

TESTAUTOMATISERING: GEEN KWESTIE VAN GELUK MAAR VAN WIJSHEID

Door Ruud Teunissen • ruud.teunissen@polteq.com



Door onze gezamenlijke inspanning in de afgelopen decennia staan het belang en de rol van testen terecht niet meer ter discussie. Om dit vast te houden moet testen inspelen op de continue vraag naar kwaliteitsverbetering, kostenreductie, verkorten van de doorlooptijd en het verhogen van het aantal releases. Dit kan niet zonder een succesvolle implementatie en toepassing van testautomatisering. Succes mag geen kwestie van toeval of geluk meer zijn, maar van bewuste keuzes en wijsheid.

Het verleden en het heden

Als we eerlijk zijn, passen we testautomatisering tot nu toe niet overal met succes toe. Het is natuurlijk verleidelijk om uitgebreid stil te staan bij het hoe en waarom dat zo is. 'Naming and blaming' is volgens velen immers een populair tijdverdrijf van testers. Laten we onze energie echter steken in het bepalen hoe we implementatie en toepassing van testautomatisering wel succesvol kunnen aanpakken. En leren van de fouten die onmiskenbaar zijn gemaakt.

In ons enthousiasme over de schijnbaar onbeperkte mogelijkheden van de beschikbare tooling en technologie, beginnen we als 'boys met toys' met testautomatisering, zonder stil te staan bij wat we willen en kunnen bereiken. Sterker nog, het tool is vaak al geselecteerd en geïnstalleerd door relatieve buitenstaanders zoals management, inkoop, ontwikkeling, voordat een tester heeft gekeken of het tool überhaupt bijdraagt.

Het tool wordt vaak gezien als dé oplossing. Maar voor welk probleem? Belangrijke les uit het verleden en een van de essenties van succesvolle automatisering: 'Wees doelgericht, niet tool-gericht'. En dat doel is nooit testautomatisering zelf.

Overigens, tot nu toe zijn we als testers schijnbaar ongestraft weggekomen met het tekortschieten van test-automatisering. Misschien wel omdat testautomatisering door zowel testers als niet-testers tot nog toe te veel gezien is als een speeltje en te weinig als een hulpmiddel. In de huidige context is deze opstelling echter niet meer houdbaar.

Softwareontwikkeling wordt meer en meer Agile, projectmanagement gaat volgens Scrum en 'continuous delivery' is steeds vaker een feit. Logisch gevolg: om in deze continu veranderende en veeleisende context goed en gedegen te blijven testen en toegevoegde waarde te blijven leveren is succesvol toepassen van testautomatisering noodzakelijk. Laten we dus bij het begin beginnen en testautomatisering behandelen zoals het hoort: het gaat om het automatiseren van het testen.



Wanneer is testautomatisering succesvol?

Het eerste dat moet gebeuren is het gezamenlijk vaststellen van de 'acceptatiecriteria'. Wanneer is testautomatisering succesvol? Welke doelen willen we bereiken? Uitermate belangrijk is hierbij dat de verwachtingen en doelstellingen realistisch zijn. →

Om een parallel te trekken met automatiseren van de boekhouding: na selectie en implementatie van de boekhouding is er nog steeds een boekhouder nodig. De logica, vastlegging en standaardcontroles kunnen worden geautomatiseerd, de intelligentie en interpretatie blijft mensenwerk. Met andere woorden, na het automatiseren van het testen is nog steeds een tester nodig. Gelukkig wordt dit onderschreven door een van de vele uitspraken van 'orakel' en 'goeroe' James Bach.

'I love test automation, but I rarely approach it by looking at manual tests and asking myself 'how can I make the computer do that?'. Instead, I ask myself how I can use tools to augment and improve the human testing activity. I also consider what things the computers can do without humans around, but again, that is not automating good manual tests, it is creating something new.' - **James Bach**

Alles automatiseren en handmatig testen overbodig maken is geen realistisch doel. Het geautomatiseerd uitvoeren van een groot deel – bijvoorbeeld 80 procent - van de regressietest wel. Net als het automatiseren van de unittest binnen een Agile aanpak of het geautomatiseerd testen van veelvoorkomende handelingen binnen een business proces, zoals het vastleggen en/of wijzigen van klantgegevens.

