**Write a short summary for each methodologies.**

**Development methodologies:**

Development methodologies are like recipes or blueprints that teams follow to build something, be it a website, app, or even a new product. They provide a structured way to organize work, ensuring everything is done in an efficient and organized manner.

There are 4 types:

1. Domain Driven Design
2. Test Driven Design
3. Behavior Driven Design

**Domain-Driven Design (DDD):**

Domain Driven Design is a software development philosophy that emphasizes the importance of understanding and modeling the business domain. It is a strategy aimed at improving the quality of software by aligning it more closely with the business needs it serves.

Example:

1. If software processes loan applications, it might have classes like "loan application", "customers", and methods such as "accept offer" and "withdraw".
2. If we take the retail domain it is having the classes like “sales” ,”customers”,”product”, and “order”.

**Test Driven Design (TDD):**

Test-Driven Development (TDD) is a software development approach that enhances code quality and reliability by ensuring that tests are written before the actual code. This methodology fosters better software design and encourages developers to focus on the desired behavior of the system.

**Behavior Driven Design (BDD):**

Behavior-Driven Development (BDD) is a way of building software where the team defines how the software should behave, in a way that everyone (including non-technical people) can understand. It focuses on user interaction and using simple language to describe how the application should respond to different actions.

**Feature Driven Domain (FDD):**

Feature-driven design focuses on building software by breaking it down into smaller, manageable units called "features". Instead of starting with a large project, teams identify specific features they want to deliver, design those features, and then implement and test them before moving on to the next set. This approach helps teams deliver working software incrementally and efficiently.