Sets of Software engineering

### Set A: Final Paper (Based on Most Repeated Questions)

**Full Marks: 100**  
**Pass Marks: 45**  
**Time: 3 hours**

#### **1. Software Engineering Fundamentals**

**a)** Define Software Engineering. Explain the **4 P’s** of Software Project Management.  
**b)** What is **Agile Project Management**? Why is the **Prototype Model** better than the **Waterfall Model**?  
*(7 + 8 = 15 marks)*

#### **2. Software Metrics and Estimation**

**a)** Calculate the **Function Point (FP)** for a project with the following data:

* Number of inputs: 44
* Number of outputs: 80
* Number of inquiries: 30
* Number of files: 20
* Number of external interfaces: 4  
  Assume all complexity adjustment values are average. If the productivity is 40 FP/PM and the labor rate is Rs. 27,440 per month, calculate the estimated project cost and effort.  
  **b)** Explain the **COCOMO Model** and calculate the effort, development time, and average staff size for a project of 200 KLOC (Organic type).  
  *(8 + 7 = 15 marks)*

#### **3. Software Requirements and Design**

**a)** What are **functional** and **non-functional requirements**? Explain the **requirement elicitation techniques**.  
**b)** Draw a **Level-1 DFD** for a **Library Management System** (include processes like Search Book, Issue Book, and Pay Fine).  
*(7 + 8 = 15 marks)*

#### **4. Software Testing and Quality Assurance**

**a)** Differentiate between **Verification** and **Validation**. Explain **Boundary Value Analysis (BVA)** with an example.  
**b)** What is **Formal Technical Review (FTR)**? Why is it important in **Software Quality Assurance (SQA)** activities?  
*(7 + 8 = 15 marks)*

#### **5. Software Configuration Management and Maintenance**

**a)** What is **Software Configuration Management (SCM)**? Explain the **change management process** in SCM.  
**b)** Define **software maintenance**. Explain the different types of software maintenance with suitable examples.  
*(7 + 8 = 15 marks)*

#### **6. Short Notes** (Any Two)

**a)** **Cyclomatic Complexity**  
**b)** **Data Dictionary**  
**c)** **Alpha and Beta Testing**  
*(2 x 5 = 10 marks)*

#### **7. Case Study**

**Scenario:** A **Metro Ticketing System** allows users to register, purchase tickets (single-journey, day pass, monthly pass), validate tickets, and receive real-time updates.  
**Tasks:**

* Identify the **classes** and model them in a **class diagram**.
* Identify the **use cases** and expand one use case (include name, actor, pre-conditions, post-conditions, description, and steps of execution).  
  *(15 marks)*

### Set B: Final Paper (Based on Most Repeated Questions)

**Full Marks: 100**  
**Pass Marks: 45**  
**Time: 3 hours**

#### **1. Software Engineering Fundamentals**

**a)** Define **Software Engineering**. Explain the **V-Model** with its advantages and disadvantages.  
**b)** What is **Agile Development**? Explain the **Scrum** software development process in detail.  
*(7 + 8 = 15 marks)*

#### **2. Software Metrics and Estimation**

**a)** Calculate the **Function Point (FP)** for a project with the following data:

* Number of inputs: 22
* Number of outputs: 45
* Number of inquiries: 6
* Number of files: 5
* Number of external interfaces: 2  
  Assume all complexity adjustment values are average. If the productivity is 37 FP/PM and the labor rate is $7,520 per month, calculate the estimated project cost and effort.  
  **b)** Explain the **COCOMO Model** and calculate the effort, development time, and average staff size for a project of 300 KLOC (Semi-Detached type).  
  *(8 + 7 = 15 marks)*

#### **3. Software Requirements and Design**

**a)** What is **Domain Analysis**? Explain the different components of the **Object-Oriented Analysis Model**.  
**b)** Draw a **Use Case Diagram** for an **ATM Machine** (include actors and all relationships).  
*(7 + 8 = 15 marks)*

#### **4. Software Testing and Quality Assurance**

**a)** What is **Control Structure Testing**? Explain **Basis Path Testing** with an example.  
**b)** Define **Software Reliability**. Explain the relationship between **MTBF**, **MTTF**, and **service availability**.  
*(7 + 8 = 15 marks)*

