

Test Driven Development

TDD är en mjukvaruutvecklingsmetodik där tester skrivs innan själva koden implementeras. TDD hjälper utvecklare att fokusera på att skapa funktionalitet som uppfyller specifika krav, och resultatet är renare, mer underhållbar och mer pålitlig kod.

Syfte: Att säkerställa att all kod som skrivs är testbar, fungerar enligt kravspecifikationen och är enkel att vidareutveckla.

Kärnkoncept för TDD

- **Test First:** I TDD skapas ett test som definierar den önskade funktionaliteten innan någon produktionskod skrivs.
- **Refactoring:** Koden förbättras och förenklas kontinuerligt för att göra den mer läsbar och effektiv, utan att förändra dess beteende.
- **Continuous Feedback:** Genom att köra tester ofta får utvecklaren snabb feedback om eventuella fel eller förändringar i beteende.

De Tre Huvudstegen i TDD

1. Red
2. Green
3. Refactor

Red

Skriv ett Misslyckat Test

Det första steget i TDD är att skriva ett test för den funktionalitet som ska implementeras. Eftersom koden ännu inte finns, kommer testet att misslyckas.

Exempel:

Skriv ett test som verifierar att en ny användare kan registreras i systemet.

Resultat:

Testet misslyckas (röd status) eftersom ingen kod finns ännu.

Green

Skriv Minsta Mängd Kod för att Passera Testet

Det här steget skriver utvecklaren den minsta möjliga koden för att få testet att gå igenom (grön status). Målet är att fokusera på att uppfylla testets krav, inte på perfektion.

Exempel:

Implementera en metod som skapar en användare, men gör det så enkelt som möjligt.

Resultat:

Testet lyckas (grön status).

Refactor

Förbättra Koden

När testet går igenom är nästa steg att refaktorera koden. Målet är att göra koden tydligare, effektivare och mer läsbar, utan att förändra dess beteende.

Exempel:

Förenkla metoden, lägga till variabelnamn som är mer beskrivande, eller bryta ut logik till mindre funktioner.

Resultat:

Koden är nu bättre strukturerad och lättare att underhålla, och testet går fortfarande igenom.

Exempel på TDD-cykel

1. **Red** – Skriv ett test som testar att ett nytt konto kan skapas.
2. **Green** – Implementera en enkel version av kontoskapningsfunktionen som får testet att passera.
3. **Refactor** – Förbättra kontoskapningsfunktionen genom att lägga till verifieringar och mer tydlig kodstruktur.

Fördelar med TDD

- **Bättre Kodkvalitet:** TDD säkerställer att koden är testbar och fungerar enligt kravspecifikationen.
- **Färre Buggar:** Eftersom tester körs kontinuerligt upptäcks buggar tidigt i utvecklingsprocessen.
- **Enklare Underhåll:** TDD leder till mer modulär och lättläst kod, vilket gör framtida ändringar enklare att genomföra.

Sammanfattning

Test-Driven Development är en metodik som fokuserar på att skriva tester först, implementera minsta möjliga funktionalitet för att passera tester, och sedan refaktorera koden. Genom att följa denna process säkerställer TDD att koden är robust, testbar och uppfyller kraven.

Verktyg för TDD

- **xUnit**: Ett ramverk för enhetstester i .NET.