

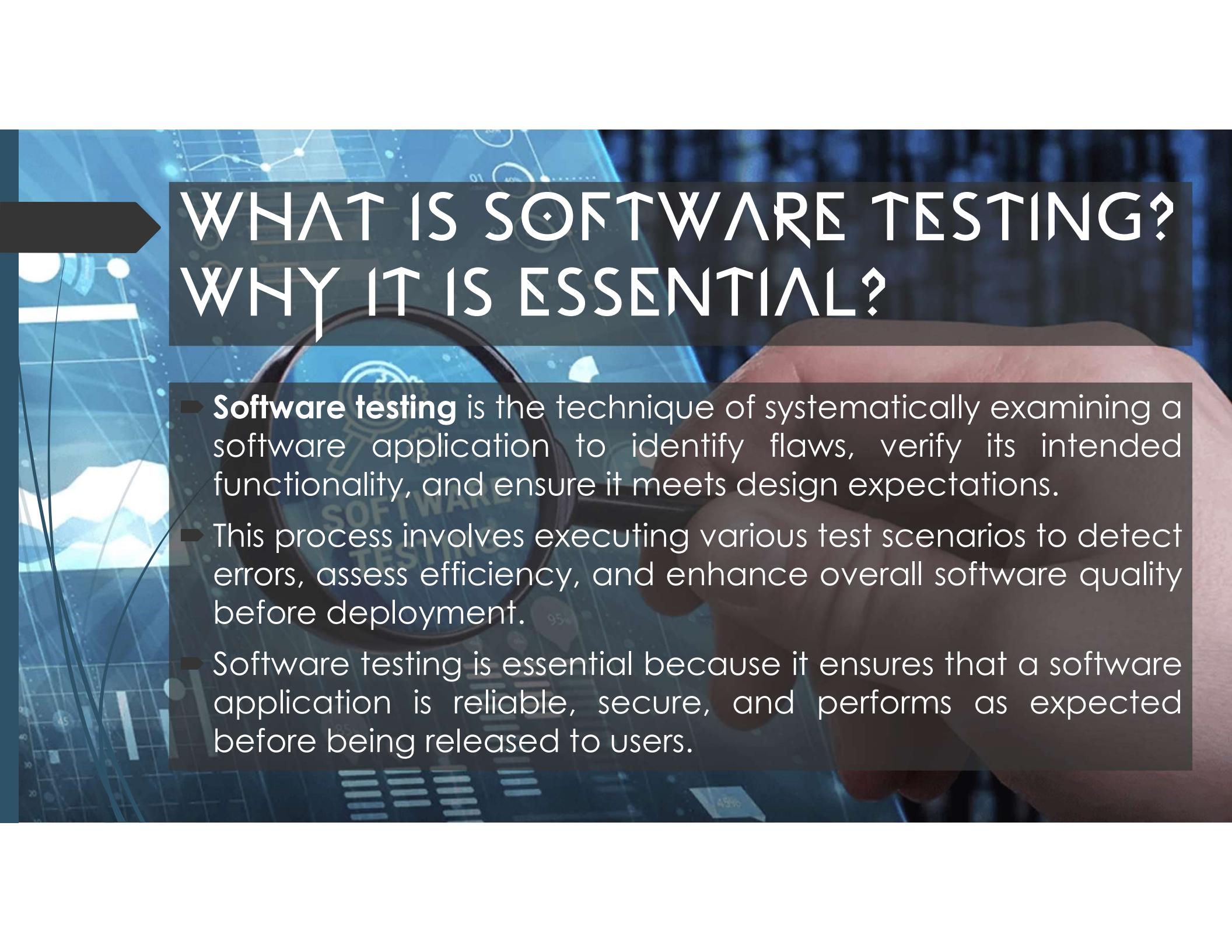


QUALITY ASSURANCE

SOFTWARE TESTING ASSIGNMENT

MODULE - 1

DARSHAN GONIL



WHAT IS SOFTWARE TESTING? WHY IT IS ESSENTIAL?

