

Aanvulling en reactie op de aanzet tot een IT-Deltaplan voor de overheid

Paul Mulder (20 januari 2020), [reactie op persoonlijke titel](#).

1. Ter inleiding

Ik heb van 1985 tot 2006 bij de IT-organisatie van de Belastingdienst gewerkt in diverse IT-functies. Daarna ben ik als architect ingezet geweest bij diverse overheidsorganisaties, waaronder ook weer twee jaar bij de Belastingdienst, maar ook bij grote organisaties in het bedrijfsleven. Bij die laatsten zie je dezelfde dingen fout gaan als bij bijvoorbeeld bij de Belastingdienst.

2. Is het zo slecht?

Allereerst wil ik er op wijzen dat je het als grote overheidsorganisatie zoals de Belastingdienst bijna niet goed kunt doen. Een kabinet bedenkt in het begin van de kabinetsperiode nieuwe wetten die in het derde jaar ingevoerd moeten worden. In het vierde jaar is niet gewenst omdat de negatieve effecten van de wet wel eens stemmen zouden kunnen kosten. Vaak is alleen de invoeringsdatum van een nieuw of te wijzigen systeem bekend. Requirements zijn slechts in hoofdlijnen bekend. Deze worden in de aanloop naar de invoeringsdatum pas uitgewerkt. Soms worden tot kort voor de deadline nog amendementen uit de Tweede Kamer meegenomen. Dit leidt er soms toe dat meerdere scenario's worden ontwikkeld, omdat niet zeker is welke kant de politiek op zal gaan, maar de tijd ontbreekt om daarop te wachten. In het slechtste geval komt er een compromis uit dat niet voorzien was.

De invoering van het eerste systeem van Toeslagen was een goed voorbeeld hiervan. Gezien de onzekerheid, de complexiteit en krappe planning was er de overeenstemming dat bij de start 90% van de aanvragen geautomatiseerd afgehandeld zou kunnen worden. 10% mocht dus uitvallen voor handmatige verwerking. Toen er een maand na invoering nog 50.000 aanvragen in behandeling waren, kwamen er vragen in de Tweede Kamer. De Belastingdienst bleef ernstig in gebreke, was de teneur, terwijl dit nog geen 1% van de 6 miljoen aanvragen betrof!

Iedere fout bij overheidsdiensten, zoals de Belastingdienst en het UWV, leidt tot koppen in de Telegraaf en tot Kamervragen. Vaak is het stemmingmakerij met een politiek doel. Ik weet dat vele systemen bij de Belastingdienst al jaren vrijwel probleemloos draaien. Ook veel projecten en agile ontwikkeltrajecten zijn best succesvol. Het beeld dat naar buiten komt is negatiever dan het in werkelijkheid is.

Toch vind ik ook wel dat er veel verbeterd kan worden. Hieronder beschrijf ik wat mij dikwijls is opgevallen en waardoor de doorlooptijd en de kosten van IT-nieuwbouw, IT-onderhoud en IT-beheer uit de hand lopen.

Sommige onderwerpen komen ook voor in het document van Daan.

3. Wat gaat er mis?

3.1 *Projecten zijn te groot*

Projecten hebben doorlooptijden van meerdere jaren en hebben soms honderden projectmedewerkers. Enorm veel tijd en energie wordt besteed aan overleg. Meningsverschillen over architectuur en projectaanpak nemen toe naarmate meer mensen betrokken zijn. Verloop in de langlopende projecten leidt bovendien tot weer nieuwe meningen en herhaalde discussies.

Opdeling in deelprojecten leidt tot eilandjes die door focus op de eigen belangen slecht samenwerken. Deelprojecten rapporteren dat zij hun deel van de functionaliteit werkend hebben. Het geheel werkt echter niet.

Advies: knip projecten op in kleine delen en ontwikkel de functionaliteit incrementeel.

3.2 *Verkeerde aanpak bij onduidelijke requirements*

Veranderprogramma's worden projectmatig aangevlogen, ook als de beoogde doorlooptijd meerdere jaren is en het onduidelijk is wat het uiteindelijke resultaat zal zijn. Bij onzekere requirements past een agile ontwikkelaanpak. Toch wordt aanbesteed alsof het een waternalproject betreft waarbij bij aanvang van de realisatie alle requirements tot in detail bekend zijn. Gevolg is dat ieder gewijzigd of toegevoegd requirement leidt tot meerwerk tijdens de realisatie. Grote budgetoverschrijdingen zijn het gevolg.

