

Deel 4: Test frameworks

In de [vorige blog](#) hebben we gekeken naar de 100% code coverage illusie en zijn we dieper ingegaan op de aanpak van testautomatisering voor de verschillende lagen van de testpiramide.

In deze blog behandelen we de opbouw van veelvoorkomende test frameworks.

Test frameworks

Test frameworks hebben tot doel het testen te vergemakkelijken. Ze regelen de randvoorwaarden en integreren met de ondersteunende tools waarmee tests eenvoudig kunnen worden opgesteld en uitgevoerd zodat de gebruiker van het framework alleen nog maar over de te implementeren tests hoeft na te denken. Testframeworks kom je tegen op elk testniveau, er zijn unittest frameworks, API-test frameworks en E2E-test frameworks.

Vaak vind je in deze frameworks drie niveaus van onder naar boven bestaande uit een adaptatieniveau een stapdefinitieniveau en een testinterpretatieniveau. Elke laag biedt een verdere vorm van abstractie van de code. Onderstaande figuur laat de drie lagen zien met de veel voorkomende frameworks per laag.



Adaptatieniveau

Het onderste niveau, het adaptatieniveau, is de laag die in geval van het API framework de

API-calls doet en de responses ontvangt. Bij E2E frameworks bevat het adaptatieniveau het page-object-model van de te testen pagina. Het niveau bevat ook de elementaire handelingen die nodig zijn om de tests uit te voeren. In geval van een E2E testframework vind je er handelingen als: klik-op-een-knop, lees-een-schermelement-uit, scroll-door-een-lijst etc.

Stapdefinitieniveau

Boven op het adaptatieniveau vind je in een goed framework een laag met stapdefinities (wordt ook wel glue-code genoemd). Deze laag vertaalt de zinnen uit de (Gherkin) feature file naar functionele, gebruikers gefocuste zaken. Deze geeft hij vervolgens door aan het adaptatieniveau. Voorbeelden van stapdefinities zijn: Selecteer product in de webshop, Ga naar het winkelmandje, Check out, Voer de betaling uit etc.

Testinterpretatieniveau

Het testinterpretatieniveau bevat in geval van een BDD testframework de Gherkin syntax zodat elke test die geschreven is volgens deze syntax uitgevoerd kan worden door het framework.

Vooruitblik volgende blog

Tot zover deze blog. In de volgende blog staan we stil bij de voordelen van goede invoering en de risico's bij foute invoering van testautomatisering.

STAY TUNED!

Wim ten Tusscher