

# WHITEPAPER

## *Test Tool Management*



## COLOFON

Dit whitepaper is gepubliceerd door de TestNet werkgroep Test Tool Management.

*TestNet is een in Nederland gevestigde professionele organisatie voor softwaretesters door softwaretesters, die een platform biedt voor kennisuitwisseling in ICT-producttesten. Het faciliteert netwerken, organiseert regelmatig evenementen en werkgroepen, en publiceert een online tijdschrift en een "Flash". Opgericht in 1997, heeft het meer dan 1500 leden en streeft het ernaar om het testen in de IT-wereld te professionaliseren en het bewustzijn ervan te vergroten.*

De test tool management werkgroep heeft tot doel de probleemstellingen rondom test tool management en het test tool managementproces in kaart te brengen en deze uit te werken. Huidige deelnemers van de werkgroep en auteurs van deze whitepaper zijn:

- Ronald Kerkhoff (lead),
- Otman Zemouri,
- Mickel Jacobs,
- Bert Veldman.

Naast de auteurs hebben ook een aantal collega's en vakgenoten waardevolle bijdragen geleverd aan de totstandkoming van dit whitepaper. We willen de volgende personen bedanken voor hun input, feedback en ondersteuning:

- Rik Marselis,
- Marco Jansen van Doorn,
- Noud Esselink,
- Dolf Cornelius,
- Danny van Loenen,
- Jos van Rooyen.

Als je lid bent van TestNet en geïnteresseerd bent in het deelnemen aan deze werkgroep, dan ben je van harte welkom. Stuur in dat geval een email naar de werkgroep coördinator: [wergroepen@testnet.org](mailto:wergroepen@testnet.org)

Whitepaper Test Tool Management Versie 1.1,  
23 April 2024

© 2024 Vereniging TestNet. Alle rechten voorbehouden.

Dit whitepaper is beschermd door auteursrecht en mag niet geheel of gedeeltelijk worden gereproduceerd, gedistribueerd, verzonden, weergegeven, gepubliceerd of uitgezonden zonder de voorafgaande schriftelijke toestemming van de auteurs. U mag geen auteursrechtelijk beschermd werk of andere in dit whitepaper opgeslagen informatie kopiëren of opslaan in enige vorm zonder voorafgaande toestemming van de auteurs.

Voor toestemming om het materiaal uit dit whitepaper te gebruiken, neemt u contact op met de auteurs via Vereniging TestNet.

# **Inhoudsopgave**

<b>INLEIDING</b>	<b>4</b>
<b>DE EVOLUTIE VAN SOFTWARETESTEN: TECHNOLOGIE, ORGANISATIE EN TOOLS</b>	<b>4</b>
<b>EEN VIRTUEEL SCENARIO UIT DE PRAKTIJK: HET VERHAAL VAN MOBYIT</b>	<b>7</b>
<b>DE ROL VAN TEST TOOLS</b>	<b>8</b>
<b>TEST TOOL MANAGEMENT PROCESSEN</b>	<b>9</b>
<b>Test Tool Strategie</b>	<b>9</b>
<b>Architectuur</b>	<b>9</b>
<b>Selectie</b>	<b>10</b>
<b>Beheer &amp; Onderhoud</b>	<b>11</b>
<b>Verwijdering</b>	<b>11</b>
<b>ROLLEN EN VERANTWOORDELIJKHEDEN</b>	<b>12</b>
<b>Test tool eigenaar</b>	<b>12</b>
<b>Test tool beheerder</b>	<b>13</b>
<b>Test tool gebruiker</b>	<b>13</b>
<b>HET BELANG VAN TEST TOOL MANAGEMENT</b>	<b>14</b>
<b>De voordelen van test tool management</b>	<b>14</b>
<b>De uitdagingen van test tool management</b>	<b>15</b>
<b>TEST TOOL MANAGEMENT IMPLEMENTEREN?</b>	<b>16</b>
<b>TEST TOOL CLASSIFICATIESYSTEEM (OF TAXONOMIE):</b>	<b>17</b>
<b>Test tool classificatiesysteem in test tool management:</b>	<b>17</b>
<b>CONCLUSIE</b>	<b>17</b>
<b>BRONVERMELDING</b>	<b>18</b>

## Inleiding

Dit whitepaper behandelt de verschillende probleemstellingen rondom test tool management en de daaropvolgende processtappen die moeten worden gevolgd voor een volwassen implementatie en optimalisatie van test tool management. We bieden een kijk op test tool management, inclusief de betekenis ervan, de voordelen en de cruciale rol die het speelt bij het verbeteren van het testproces. Verder gaan we in op de risico's van onvoldoende test tool management en bieden we oplossingen om dit te voorkomen.

## De Evolutie van Softwaretesten: Technologie, Organisatie en Tools

Het is opmerkelijk om te zien hoe niet alleen technologische maar ook organisatorische vooruitgang van het afgelopen decennium een aanzienlijke invloed heeft gehad op de wereld van softwaretesten. Een duidelijk bewijs hiervan is de significante toename van het aantal beschikbare test tools, maar ook de opkomst van Agile ontwikkelmethoden en de recente trend van DevOps.

Technologische vooruitgang heeft geleid tot de ontwikkeling van nieuwe test tools die het testen van software efficiënter en effectiever maken. Zo zijn er tools beschikbaar voor het automatiseren van testscenario's (bijvoorbeeld Unit Test en Functionele tests), het beheren van testgevallen (bijvoorbeeld het analyseren van testgegevens en het rapporteren van testresultaten), zoals geïllustreerd in figuur 1, ingedeeld in tool specifieke aandachtsgebieden.



Figuur 1 - Software test tool landschap [5]

Organisatorische vooruitgang heeft geleid tot een verschuiving van traditionele waterval ontwikkelingsmethoden naar Agile- en DevOps-methoden. Agile ontwikkelmethoden zijn gebaseerd op het concept van iteratieve ontwikkeling, waarbij software in korte cycli/sprints wordt ontwikkeld, getest en beheerd. DevOps is een aanpak die softwareontwikkeling en -operatie combineert.

Deze organisatorische vooruitgang heeft een aantal gevolgen voor softwaretesten. Zo moet softwaretesten sneller en flexibeler worden om aan te kunnen sluiten bij de snelle ontwikkelingscycli/sprints van Agile en DevOps.

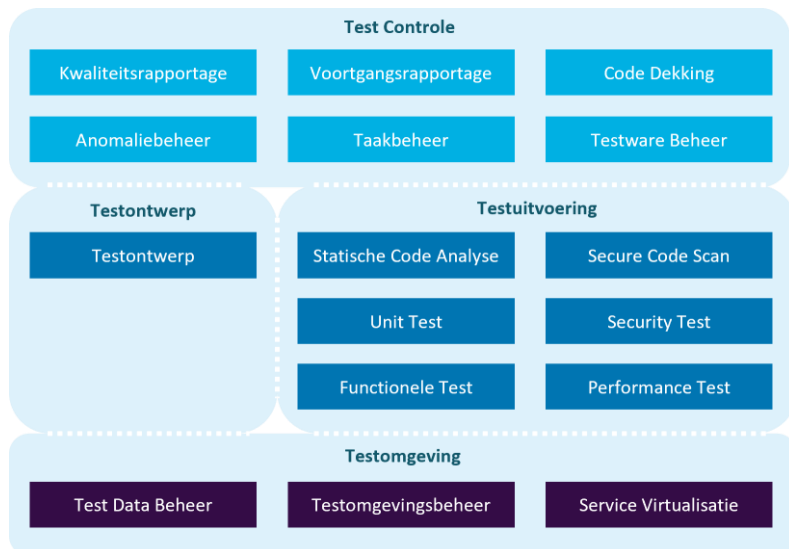
Een voorbeeld van hoe technologische en organisatorische vooruitgang elkaar hebben beïnvloed, is de ontwikkeling van "Continuous Integration & Continuous Delivery" (CI/CD). CI/CD is een proces dat het automatisch bouwen, testen en deployen van software mogelijk maakt. CI/CD wordt vaak gebruikt in combinatie met Agile en DevOps.

