

De hedendaagse wereld is snel en competitief. Om als bedrijf succesvol te zijn heb je snel veranderende producten van hoge kwaliteit nodig. Dit legt de nadruk op aanpasbaarheid en testbaarheid van je product.

Testbaarheid verhoogt aanpasbaarheid, want als je je aanpassingen goed kunt testen, weet je dat de aanpassing goed is en het niet-aangepaste nog steeds goed werkt. *Snel* kunnen testen vraagt om testautomatisering. Echter, als testautomatiseringcode slecht ontwikkeld is, dan bereik je niet wat je wilt. De volgende drie aspecten zijn belangrijk voor succesvolle testautomatisering:

1. Doelgericht NIET Toolgericht

Denk na over productrisico's en met welke tests je deze risico's kunt afdekken. Denk na over welke doelen je wel en niet met automatische tests kunt en wilt bereiken en kies daarna pas voor de meest geschikte tool. Niet alles is te automatiseren. Zaken als usability (look and feel) zijn heel lastig in automatisering te vangen. Daarnaast moet het product ook nog goed exploratory aan de tand gevoeld worden, deze tests zijn per definitie niet te automatiseren. De tester is dus nog steeds in command.

2. Korte terugkoppelingen

Automatiseer niet je handmatige tests, deze zijn vaak end2end, starten vaak op de GUI en zijn daarmee complex en traag in de terugkoppeling of iets goed of fout is. Ook is het lastig om een falende end2end test te debuggen, de fout kan namelijk overal in de keten ontstaan. Testautomatisering faciliteert testen onder de motorkap. Het stelt je in staat de keten op te breken in kleinere stukken en deze zelfstandig te testen. De keten is namelijk zo sterk als de zwakste schakel. Begin met veel coverage op kleine deelfunctie. Dit zijn kleine, snelle en goedkope tests die bij elke codewijziging kunnen worden uitgevoerd. Ze hebben dus een erg kort terugkoppelus. Vul de testdekking aan met een set goed gekozen tests van geïntegreerde features, APIs en/of services. Sluit af met slechts enkele E2E (GUI)tests om de business processen te dekken. De feature tests en zeker de E2E tests zijn complexer, de terugkoppelingen zijn dus groter. Ook zijn ze trager waardoor de uitvoerfrequentie vaak lager is (bv. elke nacht op de nightly build).

3. Onderhoudbaarheid en schaalbaarheid

Geautomatiseerde tests zijn een software product met een gewenste levensduur die net zo lang is als je product. Onderhoudbaarheid en schaalbaarheid zijn dus belangrijk om op lange termijn succesvol te zijn, je wilt niet dat het onderhoud van je testautomatisering een rem wordt op de snelheid van productaanpassingen, dan zet je de wereld op de kop! Denk dus continue na over de architectuur van je testautomatisering, bouw abstractielagen in en

refactor vaak. Maak de testautomatiseringscode onderdeel van je code review en claim tijd voor alle activiteiten in je planning. Testautomatisering is een teamactiviteit, niet het domein van alleen de tester.

Volg je deze drie principes dan bouw je als team een testpiramide van solide bouwblokken. Doe je het fout, dan produceer je een test-ijshoortje. En we weten allemaal welke van deze twee het langst stand houdt in de tijd!

Wim ten Tusscher