

**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI**

**WORK INTEGRATED LEARNING PROGRAMMES**

**COURSE HANDOUT – Flipped**

**Part A: Content Design**

|  |  |
| --- | --- |
| **Course Title** | Software Quality Assurance and Testing |
| **Course No(s)** | SE\* ZG501 |
| **Credit Units** | 4 |
| **Course Author** | Bhaskar Zeminder |
| **Version No** | 1.0 |
| **Date** | 29- July-2023 |

**Course Objectives**

|  |  |
| --- | --- |
| **No** | Objective |
| **CO1** | Develop a comprehensive understanding of software quality assurance principles, methodologies, and industry best practices to ensure the delivery of high-quality software products |
| **CO2** | Master the various techniques and tools used in software testing, verification, and validation to effectively identify and rectify defects throughout the software development lifecycle. |
| **CO3** | Analyse and apply advanced quality assurance strategies such as test automation, continuous integration, and agile testing, to enhance the efficiency and effectiveness of software development teams in delivering top-notch software products |

**Textbook(s)**

|  |  |
| --- | --- |
| T1 | Software Quality Assurance Book by Alain April and Claude Y. Laporte |
| T2 | Software Quality Assurance (From Theory to Implementation) by Daniel Galin |

**Reference Book(s) & other resources**

|  |  |
| --- | --- |
| R1 | Software Quality Assurance By Ivan Mistrik, Richard M Soley, Nour Ali, John Grundy, Bedir Tekinerdogan |
| R2 | Software Testing: Concepts and Operations by Rajiv Chopra |
| R3 | Software Testing and Quality Assurance – Theory and Practice, Kshirasagar Naik, Priyadarshi Tripathy, Wiley, 2013 |
| R4 | Software Quality Engineering – Jeff Tian, Wiley India, 2015 |
| R5 | Quality Planning and Assurance Book by Herman Tang |

**Learning Outcomes:**

|  |  |  |
| --- | --- | --- |
| No | Learning Outcomes | Objectives |
| LO1 | Advanced knowledge of Software Quality Assurance principles and methodologies | By the end of the course, participants should have an in-depth understanding of the advanced principles, concepts, and methodologies of Software Quality Assurance. They should be well-versed in topics like quality models, process improvement frameworks (e.g., CMMI, Six Sigma), risk management, and the application of SQA in different software development models (e.g., Agile, DevOps). |
| LO2 | Master testing techniques and tools for comprehensive software testing. | Upon completing the course, participants should have a solid knowledge of various testing techniques, such as unit testing, integration testing, system testing, and acceptance testing. They should also be proficient in using popular testing tools and frameworks to create, execute, and analyse test cases effectively, ensuring comprehensive test coverage and defect detection. |
| LO3 | Proficiency in designing and implementing robust quality assurance strategies | Upon completing the course, participants should be capable of developing comprehensive and effective quality assurance strategies tailored to specific software projects. They should be skilled in devising test plans, defining test cases, and establishing quality metrics to measure the performance and reliability of software applications across different domains and industries. |

**Content Structure**

**Module 1: Essential SQA: Processes and Success Factors**

|  |  |  |
| --- | --- | --- |
| Topic No. | Topic Title | Reference |
| 1.1 | Definition and importance of software quality assurance | T1 Chapter 1 |
| 1.2 | Distinction between Quality Assurance and Quality Control | Lecture Notes |
| 1.3 | Success Factors in Quality Assurance | T1 Chapter 1 &  T2 Chapter 1 |
| 1.4 | Cost of Quality and Quality Culture | T1 Chapter 2 |
| 1.5 | Role of SQA in software development life cycle | Lecture Notes |

**Module 2: Standardizing SQA: Quality Models and Management**

|  |  |  |
| --- | --- | --- |
| Topic No. | Topic Title | Reference |
| 2.1 | Software Quality Models | T1 Chapter 3,  R1 Chapter 2 |
| 2.2 | Specifying Quality Requirements and Plan | T1 Chapter 3 |
| 2.3 | Requirement Traceability During Software Lifecycle | T1 Chapter 3 |
| 2.4 | Standards for Quality Management | T1 Chapter 4 |
| 2.5 | Frameworks (ITIL, ISO, CMMi) | T1 Chapter 4 |

