

**Moderne applicaties zijn modulair en gedistribueerd en hebben hierdoor vele externe integratiepunten (en daarmee afhankelijkheden). Het in isolatie testen van een dergelijke moderne applicatie is daardoor niet zonder meer mogelijk. Een primaire voorwaarde voor het testen is toegang tot alle (voor de te testen applicatie) externe applicaties.**

## Gebruik van simulatie in een testomgeving

### Alle voordelen en vooroordelen

**D**e externe applicaties variëren in aantallen van enkelen tientallen tot soms wel honderden in meer complexe situaties. Het is een hele uitdaging om een testomgeving te verkrijgen waarin al deze externe applicaties aanwezig zijn, beschikbaar zijn en het juiste gedrag vertonen. In dit artikel wordt betoogd dat het gebruik van een virtuele testomgeving vele voordelen biedt boven een 'echte' testomgeving.

Onder een virtuele testomgeving wordt een testomgeving verstaan waarin het gedrag van externe applicaties wordt esimuleerd. Met andere woorden: de applicatie die wordt getest communiceert met een virtuele versie van de externe applicatie in plaats van met een echte versie. In dit artikel worden echt en virtueel met elkaar vergeleken op achtereenvolgens de volgende onderwerpen: beschikbaarheid, compleetheid, gedrag, testdatamanagement en natuurlijk de kosten.

tuig op het water, direct de eerste keer met succes door de piloot wordt uitgevoerd? Het uitvoeren van deze handeling in een echt vliegtuig was vast geen onderdeel van de opleiding. Het laten landen van een miljoenen kostend vliegtuig in een rivier is uiteraard onmogelijk om in het echt te oefenen. Toch heeft deze piloot dit soort uitzonderlijke situaties waarschijnlijk meerdere malen geoefend: niet in het echt maar in een vluchtsimulator. Piloten zitten gedurende een groot deel van hun training niet achter de knuppel van een echt vliegtuig maar veilig op de grond achter de knuppel van een simulator. Dit voorbeeld laat zien dat simulatoren in staat zijn om de werkelijkheid, in dit geval een landing waarbij beide motoren kort na de start uitvallen, goed na te bootsen. Sterker nog, veel situaties zijn alleen in een simulator te trainen en niet in een echt vliegtuig.



Dit soort situaties wil je niet in de echte wereld trainen.

#### US Airways 1549

Op 15 januari vertrekt US Airways vlucht 1549 vanaf een vliegveld bij New York. Wat een routinevlucht had moeten zijn eindigt enkele minuten na het opstijgen in een historische landing op de Hudson rivier. Kort na de start wordt het vliegtuig geraakt door een zwerm vogels waardoor alle twee de motoren uitvallen. Een situatie als deze is een nachtmerrie voor piloten. Door op miraculeuze wijze te landen op de Hudson rivier worden de levens van honderden mensen in het vliegtuig en op de grond gespaard.

Hoe is het mogelijk dat een dergelijke complexe handeling, het landen van een verkeersvlieg-



**Edwin van Asch**  
Solution Consultant bij  
Systemation



**Jean Paul Verkade,**  
Solution Consultant bij  
Systemation

**Door  
applicatie-  
simulatie  
kunnen  
teams  
parallel  
testen**

Het behoeft weinig toelichting dat in de luchtvaart het gebruik van simulatie in veel gevallen de voorkeur heeft boven het vliegen in een echt vliegtuig. Wat wellicht meer mensen zal verbazen is dat er een grote parallel is tussen de voordelen van simulatie in de luchtvaart en de voordelen van simulatie tijdens applicatieontwikkeling in het algemeen en het testen van applicaties in het bijzonder.

