

1. a) What is the importance of software Engineering? Briefly describe what should be steps taken under the process of developing a software system.

b) Explain the principles which play a major role in development of software.

c) Describe the components and quality which is necessary for the documents of software specification.

d) What are the benefits of metrics in software engineering?
2. a) Define the blue print methodology.

b) Give the benefits of verification and validation in software development and tell about the techniques of verification and validation in the process of software development.

c) Define the meaning of quality assurance. Explain the role of testing in quality assurance.

d) Write short note on software failure, black box testing, white box testing and stress Testing.
3. a) Explain the various types of models which used in Software Engineering.

b) Write down the concept of data flow diagram.

c) Describe the objectives of a) coding b) structured programming in terms of software engineering.

d) You have been asked to develop a system that will help with planning large-scale events and parties such as weddings, graduation celebrations, and birthday parties. Using an activity diagram, model the process context for such a system that shows the activities involved in planning a party (booking a venue, organizing invitations, etc.) and the system elements that might be used at each stage
4. a) At the end of their study program, students in a software engineering course are typically expected to complete a major project. Explain how the agile methodology may be very useful for the students to use in this case.

b) To reduce costs and the environmental impact of commuting, your company decides to close a number of offices and to provide support for staff to work from home. However, the senior management who introduce the policy, is unaware that software is developed using Scrum. Explain how you could use technology to support Scrum in a distributed environment to make this possible. What problems are you likely to encounter using this approach?

c) You have taken a job with a software user who has contracted your previous employer to develop a system for them. You discover that your company's interpretation of the requirements is different from the interpretation taken by your previous employer. Discuss what you should do in such a situation? You know that the costs to your current employer will increase if the ambiguities are not resolved. However, you also have a responsibility of confidentiality to your previous employer.
5. a) Why a software project manager require for a software Industry? Show the responsibilities of project manager.

b) Briefly describe the project planning, scope management and project estimation in terms of software management activities.

c) Write short note about project scheduling, resource management, project communication management, configuration management.

d) Suppose you are a project manager of XYZ software development team. Company authority asks you to present a project schedule for a client. Draw a Gantt chart and PERT chart for that software scheduling.
6. a) What is Entity-Relationship model? Define data dictionary and show the requirement of data dictionary.

b) Differentiate among software design strategies like structured design, function oriented design and object oriented design.

c) Compare between Top-down design and Bottom-up design.

d) Show the user interface design activities.
7. a) List out the characteristics of good software.

b) Discuss about the Big Bang model with its advantage and disadvantage. Show the V-Model structure.

c) Describe about the data flow diagram components with their three levels of architecture.  
Or  
Write down each of the clauses in the ACM/IEEE Code of ethics for software engineers.
8. a) Imagine that a government wants a software program that helps to keep track of the utilization of the country's vast mineral resources. Although the requirements put forward by the government were not very clear, a software company was tasked with the development of a prototype. The government found the prototype impressive, and asked it be extended to be the actual system that would be used. Discuss the pros and cons of taking this approach

b) Suggest the most appropriate generic software process model that might be used as a basis for managing the development of the following systems. Explain your answer according to the type of system being developed:

  - A system to control antilock braking in a car
  - A virtual reality system to support software maintenance
  - A university accounting system that replaces an existing system
  - An interactive travel planning system that helps users plan journeys with the lowest environmental impact

c) Should there be a separate profession of 'software architect' whose role is to work independently with a customer to design the software system architecture? A separate software company would then implement the system. What might be the difficulties of establishing such a profession?

## Answer 1:

### (a) Importance of Software Engineering

**Importance:** Software Engineering ensures the creation of reliable and efficient software that meets user needs.

**প্রয়োজনীয়তা:** সফটওয়্যার ইঞ্জিনিয়ারিং নির্ভরযোগ্য এবং কার্যকর সফটওয়্যার তৈরি নিশ্চিত করে যা ব্যবহারকারীদের প্রয়োজনীয়তা পূরণ করে।

### Steps in Development:

1. **Requirement Analysis:** Understand what the users need.  
**প্রয়োজন বিশ্লেষণ:** ব্যবহারকারীরা কি প্রয়োজন তা বুঝুন।
2. **System Design:** Plan the software structure.  
**সিস্টেম ডিজাইন:** সফটওয়্যারের কাঠামো পরিকল্পনা করুন।
3. **Implementation:** Write the code.  
**বাস্তবায়ন:** কোড লিখুন।
4. **Testing:** Check for errors and ensure it works.  
**পরীক্ষা:** ত্রুটি পরীক্ষা করুন এবং এটি কাজ করছে তা নিশ্চিত করুন।
5. **Deployment:** Release the software to users.  
**মোতায়েন:** সফটওয়্যারটি ব্যবহারকারীদের কাছে প্রকাশ করুন।
6. **Maintenance:** Update and fix issues as needed.  
**রক্ষণাবেক্ষণ:** প্রয়োজন অনুসারে আপডেট এবং সমস্যা সমাধান করুন।

### b. Principles in Software Development

#### Key Principles:

1. **Modularity:** Break the software into smaller parts.  
**মূলনীতি:** সফটওয়্যারকে ছোট ছোট অংশে বিভক্ত করুন।
2. **Reusability:** Use existing components to save time.  
**পুনঃব্যবহারযোগ্যতা:** সময় সাশ্রয়ের জন্য বিদ্যমান উপাদানগুলি ব্যবহার করুন।
3. **Maintainability:** Make it easy to fix and update.  
**রক্ষণাবেক্ষণযোগ্যতা:** এটি মেরামত এবং আপডেট করা সহজ করুন।
4. **Scalability:** Allow the software to grow with demand.  
**স্কেলেবিলিটি:** সফটওয়্যারটিকে চাহিদার সাথে বৃদ্ধি হতে দিন।
5. **User-Centered Design:** Focus on making it easy for users.  
**ব্যবহারকারী-কেন্দ্রিক ডিজাইন:** ব্যবহারকারীদের জন্য সহজ করে তোলার উপর মনোযোগ দিন।