Advies: kies bij onduidelijke requirements voor een agile aanpak. Dit niet alleen in het realisatietraject, maar ook in de voorbereiding daarvan. Deel grote veranderingen op in epics op de portfoliobacklog, prioriteer deze voortdurend en pak de epics één voor één op.

Een dergelijke agile aanpak maakt het mogelijk in ontwikkeltrajecten naar voortschrijdend inzicht de koers bij te stellen. Door incrementeel werkende software op te leveren wordt telkens waarde opgeleverd. Eventueel stopzetten van de ontwikkeling leidt er niet toe dat alle ontwikkelkosten weggegooid geld zijn.

Ga niet uitbesteden bij onzekere requirements. Neem als organisatie zelf de regie over ontwikkeltrajecten. Sluit raamcontracten af met leveranciers van IT-specialisten, testers en Scrum masters. Geen resultaatverplichting maar een inspanningsverplichting die continuïteit garandeert in de ontwikkelteams. Voeg product owners uit de eigen business toe aan deze ontwikkelteams. Laat deze product owners adviseren door architecten, materiedeskundigen, security en privacy officers bij het prioriteren van de backlogs.

Deze aanpak kent minder overhead dan uitbesteden en voorkomt dat de kosten uit de hand lopen.

Agile ontwikkelen is overigens niet goedkoper dan waternal. De werkwijze met backlogs prioriteren, bigroom- en sprintplanningen, refinements en demo's is arbeidsintensief. Mits goed toegepast levert het echter al vanaf de oplevering van het

eerste increment waarde op. De kans op het ontwikkelen van achterhaalde functionaliteit is bovendien klein en de betrokkenheid van de business is groter dan bij waterval. De kosten zijn beter beheersbaar.

Waterval blijft een efficiëntere aanpak als alle requirements vooraf duidelijk zijn. Helaas is dat zelden het geval.

3.3 Aanbestedingen zijn lastig

Aanbestedingsregels en -procedures belemmeren snelle besluitvorming en resultaatgericht werken. De procedures zijn vooral bedacht vanuit juridisch perspectief, waarbij iedere marktpartij evenveel kans moet krijgen om mee te dingen. Europees aanbesteden kent een lange doorlooptijd. Tijd die er vaak niet is omdat de invoeringsdatum van nieuwe wetgeving al vast ligt. Projecten zijn daardoor vooral druk om te proberen onder aanbestedingsgrenzen te blijven. Dit leidt tot suboptimale oplossingen.

Vermijdingsgedrag leidt er soms ook toe dat ontwikkelprojecten aan ontwikkelorganisaties binnen de overheid worden gegund. Dat hoeft niet erg te zijn, maar als die organisaties niet de juiste competenties in huis hebben, leidt dat tot inzet van platformen die niet optimaal zijn voor de gevraagde functionaliteit.

Aanbestedingsregels leiden er ook toe dat overheidsorganisaties ongewenst met bepaalde technologie en dienstverleners opgezadeld worden. De Belastingdienst zit daardoor bijvoorbeeld met IBM technologie voor belastingen en douane en Microsoft technologie voor toeslagen. Kruisbestuiving en hergebruik van oplossingen tussen belastingen/douane en toeslagen is daardoor zeer beperkt.

Advies: toets (als politiek) de aanbestedingsregels op uitvoerbaarheid en ongewenste effecten. Stel de regels waar mogelijk bij.
--

3.4 Complexe materie

De versnippering van het politieke landschap en de lobbycultuur, leiden tot veel compromissen en uitzonderingen op wet- en regelgeving voor specifieke doelgroepen. Het gevolg is complexe wetgeving, moeilijke uitvoerbaarheid en zeer complexe IT-oplossingen in de uitvoering.

Bijvoorbeeld de politieke wens om in de actualiteit te werken heeft ertoe geleid dat aan miljoenen burgers op basis van aannames allerlei toeslagen worden uitgekeerd en dat een groot deel daarvan één of twee jaar later weer wordt teruggevorderd. Omdat het concept zeer fraudegevoelig is en de aantallen erg groot zijn, worden als tegenmaatregel complexe algoritmen ingebouwd die potentiële fraudegevallen eruit filteren. Dit leidt vervolgens ook tot onterechte afwijzingen en veel onbegrip in de samenleving.

