

Over DevOps is al veel geschreven. Een Google search op DevOps op het moment van schrijven van deze blog geeft 60.200.000 hits! Als je dus wilt weten wat DevOps is, is er genoeg informatie te vinden, daar ga ik niets aan toevoegen. Waar ik het wel over wil hebben is een fout, of beter gezegd, een omissie die regelmatig voorkomt bij DevOps teams. Deze omissie belemmert hen om echt succesvol te zijn.

Shift-left, shift-right...

De kern van DevOps is dat een team integraal verantwoordelijk is voor de ontwikkeling en de operatie van een product in al haar facetten. De beperkte capaciteit van het DevOps team moet daarbij verdeeld worden tussen het (door)ontwikkelen van het product en het operationeel ondersteunen van het product in productie.

Een goed functionerend DevOps team kan de meeste tijd/capaciteit besteden aan innovatie. In mijn opdrachten zie ik regelmatig het omgekeerde; DevOps teams die veel tijd besteden aan de operatie en maar heel beperkt toekomen aan innovatie. Vaak zie je dan dat er veel problemen zijn met het product in productie en dat het oplossen hiervan veel tijd kost. Dit niet geplande werk zorgt voor verstoring op de agile leverbetrouwbaarheid, omdat de problemen meestal met prioriteit moeten worden opgelost. Als ik wat dieper kijk naar hoe deze teams werken, zie ik vaak dat activiteiten op het gebied van kwaliteit en testen niet goed passen binnen de DevOps aanpak. Vaak zie ik dat genoemde activiteiten nog op de oude manier worden uitgevoerd door teamleden in de rol van een traditionele tester. Deze teams hebben onvoldoende de shift-left en shift-right beweging gemaakt op het gebied van kwaliteitsborging. In deze blog zal ik uitleggen wat ik hiermee bedoel.

In het artikel gebruik ik veel termen die een relatie hebben met DevOps. Het voert te ver om al deze termen in dit artikel uit te leggen. Google biedt hier verdieping indien gewenst.

Shiften ten opzichte van wat?

Het referentiepunt is het moment van inproductiename. Dat is namelijk in de traditionele Waterval ontwikkelmethodiek het punt waar, net ervoor, het zwaartepunt van kwaliteitsborging ligt. De kwaliteitsborging bestaat dan uit het testen van het opgeleverde product door testers. De testers opereren daarbij als onafhankelijke partij en vormen een soort quality-gate naar productie. In de meest extreme variant wordt de kwaliteit in het product getest door heen-en-weer te pingpongen tussen de ontwikkelaars en de testers.

Shift-left is de beweging van kwaliteitsborging naar een punt verder links van inproductiename, dus eerder in het ontwikkelproces. Hier was al in de vroege testliteratuur aandacht voor, maar het bleek in de praktijk vaak lastig om te verwezenlijken.

Shift-right is de beweging van kwaliteitsborging naar een punt verder rechts van

inproductiename, dus in productie! Traditioneel zag je hier vaak alleen het uitvoeren van infrastructuurmonitoring.

Waarom moeten we shiften?

Bij agile productontwikkeling, waar DevOps onder valt, ligt de nadruk op het snel leveren van waarde aan gebruikers in een competitieve wereld. Dit vraagt flexibiliteit in ontwikkeling om in te kunnen spelen op de veranderende wensen en behoeften van de gebruikers en vereist een grote focus op kwaliteit; in een competitieve omgeving heeft de eindgebruiker namelijk de optie om, als het hem niet bevalt, jouw product te verruilen voor het product van de concurrent.

De benodigde snelheid van ontwikkelen, in combinatie met de vereiste hoge kwaliteit, vragen een andere invulling van kwaliteitsborging. Kwaliteit moet vanaf het eerste moment worden ingebouwd (quality-by-design) en in alle stappen van het ontwikkelproces worden bewaakt en dus niet zoals in waterval pas vlak voor inproductiename.

Het doel is dus producten leveren die waarde hebben voor de gebruikers. De vraag is hoe we kunnen vaststellen of we ons doel hebben bereikt. Hoe weten we of we de gewenste waarde hebben geleverd? Dat kun je alleen te weten komen door de waarde echt te meten. Liefst wil je ook nog verschillende oplossingsrichtingen uittesten om vast te stellen welke het meest gewaardeerd wordt door je gebruikers. Beide zaken kun je alleen in productie vaststellen; je zult dus naar rechts moeten schuiven.

Shift-left

Voorbeelden van shift-left activiteiten zijn user-story reviews, BDD, unit testen, testen in isolatie m.b.v. stubbing en/of mocking en integratietesten.

Zoals ik eerder in het artikel al aangaf is stond shift-left altijd al op de agenda van de testliteratuur maar kwam het vaak niet van de grond. Waarom kan het nu wel succesvol zijn?

Allereerst zijn er grote stappen gemaakt op het gebied van tooling. Het bouwen van een volledig geautomatiseerde CI/CD pipeline met geïntegreerde kwaliteitsborging op alle gewenste niveaus is nu met een ruime keuze aan tools (zowel betaald als Open Source) mogelijk.

Daarnaast zorgt de nauwere samenwerking binnen agile/DevOps er voor dat traditionele muren verdwijnen (denk aan de muur tussen ontwikkeling en test) en bevordert het de samenwerking. Het team is in DevOps volledig verantwoordelijk voor de kwaliteit en zal dus eerder dan in een niet-agile setting geneigd zijn om samen op zoek te gaan naar de beste invulling van kwaliteitsborging met de minste inspanning en kosten. [Barry Boehm](#) wist in

1979 al dat hiervoor shift-left de oplossing was!

Shift-right

Van oudsher zagen we in productie veelal technische infrastructuurmonitoring. Hierbij werden de servers gemonitord op CPU- en RAM-gebruik en het netwerk op nog beschikbare bandbreedte.

Op dit gebied is nu veel meer mogelijk. Naast de traditionele technische monitoring, behoort nu ook functionele-, performance- en zelfs businessmonitoring tot de mogelijkheden. Het laatste maakt het mogelijk om real-time in productie te monitoren of de core businessprocessen nog draaien en hoeveel waarde elk proces levert. Alarmen geven automatisch een signaal als de geleverde businesswaarde onder het verwachte niveau komt waarop vroegtijdig, vaak proactief nog voor de klant begint te klagen, kan worden ingegrepen.

Business tooling maakt het mogelijk om A/B testen in productie te doen. Het ontwikkelteam bouwt in nauwe samenwerking met de Business meerdere smaken van een oplossing en beide oplossingen worden in productie worden gezet; elke oplossing voor een geselecteerd publiek. Vervolgens geeft monitoring inzicht in welke oplossing het meest gebruikt wordt en de meeste waarde levert.

Conclusie

Levert een [DevOps](#) setting niet het gewenste resultaat op het gebied van snelle innovatie in combinatie met een tevreden klant? Houd dan de kwaliteitsborging eens tegen het licht van dit artikel en maak een beweging shift-left en/of shift-right al naar gelang waar de verbeterpotentie ligt en/of het eenvoudigst kan worden bereikt. Vergeet daarbij niet om te investeren in de vereiste kennis voor het implementeren van de nieuwe tools en technieken.

Doe dit met zorg en niets staat je meer in de weg om uit DevOps te halen wat er in al die mooie artikelen wordt beloofd!

Wim ten Tusscher