### c. Components and Quality in Software Specifications

#### Components:

1. **Functional Requirements:** What the software should do.  
**কার্যকরী প্রয়োজনীয়তা:** সফটওয়্যারটি কি করা উচিত।

2. **Non-Functional Requirements:** Performance and security needs.  
অকার্যকরী প্রয়োজনীয়তা: কর্মক্ষমতা এবং সুরক্ষার প্রয়োজন।
3. **Design Specifications:** How the software is structured.  
ডিজাইন স্পেসিফিকেশন: সফটওয়্যারটি কিভাবে গঠন করা হয়েছে।
4. **Use Cases:** Scenarios of user interactions.  
ব্যবহারের ক্ষেত্রে: ব্যবহারকারীর মিথস্ক্রিয়ার দৃশ্যপট।

#### Quality Attributes:

- **Correctness:** Works as intended.  
সঠিকতা: উদ্দেশ্যমত কাজ করে।
- **Efficiency:** Uses resources wisely.  
কার্যকারিতা: সংস্থানগুলি দক্ষতার সাথে ব্যবহার করে।
- **Reliability:** Consistent performance.  
নির্ভরযোগ্যতা: ধারাবাহিক কর্মক্ষমতা।
- **Usability:** Easy to use.  
ব্যবহারযোগ্যতা: ব্যবহার করা সহজ।
- **Maintainability:** Simple to update.  
রক্ষণাবেক্ষণযোগ্যতা: আপডেট করা সহজ।

#### d. Benefits of Metrics in Software Engineering

##### Benefits:

1. **Performance Measurement:** Identify improvement areas.  
উপকারিতা: কর্মক্ষমতা পরিমাপ: উন্নতির ক্ষেত্র চিহ্নিত করুন।
2. **Progress Tracking:** Monitor development stages.  
অগ্রগতি ট্র্যাকিং: উন্নয়নের পর্যায়গুলি পর্যবেক্ষণ করুন।
3. **Quality Assurance:** Reduce defects and improve satisfaction.  
গুণমানের নিশ্চয়তা: ত্রুটি হ্রাস করুন এবং সন্তুষ্টি বাড়ান।
4. **Risk Management:** Spot potential issues early.  
ঝুঁকি ব্যবস্থাপনা: সম্ভাব্য সমস্যাগুলি শুরুর দিকে চিহ্নিত করুন।
5. **Resource Management:** Use resources effectively.  
সম্পদ ব্যবস্থাপনা: কার্যকরভাবে সম্পদ ব্যবহার করুন।

## Answer 2:

### Software Engineering - Key Concepts

#### 2(a) Blue Print Methodology

**Definition:** A structured approach in software development that involves creating a detailed design or blueprint before coding. This ensures all aspects are considered, reducing errors and improving efficiency.

**বর্ণনা:** সফটওয়্যার উন্নয়নে একটি কাঠামোগত পদ্ধতি যা কোডিংয়ের আগে একটি বিস্তারিত

ডিজাইন বা নীলনকশা তৈরি করা অন্তর্ভুক্ত করে। এটি নিশ্চিত করে যে সমস্ত দিকগুলি বিবেচিত হয়েছে, ভুলগুলি হ্রাস করে এবং কার্যকারিতা বাড়ায়।

---

## 2(b) Benefits of Verification and Validation

### Benefits:

1. **Quality Improvement:** Ensures software meets requirements and is defect-free.  
**গুণমানের উন্নতি:** নিশ্চিত করে যে সফটওয়্যারটি প্রয়োজনীয়তা পূরণ করে এবং ত্রুটিমুক্ত।
2. **Cost Efficiency:** Identifies issues early, reducing fix costs.  
**খরচের দক্ষতা:** শুরুর দিকে সমস্যা চিহ্নিত করে, মেরামতের খরচ কমায়।
3. **User Satisfaction:** Confirms the product meets user needs.  
**ব্যবহারকারীর সন্তুষ্টি:** নিশ্চিত করে যে পণ্যটি ব্যবহারকারীর প্রয়োজনীয়তা পূরণ করে।
4. **Risk Reduction:** Minimizes the chances of project failure.  
**ঝুঁকি হ্রাস:** প্রকল্পের ব্যর্থতার সম্ভাবনা কমায়।

### Techniques:

- Code Reviews  
**কোড পর্যালোচনা**
  - Testing (unit, integration, system)  
**পরীক্ষা (ইউনিট, ইন্টিগ্রেশন, সিস্টেম)**
  - Static Analysis  
**স্ট্যাটিক বিশ্লেষণ**
- 

## 2(c) Quality Assurance

**Definition:** A systematic process to ensure software meets requirements and is defect-free.

**বর্ণনা:** একটি পদ্ধতিগত প্রক্রিয়া যা নিশ্চিত করে যে সফটওয়্যারটি প্রয়োজনীয়তা পূরণ করে এবং ত্রুটিমুক্ত।

### Role of Testing:

- Identifies defects before release.  
**রিলিজের আগে ত্রুটিগুলি চিহ্নিত করে।**
  - Ensures compliance with standards.  
**মানদণ্ডের সাথে সামঞ্জস্য নিশ্চিত করে।**
  - Validates intended functionality.  
**প্রত্যাশিত কার্যকারিতা যাচাই করে।**
-

