligence. I was also appreciated for maintaining the code quality and reducing the cost that used to occurred for executing the regression suite manually.  Upon successful implementation of automation suite, I have given walkthrough to clients, onsite-coordinators, team members & managers and demonstrated the usage, advantages and limitations of the automation framework and suite. A complete presentation of this suite was carried out on WebEx sessions, where I projected the savings that can be made by using this automation suite. By implementing this automation suite, I was able to save 70k$ per year for HSBC.  All the automation tools and technologies that I have used in preparing this automation suite were licensed and authentic, ensuring safe practices of experimental procedures.  In every phase of development, I used to ask my peers to carry out peer reviews, brain storming sessions, knowledge transfer sessions to improve the quality of the automation suites. All the procedures that were carried out in preparing this automation suite followed the quality management processes as well.  In this way, I used and followed the ethical behaviour, sustainable development methodologies and quality management systems tools and processes. | **Element**  **Engineering knowledge referred in Element 3**  **Engineering discipline referred in Element 6**  **Engineering design referred in Element 5**  **Element 4**  **Element 1**  **Element 2**  **Element 7** |
| **Work/Study Episode 2**  **(Jan 2016 - Present)**  **Overview of the project**  Citigroup Inc. or Citi is an American multinational investment banking and financial services corporation headquartered in Manhattan, New York City. Citigroup was formed from one of the world's largest mergers in history by combining the banking giant Citicorp and financial conglomerate Travellers. It is the third largest bank holding company in the US by assets. Citigroup has its services all across the world majorly classified into 3 regions APAC (Asia Pacific), EMEA (Europe & Middle East) and NAM (North America).  I work for the Institutional Clients Group (ICG’s) as an automation test architect in all 3 regions mentioned above. One of the project that I am currently working on is -  **Create end to end automation solution for Optimus application. This project consists of 2 parts -**   1. **Develop integration tests for Optimus components using Java.** 2. **Develop end to UAT tests for QA using TDM (Test Data Manager) automation tool.**   **Your role and responsibilities**   * Understand the requirements and assessed the feasibility of creating the automation suite. * Create modules (classes) for each components of Optimus application using Java programming language for integration tests * Build feature files for each component under test using Cucumber framework with Gherkins language. * Use TDM tool to build the end to end automation flow integrating CitiSmart application. * Handle errors using exception handling techniques * Prepare execution reports showing results of integration and UAT automation tests * Participate in daily scrum calls and provide an update on the automation tests   **Complexities (using the complexity definitions) and challenges of the project**  Integration tests are built using cucumber framework and it interacts with several Java classes of each Optimus module in order to validate the order flow and recommendations. Since, I working with Cucumber and Gherkins for the first time, it was very much challenging for me to understand the whole structure of the framework and find a link between the classes, input data and output data.  It was very important for me to understand the API’s of each Optimus component because I cannot create feature file until and unless I get to know the behaviour of each API’s.  Eventually, by attending walkthrough sessions with developers of Optimus application, I started grasping the behaviour and created the feature files, though it was very much challenging task.  **How does this project demonstrate application of your engineering knowledge?**  Optimus application provides a platform for the Citi customers to get recommendations based on the order input parameters received from the customers. It generates recommendations to serve client objectives and to provide best services in terms of order executions based on the historical data and analytics available from Citi. The platform evaluates multiple models, uses analytics and generates recommendations within acceptable timeframe. It is real time and based on the market data. It also provides transparency to client in every phase of trading life cycle (pre-trade, trade and post-trade). It also enables clients to visualize order state and execution performance of trade in real-time.  **Hi-Level Logical Architecture Design**  -    Optimus is built using the latest technologies of real time interactions using Java, Kafa topics, Redis Cache, KDB services and multi-level architecture. It is a bridging engine between the client facing applications and the algorithm engines of Citi trading entity.  Being new to Investment banking and project, I grasped the technologies very quickly showing my intention and dedication towards learning the advanced technologies and implementing them in this project.   1. **Develop integration tests for Optimus components using Java and Cucumber.**   The Optimus Integration Framework is an approach that brings together Optimus components and wires them to communicate with each other. The integration framework tests the end to end interaction of components by controlling inputs, injecting controllable models and validating outputs. The framework utilizes Behaviour Driven Development methods.  Behavior-driven Development (BDD) is a combination of Test-driven Development (TDD) and Domain-driven Design which can be used by both business and software development users. Gherkins is a business readable language that lets you describe behavior without specifying implementation details. Cucumber is the tool that executes the plain-text functional descriptions as automated tests. All tests are executed regularly using Continuous Integration (CI) software. The result is a suite of components that are tested after every change and have business involvement in writing testing scenario.  After careful understanding of applications, I started creating the feature files using the cucumber framework, wherein I again followed the concepts of software development like integration, polymorphism, inheritance, cohesion. I also used these development concepts in building the class files to pull the data from KDB services and use them in the integration tests.    Reference java classes for Then conditions –    Used Gherkins to build the automation suites –    In this way, I used the concepts of software development like stability, polymorphism, encapsulation, inheritance, modularity and coupling in building the integration tests using Cucumber framwowrk and gherkins and evident that I have specialist engineering knowledge of automation development and understanding of systems involved in it to solve complex engineering problems.  AND  I used mathematical and algebra calculation in integration tests to handle the part of logics and simulate the behaviour of the functionalities using automation.  Since writing integration tests requires software’s like IntelliJ IDEA editor, Java versions, Unix boxes to validate the unix logs, KDB API’s and whenever integration test are executed on a system, the system should have high configurations like 64bit Operating system, around 10 GB ram to handle the load on the systems, 1 Tera byte hard disk space to accommodate heavy applications like IntelliJ editors.  All these configurations are required for implementing the API’s that can interact with the Optimus components. Please find the Optimus components below –    This gave me exposure on the configurations required to build the integration tests. Hence, I was able to implement the concepts of natural science such as interaction, concurrency and causality, synchrony and asynchrony, compositional modelling and reasoning qualitative versus quantitative reasoning, operational methodologies, continuous versus discrete, etc. as models from physics.   1. **Develop end to UAT tests for QA using TDM (Test Data Manager) automation tool.