Privacywetgeving en eisen aan transparantie (bijvoorbeeld de Wet open overheid) leiden tot grote complexiteit omdat iedere stap verantwoord moet kunnen worden en over alles gerapporteerd moet kunnen worden. Op zich is transparantie zeer wenselijk, maar klaag dan niet over de complexiteit die dit met zich meebrengt.

Advies: Probeer als politiek en als ambtelijke top, ondanks de versnippering, de regelgeving simpel en uitvoerbaar te houden. De IT-ondersteuning is dan sneller, goedkoper en met meer kwaliteit te leveren. De uitvoeringstoets die tegenwoordig gedaan wordt is overigens een goede maatregel om de wetgeving beter aan te laten sluiten bij wat haalbaar is in de uitvoering.

3.5 Strijdige belangen en verkeerde prikkels

Veel gaat fout door belangen die niet in lijn liggen met de doelen van de overheid of de gebruikers (burger of het bedrijfsleven) van de overheidsdiensten.

Belangen van IT-dienstverleners staan soms goede en duurzame oplossingen in de weg. De intentie is uiteraard om goede systemen te leveren die voldoen aan de eisen. Zij hebben echter ook belang bij hoge ontwikkelkosten en hoge kosten voor beheer en onderhoud.

Het onvoldoende ondersteunen van open standaarden en gebrek aan transparantie dwingen organisaties tot de inzet van meerdere pakketten van dezelfde leverancier. Deze vendor lock-in verhindert het sturen op flexibiliteit en reductie van IT-kosten.

Niet-onafhankelijke architecten ontwerpen architecturen die vooral ook gunstig zijn voor de software leverancier of IT-dienstverlener waaraan de architect verbonden is.

Persoonlijke belangen van stakeholders zoals bestuurders, managers, projectleiders, architecten leiden tot verkeerde beslissingen. Te innige relaties met leveranciers leiden tot deals die slecht doordacht zijn. Ze krijgen zelden de goedkeuring van architecten die vraagstukken benaderen vanuit meerdere perspectieven.

Soms is er sprake van een afrekencultuur, waarbij het onmogelijk wordt gemaakt om een afwijkende mening te uiten. Ook als er geen sprake is van een afrekencultuur is er dikwijls risicomijdend gedrag te bespeuren bij projectleiders, afdelingsmanagers en architecten. Meepraten met meerderen en meer groene smileys opnemen in de rapportages dan reëel is, verhullen de problemen die op de werkvloer al lang bekend zijn. De eigen carrièrekansen zijn belangrijker dan de performance van de organisatie.

Te grote ego's leiden tot gebrek aan samenwerking tussen organisaties en organisatieonderdelen.

Organisaties zijn dikwijls ongelukkig ingedeeld. Bij ontwikkeltrajecten zijn vele afdelingen betrokken die belangen hebben die niet in lijn liggen met de doelen van het ontwikkeltraject. Bestuurders denken telkens problemen op te lossen door een nieuwe harkje te tekenen. Het herverdelen van verantwoordelijkheden en andere poppetjes op managementposities kan soms werken. Helaas wordt echter zelden of nooit een organisatie ontworpen vanuit de gedachte dat processen zo weinig mogelijk organisatieonderdelen raken. Bij de vele reorganisaties die ik in 21 jaar bij de Belastingdienst ICT-organisatie heb meegemaakt heb ik dat tenminste niet gezien. Reorganisaties leiden tot onrust, kosten veel tijd aan inrichtingsactiviteiten en zorgen ervoor dat mensen drukker zijn met hun eigenbelangen en afdelingsbelangen dan met het IT-project waaraan ze een bijdrage moeten leveren. Telkens gingen wij

weer de hei op om met een nieuwe afdeling 'piketpaaltjes te slaan', m.a.w. het bepalen van 'wie is onze klant', 'waar zijn wij van' en vooral ook 'waar zijn wij niet van'.