**Module 3: Fundamentals of SQA: Software Quality Attributes**

|  |  |  |
| --- | --- | --- |
| Topic No. | Topic Title | Reference |
| 3.1 | Software Requirements into Software Quality Factors | T2 Chapter 3 |
| 3.2 | Understanding quality attributes   * Reliability * Usability * Maintainability * Other quality attributes | T2 Chapter 3,  R1 Chapter 2 |
| 3.3 | Alternative models of Software Quality Factors | T2 Chapter 3 |

**Module 4: Deep driving SQA: Software Testing Techniques**

|  |  |  |
| --- | --- | --- |
| Topic No. | Topic Title | Reference |
| 4.1 | Software Testing Fundamentals | T2 Chapter 9,  R2 Chapter 1 |
| 4.2 | Software Verification and Validation | R2 Chapter 2 |
| 4.3 | Test design techniques (black-box testing, white-box testing, boundary value analysis, equivalence partitioning, etc.) | R2 Chapter 3  R2 Chapter 4 |
| 4.4 | Test levels and types (unit testing, integration testing, system testing, etc.) | R2 Chapter 7 |

**Module 5: Mastering SQA: Test Execution and Automated Testing**

|  |  |  |
| --- | --- | --- |
| Topic No. | Topic Title | Reference |
| 5.1 | Test Execution Process | T2 Chapter 10 |
| 5.2 | Test Case Design | T2 Chapter 10 |
| 5.3 | Automated testing | T2 Chapter 10,  R2 Chapter 9 |
| 5.4 | Alpha and Beta site testing programs | T2 Chapter 10 |
| 5.5 | Regression Testing Strategies | R2 Chapter 6,  R2 Chapter 12 |
| 5.6 | Case Study: Exploring Automated Source Code Analyzers and Software Composition Analysis Tools | Lecture Notes |

**Module 6: Effective SQA: Quality Audits and Project Assessments**

|  |  |  |
| --- | --- | --- |
| Topic No. | Topic Title | Reference |
| 6.1 | Personal Review, Inspection Review and Project Assessments | T1 Chapter 5,  T2 Chapter 8 |
| 6.2 | Types of Audits (Internal, Third Party) | T1 Chapter 6 |
| 6.3 | Project Assessment and Control Process | T1 Chapter 6,  T2 Chapter 8 |
| 6.4 | Corrective Actions | T1 Chapter 8 |

**Module 7: Comprehensive SQA: Effective Test Management and Planning**

|  |  |  |
| --- | --- | --- |
| Topic No. | Topic Title | Reference |
| 7.1 | Test Organization and Team Management | T1 Chapter 5 |
| 7.2 | Test Estimation and Scheduling | R1 Chapter 7 |
| 7.3 | Test Data Management | Lecture Notes |
| 7.4 | Configuration Management and Change Control | T1 Chapter 8,  T1 Chapter 5 |
| 7.5 | Case Study: Develop a test plan and design test cases for a given software application | Lecture Notes |

**Module 8: Enhancing SQA: Process Improvement and Metrics**

|  |  |  |
| --- | --- | --- |
| Topic No. | Topic Title | Reference |
| 8.1 | Introduction to Test Process Improvement | T1 Chapter 9 |
| 8.2 | Capability Maturity Model Integration (CMMI) for Testing | T1 Chapter 10 |
| 8.3 | Six Sigma in Software Testing | Lecture Notes |
| 8.4 | Test Metrics for Process Improvement | T2 Chapter 21 |

**Module 9: Optimizing SQA: Agile Testing and DevOps Integration**

|  |  |  |
| --- | --- | --- |
| Topic No. | Topic Title | Reference |
| 9.1 | Introduction to Agile Methodology and Testing | T1 Chapter 4 |
| 9.2 | Agile Test Planning and Execution | T1 Chapter 5 |
| 9.3 | Continuous Testing in DevOps | Lecture Notes |
| 9.4 | Test Environment and Test Data Management in DevOps | Lecture Notes |

**Module 10: Excelling SQA: Best Practices and Case Studies**

|  |  |  |
| --- | --- | --- |
| Topic No. | Topic Title | Reference |
| 10.1 | Best Practices for SQA implementation | T1 Chapter 4,  Lecture Notes |
| 10.2 | Quality assurance in different development methodologies (Waterfall, Agile, etc.) | T2 Chapter 7 |
| 10.3 | Building a quality culture in organizations | R1 Chapter 7 |
| 10.4 | Case studies of successful SQA implementations | Lecture Notes |
| 10.5 | Lessons learned from successful software quality assurance projects | Lecture Notes |