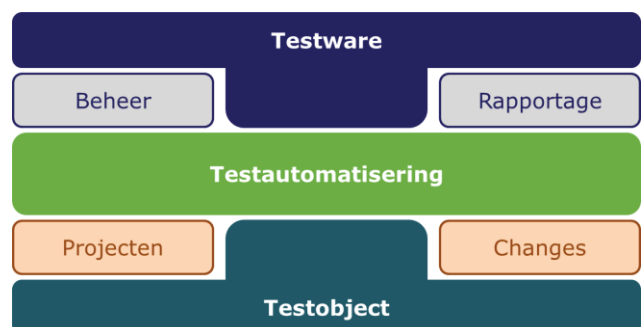
Een van de belangrijke kenmerken van bijvoorbeeld Agile en Scrum is dat we bij iedere stap of activiteit nadenken over de toegevoegde waarde ofwel de business case. Laten we dat ook doen bij testautomatisering. Stel dus steeds de vraag in hoeverre automatiseren van deze testen bijdraagt aan het bereiken van de afgesproken doelen. Ervaring leert dat ook hier de 80 – 20 regel van toepassing is: 80 procent van de inspanning gaat zitten in 20 procent van de automatiseerbare testen. Belangrijk dus om regelmatig vast te stellen of de inspanning van mensen en inzet van middelen in balans blijft met de opbrengst. Niet alleen op korte termijn, maar ook in de toekomst. Geautomatiseerde systemen vragen na implementatie immers om onderhoud.

Context driven

Yes, de veelgehoorde kreet 'context driven' is gelukkig ook op testautomatisering van toepassing. Succesvolle testautomatisering sluit immers nauw aan bij de gehanteerde test-, ontwikkel- en beheeraanpak en past in het systeemlandschap. Dus testautomatisering past in de context, is daarmee context driven. Maar welke aspecten van de context zijn daadwerkelijk van belang voor testautomatisering?

De verleiding is groot om te focussen op de (bestaande) testware en de te testen objecten. Hoe 'automatiseerbaar' zijn de huidige en toekomstige testgevallen? En hoe 'benaderbaar' zijn de te testen objecten voor de gekozen tooling? Ervaring heeft geleerd dat deze twee aspecten een 'gegeven' zijn. De impact op testautomatisering is dan ook veeleer dat zij in grote mate bepalend zijn voor wat bereikt – of niet bereikt – kan worden. Vergelijkbaar met de 'enablers' in veelgebruikte verbetermodellen.

Testautomatisering is hooguit een trigger voor verbetering van het testproces en de testware, het softwareproces en de software, maar zal daarbij nooit een sturende rol krijgen. →



In de praktijk is gebleken dat – binnen deze context – de mate waarin testautomatisering kan bijdragen vooral wordt bepaald door de architectuur voor testautomatisering, de wijze waarop de –geautomatiseerde– testscripts worden ontwikkeld en onderhouden en hoe wordt geprogrammeerd. Testautomatisering is immers automatisering. En succesvolle automatisering past binnen de context en focust op de bijdrage die het kan leveren: 'wat is nodig' staat centraal, niet 'wat kan'.

Testautomatisering moet bijdragen aan testdoelen

Laten we nog eens gaan kijken bij de boekhouder. Wat was veelal het oorspronkelijke doel van automatisering van de boekhouding? Zorgen dat de boekhouder zijn werk beter of sneller kan doen en zich kan concentreren op de kern van zijn werk: nadenken en interpreteren. Met andere woorden, automatisering helpt de boekhouder bij het behalen van zijn doelstellingen. En dit is precies wat als eerste met testautomatisering bereikt kan en moet worden. De tester kan zich concentreren op zijn belangrijkste taak: op een slimme manier testen, resultaten analyseren en inzicht geven in de kwaliteit. Automatisering neemt het routinematig werk uit handen en ondersteunt de tester. Het eerste niveau van testautomatisering is hiermee bereikt.

Testen bereikt meer met testautomatisering dan zonder

Gericht toepassen van de mogelijkheden die een gedegen architectuur en de bijbehorende tooling biedt, stelt testen in staat meer te bereiken met automatisering dan zonder. Denk hierbij niet alleen aan het uitvoeren van meer testgevallen door middel van het toepassen van data-driven technieken en het vaker draaien van geautomatiseerd testen buiten de normale werktijden van de testers, maar ook door geautomatiseerde integratie met andere tooling – zoals defectmanagement en configuratiemanagement – die gebruikt wordt binnen projecten en organisaties. Een logisch vervolg op het uit handen nemen van routinematig werk en daarmee het tweede niveau van testautomatisering.

