



Van handwerk naar continuous delivery in 1 jaar

Erick Segaar is Application Lifecycle Management-specialist bij Devoteam. In die rol kwam hij ook binnen bij een overheidsinstantie, waar het fenomeen ALM nog in de kinderschoenen stond. “Er was helemaal niks toen wij binnenkwamen”, zegt Erick er zelf over. Een jaar later is alles anders.

Handmatig

‘Helemaal niks’ is wellicht een beetje overdreven. De organisatie maakte gebruik van Team Foundation Server 2012. Behalve dat dat een wat oudere versie is, werd niet volledig gebruik gemaakt van de mogelijkheden van TFS. De server werd eigenlijk alleen gebruikt voor *version control*. De rest ging handmatig, zegt Erick: “De developer bouwde op zijn eigen computer. Hij zette zijn binaries handmatig op locatie, waar de testers de producten konden testen. Een trage, foutgevoelige, niet betrouwbare, niet-*repeatable* manier van werken.”

Continuous Integration

De eerste stap was de implementatie van een centraal buildsysteem en het bewerkstelligen van *continuous integration* (CI). Een complexe branching strategy werd overboord gezet. Iedereen werkt sindsdien op de *main branch* en elke developer checkt minimaal elke 24 uur in. “Die check-in fungeert als trigger voor de build server”, zegt Erick. “Die haalt de code op die jij net hebt ingecheckt en begint dan met bouwen. De tester kan dan ook aan de slag. Het is een onafhankelijk, betrouwbaar en stabiel centraal systeem. Als developer weet je zeker dat er met jouw code verder gebouwd kan worden en dat je code sowieso werkt.” In aanvulling op de continuous integration hebben testers nu zelf de gelegenheid om de laatste succesvolle build te publiceren naar hun eigen omgeving. Daar kunnen de beheerders de volgende versie ophalen, op het moment dat zij er klaar voor zijn.

Scrum

Devoteam pakte ook het scrumproces aan. Met name op het regressietest-onderdeel, dat maar bleef uitlopen. De verschillende scrumteams zijn nu opgelijnd om elke donderdag hun sprint af te hebben; dat maakt het mogelijk om tijdig een centrale regressietest uit te voeren op de vrijdag waardoor er maandag altijd begonnen kan worden met een nieuwe sprint. En dat is nog slechts een tussenfase, zoals verderop wordt uitgelegd...

Migratie

In september 2016 werd de migratie naar Team Foundation Server 2017 voltooid. Erick: “Daarmee is het mogelijk geworden om verschillende omgevingen aan elkaar te linken. Zo kunnen we CI-builds overzetten van de ontwikkelomgeving naar de testomgeving en uiteindelijk naar de acceptatie-omgeving. Voordat we kunnen opleveren naar de acceptatie-omgeving moeten we valideren of de deployment van de CI-builds is geslaagd. We concentreren ons daarbij op de omgevingsafhankelijke parameters. Als je wilt overbrengen naar een volgende omgeving, veranderen de binaries zelf immers niet, maar de omgevingsfactoren wel. Door middel van *smoke testing* hebben we omgevingsafhankelijke parameters getest, waarbij alle *evidence* wordt teruggekoppeld naar de centrale releasedefinitie. Op die manier wordt inzichtelijk dat alle cruciale stappen zijn doorlopen. Er is getest, alles is goed gegaan en er is niet aan de release gemorreld in aanloop naar het acceptatiestadium.”

De huidige status

Erick houdt zich nu bezig met de automatisering van functionele testen. Daar wordt het *continuous testing platform* Tosca voor gebruikt. Erick: “De Tosca-tests worden nu nog handmatig afgetrapt en er worden handmatig keuzes gemaakt over de tests die moeten worden uitgevoerd tijdens de regressietest. Straks zal, zo gauw de code wordt ingecheckt, de pijplijn zelf checken wat er is aangepast ten opzichte van de vorige versie en daar zelf een subtest op draaien. En elke nacht wordt een volledige regressietest gedraaid.” Hierdoor kan het ontwikkelproces ook op de vrijdag verdergaan. Die dag hoeft dan immers niet meer te worden gereserveerd voor de centrale regressietest. Erick: “Dat maakt het proces straks toch weer een stuk efficiënter.”

Conclusie

Door alle aanpassingen die Erick en zijn team hebben doorgevoerd, zijn de teams in staat langer door te ontwikkelen in een sprint. Dat betekent dat er meer effectieve tijd is om features toe te voegen. Door de verregaand geautomatiseerde deployment behoort de foutgevoeligheid die gepaard gaat aan handmatige acties tot het verleden; het is nu *push & go*. Fouten worden bovendien al in de ontwikkelomgeving opgemerkt en niet pas bij productie.

Door structureel goede aanpassingen te maken aan de bestaande grote monolieten in de organisatie, is Devoteam erin geslaagd de feedbackcycle te verkorten. De organisatie heeft minder tijd nodig om nieuwe features toe te voegen en loopt minder risico bij het aanpassen van oude. De organisatie is, kortom, agile geworden.

Van handwerk naar continuous delivery in een jaar? Het blijkt te kunnen!