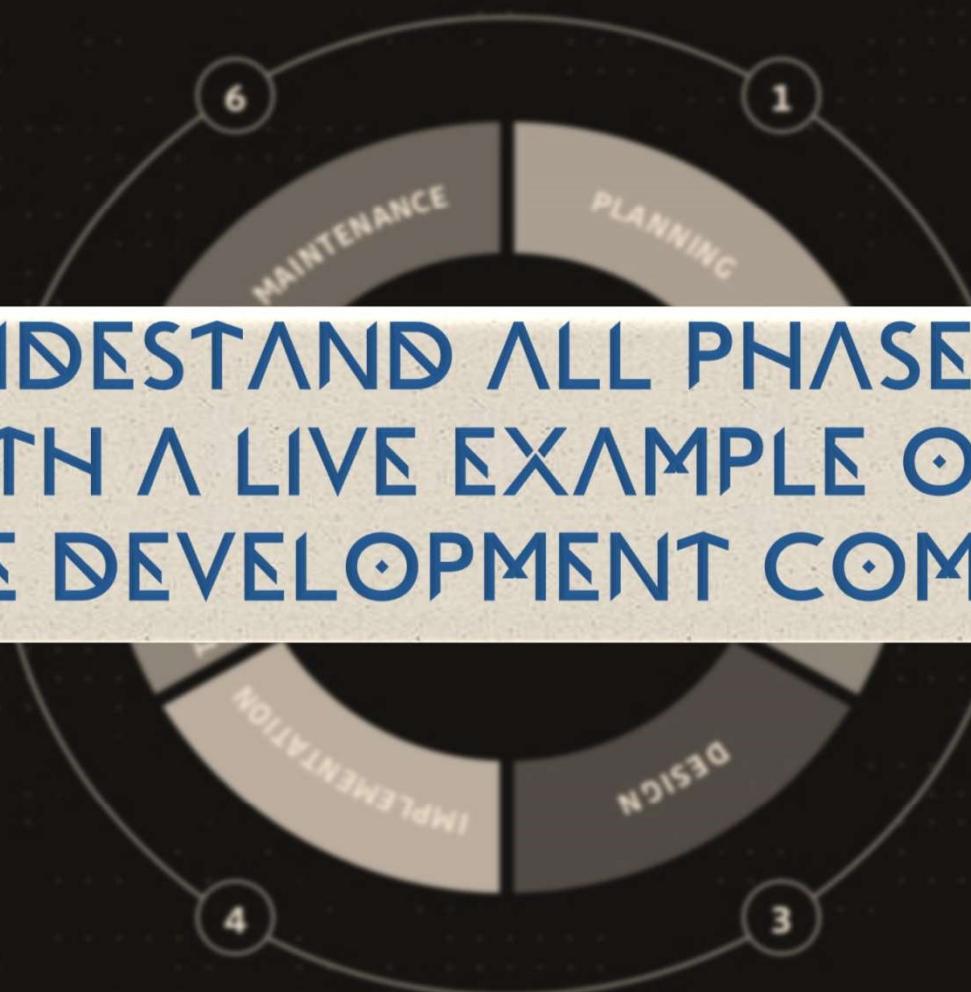
- **Software testing** is the technique of systematically examining a software application to identify flaws, verify its intended functionality, and ensure it meets design expectations.
- This process involves executing various test scenarios to detect errors, assess efficiency, and enhance overall software quality before deployment.
- Software testing is essential because it ensures that a software application is reliable, secure, and performs as expected before being released to users.

WHAT IS SDLC?

- ▶ SDLC is called as SOFTWARE DEVELOPMENT LIDE CYCLE.
- ▶ It is a process use to design, develop, test & deploy software efficiently.
- ▶ Without this process software can't begin.
- ▶ It consists six major elements which as follows:
 - ▶ REQUIREMENT & PLANNING
 - ▶ ANALYSIS
 - ▶ DESIGN
 - ▶ IMPLEMENTATION
 - ▶ TESTING & INTEGRATION
 - ▶ MAINTENANCE



