Klomps & Snels Klomps & Snels Klomps & Snels

Het 100.000-dingen doekje

"Als we naar dit scherm kijken, dan ziet het er wel goed uit, alleen zou het handig zijn als er ook nog gegevens uit dat andere scherm op te zien zijn." "Ja, en dat je dan die wel direct hier kan aanpassen." "En het liefste dan ook nog dat je door kan klikken naar dat derde scherm waarop je de actuele statussen ziet". "OK, ik snap het. En willen jullie dan een blauw of een grijs knopje?"

Er kwam een blauw knopje en voor de ontwikkeling van dit deel van het systeem was ongeveer twee keer zoveel inspanning nodig als vooraf ingeschat. Bovendien bleek de performance van het scherm slecht: het aggregeren en sorteren van een grote hoeveelheid gegevens uit veel verschillende tabellen zorgt daar wel voor. Over de hoeveelheid mouseclicks waren de eindgebruikers helemaal niet te spreken. Wel goed geluisterd, toch een ontevreden klant...wat ging er mis?

In iteratieve projecten komt de kwaliteit van een IT-oplossing meer en meer op de schouders van de developer terecht. En dan bedoelen we kwaliteit als in "en dit hebben we nodig om ons bedrijfsprobleem op te lossen". Da's wel eng, want zet dat ons niet weer een aantal decennia terug, toen de kwaliteit van de software (lees hier: de kwaliteit van de code) bijzonder afhankelijk was van degene die gecodeerd had? Zijn we dan met iteratieve projecten weer terug waar we begonnen zijn?

De iteratieve developer herken je aan de gloed van absolute, niets ontziende doelgerichtheid. Alles wat gedaan wordt, wordt afgewogen tegen het doel – of het nu gaat om het implementeren van een pakket of het ontwikkelen van een maatwerksysteem, het moet een bedrijfsmatig doel ondersteunen. Verscheidene iteratieve ontwikkelmethodieken ondersteunen dit alles goed en gebruiken prototyping, veelvuldig en vroeg testen van onderdelen samen met of door de eindgebruikers, XP, daily builds en sterke sociale vaardigheden om die doelgerichtheid te ondersteunen.

Discussies over het waarom van bepaalde functionele vereisten schuwt hij niet; de iteratieve developer overziet namelijk de impact van de gebruikerswensen. Hij vertaalt ze (vrijwel) direct naar technische, functionele en planmatige gevolgen: "Dit druist wel tegen de nu gekozen architectuur in, weet je zeker dat je dit nodig hebt? Als we dit realiseren, dan moeten de eindgebruikers ongeveer vier keer zoveel handelingen uitvoeren, wil je dat? OK, maar dit heeft direct gevolgen voor de hoeveelheid te besteden tijd. Dan kunnen we dus iets anders niet realiseren.". Hij schuwt de discussie niet en voert deze op een manier die getuigt van respect voor de belangen van de ander, gericht op het gezamenlijk bereiken van het gemeenschappelijke doel. En zinsneden als "Oh, dat kost niet zoveel tijd, ja hoor, dat krijgen we er wel in", zonder dat de iteratieve developer een goed gevoel heeft bij de aspectgebieden technologie, functionaliteit en planmatigheid, zal hij niet uiten.

Iteratieve ontwikkelmethodieken, technieken én attitude; dat zijn gelukkig andere ingrediënten dan de persoonsgebonden afhankelijkheid van jaren geleden. Maar ook nu gaat het weer om de balans, niet zozeer over wie de kennis heeft en wie gestructureerd kan programmeren, maar de juiste mix van technologische en bedrijfskundige kennis bij de developer. Het zijn zaken die je niet in een cursus van een week of wat leert en de balans zal niet iedere developer zich eigen kunnen maken. Moet een iteratieve developer dan een soort 100.000-dingen doekje zijn? Tot op zekere hoogte wel. Hij hoeft niet in alle aspectgebieden uit te blinken, maar als hij maar allround genoeg en een goede teamspeler is, kunnen zijn teamgenoten zijn zwakkere punten compenseren. Een goed gebalanceerd team is hierbij een must, met een gefaciliteerd spel van duidelijke afspraken over rolverdelingen, huisregels als elkaar ergens op aan mogen spreken, en een goede sfeer. Iteratief teamwork dus.

Birgit Klomps en Tommes Snels, beiden managing consultants bij Capgemini, schrijven op persoonlijke titel over het werk in projecten, in workshops en alles wat daarbij komt kijken.