

1. Explain why having a standardized system development process is important to an organization.

Answer:

Importance of a Standardized System Development Process

A standardized system development process is essential for organizations because it:

1. **Ensures Consistency and Quality** – Reduces defects and improves system quality (as seen in the CMM framework).
2. **Boosts Efficiency** – Increases productivity by using repeatable methods like FAST.
3. **Improves Resource Management** – Makes it easier to shift resources across projects.
4. **Supports Maintenance** – Standard documentation lowers maintenance costs and aids system updates.
5. **Reduces Risks** – Uses checkpoints to catch issues early and avoid failure.
6. **Meets Compliance** – Necessary for industries with strict standards, offering competitive advantage.
7. **Aligns Stakeholders** – Involves key users and ensures goals are clearly understood.

2. How are system life cycle and system development methodology related ?

Relation Between System Life Cycle and System Development Methodology

The system life cycle is the overall process of developing and maintaining an information system, typically divided into phases like planning, analysis, design, implementation, and maintenance.

The **system development methodology** is the specific, structured approach or set of rules followed to carry out each phase of the system life cycle (e.g. Waterfall, Agile, or FAST).

How They Are Related:

1. **Structure vs. Process:**
 - The **system life cycle** defines **what** phases happen.
 - The **methodology** defines **how** each phase is done.
2. **Guidance:**
 - Methodologies provide step-by-step guidance for completing each life cycle phase.
3. **Consistency:**
 - Using a methodology ensures the life cycle is followed consistently across projects.
4. **Quality and Control:**
 - Methodologies improve the quality and manageability of the entire life cycle process.

In summary: The system life cycle is the *framework*, and the system development methodology is the *toolset* used to effectively carry out that framework.

✓ 3. What are the 10 underlying principles for systems development?

10 Easy Principles of Systems Development:

1. **Involve users** – Work with the people who will use the system.
2. **Solve real problems** – Focus on fixing business issues.
3. **Follow steps** – Do the work in clear phases (like planning, design).
4. **Write everything down** – Keep good records.
5. **Use standards** – Follow rules to keep everything consistent.
6. **Manage well** – Control time, cost, and tasks.
7. **Check cost vs. benefit** – Make sure it's worth the money.
8. **Be flexible** – Change or stop if things don't work.
9. **Break into parts** – Make the system in small, easy pieces.
10. **Plan for the future** – Build systems that can grow and change.

✓ 4. Why is documentation important throughout the development process?

Why is Documentation Important in System Development?

1. **Keeps everything clear** – Helps everyone understand what's being done.
2. **Helps new team members** – New people can quickly learn the project.
3. **Supports maintenance** – Easier to fix or update the system later.
4. **Tracks progress** – Shows what work is done and what's left.
5. **Avoids confusion** – Reduces mistakes and miscommunication.
6. **Is needed for training** – Helps users learn how to use the system.
7. **Useful for future projects** – Acts as a reference for other systems.

✓ 5. Why are process management and project management necessary?

Why Are Process Management and Project Management Necessary?

1. **Keep work organized** – Helps plan and control each step of development.
2. **Save time and money** – Avoids delays and keeps the project within budget.
3. **Reach goals** – Makes sure the final system meets the users' needs.
4. **Reduce mistakes** – Helps find and fix problems early.

5. **Improve teamwork** – Everyone knows their roles and tasks.
6. **Track progress** – Makes it easy to check what's done and what's left.

6. What is risk management? Why is it necessary?

What is Risk Management?

Risk management means **finding problems before they happen** and **making a plan to avoid or reduce them**.

Why Is It Necessary?

1. **Avoid failure** – Helps stop big problems before they ruin the project.
2. **Save time and money** – Prevents delays and extra costs.
3. **Plan better** – Teams can prepare for possible risks.
4. **Protect quality** – Ensures the final system works well.
5. **Increases confidence** – Everyone feels safer and more in control.

7. Which stakeholders initiate most projects? What is the impetus for most projects?

Who Initiates Most Projects?

Most projects are started by system owners and system users.
They are the ones who experience problems, see opportunities, or need improvements in their daily work.

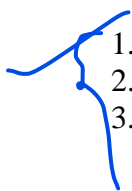
What Is the Main Reason (Impetus) for Most Projects?