CI/CD maakt gebruik van geautomatiseerde test tools om software te testen bij elke wijziging. Dit helpt om bevindingen vroeg in het ontwikkelingsproces te identificeren en te corrigeren.

De groei van het aantal test tools heeft geleid tot de ontwikkeling van tools die specifiek zijn ontworpen om

verschillende aspecten van het testproces te ondersteunen.

Deze aspecten kunnen worden onderverdeeld in vier hoofdcategorieën: Test Controle, Testontwerp, Testuitvoering en Testomgeving (zie figuur 2).



Figuur 2 – Test tool capabilities [4]

TMap Quality For DevOps teams [4] beschrijft deze vier categorieën als volgt:

**Test Controle** omvat:

- Kwaliteitsrapportage: het verzamelen en rapporteren van informatie over de kwaliteit van de software
- Voortgangsrapportage: het verzamelen en rapporteren van informatie over de voortgang van het testproces
- Code dekking: het meten van de mate waarin de code van de software is getest
- Anomaliebeheer: het beheren van afwijkingen van de verwachte resultaten
- Taakbeheer: het plannen en beheren van testtaken
- Testware beheer: het beheren van testgegevens, testscenario's en andere testmiddelen

**Testontwerp** omvat:

- Het ontwerpen van testgevallen en testscenario's

**Testuitvoering** omvat:

- Statische codeanalyse: het automatisch analyseren van de code op mogelijke fouten
- Secure code scan: het automatisch scannen van de code op kwetsbaarheden
- Unit test: het testen van individuele eenheden van code
- Security test: het testen van de beveiliging van de software
- Functionele test: het testen van de functionaliteit van de software
- Performance test: het testen van de prestaties van de software

**Testomgeving** omvat:

- Testdatabeheer: het beheren van testgegevens
- Testomgevingsbeheer: het beheren van testomgevingen
- Service virtualisatie: het virtualiseren van testomgevingen

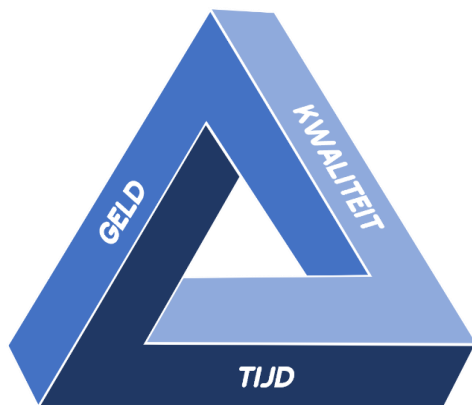
De ontwikkeling van test tools voor deze vier categorieën heeft een aantal voordelen voor softwaretesten. Zo kunnen test tools helpen om:

- Het testproces efficiënter en effectiever te maken
- Beter inzicht te krijgen in de kwaliteit en de risico's van het softwareproduct
- De kosten van softwaretesten te verlagen

Door de toename van het aantal test tools wordt het voor teams niet alleen moeilijker de juiste test tools te vinden en te implementeren maar ook om deze te onderhouden. Dit kan leiden tot een aantal uitdagingen, zoals:

- *Inefficiëntie*: Teams kunnen tijd en geld verspillen aan het leren en gebruiken van tools die niet geschikt zijn voor hun behoeften.
- *Ineffectiviteit*: Teams kunnen niet alle functies van de tools benutten, waardoor ze de kwaliteit van het testproces niet kunnen verbeteren.
- *Kwaliteit*: Teams kunnen fouten maken bij het gebruik van tools, waardoor de kwaliteit van de testgevallen kan worden aangetast. Wat weer tot gevolg heeft dat er minder inzicht wordt verkregen op de kwaliteit van het softwareproduct.
- *Doublures*: Test tools hebben vaak overlappende functionaliteiten. Een tool die specifiek is ontworpen voor één taak, kan ook functies bevatten die overlappen met andere tools. Dit kan leiden tot redundantie, waarbij meerdere tools worden gebruikt voor hetzelfde doel, wat kosteninefficiënt is.
- *ROI (Return on Investment)*: Dit houdt rekening met de functionaliteit die een tool biedt in verhouding tot de kosten. Deze kosten kunnen bestaan uit projectkosten, personeelskosten (zoals het opleiden van engineers en het gebruik van de tool), en licentiekosten. De ROI wordt beoordeeld op basis van de bijdrage die de tool levert aan een effectiever en efficiënter testproces.

Kijken we naar bovenstaande uitdagingen, dan kunnen we niet om het zogenaamde “duivelsdriehoek” fenomeen heen. Elke (project)organisatie heeft weleens te kampen gehad met dit fenomeen, waarbij de aspecten tijd, geld en kwaliteit elkaar altijd beïnvloeden.



Figuur 3 – Duivelsdriehoek [6]

Testen is een belangrijke manier om inzicht in kwaliteit te geven en dit te borgen. Het kan informatie geven over onder andere de aanwezigheid van fouten, de functionaliteit van het product of de dienst, en de gebruiksvriendelijkheid. Door testen te integreren in het hele proces, van ontwerp tot implementatie, geeft dit inzicht in de kwaliteit van de software.

Dit is belangrijk om te voorkomen dat de software niet (meer) voldoet aan de verwachtingen van de gebruikers, of aan de vereiste functionele ontwerpen/specificaties.

De hoeveelheid inspanning die nodig is om meer inzicht in kwaliteit te geven, hangt af van een aantal factoren, zoals de complexiteit van het product of de dienst, de aard van de kwaliteitseisen, en de beschikbare middelen, zoals tijd, geld en expertise.

Een kosten-batenanalyse helpt bij het bepalen van de benodigde inspanning. Hierbij worden de kosten van inspanningen afgewogen tegen de baten zoals:

- verminderde kosten door minder fouten,
- verhoogde klant- en teamtevredenheid,
- verhoogde business value door gedetecteerde fouten per opgeleverde sprint/functionaliteit.

Bezuinigen op tijd en geld kan leiden tot een teleurstellend kwaliteitsniveau en onvoldoende business value, met mogelijke productie-incidenten als gevolg. Deze overweging geldt ook voor de keuze tussen testtools en handmatig testen, en de selectie van de juiste tool voor het project, de afdeling, het systeemlandschap. Terwijl testtools de testtijd aanzienlijk kunnen verkorten en kosten kunnen besparen, is het vermeldenswaardig dat de initiële investering en lopende onderhoudskosten aanzienlijk kunnen zijn. Handmatig testen kan minder kosten in termen van initiële investering en opleiding, maar kan meer tijd kosten en mogelijk minder grondig zijn, vooral bij regressietests (tester fatigue).

Het is een voortdurende uitdaging om de juiste balans te vinden tussen deze factoren voor de meest effectieve en efficiënte teststrategie. Het is belangrijk om de specifieke behoeften en context van het project te overwegen bij het nemen van deze beslissing.

## Een virtueel scenario uit de praktijk: Het verhaal van MobyIT

*Een bloeiend technologiebedrijf genaamd MobyIT, werkte aan een revolutionaire mobiele app "SmartConnect". Het DevOps-team was vastbesloten om een product te leveren dat voldeed aan de verwachtingen van de gebruikers.*

*Om hun doel te bereiken, besloot het DevOps-team de test tool TestQuestPro te gaan gebruiken die hen zou helpen niet alleen om betere inzichten op de kwaliteit van de app te verkrijgen, maar ook om de kwaliteit en herbruikbaarheid van de test cases te vergroten. Vol enthousiasme implementeerden ze de nieuwe tool in hun testproces.*

*In het begin leek de tool hen te helpen inzicht te krijgen op de kwaliteit. Het team was onder de indruk van de uitgebreide mogelijkheden van TestQuestPro en de veelbelovende resultaten die het opleverde. Echter, na verloop van tijd begonnen er uitdagingen te ontstaan. Het team realiseerde zich dat de tool complexer was dan ze eerst hadden gedacht. Het vereiste diepgaande training en oefening om de functionaliteiten volledig te benutten. Het team merkte dat ze veel tijd besteedden aan het begrijpen en configureren van de tool, wat resulteerde in vertragingen in het testproces.*

