Software testing Assignment

Module -1(Fundamental)

Name: - Hemal Patil

1.what is SDLC?

A-SDLC is a structure imposed on the development of a software product that defines the process for planning, implementation, testing, documentation, deployment, and ongoing maintenance and support. There are a number of different development models. A Software Development Life Cycle is essentially a series of steps, or phases, that provide a model for the development and lifecycle management of an application or piece of software.

2. what is software testing?

A-Software testing is a process of executing a program or application with the intent of finding the software bugs. Testing is the process of evaluating a system or its component(s) with the intent to find that whether it satisfies the specified requirements or not. This activity results in the actual, expected and difference between their results. In simple words testing is executing a system in order to identify any gaps, errors or missing requirements in contrary to the actual desire or requirements

3.what is Agile Methodology?

A-Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing.

4.what is SRS?

A-A software requirements specification (SRS) is a complete description of the behavior of the system to be developed. It includes a set of use cases that describe all of the interactions that the users will have with the software. Use cases are also known as functional requirements. In addition to use cases, the SRS also contains nonfunctional (or supplementary) requirements. Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance requirements, quality standards, or design constraints).

5.what is OOPS?

A-Object Oriented Programming System , Object means a real word entity such as pen, chair, table etc. Object-Oriented Programming is a methodology or paradigm to design a program using classes and objects.

6.Writedown basic concept of OOPS?

A-It simplifies the software development and maintenance by providing some concepts:

• Object • Class • Inheritance • Polymorphism • Abstraction • Encapsulation

7.what is object?

A-object gives the permission to Access functionality of class.

8.what is class?

A-class contains data member and member function

9.what is encapsulation?

A-Wrapping of Data.Adv- to secure the data.

10.what is inheritance?

A-creating a class from an existing class. Adv- code Redundancy.

11.write SDLC phases with basic introduction?

A-1.Requirements Collection/Gathering Establish Customer Needs

Requirement Gathering Features Usage scenarios Although requirements may be documented in written form, they may be incomplete, unambiguous, or even incorrect. Requirements will Change! Inadequately captured or expressed in the first place User and business needs change during the project

2.Analysis Model And Specify the requirements- “What”

The analysis phase defines the requirements of the system, independent of how these requirements will be accomplished. This phase defines the problem that the customer is trying to solve. The deliverable result at the end of this phase is a requirement document. Ideally, this document states in a clear and precise fashion what is to be built. This analysis represents the “what” phase

3.Design Model And Specify a Solution – “Why”

Design Phase Design Architecture Document Implementation Plan Critical Priority Analysis Performance Analysis Test Plan The Design team can now expand upon the information established in the requirement document.

4.Implementation Construct a Solution In Software

In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be

narrowly designed for this particular system, or the component may be made more general to satisfy a reusability guideline. Implementation - Code

5.Testing Validate the solution against the requirements

A customer satisfied with the quality of a product will remain loyal and wait for new functionality in the next version. Quality is a distinguishing attribute of a system indicating the degree of excellence. Regression Testing Internal Testing Unit Testing Application Testing Stress Testing

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

The testing phase is a separate phase which is performed by a different team after the implementation is completed. There is merit in this approach; it is hard to see one's own mistakes, and a fresh eye can discover obvious errors much faster than the person who has read and re-read the material many times. Unfortunately, delegating (alternate) testing to another team leads to as lack (dull) attitude regarding quality by the implementation team. If the teams are to be known as craftsmen, then the teams should be responsible for establishing high quality across all phases.

12.explain phases of water fall model?

A- The waterfall is unrealistic for many reasons, especially: Requirements must be "frozen" to early in the life cycle Requirements are validated too late

A-1.Requirements Collection/Gathering Establish Customer Needs

Requirement Gathering Features Usage scenarios Although requirements may be documented in written form, they may be incomplete, unambiguous, or even incorrect. Requirements will Change! Inadequately captured or expressed in the first place User and business needs change during the project

2.Analysis Model And Specify the requirements- "What"

The analysis phase defines the requirements of the system, independent of how these requirements will be accomplished. This phase defines the problem that the customer is trying to solve. The deliverable result at the end of this phase is a requirement document. Ideally, this document states in a clear and precise fashion what is to be built. This analysis represents the "what" phase

3.Design Model And Specify a Solution – "Why"

Design Phase Design Architecture Document Implementation Plan Critical Priority Analysis Performance Analysis Test Plan The Design team can now expand upon the information established in the requirement document.

4.Implementation Construct a Solution In Software

In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be

narrowly designed for this particular system, or the component may be made more general to satisfy a reusability guideline. Implementation - Code

5.Testing Validate the solution against the requirements

A customer satisfied with the quality of a product will remain loyal and wait for new functionality in the next version. Quality is a distinguishing attribute of a system indicating the degree of excellence. Regression Testing Internal Testing Unit Testing Application Testing Stress Testing

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

The testing phase is a separate phase which is performed by a different team after the implementation is completed. There is merit in this approach; it is hard to see one's own mistakes, and a fresh eye can discover obvious errors much faster than the person who has read and re-read the material many times. Unfortunately, delegating (alternate) testing to another team leads to as lack (dull) attitude regarding quality by the implementation team. If the teams are to be known as craftsmen, then the teams should be responsible for establishing high quality across all phases.

13.write phases of spiral model?

A-Spiral Model is very widely used in the software industry as it is in synch with the natural development process of any product i.e. learning with maturity and also involves minimum risk for the customer as well as the development firms. Following are the typical uses of Spiral model: When costs there are a budget constraint and risk evaluation is important. For medium to high-risk projects. Long-term project commitment because of potential changes to economic priorities as the requirements change with time. Customer is not sure of their requirements which are usually the case. Requirements are complex and need evaluation to get clarity. New product line which should be released in phases to get enough customer feedback. Significant changes are expected in the product during the development cycle

Pros (Why It works) -Changing requirements can be accommodated. Allows for extensive use of prototypes Requirements can be captured more accurately. Users see the system early. Development can be divided into smaller parts and more risky parts can be developed earlier which helps better risk management.

Cons (Why It doesn't work) -Management is more complex. End of project may not be known early. Not suitable for small or low risk projects and could be expensive for small projects. Process is complex Spiral may go indefinitely. Large number of intermediate stages requires excessive documentation

14.write Agile Manifesto principle?

Agile Principles

● Customer satisfaction through early and continuous software delivery – Customers are happier when they receive working software at regular intervals, rather than waiting extended periods of time between releases.

● Accommodate changing requirements throughout the development process – The ability to avoid delays when a requirement or feature request changes.

● Frequent delivery of working software – Scrum accommodates this principle since the team operates in software sprints or iterations that ensure regular delivery of working software.

● Collaboration between the business stakeholders and developers throughout the project – Better decisions are made when the business and technical team are aligned.

● Support, trust, and motivate the people involved – Motivated teams are more likely to deliver their best work than unhappy teams.

● Enable face-to-face interactions – Communication is more successful when developmentteamsareco-located396

● Working software is the primary measure of progress – Delivering functional software to the customer is the ultimate factor that measures progress.

● Agile processes to support a consistent development pace – Teams establish a repeatable and maintainable speed at which they can deliver working software, and they repeat it with each release.

● Attention to technical detail and design enhances agility – The right skills and good design ensures the team can maintain the pace, constantly improve the product, and sustain change.

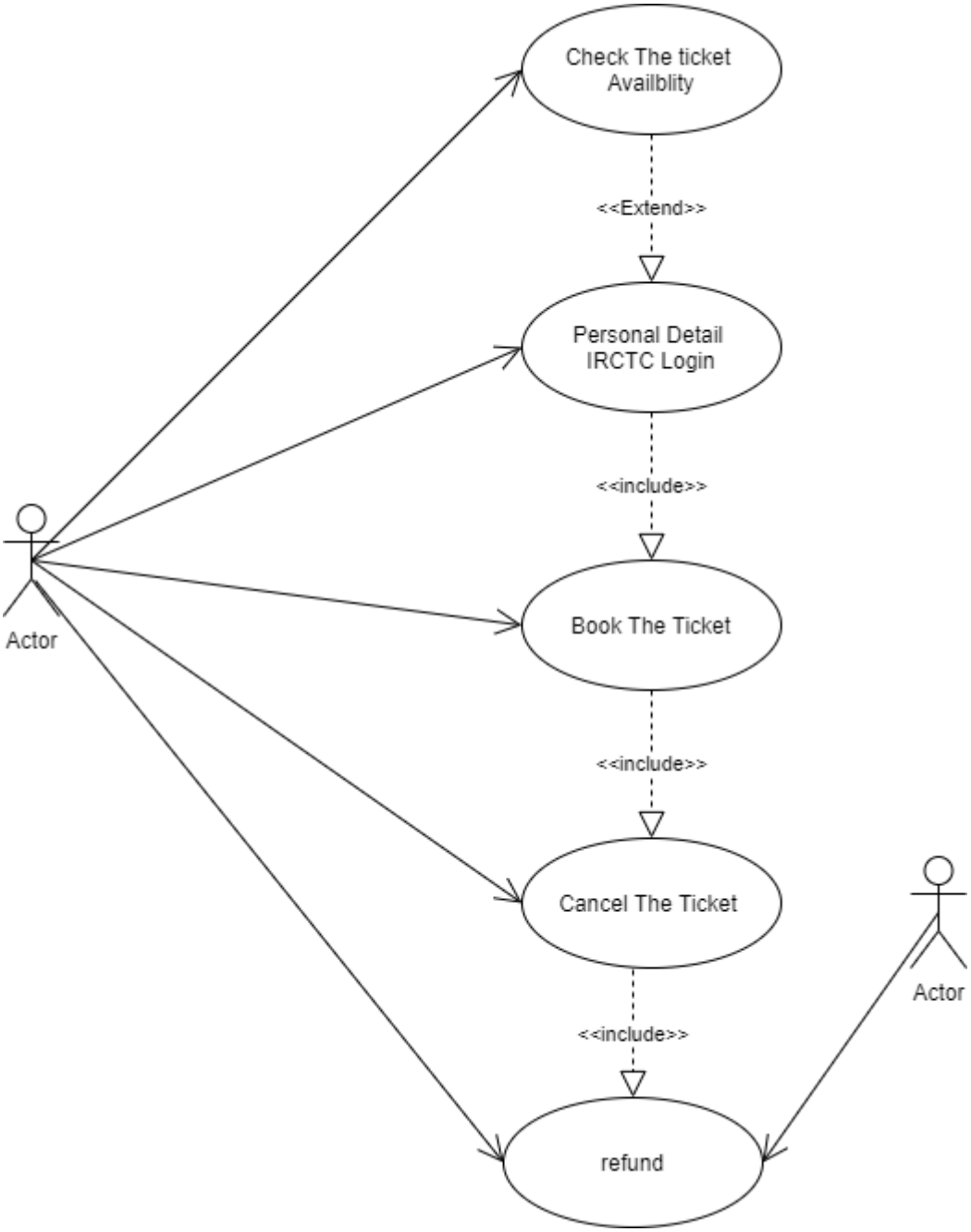
● Simplicity – Develop just enough to get the job done for right now.

● Self-organizing teams encourage great architectures, requirements, and designs – Skilled and motivated team members who have decision-making power, take ownership, communicate regularly with other team members, and share ideas that deliver quality products.

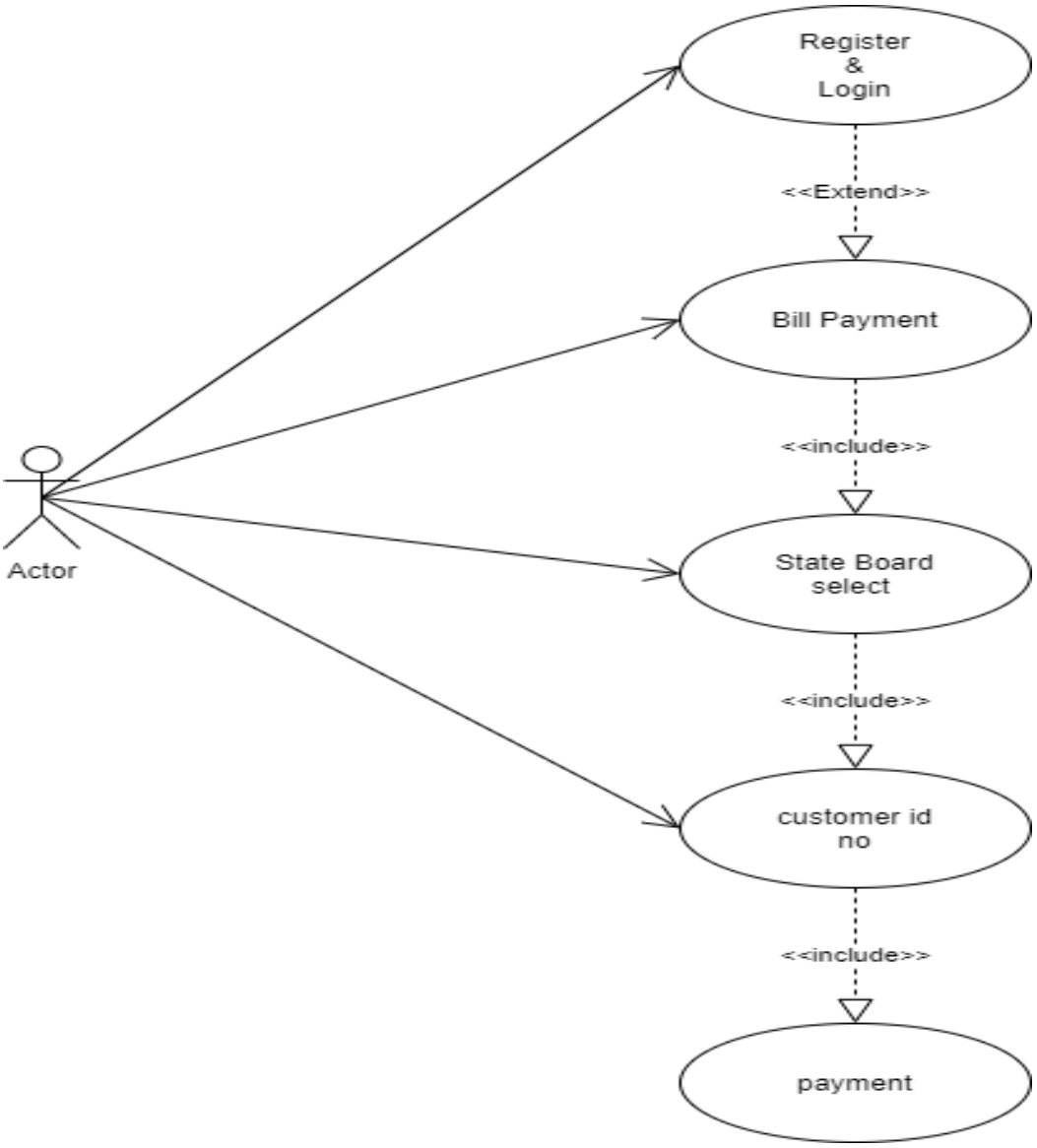
● Regular reflections on how to become more effective – Self-improvement, process improvement, advancing skills, and techniques help team members work more efficiently. 397 Scrum

● Scrum: SCRUM is an agile development met

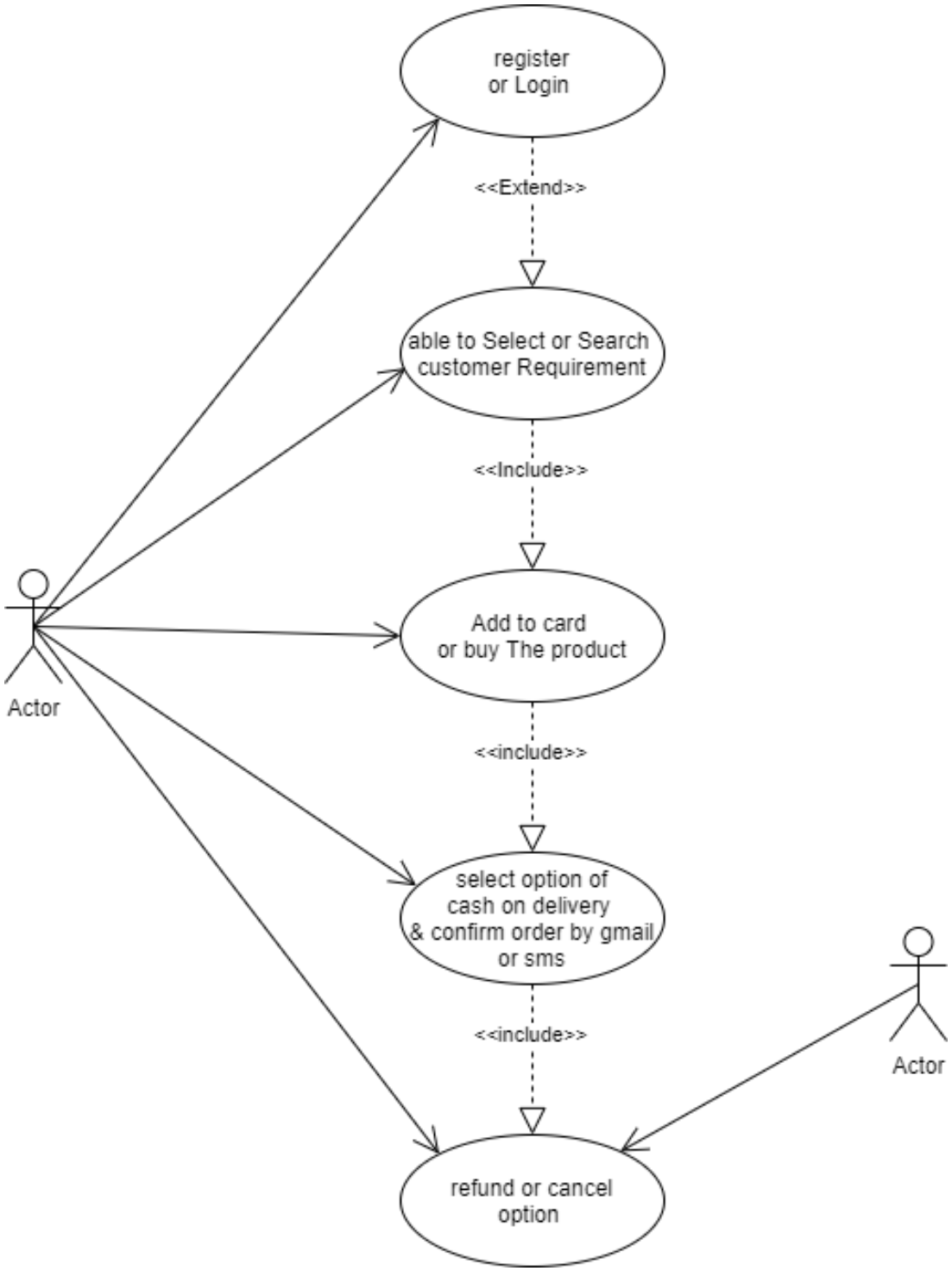
15.draw use case on online book a Ticket ?



16.draw the usercase online bill payment system?



17.draw use case on online shopping product using cod?



18. draw use case on online shopping product using payment gateway?