

KEES STUDY: ERVARINGEN MET ACCEPTATIE TESTAUTOMATISERING

Door Kees Blokland • kees.blokland@polteq.com



Twee jaar geleden kreeg ik de vraag: 'Hier is een plan met een begroting voor het automatiseren van de regressietest voor de business testgevallen, maar er zijn nog vragen, waarop het antwoord te lang uitblijft. Kun jij onderzoeken wat er nodig is om het proces vlot te trekken?' Zo startte voor mij een interessant en leerzaam traject.

Analyse van het probleem

Het plan met de schatting bevatte een lijst met te automatiseren tests voor vier verschillende applicaties en benoemde daarnaast een aantal technische zaken die geregeld moesten worden. Om te onderzoeken wat er moest gebeuren om het plan vlot te trekken, plande ik gesprekken met de betrokkenen. Omdat het proces rond het plan niet voor niets stil was komen te liggen, bereidde ik de gesprekken goed voor. Dit deed ik aan de hand van een checklist, gebaseerd op een verbeteraanpak voor testautomatisering (TI4Automation, REF³). Ik bedacht van tevoren welke informatie ik bij welke bron zou willen halen (niet alleen via gesprekken overigens, ik gebruikte ook andere bronnen).

TI4Automation checklist	Test manager	Project leader	Test automation group	Customers of the tests	Test plan	Dashboard daily test
01. Automation architecture	X		X			
02. Automation scripts			X	X		X
03. Automation standards			X			
04. Tooling	?		X			
05. Test environment			X			
06. Test data			X			
07. Tool integration	X		X			
08. Test automation team	X		X	X		
09. Automation strategy	X	X	X	X	X	
10. Planning & Estimation	X	X	X			

Resultaten van de analyse

De belangrijkste bevindingen uit het vooronderzoek waren:

1. het inrichten van de testautomatiseringsarchitectuur was nog niet in het plan meegenomen;
2. de lijst met te automatiseren testgevallen moest worden bijgewerkt naar de nieuwe release;
3. de lijst van applicaties waarvoor tests geautomatiseerd moesten worden klopte niet;
4. de 'automatiseerbaarheid' van de testgevallen was onvoldoende (zo was niet inzichtelijk hoe testgevallen uitgevoerd moesten worden en welke checks moesten plaatsvinden; dat zat alleen in de hoofden van mensen);
5. voor enkele applicaties draaide al dagelijks een setje geautomatiseerde tests om te kunnen zien dat de testomgeving beschikbaar was;
6. rond één van de applicaties was de nood hoog: een regressietestcyclus kostte twee weken testen door twee mensen; de menskracht was een probleem maar ook de lange doorlooptijd en de testdekking was eveneens onvoldoende.

Punt zes illustreerde meteen goed waarom de business belang had bij testautomatisering: zorgen

Problem

- Manual testing legacy systems costs too much time

Risks

- Problems remain undetected and show in production
- Other important tasks of involved people are squeezed
- High testing cost

Root causes

- Regression test sets are large
- Running test cases requires a lot of domain knowledge
- (Regression in the applications → no testing problem)

Recommendation

- Automate as much testing as possible to free up people and to increase regression test coverage

³ <https://improvement.polteq.com/ti4automation>

dat de beheerders en gebruikers minder tijd kwijt zijn aan regressietesten. Dat de regressietest vooral een grotere dekking krijgt en ook vaker uitgevoerd kan worden. Dit moest de inspanning van de beheerders terugbrengen en het aantal incidenten in productie verkleinen.

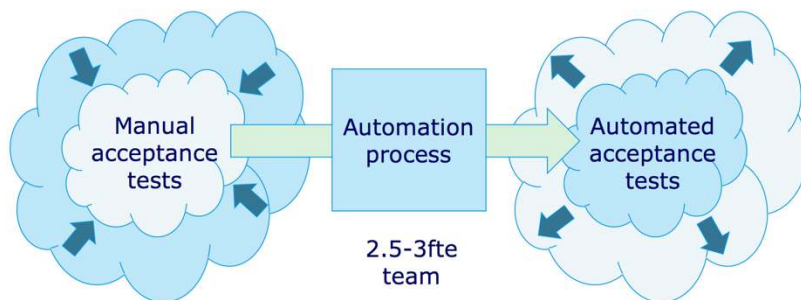
Voorstel voor vervolg

'Wat is je voorstel?', was de vraag die ik zag aankomen 😊. Uit het vooronderzoek was me nog meer duidelijk geworden. Zo bleek het cruciaal dat het testautomatiseringsteam (het team) en de deskundigen (SME's, subject matter experts) direct bij elkaar in de buurt kwamen te zitten. Het team moet veel en vaak informatie halen en moet de gemaakte scripts veel en vaak laten valideren door de SME's en dus was werken op dezelfde locatie een belangrijk startpunt in het voorstel.

Andere pijlers van de voorgestelde aanpak waren een kort-cyclische werkwijze en het starten met een POC (proof of concept). De POC was bedoeld om de aanpak uit te proberen en om het vertrouwen van de business te winnen, zodat zij daadwerkelijk in de testautomatisering wilden investeren.

Het voorstel viel in goede aarde en business vroeg ons (de afdeling testen en testautomatisering) om een POC uit te voeren op de tests van de applicatie waar de nood het hoogst was. De POC maakte nog meer duidelijk:

- een bestaande lijst van testgevallen van de beheerders kon prima als startpunt dienen;
- automatiseren van de tests was technisch gezien niet bijzonder moeilijk;
- tooling hiervoor was direct beschikbaar (HP UFT en HP QC);
- de beheerders moesten wel voldoende tijd beschikbaar maken ter ondersteuning;
- werken op dezelfde locatie was inderdaad nodig;
- het is haalbaar!.