The main reason is to solve business problems or take advantage of new opportunities.
This could mean improving productivity, saving time or money, improving customer service, or staying ahead of competitors.

8. Who are the main participants in the scope definition? What are their goals in the scope definition?

Who Are the Main Participants in Scope Definition?

The main participants are:

- 
1. **System owners**
 2. **System users**
 3. **System analysts**
-

What Are Their Goals?

- **System Owners:**
Want to define the business goals, budgets, and what the system should achieve.
 - **System Users:**
Want to make sure the system supports their daily work and solves their problems.
 - **System Analysts:**
Help gather information, understand needs, and clearly define what the system will do.
-



9. What are the three most important deliverables in scope definition?


Three Most Important Deliverables in Scope Definition:

1. **Project Scope Statement:**
 - **What it is:** A detailed description of the project's objectives, goals, deliverables, and boundaries.
 - **Why it's important:** It outlines what will be included and excluded in the project. This helps set clear expectations for stakeholders and avoids scope creep (uncontrolled changes).
 2. **Requirements Document:**
 - **What it is:** A comprehensive document that captures the functional and non-functional requirements of the system. It details the features, performance needs, and user requirements.
 - **Why it's important:** It ensures that the system will meet the needs of the users and stakeholders. This document serves as the foundation for system design and development.
 3. **Work Breakdown Structure (WBS):**
 - **What it is:** A visual representation of the project that breaks down the entire scope into smaller, manageable tasks and sub-tasks. It shows the relationship between tasks and their dependencies.
 - **Why it's important:** It helps project managers organize and prioritize work, assign resources effectively, and track progress. The WBS provides a roadmap for the project and ensures that all components are addressed.
-


✓ 10. Who are the main participants in the requirements analysis phase? Why are they the main participants?

Who Are the Main Participants in the Requirements Analysis Phase?

The main participants are:

- 
- **System Users**
 - **System Owners**
 - **System Analysts**
 - **Developers**
 - **QA (Quality Assurance) Team**
-


Why Are They the Main Participants?

- 
- **System Users:** They use the system, so they know what features and functions are needed.
 - **System Owners:** They set business goals and provide money and resources.
 - **System Analysts:** They collect and organize all the requirements clearly.
 - **Developers:** They check if the system can be built and suggest technical solutions.
 - **QA Team:** They make sure the requirements can be tested and that the system will work properly.
-

✓ 11. What feasibility analyses are made in the decision analysis?

Feasibility Analyses in Decision Analysis:

During decision analysis, the following types of **feasibility analyses** are done:

- 
1. **Technical Feasibility** – Checks if the technology needed for the system is available and works properly.
 2. **Economic Feasibility** – Checks if the project is affordable and worth the cost (cost vs. benefit).
 3. **Operational Feasibility** – Checks if the system will work well in the company and if people will use it.
 4. **Schedule Feasibility** – Checks if the system can be finished on time.
 5. **Legal and Ethical Feasibility** – Checks if the system follows laws and ethical rules.
-

12. What is model-driven development?

What is Model-Driven Development?

English:

Model-driven development (MDD) is a system development approach where developers create models (like diagrams) to show how the system will work. These models guide the design, coding, and testing of the system. The main goal is to focus more on understanding the system through visual models before writing code.

Bangla:

মডেল-ড্রিভেন ডেভেলপমেন্ট (Model-Driven Development) হলো একটি সিস্টেম তৈরি করার পদ্ধতি, যেখানে প্রথমে ডায়াগ্রামের মতো মডেল তৈরি করা হয় — যা সিস্টেমটি কীভাবে কাজ করবে তা দেখায়। এই মডেলগুলো ব্যবহার করে পরে ডিজাইন, কোডিং এবং টেস্টিং করা হয়। এতে কোড লেখার আগে পুরো সিস্টেমটি ভালোভাবে বোঝা যায়।

13. Why is model-driven development popular?

Why is Model-Driven Development Popular?

English:

Model-driven development is popular because it helps developers and users understand the system easily using diagrams and models. It reduces mistakes, saves time, and improves communication between team members. It also helps in planning before writing any code, which leads to better system design.