In het hedendaagse complexe applicatielandschap is het steeds moeilijker om bepaalde delen te testen zonder genoodzaakt te zijn om het complete landschap in te richten. Moderne applicaties zijn modulair en gedistribueerd en hebben hierdoor vele integratiepunten (en daarmee afhankelijkheden) met andere applicaties. Voor een tester betekent het dat het opzetten van een testomgeving steeds moeilijker wordt. Niet alleen moeten alle omliggende systemen en componenten fysiek aanwezig zijn om te kunnen testen, ze moeten ook nog eens juist zijn geconfigureerd, van de correcte testdata zijn voorzien en beschikbaar zijn voor de test op het moment dat de test wordt uitgevoerd. Deze omliggende systemen variëren in aantallen van enkelen tientallen tot soms wel honderden in meer complexe situaties. Dit betekent dat het opzetten, bijwerken en na een test weer terug in de originele toestand brengen van een testomgeving een steeds grotere investering en een steeds langere doorlooptijd vraagt.

Een ander aspect is het gedrag van bepaalde systemen, net als bij het vliegtuig is het erg lastig om zeldzaam of afwijkend gedrag van een systeem te forceren tijdens een test. Hoe kun je het uitermate traag reageren van een backend systeem als gevolg van een batch job op dat systeem forceren? Of hoe test je de reactie van het systeem dat je aan het testen bent op het terugkrijgen van een bepaalde foutmelding van een externe webservice? Testers willen voor deze scenario's de reacties van het systeem onder test zien en dat niet eenmalig, maar deze test moet elke dag worden meegenomen in de regressietest om te controleren of het systeem blijft voldoen aan de specificaties.

### **Veel voordelen**

Het gebruik van een virtuele testomgeving biedt vele voordelen boven een 'echte' testomgeving. Onder een virtuele testomgeving wordt een testomgeving verstaan waarin het gedrag van externe applicaties wordt gevirtualiseerd (dit wordt ook wel simulatie genoemd). Met andere woorden: de applicatie die wordt getest communiceert met een virtuele versie van de externe applicatie in plaats van met een echte versie. Deze virtuele versie wordt een virtual-service genoemd. Een virtual-service simuleert de echte applicatie op basis van input en output berichten. Voor elk bericht dat de virtual-service ontvangt wordt op basis van een aantal logische

stappen en regels een bijbehorend realistisch antwoord samengesteld of opgezocht in een set van klaarstaande antwoorden. Deze antwoorden kunnen zijn opgenomen in een productieomgeving door gebruik te maken van de opnamefaciliteit van de virtual-service omgeving of zijn ingebracht door de tester die verantwoordelijk is voor de virtual-service.

De virtualisatie waar in dit artikel over gesproken wordt is een ander concept dan de virtualisatie van hardware en operating systemen zoals we die kennen van producten zoals VMware. Bij deze producten draaien de echte applicaties op een nagebootst platform. In het virtual-service concept worden de applicaties zelf nagebootst, of eigenlijk de communicatie tussen de applicaties.

We zullen echt en virtueel met elkaar vergelijken op achtereenvolgens de volgende onderwerpen: beschikbaarheid, compleetheid, gedrag, testdatamanagement en natuurlijk de kosten.

### **Beschikbaarheid**

Bij het opleiden van piloten gelden een aantal belangrijke beperkingen. Er is niet altijd een lesvliegtuig beschikbaar voor iedere individuele piloot in opleiding. Er is wellicht helemaal nog geen beschikbaarheid op het moment dat het om een nieuw (te ontwikkelen) type vliegtuig gaat. Natuurlijk is het mogelijk om een rooster op te stellen zodat de piloten gedurende een specifiek tijdslot mogen oefenen. In het geval van een nieuw type vliegtuig is er geen andere optie dan te wachten totdat dit vliegtuig uitontwikkeld, getest en opgeleverd is. Beide beperkingen resulteren in een doorlooptijd van de pilotenopleiding (time to market) die veel en veel langer is dan gewenst.

Om deze doorlooptijd terug te dringen maken vliegopleidingen gebruik van vliegsimulators. Hierdoor kunnen er meerdere piloten in opleiding tegelijkertijd oefenen (testen) en is het zelfs mogelijk om te oefenen in toestellen die nog niet fysiek beschikbaar zijn, bijvoorbeeld omdat ze nog ontwikkeld worden. Het behoeft weinig betoog om aan te nemen dat de introductie van simulatie bij een vliegopleiding heeft geleid tot sterke stijging van de beschikbaarheid van 'trainingsomgevingen' en daarmee het verkorten van de time to market voor een piloot in opleiding.