**Module 11: Shaping SQA: An Outlook for the Future**

|  |  |  |
| --- | --- | --- |
| Topic No. | Topic Title | Reference |
| 11.1 | Emerging technologies and their impact on SQA | Lecture Notes |
| 11.2 | Artificial intelligence and machine learning in quality assurance | Lecture Notes |
| 11.3 | Blockchain and quality assurance | Lecture Notes |
| 11.4 | Future directions and career opportunities in SQA | Lecture Notes |

**Part B: Contact Session Plan**

|  |  |
| --- | --- |
| **Academic Term** |  |
| **Course Title** | Software Quality Assurance and Testing |
| **Course No** |  |
| **Lead Instructor** |  |

**Glossary of Terms**

|  |  |  |
| --- | --- | --- |
| **Module** | **M** | Module is a standalone quantum of designed content. A typical course is delivered using a string of modules. M2 means module 2. |
| **Contact Hour** | **CH** | Contact Hour (CH) stands for an hour long live session with students conducted either in a physical classroom or enabled through technology. In this model of instruction, instructor led sessions will be for 32 CH. |
| **Recorded Lecture** | **RL** | RL stands for Recorded Lecture or Recorded Lesson. It is presented to the student through an online portal. A given RL unfolds as a sequences of video segments interleaved with exercises. |
| **Lab Exercises** | **LE** | Lab exercises associated with various modules |
| **Self-Study** | **SS** | Specific content assigned for self study |
| **Homework** | **HW** | Specific problems/design/lab exercises assigned as homework |

**Module Summary**

|  |  |
| --- | --- |
| **Module No** | **Title of Module** |
| M1 | Essential SQA: Processes and Success Factors |
| M2 | Standardising SQA: Quality Models and Management |
| M3 | Fundamentals of SQA: Software Quality Attributes |
| M4 | Deep driving SQA: Software Testing Techniques |
| M5 | Mastering SQA: Test Execution and Automated Testing |
| M6 | Effective SQA: Quality Audits and Project Assessments |
| M7 | Comprehensive SQA: Effective Test Management and Planning |
| M8 | Enhancing SQA: Process Improvement and Metrics |
| M9 | Optimizing SQA: Agile Testing and DevOps Integration |
| M10 | Excelling SQA: Best Practices and Case Studies |
| M11 | Shaping SQA: An Outlook for the Future |

**Detailed Structure**

**M1: Essential SQA: Processes and Success Factors**

|  |  |
| --- | --- |
| **Type** | **Description/Plan/Reference** |
| SQAT\_RL 1.1.1 | Definition of Software |
| SQAT\_RL 1.1.2 | Definition of Software Quality and Quality Assurance |
| SQAT\_RL 1.2.1 | Quality Control Vs Quality Assurance |
| SQAT\_RL 1.2.2 | Quality Assurance Vs Software Testing |
| SQAT\_RL 1.2.3 | Importance of QA |
| SQAT\_RL 1.3.1 | Software components |
| SQAT\_RL 1.3.2 | Defect Causes |
| SQAT\_RL 1.3.3 | Software Delivery Issues |
| SQAT\_RL 1.3.4 | Software Business Models |
| SQAT\_RL 1.3.5 | Cost of Quality |
| SQAT\_RL 1.3.6 | Quality Culture |
| CS 1.1 | * Definitions of Software, Software Quality, Quality Assurance * Understanding Quality Assurance Vs Quality Control |
| CS 1.2 | * Understanding Importance of QA * Understanding Cost of Quality and Quality Culture |
| SS 1 |  |

**M2: Standardising SQA: Quality Models and Management**

|  |  |
| --- | --- |
| **Type** | **Description/Plan/Reference** |
| SQAT\_RL 2.1.1 | Quality Perspectives |
| SQAT\_RL 2.1.2 | McCall Quality Model |
| SQAT\_RL 2.1.3 | Internal Vs External Quality |
| SQAT\_RL 2.2.1 | IEEE 1061 Quality Model |
| SQAT\_RL 2.2.2 | ISO 25000, ISO 25010 |
| SQAT\_RL 2.3.1 | Requirements – Definition |
| SQAT\_RL 2.3.2 | Software Quality Requirements |
| SQAT\_RL 2.4.1 | ISO/IEC/IEEE 12207 Software Lifecycle Processes |
| SQAT\_RL 2.4.2 | Software Quality Assurance Process |
| SQAT\_RL 2.5.1 | CMMI Model |
| SQAT\_RL 2.5.2 | ITIL Framework |
| SQAT\_RL 2.5.3 | Other Frameworks |
| CS 2.1 | * Quality Perspectives * McCall Quality Model * Internal Quality Vs External Quality |
| CS 2.2 | * IEEE 1061, ISO 25000, ISO 25010 * Software quality Assurance process in detail |
| CS 3.1 | * CMMI, ITIL and other Frameworks |
| SS 2 |  |

