**Nguyễn Hải Minh - 21522342**

**Question 1: Project Initiation**

a) Explain the importance of project initiation in the software development lifecycle.  
b) List and briefly describe the key components of a project charter.

**Answer Template:**

a) Importance of Project Initiation:

Project initiation in the software development lifecycle is pivotal as it lays the groundwork for project success. It clarifies objectives and scope, establishes governance and leadership structures, identifies, and manages risks, defines the project approach and methodology, sets baselines for time, cost, and quality, engages stakeholders, and facilitates detailed planning. By addressing these aspects early on, project initiation ensures alignment between project goals and stakeholder expectations, mitigates risks, and provides a clear roadmap for project execution, ultimately increasing the likelihood of delivering a high-quality software product on time and within budget.

b) Key Components of a Project Charter:

1. Purpose and objectives.

2. Scope and deliverables

3. Stakeholders and roles.

**Question 2: Stakeholder Management**

Describe the significance of stakeholder management in software project management. Provide examples of strategies for effectively engaging and managing stakeholders throughout the project lifecycle.

**Answer Template:**

\* Significance of Stakeholder Management:

Stakeholder management in software project management is crucial for project success as it ensures that the needs, expectations, and concerns of all stakeholders are identified, understood, and addressed throughout the project lifecycle. Effective stakeholder management involves engaging stakeholders early and often, communicating transparently, and actively managing relationships to build trust and alignment. By involving stakeholders in decision-making, managing their expectations, and addressing their concerns, software project managers can mitigate risks, foster stakeholder buy-in, and ultimately increase the likelihood of delivering a product that meets stakeholder needs and achieves organizational objectives.

\* Strategies for Stakeholder Engagement:

1. Active Communication: Maintain regular, transparent communication with stakeholders, addressing their concerns promptly and using appropriate channels and formats.

2. Stakeholder Analysis and Mapping: Identify key stakeholders, assess their interests, influence, and impact on the project, and prioritize engagement efforts accordingly.

3. Engagement Activities and Participation: Involve stakeholders in decision-making processes, solicit their input through workshops or meetings, and create opportunities for them to contribute to project planning and execution.

**Question 3: Agile Project Management**

a) Compare and contrast traditional project management with Agile project management methodologies.  
b) Discuss the benefits and challenges of adopting Agile methodologies in software development projects.

**Answer Template:**

a) Comparison of Traditional and Agile Project Management:

|  |  |
| --- | --- |
| **Traditional Project Management** | **Agile Project Management** |
| Sequential approach to steps | Iterative and incremental |
| Predictive planning | Adaptive planning throughout the process |
| High emphasis on documenting | Customer collaboration |

b) Benefits and Challenges of Agile Methodologies:

Benefits:

1. Flexibility: Agile allows for adaptability to changing requirements and market conditions.

2. Customer Satisfaction: Agile involves customers throughout the process, leading to greater alignment with their needs.

Challenges:

1. Lack of Predictability: Due to its iterative nature, predicting project timelines and outcomes can be challenging.
2. Flexibility: Agile allows for adaptability to changing requirements and market conditions.

**Question 4: Risk Management**

Explain the process of risk management in software project management. Provide examples of risk identification techniques and mitigation strategies.

**Answer Template:**

Process of Risk Management:

Risk management in software project management involves identifying, analyzing, and mitigating potential risks to project objectives. This includes identifying threats and opportunities, analyzing their likelihood and impact, developing mitigation strategies, and continuously monitoring and managing risks throughout the project lifecycle to minimize their impact on project success.

Risk Identification Techniques:

1. Brainstorming: Gather project team members and stakeholders to generate a list of potential risks based on their expertise and perspectives.

2. Risk Checklists: Utilize predefined risk checklists or templates tailored to the specific domain or industry of the software project.

Mitigation Strategies:

1. Risk Avoidance: Take proactive measures to avoid or eliminate risks altogether by altering project plans, processes, or activities.

2. Risk Transfer: Transfer the responsibility for managing certain risks to external parties, such as vendors or insurance providers.

**Question 5: Project Monitoring and Control**

a) Define project monitoring and control and its significance in project management.  
b) List and briefly explain the key performance indicators (KPIs) used for monitoring and controlling software development projects.

**Answer Template:**

a) Definition and Significance of Project Monitoring and Control:

Project monitoring and control involve tracking project performance, comparing it to the planned objectives, identifying deviations, and taking corrective actions to ensure project objectives are met. This process includes gathering project data, analyzing progress against the project plan, identifying variances, and implementing changes as needed to keep the project on track. The significance of project monitoring and control lies in its ability to provide visibility into project progress, identify issues early, and take proactive measures to address them, ultimately ensuring project success within scope, time, and budget constraints.

b) Key Performance Indicators (KPIs):

1. Velocity: Measures the amount of work completed by the development team in each iteration, providing insight into productivity and capacity.

2. Defect Density: Tracks the number of defects identified relative to the size of the software product, indicating product quality and effectiveness of quality assurance efforts.

3. Defect Density: Tracks the number of defects identified relative to the size of the software product, indicating product quality and effectiveness of quality assurance efforts.