

Binnen de testwereld is testautomatisering de laatste tijd hot. Mijn eerste ervaringen met testautomatisering gaan echter al terug naar 2003 en 2009. Beide keren flopte de testautomatisering.

In beide gevallen was er veel code geschreven, maar waren er geen of heel weinig werkende testgevallen. Na deze twee flopervaringen had ik testautomatisering eigenlijk afgeschreven. Terugkijkend zie ik nu dat er in beide situaties dezelfde verkeerde keuzes zijn gemaakt:

- Opgezet als een apart technologieproject met lagere prio als de productontwikkeling;
- Er waren vage doelen zonder duidelijke strategie en deadlines;
- Doel was het automatiseren van bestaande handmatige tests;
- De tools waren leidend.

Tijden veranderen en de industrie ontwikkelt zich verder. In de nieuwe wereld van Agile, CI/CD en DevOps is het niet meer mogelijk al het testen handmatig uit te voeren. Ook zijn er inmiddels nieuwe tools met vele nieuwe mogelijkheden. De tijd is dus rijp voor top-testautomatisering! Hierbij is het wel van belang te realiseren dat alleen het verifiëren van gewenste functionaliteit (het zogenaamde “checken”) te automatiseren is. Geautomatiseerd checken wordt in een goede teststrategie altijd aangevuld met (handmatig) exploratory testen.

Verder is het van belang om een goede automatiseringsstrategie te hebben. Deze is gebaseerd op de testpiramide, waarbij de grootste testdekking in het onderste test level (unit test) wordt behaald. Naar behoefte (risico gedreven) wordt de unit testdekking aangevuld met integratie- en service- of API testen. Testautomatisering op GUI level moet beperkt van omvang zijn vanwege de hoge onderhoudsgevoeligheid en trage executiesnelheid. Het is van belang om voor elke user story en elke sprint de benodigde testautomatiseringsdekking op de verschillende test levels te bepalen. Hiermee bouw je elke sprint aan een groeiende set regressietesten op elk testlevel en blijft er ruimte over voor handmatig exploratory testen.

Voor top-testautomatisering is top-code nodig

Testautomatisering is code schrijven. Voor top-testautomatisering is dus top-code nodig. Top-code voldoet aan codestandaarden en ondergaat code review. Top-code heeft een goede architectuur en wordt, indien nodig, gerefactored. Daarnaast is top-code getest, dus test je geautomatiseerde tests op false positives en false negatives. Deze zijn namelijk sterk ondermijnend voor het vertrouwen in de geautomatiseerde tests.

Top-testautomatisering is onlosmakelijk verbonden met de productontwikkeling. Het is een teaminspanning die al begint tijdens de refinement; daar bepaal je namelijk je testdoelen, de testfocus (hoog/laag risico) en de testaanpak (geautomatiseerd, handmatig). Verder is het voor top-testautomatisering van belang dat het team stilstaat bij testbaarheid. Als er tijdens

de bouw van het product aandacht is voor testbaarheid, komt dat het (automatisch) testen zeer ten goede.

Vanwege de verstrengeling met productontwikkeling is de sprint pas gedaan (Done) als alle testautomatisering is gedaan. Dat wil zeggen: nieuwe testen gemaakt en getest, bestaande testen aangepast en getest of end-of-life verklaard. Dit kost inspanning en moet je dus meenemen in de inschatting van de hoeveelheid werk. Story points omvatten dus ook het werk om testautomatisering voor je te laten werken! Het is van groot belang dit heel serieus te nemen. Testautomatiserings debt en het "uitzetten" van niet werkende testen zijn dodelijk en leiden niet tot top-testautomatisering.

Last but not least: testautomatisering is geen doel op zich, het moet ondersteunend zijn aan het uiteindelijke product. Dit betekent dat je just-enough testautomatisering moet maken om de productrisico's, binnen de context waarin het product waarde voor de klant moet leveren, af te dekken. Teveel testautomatisering is een last omdat het onderhouden van de code dan een last wordt. Te weinig testautomatisering leidt tot ongedekte (regressie) risico's en te weinig tijd voor handmatig exploratory testen.

Dit artikel is een samenvatting van [de presentatie](#) die Wim ten Tusscher gaf tijdens de Polteq Conferentie op 7 juni 2018.