**M3: Fundamentals of SQA: Software Quality Attributes**

|  |  |
| --- | --- |
| **Type** | **Description/Plan/Reference** |
| SQAT\_RL 3.1.1 | McCall Quality Model |
| SQAT\_RL 3.1.2 | ISO 25000 Series |
| SQAT\_RL 3.2.1 | Introduction to ISO 25010 |
| SQAT\_RL 3.2.2 | Quality Attributes – Functionality, Performance |
| SQAT\_RL 3.2.3 | Quality Attributes – Compatibility, Usability |
| SQAT\_RL 3.2.4 | Quality Attributes – Reliability, Security |
| SQAT\_RL 3.2.5 | Quality Attributes – Maintainability, Portability |
| CS 3.2 | * Understanding ISO 25000 Series * Introduction to ISO 25010 * Understanding Quality Attributes |
| CS 4.1 | * Functionality, Performance * Compatibility, Usability |
| CS 4.2 | * Reliability, Security * Maintainability, Portability |
| SS 3 |  |

**M4: Deep driving SQA: Software Testing Techniques**

|  |  |
| --- | --- |
| **Type** | **Description/Plan/Reference** |
| SQAT\_RL 4.1.1 | Software Testing – Definitions and Fundamentals |
| SQAT\_RL 4.1.2 | Software Testing – Why needed |
| SQAT\_RL 4.1.3 | How much to Test |
| SQAT\_RL 4.1.4 | Software Testing – Terms and Principles |
| SQAT\_RL 4.2.1 | Verification and Validation – Definition, Comparison |
| SQAT\_RL 4.2.2 | V&V – Techniques |
| SQAT\_RL 4.2.3 | V&V – Object, Activities, Goals |
| SQAT\_RL 4.3.1 | Boundary Value Analysis |
| SQAT\_RL 4.3.2 | Equivalence class Partition |
| SQAT\_RL 4.4.1 | Unit Testing, Integration Testing, System Testing |
| CS 5.1 | * Software Testing Definitions, Fundamentals * Software Testing – Terms and Principles |
| CS 5.2 | * Verification and Validation – Definition and Comparison * V&V Techniques – BVA |
| CS 6.1 | * Equivalence Class Partition * Unit Testing, Integration Testing, System Testing |
| SS 4 |  |

**M5: Mastering SQA: Test Execution and Automated Testing**

|  |  |
| --- | --- |
| **Type** | **Description/Plan/Reference** |
| SQAT\_RL 5.1.1 | Software Test Execution – Goals and Phases |
| SQAT\_RL 5.1.2 | Software Testing – Strategy and Planning |
| SQAT\_RL 5.1.3 | Software Test – Design and Prioritization |
| SQAT\_RL 5.1.4 | Software Test Execution and Reporting |
| SQAT\_RL 5.2.1 | Automated Testing Process and Types |
| SQAT\_RL 5.2.2 | Test automation Tools |
| SQAT\_RL 5.3.1 | Automate Testing Advantages and Disadvantages |
| SQAT\_RL 5.4.1 | Alpha and Beta Testing |
| SQAT\_RL 5.5.1 | Defect Prioritization |
| SQAT\_RL 5.5.2 | Regression Testing – Fundamentals and Types of Regression Testing |
| SQAT\_RL 5.6.1 | Source code Analysers. |
| SQAT\_RL 5.6.2 | Software composition Tools |
| CS 6.2 | * Software Test Design, Execution, Prioritization, Reporting |
| CS 7.1 | * Understanding Alpa and Beta Testing * Automation Testing – Tools, Advantages, Disadvantages |
| CS 7.2 | * Defect reporting, Prioritization * Understanding Regression Testing * Static code Analysis and code composition Tools |
| SS 5 |  |