*Bovendien ontdekte het team dat TestQuestPro regelmatig niet naar verwachting functioneerde, waardoor ze gegevens en voortgang verloren. Dit veroorzaakte frustratie wat ten koste ging van de efficiëntie van het team. Ze waren gefrustreerd door de gebrekkige ondersteuning van de toolleverancier, wat hun inspanningen om deze problemen op te lossen bemoeilijkte. Bovendien was er een steile leercurve geassocieerd met TestQuestPro, wat betekent dat het veel tijd kostte om de tool volledig te begrijpen.*

*Naast deze technische uitdagingen, ontving het DevOps-team een onverwachte factuur voor de licentiekosten van TestQuestPro, die het budget van het project overschreed. Dit zette het team onder druk om alternatieve oplossingen te vinden om binnen de financiële grenzen te blijven.*

*Uiteindelijk realiseerde het DevOps-team zich dat ze zich hadden laten verleiden door de geavanceerde functies van TestQuestPro zonder grondig rekening te houden met de trainingsbehoeften, technische ondersteuning en kosten. Het team besloot dat ze terug moesten keren naar hun vertrouwde test tool en zich moesten richten op een effectiever test tool management.*

*Concluderend, het verhaal van MobyIT illustreert de uitdagingen die kunnen ontstaan als teams en beheerteams niet effectief test tool management toepassen. Zonder een duidelijk plan voor de implementatie en het gebruik van test tools, kunnen deze teams problemen ondervinden met de complexiteit, functionaliteit, ondersteuning en kosten van de test tool. Dit kan leiden tot een lagere volwassenheid van het test tool managementproces, wat invloed kan hebben op de algehele prestaties van het team en de efficiëntie van het beheer binnen het project.*

In bovenstaande voorbeeld wordt het probleem van gebrek aan planning en coördinatie geïllustreerd door het verhaal van MobyIT. Het bedrijf implementeerde een nieuwe test tool zonder eerst de capaciteiten van het team te analyseren. Dit is een voorbeeld van een gebrek aan planning en coördinatie, omdat het bedrijf niet eerst had nagedacht over de capaciteiten die het team nodig had om de doelstellingen van het testproces te bereiken.

Dit leidde tot een aantal problemen, zoals:

- Het team had moeite om de tool te gebruiken,
- De tool functioneerde niet naar verwachting,
- De tool was te duur.

### **Maar wat is nu precies test tool management en wat heeft het voor doel?**

In de volgende paragrafen gaan we hier dieper op in.

## Test tool management

Test tool management is een proces dat helpt om de juiste test tools te kiezen, te implementeren en te beheren om de efficiëntie, effectiviteit en kwaliteit van het testproces te verbeteren en daarmee ook de kosten te reduceren door o.a. verhoogde testproductiviteit en makkelijke toegankelijkheid van de tool.

### Doel van test tool management

Het doel van test tool management is om ontwikkel/DevOps-teams tijd, geld en moeite te laten besparen. Dit omvat niet alleen het beheer van de tools zelf, maar ook het bieden van ondersteuning en training aan de gebruikers die testwerkzaamheden uitvoeren. Dit geldt ook voor business analisten die betrokken zijn bij het opstellen van use cases en specificaties. Op deze manier wordt ervoor gezorgd dat de beheerde tools snel en optimaal kunnen worden gebruikt.

## De rol van test tools

Test tools spelen een onmisbare rol bij het uitvoeren en ondersteunen van verschillende testactiviteiten. Deze activiteiten omvatten de vier belangrijke aspecten van testen: Test Controle, Test Ontwerp, Test Uitvoering en Test Omgeving. De tools variëren van geautomatiseerde testhulpmiddelen tot testmanagementsoftware en bevindingenbeheer-systemen.

Het gebruik van de juiste test tools kan de volgende voordelen opleveren:

- **Efficiëntie:** Test tools kunnen worden gebruikt om testtaken te automatiseren, zoals het uitvoeren van testgevallen en het analyseren van testresultaten.
- **Effectiviteit:** Test tools kunnen worden gebruikt om meer gedetailleerde en geavanceerde tests uit te voeren, zoals het testen van randgevallen en het uitvoeren van stresstests.
- **Kwaliteit:** Test tools kunnen worden gebruikt om de dekking van testgevallen te verbeteren, zoals het testen van alle belangrijke functies en het identificeren van fouten.

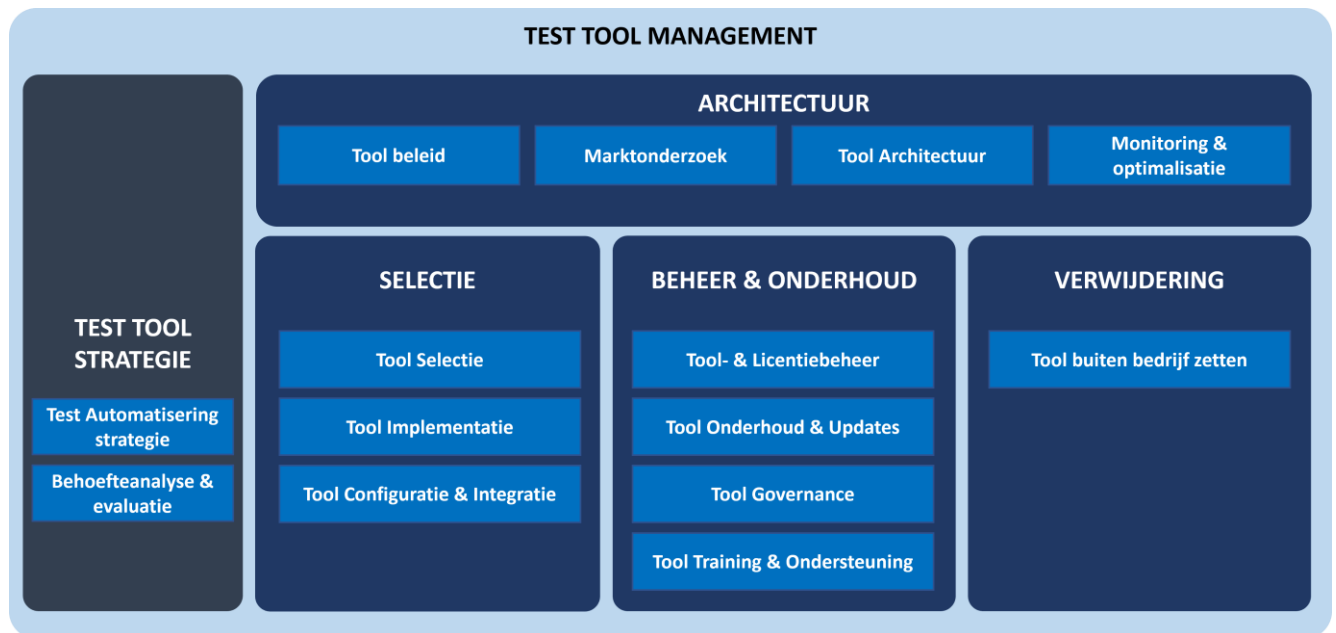
Voorbeelden van hoe test tools worden gebruikt in de praktijk, gerelateerd aan de vier aspecten van testen, zoals weergegeven in Figuur 2, zijn:

- **Testautomatisering [Testuitvoering]:** Een bedrijf gebruikt een testautomatiseringstool om automatisch testgevallen uit te voeren voor zowel nieuwe als aangepaste softwareapplicaties. Dit maakt het duidelijk dat de tool nuttig is in verschillende scenario's en helpt het bedrijf om de testefficiëntie te verbeteren en snel inzicht te geven in de mate van kwaliteit van het softwareproduct
- **Testdata generator [Testomgeving]:** Testdata generators kunnen voor organisaties die nieuwe applicaties testen, afhankelijkheden reduceren door testgegevens te genereren die rekening houden met deze afhankelijkheden. Dit kan de efficiëntie en de kwaliteit van de tests verbeteren.
- **Testmanagement [Test Controle]:** Een bedrijf gebruikt een testmanagementtool om alle testactiviteiten te beheren, van het plannen en uitvoeren van testgevallen tot het analyseren van testresultaten. Dit helpt het bedrijf bij het creëren van een consistent en efficiënt testproces.
- **Testomgevingsbeheer [Testomgeving]:** Een bedrijf heeft een beheerteam die gebruikt maakt van een testomgevingsbeheertool om de testomgevingen te beheren en te automatiseren. Dit helpt het bedrijf om testomgevingen consistent te maken. Het helpt ook om testomgevingen efficiënt te maken. Dit verbetert de testefficiëntie.
- **CI/CD (continuous integration/continuous delivery) [Testuitvoering]:** Een bedrijf gebruikt een test tool om geautomatiseerde testscripts uit te voeren in een CI/CD-pipeline. Dit helpt het bedrijf om testautomatisering te integreren met de ontwikkelingscyclus, waardoor de testefficiëntie wordt verbeterd en de zichtbaarheid van de softwarekwaliteit wordt vergroot.
- **Service virtualization [Testomgeving]:** Een bedrijf gebruikt een service virtualization-tool om afhankelijkheden te simuleren. In deze context betekent simuleren het nabootsen van een service. De service virtualization-tool creëert een nabootsing van een afhankelijk systeem. Dit heeft diverse voordelen, waaronder de mogelijkheid om tests uit te voeren onafhankelijk van de beschikbaarheid van andere systemen, de aanwezigheid van testdata en de snelheid van deze systemen.



## Test tool managementprocessen

Test tool management bestaat uit de volgende fasen en processen (zie figuur 4):



Figuur 4 - Test tool managementmodel

In de volgende paragrafen wordt per fase een uitgebreidere uitleg gegeven over de verschillende proces onderdelen.

### Test Tool Strategie

De test tool strategie is een set strategische keuzes die DevOps-teams binnen de organisatie maken met betrekking tot de test tools die ze willen gebruiken. Het is belangrijk om te vermelden dat de test tool strategie ook rekening houdt met de vier categorieën van “test tool capabilities” zoals eerder beschreven. Deze categorieën helpen teams bij het selecteren van de juiste tools die passen bij hun specifieke behoeften en doelstellingen. De strategie is gebaseerd op de test automatiseringsstrategie en de behoefteanalyse & evaluatie.

1. **Test automatiseringsstrategie:** De test automatiseringsstrategie vormt de basis voor een test tool strategie. Dit proces omvat het:
  - Identificeren van de tests die geautomatiseerd moeten worden,
  - Bepalen van de benodigde middelen,
  - Opstellen van een implementatieplan.Deze stap in het proces kan worden overgeslagen als de tests niet worden geautomatiseerd.
2. **Behoefteanalyse & evaluatie:** Zodra de test automatiseringsstrategie duidelijk is of niet nodig is omdat er niet aan testautomatisering wordt gedaan, kan het team nadenken over de specifieke criteria waaraan de test tools moeten voldoen. Het team of de afdeling die over het beheer van de test tools gaat, moet toezien op de evaluatie van de bestaande test tools. Ze moeten beoordelen of deze tools nog steeds voldoen aan de huidige behoeften. Daarnaast moeten ze eventuele lacunes identificeren die moeten worden aangepakt. Door deze analyse zorgvuldig uit te voeren, kan de organisatie efficiënter en effectiever reageren op de unieke eisen van elk project.

### Architectuur

Deze fase richt zich op hoe de organisatie omgaat met test tooling. Het omvat de volgende processtappen:

1. **Test tool beleid:** Stel duidelijke beleidlijnen en normen op voor het gebruik van test tools binnen de organisatie. Stem dit eventueel af met afdeling inkoop en definieer specifieke criteria waaraan test tools moeten voldoen om goedgekeurd te worden voor intern gebruik. Dit helpt bij het voorkomen van wildgroei van tools en zorgt ervoor dat alleen goedgekeurde en geschikte tools worden gebruikt. Het waarborgt dat de test tools bijdragen aan de algehele kwaliteit en effectiviteit van het testproces.

2. **Marktonderzoek:** Voer regelmatig marktonderzoek uit om op de hoogte te blijven van nieuwe en opkomende test tools en technologieën, door vakbeurzen te bezoeken zoals de TestNet evenementen. Evalueer hoe deze nieuwe tools kunnen bijdragen aan de teststrategie van de organisatie. Dit stelt de organisatie in staat om proactief in te spelen op veranderingen in de markt en te onderzoeken hoe deze nieuwe tools kunnen bijdragen aan de test tool strategie.
3. **Tool-architectuur:** Ontwerp een architectuur voor het test tool-ecosysteem dat de behoeften van verschillende teams en projecten adresseert. Definieer hoe de test tools zullen samenwerken en gegevens zullen delen binnen de organisatie.
4. **Continue monitoring en optimalisatie:** Blijf het gebruik van test tools binnen de organisatie monitoren en verzamel feedback van gebruikers. Optimaliseer de toolset op basis van veranderende behoeften en technologische ontwikkelingen. Dit proces maximaliseert de waarde die test tools aan de organisatie toevoegen, wat niet alleen grip geeft op de kosten (project/beheer/licentie) van test tools maar ook bijdrage levert aan de algemene efficiëntie van het testproces.

Fase architectuur heeft raakvlakken met de overige fasen.

## Selectie

In deze fase wordt de behoefte aan test tools geïdentificeerd en getoetst. De volgende processtappen zijn hierbij van belang:

1. **Toolselectie [3]:** Het identificeren, evalueren en selecteren van geschikte test tools op basis van de behoeften en vereisten van het specifieke project. Dit is een belangrijke factor in het verbeteren van de volwassenheid van testprocessen. Door de juiste tools te selecteren, kunnen organisaties de efficiëntie en effectiviteit van het testproces vergroten, wat kan leiden tot een hogere kwaliteit van de software. Toolselectie kent vier stappen:
  - 1) **Identificeer de vereisten voor tools:** Duidelijk aangeven waarvoor de test tool nodig is en waaraan deze moet voldoen. Bijvoorbeeld, als een tool nodig is voor geautomatiseerd testen, moet worden gespecificeerd welke soorten tests moeten worden geautomatiseerd (zoals unit tests, integratietests, enz.), welke programmeertalen de tool moet ondersteunen, en andere specifieke functies die nodig zijn.
  - 2) **Evalueer de tools en leveranciers:** Nadat de vereisten zijn geïdentificeerd, moeten de verschillende tools en hun leveranciers worden geëvalueerd om te zien welke het beste aan de behoeften voldoen. Dit kan zijn het lezen van productrecensies, het uitvoeren van proefversies van de tools, en het voeren van gesprekken met de vertegenwoordigers van de testtoolleveranciers.
  - 3) **Schat de kosten en baten in:** Weeg de kosten van de aanschaf en het onderhoud van de tool af, tegen de voordelen die het zal bieden. Denk hierbij aan licentie en onderhoudskosten en de tijd en middelen die nodig zijn om het team op te leiden om de tool te gebruiken.
  - 4) **Neem de uiteindelijke beslissing:** Na het doorlopen van de bovenstaande stappen, zou er voldoende informatie moeten zijn om een geïnformeerde beslissing te nemen over welke tool het beste past bij de behoeften van het team en het beschikbare budget. Het is belangrijk om te onthouden dat geen enkele tool perfect is, dus het is mogelijk dat er compromissen moeten worden gesloten. Maar met zorgvuldige overweging, zou een tool moeten kunnen worden gekozen die het meest geschikt is voor het team.
2. **Toolimplementatie:** Het installeren, configureren en in gebruik nemen van de geselecteerde test tools maakt deel uit van dit proces. Dit zorgt ervoor dat de geselecteerde tools volledig worden geïntegreerd in het softwareontwikkelingsproces. Daarnaast omvat dit proces ook het inrichten van het beheer van deze tools, waardoor het team de test tools optimaal kan benutten en het testproces efficiënter en effectiever verloopt.
3. **Toolconfiguratie en integratie:** Het aanpassen en configureren van de test tools om aan de behoeften van het specifieke project te voldoen en ze te integreren met andere ontwikkelings- en operationele tools. Deze processtap helpt bij in het minimaliseren van de kosten en het optimaal benutten van de tools zonder extra investeringen. Het stelt het team in staat om een naadloze samenwerking tussen verschillende tools en processen te creëren.