#### **5. Software Configuration Management and Maintenance**

**a)** What is **Software Reengineering**? Explain its process and need.  
**b)** Define **Software Reuse**. Discuss the benefits and problems associated with software reuse.  
*(7 + 8 = 15 marks)*

#### **6. Short Notes** (Any Two)

**a)** **MVC Framework**  
**b)** **Software Engineering Ethics**  
**c)** **Design Patterns**  
*(2 x 5 = 10 marks)*

#### **7. Case Study**

**Scenario:** A **Restaurant Management System** allows customers to order food through an application, which is served by a robot. The system also handles payments and feedback.  
**Tasks:**

* Draw a **Sequence Diagram** for the food ordering process.
* Identify the **classes** and model them in a **class diagram**.  
  *(15 marks)*

### Set C: Final Paper (Based on Most Repeated Questions)

**Full Marks: 100**  
**Pass Marks: 45**  
**Time: 3 hours**

#### **1. Software Engineering Fundamentals**

**a)** Define **Software Engineering**. Explain the **Waterfall Model** and its limitations.  
**b)** What is **Agile Methodology**? Explain the **Extreme Programming (XP)** model with its features.  
*(7 + 8 = 15 marks)*

#### **2. Software Metrics and Estimation**

**a)** Calculate the **Function Point (FP)** for a project with the following data:

* Number of inputs: 12
* Number of outputs: 8
* Number of inquiries: 5
* Number of files: 4
* Number of external interfaces: 3  
  Assume all complexity adjustment values are average. Show all steps.  
  **b)** Explain the **COCOMO Model** and calculate the effort, development time, and average staff size for a project of 200 KLOC (Embedded type).  
  *(8 + 7 = 15 marks)*

#### **3. Software Requirements and Design**

**a)** What is **Scenario-Based Modeling**? Explain with an example.  
**b)** Draw a **Level-0 DFD** for a **Student Registration System**.  
*(7 + 8 = 15 marks)*

#### **4. Software Testing and Quality Assurance**

**a)** What is **Alpha Testing** and **Beta Testing**? Explain the **software testing process model**.  
**b)** Define **Cyclomatic Complexity**. Calculate the cyclomatic complexity of the given code:

int fun(int x, int y) {  
 while(x != y) {  
 if(x > y) x = x - y;  
 else y = y - x;  
 }  
 return x;  
}

*(7 + 8 = 15 marks)*

#### **5. Software Configuration Management and Maintenance**

**a)** What is **Software Configuration Management (SCM)**? Explain the **version management process**.  
**b)** Define **Software Maintenance**. Explain the different types of software maintenance with suitable examples.  
*(7 + 8 = 15 marks)*

#### **6. Short Notes** (Any Two)

**a)** **Boundary Value Analysis (BVA)**  
**b)** **Software Reengineering**  
**c)** **Cloud-Based Software Engineering**  
*(2 x 5 = 10 marks)*

#### **7. Case Study**

**Scenario:** A **Movie Ticket Booking System** allows customers to book tickets online or at the theater. The system handles ticket validation, complementary food, and customer access.  
**Tasks:**

* Draw a **Context Diagram** and **Level-1 DFD** for the system.
* Identify the **use cases** and expand one use case (include name, actor, pre-conditions, post-conditions, description, and steps of execution).  
  *(15 marks)*

### Analysis of Teacher's Mindset:

1. **Most Repeated Topics**:
   * **Function Point Calculation**, **COCOMO Model**, **Agile Methodology**, **DFD**, **Use Case Diagram**, **Cyclomatic Complexity**, **Verification vs Validation**, **Software Maintenance**, **SCM**, and **Testing Techniques** are the most repeated topics across all papers.
   * These topics are essential for students to score at least 45 marks.
2. **Balanced Difficulty**:
   * The papers are designed to have a mix of theoretical and practical questions, ensuring that students with different strengths can score well.
   * Case studies and diagrams are included to test practical application skills.
3. **Focus on Core Concepts**:
   * The papers focus on core software engineering concepts like **metrics**, **design**, **testing**, and **management**, which are crucial for both exams and real-world applications.
4. **Encouraging Conceptual Understanding**:
   * The questions are designed to test the student's understanding of concepts rather than rote learning, ensuring that students who grasp the concepts can score well.

By following these sets, students can confidently prepare for their exams and achieve the minimum passing marks of 45.