Assignment

* what is testing?
* Testing is executing a system in order to identify and gap, error,

or missing requirement in contrary to the actual desire or [[1]](#endnote-1)

requirement.

* what is 7 key principles? explain in detail.
* Defect
* Exhaustive
* Early testing
* Defect clustering
* Pesticide paradox
* Testing is context dependent
* Absence of error fallacy.
* Defects: - testing cannot prove that there are no defects

Present.

* We test to find faults.
* Exhaustive testing is impossible: -
* Testing everything including all combination of input

And preconditions is not possible.

* Early testing: -
* Testing activity should start as early as possible in the

Software or system development life cycle and should

Be focused on defined objectives.

1. Defect clustering: -

   - Defect are not evenly spread in system

   - they are clustered

   - In other words, most defect found during

   Testing are usually confined to a small

   Number of modules.

   * Pesticide paradox: -
   * If the same test are repeated over and over again,

   Eventually the same set of test cases will no longer

   Find any new defect.

   * To overcome this “pesticide paradox” the test

   Causes need to be regularly reviewed and revised

   And new and different parts of the software or

   System to potentially find more defects.

   * Testing is context dependent: -
   * Testing is basically context dependent.
   * Testing is done differently in different contexts.

   Example -safety critical software is tested

   Differently from an e-commerce site.

   * 3 to 10 failures per thousand lines of cod ( kloc)

   Typical for commercial software

   * 1 to 3 failures per kloc typical for industrial

   Software

   * 0.01 failure per kloc for Nasa shuttle code!
   * Absence of errors fallacy
   * If the system built is unusable and dose not

   Fulfill the user’s need and expectation then

   Finding and fixing defect does not help.

   * Even after defect have been resolved it may

   Still be un usable and /or dose not fulfil the

   User’s need and expectation

   WHAT IS SDLC?

   A Software Development Life Cycle is essentially a series or

   Phases that provide a model for the development and

   Life cycle management of an application or piece of software.

   **Phases of SDLC:-**

   Requirement Gathering (what is the problem?)

   1 customer needs

   2 requirement from stakeholder, client, customer, CEO etc

   3 improvements in current soft ware

   **Analysis Phase (what we want?)**

   1 Details on computer programming languages and environments,

   machines, packages, application architecture, distributed

   architecture layering, memory size, platform, algorithms, data

   structures, global type definitions, interfaces, and many other

   2 risks of the project

   3 costs of the project

   4 time for completion

   **Design (How can we get what we want?)**

   1 Design Architecture Document

   2 Implementation Plan

   3 Critical Priority Analysis

   4 Performance Analysis

   5 Test Plan

   **Implementation (create what we want?)**

   1 In the implementation phase, the team builds the

   Components either from scratch on by composition.

   2 Implementation - Code

   3 Critical Error Removal

   **Testing (did we get what we want)**

   1 we test the build to check for defect

   2 we report the defect and get it fixed

   3 we retest the build until it fulfils customer requirement.

   **Deployment** -project live then it will become ma product

   **Maintenance**

   1 Corrective maintenance: identifying and repairing defects

   2 Adaptive maintenances: adapting the existing solution to

   the new platforms.

   3 Perfective Maintenance: implementing the new

   Requirement.

   Software Testing Methodologies

   Introduction

   Waterfall Model/Methodology (Classical

   Software Cycle)

   Iterative & Incremental Model/Methodology

   Spiral Model/Methodology

   Agile Model/Methodology

   Use Case

   Software Requirement Specification (SRS)

   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 the interactions

   users will have with the software.

   What is oops?

   Object oriented programming is way of writing the

   Programs is organized way

   Object are like a black box where data are hidden.

   Write basic concepts of oops?

   **1 what is object?**

   object gives the permission

   To access functionality of class.

   **2 what is class?**

   Class is a collection of data member and

   Member function int a=10 b=20 void

   Function (intr,inty)(a+b)

   What is encapsulation?