Schaalbaar en toekomstvast

In een wereld waar Agile Development, projectmanagement volgens Scrum en 'continuous integration/delivery/deployment/improvement/...' worden toegepast en nagestreefd is het noodzakelijk testautomatisering naar het –op dit moment– hoogst mogelijke niveau te tillen. Dit betekent dat testautomatisering zelf ook de principes toepast die passen bij de ontwikkel- en beheeraanpak in de aangegeven context. Geautomatiseerde testscripts worden object georiënteerd ingericht om onderhoudsinspanning te reduceren. Testautomatisering is continu op zoek naar mogelijkheden voor verbetering. Het team werkt nauw

In de wereld van Tolkien is sprake van: 'One ring to rule them all'. In de praktijk is duidelijk gebleken dat 'One tool to rule them all' vrijwel altijd leidt tot ongewenste complicaties. Vandaar dat bewust consequent gesproken wordt over 'tooling' – een verzameling tools – en niet over één tool.



samen met ontwikkeling om kennis, ervaringen, kunde en vaardigheden te delen. Bestaande – werkende – scripts worden 'gere-factor'd'. De testarchitectuur sluit nauw aan op de systeemarchitectuur en geautomatiseerd testen vindt, waar mogelijk, plaats op alle testlevels. Van unittest tot het testen van end-to-end business processen. Met andere woorden, Fit-for-purpose en Fit-for-context, en daarmee het derde niveau van testautomatisering. →

De tien pijlers van testautomatisering

Meer dan achttien jaar geleden hebben vier pijlers, fasering, organisatie, technieken en infrastructuur, de Nederlandse testwereld veroverd. Zij vormen, hoewel regelmatig in een nieuw jasje gestoken, aangepast, uitgebreid en verbeterd, de basis voor het testen binnen veel organisaties.

Het selecteren van tooling is bekend terrein, reden om hier nu niet al te lang bij stil te staan. Essentieel is wel dat breder gekeken wordt dan de normale selectiecriteria, zoals systeemvereisten, ondersteunde omgevingen, gebruikersgemak. De geselecteerde tooling moet – in lijn met de doelstellingen van testautomatisering – ‘fit for purpose’ en ‘fit for context’ zijn.

De vraag is nu: welke pijlers vormen de basis voor succesvol automatiseren van het testen? En hoeveel zijn er nodig? Zoals al eerder aangegeven wordt de kern gevormd door de gekozen architectuur voor testautomatisering, de wijze waarop de geautomatiseerde testscripts worden ontwikkeld en onderhouden en welke programmeerstandaarden gehanteerd worden. De vierde pijler is tooling. Deze vormt het platform waarop testautomatisering wordt ontwikkeld, onderhouden en draait, en bestaat uit een selectie van tools en – zelf ontwikkelde – software. Uiteraard zijn de testomgeving en de testdata cruciaal. De testomgeving en testdata moeten niet alleen beschikbaar, bereikbaar en (her)bruikbaar zijn. In lijn met het doel en de scope van testen en testautomatisering, moeten zij end-to-end testen over verschillende platforms ondersteunen. Integratie met andere tooling, zoals defect- en configuratiemanagement, is de zevende pijler. Gedegen testautomatisering staat of valt met een goed testautomatiseringsteam. Dat vormt dan ook pijler nummer acht. Samen met een strategie om te borgen dat de gestelde doelen realistisch zijn en op een transparante en efficiënte wijze worden geïmplementeerd, en een doordachte manier om alle activiteiten te plannen, begroten en bewaken, staat de teller op tien.

| Key area | | Contributing | | | | Adding value | | | | Optimizing | | | |
|----------|-------------------------|-------------------------------------|---|---|---|---------------------|---|---|---|---------------------------------------|---|---|---|
| 1 | Automation architecture | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | |
| 2 | Automation scripts | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | |
| 3 | Automation standards | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | |
| 4 | Tooling | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 5 | Test environment | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 6 | Test data | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 7 | Tool integration | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | |
| 8 | Test automation team | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | |
| 9 | Automation strategy | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | |
| 10 | Planning & Estimation | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | |
| | | Enable testing to achieve its goals | | | | Mitigate more risks | | | | Scalable, fit for purpose and context | | | |