Veelal leidt dat tot versnipperde verantwoordelijkheden en wegschuiven van lastige taken naar andere afdelingen. Afdelingsbelangen en het 'daar-zijn-wij-niet-van-syndroom' leiden tot tegenwerking en bureaucratie in ontwikkeltrajecten. Voorbeelden van organisatie-indelingen zijn dat er verschillende ontwikkelafdelingen zijn voor de frontoffice systemen zoals de portalen, de backoffice systemen en de integratie daarvan. Hierbij bepaalt de afdeling met de grootste invloed waar de complexiteit naar toegeschoven wordt. Dit is meestal niet de plek die de applicatieketen overziend de meeste logische is. Ook aparte afdelingen voor business architectuur, IT-architectuur, infrastructuur, ontwerp, bouw en test die deelopdrachten in een ontwikkelproject uitvoeren leiden tot vingerwijzen als er iets fout gaat en onvoldoende betrokkenheid bij het uiteindelijke projectdoel.

Advies: bepaal bij reorganiseren eerst de primaire processen en richt de organisatie zo in dat deze zo soepel mogelijk verlopen. (Uiteraard is op onderdelen functiescheiding nodig om de kwaliteit te borgen.) Zorg er voor dat doelen en KPI's van organisatieonderdelen en managers bijdragen aan de hogere doelen van de organisatie en van de overheid. Zorg voor onafhankelijke architecten. Eis dat projectmedewerkers en leveranciers zich committeren aan de doelen die met een verandertraject zijn beoogd. Zorg er ook voor dat men niet alleen verantwoordelijk is voor het projectresultaat, maar ook voor de periode daarna. Dat voorkomt dat quick and dirty software ontwikkeld wordt die slecht onderhoudbaar en beheerbaar is. DevOps is een aanpak met teams die zowel nieuwbouw als onderhoud doen. Ze worden gekenmerkt door gevoel van eigenaarschap en daardoor meer betrokkenheid bij het systeem en meer werkplezier.

3.6 Korte termijn denken

Oplossingen worden onvoldoende ontworpen vanuit meerdere relevante perspectieven. Te eenzijdig sturen op projectkosten, -risico's en -doorlooptijd leidt tot onvoldoende doordachte oplossingen. Dit wordt versterkt bij irreële deadlines. Een wel erg minimaal minimal viable product (MVP) is het gevolg. In dit MVP ontbreekt voor de klant belangrijke functionaliteit, zijn houtje-touwtje oplossingen ingebouwd en is geen rekening gehouden met beheerbaarheid en onderhoudbaarheid. De belofte om in de vervolgfase de technical debt weg te werken wordt in 99% van de gevallen niet nagekomen. Dikwijls is er geen budget meer en als er al budget is, gaat de prioriteit uit naar het toevoegen van nieuwe functionaliteit.

Korte termijn denken, veelal beperkt tot de project-deadline en -scope, leidt tot continue onenigheid met architecten, die veelal ook rekening houden met ontwikkelingen parallel aan en na afloop van het project.

Korte termijn denken houdt in dat architectuur niet serieus genomen wordt en vooral gezien wordt als een verplicht nummer in de PRINCE2 aanpak. Van daadwerkelijk werken onder architectuur – liever spreek ik van werken met architectuur – wordt niet de meerwaarde gezien en komt dan ook weinig terecht.

Korte termijn denken bij projectleiders maar ook bij bestuurders leidt vrijwel nooit tot duurzame IT-systemen. Het grootste deel van de verandercapaciteit gaat daardoor op aan beheer en onderhoud. Ruimte voor vernieuwing is er nauwelijks.

Advies: benader bedrijfsvraagstukken vanuit alle relevante perspectieven. Niet alleen de kosten en doorlooptijd van het verandertraject zijn van belang. Andere perspectieven zijn bijvoorbeeld de klant, de medewerker, de bedrijfsprocessen in de uitvoering, beheer en onderhoud, financiën (initieel en structureel), security en bestuurlijke aspecten.
Accepteer dat het verandertraject wat complexer wordt. De kosten gaan voor de baat.

