**Scheme of Evaluation (Marking Scheme) – SE ZG501 Mid-Semester Test (30 Marks Total)**

**Course Title:** Software Quality Assurance and Testing  
**Exam Type:** EC-2 Regular, Closed Book  
**Duration:** 2 Hours  
**Weightage:** 30%

**Question 1**  
**(i) Define and differentiate between defect, error, and failure**

• **Total Marks: 2**  
• **Evaluation Scheme:**

* **1 mark** for the **correct definition** of all three terms:
  + **Error**: A human mistake made during software development (e.g., a programmer writes incorrect logic).
  + **Defect (Bug)**: A flaw or fault in the software code or design caused by an error.
  + **Failure**: The observable incorrect behavior of the software when a defect is executed.
* **1 mark** for **clear differentiation using a given scenario**
  + Example scenario:

A developer misunderstands a requirement and writes incorrect code (**error**). This results in a calculation bug in the system (**defect**). When a user runs the application and sees the wrong output (**failure**).

* + **0.5 mark** for correctly applying the scenario to show the relationship among the three terms
  + **0.5 mark** for highlighting how they differ based on their role in the software lifecycle

**(ii) Methods to identify and fix issues**

* Total Marks: 3
* Evaluation:
  + 1 mark for identifying error detection methods (e.g., code review, unit testing)
  + 1 mark for identifying defect detection (e.g., functional testing, debugging)
  + 1 mark for identifying failure and proposing valid corrective strategies

**Question 2**  
**(a) Difference between Quality Assurance and Quality Control**

* Total Marks: 1
* Evaluation:
  + 0.5 mark for QA definition
  + 0.5 mark for QC definition

**(b)(i) Impact of unclear/incomplete requirements**

* Total Marks: 2
* Evaluation:
  + 1 mark for impact on quality (e.g., misinterpretation, delays)
  + 1 mark for scenario-based explanation

**(b)(ii) Best practices for defining requirements**

* Total Marks: 2
* Evaluation:
  + 0.5 mark each for any four relevant practices (e.g., stakeholder discussion, documentation, prototyping, use cases, review meetings)

**Question 3**  
**(a) Justification for investing in Software Quality Assurance**

* Total Marks: 3
* Evaluation:
  + 1 mark for cost-effectiveness
  + 1 mark for risk mitigation
  + 1 mark for long-term value

**(b) Prevention cost vs overall quality cost**

* Total Marks: 2
* Evaluation:
  + 1 mark for the role of prevention
  + 1 mark for connection to failure and overall cost

**Question 4**  
**(a) Three perspectives of McCall Model**

* Total Marks: 1
* Evaluation:
  + 0.33 mark each for: Product Revision, Product Transition, Product Operation

**(b) Primary focus of McCall Model**

* Total Marks: 1
* Evaluation:
  + 1 mark for correct identification (focus on product quality/internal perspective)

**(c) Role of quality measurement in acquisition**

* Total Marks: 1
* Evaluation:
  + 1 mark for explaining contractual, verification, or compliance benefits

**(d) PDCA cycle in ISO 9001**

* Total Marks: 1
* Evaluation:
  + 0.5 mark for process planning and execution
  + 0.5 mark for continuous improvement aspect

**(e) Five stages of ITIL service lifecycle**

* Total Marks: 1
* Evaluation:
  + 0.2 mark each for: Service Strategy, Service Design, Service Transition, Service Operation, Continual Service Improvement

**Question 5**

**(a) Static vs Dynamic Testing**

• **Total Marks: 2**  
• **Evaluation Scheme:**

* **1 mark** for explaining **Static Testing**
  + Include techniques (e.g., code reviews, walkthroughs, static analysis)
  + When it is used (early stages, before code execution)
* **1 mark** for explaining **Dynamic Testing**
  + Include techniques (e.g., unit testing, integration testing, system testing)
  + When it is used (later stages, after code is written and executable)

**→ Out of the 1 mark each for static and dynamic testing, 0.5 mark** should reflect the **difference** between them (e.g., static is without executing code, dynamic involves code execution).

**(b) Boundary Value Analysis for grade validation**

* Total Marks: 3
* Evaluation:
  + 0.5 mark each for any of the 6 valid BVA test cases with correct expected outcome

**Question 6**

**(a) Functional vs Non-functional Testing**

• **Total Marks: 2**  
• **Evaluation Scheme:**

* **1 mark** for explaining **Functional Testing**
  + Includes what it tests (system behavior/features)
  + Common techniques: unit testing, integration testing, system testing
  + Based on business/functional requirements
* **1 mark** for explaining **Non-functional Testing**
  + Includes what it tests (system performance, quality attributes)
  + Common types: performance testing, load testing, usability testing, security testing
  + Based on non-functional requirements

**→ Out of the 1 mark each, 0.5 mark** should reflect the **difference** between the two (e.g., functional = "what system does", non-functional = "how system behaves under certain

**(b) Types and advantages of automated testing**

* Total Marks: 3
* Evaluation:
  + 2 marks for describing at least three types of automated testing
  + 1 mark for listing at least three valid advantages