**   Since integration tests was meant to validate only the flows and behaviour of Optimus components, I needed a mechanism to validate the end to end trade flows integrating the client facing applications (CFORE), CitiSmart (Algo Engine), Optimus Engine (Recommendation Engine) and the stock exchanges.  To achieve this, I used TDM (Test Data Manager) automation tool to configure the actions/functions that can place orders into Citi systems (mocking client facing applications) and validate the logs in Tibco topic servers, Data in KDB services, Optimus recommendations and then citismart application behaviour and then to stock exchanges responses.    I used my functional knowledge of Data communication and software communication to understand the connectivity between these systems in the architecture and implemented the UAT automation solutions**.**  In order to build the QA automation solution, I needed inputs from the Optimus development team members like - functional behaviour of the Optimus applications, architectural structure of the applications, Data fornats required for execution, customized fix tags & their values, expected results from a particular flow, Intergation of Optimus with Citismart application etc..  In order to get these inputs I organized and attended the knowledge transfer sessions with the functional team on regular basis, which helped me in increasing my communication and coordination between the team menbers. Also, upon successful completion of automation model and solution, I have also given a presentation on the QA automation working model, usage, total cost saving in terms of effort and advantages of the solution to the entire functional team and management.  Working in this kind of environment gave me an exposure on challenges & mitigation techniques, learned new technologies functionally (Optimus), got innovative ideas from other team members, got to learn and improve coordination skills and exchange ideas technically and non-technically.  In this way, I was able to communicate both technically and non-techincally and deliver a presentation, user guide, effective cost savings on the model built for citismart application. Also, this helped me in understanding the critical issues that affects the current technical projects, got exposure of the tools & technologies that needs to implemented as a part of this model/package, designed the automation solution to accommodate all the functional requirements, assessed and implemented the resources/inputs required for the project.  Since both Integration tests and UAT automation tests are build using Java and C# programming languages, both of these languages being Object oriented programming languages, I used the objects to create the feature files and functions using polymorphism and inheritance software development concepts.  Created services and step definitions in integration framework to integrate the feature files with Optimus components following coupling/ cohesion development standards –    Created functions/actions in UAT automation tests to integrate the flows between Citismart, Optimus, Tibco topics and Exchanges following modularity development standards –    I used the concepts of software designing (engineering design) like Modularity and Coupling by implementing the concepts of common functions in automation suite development. Hence I applied appropriate design methods in solving complex engineering problems in the automation suite**.**  In delivering this automation solution I have proactively taken responsibilities of completing the tasks with loyalty and quality. I maintained the code quality by doing self-reviews and asking other team members to peer review of the code, brain storming sessions, knowledge transfer sessions. Also, the code coverage is validated using JCov code coverage tool.  The quality management was processes are carried out using JIRA tool. Each and every activity in software development life cycle of this project is tracked using this tool. The dashboards in JIRA are capable of measuring the quality of tasks and monitor the responsibilities.  Throughout the project, the risks and mitigation plans were tracked using the JIRA tool. Risks were proactively raised and escalated to concerned departments by me.  Upon successful implementation of the integration framework and UAT automation suite, I have given walkthrough to team-members, development coordinators & managers and demonstrated the usage, advantages the entire automation solution. The presentations were given through video conferences and conference calls to the relevant team in US, UK & Asia pacific region.  In this way, I used and followed the ethical behaviour, sustainable development methodologies and quality management systems tools and processes.  The entire development of the integration framework and UAT automation tests were carried out using advance technologies and tools like C#, Java, Cucumber, Maven etc…  The development methodology used in the project is Agile (Scrum-ban specifically). All the advanced scrum activities listed below are carried out in this project for end to end delivery –   * Sprints - In Scrum, work is performed in iterations or cycles of up to a calendar month called sprints. The work completed in each sprint should create something of tangible value to the customer or user. Sprints are time boxed so they always have a fixed start and end date, and generally they should all be of the same duration. * Sprint Planning - A product backlog may represent many weeks or months of work, which is much more than can be completed in a single, short sprint. To determine the most important subset of product backlog items to build in the next sprint, the product owner, development team, and ScrumMaster perform sprint planning. During sprint planning, the product owner and development team agree on a sprint goal that defines what the upcoming sprint is supposed to achieve. Using this goal, the development team reviews the product backlog and determines the high priority items that the team can realistically accomplish in the upcoming sprint while working at a sustainable pace * Sprint Execution - Once the Scrum team finishes sprint planning and agrees on the content of the next sprint, the development team, guided by the ScrumMaster’s coaching, performs all of the task-level work necessary to get the features done. “Done” means there is a high degree of confidence that all of the work necessary for producing good-quality features has been completed. Team members define their own task-level work and then self-organize in any manner they feel is best for achieving the sprint goal. * Daily Scrum -Each day of the sprint, ideally at the same time, the development team members hold a time boxed (15 minutes or less) daily scrum. This inspect-and adapt activity is sometimes referred to as the daily stand-up because of the common practice of everyone standing up during the meeting to help promote brevity. A common approach to performing the daily scrum has the ScrumMaster facilitating and each team member taking turns answering three questions for the benefit of the other team members: * What did I accomplish since the last daily scrum? * What do I plan to work on by the next daily scrum? * What are the obstacles or impediments that are preventing me from making progress? * Sprint Reviews - At the end of the sprint there are two additional inspect-and-adapt activities. One is called the sprint review. The goal of this activity is to inspect and adapt the product that is being built. Critical to this activity is the conversation that takes place among its participants, which include the Scrum team, stakeholders, sponsors, customers, and interested members of other teams. The conversation is focused on reviewing the just-completed features in the context of the overall development effort. * Sprint Retrospective - The second inspect-and-adapt activity at the end of the sprint is the sprint retrospective. Whereas the sprint review is a time to inspect and adapt the product, the sprint retrospective is an opportunity to inspect and adapt the process. During the sprint retrospective the development team, ScrumMaster, and product owner come together to discuss what is and is not working with Scrum and associated technical practices.   Again, In this way I learned advanced technologies like Agile development methodologies, Cucumber, Maven and implemented them in this project. | **Element**  **Element 8**  **Element 4**  **Element 2**  **Element 1**  **Element 3**  **Element 6**  **Element 5**  **Element 7**  **Element 8** |