**LET US UNDERSTAND ALL PHASES OF SDLC
WITH A LIVE EXAMPLE OF A
GAME DEVELOPMENT COMPANY.**



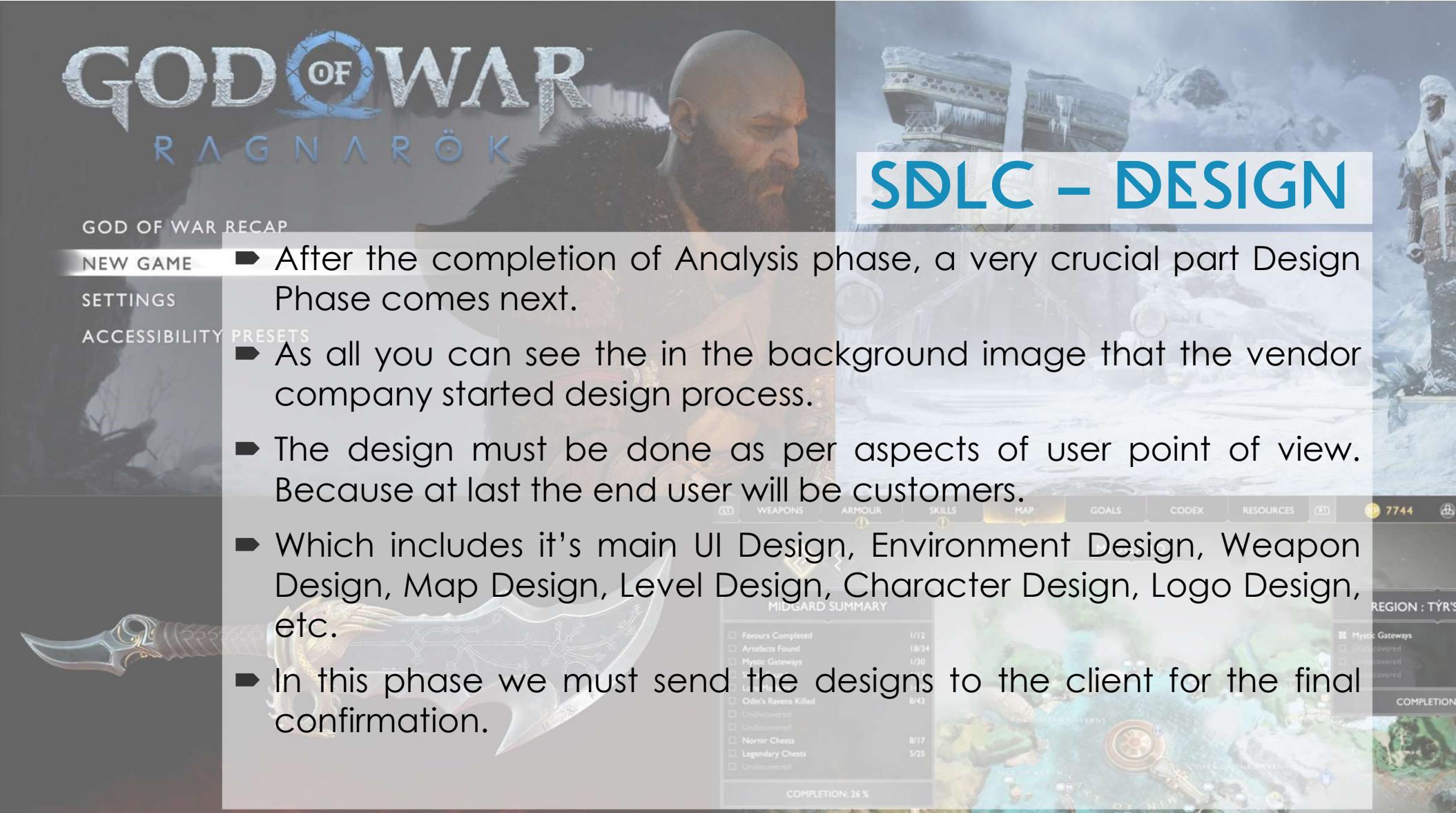
SDLC - REQUIREMENT

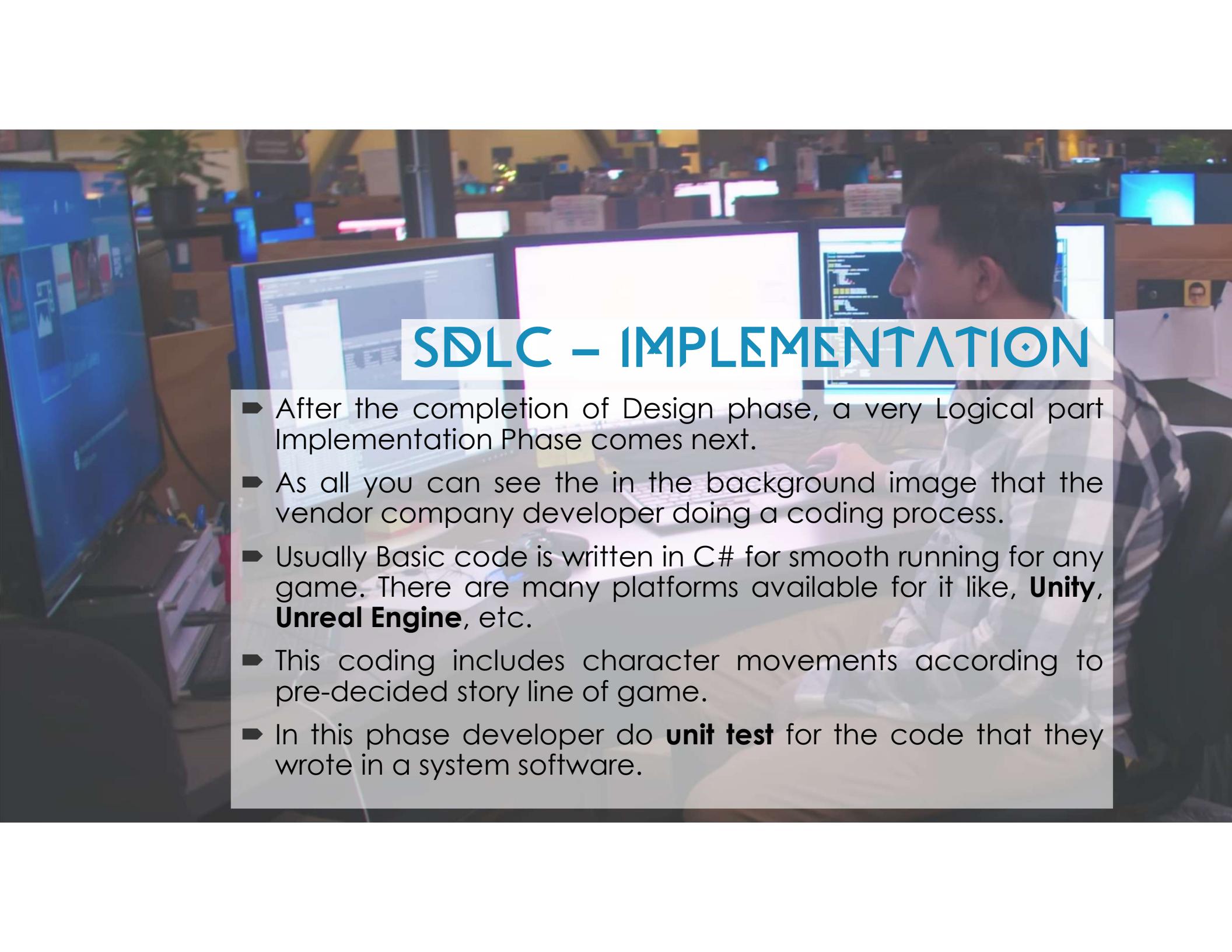
- ▶ Client Company Name: Sony Interactive Entertainment.
- ▶ Vendor Company Name: Santa Monica Studio.
- ▶ Now the client has come to the vendor with his requirements.
- ▶ Client said that I want to make a game on base of "**Norse Mythology**".
- ▶ And the game name should be "**GOD OF WAR**".
- ▶ So, Norse Mythology is the **primary requirement** of client.
- ▶ The vendor company said that we will make a game as on your requirement.
- ▶ In most of the 90% cases the client always be a non technical entity.