Bovenstaand voorbeeld is een goede analogie voor de beperkingen in beschikbaarheid van testomgevingen. De kosten voor het opzetten en gebruik van testomgevingen zijn hoog, waardoor er maar een beperkt aantal omgevingen beschikbaar is. Ook hier wordt vaak een rooster opgesteld om teams gedurende een bepaalde tijdsperiode toegang te geven. Op het moment dat nieuwe functionaliteit getest

moet worden (denk hierbij bijvoorbeeld aan een nieuwe gebruikersinterface of het uitvoeren van een ketentest) moet worden gewacht totdat de gehele testomgeving inclusief applicaties gereed is.

Door middel van applicatiesimulatie kunnen teams in parallel testen door ieder team een virtuele testomgeving ter beschikking te stellen. Een virtuele testomgeving bevat van alle applicaties waarvan de te testen applicatie afhankelijk is een gesimuleerde versie. Als er sprake is van een afhankelijkheid van een applicatie die nog niet af is kan applicatie simulatie helpen het testen naar voren te halen.

Het testen in parallel en het vroeg kunnen beginnen met testen leveren beide een grote positieve bijdrage aan het verkorten van de time to market van nieuwe applicaties en daarmee van nieuwe business functies voor een bedrijf.

### Compleetheid

Compleetheid is nauw gerelateerd aan het vorige onderdeel: beschikbaarheid. Luchtvaartmaatschappijen hebben een vloot bestaande uit vele vliegtuigen en vliegtuigtypes in verschillende configuraties. Het is voor deze luchtvaartmaatschappijen niet haalbaar om van elk van deze vliegtuigtypes in alle configuraties een oefenvliegtuig beschikbaar te hebben. Ook hiervoor maken luchtvaartmaatschappijen gebruik van simulatoren, tenminste één voor elk vliegtuigtype, welke ingesteld kan worden in verschillende configuraties. Hiermee kunnen piloten worden getraind in het vliegen met een specifiek type vliegtuig en kunnen eventuele probleemsituaties van verschillende configuraties worden nagebootst in een veilige omgeving.

### Vershillende versies

In analogie met de software wereld zien we dat in de praktijk er vaak meerdere versies van applicaties tegelijkertijd worden gebruikt. Zo kan het zijn dat in een productieomgeving gebruik gemaakt wordt van versie 1.0, in de acceptatieomgeving versie 1.1 en in de ontwikkelomgeving versie 1.2. Tussen programma's of organisatieonderdelen kunnen de verschillen nog groter zijn. Het is niet haalbaar om voor alle applicaties en alle versies van deze applicaties een complete omgeving in te richten (bijvoorbeeld een ontwikkel, test, acceptatie en productie omgeving). Een negatieve consequentie hiervan is dat productieproblemen moeilijk te reproduceren zijn in een test of ontwikkelomgeving. Immers, het applicatielandschap in productie wijkt af van de applicatielandschappen in de overige omgevingen door het gebruik van verschillende versies van bepaalde onderdelen in de andere omgevingen. De oplossing hiervoor is om van elke applicatieversie een virtuele versie beschikbaar te hebben. Deze virtuele versie is in staat het exacte gedrag van de echte versie van de applicatie te simuleren. Met deze virtuele versies kan snel



In de simulator kunnen de piloten en hun machines 'virtueel' worden getest (foto KLM).

een representatieve testomgeving worden opgezet zodat bijvoorbeeld een productieprobleem snel kan worden gereproduceerd.

### Gedrag

Het is van groot belang dat piloten kunnen oefenen onder de juiste omstandigheden. Aanvankelijk zal een stabiele omgeving gewenst zijn. Allereerst kan een piloot hiermee zijn basisvaardigheden oefenen. Ook is het hiermee mogelijk om op een objectieve manier de vaardigheden van verschillende piloten te vergelijken. Wanneer de piloten de basisvaardigheden onder de knie hebben wordt het van belang om ook calamiteiten te kunnen oefenen. Uit-

## De kosten van een virtuele applicatie omgeving zijn veel lager

zonderingssituaties zoals brandende motoren, niet werkende metertjes en landen in slecht weer zijn moeilijk en in bepaalde gevallen zelfs onwenselijk om te oefenen in een echt vliegtuig. Juist hier bewijst de simulator haar grote waarde. De gewenste uitzonderingsscenario's kunnen eenvoudig door de trainers worden aangepast en ingevoerd tegen een fractie van de werkelijke kosten en zonder de risico's van het oefenen in een echt vliegtuig met bijvoorbeeld een brandende motor.

