Day 17 Spring Profiles, (Setting Active Profiles) Environments @Value Spring Init Starters.

Day 16 Revisit –

We Created a Spring Boot based Monolith application.

Airline Reservation System (ARS)

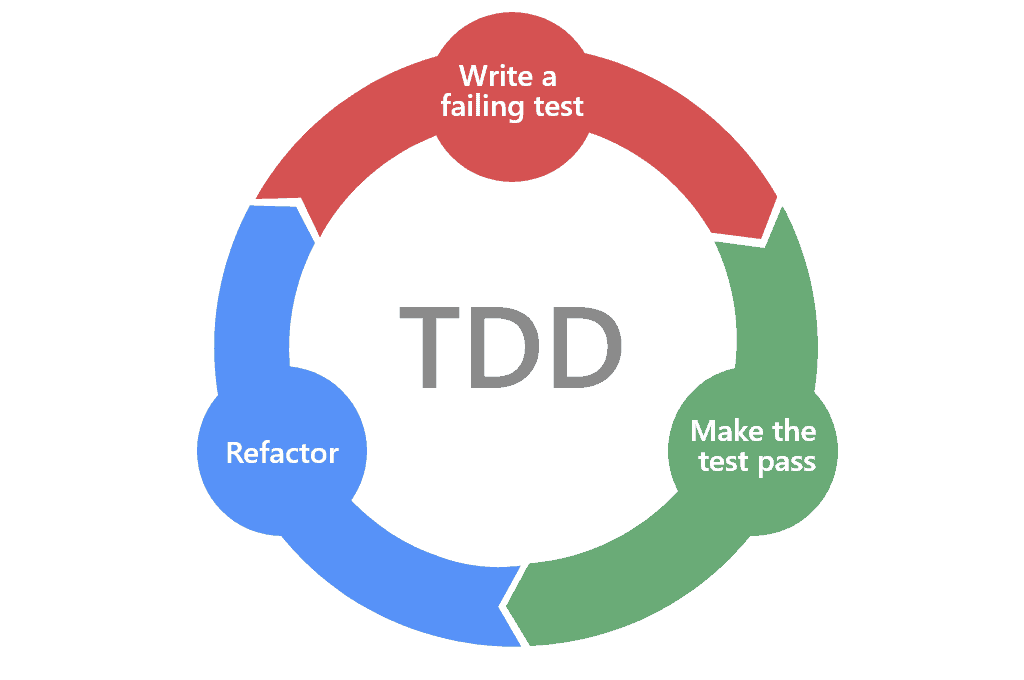
Entities.

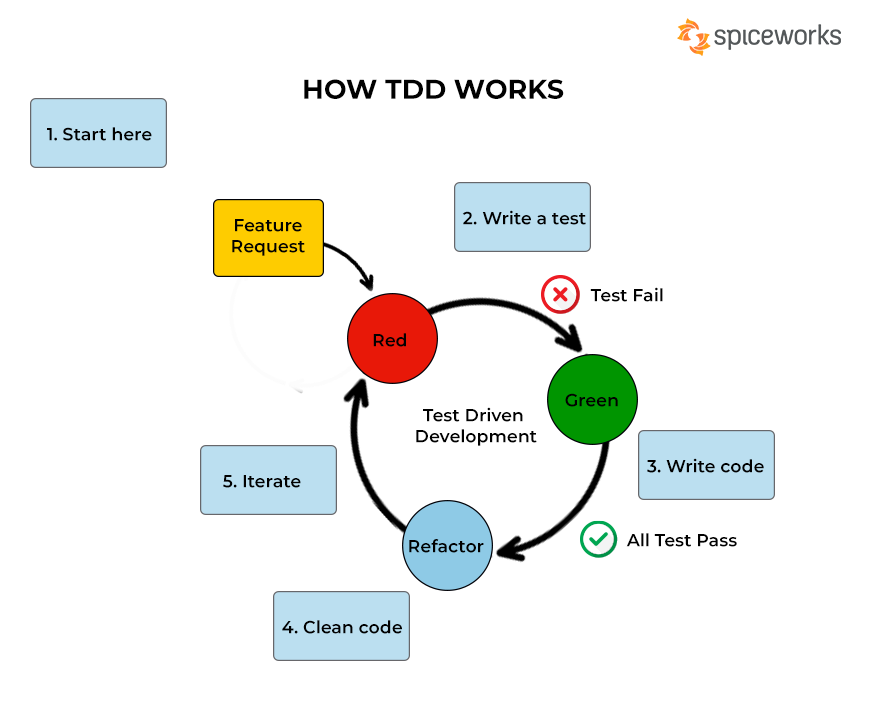
Enterprise Level Projects

1. TDD – Test Driven Development. (Write the failing test case first, before writing actual code, refactor code. Repeat. )
2. BDD – Behavior Driven Development. [ Selenium with Cucumber, Gherkin Language (Given, When, Then)]
3. DDD – Domain Driven Development.