## Beheer & Onderhoud

In deze fase worden de test tools beheerd om ervoor te zorgen dat ze correct werken en worden gebruikt. Processtappen die hierbij van belang zijn:

1. **Tool- en licentiebeheer:** Het beheren van toolassets, inclusief registratie, beschikbaarheid en gebruik, evenals het beheren van licenties en naleving van licentieovereenkomsten. Als dit proces goed is ingericht, zorgt dat ervoor dat de tools beschikbaar zijn wanneer dat nodig is, zonder verspilling van middelen of licentie-inbreuken, en waarborgt naleving van licentieovereenkomsten. Tevens kan hiermee ook het daadwerkelijk gebruik worden gemeten (bijvoorbeeld hoeveel mensen hoe vaak inloggen) om de business case te blijven bewaken.
2. **Toolonderhoud en updates:** Het regelmatig onderhouden, updaten en upgraden van geïmplementeerde test tools om de functionaliteit en prestaties te optimaliseren.
3. **Tool Governance:** Het proces van het beheren van test tools, inclusief de processtappen, aspecten en strategieën die nodig zijn om test tools effectief te gebruiken en te onderhouden. Een goed tool governance-beleid biedt duidelijke richtlijnen en regels, waardoor de consistentie en kwaliteit van het testproces worden gewaarborgd en het test tool management efficiënter verloopt.
4. **Tooltraining en ondersteuning:** Waarbij tooltraining het proces is van het geven van training aan teams en andere gebruikers over hoe test tools te gebruiken, is toolondersteuning het proces van het bieden van ondersteuning aan teams en andere gebruikers bij het gebruik van test tools. Tooltraining en ondersteuning is een belangrijk onderdeel van het proces om test tools volledig te benutten. Door gebruikers van test tools te (blijven) trainen en te ondersteunen, kunnen organisaties ervoor zorgen dat ze de test tools op een effectieve manier kunnen gebruiken.

## Verwijdering

In deze fase worden test tools verwijderd of buiten werking gesteld die niet langer nodig zijn.

1. **Tool buiten bedrijf zetten:** Het gecontroleerd verwijderen of buiten werking stellen van test tools die niet langer nodig zijn of niet meer voldoen aan de vereisten. Het effectief opruimen van tools zorgt voor een opgeruimde en actuele toolset, verhoogt de operationele efficiëntie en minimaliseert onnodige overhead, los van de besparing op licentie kosten.

### MobyIT

Als MobyIT test tool management (TTM) had geïmplementeerd dan had MobyIT de huidige knelpunten als volgt kunnen voorkomen:

#### Test Tool Strategie:

- Test automatiseringsstrategie: Definiëren van doelstellingen en vereisten voor test tools in de context van testautomatisering. Dit helpt bij het bepalen van de functionaliteit en het gewenste automatiseringsniveau van de tool.
- Behoeftanalyse & evaluatie: Een grondige analyse van de testbehoeften van de "SmartConnect"-app had de functionele en technische vereisten voor de testtool gedefinieerd. Dit helpt bij het selecteren van een tool die perfect aansluit bij de behoeften van de DevOps team.

#### Architectuur:

- Test tool beleid: Vaststellen van duidelijke criteria voor toolselectie en -gebruik, met aandacht voor complexiteit, trainingsbehoeften en ondersteuningsmogelijkheden. Dit helpt bij het kiezen van een tool die past bij de vaardigheden van het team en de infrastructuur van de organisatie.
- Marktonderzoek: Continue analyse van nieuwe trends en technologieën in TTM om op de hoogte te blijven van alternatieve tools. Dit zorgt ervoor dat MobyIT de best passende tool voor hun project kan kiezen.

#### Selectie:

- Tool selectie: Grondige evaluatie van tools op basis van de gedefinieerde criteria, inclusief gebruiksvriendelijkheid, functionaliteit en ondersteuningsniveau. Dit helpt bij het vinden van de tool die de beste waarde biedt voor de investering.
- Tool implementatie: Gefaseerde implementatie met uitgebreide training en ondersteuning om vertragingen te minimaliseren en teamvertrouwdheid te bevorderen. Dit zorgt voor een soepele integratie van de tool in het testproces.

### **Beheer & Onderhoud:**

- Tool- en licentiebeheer: Nauwkeurige budgetbewaking en licentiebeheer om onverwachte kosten te voorkomen. Dit zorgt voor een efficiënt gebruik van middelen.
- Tool onderhoud en updates: Regelmatige updates en onderhoud om bugs en compatibiliteitsproblemen te minimaliseren. Dit zorgt voor een optimale werking van de tool.

### **Verwijdering:**

- Tool buiten bedrijf stellen: Ontwikkeling van een exit strategie voor het gecontroleerd deactiveren van ongebruikte tools en het vergemakkelijken van de overstap naar nieuwe tools. Dit zorgt voor een geordende afronding van de tool lifecycle.

Door deze tips te volgen, kan MobyIT een cultuur van voortdurende verbetering creëren op het gebied van test tool management, wat de algehele efficiëntie en kwaliteit van hun softwareontwikkelingsproces zal verhogen.

De fasen Architectuur, Selectie, Beheer & Onderhouden en Verwijdering zijn toebedeeld aan de tool eigenaars en beheerders. De fase Test Tool Strategie is toebedeeld aan de tool gebruikers.

## **Rollen en verantwoordelijkheden**

Het definiëren van rollen en verantwoordelijkheden is cruciaal in elk proces, inclusief test tool management, om de volgende redenen:

- **Verantwoordelijkheid:** Het zorgt ervoor dat elke taak of activiteit een 'eigenaar' heeft, wat betekent dat er iemand is die verantwoordelijk is voor het voltooien van die taak of activiteit.
- **Efficiëntie:** Het helpt bij het stroomlijnen van processen en het verbeteren van de efficiëntie, omdat iedereen weet wat hun taken zijn en wat er van hen wordt verwacht.
- **Communicatie:** Het verbetert de communicatie en samenwerking binnen het team. Als iedereen zijn rol kent, weten ze ook met wie ze moeten communiceren over specifieke kwesties of problemen.
- **Verantwoording:** Het bevordert verantwoording omdat elke rol specifieke taken en verantwoordelijkheden heeft. Dit betekent dat als er iets misgaat, het gemakkelijker is om te bepalen waar het probleem ligt.
- **Kwaliteitscontrole:** Het helpt bij het handhaven van de kwaliteitsnormen. Als iedereen zijn rol begrijpt en de bijbehorende taken en verantwoordelijkheden kent, is de kans groter dat het werk correct wordt uitgevoerd.

Voor het toepassen van test tool management kennen we drie rollen:

### **Test tool eigenaar**

De test tool eigenaar is verantwoordelijk voor zowel het strategisch als tactisch beheer van test tools, inclusief Lifecycle Management. Dit betekent dat de test tool eigenaar niet alleen beslissingen neemt en plannen maakt voor de lange termijn (strategisch beheer), maar ook voor de korte termijn (tactisch beheer), en dat zij de gehele levenscyclus van de test tools beheren, van selectie en aanschaf, tot implementatie, gebruik, onderhoud, en uiteindelijk vervanging of verwijdering.

In traditionele organisaties is de test tool eigenaar vaak een teammanager van een (test) afdeling. Deze rol kan ook worden vervuld door een senior testengineer of een testarchitect.

In Agile organisaties is de test tool eigenaar vaak een lid van de staforganisatie. Deze rol kan ook worden vervuld door een DevOps engineer of een product owner.

De test tool eigenaar is verantwoordelijk voor het:

- Beheren van het test toolbudget,
- Aanschaffen en verwijderen van test tools,
- Definiëren van een beleid en criteria voor het gebruik van test tools,
- Onderhouden van relaties met de leveranciers van test tools,
- Onderhouden van relaties met de test tool beheerders en test tool gebruikers,
- Rapporteren over de resultaten van het test tool managementproces,

- Definiëren van de behoeften en vereisten voor test tooling,
- Overzien van de architectuur voor het test tool-ecosysteem,
- Bepalen van de criteria waaraan test tools moeten voldoen om goedgekeurd te worden voor intern gebruik.