Ook in de applicatiewereld komen we dergelijke fenomenen tegen. Moderne applicaties hebben vaak vele koppelingen met andere applicaties. Gedurende de ontwikkeling van een applicatie is het in eerste instantie van belang dat de gekoppelde applicaties stabiel gedrag vertonen. Instabiel gedrag van gekoppelde applicaties compliceren de automatisering van testen en de diagnose van fouten. De testresultaten zijn dan immers niet eenduidig en issues in de te testen applicatie zijn moeilijk te reproduceren.

Reële systemen zijn complex en worden binnen testomgevingen regelmatig opnieuw geconfigureerd of gewijzigd met een hoog risico op fouten en daarmee ongewenst instabiel gedrag.

Door het gebruik van applicatiesimulatie kan een stabiele omgeving worden gerealiseerd zodat de desbetreffende applicatie gedurende een bepaalde tijd 'in isolatie' kan worden getest.

### Uitzonderingssituaties

Zodra de belangrijkste functionaliteiten van een applicatie zijn gerealiseerd en getest wordt het testen van uitzonderingssituaties van belang. Het genereren van exceptionele condities (infrastructuurfouten, applicatie-excepties, etc) in een echte applicatie is niet evident en daardoor tijdrovend, niet reproduceerbaar en in een groot aantal gevallen in het geheel niet mogelijk.

Ook hierbij helpt simulatie om eenvoudig exceptionele condities door gekoppelde applicaties te laten genereren. Op basis van bepaalde input patronen kan een virtual-service worden geconfigureerd om bijvoorbeeld altijd bepaalde exceptie berichten terug te sturen of juist helemaal niets terug te sturen (voor het simuleren van een applicatie die niet reageert binnen een bepaalde tijd). Ook is het mogelijk om onderscheid te maken voor wat betreft de antwoorden op basis van de diverse test teams die aan het testen zijn: het ene team krijgt altijd normale antwoorden en een ander team krijgt altijd exceptionele antwoorden.

Bovendien kunnen deze exceptionele condities in een gesimuleerde omgeving eenvoudig door testers zelf worden geconfigureerd. Hierdoor zijn er geen afhankelijkheden naar bijvoorbeeld operationeel beheer en ontwikkelteams.

### Testdatamanagement

Piloten moeten geoefend zijn in vele verschillende scenario's. Zoals hierboven al staat beschreven is het hierbij van belang dat de omgeving het desbetreffende scenario kan ondersteunen. In het echt is dat natuurlijk erg lastig, een piloot moet leren landen op allerlei verschillende luchthavens onder verschillende omstandigheden (diverse windrichtingen, dag, nacht, slechtweer, etc.). Vanwege deze reden kunnen vluchtsimulatoren worden geconfigureerd om de juiste omgeving te representeren. Hierbij kan worden gedacht aan een specifiek vliegveld een specifieke windrichting (en dus landingsbaan) en specifieke weersomstandigheden (bijvoorbeeld mist).

Ook hier is weer een parallel te trekken met de applicatie test wereld. Voordat een test uitgevoerd kan worden moeten eerst alle applicaties op de juiste wijze worden geconfigureerd en moeten deze de juiste testdata bevatten. Een voorbeeld in de telecommunicatie is de portering van een klant (het van de ene aanbieder naar een andere aanbieder verhuizen van een klant). Hierbij moet de klant als actieve klant bekend zijn in alle systemen en moet de klant dezelfde eigenschappen hebben in alle systemen. Op het moment dat de test eenmalig is uitgevoerd begint het spel weer opnieuw, de huidige data is in alle systemen gewijzigd en dus is de data voor deze klant niet meer bruikbaar voor de volgende (regressie)test. Door het simuleren van gekoppelde systemen kan deze situatie eenvoudig worden voorkomen. De simulator kan eenvoudig weer in dezelfde status terug worden gezet. Dit voordeel is met name evident bij performance en load testen. Hierbij moeten duizenden tot miljoenen testgevallen worden geconfigureerd. Zonder applicatiesimulatie kost het opzetten van de testomgeving en de configuratie van de testdata factoren meer tijd dan de daadwerkelijke uitvoering van de test.

