

Enhetstestning

Enhetstestning är en testteknik där individuella enheter/funktioner testas isolerat. Syftet med testerna är att validera att varje enhet fungerar som förväntat samt att de funktionella kraven uppfylls. Detta bidrar även till ökad pålitlighet och kvaliteten i mjukvaran.

De fungerar även som ett slags skyddsnät för utvecklaren då hen kan refaktorera och göra förändringar med vetskapen om att det inte introducerar nya buggar eller förstör existerande funktionalitet.

Enhetstesternas roll i utvecklingscykeln är att hitta buggar tidigt, innan de hinner sprida sig till mer komplexa delar av systemet. Och eftersom utvecklaren kan se exakt var buggen har uppstått går den oftast väldigt snabbt att åtgärda.

I testkedjan spelar enhetstester en mycket stor roll eftersom de utgör grunden för alla andra tester. Väl alla individuella enheter har blivit grundligt testade kan de integreras och testas tillsammans som en modul, och efter det kan systemtester köras för att validera hela produkten.

Automatiserade enhetstester

Automatisering av enhetstester kan ge flera fördelar:

- De tvingar utvecklaren att skriva testbar kod, vilket bidrar till ökad kodkvalitet och kod som är lättare att underhålla
- Testerna går (oftast) snabbt att köra och är enkla att repetera, vilket gör att utvecklaren kan köra testerna ofta och därmed hitta buggar tidigt
- De körs på samma sätt varje gång

Regressionstestning

Regressionstestning görs för att säkerställa att förändringar inte har introducerat nya buggar eller gjort sönder existerande funktionalitet. Testningen utförs genom att man kör igenom alla testerna igen och verifierar att allt fortfarande fungerar som förväntat.

Enhetstester är grunden till en fungerande regressionstestning då de säkerställer de individuella enheternas funktionalitet. Och om enhetstesterna är automatiserade kan de enkelt köras som en del i utvecklingsprocessen.

Testdriven utveckling (TDD)

TDD är en iterativ utvecklingsmetod där man skriver testerna innan den faktiska koden:

- Skriv ett test som inte blir godkänt. Testet fungerar som en specifikation för implementationen som skrivs senare
- Implementera den minsta mängd kod som krävs för att testet ska bli godkänt
- Refaktorera koden för att förbättra och göra den lättare att underhålla
- Repetera

TDD säkerställer att implementationen designas för testbarhet samt att den lever upp till de funktionella kraven. Genom att tillämpa TDD-metoden kan buggar hittas tidigt och man säkerställer att koden är vältestad innan den släpps.