Aan de slag

De business had vertrouwen in onze aanpak en gaf ons de opdracht om in drie maanden te laten zien wat we samen met de beheerders konden realiseren. Bij succes zouden we ook de rest van het jaar doorgaan. Een afdelingsmanager binnen de business werd onze Product Owner. Hij gaf ons een aantal specifieke opdrachten mee. Zo wilde hij dat wij een check uitvoerden op de dekking van de tests, dat alle testgevallen in HP Quality Center kwamen en dat de geautomatiseerde tests door de betrokken beheerders zelf konden worden gedraaid en geanalyseerd. De testautomatisering in de drie maanden verliep succesvol en inmiddels zijn we (dik anderhalf jaar na de POC) bezig met het vierde automatiseringsproject.

Hoe werken wij en wat maakt het succesvol?

Stap één: testanalyse

Het uitgangsmateriaal is een lijst met testgevallen, vaak met nog vrij weinig detailinformatie. De testanalist in het team speelt met de beheerders de testgevallen stap-voor-stap na en voegt aan de testgevallen toe op welke punten controles in de applicatie moeten plaatsvinden. Met name dat laatste vergt veel aandacht: zonder deze checks krijgen de geautomatiseerde tests niet hun volledige waarde. De beheerders en de testanalist bespreken de noodzaak om tests te handhaven en kijken naar mogelijkheden om tests te combineren. De totale testdekking van de testset wordt visueel gemaakt, bijvoorbeeld in een classification tree. Daarmee worden gaten in de dekking zichtbaar, waarvoor tests worden toegevoegd. Dit resulteert in een testset

- die klaar is voor automatisering;
- die ook beschikbaar is in QC voor handmatige uitvoering (sommige tests worden uiteindelijk niet geautomatiseerd);
- waarvan duidelijk is welke risico's worden afgedekt (en welke niet);
- waarvan de dekking verbeterd is.

Stap twee: automatiseren

De tests worden vervolgens in HP-UFT geprogrammeerd. Een modulaire opbouw voorkomt dat veel code wordt gekopieerd. Hoe meer herbruikbare bouwblokken er zijn, hoe sneller het automatiseren van meer tests verloopt. Robuustheid is belangrijk: zo wordt eerst gecheckt of een veld aanwezig is en daarna wordt pas een waarde ingevuld, waarmee wordt voorkomen dat de test nutteloos 'doordendert'. Ook worden consequent screenshots opgeslagen van de schermen. Deze en nog meer ontwerpregels worden door de testautomatiseerders consequent toegepast. Als een test de eindstreep niet haalt, dan is goed te zien waar en waardoor het mis gaat. De tests die af zijn worden iedere dag gedraaid. In eerste instantie is dat om de robuustheid van de testscripts te testen, maar ook om veranderingen in de applicaties te signaleren. Als dat gewenste veranderingen zijn, dan moeten de scripts worden aangepast, anders is sprake van een bevinding op een applicatie. De investeringen in de robuustheid en in de onderhoudbaarheid van de geautomatiseerde tests worden dubbel en dwars terugverdiend. In veel gevallen kunnen de scripts door mensen gedraaid worden die niet gespecialiseerd zijn in testautomatisering, zoals beheerders.

Stap drie: stoppen met automatiseren

De volgorde van automatiseren verloopt volgens de prioriteiten die daarvoor zijn afgesproken. Na een tijdje ontstaat een beeld van de 'team velocity': hoeveel tests wekelijks worden opgeleverd. Met de opdrachtgever wordt de investering per test naast de waarde van de potentieel nog te automatiseren tests gelegd. Op het moment dat de balans doorslaat, stoppen we.

	App 'X'	Benefit	Ease
	Test case 1	H	H
automate	Test case 2	H	M
don't automate	Test case n	L	L

CSF's

Tot slot een aantal succesfactoren van dit nog steeds lopende programma.

- **Het team en de beheerders werken bij elkaar in de buurt.** Het team moet gemakkelijk even bij de beheerder kunnen aankloppen en de beheerders kunnen direct met eigen ogen de resultaten zien. De beheerders kunnen niet fulltime meedraaien en dat hoeft ook niet. Wel zijn afspraken nodig voor voldoende beschikbaarheid. Soms helpt het om op vaste tijdstippen op de dag vragen met de beheerders door te nemen en met hen de tests door te lopen.
- **Beheerders regelmatig opzoeken als werken op dezelfde locatie niet mogelijk is.** Een deel van de organisatie is niet in Nederland. Door regelmatig naar de andere locatie af te reizen wordt de basis gelegd voor de benodigde samenwerking.
- **Product owner / sponsor.** De afdelingsmanager bepaalt de teamprioriteiten, maakt afspraken over de beschikbaarheid van de beheerders en gebruikt zijn invloed om de investeringsruimte te creëren voor de testautomatisering.
- **Software-engineering principes toepassen.** Dus niet quick & dirty scripting maar quick & clean programmeren, in brokken die klein en overzichtelijk zijn. Eén van de teamleden is de testautomatiseringarchitect en ziet toe op de kwaliteit van de testcode.
- **Teamsamenstelling.** Het team bestaat uit testautomatiseringsengineers en een testanalist die ook de dagelijkse coördinatie doet. Het team is in hoofdzaak zelfsturend en de teamleden helpen elkaar waar nodig. Ondergetekende is een dag in de week betrokken voor strategische zaken zoals langetermijnplanning en visieontwikkeling. ←