## 2(d) Key Testing Concepts

- **Software Failure:** When software malfunctions or does not perform as intended.  
**সফটওয়্যার ব্যর্থতা:** যখন সফটওয়্যারটি ত্রুটি করে বা প্রত্যাশিতভাবে কাজ করে না।
- **Black Box Testing:** Evaluates functionality without knowing internal code.  
**ব্ল্যাক বক্স টেস্টিং:** অভ্যন্তরীণ কোড না জানিয়ে কার্যকারিতা মূল্যায়ন করে।
- **White Box Testing:** Tests internal workings of the software.  
**হোয়াইট বক্স টেস্টিং:** সফটওয়্যারের অভ্যন্তরীণ কার্যকারিতা পরীক্ষা করে।
- **Stress Testing:** Assesses system performance under extreme conditions.  
**স্ট্রেস টেস্টিং:** চরম পরিস্থিতিতে সিস্টেমের কার্যকারিতা মূল্যায়ন করে।

## 2(a) Types of Models Used in Software Engineering

1. **Waterfall Model:** A linear process where each phase must be completed before the next begins.  
**ওটারফল মডেল:** একটি লিনিয়ার প্রক্রিয়া যেখানে প্রতিটি পর্যায় সম্পূর্ণ করতে হয় পরবর্তী পর্যায় শুরু করার আগে।
2. **Agile Model:** An iterative approach that allows for flexibility and frequent updates based on user feedback.  
**অ্যাজাইল মডেল:** একটি পুনরাবৃত্তিমূলক পদ্ধতি যা ব্যবহারকারীর প্রতিক্রিয়ার উপর ভিত্তি করে নমনীয়তা এবং নিয়মিত আপডেটের অনুমতি দেয়।
3. **V-Model:** Focuses on testing each phase of development to ensure quality.  
**ভি-মডেল:** গুণমান নিশ্চিত করতে উন্নয়নের প্রতিটি পর্যায়ে পরীক্ষার উপর জোর দেয়।
4. **Spiral Model:** Combines iterative development with risk assessment for complex projects.  
**স্পাইরাল মডেল:** জটিল প্রকল্পগুলির জন্য পুনরাবৃত্তিমূলক উন্নয়ন এবং ঝুঁকি মূল্যায়নকে একত্রিত করে।
5. **Incremental Model:** Develops the system in small parts, allowing for gradual improvement and user feedback.  
**ইনক্রিমেন্টাল মডেল:** ধীরে ধীরে উন্নতির অনুমতি দেয় এবং ব্যবহারকারীর প্রতিক্রিয়া গ্রহণের জন্য সিস্টেমটি ছোট অংশে উন্নয়ন করে।

## 2(b) Concept of Data Flow Diagram (DFD)

A Data Flow Diagram (DFD) shows how data moves through a system. Key components are:

**ডেটা ফ্লো ডায়াগ্রাম (DFD)** একটি সিস্টেমে তথ্য কিভাবে চলাচল করে তা দেখায়। মূল উপাদানগুলি হল:

- **Processes:** Activities that transform data.  
**প্রক্রিয়া:** তথ্য পরিবর্তনের কার্যক্রম।
- **Data Flows:** Arrows indicating the direction of data movement.  
**ডেটা ফ্লো:** তথ্যের গতির দিকে নির্দেশ করে এমন তীর।
- **Data Stores:** Locations where data is stored.  
**ডেটা স্টোর:** যেখানে তথ্য সংরক্ষিত হয় সেই স্থানের চিত্র।

- **External Entities:** Sources or destinations of data outside the system.  
বহিরাগত সত্তা: সিস্টেমের বাইরে তথ্যের উৎস বা গন্তব্য।

## 2(c) Objectives of Coding and Structured Programming

### a) Coding Objectives:

- Turn designs into working code.  
কোডিংয়ের উদ্দেশ্য:
  - ডিজাইনগুলিকে কার্যকরী কোডে রূপান্তরিত করা।
- Make the code easy to read.
  - কোডটি পড়তে সহজ করা।
- Ensure the code is efficient and maintainable.
  - কোডটি কার্যকর এবং রক্ষণাবেক্ষণযোগ্য করা নিশ্চিত করা।

### b) Structured Programming Objectives:

- Break code into smaller, manageable parts (modularity).  
গঠনমূলক প্রোগ্রামিংয়ের উদ্দেশ্য:
  - কোডকে ছোট, পরিচালনাযোগ্য অংশে ভাগ করা (মডুলারিটি)।
- Use clear control structures (loops, functions).
  - পরিষ্কার নিয়ন্ত্রণ কাঠামো (লুপ, ফাংশন) ব্যবহার করা।
- Handle errors effectively.
  - ত্রুটিগুলি কার্যকরভাবে পরিচালনা করা।
- Promote code reuse to avoid duplication.
  - পুনরাবৃত্তি এড়াতে কোড পুনঃব্যবহারের উত্সাহ দেওয়া।

## 2(d) Activity Diagram for Planning Large Scale Events

1. **Identify Event Type:** Decide on the type of event (wedding, graduation, birthday).  
ইভেন্টের ধরন চিহ্নিত করুন: ইভেন্টের ধরন (বিবাহ, স্নাতক, জন্মদিন) সম্পর্কে সিদ্ধান্ত নিন।
2. **Book Venue:** Reserve the location.  
ভেন্যু বুক করুন: স্থানটি সংরক্ষণ করুন।
3. **Organize Invitations:** Create and send invitations.  
নিমন্ত্রণগুলি সংগঠিত করুন: নিমন্ত্রণ তৈরি এবং পাঠান।
4. **Arrange Catering:** Select food services.  
কেটারিং ব্যবস্থা করুন: খাদ্য পরিষেবাগুলি নির্বাচন করুন।
5. **Plan Entertainment:** Hire entertainers.  
বিনোদন পরিকল্পনা করুন: বিনোদনকারী নিয়োগ করুন।
6. **Manage Guest List:** Track RSVPs.  
অতিথির তালিকা পরিচালনা করুন: RSVPs ট্র্যাক করুন।
7. **Finalize Details:** Confirm all arrangements.  
বিস্তারিত চূড়ান্ত করুন: সমস্ত ব্যবস্থা নিশ্চিত করুন।

8. **Conduct Event:** Carry out the event.