Bangla:

মডেল-ড্রিভেন ডেভেলপমেন্ট জনপ্রিয় কারণ এটি ডেভেলপার এবং ইউজারদের জন্য সিস্টেম বোঝা সহজ করে তোলে। ডায়াগ্রাম ও মডেলের মাধ্যমে কাজের ধারণা পরিষ্কার হয়। এতে ভুল কম হয়, সময় বাঁচে এবং টিমের মধ্যে যোগাযোগ ভালো হয়। কোড লেখার আগে পরিকল্পনা করা যায়, ফলে সিস্টেমের ডিজাইন আরও ভালো হয়।

14. What is rapid application development (RAD)?

What is Rapid Application Development (RAD)?

English:

Rapid Application Development (RAD) is a system development method that focuses on quickly building and improving software using user feedback. It uses prototypes, small teams, and short development cycles to deliver working systems fast. The goal is to reduce development time and increase flexibility.

Bangla:

গ্রাফিক্স অ্যাপ্লিকেশন ডেভেলপমেন্ট (RAD) হলো একটি সিস্টেম তৈরি করার পদ্ধতি, যেখানে দ্রুত সফটওয়্যার তৈরি ও উন্নয়নে গুরুত্ব দেওয়া হয়। এতে প্রোটোটাইপ, ছোট টিম, এবং ছোট সময়ের ডেভেলপমেন্ট সাইকেল ব্যবহার করা হয়। এর লক্ষ্য হলো কম সময়ে কাজ করা এবং ইউজারের ফিডব্যাক অনুযায়ী দ্রুত পরিবর্তন আনা।

15. What benefits can RAD bring to the system development process?

What Benefits Can RAD Bring to the System Development Process?

English:

RAD (Rapid Application Development) brings many benefits to system development, such as:

- **Faster development** – Systems are built quickly using short cycles.
- **Early user feedback** – Prototypes help users give feedback early.
- **Flexibility** – Changes can be made easily during development.
- **Better quality** – Frequent testing helps find and fix problems early.
- **More user satisfaction** – Users are involved in every step, so the final system matches their needs.

16. What is computer-assisted software engineering (CASE)? List some examples of CASE.

What is CASE (Computer-Assisted Software Engineering)?

CASE means using software tools to **help design, develop, and manage computer systems**. These tools help programmers, analysts, and designers make system models, write code, check errors, and create documents. It makes the software development process faster, easier, and more organized.

Examples of CASE Tools:

1. **IBM Rational Rose** – Used for making system models using UML.
 2. **Oracle Designer** – Helps design databases and business processes.
 3. **Microsoft Visio** – Used to draw flowcharts, DFDs, and ER diagrams.
 4. **CA ERwin** – A tool for database design.
 5. **Enterprise Architect** – Used for system modeling and analysis.
 6. **Visible Analyst** – Helps design systems with diagrams.
 7. **PowerDesigner (SAP)** – For data and process modeling.
-

Main Features of CASE Tools:

- **Forward Engineering:** Makes code from system models.
 - **Reverse Engineering:** Turns code back into system models.
 - **Repository:** Stores all documents and models in one place.
 - **Modeling Tools:** Draws diagrams like DFDs and ERDs.
 - **Documentation:** Creates system reports automatically.
-

এখানে উপরের লেখাটির সহজ ও সংক্ষিপ্ত **বাংলা অনুবাদ** দেওয়া হলো:

CASE (Computer-Assisted Software Engineering) কী?

CASE হলো এমন সফটওয়্যার টুল, যা সিস্টেম ডিজাইন, ডেভেলপমেন্ট, ও মেইনটেন্যান্সে সাহায্য করে। এই টুলগুলো প্রোগ্রামার, অ্যানালিস্ট ও ডিজাইনারদের জন্য মডেল তৈরি, কোড লেখা, ভুল খোঁজা, ও ডকুমেন্ট তৈরি সহজ করে। এতে সফটওয়্যার তৈরির কাজ দ্রুত, সহজ ও গুছানো হয়।

CASE টুলের উদাহরণ:

