

In een software-wereld waar steeds sneller ontwikkeld en gereleased moet worden om mee te kunnen komen in de concurrentiestrijd, is het soms lastig om kwaliteit de juiste plek te geven. Dat het proces belangrijk is voor de kwaliteit van de software, heb ik in mijn vorige opdracht duidelijk ervaren. Onder grote tijdsdruk moest er een belangrijke release uit. Op aandringen van de developers werd er drie weken voor de releasedatum besloten dat er extra testcapaciteit nodig was. En het klopte dat dit nodig was, aangezien de applicatie ongeveer zes maanden niet getest was. In die zes maanden was er hard gewerkt aan de achterliggende code en was ongeveer 80% van de app gerefactored. Het enige dat nu nog moest gebeuren was de app testen en releasen. Eitje toch?

Team zonder tester

Helaas bleek dit lastiger dan gedacht, omdat er de afgelopen zes maanden ook al de nodige tijdsdruk op stond. Onder deze tijdsdruk was besloten om het scrumproces los te laten, zodat nog meer tijd besteed kon worden aan het ontwikkelen van software. Om de ontwikkelsnelheid nog verder omhoog te krijgen, waren ook de refinements overgeslagen, zodat er constant functionaliteit opgeleverd kon worden. En zonder tester in het team was er nog geen testaanpak gemaakt, laat staan testplannen en -gevallen. Nu is het op zich geen probleem dat je de ontwikkelsnelheid omhoog wilt brengen, maar in dit geval werd de kwaliteitscontrole er volledig uitgehaald.

De juiste balans

Het weghalen van de kwaliteitscontrole levert op korte termijn een winst in de ontwikkelsnelheid op, maar het komt ook met een groot risico dat op een later moment die ontwikkelsnelheid flink omlaag geschroefd moet worden omdat er in productie toch fouten ontdekt worden. Zo kan het dat je op een gegeven moment meer fouten aan het herstellen bent dan dat je de tijd hebt om nieuwe software te maken. Een ander uiterste is dat je zo obsessief bezig bent dat je bij elke kleine aanpassing van de software een volledige regressietest doet. In dat geval weet je een stuk zekerder dat er geen fouten in de software zitten, maar gaat je ontwikkelsnelheid ook drastisch omlaag. Als goede agile tester is het belangrijk om hier de juiste balans in te vinden.

De taak van de agile tester

Maar hoe vind je die balans? Het is belangrijk om te beseffen dat in het scrumproces al best veel momenten zitten waar ruimte is voor kwaliteitscontrole. Een belangrijke manier om die balans te vinden binnen het scrumproces is de refinement. En hoewel niemand zit te wachten op refinements die halve dagen duren, zit er ook niemand te wachten op een productieverstoring op zondag. Vooral niet als die voorkomen had kunnen worden door een user story iets beter te bespreken tijdens een refinement. Uiteraard is het gehele team

verantwoordelijk voor het opleveren van kwalitatieve software, maar zeker als tester kun je een groot verschil maken in deze fase van het proces. Niet per se door zelf alle vragen te stellen, maar door je team er bewust van te maken dat hoe beter je refinet, hoe hoger de ontwikkelsnelheid uiteindelijk wordt. Hoe vroeger in het ontwikkelproces je een fout vindt, hoe goedkoper het is. Dus wanneer je tijdens de refinement reeds een fout vindt, is dat vaak in enkele minuten op te lossen, terwijl dezelfde fout tijdens een productieverstoring dagen kan kosten om op te lossen. Als agile tester is het jouw taak om de kwaliteitscontroles tijdens dit proces te bewaken en het team bewust te maken van de collectieve verantwoordelijkheid en de voordelen daarvan.

Zorg voor een duidelijke regressietestset

Een andere manier om een goede balans tussen ontwikkelsnelheid en kwaliteit te krijgen is door te zorgen dat er een duidelijke regressietestset is. Deze regressietestset moet gebaseerd zijn op basis van een product risico analyse, zodat je bij elke release op basis van risico's kunt bepalen welke delen van de regressietest prioriteit hebben. Zoals gezegd is het een ontzettend tijdrovende klus om bij elke release alle delen van de regressietest te gaan doen. Door in die regressietest prioriteit op basis van risico in te bouwen, weet je wat er elke release het belangrijkste is. Hoe groter de release, hoe meer je van de regressietestset uitvoert.