TDD – Test Driven Development

Test-Driven Development (TDD) is a software development approach where you write automated tests before writing the actual code, using a "Red-Green-Refactor" cycle. First, you write a failing test (Red), then you write the minimum amount of code to make that test pass (Green), and finally, you clean up and improve your code while ensuring the tests still pass (Refactor). This iterative process leads to higher code quality, better design, fewer bugs, and more maintainable and testable software





BDD – Behavior Driven Development

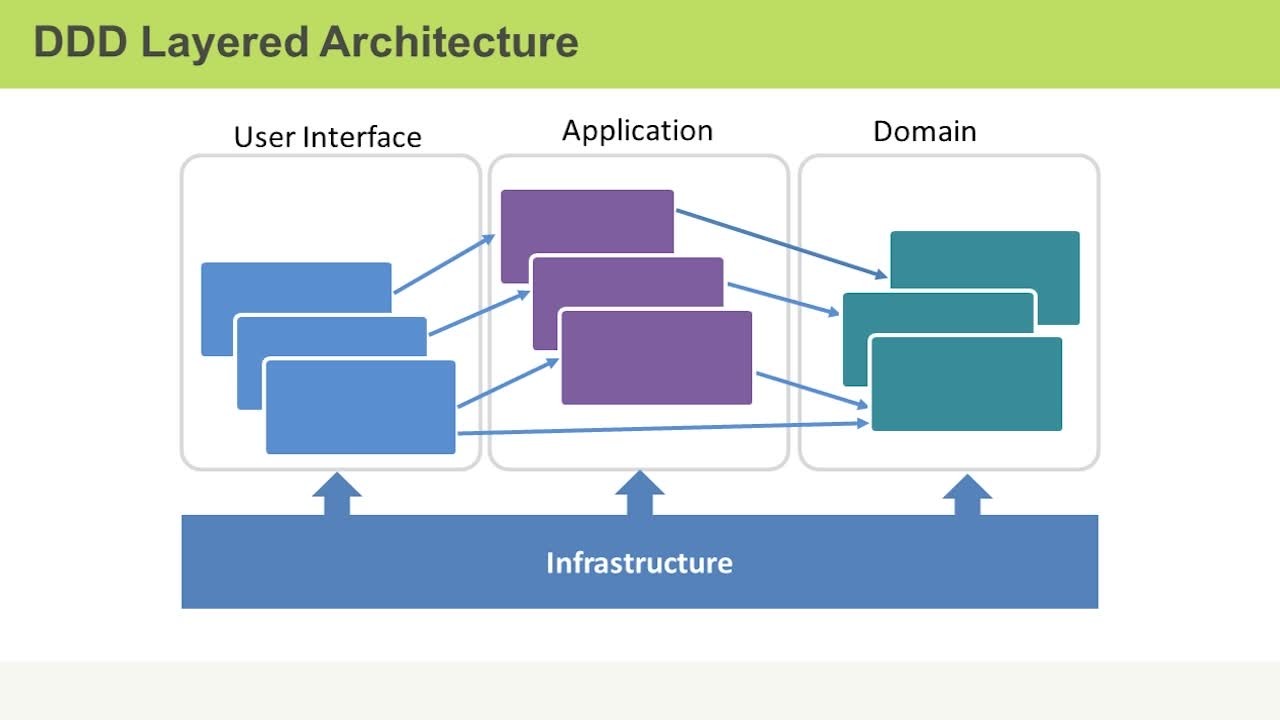
Behavior-Driven Development (BDD) is a software development methodology that uses a collaborative approach and natural language to define system behavior, ensuring that software meets user and business needs through shared examples and automated tests. Key concepts include cross-functional team collaboration, shared understanding through user-centric examples, the Gherkin language with the Given-When-Then format for scenario descriptions, and automated tests that are derived from these natural language specifications.

