Assignment 1: SDLC Overview - Create a one-page infographic that outlines the SDLC phases (Requirements, Design, Implementation, Testing, Deployment), highlighting the importance of each phase and how they interconnect.



A brief overview of the SDLC phases with a focus on an Amazon project:

**Software Development Life Cycle (SDLC) Overview:**

**1.Requirements Phase:**

• Define business needs, functionalities, and constraints.

• Collaborate with stakeholders to gather requirements.

• In Amazon's project, this phase involves identifying user needs, market analysis, and defining product features such as search functionality, product listings, and checkout processes.

**2.Design Phase:**

• Create a blueprint for the software system.

• Translate requirements into technical specifications.

• In Amazon's project, this phase includes designing the user interface, database structure, and system architecture to ensure scalability and performance.

**3.Implementation Phase:**

• Develop code based on the design specifications.

• Utilize programming languages and tools to build the software.

• In Amazon's project, developers write code to create the website's frontend, backend services, and databases, adhering to coding standards and best practices.

**4. Testing Phase:**

• Verify software functionality and performance.

• Conduct unit, integration, and system testing.

• In Amazon's project, testing ensures that features work as expected, pages load quickly, and the platform is secure against vulnerabilities and attacks.

**5. Deployment Phase:**

• Release the software to users or customers.

• Monitor system performance and address any issues.

• In Amazon's project, deployment involves rolling out updates and new features seamlessly to millions of users worldwide while ensuring high availability and reliability.

**Interconnection:**

• Each phase is interconnected and iterative.

• Feedback loops exist between phases to incorporate changes.

• For example, user feedback during deployment may inform requirements for future updates.

**Conclusion:**

• SDLC ensures systematic and efficient software development.

• Collaboration and communication are crucial throughout the process.

• Amazon's success relies on a well-executed SDLC, enabling continuous innovation and customer satisfaction.

**Assignment 2:** Explaining the Software Development Life Cycle (SDLC) using Amazon as a real-world example.

**Step-01: Requirements Gathering**

Requirements Gathering is the initial phase of the SDLC where Amazon identifies, analyzes, and documents the needs, objectives, and constraints of the software solution to be developed. Here's how Amazon approaches requirements gathering:

1.Stakeholder Identification:

• Amazon identifies stakeholders who will be affected by or have a vested interest in the software solution. Stakeholders may include end-users, customers, product managers, business analysts, executives, and regulatory bodies.

2. User Research and Feedback:

• Amazon conducts user research, surveys, interviews, and focus groups to understand the needs, preferences, and pain points of end-users and customers.

• Customer feedback, support tickets, and user analytics provide valuable insights into user behaviour, requirements, and expectations.

3. Business Analysis:

• Amazon's business analysts collaborate with stakeholders to elicit, analyze, and prioritize business requirements.

• They document functional and non-functional requirements, user stories, use cases, and acceptance criteria using tools like Confluence or Jira.

4. Market Analysis:

• Amazon conducts market analysis and competitive research to identify industry trends, emerging technologies, and customer preferences.

• Market analysis helps validate project viability, identify market gaps, and align software requirements with business goals.

5. Requirements Documentation:

• Amazon documents requirements in a Requirements Specification Document (RSD) or Product Requirements Document (PRD), detailing the scope, objectives, features, and constraints of the software solution.

• Requirements are categorized, prioritized, and traced to business goals and user needs to ensure alignment and clarity.

6. Validation and Verification:

• Amazon validates requirements with stakeholders to ensure completeness, accuracy, and alignment with business objectives.

• Verification techniques, such as requirement reviews and walkthroughs, help identify inconsistencies, ambiguities, and gaps in the requirements.

7. Change Management:

• Amazon implements change management processes to handle changes to requirements throughout the project lifecycle.

• Changes undergo impact analysis, approval, and documentation to manage scope creep and ensure project deliverables remain on track.

**Step-02: Design Phase:**

The Design Phase is a critical stage in the SDLC where Amazon's design teams translate requirements into detailed designs for the software solution. Here's how Amazon approaches the design phase:

1. Architecture Design:

• Amazon's architects define the overall system architecture, including the structure, components, modules, and interactions between subsystems.

• They select appropriate architectural patterns, such as microservices architecture or serverless architecture, based on scalability, performance, and maintainability requirements.

2. Database Design:

• Amazon's database architects design database schema, data models, tables, and relationships based on data requirements and usage patterns.

• They choose the appropriate database technologies, such as relational databases (e.g., Amazon RDS) or NoSQL databases (e.g., Amazon DynamoDB), to meet performance, scalability, and consistency requirements.

3. User Interface (UI) Design:

• Amazon's UI/UX designers create wireframes, mock-ups, and prototypes to design the user interface and user experience (UI/UX) of the software solution.

• They focus on usability, accessibility, and visual design principles to create intuitive and engaging user interfaces that meet user needs and preferences.

4. System Integration Design:

• Amazon's integration architects design interfaces, APIs, and integration points to connect the software solution with external systems, services, and data sources.

• They define message formats, communication protocols, and security mechanisms to ensure seamless integration and interoperability.

5. Security Design:

• Amazon's security architects define security requirements, controls, and mechanisms to protect the software solution against threats, vulnerabilities, and compliance requirements.

• They implement security best practices, such as encryption, authentication, authorization, and auditing, to safeguard data and infrastructure.

6.Resilience and Fault Tolerance Design:

• Amazon's resilience architects design software solutions for resilience and fault tolerance to withstand failures, disruptions, and adverse conditions.

• They implement redundancy, failover mechanisms, and distributed architectures to ensure high availability and business continuity.

**Step-03: Development Phase**

During the Development Phase, Amazon's development teams translate the design specifications and requirements into functioning software solutions. Here's how this phase unfolds at Amazon:

1. Agile Development Approach:

• Amazon employs agile development methodologies, such as Scrum or Kanban, to foster collaboration, adaptability, and iterative development.

• Development teams work in sprints, typically lasting two to four weeks, to deliver small increments of functionality called user stories or features.

2. Coding and Implementation:

• Amazon's developers write code based on the design specifications, using programming languages and frameworks suitable for the project requirements.

• They follow coding best practices, coding standards, and design patterns to ensure code quality, readability, and maintainability.

3. Collaboration and Version Control:

• Development teams collaborate using version control systems like Git, which allows them to manage code changes, track revisions, and merge code branches.

• Amazon encourages code reviews and pair programming to foster knowledge sharing, improve code quality, and identify potential issues early in the development process.

4. Continuous Integration (CI):

• Amazon implements continuous integration practices to automate the process of integrating code changes into a shared repository.

• Whenever developers commit code changes, automated build and test pipelines are triggered to validate the changes and detect any regressions or integration issues.

5. Continuous Deployment (CD):

• Amazon leverages continuous deployment pipelines to automate the deployment of software updates to production environments.

• Once code changes pass automated tests and meet quality criteria, they are automatically deployed to live environments, ensuring rapid and reliable delivery of new features and enhancements.

6. Monitoring and Feedback:

• Throughout the development phase, Amazon monitors the performance, stability, and quality of the software using monitoring tools and metrics.

• Development teams collect feedback from users, stakeholders, and automated monitoring systems to iterate on the software and make necessary adjustments.

**Step-04: Testing Phase:**

The Testing Phase is a crucial stage in the SDLC where Amazon's quality assurance (QA) teams validate the functionality, performance, and reliability of the software solution before it is deployed to production. Here's how Amazon approaches testing:

1.Test Planning:

• Amazon's QA teams collaborate with developers, product owners, and other stakeholders to create comprehensive test plans based on project requirements, user stories, and acceptance criteria.

• Test plans outline the testing objectives, scope, test cases, test data, and testing environments.

2. Types of Testing:

• Amazon conducts various types of testing to ensure the quality of its software solutions:

• Unit Testing: Developers write unit tests to verify the functionality of individual components or modules of the software.

• Integration Testing: QA teams perform integration tests to validate the interaction and interoperability of integrated components.

• System Testing: QA teams test the entire system to verify that it meets functional and non-functional requirements.

• User Acceptance Testing (UAT): End-users or stakeholders participate in UAT to validate that the software meets their expectations and business needs.

3. Test Execution:

• Amazon's QA teams execute test cases according to the test plans, using automated testing tools and manual testing techniques.

• Automated tests are run using test automation frameworks and tools, which expedite test execution, increase test coverage, and improve repeatability.

• Manual testing is performed for scenarios that cannot be easily automated, such as usability testing, exploratory testing, and ad-hoc testing.

4. Performance Testing:

• Amazon conducts performance testing to assess the responsiveness, scalability, and reliability of its software solutions under various load conditions.

• Performance tests simulate realistic user scenarios and measure key performance metrics, such as response times, throughput, and resource utilization.

5. Defect Management:

• QA teams track and manage defects using defect tracking systems or issue tracking tools, such as Jira or Bugzilla

. • Defects are categorized, prioritized, and assigned to developers for resolution. QA teams verify fixes and validate that defects are properly addressed.

**Step-05: Deployment and Maintenance:**

Deployment and Maintenance represents the stages where the software solution transitions from development to production environments and is continuously monitored, supported, and updated throughout its lifecycle. Here's how Amazon manages deployment and maintenance:

1. Continuous Deployment:

• Amazon implements continuous deployment pipelines to automate the deployment of software updates to production environments.

• Once code changes pass automated tests and meet quality criteria, they are automatically deployed to live environments using deployment automation tools and techniques.

2. Monitoring and Incident Management:

• Amazon continuously monitors the performance, stability, and availability of its software solutions using monitoring tools, metrics, and alerts.

• Automated monitoring systems detect anomalies, performance bottlenecks, and system failures, triggering alerts and notifications to operations teams.

3. Incident Response and Resolution:

• Amazon's operations teams promptly respond to incidents, outages, or performance degradations by following predefined incident response procedures and escalation paths.

• They diagnose the root cause of issues, implement remediation measures, and restore service availability as quickly as possible to minimize impact on customers and business operations.

4. Patch Management and Updates:

• Amazon regularly applies patches, updates, and security fixes to its software solutions to address vulnerabilities, bugs, and performance optimizations.

• Patch management processes ensure that software components are kept up to date with the latest releases and patches from vendors or open-source communities.

5. Capacity Planning and Scaling:

• Amazon conducts capacity planning exercises to forecast resource requirements and scalability needs for its software solutions.

• Scalability mechanisms, such as auto-scaling and load balancing, dynamically adjust resource allocation based on demand to maintain performance and availability during peak usage periods.

**Assignment 3:** Agile Principles Application - Write a two-paragraph reflection on how the Agile values of individuals and interactions, working solutions, and customer collaboration apply to the development of the community event app.

In the development of the community event app, applying Agile principles such as prioritizing individuals and interactions, working solutions, and customer collaboration is essential for success. Firstly, emphasizing individuals and interactions means fostering a collaborative and communicative environment among team members. This involves regular meetings, discussions, and brainstorming sessions to ensure everyone's voices are heard, ideas are shared, and any potential issues are addressed promptly. By prioritizing the human aspect of development, teams can leverage their collective skills and insights to create a more robust and innovative solution.

Secondly, focusing on delivering working solutions incrementally allows for rapid feedback loops and continuous improvement. Rather than waiting until the entire app is complete, Agile methodologies advocate for delivering functional pieces of software at regular intervals. This approach enables stakeholders to see tangible progress, provide feedback, and course-correct if necessary, ensuring that the final product aligns closely with user needs and expectations. Moreover, this iterative approach promotes adaptability, allowing the team to respond effectively to changes or new requirements throughout the development process. Finally, fostering strong collaboration with the community and end-users is paramount. Engaging stakeholders throughout the development lifecycle, soliciting their input, and incorporating their feedback ensures that the app addresses real-world needs and delivers value to its intended users. By involving the community in the development process, teams can build a sense of ownership and support around the app, ultimately leading to greater adoption and success.

**Assignment 4:** Scrum Framework Overview - Prepare a one-page cheat sheet on the Scrum framework that includes roles, responsibilities, artifacts, and ceremonies. Provide a brief example of a Sprint task list for the earlier mentioned app project.

**Scrum Framework Cheat Sheet**

**Roles:**

* Product Owner: Represents the stakeholders and defines the product vision, prioritizes the backlog, and accepts completed work.
* Scrum Master: Facilitates Scrum events, coaches the team, removes impediments, and ensures adherence to Scrum principles and practices.
* Development Team: Self-organizing cross-functional team responsible for delivering increments of potentially shippable products each Sprint.

**Responsibilities:**

* Product Owner: Prioritize the product backlog, define user stories, accept/reject completed work.
* Scrum Master: Facilitate Scrum events, coach the team, remove impediments, ensure adherence to Scrum values.
* Development Team: Estimate tasks, select work during Sprint planning, collaborate on development, deliver increments of working software.

**Artifacts:**

* Product Backlog: Prioritized list of features, enhancements, and fixes maintained by the Product Owner.
* Sprint Backlog: Subset of the Product Backlog items selected for the Sprint, owned by the Development Team.
* Increment: Potentially shippable product functionality delivered by the Development Team at the end of each Sprint.

**Ceremonies:**

* Sprint Planning: Collaborative session where the team selects work from the Product Backlog for the upcoming Sprint.
* Daily Standup (Daily Scrum): Daily 15-minute meeting for the Development Team to synchronize, plan the day's work, and identify any impediments.
* Sprint Review: Demo of the Increment to stakeholders, gathering feedback, and discussing adjustments to the Product Backlog.
* Sprint Retrospective: Reflection meeting where the team discusses what went well, what could be improved, and creates action items for the next Sprint.

**Example Sprint Task List for Amazon App:**

Sprint Goal: Enhance user experience and improve search functionality.

User Story 1: Improve Search Feature

* Task 1: Refine search algorithm for better accuracy.
* Task 2: Implement autocomplete feature in search bar.
* Task 3: Enhance search result filtering options.

User Story 2: Streamline Checkout Process

* Task 1: Optimize checkout page loading speed.
* Task 2: Add guest checkout option for faster purchase.
* Task 3: Implement one-click checkout feature.

User Story 3: Enhance Product Recommendations

* Task 1: Integrate machine learning model for personalized recommendations.
* Task 2: Display related products based on user browsing history.
* Task 3: Improve recommendation engine performance.

User Story 4: Implement Wishlist Feature

* Task 1: Design UI/UX for adding products to Wishlist.
* Task 2: Develop backend functionality for managing user Wishlist’s.
* Task 3: Test Wishlist feature across different devices and browsers.

User Story 5: Improve Accessibility

* Task 1: Conduct accessibility audit and address identified issues.
* Task 2: Enhance keyboard navigation for better accessibility.
* Task 3: Implement screen reader support for visually impaired users.

**Assignment 5:** Agile Project Planning - Create a one-page project plan for a new software feature using Agile planning techniques. Include backlog items with estimated story points and a prioritized list of user stories**.**

**Project Plan:** New Software Feature

**Feature Name**: Enhanced User Profile Management

**Project Overview:** The goal of this project is to improve the user profile management functionality of our application to enhance user experience and provide more personalized interactions.

**Backlog Items:**

1. Profile Creation: As a new user, I want to create a profile with basic information. (Story Points: 5)
2. Profile Editing: As a user, I want to edit my profile information (e.g., name, email, password). (Story Points: 8)
3. Profile Picture Upload: As a user, I want to upload and update my profile picture. (Story Points: 3)
4. Privacy Settings: As a user, I want to manage my privacy settings (e.g., visibility of profile information). (Story Points: 8)
5. Account Deactivation: As a user, I want to deactivate my account temporarily or permanently. (Story Points: 5)

**Prioritized User Stories:**

* + Profile Editing (8 story points)
  + Privacy Settings (8 story points)
  + Profile Creation (5 story points)
  + Account Deactivation (5 story points)
  + Profile Picture Upload (3 story points)

**Project Timeline:**

Sprint 1 (2 weeks)

* + Sprint Goal: Implement profile creation and profile editing functionality.  Backlog Items: User Story 1, User Story 2
  + Estimated Duration: 2 weeks.

Sprint 2 (2 weeks)

* + Sprint Goal: Implement profile picture upload and privacy settings functionality.  Backlog Items: User Story 3, User Story 4
  + Estimated Duration: 2 weeks.

Sprint 3 (1 week)

* + Sprint Goal: Implement account deactivation functionality and conduct testing.
  + Backlog Item: User Story 5
  + Estimated Duration: 1 week.

**Assignment 5**: Daily Standup Simulation - Write a script for a Daily Standup meeting for a development team working on the software feature from Assignment 1. Address a common challenge and incorporate a solution into the communication flow.

**Daily Standup Meeting Script** Date: 06/05/24 Time: 9:00 a.m.

Duration: 45 minutes

**Attendees:**

* [Sitamraju Venkata Vishal]
* [Shreya Addaganti]
* [Siva Veduruvada]
* [Scrum Master]

**Agenda:**

1. Brief Updates:

Each team member shares:

* + - What they worked on yesterday.
    - What they plan to work on today.
    - Any blockers or challenges they're facing.

1. Discussion:
   * + Address any blockers or challenges raised by team members.
     + Collaborate on solutions and aid overcome obstacles.
2. Action Items:
   * + Document any action items or follow-up tasks to resolve blockers.
     + Assign responsibilities and set deadlines for resolving issues.

**Meeting Script:**

* + **Scrum Master:** Good morning, everyone! Let's start our daily standup meeting. Today, we'll focus on our progress towards implementing the enhanced user profile management feature.

* + **Vishal:** Yesterday, I worked on implementing the profile creation functionality. Today, I plan to start working on the profile editing feature. No blockers.

* + **Shreya:** I worked on refining the privacy settings UI yesterday. Today, I'll continue with backend development for the privacy settings feature. No blockers.

* + **Siva:** I completed the profile picture upload functionality yesterday. Today, I'll assist with testing and start preparing for the account deactivation feature. I don't have any blockers.

* + **Scrum Master:** Thanks for the updates, everyone. It sounds like we're making good progress. However, I've noticed that we're encountering some integration issues between the frontend and backend components. This seems to be slowing down our development process. Let's discuss how we can address this challenge.

* + **Vishal:** I think we need to improve our communication and coordination between front-end and back-end teams. We should schedule regular sync-up meetings to align interface requirements and API endpoints.

* + **Shreya:** Agreed. We should also establish clear documentation for API contracts and ensure that both teams are following the same coding standards and conventions.

* + **Scrum Master:** Great suggestions, team! Let's implement these solutions immediately. I'll schedule a sync-up meeting for tomorrow to kick off the coordination efforts. In the meantime, let's update our documentation and ensure that we're following consistent coding practices.

* + **Siva:** Sounds good. I'll also reach out to the QA team to ensure that they're aware of the integration issues and can prioritize their testing accordingly.

* + **Scrum Master:** Perfect. Let's continue our momentum and keep each other informed of any progress or challenges. If anyone encounters blockers or needs assistance, don't hesitate to reach out. Remember, we're all in this together!