SDLC - ANALYSIS

- After the completion of requirement phase, an Analysis Phase comes next.
- As all you can see the in the background image that the vendor company has started the analysis of the game structure.
- The must analysis on what kind of characters does exist in Norse Mythology.
- Which includes it's main storyline, characters, weapons, etc.
- In this phase we must involve the client to explain the what vendor has an idea on his mind for the game.





SDLC – IMPLEMENTATION

- ▶ After the completion of Design phase, a very Logical part Implementation Phase comes next.
- ▶ As all you can see the in the background image that the vendor company developer doing a coding process.
- ▶ Usually Basic code is written in C# for smooth running for any game. There are many platforms available for it like, **Unity**, **Unreal Engine**, etc.
- ▶ This coding includes character movements according to pre-decided story line of game.
- ▶ In this phase developer do **unit test** for the code that they wrote in a system software.

SDLC - TESTING

- ▶ After the completion of logical phase, a quality part **TESTING** Phase occurs where we must test each and every scenario as aspects of the end user.
- ▶ As all you can see the in the background image that the we tester started testing of a developed game and found 2006 bugs in a game at different parts of it.
- ▶ Original size of the bugs was 3878 in a code.
- ▶ On the basis of that we can say that with the testing, we testers build the customer or client's "**TRUST**" on us.
- ▶ Because of this bugs the original release date was changed from **March 13th 2018** to **April 20th 2018**.