#### **MobyIT**

In de situatie van MobyIT zou een test tool eigenaar kunnen helpen om de volgende problemen te voorkomen of te verhelpen:

- Gebrek aan overeenstemming over de behoeften van het testproces: De test tool eigenaar kan helpen om de behoeften van het testproces te definiëren en te communiceren met alle betrokken partijen.
- Onvoldoende kennis en ervaring met de test tool: De test tool eigenaar kan de teams helpen om de test tool te leren gebruiken en te beheersen.
- Gebrek aan ondersteuning bij het gebruik van de test tool: De test tool eigenaar kan ondersteuning bieden aan de teams bij het gebruik van de test tool.

#### **Test tool beheerder**

De test tool beheerder is verantwoordelijk voor het tactisch beheer van de test tools. In traditionele organisaties is de test tool beheerder vaak een testengineer of een senior testengineer. Deze rol kan ook worden vervuld door een testmanager of een testarchitect.

In Agile organisaties is de test tool beheerder vaak een lid van de staforganisatie. Deze rol kan ook worden vervuld door een DevOps engineer of een product owner.

De test tool beheerder is verantwoordelijk voor het:

- Ontwikkelen en implementeren van een beleid voor het gebruik van test tools binnen de organisatie,
- Selecteren van de juiste tools voor de behoeften van de organisatie,
- Onderhouden van relaties met de test tool eigenaar en test tool gebruikers,
- Implementeren van de test tools en deze configureren voor gebruik,
- Monitoren van de prestaties van de tools en waar nodig maatregelen nemen,
- Bieden van ondersteuning aan de gebruikers,
- Het maken, beschikbaar stellen en bijhouden van trainingen en gebruikershandleidingen,
- Zorgen voor training en ondersteuning (bijvoorbeeld het organiseren van trainingssessies of workshops),
- Implementeren van de architectuur voor het test tool-ecosysteem,
- Integreëren van test tools met andere ontwikkelings- en operationele tools,
- Continue verbetering van het test tool managementproces.

#### **MobyIT**

In de specifieke situatie rondom MobyIT zou de test tool beheerder betrokken zijn bij het oplossen van de technische problemen die het productontwikkelingsteam ondervond met TestQuestPro. De test tool beheerder zou ook helpen bij het coördineren van de training en ondersteuning die het team nodig heeft om de tool effectiever te gebruiken. Samen met de test tool eigenaar zouden ze moeten beslissen of het nodig is om terug te keren naar de vorige test tool en een meer kosteneffectief en efficiënt test tool management te implementeren.

#### **Test tool gebruiker**

De test tool gebruiker is verantwoordelijk voor de tools die in het team gebruikt worden om tests uit te voeren. Hij/zij is verder verantwoordelijk voor het:

- Leren, voortdurende ontwikkeling en begeleiding bij het gebruik van tools,
- Rapporteren van bevindingen over de tools aan de beheerder,
- Opstellen van een test tool strategie voor het (DevOps) team,
- Definiëren van de behoeften en vereisten voor test tooling,
- Feedback geven over de ervaringen die met de test tool zijn verkregen.

## MobyIT

In de bovengenoemde scenario's rondom MobyIT zouden test tool gebruikers de volgende acties kunnen hebben ondernomen om de problemen te voorkomen:

- Ze zouden kunnen deelnemen aan het proces van het selecteren en implementeren van TestQuestPro. Dit zou hen hebben geholpen om de tool beter te begrijpen en om ervoor te zorgen dat de tool aan hun behoeften voldeed.
- Ze zouden kunnen vragen om training en ondersteuning bij het gebruik van TestQuestPro. Dit zou hen hebben geholpen om de tool effectiever te gebruiken en om de problemen met de tool te voorkomen.
- Ze zouden feedback kunnen geven over de problemen met TestQuestPro. Dit zou de test tool eigenaar en de test tool beheerder hebben geholpen om de problemen te identificeren en op te lossen.

De rollen en verantwoordelijkheden van de test tool eigenaar, beheerder en gebruiker zijn verschillend maar onderling verbonden. De test tool eigenaar speelt een cruciale rol bij het definiëren van toolbeleid en het beheren van het gehele succes van test tool management. De beheerder is primair verantwoordelijk voor het waarborgen dat de tools beschikbaar zijn en correct functioneren, terwijl de gebruiker de taak heeft om de tools effectief te gebruiken om te voldoen aan de testbehoeften van de organisatie. Een effectieve aanpak om de samenwerking tussen de test tool eigenaar, beheerder en gebruiker te bevorderen, is het oprichten van een team of community met vertegenwoordigers uit elke groep. Dit samenwerkende team of community kan gezamenlijk de test tool managementprocessen evalueren en verbeteren.

## Het belang van test tool management

Test tool management speelt een cruciale rol bij het waarborgen van het gebruik van de juiste test tools en hun effectieve integratie in het testproces. Dit proces omvat het identificeren van de specifieke testbehoeften van het project, het zorgvuldig evalueren van beschikbare tools, het selecteren van de meest geschikte opties, het vakkundig configureren en integreren van de gekozen tools, het grondig trainen van het team en het zorgvuldig onderhouden en updaten van de tools gedurende de levensduur van de softwareapplicatie die moet worden getest. Met name in omgevingen waar continue integratie en continue levering (CI/CD) worden toegepast, is het gebruik van test tooling randvoorwaardelijk. Een goed ingericht test tool management draagt bij aan een geoptimaliseerd testproces dat de kwaliteit van softwareproducten en het succes van projecten bevordert.

## De voordelen van test tool management

Test tool management is belangrijk omdat het een aantal voordelen biedt, waaronder:

- **Verhoogde testefficiëntie:** Test tool management kan helpen om het testproces te automatiseren en te optimaliseren. Dit kan leiden tot een vermindering van de testtijd en kosten, en een verbetering van de testkwaliteit. Bijvoorbeeld, een team gebruikt een geautomatiseerde test tool om automatisch een reeks testgevallen uit te voeren voor een nieuwe softwareapplicatie. Dit helpt het team om de testtijd te verminderen en de testkwaliteit te verbeteren.
- **Verbeterde nauwkeurigheid en grondigheid:** Test tool management stelt teams in staat om niet-functionele requirements zoals prestaties, beveiliging en bruikbaarheid grondig en nauwkeurig te testen. Bijvoorbeeld, bij het uitvoeren van performance tests, kunnen tools duizenden virtuele gebruikers simuleren om de reactietijd, doorvoer en betrouwbaarheid van een systeem onder zware belasting te meten. Dit zou bijna onmogelijk zijn om handmatig te doen. Evenzo kunnen security test tools automatisch en nauwkeurig veelvoorkomende beveiligingsproblemen identificeren. Zonder deze tools zouden dergelijke tests tijdrovend zijn en mogelijk niet alle potentiële problemen aan het licht brengen.
- **Verhoogde testdekking:** Test tools kunnen worden gebruikt om testgevallen automatisch uit te voeren, waardoor teams zich kunnen concentreren op meer complexe taken, zoals het analyseren van testresultaten. Test tools kunnen ook worden gebruikt om testgevallen te genereren op basis van verschillende criteria, zoals de functionaliteit van het softwareproduct, de risico's of de eisen van de gebruikers. Bijvoorbeeld, een team gebruikt een testautomatiseringstool om testgevallen te genereren voor een nieuwe softwareapplicatie. De tool genereert testgevallen op basis van de functionaliteit van de app, de risico's die zijn geïdentificeerd door het team en de eisen van de gebruikers. Dit helpt het team om de testdekking te verbeteren en fouten eerder op te sporen.
- **Verhoogde testkwaliteit en effectief testprocesbeheer:** Test tools kunnen worden gebruikt om

testgegevens te genereren die realistisch en consistent zijn, waardoor de nauwkeurigheid van de testresultaten wordt vergroot. Dit helpt het team om fouten eerder op te sporen, wat uiteindelijk de kwaliteit van het softwareproduct verbetert. Bovendien kunnen test tools bijdragen aan een effectieve beheersing van het testproces, inclusief het volgen van de voortgang, het bewaken van de kwaliteit, en het beheren van bevindingen. Dit leidt tot een efficiënter en effectiever testproces.

