

Behaviour Driven Development

BDD är en mjukvaruutvecklingsmetodik som fokuserar på att definiera och testa programvarans beteende genom samarbete mellan utvecklare, testare och icke-tekniska intressenter.

Syfte: Säkerställa att programvaran uppfyller affärskraven och är i linje med kundens behov genom att beskriva förväntade utfall på ett enkelt och begripligt språk.

Kärnkoncept för BDD

- **Gemensam Förståelse:** BDD främjar samarbete mellan affärs- och tekniska team för att säkerställa att alla har en gemensam förståelse av kraven.
- **Specifikation med Exempel:** Krav skrivs som exempel på hur programvaran ska bete sig i olika scenarier.
- **Körbara Specifikationer:** Dessa exempel omvandlas till automatiserade tester som fungerar som levande dokumentation.

De Tre Huvudkomponenterna i BDD

1. User Stories
2. Scenarios
3. Given-When-Then Syntax

User Stories

Beskriver vad användaren behöver och vill uppnå. De skrivs i ett enkelt format och fångar affärsvärdet.

Exempel:

"Som kund vill jag kunna logga in på mitt konto för att se min kontoinformation."

Scenarios

Ger exempel på hur användarberättelserna ska fungera i praktiken. Varje scenario beskriver ett specifikt fall och ger tydlig förväntad funktionalitet.

Exempel:

1. Scenario: *Lyckad inloggning*
2. Scenario: *Misslyckad inloggning (felaktigt lösenord)*

Given-When-Then Syntax

BDD-scenarier skrivs ofta i ett strukturerat format som kallas **Given-When-Then**, vilket gör kraven tydliga och läsbara.

1. **Given** – Förutsättning eller kontext.
2. **When** – Handlingen eller händelsen.
3. **Then** – Förväntat utfall.

Exempel:

1. **Given** att en användare är på inloggningssidan
2. **When** användaren anger sitt användarnamn och lösenord
3. **Then** ska användaren dirigeras till sin dashboard

Fördelar med BDD

- **Förbättrad Kommunikation:** BDD gör det lättare för alla teammedlemmar att förstå kraven.
- **Minskat Behov av Omskrivning:** Tydliga specifikationer minskar risken för missförstånd och fel.
- **Körbara Specifikationer:** Levande dokumentation som fungerar som automatiserade tester.

Sammanfattning

Behavior-Driven Development handlar om att skapa tydliga, läsbara specifikationer som beskriver programvarans beteende med exempel. Genom att använda ett språk som alla kan förstå skapar BDD en gemensam förståelse, minskar risken för fel och säkerställer att utvecklingen följer affärskraven.

Verktyg för BDD

SpecFlow – Ett verktyg för .NET-applikationer som möjliggör BDD i Visual Studio.