

Tien tips voor klanten om zelf te testen

Decos is hard bezig om het testen verder te professionaliseren. Alle losse testonderdelen worden samengevoegd tot één testproces. Ook is er begonnen met het uitrollen van testautomatisering, waarmee het mogelijk wordt testen vaker en op meer verschillende omgevingen te herhalen.

Ondanks dat Decos er alles aan doet om hun producten zo goed mogelijk te testen blijft het van belang dat de klanten het product ook testen voor ze het in productie nemen. Software blijft een product dat afhankelijk is van heel veel factoren. Het is onmogelijk om alle combinaties te testen. Het is belangrijk dat de klant zijn configuratie zelf test voor dat het in productie wordt genomen. Hiermee wordt het risico dat er een fout wordt gevonden, die vervelende gevolgen heeft voor gebruikers, een stuk kleiner.

Hieronder 10 tips die u helpen bij het zelf opzetten van een effectieve testprocedure.

Tip 1: Test uw configuratie, niet het product

Het is natuurlijk niet de bedoeling dat de klant het product volledig gaat testen. Dit hoort tot het werk van Decos. Wel moet de klant testen of de configuratie die het heeft, het naar wens doet. Zeker als het een nieuw product is moet er worden gekeken of het product doet wat er is afgesproken dat het zou doen. Ook moet er worden gekeken of het product goed werkt in zijn omgeving.

Tip 2: Bepaal een representatieve steekproef

Het is nooit mogelijk om alle verschillende mogelijkheden te testen. Daarom wordt er in de testwereld gewerkt met een "representatieve steekproef". Hoe groot deze steekproef moet zijn hangt af van het verwachte risico dat er wordt genomen. Is het een nieuwe product, dan is het risico natuurlijk een stuk groter dan wanneer het een kleine update is. De testinspanning zal in verhouding moeten zijn met het te verwachten risico. Tijdens het testen is het niet de bedoeling alle fouten die in het product zitten worden gevonden. Testen heeft als doel om een inschatting te maken van het risico dat wordt gelopen als een product in productie wordt genomen. Het aantal fouten dat wordt gevonden zegt natuurlijk wel iets over de kwaliteit.

Tip 3: Inventariseer de mogelijke impact

Belangrijk is het om een inventarisatie te maken welke processen door het product worden geraakt en welke impact dat heeft. Onderdelen die veel gebruikt worden of die erg belangrijk zijn voor de bedrijfsprocessen hebben een hogere impact dan onderdelen die zelden gebruikt worden of welke minder van belang zijn. Ook onderdelen die maatwerk zijn hebben een groter risico dat er iets mis gaat dan onderdelen die door alle klanten worden gebruikt.

Tip 4: Test na iedere aanpassing

Het herhalen van dezelfde testen noemt men een regressietest. Het doel van een regressietest is om te kijken af alles het nog steeds doet en er geen nieuwe problemen zijn geïntroduceerd. Een regressietest kan worden uitgevoerd na een patch of update naar een nieuwe versie, maar ook als de hardware omgeving waar de software op draait is veranderd. Het is echter niet nodig altijd met dezelfde diepgang testen. Hier kan bij het ontwerpen van de testen rekening worden gehouden.

Tip 5: Verdeel tests in categorieën

Door het tijdens het ontwerpen van de testen een indeling te maken is het snel duidelijk welke testen moeten worden uitgevoerd. Categorie 1 zijn alle testen die altijd moeten worden uitgevoerd. Met deze testen raak je alle belangrijke functionaliteit. Categorie 2 is een stap dieper die wordt uitgevoerd als er wijzigingen zijn van dit onderdeel. Deze categorie kan worden overgeslagen als er niets in dat onderdeel is veranderd. Categorie 3 wordt alleen uitgevoerd als een onderdeel heel goed getest moet worden. Dit zal alleen gedaan worden als er grote aanpassingen aan dat onderdeel zijn geweest.

Tip 6: Deel testscript op in overzichtelijke delen

Om overzicht in de testscripts te houden moet je ze opdelen in verschillende delen. Het is daarbij raadzaam om de verdeling van het te testen product aan te houden. Als het programma is opgedeeld in modules kan het testscript deze verdeling aanhouden. Ook kan er een verdeling worden gemaakt in beheerder en eindgebruiker.

Tip 7: Pas versiebeheer toe op testscripts

Versie beheer van testscripts is zeer belangrijk. Om later te kunnen kijken welke versie een testscript is en of het is bijgewerkt naar aanleiding van veranderingen in de software. Begin het testscript altijd met een versie nummer, datum waarop het gemaakt is, versie van het product dat getest moet worden en natuurlijk de naam van de persoon die het heeft gemaakt.

Tip 8: Beschrijf doel, uitgangssituatie, handelingen en verwacht resultaat

De testgevallen zelf bestaan uit de volgende onderdelen. Als eerst het doel van de test. Hier wordt kort en bondig beschreven wat het doel is van een testgeval en eventueel een koppeling naar een requirement van het te testen product. Daarna komt de uitgangssituatie die nodig is voor een testgeval. Zorg dat je bent ingelogd met een gebruiker die de rechten heeft van een administrator. De volgende stap is de handeling of de handelingen die moeten worden gedaan. En uiteindelijk het verwachte resultaat. Daarna de categorie zoals die hierboven is beschreven.

Tip 9: Voeg testgevallen samen tot een testscript

Alle testgevallen op een rij is dan het testscript. Als dit op een logische manier wordt opgedeeld kan het later door verschillende personen worden uitgevoerd. Ook is het praktisch het script niet te lang te maken, maar op te delen in meerdere korte stukken. Dit is voor het onderhoud een stuk makkelijker.

Tip 10: Leg vast wie, wat, wanneer getest heeft

Als de testscripts klaar zijn kunnen ze worden gebruikt voor de uitvoer. Hier moet dan nog worden bijgehouden wie er heeft getest, wanneer die dat heeft gedaan en op welke versie van het te testen product. En natuurlijk niet te vergeten de uitkomst van de test.

Succes met het opstellen van de testscripts en het uitvoeren ervan.

Mocht u verder nog vragen hebben over het testen of ondersteuning willen bij het testen, neemt u dan contact met ons op via 071-364 0700 of per e-mail via info@decos.nl