1. **IBM Rational Rose** – UML দিয়ে সিস্টেম মডেল তৈরি করতে ব্যবহৃত হয়।
 2. **Oracle Designer** – ডেটাবেস ও বিজনেস প্রসেস ডিজাইনের জন্য।
 3. **Microsoft Visio** – ফ্লোচার্ট, DFD ও ERD ডায়াগ্রাম আঁকতে ব্যবহৃত।
 4. **CA ERwin** – ডেটাবেস ডিজাইনের জন্য জনপ্রিয় টুল।
 5. **Enterprise Architect** – সিস্টেম মডেল ও অ্যানালাইসিসের জন্য।
 6. **Visible Analyst** – ডায়াগ্রাম দিয়ে সিস্টেম ডিজাইনের জন্য।
 7. **PowerDesigner (SAP)** – ডেটা ও প্রসেস মডেলিংয়ের জন্য ব্যবহৃত।
-

CASE টুলের প্রধান বৈশিষ্ট্য:

- **Forward Engineering:** মডেল থেকে অটো কোড তৈরি করা।
- **Reverse Engineering:** কোড থেকে মডেল তৈরি করা।
- **Repository:** সব ডকুমেন্ট ও মডেল এক জায়গায় রাখা।

- **Modeling Tools:** DFD, ERD-এর মতো ডায়াগ্রাম তৈরি করা।
- **Documentation:** অটো ডকুমেন্ট রিপোর্ট তৈরি করে।

1. The Capability Maturity Model (CMM) was developed by the Software Engineering Institute at Carnegie Mellon, and is widely used by both the private and public sectors. What is the purpose of the CMM framework and how does it achieve this?

Purpose of the Capability Maturity Model (CMM)

The CMM is a framework that helps organizations improve their software development processes. Its main goals are:

1. **Assess Process Maturity** – To check how well an organization follows a structured approach.
2. **Guide Process Improvement** – To help organizations move from chaotic to well-organized processes.
3. **Improve Quality & Efficiency** – To reduce defects and increase productivity by using best practices.
4. **Benchmarking & Compliance** – Some contracts require meeting certain CMM levels.

How CMM Works

CMM has **five levels** of process maturity:

| Level | Name | Key Characteristics |
|---------------|--------------|---|
| 1. Initial | Chaotic | Processes are random and depend on individuals. |
| 2. Repeatable | Managed | Basic project management is in place. |
| 3. Defined | Standardized | Processes are documented and followed. |
| 4. Managed | Controlled | Processes are measured and controlled. |
| 5. Optimizing | Improving | Processes are improved based on feedback. |

How CMM Helps Improve

- **Best Practices** – Provides guidelines for improvement.
 - **Step-by-Step Progress** – Allows gradual, steady growth.
 - **Measurement** – Tracks performance to find areas for improvement.
 - **Training** – Promotes continuous learning and standard practices.
-

Impact of CMM

- **Better Quality** – Fewer defects and more reliable software.
- **Cost & Time Control** – Reduces extra work and delays.
- **Competitive Advantage** – Higher CMM levels attract better projects.

By following CMM, organizations improve from disorganized development to structured, efficient processes.

2. List the five maturity levels, and briefly describe each of them.

Five Maturity Levels of CMM:

1. **Level 1: Initial**
 - **Description:** Processes are unpredictable and depend on individual efforts. Success is not consistent.
2. **Level 2: Repeatable**
 - **Description:** Basic project management is in place, which helps in repeating successful results.
3. **Level 3: Defined**
 - **Description:** Processes are well-documented and standardized across the organization.
4. **Level 4: Managed**
 - **Description:** Processes are controlled and measured using data to ensure they are working well.
5. **Level 5: Optimizing**
 - **Description:** Continuous improvements are made based on feedback and data to improve processes.

Here are the answers to the "**Problems and Exercises**" based on your uploaded PDF, in **both English and Bangla**:

1. Purpose of the CMM Framework

English:

The purpose of the Capability Maturity Model (CMM) is to provide a structured way for organizations to improve the maturity and quality of their systems development processes. It achieves this by defining five levels of maturity that guide improvement from chaotic practices to optimized processes.