**ইভেন্ট পরিচালনা করুন:** ইভেন্টটি পরিচালনা করুন।

#### System Elements:

- **User Interface:** For entering information.  
**সিস্টেম উপাদান:**
  - **ব্যবহারকারীর ইন্টারফেস:** তথ্য প্রবেশ করার জন্য।
- **Database:** To store guest and event data.
  - **ডেটাবেস:** অতিথি এবং ইভেন্টের তথ্য সংরক্ষণের জন্য।
- **Notification System:** To send reminders.
  - **নোটিফিকেশন সিস্টেম:** স্মরণ করিয়ে দেওয়ার জন্য।
- **Calendar:** To manage timelines.
  - **ক্যালেন্ডার:** সময়সীমা পরিচালনা করতে।

### Answer 3:

#### 3(a) Types of Models Used in Software Engineering

1. **Waterfall Model:** A linear process where each phase must be completed before the next begins.  
**ওটারফল মডেল:** একটি লিনিয়ার প্রক্রিয়া যেখানে প্রতিটি পর্যায় সম্পূর্ণ করতে হয় পরবর্তী পর্যায় শুরু করার আগে।
2. **Agile Model:** An iterative approach that allows for flexibility and frequent updates based on user feedback.  
**অ্যাজাইল মডেল:** একটি পুনরাবৃত্তিমূলক পদ্ধতি যা ব্যবহারকারীর প্রতিক্রিয়ার উপর ভিত্তি করে নমনীয়তা এবং নিয়মিত আপডেটের অনুমতি দেয়।
3. **V-Model:** Focuses on testing each phase of development to ensure quality.  
**ভি-মডেল:** গুণমান নিশ্চিত করতে উন্নয়নের প্রতিটি পর্যায়ে পরীক্ষার উপর জোর দেয়।
4. **Spiral Model:** Combines iterative development with risk assessment for complex projects.  
**স্পাইরাল মডেল:** জটিল প্রকল্পগুলির জন্য পুনরাবৃত্তিমূলক উন্নয়ন এবং ঝুঁকি মূল্যায়নকে একত্রিত করে।
5. **Incremental Model:** Develops the system in small parts, allowing for gradual improvement and user feedback.  
**ইনক্রিমেন্টাল মডেল:** ধীরে ধীরে উন্নতির অনুমতি দেয় এবং ব্যবহারকারীর প্রতিক্রিয়া গ্রহণের জন্য সিস্টেমটি ছোট অংশে উন্নয়ন করে।

#### 3(b) Concept of Data Flow Diagram (DFD)

A Data Flow Diagram (DFD) shows how **data moves through a system**. Key components are:

**ডেটা ফ্লো ডায়াগ্রাম (DFD)** একটি সিস্টেমে তথ্য কিভাবে চলাচল করে তা দেখায়। মূল উপাদানগুলি হল:



- **Processes:** Activities that transform data.  
প্রক্রিয়া: তথ্য পরিবর্তনের কার্যক্রম।
- **Data Flows:** Arrows indicating the direction of data movement.  
ডেটা ফ্লো: তথ্যের গতির দিকে নির্দেশ করে এমন তীর।
- **Data Stores:** Locations where data is stored.  
ডেটা স্টোর: যেখানে তথ্য সংরক্ষিত হয় সেই স্থানের চিত্র।
- **External Entities:** Sources or destinations of data outside the system.  
বহিরাগত সত্তা: সিস্টেমের বাইরে তথ্যের উৎস বা গন্তব্য।

### 3(c) Objectives of Coding and Structured Programming

#### a) Coding Objectives:

- Turn designs into working code.  
কোডিংয়ের উদ্দেশ্য:  
◦ ডিজাইনগুলিকে কার্যকরী কোডে রূপান্তরিত করা।
- Make the code easy to read.  
◦ কোডটি পড়তে সহজ করা।
- Ensure the code is efficient and maintainable.  
◦ কোডটি কার্যকর এবং রক্ষণাবেক্ষণযোগ্য করা নিশ্চিত করা।

#### b) Structured Programming Objectives:

- Break code into smaller, manageable parts (modularity).  
গঠনমূলক প্রোগ্রামিংয়ের উদ্দেশ্য:  
◦ কোডকে ছোট, পরিচালনাযোগ্য অংশে ভাগ করা (মডুলারিটি)।
- Use clear control structures (loops, functions).  
◦ পরিষ্কার নিয়ন্ত্রণ কাঠামো (লুপ, ফাংশন) ব্যবহার করা।
- Handle errors effectively.  
◦ ত্রুটিগুলি কার্যকরভাবে পরিচালনা করা।
- Promote code reuse to avoid duplication.  
◦ পুনরাবৃত্তি এড়াতে কোড পুনঃব্যবহারের উত্সাহ দেওয়া।

### 3(d) Activity Diagram for Planning Large Scale Events

1. **Identify Event Type:** Decide on the type of event (wedding, graduation, birthday).  
ইভেন্টের ধরন চিহ্নিত করুন: ইভেন্টের ধরন (বিবাহ, স্নাতক, জন্মদিন) সম্পর্কে সিদ্ধান্ত নিন।
2. **Book Venue:** Reserve the location.  
ভেন্যু বুক করুন: স্থানটি সংরক্ষণ করুন।
3. **Organize Invitations:** Create and send invitations.  
নিমন্ত্রণগুলি সংগঠিত করুন: নিমন্ত্রণ তৈরি এবং পাঠান।
4. **Arrange Catering:** Select food services.  
কেটারিং ব্যবস্থা করুন: খাদ্য পরিষেবাগুলি নির্বাচন করুন।