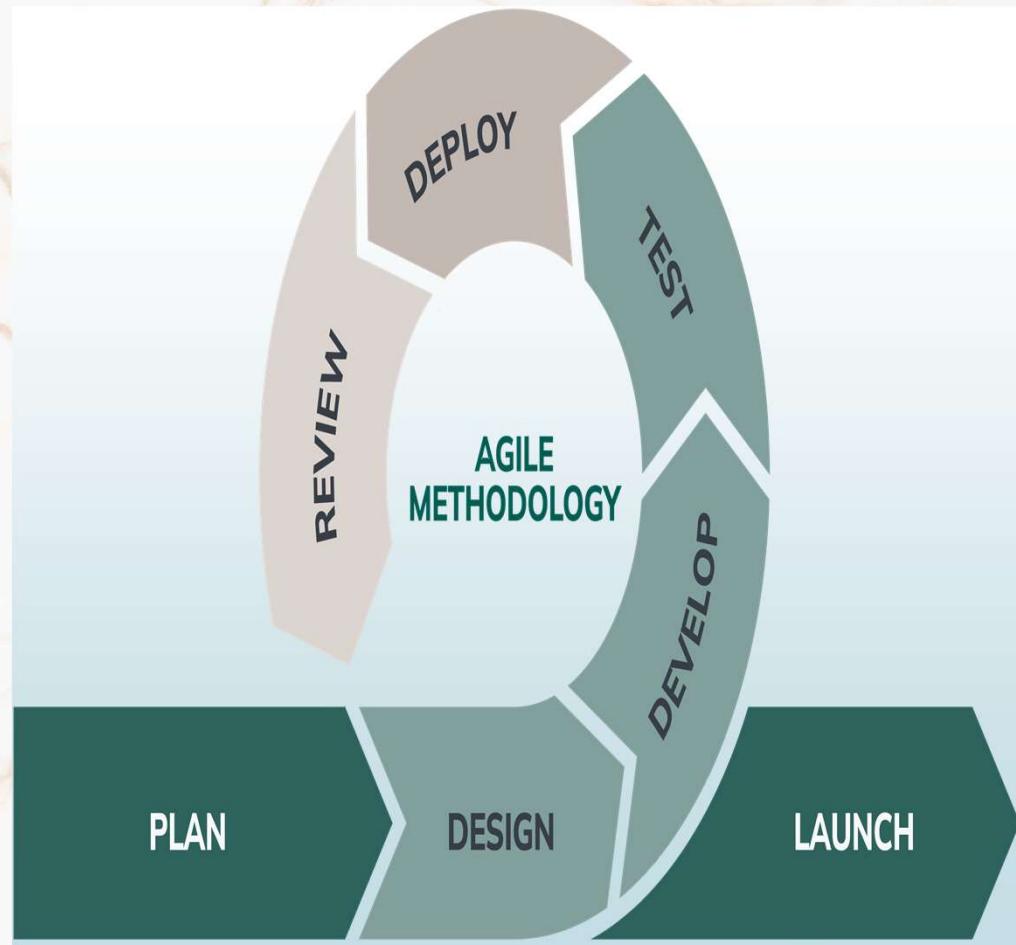
SDLC - MAINTENANCE



- After the completion of testing phase, a maintenance phase occurs where the developer do changes on a saying of a client.
- In background image you all can see the new game + difficulty option.
- It allows you play the new game again without an adventure that's equally focused on previous upgrade levels.
- This phase is called the maintenance of a game.

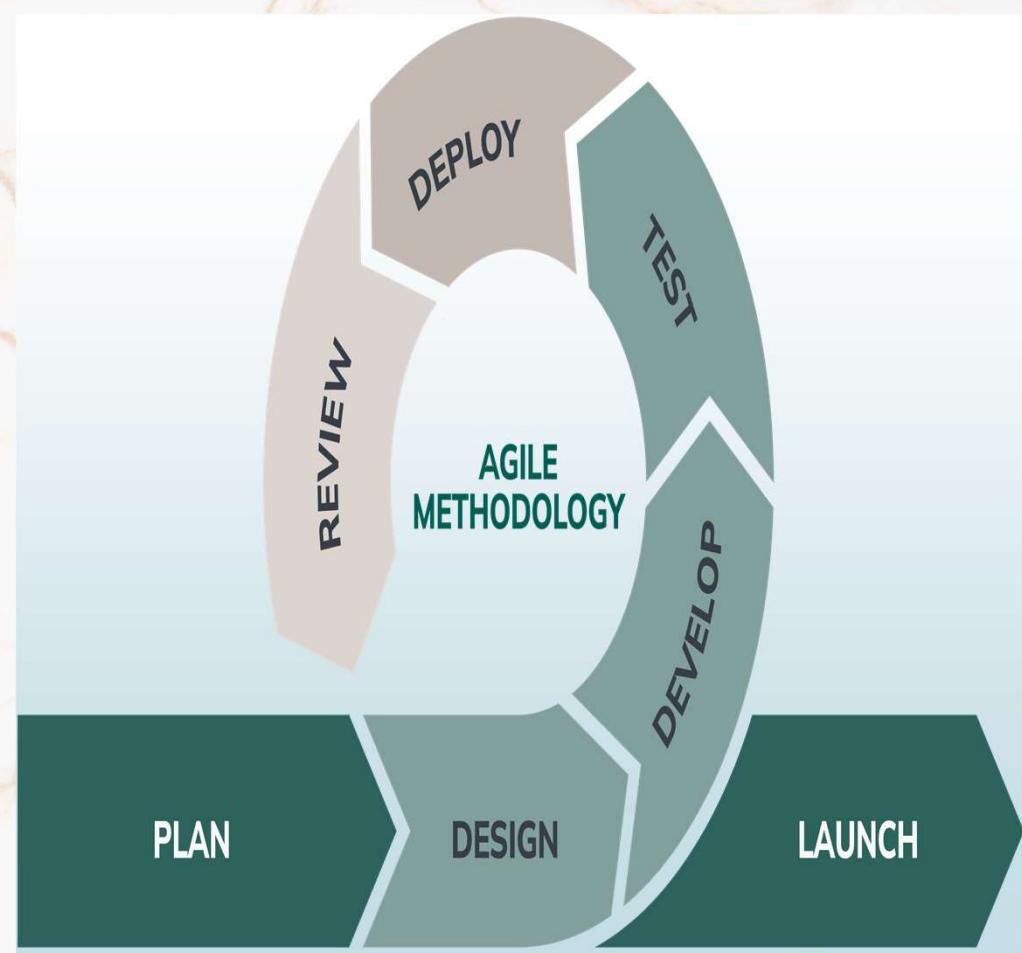
WHAT IS AGILE METHODOLOGY?

- Agile method is iterative & incremental which means break the project into smaller compartments.
- Each iteration lasts for 3 weeks & follows the steps of SDLC.
- This method we must stay in contact with the client for the review of finalization each iteration."
- This iteration also called as "**SPRINT**".
- Previous example of the game development is the best example of Agile Method.
- Agile stands for the continuous development with unique ideas like Japanese term "**KAIZEN**" method.



NOW AGILE METHODOLOGY WORKS.