- **Verbeterde rapportage en communicatie:** Veel test tools bieden ingebouwde rapportagefuncties die het gemakkelijk maken om testresultaten te delen en te communiceren met het team en belanghebbenden. Bijvoorbeeld, een team gebruikt een test tool om testresultaten te rapporteren in een visueel aantrekkelijke en gemakkelijk te begrijpen format. Dit helpt het team om de testresultaten snel en gemakkelijk te begrijpen en te communiceren met de belanghebbenden, zoals ontwikkelaars, productmanagers en klanten.
- **Betere samenwerking en teamwerk:** Test tools kunnen helpen bij het verbeteren van de samenwerking binnen het team door het mogelijk te maken om testgevallen, testgegevens en testresultaten te delen. Bijvoorbeeld, een team gebruikt een test tool om testgegevens te delen met verschillende testers, ontwikkelaars en productmanagers. Dit helpt de teams om samen te werken en te coördineren om de kwaliteit van het softwareproduct te verbeteren.
- **Kostenbesparingen:** Hoewel de initiële kosten voor de aanschaf en implementatie van test tools aanzienlijk kunnen zijn, kunnen ze op de lange termijn kosten besparen door de efficiëntie van het testproces te verbeteren. Bijvoorbeeld, een team gebruikt een geautomatiseerde test tool om performance tests uit te voeren. Dit helpt het team om de testtijd te verkorten en de kosten van de test te verminderen.

Deze voordelen kunnen een aanzienlijke impact hebben op de algehele kwaliteit en het succes van het softwareontwikkelingsproces. Het is echter belangrijk om te onthouden dat de effectieve implementatie en het beheer van test tools essentieel zijn om deze voordelen te realiseren.

## De uitdagingen van test tool management

Hoewel test tool management een waardevolle methode is, zijn er enkele uitdagingen die bedrijven kunnen tegenkomen bij het succesvol implementeren ervan. Deze uitdagingen omvatten:

- **Complexiteit van test tools:** Test tools kunnen vaak complex zijn en vereisen technische expertise om ze effectief te kunnen gebruiken. Het ontbreken van diepgaande kennis kan leiden tot inefficiënt gebruik en beperkte voordelen van de tool. Bijvoorbeeld, het gebruik van een geavanceerde geautomatiseerde test tool vereist diepgaande programmeerkennis, waardoor testers mogelijk niet in staat zijn om alle functies optimaal te benutten.
- **Gebrek aan kennis en richtlijnen:** Het kiezen van test tools zonder kennis en richtlijnen kan leiden tot slechte softwarekwaliteit en verspilling van middelen. Organisaties moeten hun teams informeren over de test tools die beschikbaar zijn en hen helpen de beste tools te selecteren voor hun softwareprojecten. Dit kan de testresultaten, de productiviteit en de klanttevredenheid verbeteren.
- **Beperkte middelen:** Kleinere bedrijven of teams met beperkte middelen kunnen moeite hebben om te investeren in dure test tools of training voor hun personeel, waardoor hun testprocessen minder effectief worden. Bijvoorbeeld, een klein softwarebedrijf kan zich geen dure test tool veroorloven en moet zich beperken tot gratis tools met beperkte functionaliteit.
- **Weerstand tegen verandering:** Het implementeren van nieuwe test tools kan weerstand opwekken bij het team, vooral als ze al gewend zijn aan andere tools of werkwijzen. Het omgaan met veranderingsmanagement is cruciaal voor succes. Bijvoorbeeld, een team dat al jaren dezelfde tool gebruikt, kan aarzelen om over te stappen naar een nieuwe tool, waardoor het potentieel van een efficiënter proces wordt belemmerd.
- **Geen duidelijke test tool strategie:** Het ontbreken van een duidelijke test tool strategie kan leiden tot willekeurige selectie en implementatie van tools, wat de efficiëntie en effectiviteit van het testproces kan ondermijnen. Bijvoorbeeld, verschillende teams binnen een organisatie gebruiken inconsistente tools zonder duidelijke richtlijnen, wat leidt tot fragmentatie en inefficiëntie.
- **Onvoldoende training en ondersteuning:** Het gebrek aan voldoende training en continue ondersteuning kan de leercurve vergroten en het vermogen van het team om de tools effectief te gebruiken belemmeren. Bijvoorbeeld, na de implementatie van een nieuwe test tool ontvangen de testers geen formele training, waardoor ze niet het volledige potentieel van de tool kunnen benutten en het

testproces vertraging oploopt.

- **Test Tool Dekkingsgebied:** Het gebruik van software of meerdere applicaties binnen een organisatie kan ertoe leiden dat er verschillende tools worden geselecteerd en gebruikt voor diverse testaspecten. Dit kan inefficiënt en kostbaar zijn. Daarom is het belangrijk om periodiek een evaluatie uit te voeren van de gebruikte test tools om te bepalen of ze nog steeds geschikt zijn voor het gestelde doel en kosteneffectief zijn.
- **Verstoring van bestaande processen:** De implementatie van test tool management kan bestaande processen verstoren en kan weerstand oproepen bij teamleden die gewend zijn aan bepaalde manieren van werken. Bijvoorbeeld, Een team gebruikt een traditionele, manuele testmethode. De implementatie van een nieuwe geautomatiseerde test tool vereist een verandering in de manier waarop tests worden uitgevoerd. Dit kan leiden tot weerstand bij het team.
- **Leercurve:** Er kan een aanzienlijke leercurve zijn voor teamleden om nieuwe tools te leren gebruiken, wat kan leiden tot een tijdelijke daling van de productiviteit. Bijvoorbeeld, Een team implementeert een nieuwe geautomatiseerde test tool. De testers moeten leren hoe ze de tool gebruiken en hoe ze tests kunnen automatiseren. Dit kan leiden tot een tijdelijke daling van de testproductiviteit.
- **Niet altijd de perfecte pasvorm:** Soms voldoen de beschikbare tools niet perfect aan de specifieke behoeften van een project of organisatie. Dit kan leiden tot compromissen in hoe tests worden uitgevoerd. Bijvoorbeeld, Een team implementeert een nieuwe geautomatiseerde test tool. De tool is niet geschikt voor alle soorten tests die het team moet uitvoeren. Dit kan leiden tot het gebruik van suboptimale testmethoden.
- **Te veel vertrouwen op tools:** Er is een risico dat teams te afhankelijk worden van tools, waardoor ze mogelijk minder flexibel zijn in hun testprocessen en minder in staat zijn om te reageren op veranderingen. Bijvoorbeeld, Een team vertrouwt te veel op een geautomatiseerde test tool. Dit kan leiden tot het feit dat ze niet voldoende aandacht besteden aan manuele tests, wat kan leiden tot het missen van fouten.

Het is belangrijk om deze mogelijke uitdagingen in overweging te nemen bij het beslissen over de implementatie van test tool management. Het is essentieel om een evenwicht te vinden tussen de voordelen van automatisering en de mogelijke uitdagingen die het met zich meebrengt.

## Test tool management implementeren?

Het implementeren van test tool management kan een complex proces zijn, maar er zijn verschillende stappen die kan worden gevolgd om het proces soepeler te laten verlopen:

- **Inventariseer uw huidige tools:** Begin met een grondige inventarisatie van de tools die u momenteel beheert. Begrijp hun functionaliteiten, sterktes en zwaktes, en hoe ze worden gebruikt in uw huidige testprocessen.
- **Identificeer verbeterpunten:** Kijk naar de huidige uitdagingen of tekortkomingen in uw testprocessen. Zijn er aspecten die kunnen worden verbeterd met beter test tool management? Dit kan variëren van het verbeteren van de efficiëntie tot het verhogen van de testdekking of het verbeteren van de rapportage.
- **Ontwikkel een strategie:** Op basis van uw inventarisatie en identificatie van verbeterpunten, ontwikkel een strategie voor hoe u uw test tool management wilt verbeteren. Dit kan het optimaliseren van het gebruik van bestaande tools, het trainen van uw team om de tools effectiever te gebruiken, of het implementeren van nieuwe tools omvatten.
- **Implementeer de strategie:** Voer uw strategie uit en monitor de resultaten. Dit kan het implementeren van nieuwe processen, het geven van trainingen, of het maken van aanpassingen aan uw tools omvatten.
- **Evalueer en verbeter:** Na de implementatie, evalueer de effectiviteit van uw nieuwe test tool management aanpak. Gebruik feedback van uw team en meetbare resultaten om te bepalen of uw aanpak succesvol is en waar verdere verbeteringen kunnen worden aangebracht.