**M6: Effective SQA: Quality Audits and Project Assessments**

|  |  |
| --- | --- |
| **Type** | **Description/Plan/Reference** |
| SQAT\_RL 6.1.1 | Reviews – Definitions, Types of reviews |
| SQAT\_RL 6.1.2 | Personal Reviews |
| SQAT\_RL 6.1.3 | Inspection Reviews |
| SQAT\_RL 6.1.4 | Desk-Check Reviews |
| SQAT\_RL 6.1.5 | Walkthrough |
| SQAT\_RL 6.2.1 | ISO 20246 |
| SQAT\_RL 6.2.2 | IEEE 1028 |
| SQAT\_RL 6.2.3 | CMMI-Dev – Verification |
| SQAT\_RL 6.3.1 | Audits – Definitions |
| SQAT\_RL 6.3.2 | Types of Audits |
| SQAT\_RL 6.3.3 | Audit as per IEEE-1028 |
| SQAT\_RL 6.3.4 | Audit as per CMMI-Dev |
| SQAT\_RL 6.4.1 | Corrective Action |
| SQAT\_RL 6.4.2 | Corrective Action Process |
| CS 9.1 | * Reviews – Types |
| CS 9.2 | * Audits * Corrective Action |
| SS 6 |  |

**M7: Comprehensive SQA: Effective Test Management and Planning**

|  |  |
| --- | --- |
| **Type** | **Description/Plan/Reference** |
| SQAT\_RL 7.1.1 | Test Organization |
| SQAT\_RL 7.1.2 | Roles of Test organisation |
| SQAT\_RL 7.2.1 | Test Estimation |
| SQAT\_RL 7.2.2 | Test Estimation Methodologies |
| SQAT\_RL 7.2.3 | Test Estimation Best practices |
| SQAT\_RL 7.2.4 | Test Data Management |
| SQAT\_RL 7.3.1 | Software configuration Management – Definition |
| SQAT\_RL 7.3.2 | SCM - Benefits |
| SQAT\_RL 7.3.3 | SCM as per IEEE-828 |
| SQAT\_RL 7.3.4 | SCM as per CMMI-Dev |
| SQAT\_RL 7.4.1 | Baselines |
| SQAT\_RL 7.4.2 | Branches |
| SQAT\_RL 7.4.3 | Branching Strategies |
| SQAT\_RL 7.5.1 | Configuration/Change control - Definition |
| SQAT\_RL 7.5.2 | Change Phases/Steps |
| SQAT\_RL 7.5.3 | Control Board |
| SQAT\_RL 7.5.4 | Change Management Policy |
| SQAT\_RL 7.6.1 | Case Study - Test Plan and Test Design for S/W Application |
| CS 10.1 | * Test Organisation and Roles |
| CS 10.2 | * Test Estimation Methodologies |
| CS 11.1 | * Configuration Management |
| CS 11.2 | * Case Study – Test Plan and Test Case design for Application |
| SS 7 |  |

**M8: Enhancing SQA: Process Improvement and Metrics**

|  |  |
| --- | --- |
| **Type** | **Description/Plan/Reference** |
| SQAT\_RL 8.1.1 | Test Process Improvement – Fundamentals |
| SQAT\_RL 8.1.2 | Test Process Improvement – When |
| SQAT\_RL 8.1.3 | Test Process Improvement – How |
| SQAT\_RL 8.1.4 | Test Process Improvement – Models |
| SQAT\_RL 8.2.1 | CMMI-Dev for Testing |
| SQAT\_RL 8.2.2 | CMMI-Dev Processes for Testing |
| SQAT\_RL 8.3.1 | Six Sigma – An overview |
| SQAT\_RL 8.3.2 | Aspects of six sigma in software Testing |
| SQAT\_RL 8.3.3 | Six sigma Tools and techniques in software Testing |
| SQAT\_RL 8.4.1 | Quality Metrics |
| SQAT\_RL 8.4.2 | Quality Metrics – Objectives |
| SQAT\_RL 8.4.3 | Quality Metrics – Classification |
| SQAT\_RL 8.4.4 | Software Process Metrics |
| SQAT\_RL 8.4.5 | Quality Metrics – Implementation |
| CS 12.1 | * Test Process improvement |
| CS 12.2 | * Quality metrics |
| SS 8 |  |