- ▶ Let's take a previous game development example here.
- ▶ We plan the 2nd step of SDLC Analysis of the game.
- ▶ For that we have a 4 members to execute the details of whole Norse mythology.
- ▶ They read whole Norse mythology book & noted down extreme heavy points to narrate a game story.
- ▶ Story created, documentation of it submitted to the client, client suggest some corrections.
- ▶ Here we go again to improve it's best.
- ▶ Thus, how this cycle runs for only 3 weeks & after it the best pops out.



AGILE METHODOLOGY!!!

PROS.

- Software is released in small parts, making improvements quicker.
- Easy to adjust plans and features as needed.
- Regular feedback ensures the product meets user needs.
- Encourages communication between developers, testers, and client.
- Bugs are found and fixed early through regular testing.
- Continuous testing and improvements make the software more reliable.

CONS.

- Changing requirements make it hard to predict when the project will be finished.
- Frequent changes can make the project bigger than expected.
- Clients must be available for feedback, which isn't always possible.
- Teams must stay organized and follow Agile rules properly.
- Managing Agile in big, complex projects can be difficult.
- Focus on working software may lead to incomplete records.

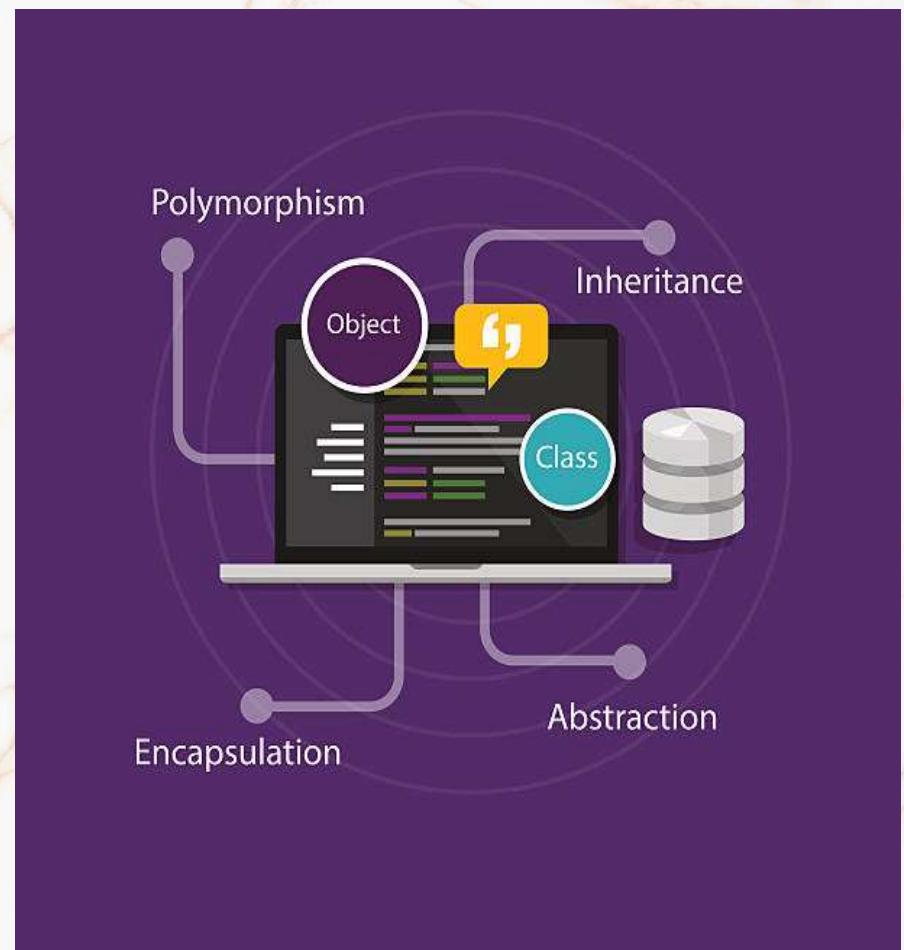
WHAT IS SRS?

- ▶ SRS stands for “**Software Requirement Specification**”.
- ▶ It is a document prepared by business analyst or system analyst.
- ▶ It describes what will be the features of software & what will be its behaviour.
- ▶ It is the detail description of software system to be developed with its “**Functional & Non-Functional Requirements**”.
- ▶ SRS document is actually agreement between client & provider.
- ▶ SRS document must be complete by taking all the requirement related to software development.
- ▶ The SRS document must be created in such a manner that it should be modifiable.



WHAT IS OOP?

- ▶ **Object Oriented Programming** (OOP) is a programming paradigm that organizes code using **objects** instead of just functions and logic. Objects are instances of **classes**, which act as blueprints defining their properties (data) and behaviors (methods).
- ▶ There are 6 Key Components of OOP.
 - ▶ Class
 - ▶ Object
 - ▶ Encapsulation
 - ▶ Inheritance
 - ▶ Polymorphism
 - ▶ Abstraction



OOP - CLASS, OBJECT, ENCAPSULATION & INHERITANCE

► CLASS:

- ▶ Let us understand this as a previous game example.
- ▶ The game they are developing should be in **Action** genre.
- ▶ In this sense the class of OOP is **Action**.