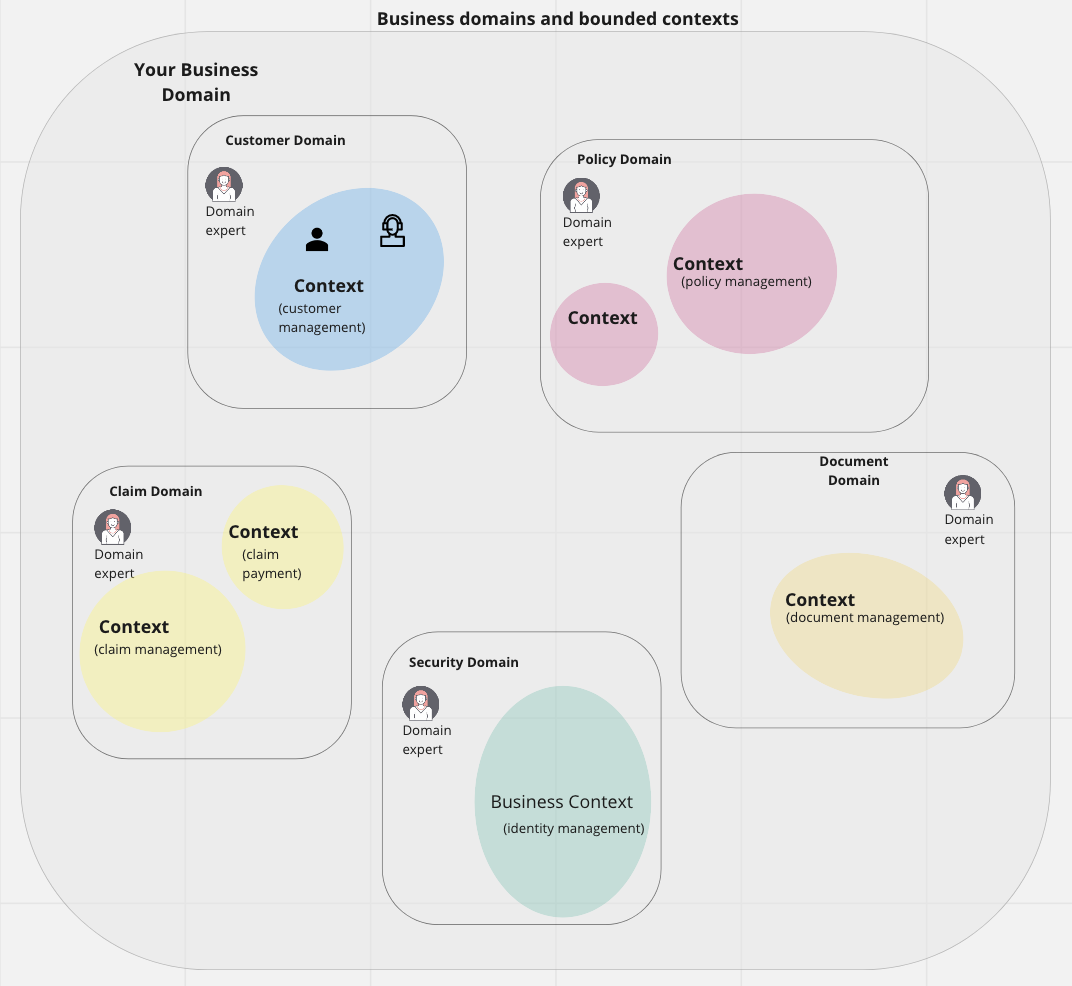
1. User Stories or Epic 🡪 Behavior (Functional Requirement)