**M9: Optimizing SQA: Agile Testing and DevOps Integration**

|  |  |
| --- | --- |
| **Type** | **Description/Plan/Reference** |
| SQAT\_RL 9.1.1 | Agile Development – Overview |
| SQAT\_RL 9.1.2 | Agile Development – Methodology |
| SQAT\_RL 9.1.3 | Agile Development – Advantages/Disadvantages |
| SQAT\_RL 9.1.4 | Agile Development – Comparison with other Methodologies |
| SQAT\_RL 9.2.1 | Agile Testing |
| SQAT\_RL 9.2.2 | Agile Testing Methodologies |
| SQAT\_RL 9.2.3 | Agile Testing Strategy |
| SQAT\_RL 9.2.4 | Benefits Of Agile Testing |
| SQAT\_RL 9.3.1 | Introduction to DevOps |
| SQAT\_RL 9.3.2 | Testing in DevOps |
| SQAT\_RL 9.3.3 | Testing in DevOps – Best Practices |
| SQAT\_RL 9.4.1 | Testing Environment in DevOps |
| SQAT\_RL 9.4.2 | Managing Test Environment in DevOps |
| SQAT\_RL 9.4.3 | Challenges in Managing Test Environment in DevOps |
| SQAT\_RL 9.5.1 | Test Data Management in DevOps |
| SQAT\_RL 9.5.2 | Test Data Management in DevOps – Challenges |
| SQAT\_RL 9.5.3 | Test Data Management in DevOps - Strategies |
| CS 13.1 | * Agile Development and Testing |
| CS 13.2 | * Test Environment and Test Data in DevOps |
| SS 9 |  |

**M10: Excelling SQA: Best Practices and Case Studies**

|  |  |
| --- | --- |
| **Type** | **Description/Plan/Reference** |
| SQAT\_RL 10.1.1 | Best practices for SQA implementation |
| SQAT\_RL 10.1.2 | Challenges for SQA implementation |
| SQAT\_RL 10.2.1 | QA in different SDLC |
| SQAT\_RL 10.2.2 | QA in different Waterfall |
| SQAT\_RL 10.2.3 | QA in different Agile |
| SQAT\_RL 10.3.1 | Case Study – Successful QA implementation |
| CS 14.1 | * SQA Best Practices/Implementation * QA in Waterfall and Agile |
| CS 14.2 | * Case Study – Successful QA implementation |
| SS 10 |  |

**M11: Shaping SQA: An Outlook for the Future**

|  |  |
| --- | --- |
| **Type** | **Description/Plan/Reference** |
| SQAT\_RL 11.1.1 | Emerging Technologies |
| SQAT\_RL 11.1.2 | Impact on QA due to emerging technologies |
| SQAT\_RL 11.1.3 | AI and Machine learning in QA |
| SQAT\_RL 11.2.1 | Blockchain Testing – Overview |
| SQAT\_RL 11.2.2 | Types of Blockchain Testing |
| SQAT\_RL 11.2.3 | Blockchain Testing – Challenges |
| SQAT\_RL 11.2.4 | Blockchain Testing – Best practices |
| SQAT\_RL 11.3.1 | Future of QA Engineers |
| SQAT\_RL 11.3.2 | Career paths for QA Engineers |
| CS 15.1 | * Emerging Technologies and QA * Blockchain Testing |
| CS 15.2 | * Future of QA * Career path for QA Engineers |
| SS 11 |  |

**Teaching Methodology *(Online Session Mode)***

The pedagogy for this course is centred around online contact sessions, which consist of 1.5-hour lecture sessions. In addition to the delivery of lessons on the topics, these contact sessions will also be enriched with discussions on organization-specific practices and case studies from experienced QA managers in the Indian IT industry.

**Course Delivery**

* There are 16 Contact Sessions (of 1.5 hours each) — before mid-semester and 8 post-mid-semester over a period of 16 weeks
* The 8th & 16th Contact Sessions are planned for review of topics pre-mid-semester and pre-comprehensive examinations.