|  |  |
| --- | --- |
| **Work/Study Episode 3**  **Overview of the project**  **Your role and responsibilities**  **Complexities (using the complexity definitions) and challenges of the project**  **How does this project demonstrate application of your engineering knowledge?** | **Element** |
| **Work/Study Episode 4**  **Overview of the project**  **Your role and responsibilities**  **Complexities (using the complexity definitions) and challenges of the project**  **How does this project demonstrate application of your engineering knowledge?** | **Element** |

**Knowledge Matrix**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Knowledge Element** | **W/S Episode 1** | **W/S Episode 2** | **W/S Episode 3** | **W/S Episode 4** |
| 1. Application of knowledge from one or more of the natural sciences | Y | Y |  |  |
| 1. Application of knowledge of mathematics | Y | Y |  |  |
| 1. Application of knowledge of engineering fundamentals | Y | Y |  |  |
| 1. Application of specialist engineering knowledge to solve complex problems | Y | Y |  |  |
| 1. Application of knowledge of design methods to solve complex problems | Y | Y |  |  |
| 1. Application of knowledge of key elements of engineering practice | Y | Y |  |  |
| 1. Role of Engineering in Society | Y | Y |  |  |
| 1. Application of advanced knowledge in an area of your discipline |  | Y |  |  |