► OBJECT:

- ▶ Now develop a **characters** according to fit on action genre game.
- ▶ In this case the **character** that we develop is an **object**.

► ENCAPSULATION:

- ▶ Encapsulation **hides the internal details** of an object and only exposes necessary functionalities.
- ▶ Example: A Player can't manually set health bar to 100%.
- ▶ It only increases through health stones.

► INHERITANCE:

- ▶ Inheritance allows a **child class to inherit properties and methods** from a parent class.
- ▶ Example: **Weapon** as a parent class, **Axe** as a child class – ability to freeze enemy.

OOP – POLYMORPHISM & ABSTRACTION

► POLYMORPHISM:

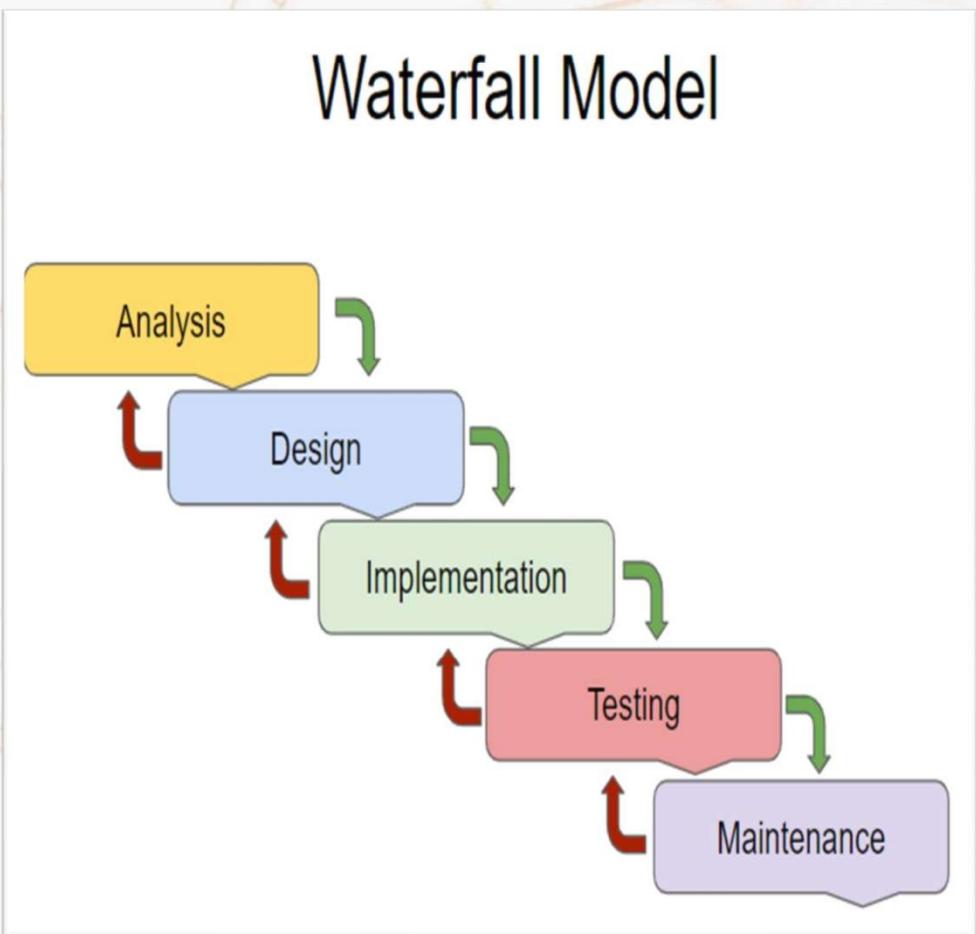
- ▶ Polymorphism allows different objects to use the **same method name but with different implementations.**
- ▶ Example: Character attacks changes base on the weapon. With Axe character can perform light & heavy attacks.

► ABSTRACTION:

- ▶ Abstraction hides unnecessary details and exposes only **essential functionalities.**
- ▶ **Example:** The game **hides** complex mechanics like physics, AI behavior, and collision detection.

WHAT IS WATERFALL MODEL?

- The Waterfall Model is a linear and sequential software development model where each phase must be completed before moving to the next. It follows a top-down approach, like a waterfall, where progress flows in one direction.
- Small to medium-sized projects with clear, well-defined requirements.
- There are mainly 5 Phases of waterfall model system.
 - Analysis
 - Design
 - Implementation
 - Testing
 - Maintenance



WATERFALL MODEL – ANALYSIS, DESIGN, IMPLEMENTATION, TESTING & MAINTENANCE

► ANALYSIS:

- The project team collects detailed requirements from client .
- The requirements are documented in a **SRS** form.

► DESIGN:

- The system architecture & design are created on the base of SRS.
- Includes UI/UX Design, database structure, & system architecture.

► IMPLEMENTATION:

- Developers write actual code based on design.
- Unit testing is done to check individual parts.