3.7 Scenario denken

De wil van bestuurders om uit meerdere scenario's te kunnen kiezen is duur, contraproductief en enorm vertragend. Natuurlijk is het schetsen van meer dan één oplossingsrichting nuttig om voor- en nadelen van oplossingen in beeld te krijgen. In de praktijk merk ik echter dat het vooral onzekerheid en besluiteloosheid van bestuurders de reden is om scenario's uit te werken. Niet zelden worden alle voor- en nadelen, risico's, mitigerende maatregelen, project- en beheerkosten per scenario uitgewerkt; ook van scenario's die geen enkele kans van slagen hebben. Alleen dat al kost enorm veel geld en doorlooptijd. Dikwijls is al zoveel tijd verstreken dat alleen een voor de langere termijn ongunstig scenario nog haalbaar is voor de project-deadline.

Ook weerstand tegen de architectuur leidt nogal eens tot het initiatief om alternatieve scenario's uit te werken. Een goede architectuur is ontworpen vanuit meerdere perspectieven en is gericht op langere termijn doelen zoals flexibiliteit en lagere kosten voor onderhoud en beheer. Niet zelden kom je dan uit op een best-of-breed architectuur. Best-of-breed architecturen, security en privacy by design, aandacht voor generieke en herbruikbare componenten, loosely coupled als principe voor integratie leiden niet tot kortere en gemakkelijker ontwikkeltrajecten. Projectleiders met alleen oog voor de project-deadline beïnvloeden de beslissers in de richting van snel en gemakkelijk ontwikkelen en het vermijden van projectrisico's. Lange termijn doelen worden dan helaas niet gerealiseerd.

Advies: bedenk alleen scenario's als het echt nodig is, bijvoorbeeld vanwege te weinig tijd tot de invoeringsdatum. Werk alleen scenario's uit die echt een kans maken om gekozen te worden.

3.8 Onjuiste competenties

Niet passende competenties is ook een veel voorkomend probleem.

Projectleiders die niet in staat zijn een complex verandertraject te managen, zullen er alles aan doen om alles wat lastig is buiten scope te plaatsen.

Slecht opgeleide IT-ontwikkelaars produceren foutgevoelige en slecht onderhoudbare software. Als 'senior' aangeboden consultants zijn soms alleen in leeftijd senior. Ontwikkelaars in low code platforms missen nogal eens een echte IT-achtergrond. Bij stand-alone toepassingen gaat dat vaak nog wel goed. In de

complexe geïntegreerde applicatielandschappen bij de overheid is dat een groot risico. Ze zijn bijvoorbeeld onvoldoende bekend met concepten als loosely coupled integration en houden onvoldoende rekening met onderhoudbaarheid van software.

Agile ontwikkelen betekent nog steeds 'eerst denken, dan doen'. Dat wordt nogal eens vergeten.

Te hoog opgeleide IT-ontwikkelaars inzetten leidt ook tot ongewenste effecten. De overheid denkt nogal eens slim te zijn door in de functieprofielen voor softwareontwikkelaars een academisch niveau als eis op te nemen. Behalve dat deze mensen in verhouding duur zijn, hebben ze vaak weinig ambitie om een goede ontwikkelaar te worden. Ze richten zich op een functie als manager of architect. In ontwikkelprojecten discussiëren ze liever over de ontwikkelmethode dan dat ze gefocust zijn op het produceren van goede code.

De beloningsstructuur voor IT-functies is er ook debet aan dat de overheid niet de juiste mensen op de juiste plek ingezet krijgt. Een architect zit in een hogere loonschaal dan een ontwerper. De ontwerper weer hogere dan een programmeur. Een programmeur vervolgens hoger dan een tester. Een manager zit altijd in een hogere schaal dan de mensen in zijn of haar team. Hierdoor gaat iedereen die meer wil verdienen zich richten op een functie als architect of manager, ook als ze goed zijn in hun vak en misschien maar matig zullen presteren in een managementfunctie.

Die beloningsstructuur leidt ook tot een cultuur waarin het interessant is jezelf manager of coördinator te noemen. Ik heb projecten gezien waarin een testmanager, een testregisseur en een testcoördinator rondliepen, waarbij ik mij afvroeg wie er eigenlijk ging testen!

Daarbij komt ook dat de overheid IT-functies niet marktconform beloont. De aanvangssalarissen zijn veelal goed, maar als het op doorgroei aankomt vertrekt een groot deel naar het bedrijfsleven of wordt ZZP'er. Het gevolg is dat de overheid structureel aangewezen is op veel inhuur van externe IT-specialisten.