5. **Plan Entertainment:** Hire entertainers.  
বিনোদন পরিকল্পনা করুন: বিনোদনকারী নিয়োগ করুন।
6. **Manage Guest List:** Track RSVPs.  
অতিথির তালিকা পরিচালনা করুন: RSVPs ট্র্যাক করুন।
7. **Finalize Details:** Confirm all arrangements.  
বিস্তারিত চূড়ান্ত করুন: সমস্ত ব্যবস্থা নিশ্চিত করুন।
8. **Conduct Event:** Carry out the event.  
ইভেন্ট পরিচালনা করুন: ইভেন্টটি পরিচালনা করুন।

#### System Elements:

- **User Interface:** For entering information.  
সিস্টেম উপাদান:
  - ব্যবহারকারীর ইন্টারফেস: তথ্য প্রবেশ করার জন্য।
- **Database:** To store guest and event data.
  - ডেটাবেস: অতিথি এবং ইভেন্টের তথ্য সংরক্ষণের জন্য।
- **Notification System:** To send reminders.
  - নোটিফিকেশন সিস্টেম: স্মরণ করিয়ে দেওয়ার জন্য।
- **Calendar:** To manage timelines.
  - ক্যালেন্ডার: সময়সীমা পরিচালনা করতে।

## Answer 4:

2F 2T SL

Small Steps: Break projects into small tasks and deliver them regularly.

### 4(a) Agile Methodology for Students

Flexibility: Easily adapt to changes based on feedback.

#### Agile Methodology:

Teamwork: Collaborate in diverse groups with daily check-ins.

- **Flexible:** Can change plans easily.  
নমনীয়: পরিকল্পনা সহজেই পরিবর্তন করা যায়।
- **Teamwork:** Students work well together.  
দলবদ্ধতা: শিক্ষার্থীরা ভালোভাবে একসঙ্গে কাজ করে।
- **User Feedback:** Adjusts based on user needs.  
ব্যবহারকারীর প্রতিক্রিয়া: ব্যবহারকারীর চাহিদার ভিত্তিতে মানিয়ে নেয়।

Feedback: Get input from instructors to improve.

Learning: Reflect on progress and apply lessons.

Time Management: Use sprints to meet deadlines and prioritize tasks.

### 4(b) Supporting Scrum in a Distributed Environment

#### Using Technology:

- **Communication Tools:** Use Zoom or Slack for meetings.  
যোগাযোগের উপকরণ: বৈঠকের জন্য জুম বা স্ল্যাক ব্যবহার করুন।
- **Task Tracking:** Use Jira or Trello to keep track of tasks.  
কাজ ট্র্যাকিং: কাজ ট্র্যাক করতে জিরা বা ট্রেলো ব্যবহার করুন।

Scrum is a framework used in Agile methodology to help teams work together more effectively. It focuses on breaking down projects into small, manageable parts (called "sprints") and encourages regular communication and feedback.

#### Problems:

- **Time Zones:** Different times can make it hard to meet.  
টাইম জোন: ভিন্ন সময়ে সাক্ষাৎ করা কঠিন হতে পারে।
- **Misunderstandings:** May lead to confusion.  
ভুল বোঝাবুঝি: বিভ্রান্তির কারণ হতে পারে।

#### 4(c) Handling Different Interpretations of Requirements

**Steps:** Resolve the requirement differences by facilitating communication between both companies, while maintaining confidentiality and staying neutral.

1. **Talk About Requirements:** Discuss with the team.  
প্রয়োজনীয়তা নিয়ে কথা বলুন: দলের সাথে আলোচনা করুন।
2. **Note Differences:** Write down what doesn't match.  
পার্থক্য নোট করুন: যা মিলছে না তা লিখুন।
3. **Propose Solutions:** Suggest ways to fix issues.  
সমাধান প্রস্তাব করুন: সমস্যা সমাধানের উপায় সুপারিশ করুন।
4. **Respect Privacy:** Keep past agreements confidential.  
গোপনীয়তা রক্ষা করুন: পূর্ববর্তী চুক্তি গোপন রাখুন।

Clarify Requirements: Discuss and resolve any requirement confusion between both parties.

Facilitate Communication: Ensure proper communication between the two companies to address the issue.

Stay Neutral: Don't take sides; focus on solving the problem.

Maintain Confidentiality: Do not disclose any confidential information from your previous employer.

### Answer 5:

#### 5(a) Importance of a Software Project Manager

##### Why Required:

A software project manager is crucial for guiding the project from start to finish. They ensure the project meets deadlines, stays within budget, and meets quality standards.

##### Responsibilities:

1. **Planning:** Create a project plan and timeline.  
পরিকল্পনা: একটি প্রকল্প পরিকল্পনা এবং সময়সূচী তৈরি করুন।
2. **Monitoring:** Track progress and adjust plans as needed.  
Monitoring: অগ্রগতির অনুসরণ করুন এবং প্রয়োজন অনুযায়ী পরিকল্পনাগুলি সামঞ্জস্য করুন।
3. **Team Management:** Lead and support the development team.  
দল ব্যবস্থাপনা: উন্নয়ন দলের নেতৃত্ব দিন এবং সমর্থন করুন।
4. **Stakeholder Communication:** Communicate with clients and stakeholders.  
হিসাবদারি যোগাযোগ: ক্লায়েন্ট এবং স্টেকহোল্ডারদের সাথে যোগাযোগ করুন।
5. **Risk Management:** Identify and mitigate project risks.  
ঝুঁকি ব্যবস্থাপনা: প্রকল্পের ঝুঁকিগুলি চিহ্নিত করুন এবং হ্রাস করুন।

#### 5(b) Project Planning, Scope Management, and Estimation

##### Project Planning:

Defining objectives and creating a roadmap to achieve them.

**প্রকল্প পরিকল্পনা:** উদ্দেশ্যগুলি নির্ধারণ করা এবং সেগুলি অর্জনের জন্য একটি রোডম্যাপ তৈরি করা।

**Scope Management:**

Ensuring the project includes all necessary tasks and avoids scope creep (unplanned additions).