**Contact Session Plan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CS** | **CH** | **Pre-CH** | **During CH** | **Post-CH** |
| 1 | 1 | SQAT\_RL 1.1.1- 1.2.2 | CS 1.1 |  |
| 2 | SQAT\_RL 1.2.3- 1.3.6 | CS1.2 | SS 1 |
| 2 | 3 | SQAT\_RL 2.1.1- 2.1.3 | CS 2.1 |  |
| 4 | SQAT\_RL 2.2.1- 2.4.2 | CS 2. 2 | SS 2 |
| 3 | 5 | SQAT\_RL 2.5.1- 2.5.3 | CS 3.1 |  |
| 6 | SQAT\_RL 3.1.1- 3.2.1 | CS 3.2 | SS 3 |
| 4 | 7 | SQAT\_RL 3.2.2- 3.2.3 | CS 4.1 |  |
| 8 | SQAT\_RL 3.2.4- 3.2.5 | CS 4.2 | SS 4 |
| 5 | 9 | SQAT\_RL 4.1.1- 4.1.4 | CS 5.1 |  |
| 10 | SQAT\_RL 4.2.1- 4.3.1 | CS 5.2 | SS 5 |
| 6 | 11 | SQAT\_RL 4.3.2- 4.4.1 | CS 6.1 |  |
| 12 | SQAT\_RL 5.1.1- 5.1.4 | CS 6.2 | SS 6 |
| 7 | 13 | SQAT\_RL 5.2.1- 5.4.1 | CS 7.1 |  |
| 14 | SQAT\_RL 5.5.1- 5.6.2 | CS 7.2 | SS 7 |
| 8 | 15-16 | Revision |  |  |
| **Mid Semester Exam** | | | | |
| 9 | 17 | SQAT\_RL 6.1.1- 6.2.3 | CS 9.1 |  |
| 18 | SQAT\_RL 6.3.1- 6.4.2 | CS 9.2 | SS 6 |
| 10 | 19 | SQAT\_RL 7.1.1- 7.1.2 | CS 10.1 |  |
| 20 | SQAT\_RL 7.2.1- 7.3.4 | CS 10.2 | SS 7 |
| 11 | 21 | SQAT\_RL 7.4.1- 7.5.4 | CS 11.1 |  |
| 22 | SQAT\_RL 7.6.1 | CS 11.2 | SS 7 |
| 12 | 23 | SQAT\_RL 8.1.1- 8.2.2 | CS 12.1 |  |
| 24 | SQAT\_RL 8.3.1- 8.4.5 | CS 12.2 | SS 8 |
| 13 | 25 | SQAT\_RL 9.1.1- 9.2.4 | CS 13.1 |  |
| 26 | SQAT\_RL 9.3.1- 9.5.3 | CS 13.2 | SS 9 |
| 14 | 27 | SQAT\_RL 10.1.1- 10.2.3 | CS 14.1 |  |
| 28 | SQAT\_RL 10.3.1 | CS 14.2 | SS 10 |
| 15 | 29 | SQAT\_RL 11.1.1- 11.2.2 | CS 15.1 |  |
| 30 | SQAT\_RL 11.2.3-11.3.2 | CS 15.2 | SS 11 |
| 16 | 31-32 | Revision |  |  |
| **End Semester Exam** | | | | |

**Assignments**

Each participant or Group of participants will be given an assignment on a topic that was discussed in class. The assignment topics will be based on practical problems experienced or part of work items or tools used by collaborating organizations.

* Assignments will be take-home and deadline-driven, typically lasting 2 weeks. Participants are expected to spend at least 16 hours on the study, research, discussion, and preparation of the report and presentation.
* As part of the deliverables, participants will prepare a report and/or make a short presentation in class.

**Experiential Learning Components**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Topic** | **Objectives** | **Hands-On-Exercises** |
| 1 | SQA Planning and Design | Learn how to create a comprehensive test plan and design effective test cases. | Develop a test plan and design test cases for a given software application. |
| 2 | Automation Frameworks | Introduction to automation tools and frameworks | Automate test cases using a popular test automation tool. |
| 3 | Usability Testing | Understand the importance of user experience and usability in software. | Conduct usability testing on a software interface and provide recommendations for improvement. |
| 4 | Continuous Integration and Continuous Testing | Explore the concepts of continuous integration and continuous testing in SQA | Demonstration of CI/CD pipeline and automate the testing process for a software application. |

**Evaluation Scheme**:

Legend: EC = Evaluation Component (*Evaluation components can be tailored.)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Name** | **Type** | **Duration** | **Weight** |
| EC-1 | Quiz-I | Online | 1 Week | 5% |
| Quiz-II | Online | 1 Week | 5% |
| Assignment | Online | 2 Weeks | 10% |
| EC-2 | Mid-Semester Test | Open Book | 2 hours | 30% |
| EC3 | Comprehensive Exam | Open Book | 3 hours | 50% |