Advies: beloon de mensen naar de waarde die ze opleveren. Een goede programmeur is meer waard dan een matige projectleider. Een MBO'er kan een prima programmeur of tester zijn, die gefocust is op zijn taak en doet wat gevraagd wordt.

Let bij inhuur goed op de achtergrond van consultants. Leg bij twijfels over de IT-achtergrond de belangrijkste principes binnen de architectuur nog eens extra uit.

3.9 Verkeerde architectuurkeuzes

Architectuur vanuit de ivoren toren en gebrek aan gezond verstand leidt nogal eens tot verkeerde oplossingsrichtingen. Het is goed om af en toe te brainstormen over bijvoorbeeld hoe het zou zijn als de burger interactief zijn belastingaangifte invult en binnen luttele seconden zijn definitieve aanslag ontvangt. Dergelijke functionaliteit vraagt om een volledige realtime zaaksgewijze verwerking. Dit vraagt nogal wat van de IT op het gebied van functionaliteit, performance, schaalbaarheid en security.

Voorals je beseft dat het om 8 miljoen aangiftes gaat in slechts een paar maanden doorlooptijd met piekverwerkingen van 40 tot 50 duizend concurrent users.

Als je vervolgens weer met beide benen op de grond staat zul je beseffen dat de meeste burgers echt geen moeite hebben met een poosje wachten op de definitieve aanslag. Batchverwerkingen in bepaalde processtappen in een aangifteproces met zeer grote aantallen zijn zo gek nog niet. Batchverwerking wordt nogal eens beschouwd als ouderwets. Het is echter zeer robuust en beheersbaar, vooral bij grote aantallen. Als een batch niet goed gaat is, eenvoudig de verwerking nog een keer uit te voeren.

Een ander voorbeeld van luchtfietsen is de customer journey waarbij een ondernemer 's avonds thuis op z'n gemak in het ondernemersportaal van de Belastingdienst gaat kijken wat hij zoal aan belasting betaalt. Het portaal moet functionaliteit bieden om jaren terug te gaan in het dossier. De realiteit is dat een ondernemer zijn eigen administratie gebruikt en meer vertrouwt om inzicht in zijn boekhouding te krijgen. Hij komt het liefst alleen op het ondernemersportaal op de laatste dag van de maand. Hij wil dan inloggen en meteen de Btw-aangifte op het scherm zien.

Data analytics als techniek om belastingcontrole te ondersteunen is interessant en het onderzoeken waard. Echter, op basis van een beperkte proef met gekleurde uitkomsten gaan roepen dat het bestaande controleproces achterhaald is getuigt van weinig realiteitszin. Als vervolgens een afvloeiingsregeling aangeboden, wordt mag je er niet van opkijken dat medewerkers uit dat controleproces er massaal gebruik van maken.

Advies: design thinking als methode om vanuit de burger en de ondernemer te denken bij het creëren van oplossingsrichtingen is een goede aanpak. Blijf daarbij wel realistisch.

Onderzoek nieuwe technieken op het gebied van data analytics, rule based verwerking of bijvoorbeeld blockchain en AI, maar houd het doel van de bedrijfsprocessen voor ogen. Denk niet dat moderne aanpakken of technieken de oplossing zijn voor alles.

4. Wat zou de burger/belastingbetaler digitaal moeten kunnen vragen aan de gedigitaliseerde overheid?

MijnOverheid.nl is een goede ontwikkeling om burgers vanuit één portaal toegang te geven tot alle diensten van de overheid en gegevens die over je zijn geregistreerd bij de overheid. Als burger zou ik in dit portaal een overzicht willen hebben van alle belastingen die ik moet betalen en toeslagen, uitkeringen en toelagen die ik ontvang:

- inkomstenbelasting en voorlopige teruggave of aanslag (Belastingdienst);
- motorrijtuigenbelasting (Belastingdienst);
- gemeentelijke belastingen (gemeente of samenwerkingsverband);
- waterschapslasten (waterschap of samenwerkingsverband);
- uitkeringen (UWV);
- toeslagen (Belastingdienst/Toeslagen);
- kinderbijslag (Sociale Verzekeringsbank);

- studielening (DUO);

Idealiter zou de overheid één invorderingsorganisatie hebben die wat ik moet betalen en weer terug ontvang met elkaar verrekend. Een maandelijkse incasso/excasso is wat overblijft.