**স্কোপ ব্যবস্থাপনা:** প্রকল্পে সমস্ত প্রয়োজনীয় কাজ অন্তর্ভুক্ত আছে তা নিশ্চিত করা এবং অপ্রত্যাশিত সংযোজন এড়ানো।

**Project Estimation:**

Estimating time and resources needed to complete tasks.

**প্রকল্প অনুমান:** কাজ সম্পন্ন করার জন্য সময় এবং সম্পদের প্রয়োজন অনুমান করা।

### 5(c) Project Management Activities

1. **Project Scheduling:** Creating a timeline for project tasks.

**প্রকল্প সময়সূচী:** প্রকল্পের কাজগুলির জন্য একটি সময়সূচী তৈরি করা।

2. **Resource Management:** Allocating resources like team members and tools effectively.

**সম্পদ ব্যবস্থাপনা:** দলের সদস্য এবং সরঞ্জামের মতো সম্পদগুলি কার্যকরভাবে বরাদ্দ করা।

3. **Project Communication Management:** Ensuring clear communication among team members and stakeholders.

**প্রকল্প যোগাযোগ ব্যবস্থাপনা:** দলের সদস্য এবং স্টেকহোল্ডারদের মধ্যে পরিষ্কার যোগাযোগ নিশ্চিত করা।

4. **Configuration Management:** Managing changes to software and ensuring that the project remains consistent.

**কনফিগারেশন ব্যবস্থাপনা:** সফটওয়্যার পরিবর্তনগুলি পরিচালনা করা এবং প্রকল্পটি সঙ্গতিপূর্ণ রাখা নিশ্চিত করা।

### 5(d) PERT Chart for Software Scheduling

**Example PERT Chart Activities:**

1. **Requirement Gathering (2 weeks)**  
প্রয়োজনীয়তা সংগ্রহ (২ সপ্তাহ)
2. **Design Phase (3 weeks)**  
ডিজাইন ধাপ (৩ সপ্তাহ)
3. **Development Phase (4 weeks)**  
উন্নয়ন ধাপ (৪ সপ্তাহ)
4. **Testing Phase (2 weeks)**  
পরীক্ষণ ধাপ (২ সপ্তাহ)
5. **Deployment (1 week)**  
প্রস্তুতি (১ সপ্তাহ)

## Overall Timeline:

Total project duration: 12 weeks.

## Answer 6:

### 6(a) Entity-Relationship Model & Data Dictionary

#### Entity-Relationship Model:

The Entity-Relationship (ER) model is a diagrammatic representation of the data structure of a system. It shows how entities (like users, products) relate to each other.

#### এন্টিটি-রিলেশনশিপ মডেল:

এন্টিটি-রিলেশনশিপ (ER) মডেল একটি সিস্টেমের ডেটা স্ট্রাকচারের একটি চিত্রায়িত উপস্থাপনা। এটি দেখায় কীভাবে এন্টিটিগুলি (যেমন ব্যবহারকারী, পণ্য) একে অপরের সাথে সম্পর্কিত।

#### Data Dictionary:

A data dictionary is a collection of metadata that describes the structure, relationships, and constraints of the data in the database. It includes definitions of data elements, data types, and formats.

#### ডেটা ডিকশনারি:

একটি ডেটা ডিকশনারি হল একটি মেটাডেটার সংগ্রহ যা ডেটাবেসে ডেটার কাঠামো, সম্পর্ক এবং বাধা বর্ণনা করে। এটি ডেটা উপাদানের সংজ্ঞা, ডেটা প্রকার এবং ফর্ম্যাট অন্তর্ভুক্ত করে।

#### Requirements of Data Dictionary:

- Consistency:** Ensures uniform definitions across the database.  
**একরূপতা:** ডেটাবেস জুড়ে একক সংজ্ঞা নিশ্চিত করে।
- Standardization:** Establishes naming conventions and data formats.  
**মানকরণ:** নামকরণ স্ট্যান্ডার্ড এবং ডেটা ফর্ম্যাট প্রতিষ্ঠা করে।
- Documentation:** Provides a comprehensive reference for developers and users.  
**ডকুমেন্টেশন:** উন্নয়নকারী এবং ব্যবহারকারীদের জন্য একটি ব্যাপক রেফারেন্স প্রদান করে।

### 6(b) Differentiating Design Strategies

#### Structured Design:

Focuses on decomposing the system into smaller, manageable modules. It emphasizes a clear flow of control and data.

#### স্ট্রাকচারড ডিজাইন:

সিস্টেমটিকে ছোট, পরিচালনাযোগ্য মডিউলগুলিতে বিভক্ত করার উপর দৃষ্টি নিবদ্ধ করে। এটি নিয়ন্ত্রণ এবং ডেটার পরিষ্কার প্রবাহকে গুরুত্ব দেয়।

#### Object-Oriented Design:

Uses objects to represent data and methods. It emphasizes encapsulation, inheritance, and polymorphism.

## অবজেক্ট-ওরিয়েন্টেড ডিজাইন:

ডেটা এবং পদ্ধতিগুলি উপস্থাপন করতে অবজেক্ট ব্যবহার করে। এটি এনক্যাপসুলেশন, ইনহেরিট্যান্স এবং পলিমরফিজমকে গুরুত্ব দেয়।

### 6(c) Top-down vs. Bottom-up Design

#### Top-down Design:

Starts with the highest level of abstraction and breaks it down into smaller components.

#### টপ-ডাউন ডিজাইন:

সর্বোচ্চ স্তরের বিমূর্ততা দিয়ে শুরু করে এবং এটিকে ছোট উপাদানে বিভক্ত করে।

#### Bottom-up Design:

Begins with the smallest components and integrates them to form larger systems.

