Chapter: What is BDD? (1 Hour)

🧠 Objective

Understand the core principles of Behavior Driven Development (BDD), how it differs from Test Driven Development (TDD) and Acceptance Test Driven Development (ATDD), and why it's a powerful tool for improving collaboration, clarity, and client alignment in software projects.

📝 What is BDD?

BDD (Behavior Driven Development) is a collaborative approach to software development that bridges the communication gap between business and technical teams.

- It builds on top of **Test Driven Development (TDD)** and **Acceptance Test Driven Development (ATDD)**.
- It uses a simple **Given–When–Then** syntax to describe system behavior from a user's perspective.
- Feature descriptions are written in **Gherkin**, a human-readable DSL (domain-specific language).

> Think of BDD as writing down user expectations in a way that both humans and machines can understand and execute.

🔍 BDD vs TDD vs ATDD

Feature	TDD	ATDD	BDD
Focus	Developer's perspective	Acceptance criteria from stakeholders	Behavior of the system
Tests Written By	Developers	Developers + Testers + Customers	Developers + QA + Business Analysts
Test Language	Code	Natural language or code	Gherkin (natural language)
Collaboration Level	Low	Medium	High
Tool Examples	`testing`, `JUnit`, `Go test`	`FitNesse`, `Robot Framework`	`Cucumber`, `godog`, `SpecFlow`

🌍 Real-World Analogy

☑ TDD (Test Driven Development)

A **chef** writes a recipe *only after* cooking the dish and tweaking it until it's perfect.

- Focus is on implementation.

- Example in Go:

```
func TestAdd(t *testing.T) {
    result := Add(2, 3)
    if result != 5 {
        t.Errorf("Expected 5, got %d", result)
    }
}
```

☑ ATDD (Acceptance Test Driven Development)

A **restaurant manager** first agrees with the customer on what the meal should taste and look like.

- Involves QA and business upfront to define acceptable behavior.
- Tests are written in acceptance criteria format.

☑ BDD (Behavior Driven Development)

The **chef, waiter, and customer** all agree on the dish — how it should look, smell, and taste — *before* it's even prepared. They write it down as:

> Given I'm hungry, When I order a pizza, Then I should receive a hot Margherita within 30 minutes.

- Everyone is aligned before the development starts.
- Focuses on **communication, clarity, and collaboration**.

🤖 Simple Hands-On Example

Use Case: Login Feature

📝 Feature: User Login

```
Feature: User Login

  Scenario: Successful login with valid credentials
    Given a registered user with username "rahul" and password "12345"
    When the user logs in with username "rahul" and password "12345"
    Then the user should see the dashboard
```

🥷 Corresponding Step Definitions (Go using godog)

```
func (s *loginSuite) iHaveARegisteredUser(username, password string) error {
    s.users[username] = password
    return nil
}
```

```go
func (s *loginSuite) iLoginWithCredentials(username, password string) error {
    s.result = (s.users[username] == password)
    return nil
}

func (s *loginSuite) iShouldSeeTheDashboard() error {
    if !s.result {
        return errors.New("login failed")
    }
    return nil
}
```

🎯 Benefits of BDD

🤝 1. **Improved Collaboration**

- Developers, QA, and business teams speak a common language (Gherkin).
- Everyone is involved from the beginning.

📝 2. **Better Clarity of Requirements**

- Requirements are written as executable examples.
- Reduces ambiguity and missing edge cases.

👩‍💼 3. **Client Alignment**

- Clients and stakeholders can *read and understand* the tests.
- Faster feedback loops lead to fewer reworks and better ROI.

📺 Recommended YouTube Videos

1. **BDD Explained in 5 Minutes (Simplilearn)**
   https://youtu.be/49gSJavp0H0

2. **Cucumber and BDD Overview (Ministry of Testing)**
   https://youtu.be/4GosOV2vqck

3. **Test Driven vs Behavior Driven vs Acceptance Test Driven**
   https://youtu.be/WyT3zJ8pU3U

❓ Interview Questions

Question	Suggested Answer Hint
What is BDD and how does it differ from TDD and ATDD?	Talk about collaboration, feature files, Given/When/Then

Question	Suggested Answer Hint
Why is Gherkin used in BDD?	Human-readable DSL, bridges business and tech
How does BDD improve software quality?	Clear requirements, early validation, automated tests
What are the advantages of BDD in agile teams?	Better communication, shared understanding
Give a real-world analogy of BDD.	Chef, customer, waiter scenario
What tools are used in Go for BDD?	godog, gherkin-lint
What is a feature file and what does it contain?	Feature, Scenario, Given/When/Then steps
What role does a QA play in a BDD project?	Writing scenarios, collaborating on step definitions

Summary

- BDD promotes *conversation over documentation*.
- It helps align technical output with business expectations.
- In Go, the godog framework is a robust choice for BDD-based automation.
- Gherkin syntax makes your scenarios readable, reusable, and testable.

> "BDD is not just about writing tests. It's about discovering what to build — together."