Ook wil ik alle overheidsdiensten kunnen benaderen via MijnOverheid.nl. Als burger zou ik niet de kennis hoeven te hebben over waar de diensten in het complexe overheidslandschap zijn ondergebracht.

Bijvoorbeeld:

- mijn Kliko-afvalcontainer is niet geleegd. Ik hoef niet te weten dat dit bij een bepaald afvalbedrijf zit waar de gemeente een contract mee heeft.
- ik wil een verbouwing laten uitvoeren. Waar vraag ik mijn vergunning aan? (Ik weet dat het Digitaal Stelsel Omgevingswet hiermee bezig is.)
- ik wil bezwaar maken tegen mijn WOZ-beschikking. Ik hoef niet te weten dat de gemeente dit door een samenwerkingsverband laat uitvoeren.

Idealiter kan ik alles in één portaal regelen, maar ik ben realistisch genoeg dat de verschillende organisaties hun eigen portalen hebben. (Binnen één organisatie is het al lastig genoeg om het eens te worden over de functionaliteit van een portaal.) Wel moet ik zonder opnieuw in te loggen door kunnen klikken naar de verschillende portalen.

5. Wat zou ik doen, als ik CIO-Rijk zou zijn?

1. Ik zou de overheidsorganisatie adviseren grote ontwikkelingen in incrementen op te delen en stap voor stap aan te pakken, zoals hierboven beschreven is. Een agile aanpak is in de meeste gevallen de enige manier om te voorkomen dat na lange ontwikkeltijd pas tot de conclusie gekomen wordt dat wat gevraagd wordt niet haalbaar is qua techniek, geld en doorlooptijd.
2. Kleine ontwikkelingen met beperkte doorlooptijd en waarbij de specificaties volledig duidelijk en definitief zijn, kun je gerust in een waterval aanpak realiseren.
3. Denk na over de rol van het BIT, welke problemen het moet voorkomen en tot welke problemen het juist leidt. Als het BIT vrijwel alle projecten afkeurt zegt dat iets over de projecten maar ook over het BIT. Het kost organisaties enorm veel tijd en energie om door de BIT-toets te komen. Ik heb nog niemand gesproken die gelukkig was met het verloop van een BIT-toets. Het is een politiek spel aan het worden. Het gevolg is dat organisaties op allerlei manieren proberen te voorkomen dat ze langs het BIT moeten.
4. Als de overheidsorganisaties in staat zijn nieuwe systemen incrementeel te ontwikkelen en daarbij opdelen in kleinere deelprojecten die na elkaar worden opgepakt, dan wordt het lastig voor het BIT het geheel te beoordelen. Opgeteld komen de projecten wel boven de BIT-norm uit, maar vooral bij agile ontwikkelen is in het begin alleen het eerste deelproject in detail bekend en is het geheel nog een globaal scenario dat in de loop van het traject nog kan wijzigen. Wat gaat het BIT hier dan van vinden? De agile aanpak reduceert bovendien al het risico op grote mislukkingen.
5. Het BIT kan een adviesrol hebben en bijvoorbeeld richtlijnen opstellen waaraan een project of een agile ontwikkeltraject moet voldoen. Ook de business case toetsen op realiteitszin kan een taak van het BIT zijn.

6. Ga bij grote veranderprojecten niet alleen praten met de opdrachtgevers, projectleiders en architecten van het project, maar doe dat ook met mensen binnen dezelfde organisatie die niet rechtstreeks bij het project betrokken zijn. Zij zien dikwijls andere risico's dan de mensen die belang hebben bij het programma.
7. Bedenk als CIO-Rijk ook wat voor rol de overheid zou moeten nemen op het gebied van innovatie. Binnen de overheid is dikwijls gemakkelijker om budget voor innovatie te regelen dan binnen het bedrijfsleven. Naar mijn mening wordt het werken in de IT van de overheid aantrekkelijker als er innovatieve ontwikkelingen plaatsvinden. Uiteraard lopen de risico's en kosten op als je innovatieve aanpakken en technieken toepast in ontwikkeltrajecten die gerelateerd zijn aan wetgeving met een vooraf bepaalde invoeringsdatum. Ga daar bewust mee om en communiceer dit ook naar de stakeholders in de politiek.