#### বটম-আপ ডিজাইন:

ছোট ছোট উপাদানগুলি দিয়ে শুরু করে এবং সেগুলিকে একত্রিত করে বৃহত্তর সিস্টেম গঠন করে।

### 6(d) User Interface Design Activities

1. **User Research:** Understand user needs and preferences.  
**ব্যবহারকারী গবেষণা:** ব্যবহারকারীর চাহিদা এবং পছন্দ বোঝা।
2. **Wireframing:** Create visual representations of the interface layout.  
**ওয়াयरফ্রেমিং:** ইন্টারফেস লেআউটের চিত্রায়িত উপস্থাপনাগুলি তৈরি করা।
3. **Prototyping:** Develop interactive prototypes for user testing.  
**প্রোটোটাইপিং:** ব্যবহারকারী পরীক্ষার জন্য আন্তঃক্রিয়াশীল প্রোটোটাইপ তৈরি করা।
4. **User Testing:** Gather feedback from users to improve the design.  
**ব্যবহারকারী পরীক্ষা:** ডিজাইন উন্নত করার জন্য ব্যবহারকারীদের কাছ থেকে প্রতিক্রিয়া সংগ্রহ করা।
5. **Final Design:** Create the final version of the user interface.  
**চূড়ান্ত ডিজাইন:** ব্যবহারকারীর ইন্টারফেসের চূড়ান্ত সংস্করণ তৈরি করা।

## Answer 7:

### 7(a) Characteristics of Good Software

1. **Functionality:** Meets the specified requirements and performs intended tasks.  
**ফাংশনালিটি:** নির্দিষ্ট চাহিদাগুলি পূরণ করে এবং উদ্দেশ্য অনুযায়ী কাজ করে।
2. **Reliability:** Works correctly and consistently under expected conditions.  
**বিশ্বাসযোগ্যতা:** প্রত্যাশিত অবস্থায় সঠিকভাবে এবং নিয়মিতভাবে কাজ করে।
3. **Usability:** User-friendly and easy to understand.  
**ব্যবহারযোগ্যতা:** ব্যবহারকারী-বান্ধব এবং বোঝা সহজ।
4. **Efficiency:** Uses system resources effectively without waste.  
**কার্যকারিতা:** সিস্টেমের সম্পদগুলি কার্যকরভাবে ব্যবহার করে।
5. **Maintainability:** Easy to update and fix bugs.  
**রক্ষণাবেক্ষণযোগ্যতা:** আপডেট এবং বাগ মেরামত করা সহজ।

6. **Portability:** Can be used on different platforms without major changes.  
**পোর্টেবিলিটি:** প্রধান পরিবর্তন ছাড়াই বিভিন্ন প্ল্যাটফর্মে ব্যবহার করা যেতে পারে।
7. **Scalability:** Can handle growth in users or data.  
**স্কেলেবিলিটি:** ব্যবহারকারী বা ডেটার বৃদ্ধিতে পরিচালনা করতে সক্ষম।

## 7(b) Big Bang Model

### Definition:

The Big Bang model is a software development approach where coding starts without any formal planning. Developers begin writing code as soon as they understand the project.

### Advantages:

- **Simplicity:** Easy to start, especially for small projects.  
**সরলতা:** শুরু করা সহজ, বিশেষ করে ছোট প্রকল্পের জন্য।
- **Flexibility:** Developers can make quick changes.  
**লচনশীলতা:** উন্নয়নকারীরা দ্রুত পরিবর্তন করতে পারে।

### Disadvantages:

- **Unpredictability:** Difficult to manage as project requirements can change rapidly.  
**অনিশ্চয়তা:** পরিচালনা করা কঠিন কারণ প্রকল্পের চাহিদা দ্রুত পরিবর্তিত হতে পারে।
- **Risk of Failure:** High chance of project failure due to lack of planning.  
**ব্যর্থতার ঝুঁকি:** পরিকল্পনার অভাবে প্রকল্পের ব্যর্থতার উচ্চ সম্ভাবনা।

## V-Model Structure

The V-Model is an extension of the waterfall model, where each phase of development is associated with a corresponding testing phase. It emphasizes verification and validation at every stage.

Copy code

```
Requirements Analysis
↓
System Design (High-Level)
↓
Architecture Design
↓
Module Design
↓
Implementation
↓
Unit Testing
↓
Integration Testing
↓
System Testing
↓
Acceptance Testing
```

## 7(c) Data Flow Diagram (DFD) Components and Levels

### Components of DFD:

1. **Processes:** Represent the transformations of data (circles or ovals).
2. **Data Stores:** Where data is stored (open-ended rectangles).
3. **Data Flows:** The movement of data (arrows).
4. **External Entities:** Sources or destinations of data (squares).

### Three Levels of DFD:

1. **Level 0 (Context Diagram):** Shows the system as a whole and its interaction with external entities.
2. **Level 1:** Breaks down the main process into sub-processes.
3. **Level 2:** Further details sub-processes into smaller components, showing more data flows.

---

### c. Alternatively, ACM/IEEE Code of **Ethics for Software Engineers**

1. **Public:** Software engineers shall act in the public interest.  
**সার্বজনীন:** সফটওয়্যার ইঞ্জিনিয়াররা পাবলিকের স্বার্থে কাজ করবে।
2. **Client and Employer:** Software engineers shall act in a manner that is in the best interests of their client and employer.  
**ক্লায়েন্ট এবং নিয়োগকর্তা:** সফটওয়্যার ইঞ্জিনিয়াররা তাদের ক্লায়েন্ট এবং নিয়োগকর্তার স্বার্থে কাজ করবে।
3. **Product:** Software engineers shall ensure that their products meet the highest professional standards.  
**পণ্য:** সফটওয়্যার ইঞ্জিনিয়াররা তাদের পণ্যগুলি সর্বোচ্চ পেশাদার মান পূরণ করে তা নিশ্চিত করবে।
4. **Judgment:** Software engineers shall maintain integrity and independence in their professional judgment.  
**বিবেক:** সফটওয়্যার ইঞ্জিনিয়াররা তাদের পেশাদার বিচার-বিবেচনায় সততা এবং স্বাধীনতা বজায় রাখবে।
5. **Management:** Software engineering managers and leaders shall promote an ethical approach to software development.  
**ব্যবস্থাপনা:** সফটওয়্যার ইঞ্জিনিয়ারিং ম্যানেজার এবং নেতারা সফটওয়্যার উন্নয়নের জন্য একটি নৈতিক পন্থা প্রচার করবে।
6. **Profession:** Software engineers shall advance the integrity and reputation of the profession.  
**পেশা:** সফটওয়্যার ইঞ্জিনিয়াররা পেশার সততা এবং সুনামকে উন্নত করবে।
7. **Colleagues:** Software engineers shall be fair and supportive to their colleagues.  
**সহকর্মীরা:** সফটওয়্যার ইঞ্জিনিয়াররা তাদের সহকর্মীদের প্রতি ন্যায্যবান এবং সহায়ক হবে।
8. **Self:** Software engineers shall participate in lifelong learning and career development.  
**নিজে:** সফটওয়্যার ইঞ্জিনিয়াররা জীবনব্যাপী শেখার এবং ক্যারিয়ার উন্নয়নে অংশগ্রহণ করবে।