### Kosten en opbrengsten

Het doel van de pilotenopleiding is om piloten zo snel mogelijk op te leiden tegen een van te voren gedefinieerd kwaliteitsniveau en met zo min mogelijk kosten. Namelijk, hoe sneller en goedkoper een piloot is opgeleid des te sneller hij kan worden ingezet en zijn opleidingstraject gaan terugverdienen. De opleidingstijd van piloten kan worden verkort door de inzet van extra vliegtuigen. Extra vliegtuigen zijn echter duur en hebben de nodige beperkingen. Vluchtsimulatoren zijn factoren goedkoper dan echte vliegtuigen en kunnen daarom worden gebruikt om de opleiding te versnellen en om de opleiding goedkoper te maken.

In applicatieontwikkeltrajecten is er een vergelijkbare situatie. Functionaliteit moet zo snel mogelijk



De kosten van het opzetten en onderhouden van een virtuele applicatie omgeving zijn een fractie van de kosten van een echte omgeving.

worden opgeleverd tegen een van te voren vastgesteld kwaliteitsniveau en tegen zo min mogelijk kosten. De oplevering kan (in beperkte mate) worden verkort door het beschikbaar stellen van extra testomgevingen. Extra testomgevingen zijn echter duur en hebben de nodige beperkingen. Moderne applicatielandschappen bestaan uit tientallen gekoppelde applicaties en de benodigde investering in hardware en licentiekosten voor deze applicaties zijn fors. Denk hierbij aan bijvoorbeeld licenties voor databases, ERP applicaties, applicatie servers en middleware.

Met name spelen deze investeringen een rol bij het uitvoeren van grootschalige load en performance testen. Deze tests worden vaak niet of onvoldoende uitgevoerd vanwege de kosten voor het creëren van een testomgeving die gelijk is aan de productie omgeving.

Virtualisatie van applicaties in een testomgeving kan helpen de kosten voor het opzetten van nieuwe testomgevingen voor een groot deel te voorkomen. De kosten van het opzetten en onderhouden van een virtuele applicatie omgeving zijn een fractie van de kosten van een echte omgeving.

### Conclusie

Net als dat een piloot niet zijn hele opleiding doorbrengt in een simulator; er zullen ook voldoende vliegreizen gemaakt moeten worden in een echt vlieg-

tuig, is het niet de bedoeling om simulatie gedurende de gehele ontwikkelcyclus van een applicatie te gebruiken. Net als een piloot moet ook een applicatie een aantal 'vliegreizen' maken in een omgeving waarin de complete reële applicatieketen aanwezig is. Echter voordat de applicatie dit stadium bereikt, heeft de voorafgaande simulatie cyclus er voor gezorgd dat alle basishandelingen en uitzonderingssituaties al een keer de revue zijn gepasseerd en dat de bij deze scenario's optredende basale fouten inmiddels niet meer voorkomen.

De finale test in de echte omgeving is te vergelijken met het examen van een piloot, de applicatie ondergaat een aantal reguliere controles en de totale functionaliteit van de keten wordt doorlopen om een eindoordeel te geven over de kwaliteit van de gehele toepassing.

In dit artikel hebben we getracht aan te tonen dat applicatiesimulatie in veel gevallen een beter resultaat oplevert dan het gebruik van echte systemen. In bijna alle gevallen zullen gesimuleerde systemen sneller inzetbaar zijn, goedkoper zijn bij aanschaf, opzet en gebruik en een betrouwbaarder en eenvoudiger reproduceerbaar resultaat geven. Ook bij het testen van de zogenaamde uitzonderingssituaties levert simulatie een grote bijdrage aan de testbaarheid van een bepaald scenario. «

**Exceptionele  
condities  
genereren  
door echte  
applicaties is  
niet evident**