বাংলা:

Capability Maturity Model (CMM)-এর উদ্দেশ্য হলো একটি সংগঠনের সিস্টেম ডেভেলপমেন্ট প্রক্রিয়ার পরিপক্বতা এবং গুণমান উন্নয়নের জন্য একটি কাঠামোবদ্ধ উপায় প্রদান করা। এটি পাঁচটি পরিপক্বতার স্তর নির্ধারণ করে যা এলোমেলো অবস্থা থেকে উন্নত ও অপ্টিমাইজড প্রক্রিয়ার দিকে সংগঠনকে পরিচালিত করে।

2. Five Maturity Levels of CMM

English:

1. **Initial:** Unpredictable, chaotic, success depends on individual effort.
2. **Repeatable:** Basic project management is established, repeatable success.
3. **Defined:** Standard processes are documented and used organization-wide.
4. **Managed:** Quantitative goals are set and measured.
5. **Optimizing:** Continuous improvement based on feedback and innovation.

বাংলা:

1. **Initial (প্রাথমিক):** এলোমেলো, অনিয়মিত, ব্যক্তিগত প্রচেষ্টার ওপর নির্ভর।
 2. **Repeatable (পুনরাবৃত্তিযোগ্য):** মৌলিক প্রজেক্ট ম্যানেজমেন্ট প্রতিষ্ঠিত, আগের সফলতা পুনরায় অর্জনযোগ্য।
 3. **Defined (সংজ্ঞায়িত):** সংগঠনজুড়ে প্রক্রিয়াগুলো নথিভুক্ত এবং ব্যবহার হয়।
 4. **Managed (পরিচালিত):** গুণমান ও উৎপাদনক্ষমতার পরিমাপযোগ্য লক্ষ্য নির্ধারণ ও মূল্যায়ন করা হয়।
 5. **Optimizing (উন্নয়নশীল):** প্রতিনিয়ত উন্নয়ন হয় প্রতিক্রিয়া ও উদ্ভাবনের মাধ্যমে।
-

3. Greatest Benefit Between CMM Levels

English:

The greatest improvement occurs between **Level 1 (Initial)** and **Level 2 (Repeatable)**.

Reason: Moving from chaos to basic process control greatly enhances predictability, reduces risk, and improves quality.

বাংলা:

সর্বোচ্চ উন্নতি হয় **Level 1 (Initial)** থেকে **Level 2 (Repeatable)**-এ যাওয়ার সময়।

কারণ: এলোমেলো অবস্থা থেকে মৌলিক প্রক্রিয়ার দিকে অগ্রসর হওয়া প্রজেক্টের পূর্বাভাসযোগ্যতা বাড়ায়, ঝুঁকি কমায় এবং গুণমান উন্নত করে।

4. Difference Between System Life Cycle and Development Methodology

English:

- **System Life Cycle** is the entire lifespan of an information system (development + operation).
- **Development Methodology** refers to the step-by-step approach used **during** development.

বাংলা:

- **System Life Cycle** হলো একটি ইনফরমেশন সিস্টেমের সম্পূর্ণ জীবনকাল (ডেভেলপমেন্ট + পরিচালনা)।
 - **Development Methodology** হলো কেবলমাত্র **ডেভেলপমেন্ট** পর্যায়ে ব্যবহৃত ধাপে ধাপে নির্দেশিকা বা কৌশল।
-

5. How Methodology Aligns with CMM

English:

A systems development methodology introduces repeatable, standardized processes, which aligns directly with CMM goals, helping organizations improve their maturity level.

বাংলা:

সিস্টেম ডেভেলপমেন্ট মেথডোলজি পুনরাবৃত্তিযোগ্য ও মানসম্মত প্রক্রিয়া চালু করে যা CMM-এর লক্ষ্যগুলোর সাথে সামঞ্জস্যপূর্ণ এবং সংগঠনকে তাদের পরিপক্বতার স্তর উন্নত করতে সাহায্য করে।

6. Underlying Principles of Systems Development

English:

1. Involve users.
2. Use problem-solving approach.
3. Use phased activities.
4. Document everything.
5. Set standards.
6. Manage process and project.
7. Justify as investment.
8. Be flexible.
9. Divide and conquer.
10. Design for growth and change.