## Answer 8:

### Effective Software Development Approaches

#### a) Pros and Cons of Using Prototypes

Faster Development:  
Early Feedback:  
Risk Identification  
Cost Efficiency

Unclear Requirements:  
Scalability Issues:  
Low Code Quality:  
Higher Costs Long-Term:

##### Pros:

1. **Immediate Feedback:** Users can review quickly.  
(তাত্ক্ষণিক প্রতিক্রিয়া: ব্যবহারকারীরা দ্রুত পর্যালোচনা করতে পারেন।)
2. **Engaged Users:** Users involved in the process feel satisfied.  
(সংশ্লিষ্ট ব্যবহারকারীরা: প্রক্রিয়ায় যুক্ত ব্যবহারকারীরা সন্তুষ্ট হন।)
3. **Risk Identification:** Potential issues can be identified early.  
(ঝুঁকি চিহ্নিতকরণ: সম্ভাব্য সমস্যা আগে থেকেই জানা যায়।)
4. **Proof of Concept:** Demonstrates the software's functionality.  
(বাস্তবায়নের প্রমাণ: সফটওয়্যার কার্যকারিতা দেখায়।)

##### Cons:

1. **Incomplete Requirements:** Important requirements may be overlooked.  
(অপূর্ণ চাহিদা: গুরুত্বপূর্ণ চাহিদাগুলি অবহেলা হতে পারে।)
2. **Scope Creep:** Requests for changes can increase.  
(স্কেপ ক্রিপ: পরিবর্তনের অনুরোধ বাড়তে পারে।)
3. **False Sense of Completeness:** The prototype may seem complete.  
(ভ্রান্ত সম্পূর্ণতা: প্রোটোটাইপ সম্পূর্ণ বলে মনে হতে পারে।)
4. **Increased Costs:** Additional expenses for prototype development.  
(বাড়তি খরচ: প্রোটোটাইপ তৈরি করতে ব্যয় বাড়ে।)

#### b) Appropriate Software Process Models for Different Systems

1. **System to Control Antilock Braking in a Car**
  - **Model:** V-Model  
(মডেল: V-মডেল)
  - **Reason:** Ideal for complex systems.  
(কারণ: এটি কঠিন সিস্টেমের জন্য আদর্শ।)
2. **Virtual Reality System**
  - **Model:** Agile Methodology  
(মডেল: অ্যাডাইল মেথডোলজি)
  - **Reason:** Suitable for quick changes and user input.  
(কারণ: দ্রুত পরিবর্তন এবং ব্যবহারকারীর ইনপুটের জন্য উপযুক্ত।)
3. **University Accounting System Replacement**
  - **Model:** Waterfall Model  
(মডেল: ওয়াটারফল মডেল)
  - **Reason:** Effective for stable requirements.  
(কারণ: নির্ভরযোগ্য প্রয়োজনীয়তার জন্য কার্যকর।)

- 4. **Interactive Travel Planning System**
  - **Model:** Spiral Model  
(মডেল: স্পাইরাল মডেল)
  - **Reason:** Good for risk management.  
(কারণ: ঝুঁকি পরিচালনার জন্য ভালো।)

**c) Should there be a separate profession of 'software architect'? (সফটওয়্যার আর্কিটেক্টের আলাদা পেশা থাকা উচিত?)**

Yes, a separate software architect profession can be beneficial. (হ্যাঁ, সফটওয়্যার আর্কিটেক্টের আলাদা পেশা থাকা উপকারী।)

Software architects design the structure of software systems, ensuring they meet requirements.

**Challenges (চ্যালেঞ্জ):**

1. **No Clear Standards (কোনও স্পষ্ট মান নেই):**  
Lack of specific guidelines. (নির্দিষ্ট গাইডলাইন নেই। )
2. **Communication Issues (যোগাযোগের সমস্যা):**  
Difficulty in effective communication among teams. (দলগুলোর মধ্যে কার্যকর যোগাযোগে সমস্যা। )
3. **Resistance to Change (পরিবর্তনের বিরুদ্ধে প্রতিরোধ):**  
Current engineers may resist role changes. (বর্তমান ইঞ্জিনিয়াররা পরিবর্তনে অনিচ্ছুক হতে পারে। )
4. **High Skill Requirements (উচ্চ দক্ষতার প্রয়োজন):**  
Need for significant experience and knowledge. (বহু অভিজ্ঞতা ও জ্ঞানের প্রয়োজন। )
5. **Cost Issues (খরচের সমস্যা):**  
Hiring separate architects increases costs. (আলাদা আর্কিটেক্ট নিয়োগের খরচ বাড়ায়। )

1. Communication Gap Between Architect and Developers:

2. Responsibility and Accountability Issues:

3. Lack of Flexibility:

4. Coordination Challenges:

5. Cost Concerns: