Module-1

Assignment

QUESTION: What is SDLC?

ANSWER: The software development lifecycle (SDLC) is the cost-effective and time-efficient process that development teams use to design and build high-quality software. The goal of SDLC is to minimize project risks through forward planning so that software meets customer expectations during production and beyond.

SDLC is essentially a series of steps or phases that provide a model for the development and lifecycle management of an application or pieces of software.

QUESTION: What is software testing?

ANSWER: Software testing is the process of checking the quality, functionality, and performance of a software product before launching. To do software testing, testers either interact with the software manually or execute test scripts to find bugs and errors, ensuring that the software works as expected**.**

QUESTION: What is agile methodology?

ANSWER: The Agile methodology is a project management approach that involves breaking the project into phases and emphasizes continuous collaboration and improvement.

QUESTION: What is SRS?

ANSWER: System Requirements Specification (SRS), also known as Software Requirements Specification, is a document or set of documentation that describes the features and behavior of a software application

QUESTION: What is oops?

ANSWER: Object-oriented programming is based on the concept of objects. In object-oriented programming data structures, or objects are defined, each with its own properties or attributes. Each object can also contain its own procedures or methods. Software is designed by using objects that interact with one another.

QUESTION: Write Basic Concepts of oops

ANSWER:

Class: A blueprint or template for creating objects. It defines the attributes (data members) and methods (functions) common to all objects of that type**.**

Object: An instance of a class. It is a concrete realization of the class blueprint, possessing its own set of attributes and methods.

Encapsulation: Encapsulation is the bundling of data (attributes) and methods that operate on the data into a single unit or class.

It hides the internal state of an object from the outside world and only exposes the necessary functionality through methods.

Inheritance: Inheritance is the mechanism by which a new class (derived class or subclass) is created from an existing class (base class or superclass).

The subclass inherits attributes and methods from its superclass, allowing for code reuse and creating a hierarchical relationship between classes.

Polymorphism: Polymorphism allows objects of different classes to be treated as objects of a common superclass.

It enables methods to behave differently based on the object that calls them. This is achieved through method overriding and method overloading.

Abstraction: Abstraction involves hiding the complex implementation details of a class and only showing the essential features of the object.

It allows programmers to focus on what an object does rather than how it does it, making the code more manageable and scalable.

These concepts form the foundation of Object-Oriented Programming and are fundamental for designing modular, reusable, and maintainable software systems.

QUESTION: What is object?

ANSWER: Object gives the permission to access functionality of class.

QUESTION: What is Class?

ANSWER: Class is a collection of data member and member function.

QUESTION: What is encapsulation?

ANSWER: The process wrapping the data in a single limit and secure the data from outside world.

QUESTION: What is inheritance?

ANSWER: Making a class from an existing class deriving the attribute of some other class.

QUESTION: What is polymorphism?

ANSWER: One name and multiple form

QUESTION: Write SDLC phases with basic introduction

ANSWER:

Requirement gathering/collection: Establish customer needs.

Analysis: Model and specify the requirements what.

Design: Model and specify a solution why.

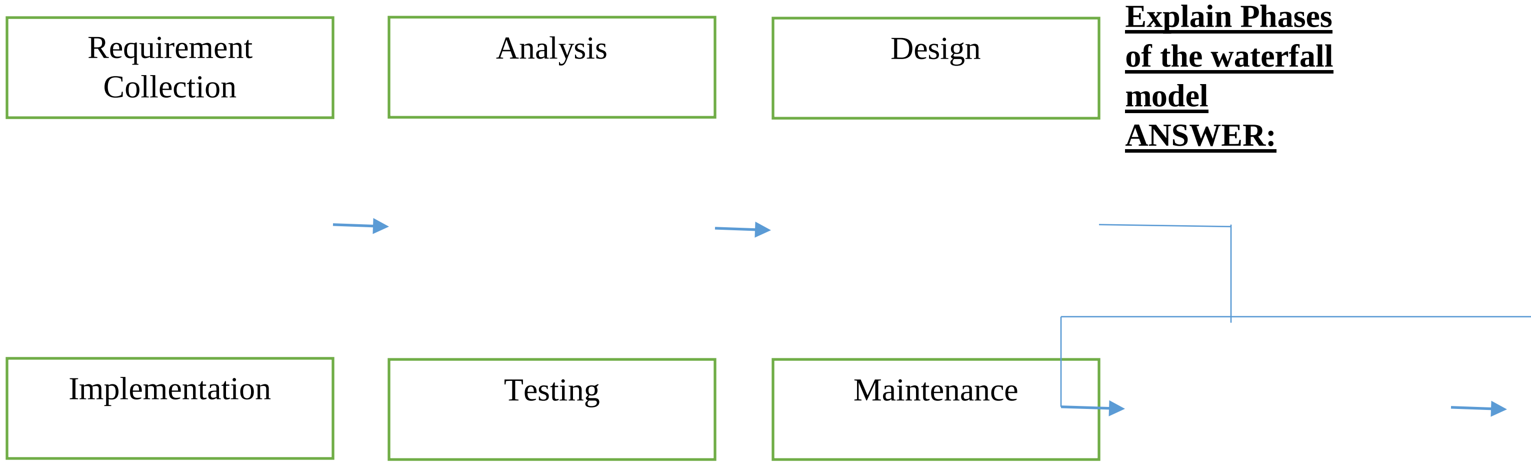
Implementation: Construct a solution in software.

Testing: Validate the solution against the requirement.

Maintenance: Repair defects and adapt the solution to the new requirements.

QUESTION: Explain Phases of the waterfall model

ANSWER:



QUESTION: Write phases of spiral model

Answer:

Planning: In this phase, the objectives, constraints, and alternatives for software development are identified.

Key activities include determining project scope, defining deliverables, establishing schedules, and identifying resources.

Risk Analysis: The primary focus of this phase is to identify potential risks and uncertainties associated with the project.

Risks are analyzed in terms of their impact on project objectives, and strategies are devised to mitigate or manage these risks.

Engineering (Development): In this phase, the actual development of the software occurs.

Requirements are gathered, design decisions are made, and code is written and tested.

This phase may involve multiple iterations, with each iteration adding new functionality or refining existing features.

Evaluation (Review): After each iteration of development, the project is evaluated to assess progress, identify issues, and review the product against the requirements.

Stakeholders provide feedback, and necessary adjustments are made to the project plan or product.

These phases form a spiral, where each loop represents a cycle of development. The spiral model emphasizes the iterative nature of software development, allowing for flexibility, risk management, and continuous improvement throughout the project lifecycle. Each cycle through the spiral provides an opportunity to refine the product based on feedback and insights gained from previous iterations.

QUESTION: Write agile manifesto principle

Answer:

The Agile Manifesto outlines four key principles that guide agile software development. These principles prioritize individuals and interactions, working software, customer collaboration, and responding to change over following rigid processes and documentation. Here are the principles:

Individuals and Interactions over Processes and Tools: Agile values the importance of people and their interactions within a team over relying solely on processes and tools. It emphasizes effective communication, collaboration, and teamwork to achieve project goals.

Working Software over Comprehensive Documentation: Agile prioritizes delivering working software to customers at regular intervals over extensive documentation. While documentation is important, Agile teams focus on producing tangible results and continuously improving the product based on feedback.

Customer Collaboration over Contract Negotiation: Agile encourages active involvement and collaboration with customers throughout the development process. Rather than relying on fixed contracts and negotiations, Agile teams work closely with customers to understand their needs, gather feedback, and adapt the product accordingly.

Responding to Change over Following a Plan: Agile recognizes the inevitability of change in software development and values the ability to respond to change quickly and effectively. Agile teams prioritize flexibility and adaptability, adjusting plans and priorities as new information emerges or customer needs evolve.

These principles guide agile teams in delivering high-quality software that meets customer requirements while fostering collaboration, adaptability, and continuous improvement throughout the development process.

QUESTION: Explain working methodology of agile model and also write pros and cons.

Answer: The Agile methodology is an iterative and incremental approach to software development, emphasizing flexibility, collaboration, and customer satisfaction. It is based on the Agile Manifesto and its principles, which prioritize individuals and interactions, working software, customer collaboration, and responding to change over following rigid processes and documentation. Here's an overview of the working methodology of Agile:

Iterative Development: Agile projects are divided into small, manageable increments called iterations or sprints.
Each iteration typically lasts from one to four weeks and results in a potentially shippable product increment.
At the end of each iteration, the team delivers working software with added or enhanced features.
Cross-functional Teams: Agile teams are typically cross-functional, comprising members with diverse skills (e.g., developers, testers, designers).
The team collaborates closely throughout the project, working together to deliver value to the customer.
Continuous Planning and Feedback: Agile projects involve continuous planning, with requirements and priorities evolving throughout the development process**.**

Stakeholders, including customers and end-users, provide feedback on each increment, guiding future iterations and ensuring alignment with their needs.

Emphasis on Communication and Collaboration: Agile emphasizes communication and collaboration within the team and with stakeholders.

Daily stand-up meetings, regular demos, and retrospectives facilitate transparency, feedback, and continuous improvement.

Adaptability and Flexibility: Agile embraces change and responds to it quickly, allowing for adjustments to requirements, priorities, and project scope.

The iterative nature of Agile enables teams to adapt to evolving circumstances and customer feedback.

Pros of Agile methodology:

Flexibility: Agile allows for flexibility and adaptability, enabling teams to respond to change quickly and effectively.

Customer Satisfaction: Agile prioritizes customer collaboration and delivers working software increments regularly, leading to increased customer satisfaction.

Early and Continuous Delivery: Agile focuses on delivering value early and frequently, allowing for faster time-to-market and feedback-driven improvements.

Transparency and Collaboration: Agile fosters transparency, communication, and collaboration within the team and with stakeholders, leading to better alignment and shared understanding.

Reduced Risk: By breaking the project into small iterations, Agile reduces the risk of project failure and enables early identification and mitigation of issues.

Cons of Agile methodology:

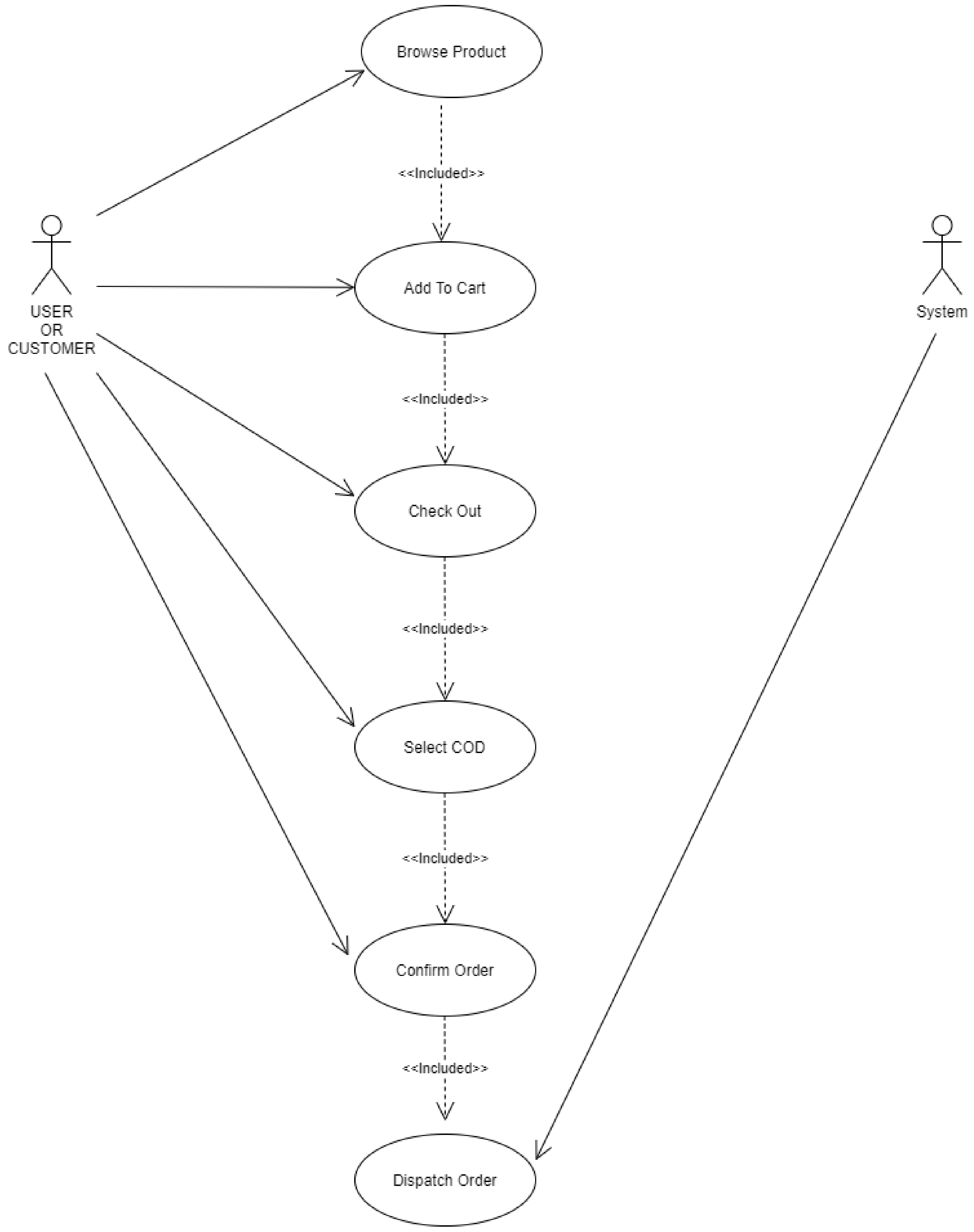
Requires Experienced Team: Agile requires a skilled and experienced team capable of self-organization and collaboration.

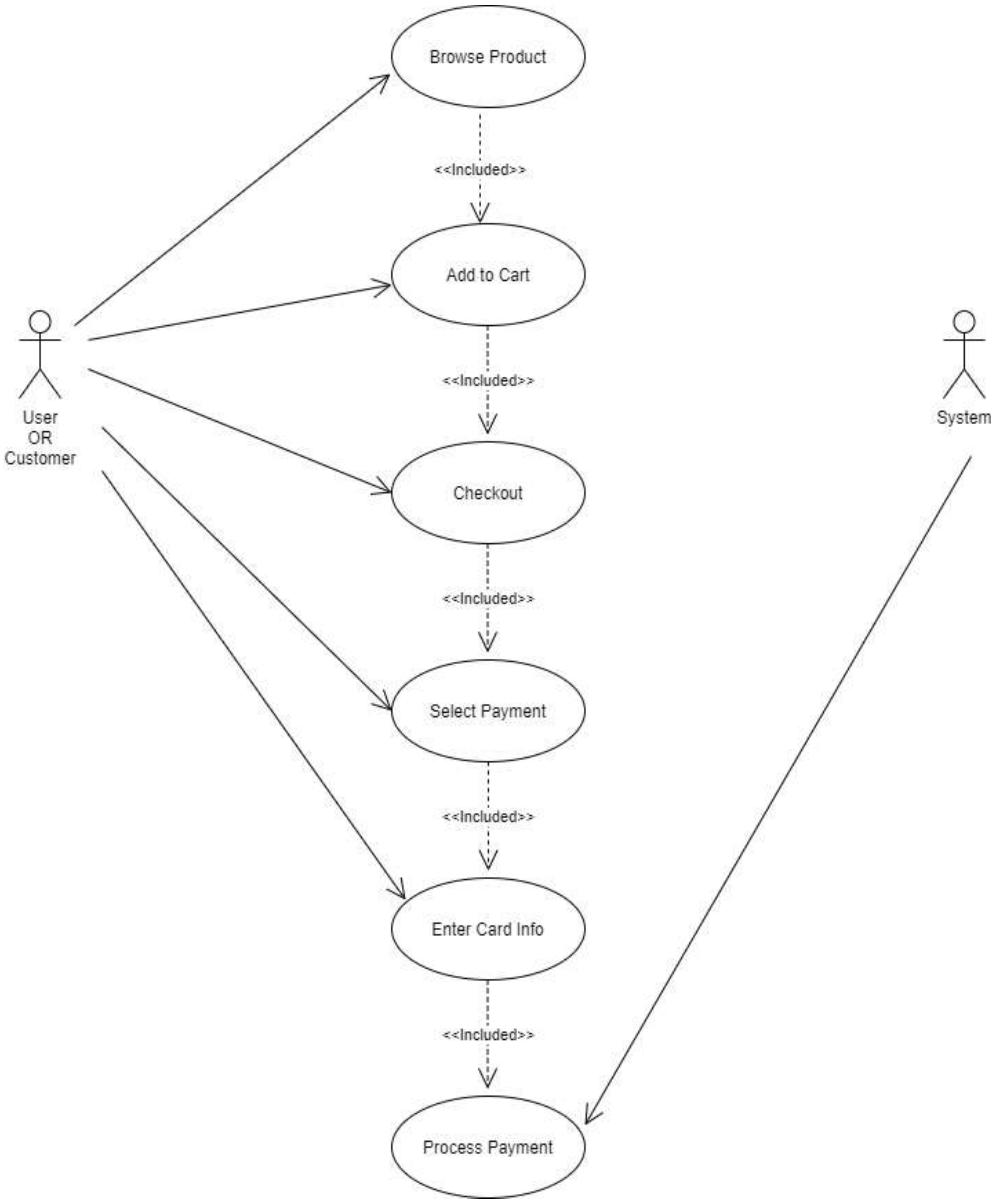
Lack of Predictability: The flexible nature of Agile can make it challenging to predict project timelines and costs accurately.

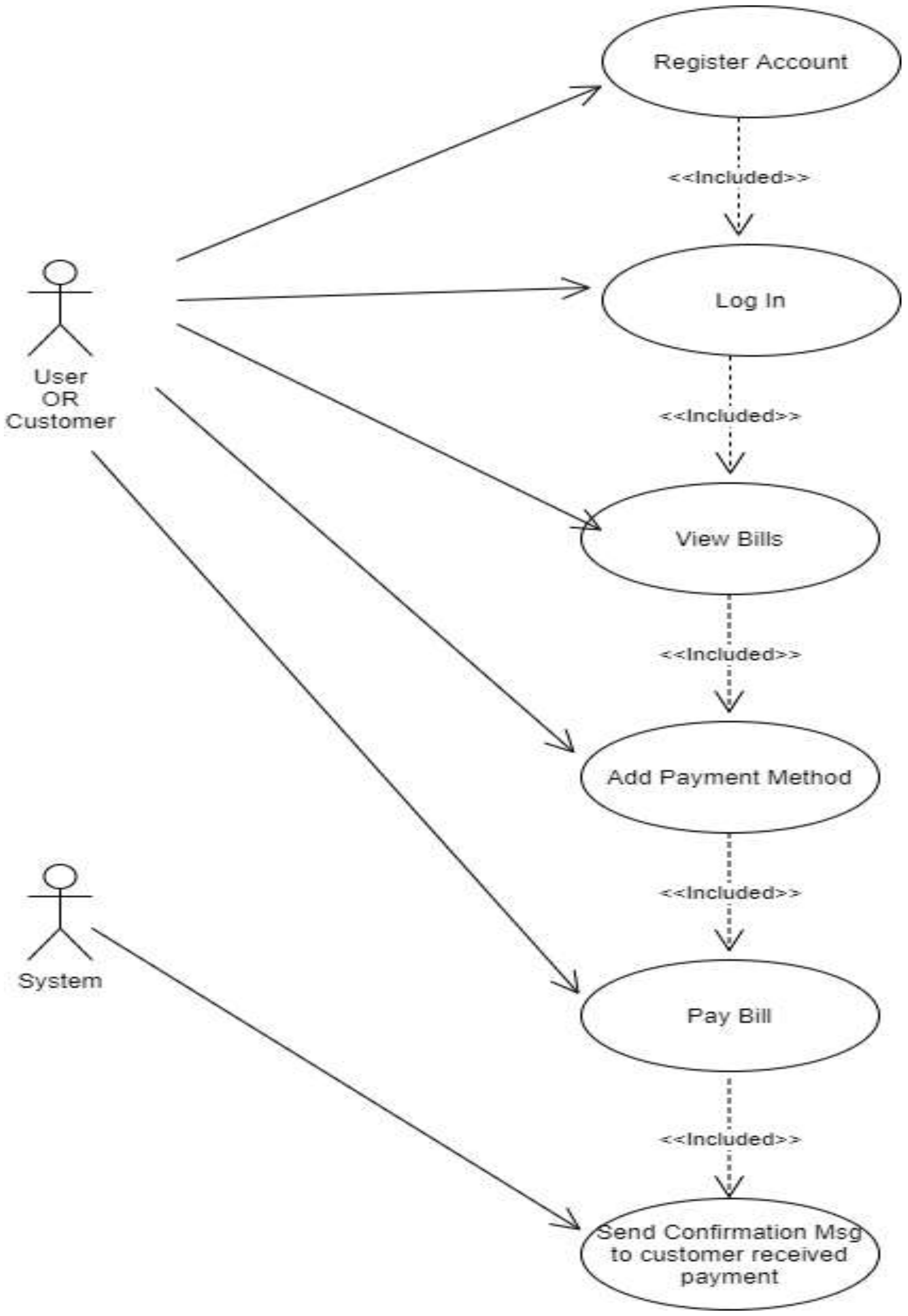
Dependency on Customer Availability: Agile relies on frequent customer collaboration, which may be challenging if customers are not readily available or engaged.

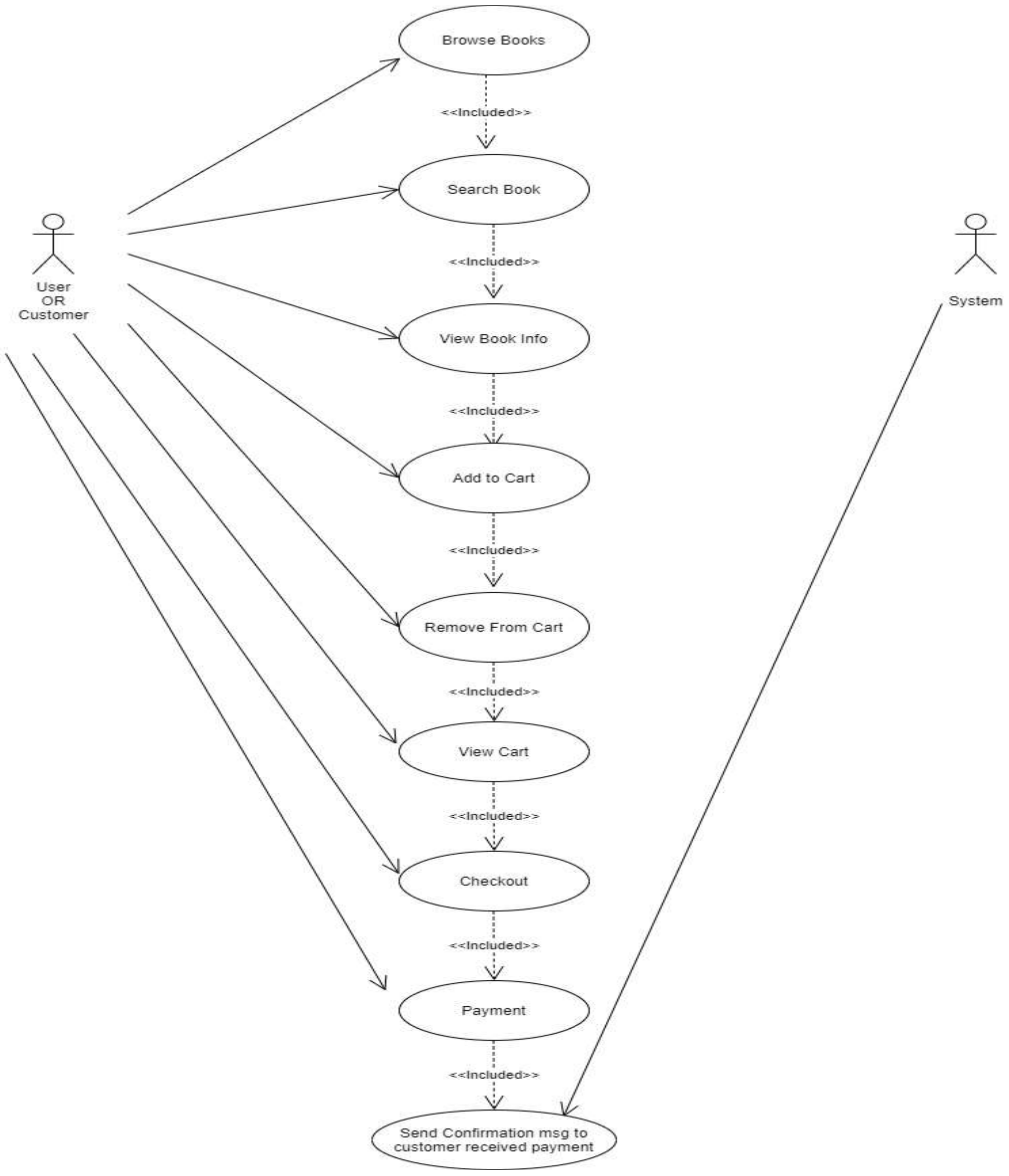
Documentation Challenges: Agile prioritizes working software over comprehensive documentation, which may lead to documentation gaps or inconsistencies.

Scope Creep: Agile's emphasis on accommodating change may lead to scope creep if not managed effectively, potentially impacting project timelines and budgets.


Browse Product
<<Included>>
Add To Cart
USER
OR
CUSTOMER
System
<<Included>>
Check Out
<<Included>>
Select COD
<<Included>>
Confirm Order
<<Included>>
Dispatch Order


Browse Product
<<Included>>
Add to Cart
<<Included>>
Checkout
<<Included>>
Select Payment
<<Included>>
Enter Card Info
<<Included>>
Process Payment
User
OR
Customer
System


Register Account
<<Included>>
Log In
<<Included>>
View Bills
<<Included>>
Add Payment Method
<<Included>>
Pay Bill
<<Included>>
Send Confirmation Msg to customer received payment
User OR Customer
System


Browse Books
<<Included>>
Search Book
<<Included>>
View Book Info
<<Included>>
Add to Cart
<<Included>>
Remove From Cart
<<Included>>
View Cart
<<Included>>
Checkout
<<Included>>
Payment
<<Included>>
Send Confirmation msg to customer received payment
User OR Customer
System