Daarnaast kan de volgende strategie worden toegepast:

In organisaties waar test tooling worden beheerd zijn test tool management-processen in de meeste gevallen impliciet aanwezig. Door het test tool managementmodel te implementeren worden ze expliciet gemaakt. Hierbij zal ook duidelijk worden welke fasen en processen ontbreken. Voor het verbeteren of implementeren van het test tool management-model kan het “people, process and technology” (PPT)-framework in



combinatie met het TMMi model worden toegepast [7]. Het PPT-framework kan ook de samenwerking en tevredenheid onder degenen die betrokken zijn bij het testen vergroten en innovatie en ontwikkeling van de testtools stimuleren.

### Test tool classificatiesysteem (of Taxonomie):

Een test tool classificatiesysteem helpt ons om test tools te sorteren en te vergelijken. Het geeft inzicht in de verschillende kenmerken van de tools, zoals wat ze doen, hoe goed ze werken, en of ze goed samenwerken met andere systemen. Bovendien helpt dit systeem ons ook om de beste tools te kiezen voor onze specifieke behoeften. Daarnaast gebruiken we een **multicriteria selectiemethode** [1][2] die verschillende criteria overweegt om de tools te rangschikken en te selecteren. Deze methode houdt ook rekening met wat het meest belangrijk is voor de mensen die de beslissingen nemen.

Het test tool classificatiesysteem en de multicriteria selectiemethode zijn samen een handige manier om de beste en voor een organisatie best passende test tools te kiezen. Ze helpen ook om de test tools die beschikbaar zijn te verbeteren.

Gezien het grote aantal test tools op de markt, zou een applicatie die deze classificatie- en selectiemethoden kan automatiseren zeer wenselijk zijn. Een dergelijke applicatie zou het proces van het kiezen van de juiste tools aanzienlijk kunnen vereenvoudigen en versnellen.

### Test tool classificatiesysteem in test tool management:

- **Classificatie van tools:** We gebruiken het classificatiesysteem om onze test tools op een georganiseerde manier te sorteren. Dit helpt ons om onze tools beter te begrijpen en te zien hoe ze met elkaar samenwerken.
- **Selectie van tools:** We gebruiken het classificatiesysteem en de *multicriteria selectiemethoden* om de beste tools voor onze behoeften te kiezen. Dit zorgt ervoor dat we de juiste tools voor onze taken hebben.
- **Evaluatie van tools:** We gebruiken het classificatiesysteem om te kijken hoe goed onze tools werken. Dit helpt ons om te zien waar we verbeteringen kunnen aanbrengen.
- **Ontwikkeling van tools:** De inzichten die we krijgen van het classificatiesysteem en de selectiemethoden helpen ons om de test tools die beschikbaar zijn te verbeteren.

Een **multicriteria selectiemethode** verwijst naar de procedure of het proces dat wordt gebruikt om iets te kiezen uit een groep mogelijkheden. In de context van Test Tool Management, verwijst de selectiemethode naar het proces van het kiezen van de meest geschikte test tools uit een reeks beschikbare opties. Deze methode kan verschillende criteria omvatten, zoals de functionaliteit van de tool, de prestaties, de compatibiliteit met andere systemen, enzovoort. Deze criteria worden vervolgens gewogen op basis van hun relatieve belangrijkheid voor de specifieke behoeften en eisen van de software. Het doel van de selectiemethode is om een objectieve en geautomatiseerde manier te bieden om de test tools te rangschikken en te selecteren. Dit helpt bij het waarborgen, zodat de meest geschikte tools worden gekozen voor de taak, rekening houdend met de voorkeuren van de besluitvormers

### Conclusie

Test tools kunnen een krachtig hulpmiddel zijn voor het verbeteren van de efficiëntie, effectiviteit en kwaliteit van het testproces. Door de verschillende manieren waarop test tools kunnen worden gebruikt te begrijpen, kunnen teams de juiste tools kiezen en gebruiken om hun testproces te optimaliseren.

Verder is test tool management een belangrijk onderdeel van het softwaretestproces. Het selecteren van de juiste test tool, kan de efficiëntie en effectiviteit van het test proces te verbeteren. Door de uitdagingen bij test toolselectie te begrijpen en de juiste oplossingen te implementeren, kan een test tool worden geselecteerd die aan de specifieke behoeften van de organisatie voldoet.

Als je geïnteresseerd bent om test tool management in je organisatie te implementeren of te verbeteren, zijn er een paar dingen die je kunt doen:

- Begin met het identificeren van de test tool behoeften van je team/project/afdeling. Welke soorten testactiviteiten moeten worden uitgevoerd? Wat zijn de eisen om deze activiteiten efficiënt en effectief uit te voeren?
- Maak de test tool managementprocessen expliciet en identificeer welke fasen en processen verbeterd c.q. geïmplementeerd moeten worden.
- Evalueer de beschikbare test tools door middel van een Proof of Concept (PoC). Vergelijk de verschillende tools op basis van hun functionaliteit, prijs en ondersteuning. Maak hierbij de gestelde doelen en resultaten meetbaar.
- Ontwikkel een plan voor de implementatie van test tools. Dit plan moet minimaal de volgende zaken omvatten:
  - De verantwoordelijkheden voor de implementatie
  - De training en ondersteuning die nodig is voor het team
- Voer regelmatige audits uit om de effectiviteit van test tool management te evalueren.

Door deze stappen te volgen, kun je ervoor zorgen dat je team de juiste middelen hanteert en de meest geschikte test tools gebruiken om de kwaliteit van het test proces te verhogen. Dit resulteert dat de tests efficiënter worden uitgevoerd. Met uiteindelijk als doel meer inzicht te krijgen in de kwaliteit van het softwareproduct.

## Bronvermelding

- [1] Abdulwareth, A.J. & Al-Shargabi, A.A. (2021). *Toward a Multi-Criteria Framework for Selecting Software Testing Tools* . Retrieved from IEEE Xplore: <https://ieeexplore.ieee.org/document/9614194>
- [2] Hamed Taherdoost en Mitra Madanchian. (2023, January 9). *Multi-Criteria Decision Making (MCDM) Methods and Concepts*. Retrieved from MDPI.com: <https://www.mdpi.com/2673-8392/3/1/6>
- [3] Hamilton, T. (2023, November 30). *How to Select Best Automation Testing Tool?* . Retrieved from Guru99: <https://www.guru99.com/testing-automation-why-right-tools-are-necessary-for-testing-success.html>
- [4] Marselis, R.; van Veenendaal, B.; Geurts, D. & Ruigrok, W. (2022). *TMAP® Quality for DevOps teams*. Sogeti.
- [5] Myers, A. (2018, June 5). *Introducing Redpoint's Software Testing Landscape*. Retrieved from Medium.com: <https://medium.com/memory-leak/introducing-redpoints-software-testing-landscape-3c5615f7eeae>
- [6] Nieuwenhuis, M. (2017). *Wat is de duivelsdriehoek?* Retrieved from 123management.nl: [http://123management.nl/0/070\\_methode/a720\\_methode\\_02\\_duivelsdriehoek.html](http://123management.nl/0/070_methode/a720_methode_02_duivelsdriehoek.html)
- [7] Veenendaal, E. van & Al-Shargabi, A.A. (2018). ISTQB, TMMi or test automation three pillars for success: the PPT framework. *SQ Magazine #12*, 11 - 14. Retrieved from TMMi.