Section four – Supplementary evidence

**Academic Transcript(s)**

Please attach a certified copy of your academic transcript(s) if you have not already supplied one to IPENZ

Work history summary

List your employment history starting from your most recent employment and then chronologically back to the start of your first job.

| **Ref No** | **Name of Employing Organisation** | **Position Title** | **End mm/yy Start mm/yy** | **Key responsibilities, activities undertaken, major achievements and/or projects. These should relate to your practice area description.** |
| --- | --- | --- | --- | --- |
|  |  |  | Present  Start at |  |
|  |  |  | End date:  Start date: |  |
|  |  |  | End date:  Start date: |  |
|  |  |  | End date:  Start date: |  |
|  |  |  | End date:  Start date: |  |
|  |  |  | End date:  Start date: |  |
|  |  |  | End date:  Start date: |  |

continued professional development (CPD) activities Summary

| **Description of Activity and Learning.**  **Please record all relevant CPD activities (eg. short course, conference, reading, technical lectures, formal study towards qualification, research, discussion groups, workshops, symposia, voluntary service roles) that have extended your professional engineering knowledge and have assisted you to develop the knowledge profile of a professional engineer. Describe the learning outcomes and how these have contributed to your acquiring a Washington Accord level of knowledge..** | | | | | **Was Formal Assessment involved?**  **What was the outcome?** |
| --- | --- | --- | --- | --- | --- |
|
| **Date(s)** | **Actual Hours** | **Form of Activity** | **Title of activity** | **What was the knowledge you acquired?  How have you applied this knowledge in your engineering practice?** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Section Five - Payment

knowledge assessment (Level 2) Fee Payment

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Assessment Fee (incl GST) in NZD | | | | | | | | | | **NZ$1,351.25** | | | | | | | |  | | | Please send a receipt | | | | | | | | |  | |
|  | Credit Card Details: | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Visa | | Bankcard / Mastercard | | | | | | | | | | American Express | | | | | |  | | | | | Diners Card | | | | |
| **Credit Card Number** | | |  |  | |  |  |  |  | |  | |  | |  |  |  | |  |  | |  | |  |  |  |  | |  | |  | |
| **Name on card** | | |  | | | | | | | | | | | **Expiry Date** | | | |  | | | **CVV** | | | | | | |
|  | | |  | | | | | | | | | | |  | | | |  | | |  | | | | | | |
| **Cardholders Signature** | | |  | |  | | | | | | | | | | | | | | | | | | | | | | |