   The process wrapping the data in a single

   Unit to secure the data from outside world

   What is inheritance?

   Making a class from an existing class. Delving

   The attribute of some other class.

   ADV redundancy, extensibility

   What is polymorphism?

   One name multiple from

   **Type over riding**

   Same name of function with same parameter but

   Definition will be different.

   **Over loading**

   Function over loding same function name but

   Different parameter.

   **Abstraction**

   Hiding detail and showing only essential

   Information.

   What is agile mythology?

   It is combination iterative and increment modal

   It is dividing the software in to small incremental

   Builds this are provided in iteration that means

   The big project are divided in to small chunks.

   Each iteration last about one to three week

   Each iteration involves all the team member working

   Each iteration involves all the team member working

   Simultaneously on areas like planning requirement

   Analysis design coding util testing and acceptance

   Testing .

   After the resease we check for the feedbacks of the

   Deployed soft ware.

   * Pesticide paradox:-
   * If the same test are repeated over and over again,

   Eventually the same set of test cases will no longer

   Find any new defect.

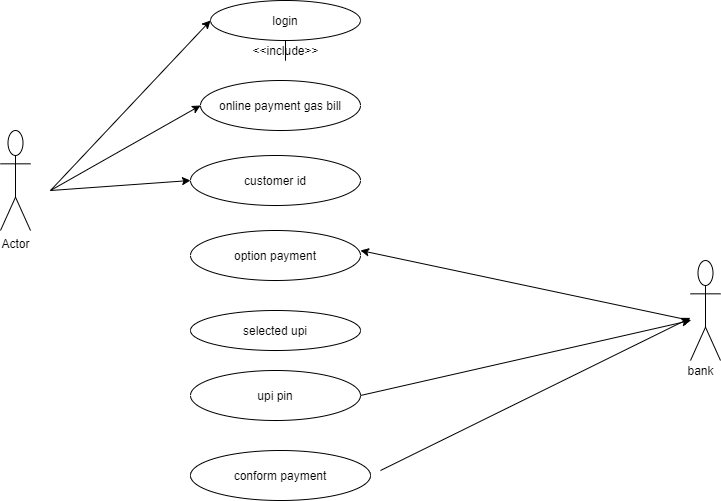
   * To overcome this “ pesticide paradox” the test

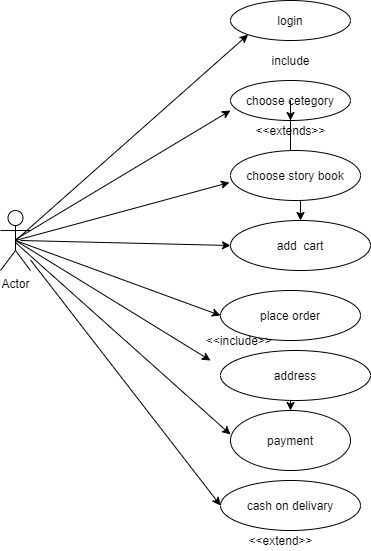
   Causes need to be regularly reviewed and revised

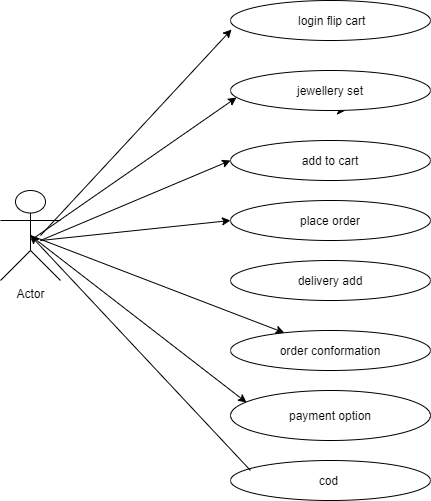
   And new and different parts of the software or

   System to potentially find more defect.

   Use case

   [↑](#endnote-ref-1)