বাংলা:

১. ব্যবহারকারীদের অন্তর্ভুক্ত করুন।
২. সমস্যা-সমাধান ভিত্তিক পদ্ধতি ব্যবহার করুন।
৩. ধাপে ধাপে কার্যক্রম ভাগ করুন।
৪. পুরো প্রক্রিয়ায় ডকুমেন্টেশন করুন।
৫. মান নির্ধারণ করুন।
৬. প্রক্রিয়া ও প্রজেক্ট পরিচালনা করুন।
৭. বিনিয়োগ হিসেবে যুক্তিসঙ্গত করুন।
৮. প্রয়োজনে পরিবর্তন বা বাতিল করুন।
৯. সমস্যাকে ছোট ছোট অংশে ভাগ করুন।
১০. ভবিষ্যৎ উন্নয়নের জন্য ডিজাইন করুন।

7. PIECES Framework Classification

| Problem | PIECES Category | বাংলা |
|------------------------|-----------------|-------------------|
| a. Duplicate data | I (Information) | তথ্য সমস্যা |
| b. Port to PDA | S (Service) | সেবার উন্নয়ন |
| c. Auto sales reports | E (Efficiency) | দক্ষতা বৃদ্ধি |
| d. Unauthorized access | C (Control) | নিয়ন্ত্রণ সমস্যা |
| e. Confusing UI | S (Service) | সেবার গুণমান |

8. FAST Phase Deliverables

English:

- **Requirements Analysis:** Business requirements statement.
- **Logical Design:** Logical models (data, process, interface).
- **Physical Design:** Technical specs, prototypes.

বাংলা:

- **Requirements Analysis:** ব্যবসায়িক চাহিদার বিবৃতি।
 - **Logical Design:** ডেটা, প্রসেস এবং ইন্টারফেসের লজিক্যাল মডেল।
 - **Physical Design:** প্রযুক্তিগত স্পেসিফিকেশন, প্রোটোটাইপ।
-

9. Scope Definition Phase

English:

- **Triggers:** Problems, opportunities, directives.
- **Participants:** Owners, analysts, managers.
- **Questions:** Is the problem worth solving? What's the scope?
- **Deliverables:** Problem statement, scope statement, work contract.

বাংলা:

- **ট্রিগার:** সমস্যা, সুযোগ, নির্দেশনা।
 - **অংশগ্রহণকারী:** সিস্টেম মালিক, বিশ্লেষক, ব্যবস্থাপক।
 - **প্রশ্ন:** এই সমস্যা সমাধান করা কি যৌক্তিক? এর পরিধি কতটুকু?
 - **ডেলিভারেবলস:** সমস্যা বিবৃতি, স্কোপ বিবৃতি, কাজের চুক্তি।
-

10. Requirements Analysis Phase

English:

- **Participants:** Users, analysts, managers.
- **Focus:** What the system should do.
- **Not Focus:** Technical solutions.
- **Evaluation:** Does each requirement help improvement?
- **Avoid:** Rushing or skipping this phase.

বাংলা:

- **অংশগ্রহণকারী:** ব্যবহারকারী, বিশ্লেষক, ব্যবস্থাপক।
 - **কেন্দ্রবিন্দু:** সিস্টেম কী করবে।
 - **যা নয়:** প্রযুক্তিগত সমাধান।
 - **মূল্যায়ন:** প্রতিটি চাহিদা উন্নয়নে সহায়ক কি না।
 - **এড়ানো উচিত:** তাড়াহুড়ো বা এই ধাপ বাদ দেওয়া।
-

11. Why Owners and Designers Are Excluded

English:

They may bias the process with high-level or technical focus, which interferes with clear, user-focused requirement gathering.

বাংলা:

তারা উচ্চ পর্যায়ের লক্ষ্য বা প্রযুক্তিগত দৃষ্টিভঙ্গি জোর করে প্রয়োগ করতে পারেন, যা ব্যবহারকারী-কেন্দ্রিক চাহিদা সংগ্রহে ব্যাঘাত ঘটায়।

12. Logical Design Phase

English:

- **Purpose:** Convert business needs into logical models.
- **Not Tech-Based:** No specific technology yet.
- **Synonyms:** Conceptual or essential design.
- **Participants:** Analysts, users.
- **Agile Modeling:** Minimal modeling to stay efficient.
- **Deliverables:** Logical models.
- **Transition:** From business to technical planning.