Where to send Completed documents

Send the completed form and associated documents to the **IPENZ Membership Manager** at one of the addresses below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Courier Address:** | IPENZ National Office,  Level Three  50 Customhouse Quay  Wellington 6144,  New Zealand | **Postal Address:** | IPENZ National Office  PO Box 12-241  Wellington 6144  New Zealand |

**Appendix One**

Complexity Definitions

**Complex engineering problems**

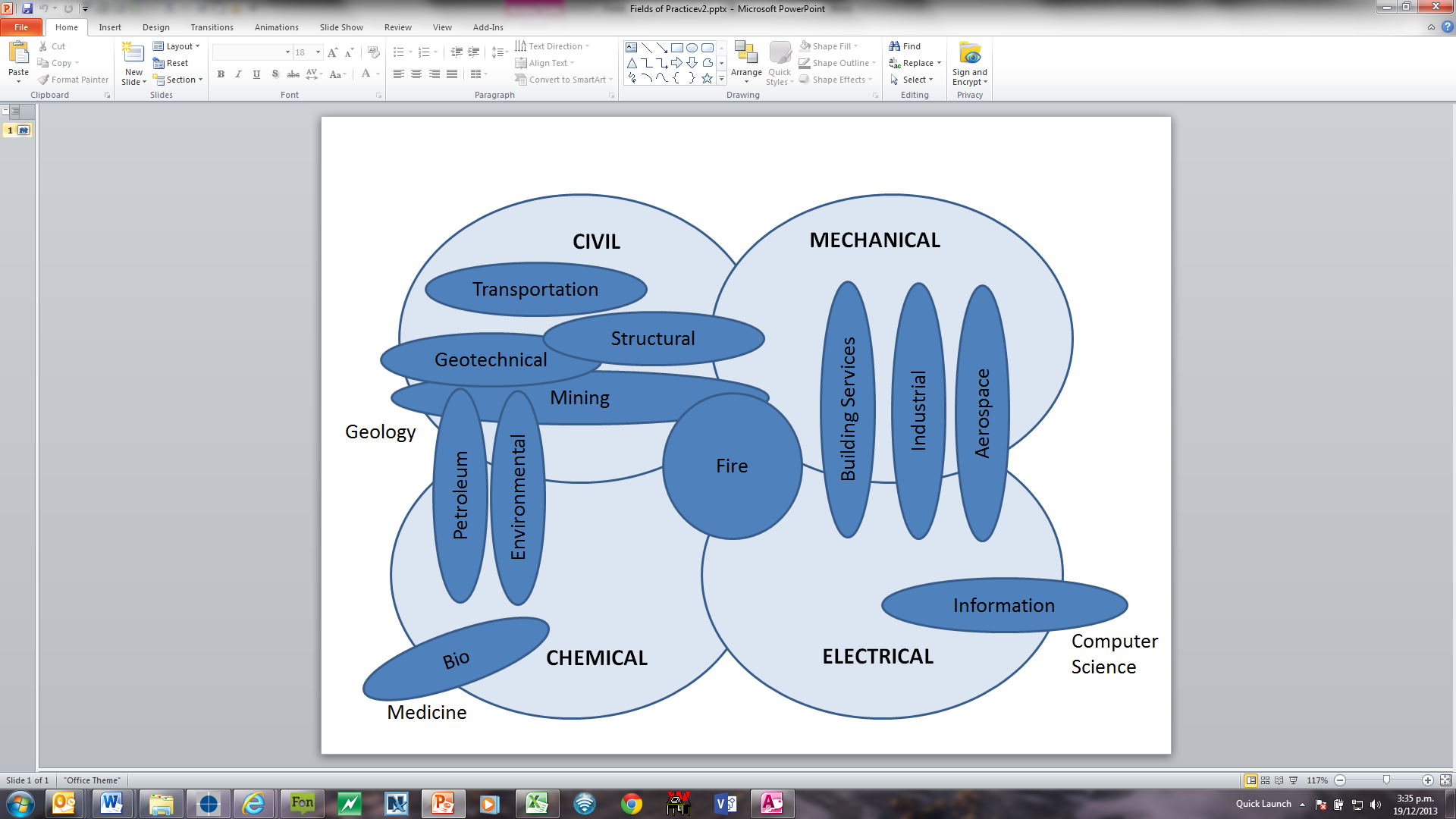
Complex engineering problems have some or all of the following characteristics:

* Involve wide-ranging or conflicting technical, engineering, and other issues;
* Have no obvious solution and require originality in analysis;
* Involve infrequently encountered issues;
* Are outside problems encompassed by standards and codes of practice for professional engineering;
* Involve diverse groups of stakeholders with widely varying needs;
* Have significant consequences in a range of contexts;
* Cannot be resolved without in-depth engineering knowledge

Appendix Two

Disciplines and fields of engineering

Engineering practice fields are loosely defined terms and are used as an indication of the nature of engineering work carried out by engineers practising in an engineering field of practice. The following diagram is a graphical display of the relationships between the various fields and the four core disciplines. Some fields may extend into other fields of scientific endeavour.



**Aerospace Engineering**

Aerospace engineering is the design, development, and production of aircraft (aeronautical engineering), spacecraft (astronautical engineering) and related systems. Aerospace engineers may specialise in aerodynamics, avionics, structures, control systems or propulsion systems. It may involve planning maintenance programmes, designing repairs and modifications and exercising strict safety and quality controls to ensure airworthy operations.

**Bio Engineering**

Bioengineering draws heavily on the Chemical Engineering discipline and involves the engineered development of raw materials to produce higher value products, using biological systems (biological catalysts). The description also encompasses the general application of engineering to biological systems to develop new products or solve problems in existing production processes. As examples, bioengineers are found in medical research, genetic science, fermentation industries and industries treating biological wastes.

**Building Services**

Building Services engineering is the application of mechanical or electrical engineering principles, and an understanding of building structure, to enhance all aspects of the built environment from air conditioning and mechanical ventilation, electrical light and power, fire services, fire safety engineering, water and waste services, data and communications, security and access control, vertical transportation, acoustics and energy management.

**Chemical Engineering**

Chemical engineering is concerned with the ways in which raw materials are changed into useful and commercial end products such as food, petrol, plastics, paints, paper, ceramics, minerals and metals. Often these processes are carried out at large scale plants. Research of raw materials and their properties, design and development of equipment and the evaluation of operating processes are all part of chemical engineering.

**Civil Engineering**

Civil engineering is a broad field of engineering concerned with the, design, construction, operation and maintenance of structures (buildings, bridges, dams, ports) and infrastructure assets (road, rail, water, sewerage). The Civil engineering discipline underpins several engineering fields such as Structural, Mining, Geotechnical and Transportation engineering, in which civil engineers often specialise. General Civil engineers are likely to be competent to undertake work that relates to one or more of these areas.

**Electrical Engineering**

Electrical engineering is the field of engineering which deals with the practical application of electricity. It deals with the aspects of planning, design, operation and maintenance of electricity generation and distribution, and use of electricity as a source of energy within major buildings, industrial processing complexes, facilities and transport systems. It includes the associated networks and the equipment involved such as switchboards, cabling, overhead lines/catenaries, earthing, control and instrumentation systems.

Areas of specialisation within the wider electrical engineering discipline, such as electronics and telecommunications are usually concerned with using electricity to transmit information rather than energy. For this reason electronics and radiocommunications/telecommunications are captured under the field of Information Engineering.