Maak testautomatisering onderdeel van je routine

Wanneer je de ontwikkelsnelheid nog meer wilt opvoeren, wordt de volgende stap om deze regressietest te automatiseren. Hoewel dit technisch gezien natuurlijk geen onderdeel is van je (scrum)proces, zul je, wanneer je echt een hoge ontwikkelsnelheid wilt hebben, testautomatisering wel onderdeel moeten maken van je routine. Er zal tijd gemaakt moeten worden om testen te gaan automatiseren en je regressietestset zal op orde moeten zijn, voordat testautomatisering echt grote voordelen gaat maken. Denk hierbij ook aan hoe er met de resultaten van de automatische testen omgegaan wordt. Het moet onderdeel van je routine worden om te zorgen dat er iets gedaan wordt met de resultaten, om op die manier de testen en de kwaliteit van de software te verbeteren. Hierin is het jouw taak als tester om je teamgenoten te overtuigen waarom het belangrijk is om een regressietest te doen bij een release en hoe je de ontwikkelsnelheid kan vergroten door de juiste testen te automatiseren.

Houd een goede retrospective

Tot slot is het van groot belang voor de kwaliteit van de software om een goede retrospective te houden. Tijdens een retrospective kun je bespreken waar het bijvoorbeeld mis ging in de kwaliteitscontrole gedurende het proces. Het is erg belangrijk om ook de kwaliteit van het uitvoeren van het proces hierin te betrekken, omdat wanneer het proces door het hele team

goed uitgevoerd wordt, de kans op fouten in productie aanzienlijk kleiner wordt. Wees tijdens een retro dan ook niet bang om duidelijk aan te geven waar het volgens jou beter kan en waarom eventuele verbeterpunten leiden tot een verbetering van de kwaliteit van de software. Ook hierbinnen heb jij als tester de taak om je team te overtuigen dat jullie samen verantwoordelijk zijn voor de kwaliteit van de software.

Conclusie

Als tester in een agile omgeving heb je niet alleen de taak om de kwaliteit van de nieuw opgeleverde software te controleren. Je hebt ook de taak om het team bewust te maken van de gezamenlijke verantwoordelijkheid voor de kwaliteit tijdens het proces. In mijn vorige opdracht heb ik zelf gemerkt dat het proces bijdraagt aan de uiteindelijke kwaliteit van de software. Toen we het scrumproces beter gingen naleven en de drie hierboven beschreven punten duidelijker en beter gingen uitvoeren, ging de kwaliteit van de software aanzienlijk omhoog. Doordat we een release-cyclus hadden van twee weken, werd het ook veel duidelijker voor de klant wat ze konden verwachten per release. De kwaliteit ging omhoog omdat we meer tijd namen voor het proces en de kwaliteit binnen het proces. Wanneer iets onduidelijk was tijdens de refinement kon dit direct daar opgelost worden en niet pas bij een acceptatietest. Het zorgde uiteindelijk voor rust, duidelijkheid en een hogere ontwikkelsnelheid, terwijl de kwaliteit van de software omhoog ging. Durf daarom als tester niet alleen kritisch te zijn op de gemaakte software, maar ook op de uitvoering van het proces, want ook dat zal leiden tot een verhoging van de kwaliteit van de software!

Kasper Faber, testspecialist

Op **donderdag 24 juni a.s.** spreekt Kasper Faber op het gratis webinar: '**Assessing and pushing the quality of Software Development**'.

Hoe krijg ik inzicht in de kwaliteit van mijn ontwikkelproces en hoe kan ik het verbeteren?

Tijdens deze online lunch-presentatie nemen we je van 12.30 tot 13.30 uur mee in de wereld van testverbetering, assessments, patterns en anti-patterns van softwarekwaliteit die jouw organisatie kunnen helpen bij het verbeteren van kwaliteitsborging (zowel binnen de teams, als over de eigen teamgrenzen heen).

[AANMELDEN](#) is gratis!



Testspecialist Kasper Faber