Volwassenheidsmatrix

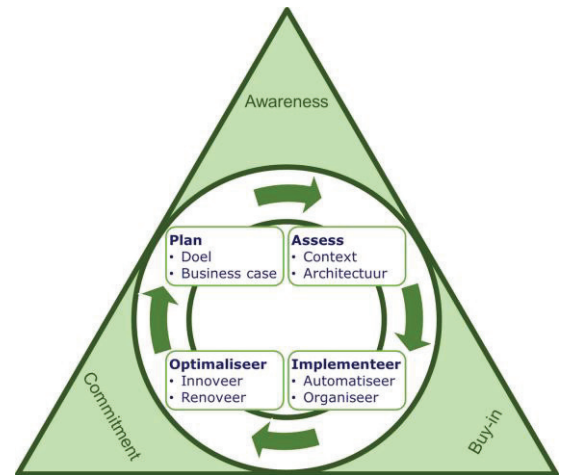
Stapsgewijze implementatie

Het vaststellen van het doel, realistisch en haalbaar, inzicht verkrijgen in de context en een analyse van het huidige en vereiste niveau, vormen de cruciale eerste stappen van succesvolle implementatie en verbetering van testautomatisering. Een goed doordachte architectuur en een pragmatisch implementatieplan zijn een logisch vervolg. →

Om testautomatisering die bijdrage te laten leveren die nodig is, is per pijler gedetailleerd en objectief aangegeven aan welke eisen moet worden voldaan om die bijdrage te leveren. Het resultaat is een volwassenheidsmatrix die het mogelijk maakt om testautomatisering van scratch af aan stap voor stap in te richten. Maar ook het naar een hoger niveau tillen van reeds bestaande testautomatisering wordt op transparante wijze ondersteund. In beide gevallen ligt de focus op de toegevoegde waarde die testautomatisering kan leveren en geldt: ‘doelgericht, niet toolgericht’.

Hierbij worden alle verzamelde ingrediënten samengevoegd tot een consistent geheel: test-automatisering die voldoet aan de verwachtingen en een organisatie die in staat is dit niet alleen neer te zetten, maar ook te onderhouden en optimaliseren.

In de huidige context is het uiteraard geen toeval dat de voorkeur uitgaat naar een Agile/Scrum-aanpak. Doel van de eerste sprint is het inrichten van de benodigde omgevingen, aanschaf en installatie van de benodigde tooling en het technisch 'aan de praat' krijgen van de omgeving. De architectuur, het ontwerp, wordt omgezet in een werkend testautomatiseringsplatform.



Tijdens de volgende sprint automatiseert het team een beperkt aantal representatieve scripts en voeren zij een soort 'Proof of Concept' van de gekozen oplossing uit. Aan het eind van deze sprint 'staat' het platform. De afsluitende demo geeft alle betrokkenen inzicht en vertrouwen in de gekozen oplossing. Hoeveel sprints uiteindelijk nodig zijn voor het behalen van de doelstellingen is uiteraard afhankelijk van de omvang en de context. Hierbij komen de testware en het testproces, de software en het softwareproces nadrukkelijk in beeld. Bij testware gaat het om de intrinsieke kwaliteit: mate van detail, consistentie, inzicht in wat getest wordt en gestelde prioriteiten. Bij software gaat het veeleer om de toegankelijkheid en mate waarin men zich aan standaarden heeft gehouden: input, output, herkenbare en benaderbare objecten, elementen, controls en eigenschappen. Testautomatisering is pas succesvol als het duurzaam is. Daarom is het van belang al vanaf het begin aandacht te besteden aan de aansluiting op de test-, ontwikkel- en beheeraanpak. De business blijft immers niet stil staan, de wijzigingen en projecten gaan door en testautomatisering dient hier naadloos op aan te sluiten. Dit zie je dan ook terug op diverse punten binnen de tien pijlers. Maar het is ook goed om na te denken waar testautomatisering gaat 'landen' in de organisatie. Parallel aan de implementatie moet de organisatie worden voorbereid op de nieuwe en uitdagende taak.

De toekomst

'A fool with a tool is still a fool' is en blijft waar. Maar het nadrukkelijke advies 'Structure then tool', dat Testen volgens TMap® (1995, pagina 441) gaf, is echt achterhaald. Sterker nog, ik denk dat succesvolle implementatie en toepassing van testautomatisering, inclusief de inzet van tooling, juist helpen bij het bereiken van de optimale balans tussen structuur en vrijheid van handelen die nodig is om in de toekomst als testers een essentiële bijdrage te blijven leveren. Testautomatisering met de huidige beschikbare tooling biedt ongekennde mogelijkheden en het is uitdagend voor testers en ontwikkelaars. Wie weet gaan ontwikkelaars testen nu wel een leuk vak vinden. Ik neem in ieder geval de uitdaging aan om automatisering van het testen tot een succes te maken. Wie weet ga ik ontwikkeling ook wel een leuk vak vinden... ←

