

Technische productkwaliteit is vaak onderbelicht

Hans Kuijpers (14 maart 2020)

Software wordt steeds belangrijker, ook bij de overheid. In tegenstelling tot wat je in de media vaak leest, heeft de overheid niet meer of minder inadequate IT dan andere sectoren. Sterker nog: de overheid staat in de top 3 samen met verzekeraars en banken als je naar de onderhoudbaarheid van de software kijkt. Dit in tegenstelling tot de telecommarkt en logistieke sector, zie artikel iBestuur Online 'Hoorzittingen geven vertekend beeld' van 13 mei 2014¹. Ook op de investeringen per systeem scoort de overheid in de top 3 na de financiële en transportsector. De inzendingen van Franks Derks "Is de overheid gewoon een bedrijf zoals alle andere?" en Martin van Egmond en Otto Duindam "IT-succesverhaal uit de overheid", laten zien dat het niet altijd kommer en kwel is en dat er veel relevante kennis binnen de overheid aanwezig is.

Een belangrijk verschil is natuurlijk dat overheidsprojecten meer in de spotlights staan: vooral als het fout gaat liggen ze onder een vergrootglas. En dat is logisch, het gaat immers om ons belastinggeld en we hebben er allemaal last van als de software niet doet wat we zouden mogen verwachten. Daarnaast is de schaal vaak groot (er zijn immers 17,5 miljoen klanten) en de vele (gestapelde) uitzonderingen en afhankelijkheden die wettelijk zijn vastgelegd, maken het niet eenvoudiger. Wat complexiteit betreft wil ik een aanvulling doen op de inzending van Rutger Gooszen "Voor complexe vraagstukken zijn geen simpele oplossingen": ondanks dat complexe vraagstukken veel als (mede)oorzaak worden genoemd waarom IT-projecten mislukken, mag het geen excuus zijn om geen toekomstvaste en aanpasbare software te schrijven.

We zien dat de markt zich positief ontwikkelt en wat we daarbij gelukkig ook zien is dat de systemen kleiner worden. Dit is niet alleen het geval bij de overheid, ik denk dat de parlementaire enquête ook invloed heeft gehad op de andere sectoren.

I. Onderwerpen die niet of beperkt staan in het Deltaplan en de inzendingen

1. Innovatiekracht via technische kwaliteit

Bij Rijkswaterstaat is in 2019 een systeem uit de jaren 90 vervangen. Ondanks dat het aardig wat voeten in aarde heeft gehad, is het systeem nu vijfmaal kleiner en vele malen eenvoudiger te onderhouden en daardoor aan te passen. Zolang men voldoende energie blijft steken in de technische kwaliteit, kan het systeem toekomstvast blijven. Continue innovatiekracht vergt dat niet alleen bij de initiële ontwikkeling, maar ook gedurende de onderhoudsfase wordt geïnvesteerd in kennis, tooling en technische kwaliteit voor toekomstvaste applicaties.

2. Inbedding van kwaliteit in contracten

Productkwaliteit is goed vast te leggen in contracten. Diverse aspecten zijn in alle levensfasen van een project (vanaf verzamelen van eisen tot en met de beheerfase)

op een objectieve manier te definiëren en te meten. Metrieken zoals productiviteit, defects en onderhoudbaarheid worden zelfs gebenchmarkt door onafhankelijke marktpartijen; denk aan ISBSG, QSM en SIG. De Nesma heeft diverse richtlijnen opgesteld die hierbij kunnen helpen.

3. Keuze tussen agile en waterval

Vervanging van een systeem (bv. omdat de technologie onvoldoende in staat is om aan een groeiend aantal transacties te voldoen, of omdat het te lang duurt om een nieuwe functionaliteit toe te voegen), kan vaak via een traditionele watervalmethode; de (meeste) functionele eisen zijn er immers al. De niet-functionele eisen moeten in ieder geval de toekomstvastheid en aanpasbaarheid nader duiden. Dit betekent dat de business de functionele wensen na de initiële vervanging eenvoudig met een agile aanpak kan laten uitbreiden. Omdat de business geduld moet hebben tot na de vervanging, dient tijdig over vernieuwing nagedacht te worden. Dit proces kan enige jaren duren voor een systeem met veel functies.