DDD – Domain Driven Development

DDD concepts focus on aligning software design with a complex business domain through strategic and tactical approaches. Key concepts include a ubiquitous language shared by developers and domain experts, [bounded contexts](https://www.google.com/search?sca_esv=a6493d853efe4d30&biw=1536&bih=695&sxsrf=AE3TifP35kWeOOoRniY9fcO3rhHz94_rEA%3A1758276019475&q=bounded+contexts&sa=X&ved=2ahUKEwiPibvMyOSPAxUNRmwGHbBfGMAQxccNegQIIhAB&mstk=AUtExfCVfAl6PAAXPT3nmNZ80A4QMuokKH0RsFXDLWUivQzUGM4Q-bNifl7iuOtMS2JhakdydQde2KffNrdIzom-bhr7hhpkh6wGydHfvzbPZUXPfJAiEVsHiWoLtdrVt997m4LR5KXJ6FsuinjT50f3QGFKgemYywsMgo2Lj-9-0AH-JleANN0u33iV0-HMCQOKmWjKsSwBqS_GwhwcVDwULaxQb5ioPHDa3tZ3FhBwEXIs-Y_NDnP9mh73X7cSt5694uEMt8Fjq4YwwetUqEcU_Y8AkS8ho1OsHxeI2-WFSHvCWQ&csui=3) to divide the system, entities and value objects to model data, [aggregates](https://www.google.com/search?sca_esv=a6493d853efe4d30&biw=1536&bih=695&sxsrf=AE3TifP35kWeOOoRniY9fcO3rhHz94_rEA%3A1758276019475&q=aggregates&sa=X&ved=2ahUKEwiPibvMyOSPAxUNRmwGHbBfGMAQxccNegQIIhAC&mstk=AUtExfCVfAl6PAAXPT3nmNZ80A4QMuokKH0RsFXDLWUivQzUGM4Q-bNifl7iuOtMS2JhakdydQde2KffNrdIzom-bhr7hhpkh6wGydHfvzbPZUXPfJAiEVsHiWoLtdrVt997m4LR5KXJ6FsuinjT50f3QGFKgemYywsMgo2Lj-9-0AH-JleANN0u33iV0-HMCQOKmWjKsSwBqS_GwhwcVDwULaxQb5ioPHDa3tZ3FhBwEXIs-Y_NDnP9mh73X7cSt5694uEMt8Fjq4YwwetUqEcU_Y8AkS8ho1OsHxeI2-WFSHvCWQ&csui=3) to group related objects, and domain events to capture changes. The ultimate goal is to create software that accurately reflects and serves the business, improving communication and maintainability.





SOA – Service Oriented Architecture (Web Service – REST based)

a software design approach using reusable, interoperable services to build complex applications.

Users

Passengers

Reservations

Ratings

Reviews

Offers

Payments

Aircrafts

Roles

User\_roles

1. Monolith (Single Project for the entire application)
2. Microservice (One Spring Boot Project for each service)

Version 1 --- CLI based ARS (Core Java and JDBC concepts only – Exceptions, Threads)

Version 2 – Monolith Application (Spring Boot with MySQL/Postgres/MS-SQL)

Version 3 – Microservice based Application (Refactoring existing monolith app)

Version 2 – ARS (Monolith Spring Boot Based Application)

1. User (Entity)
2. Repository (DAO – Data Access Object – Any DB related task will be done here only)
3. Service – (Business Logic )
4. Controller – (It’s Controls data flow between layers)

Testing End points

1. Using Swagger (API Documentation) – Simplified way of Testing all the endpoints in one place.
2. Using External Tools (Postman, SoapUI, Bruno, ThunderClient etc.,)
3. Using CLI command (cURL) – Learning the syntax is challenging.

Environments.

1. Dev (Development)
2. Test (Testing)
3. UAT env, (User Acceptance Testing env)
4. Pre-prod (Staging) [similar to actual production env with dummy data]
5. Prod (Actual live env) Highly available, load balanced env with actual/secured data.

Spring Active profile.

spring.profiles.active=dev

H2 - Dev, MySQL - Test, Postgres - Staging , MS-SQL (Enterprise) – Prod

application.properties

spring.application.name=ars

spring.profiles.active=dev

server.port = 8085

application-dev.properties

# h2 Datasource Properties

spring.datasource.url=jdbc:h2:mem:arsdb

spring.datasource.driverClassName=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=

spring.h2.console.enabled=true

spring.h2.console.path=/h2-ui

spring.h2.console.settings.trace=false

spring.h2.console.settings.web-allow-others=false

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.format\_sql=true

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.H2Dialect

spring.jpa.hibernate.ddl-auto=update

application-test.properties

# MySQL Datasource Properties

spring.datasource.url=jdbc:mysql://localhost:3306/arsdb?createDatabaseIfNotExist=true&useUnicode=true&characterEncoding=utf-8&autoReconnect=true

spring.datasource.username=root

spring.datasource.password=root

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

application-staging.properties

# Postgres Datasource Properties

spring.datasource.url=jdbc:postgresql://localhost:5432/arsdb

spring.datasource.username=postgres

spring.datasource.password=postgres

application-prod.properties

# MSSQL DataSource Configuration

spring.datasource.url=jdbc:sqlserver://<hostname>:<port>;databaseName=<database\_name>;encrypt=false;trustServerCertificate=true

spring.datasource.username=<your\_username>

spring.datasource.password=<your\_password>

spring.datasource.driverClassName=com.microsoft.sqlserver.jdbc.SQLServerDriver

context-path

active profile

Multiple environments

OAS – Open API Specification