**Engineering management**

The Engineering Management practice field is used by engineers who manage multi-disciplinary engineering activities that are so multi-disciplined that it is difficult to readily link their engineering practice with any other specific practice field. Project managers, asset managers and engineers working in policy development are likely to use the ‘Engineering Management’ field.

**Environmental Engineering**

Environmental engineering draws on the Civil and Chemical engineering disciplines to provide healthy water, air and land to enhance human habitation. Environmental engineers devise, implement and manage solutions to protect and restore the environment, within an overall framework of sustainable development. The role of the environmental engineer embraces all of the air, water and soil environments, and the interactions between them.

**Fire Engineering**

Fire engineering draws on knowledge from the range of engineering disciplines to minimise the risk from fire to health and safety and damage to property through careful design and construction. It requires an understanding of the behaviour of fires and smoke, the behaviour of people exposed to fires and the performance of burning materials and structures, as well as the impact of fire protection systems including detection, alarm and extinguishing systems.

**Geotechnical Engineering**

Geotechnical engineering involves application of knowledge of earth materials in the design of structures, such as foundations, retaining walls, tunnels, dams and embankments. Geotechnical engineers assess the properties and performance of earth materials such as their stability and strength, and the impact of groundwater.

**industrial engineering**

Industrial engineering is the application of mechanical and electrical engineering principles to the design and operation of production equipment, production lines and production processes for the efficient production of industrial goods. Industrial engineers understand plant and procedural design, the management of materials and energy, and human factors associated with worker integration with systems. Industrial engineers increasingly draw on specialised knowledge of robotics, mechatronics, and artificial intelligence.

**Information engineering**

The field of Information engineering is based on the Electrical engineering discipline but also draws heavily from Computer Science. Three areas of further specialisation can be identified:

Software engineering - The development and operation of software-intensive systems that capture, store and process data.

Telecommunications engineering - The development and operation of systems that encode, transmit and decode data via cable systems (including fibre optics) and wireless systems (radiocommunications).

Electronics engineering - The design, development and testing of electronic circuits and networks that use the electrical and electromagnetic properties of electronic components integrated circuits and microprocessors to sense, measure and control processes and systems.

**Mechanical Engineering**

Mechanical Engineering involves the design, manufacture and maintenance of mechanical systems. Mechanical engineers work across a range of industries and are involved with the design and manufacture of a range of machines or mechanical systems, typically applying principles of hydraulics (fluid control), pneumatics (air pressure control) or thermodynamics (heat energy transfer). Mechanical engineers may specialise in the Building Services or Industrial engineering field.

**Mining Engineering**

Mining engineering involves extracting and processing minerals from the earth. This may involve investigations, design, construction and operation of mining, extraction and processing facilities.

**Petroleum Engineering**

Petroleum engineering is a field of engineering relating to oil and gas exploration and production. Petroleum engineers typically combine knowledge of geology and earth sciences with specialised Chemical engineering skills, but may also draw on Mechanical engineering expertise to design extraction and production methods and equipment. Petroleum engineering activities are divided into two broad categories:

Upstream - locating oil and gas beneath the earth's surface and then developing methods to bring them out of the ground.

Downstream - the design and development of plant and infrastructure for the refinement and distribution of the mixture of oil, gas and water components that are extracted

**Structural Engineering**

Structural Engineering is a specialised field within the broader Civil engineering discipline that is concerned with the design and construction of structures. Structures might include buildings, bridges, in-ground structures, footings, frameworks and space frames, including those for motor vehicles, space vehicles, ships, aeroplanes and cranes, composed of any structural material including composites and novel materials.

**Transportation**

Transportation engineering is a specialised field of practice in the civil engineering discipline relating to the movement of goods and people by road, water, rail and air.

A Transportation engineer might specialise in one or more of: pavement design, asset maintenance/management, construction/project management, traffic operations and control, transportation planning and systems analysis, freight transportation and logistics, road safety, railways or public transport systems.