Wordt er gekozen voor agile werken, dan dient ook het management agile te denken en te sturen. Dit komt erop neer dat het management accepteert dat het minder invloed heeft op welke functionaliteit wanneer wordt opgeleverd.

Belangrijk is dus de weloverwogen keuze voor de agile of waterval aanpak. Een IT-organisatie volledig omzetten naar een agile organisatie is niet de *silver bullet* voor falende IT-projecten als waterval methodieken toepasselijker zijn.

II. Valkuilen van low code

Binnen alle sectoren wordt steeds meer gebruik gemaakt van low code platformen. Low code is mede ontstaan om iemand met domeinkennis snel nieuwe functionaliteiten te laten bouwen zonder al te veel kennis nodig te hebben van software engineering. Het is daardoor een goed middel gebleken om een proof of concept (PoC) uit te voeren of om snel een app in de markt te zetten.

In de praktijk bestaan wel een aantal valkuilen:

- a. De business is snel verwend en geneigd om de snelheid van ontwikkelen ook te verwachten bij traditionele applicaties. Het is vaak de bedoeling dat delen van functionaliteiten worden vervangen door een app. Vervolgens blijken 'oude' systemen aan te moeten blijven staan, omdat niet alles van het oude systeem wordt vervangen. Dit maakt het onderhoud van het resulterende portfolio alleen maar kostbaarder.
- b. Ook het ontwikkelen van low code systemen is onderhevig aan software engineering principes. Dit impliceert dat louter domeinkennis onvoldoende is om een toekomstvast systeem te bouwen. Met andere woorden: het is geen vanzelfsprekendheid dat low code applicaties goed onderhoudbaar zijn. Zie blog Luc Brandts: [Does low code mean better code?](#)²
- c. Voor grote organisaties, en dus ook de overheid, is het omzetten van applicaties voor de *core business* naar low code applicaties niet vanzelfsprekend. Het is essentieel dat een gedragen enterprise architectuur wordt vastgesteld en de vernieuwingen hieraan worden getoetst. Bedenk daarbij dat een PoC op een low code platform heel nuttig kan zijn om het concept te toetsen. Of een oplossing in

de vorm van low code ook toekomstvast is, zal de architectuurboard dienen te beslissen, waarbij ze het mandaat moet hebben om 'nee' te zeggen tegen de business.

III. Technische kwaliteit in Industriële Automatisering (IA of ICS)

Naast “gewone” software wordt voor de bediening, besturing en bewaking van machines, verkeerslichten, bruggen, sluizen, tunnels, maar ook voor treinen en vliegtuigen, software gebruikt die cruciaal is voor onze samenleving.

Deze software dient robuust te zijn, maar ook veilig. Deze niet-functionele aspecten worden vaak vergeten in contracten doordat ze abstract en lastig te benoemen lijken.

Daarnaast is de investering in de software voor grote infrastructurele projecten slechts een kleine kostenpost voor opdrachtgever en opdrachtnemers. Er zijn diverse projecten in Nederland geweest waarbij nieuwgebouwde wegen pas na grote vertraging konden worden geopend doordat de tunnelsoftware niet voldeed aan de eisen, en/of waarbij achteraf bleek dat de eisen niet expliciet genoeg waren (wie mag wanneer de slagboom sluiten?). Dan wordt het vrijgeven zonder testacceptatie wel heel lastig voor een wegbeheerder.

¹ <https://ibestuur.nl/weblog/hoorzittingen-geven-vertekend-beeld>

² <https://www.softwareimprovementgroup.com/resources/does-low-code-mean-better-code/>