

Sinds hun start in 2000 worden bij [Enrise](#) bedrijfsprocessen van klanten geautomatiseerd tot onder andere webapplicaties, sites en api's. Maar hoe zit het met hun eigen proces? Software-ontwikkeling is een ambacht en blijft mensenwerk, maar je kunt er veel in automatiseren. Dat Enrise daar heel ver in gaat, hebben ze laten zien tijdens de vierde editie van CodeCuisine® Live, het kennisevenement waar tech en business elkaar ontmoeten.

Goede software-ontwikkeling kan niet zonder een degelijk testproces. Enrise werkt daarom samen met softwaretesters van Polteq en vroegen ons als gastspreker meer te vertellen over hoe je een goed geautomatiseerd testproces opzet. Wim ten Tusscher gaf de presentatie: "Testautomatisering van Flop naar Top".

Daarnaast vertelden de experts van Enrise hoe je je een *ontwikkel-test-acceptatie-productie*-proces supersnel en flexibel kunt maken, zodat je op feature-niveau vele malen sneller live kan met vernieuwingen aan je online product.



Testautomatisering van Flop naar Top

Gedurende een ontwikkeltraject van de software die Enrise bouwt, zijn de testers onderdeel van hun ontwikkelteams. Zij controleren, onderzoeken en beproeven de functionaliteit en de processen die we maken. De testers krijgen dus in feite de taak om de software kapot te krijgen, zodat al in een vroeg stadium naar voren komt waar het mis kan gaan.

Flop-testautomatisering

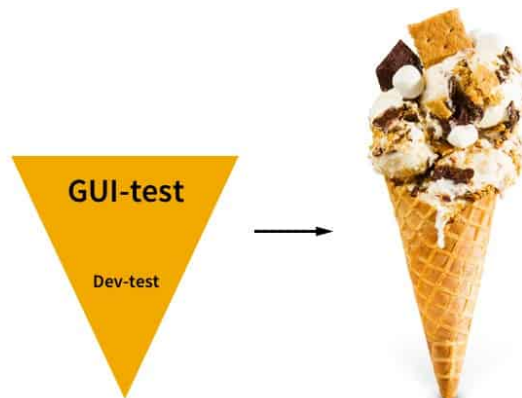
Sprekers Frans van As en Wim ten Tusscher, testspecialisten bij Polteq, namen de 52 bezoekers mee naar wat je wel en niet moet doen als het gaat om het opzetten van een testproces. Zij pleiten voor vaker, beter en eerder testen. Zo voorkom je dat je naar een

speld in een hooiberg zoekt als fouten na livegang aan het licht komen.

Je geautomatiseerde testproces flopt als je:

- Testen ziet als apart technologieproject met lage prioriteit;
- Geen strategie en deadlines hebt als het gaat om testen en uitkomsten;
- Bestaande handmatige tests automatiseert;
- Alleen tools die checken leidend laat zijn.

Te vaak is het testen van software een sluitpost van een project. En wordt er getest, dan zijn het vooral de visuele checks (GUI-tests) die worden gedaan: Zit alles erin wat we hadden afgesproken, werkt het? Dan is het goed. Development-tests worden minder vaak uitgevoerd, maar zouden eigenlijk belangrijker moeten zijn dan de GUI-tests. Want door op codeniveau te testen wat er niet zichtbaar is, kom je erachter wat niet werkt of wat niet werkt in hele specifieke situaties die je vooraf niet kon weten.



De ijshoorn

Deze manier van testen (Meer op zichtbaarheid en minder op code/onzichtbaarheid) is als een omgekeerde driehoek, met veel aandacht bovenin en weinig aandacht onderin. Daarmee laat het zich nog het best vergelijken met een ijshoortje. Het ziet er prima uit en iedereen kan er mee uit de voeten, maar een ijsje houdt nooit lang stand: vroeg of laat is het ijs gesmolten of het hoortje gebroken.

Top-testautomatisering

Zeker in complexe developmentprojecten moet je testen belangrijk maken. Enrise noemt dat een [testgedreven](#) ontwikkelproces. Daarbij werk je gezamenlijk aan nieuwe code en zodra het eerste brokje functionaliteit klaar is, gaat een tester ermee aan de slag. 'Breekt' er iets, dan kun je snel schakelen door dat brokje functionaliteit beter te maken.

Je geautomatiseerde testproces is top als je:

- Voor alles wat je van tevoren weet checks kunt opzetten en automatiseren;

- Alles wat je vooraf niet kunt weten laat testen door mensen;
- Als je alles wat je vooraf niet kunt weten laat onderzoeken door een goede tester.


Een tester legt vooral geen nadruk op die GUI-test. Dat zijn namelijk controles die geautomatiseerd kunnen worden. Het zijn de checks waarvan je weet wat ze moeten doen, wat de uitkomst moet zijn. Dus kun je die automatiseren als test. Desgewenst met groene lampjes bij een geslaagde test en rode lampjes voor een falende test.

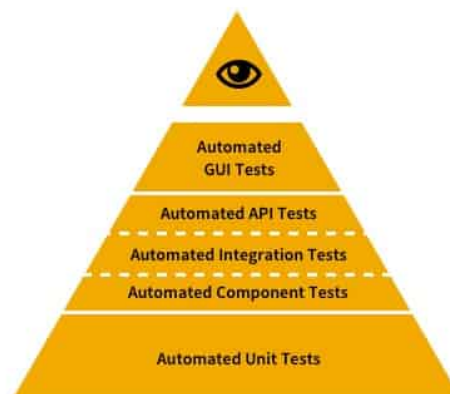
Een goede tester is een onderzoeker die software zodanig aan de tand voelt, dat je heel veel scenario's afvangt. Situaties van wegvallende verbindingen tot onmogelijke apparaten met een scherm en een browser. Bij wijze van spreken zelfs van situaties waarbij een kat over een toetsenbord loopt van de gebruiker loopt. Een goede tester kan z'n creativiteit maximaal kwijt in zijn onderzoeken en tests.

De belangrijkste genoemde punten uit de presentatie van Wim ten Tusscher:

- (Geautomatiseerd) testen is een must voor deugdelijke software die niet stuk gaat;
- Er is kennis en ervaring nodig van goede testers om scenario's te bedenken, te onderzoeken en te testen;
- Testen begint al vroeg, bij de ontwikkeling van code, niet pas aan het eind, als de deadline morgen is;
- Testen zorgt voor vertrouwen in je product, bij jezelf als producteigenaar en je mensen, maar ook bij de eindgebruiker.

De piramide

Met deze manier van automatiseren werk je ook volgens een driehoek, maar dan eentje die rechtop staan, zoals een stevige, degelijke piramide.  Piramides zijn 4000 jaar geleden gebouwd en staan nog steeds. Door je testautomatisering als een piramide in plaats van als ijshoorntje te benaderen, heb je een bestendige testoplossing. Piramides blijven staan,



ijshoorntjes breken.

Lees het complete verslag van dit kennisevenement op de [website van Enrise](#).