বাংলা:

- **উদ্দেশ্য:** ব্যবসায়িক চাহিদাকে প্রযুক্তি-নিরপেক্ষ লজিক্যাল মডেলে রূপান্তর।
 - **প্রযুক্তিনিরপেক্ষ:** এখনো নির্দিষ্ট প্রযুক্তির প্রয়োগ নয়।
 - **সমার্থক শব্দ:** কনসেপচুয়াল ডিজাইন, এসেনশিয়াল ডিজাইন।
 - **অংশগ্রহণকারী:** বিশ্লেষক ও ব্যবহারকারী।
 - **Agile মডেলিং:** শুধুমাত্র যতটুকু প্রয়োজন ততটুকু মডেলিং।
 - **ডেলিভারেবল:** লজিক্যাল মডেলসমূহ।
 - **ট্রানজিশন:** ব্যবসা থেকে প্রযুক্তিগত পর্যায়ে স্থানান্তর।
-

13. Physical Design Phase

English:

- **Purpose:** Convert logical models to technical specs.
- **Participants:** Designers, analysts, users (for feedback).
- **Philosophies:**
 - Design by specification = detailed planning.
 - Design by prototyping = iterative improvement.

- **Cancellation:** Rare.
- **Overlap:** Construction phase—because building starts while designing.

বাংলা:

- **উদ্দেশ্য:** লজিক্যাল মডেলকে প্রযুক্তিগত নির্দিষ্টকরণে রূপান্তর করা।
- **অংশগ্রহণকারী:** ডিজাইনার, বিশ্লেষক, ব্যবহারকারী (ফিডব্যাকের জন্য)।
- **দর্শন:**
 - স্পেসিফিকেশন অনুযায়ী ডিজাইন = বিস্তারিত পরিকল্পনা।
 - প্রোটোটাইপ অনুযায়ী ডিজাইন = ধাপে ধাপে উন্নয়ন।
- **বাতিল হওয়ার সম্ভাবনা:** খুব কম।
- **ওভারল্যাপ:** কনস্ট্রাকশন ধাপের সাথে, কারণ ডিজাইন চলাকালীনই নির্মাণ শুরু হয়।

Let me know if you'd like a printable version or a summarized chart version!:

14. What are two system development strategies that might be advantageous in this situation?

Answer (4 Marks):

Two suitable strategies are:

1. **Prototyping:** Allows rapid development of a working model to clarify vague requirements and gather feedback from users early.
2. **Timeboxing (or Iterative/Agile Development):** Delivers a basic, functional system within a fixed time frame by prioritizing essential features and deferring less critical ones to future releases.

15. What is the potential downside to using the strategies described above?

Answer (4 Marks):

The downsides include:

- **Prototyping:** May lead to poorly engineered final systems if prototypes are not properly refined or if short-term fixes are retained.
 - **Timeboxing/Agile:** Important features might be left out if not prioritized correctly, and tight deadlines can compromise testing or documentation.
-

১৪. এই অবস্থায় উপযোগী দুটি সিস্টেম ডেভেলপমেন্ট কৌশল কী হতে পারে?

উত্তর (৪ নম্বর):

১. প্রোটোটাইপিং (Prototyping): দ্রুত একটি কাজের মতো মডেল তৈরি করে ব্যবহারকারীদের অস্পষ্ট চাহিদা পরিষ্কার করা যায় এবং তাদের কাছ থেকে দ্রুত ফিডব্যাক নেওয়া যায়।
 ২. টাইমবক্সিং বা অ্যাজাইল উন্নয়ন (Timeboxing/Agile): নির্দিষ্ট সময়ের মধ্যে সবচেয়ে গুরুত্বপূর্ণ ফিচারগুলো দ্রুত ডেলিভারি দেওয়া হয় এবং কম গুরুত্বপূর্ণ ফিচারগুলো পরবর্তীতে যুক্ত করা হয়।
-

১৫. উপরের কৌশলগুলো ব্যবহারের সম্ভাব্য অসুবিধাগুলো কী?

উত্তর (৪ নম্বর):

- প্রোটোটাইপিং: যদি প্রোটোটাইপ ঠিকভাবে উন্নত করা না হয়, তাহলে চূড়ান্ত সিস্টেম দুর্বল হতে পারে।
 - টাইমবক্সিং/অ্যাজাইল: যদি সঠিকভাবে অগ্রাধিকার না দেওয়া হয়, তবে গুরুত্বপূর্ণ ফিচার বাদ পড়ে যেতে পারে। তাড়াহুড়ো করে কাজ করলে টেস্টিং ও ডকুমেন্টেশন দুর্বল হতে পারে।
-