► TESTING:

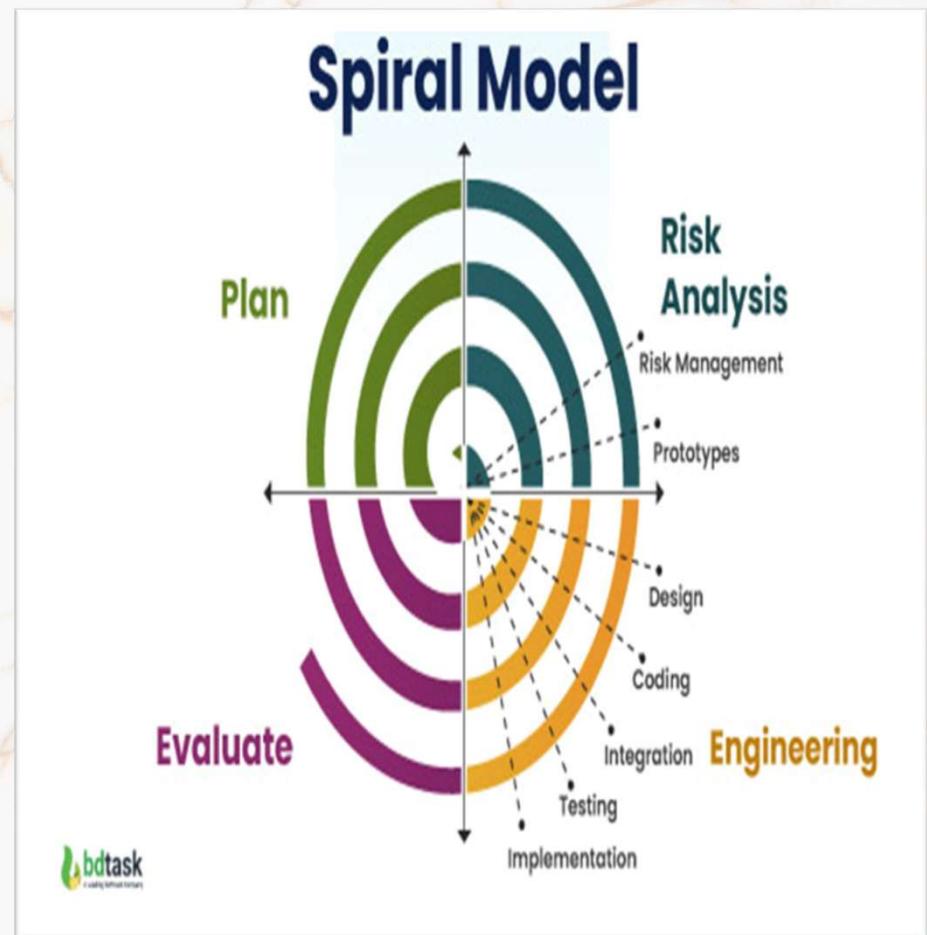
- The software tested for bugs, security flaws, & performance issues.
- Testers test software in every aspects of user's POV.

► MAINTENANCE:

- Enhancements or new features may be added based on feedback of users.

WHAT IS SPIRAL MODEL?

- ▶ The Spiral Model is a risk-driven model, meaning that the focus is on managing risk through multiple iterations of the software development process.
- ▶ The Spiral Model is often used for complex and large software development projects, as it allows for a more flexible and adaptable approach to software development.
- ▶ It is also well-suited to projects with significant uncertainty or high levels of risk.
- ▶ It consists of the following phases:
 - ▶ Planning
 - ▶ Risk Analysis & Prototyping
 - ▶ Development & Testing
 - ▶ Review & Development



SPIRAL MODEL – PLANNING, RISK ANALYSIS & PROTOTYPING, DEVELOPMENT & TESTING, REVIEW & DEPLOYMENT

► PLANNING:

- ▶ Gather requirements from clients or users.
- ▶ Identify goals, features, budget, and timelines.
- ▶ **Example:** In Android OS development, Google decides to improve battery life and security in the next update.

► RISK ANALYSIS & PROTOTYPING:

- ▶ Identify potential risks (security, performance, compatibility issues).
- ▶ Create a prototype (early version) to test solutions.
- ▶ **Example:** If a new AI feature drains too much battery, developers refine it before full implementation.

► DEVELOPMENT & TESTING:

- ▶ Develop the software based on approved prototypes.
- ▶ Conduct unit testing, integration testing & performance testing.
- ▶ **Example:** Android Beta is released to Pixel users for feedback on UI, speed & app compatibility.

► REVIEW & DEPLOYMENT:

- ▶ Gather feedback, fix bugs, & performance issues.
- ▶ Deploy the final version to the public.
- ▶ **Example:** Android releases Android 14 officially, then provides security patches & updates.

FLOW DIAGRAM GITHUB LINKS

- https://github.com/darshangohil7/Testing_Workspace/tree/main

RAISING KRATOS LINK - YOUTUBE

- https://www.youtube.com/watch?v=IJZWKBDXXFY&ab_channel=PlayStation
- Above Link is a Best example of SDLC & How industry making a software in aspects of client.
- Some of you may find this waste of time. But Please Do watch this.
- If you click if you want to go in testing part, just jump to **01:23:00**.