Adviescollege ICT-toetsing

Handreiking

Wie A(gile) zegt moet ook B zeggen

Agile werken bij de Rijksoverheid: ‘Wie A(gile) zegt moet ook B zeggen’

Agile werken is een manier van werk organiseren met zelfsturende teams waarbij men behendig kan inspelen op veranderingen. Binnen veel projecten bij de overheid wordt agile werken inmiddels toegepast of gaat dat binnenkort gebeuren. Kenmerkend voor zo’n op agile principes[1] gebaseerde werkwijze is het kortcylisch opleveren van werkende software. De verwachtingen van agile zijn vaak hoog: snel kunnen beginnen, kunnen inspelen op veranderingen, en effectiever en efficiënter software opleveren. In onze onderzoeken zien we dat daar waar, vanaf het begin van het project, de aansturing gebeurt met een stevig mandaat en de omgeving goed is ingericht, inderdaad voortvarend software wordt ontwikkeld. Bovendien nemen de voorspelbaarheid van planning en kwaliteit toe en is de kans groot dat door meer autonomie bij de ontwikkelteams het werkplezier toeneemt.

We zien echter ook dat agile werken niet altijd de verwachtingen inlost. In deze handreiking bespreken we eerst drie veel voorkomende valkuilen van deze manier van werken. Daarna geven we adviezen om die te vermijden. We belichten daarbij achtereenvolgens de rol van de opdrachtgever, de organisatie en het project.

Samengevat komen de adviezen neer op: wie A(gile) zegt, moet ook B zeggen. Een organisatie die weloverwogen kiest voor agile, moet de inrichting goed aanpakken, en bereid en in staat zijn om de benodigde randvoorwaarden in te vullen. Alleen dan kunnen de voordelen van agile werken volledig worden benut.

Valkuilen bij agile werken

Dat agile werken niet altijd oplevert wat ervan verwacht wordt, komt mede doordat agile principes niet goed ingevuld worden. In onze onderzoeken zien we drie valkuilen die vaak leiden tot onnodige vertragingen en tot verhoging van de complexiteit van toch al niet eenvoudige projecten.

- Goede sturing op het bereiken van doelen ontbreekt. Er ontstaat dan een onbalans tussen de mate van zelfsturing bij de ontwikkeling enerzijds, en prioritering en planning op de doelen anderzijds. In veel gevallen is de oorzaak hiervan dat niet wordt voldaan aan het stevige opdrachtgeverschap met mandatering naar een product owner dat voor de agile werkwijze noodzakelijk is.
- Alleen het (of slechts één) ontwikkelteam werkt volgens agile principes. Het eindproduct kan dan vaak niet volledig worden opgeleverd omdat andere teams niet zijn ingesteld op kortcyclisch bouwen, testen en uitrollen. Het kan ook zijn dat de juiste technische randvoorwaarden niet aanwezig zijn of dat men spanning ervaart tussen de manier van ontwikkelen en de manier waarop de organisatie is ingericht en werkt.
- Vooraf zijn uitgangspunten voor softwareontwikkeling niet goed vastgesteld, bijvoorbeeld op het gebied van niet-functionele eisen zoals performance en beveiliging. Ook voor (toekomstig) beheer, proefconversies, gegevensbeheer en migratie zijn uitgangspunten soms onduidelijk. Vaak zijn er geen (architectuur)kaders opgesteld of wil men deze gaandeweg ontwikkelen. Dit geldt ook voor functionele kaders. Detailontwerpen hoeven niet altijd van tevoren helemaal uitgewerkt te zijn; exploratie kan goed werken bij kleinschalige innovatie. Toch blijft het noodzakelijk dat vóórdat begonnen wordt met een softwareontwikkeltraject, een aantal fundamentele (architectuur)kaders en eisen plus de basis van het ontwerp zijn uitgewerkt.

Om deze valkuilen te vermijden, hebben we drie adviezen.

[1] Zie: Principles behind the Agile Manifesto, https://agilemanifesto.org/principles.html

Advies 1 | Opdrachtgevers, geef de product owner een stevig mandaat

De opdrachtgever maakt de doelen (het uiteindelijke product) expliciet, stelt budget beschikbaar en geeft kaders voor het tijdspad. Voor de dagelijkse sturing op de realisatie van die doelen is een krachtige product owner nodig.

Een product owner is verantwoordelijk voor het maximaliseren van de waarde die ontwikkelteams leveren. Hij of zij is iemand met functionele kennis over het gebruikersproces en heeft affiniteit met IT. Deze kennis is nodig om op het juiste niveau de planning te bepalen met enerzijds de gebruikers en het betrokken managementteam, en anderzijds het ontwikkelteam. Bij complexe projecten waar veel expertise nodig is en waar ook de IT erg complex is, kan het lastig zijn zowel het functionele als het noodzakelijke inzicht in IT te verenigen in één persoon. In dat geval kan de product owner bijvoorbeeld bijgestaan worden door technisch inhoudelijke experts. We zien dat deze samenwerking goed kan werken.

Om slagvaardig beslissingen te kunnen nemen heeft de product owner een krachtig mandaat van de opdrachtgever nodig. Alleen dan kan hij of zij verantwoord richting geven aan de ontwikkeling. Welke taken daarbij horen, beschrijven we hieronder.

Bewaken van korte- en langetermijndoelen

De product owner moet voldoende toegerust zijn om de korte- en langetermijndoelen te bewaken door deze te vertalen naar een backlog (lijst met werkvoorraad) en ze te prioriteren. Hiervoor is afstemming nodig met alle belanghebbenden. Bij het realiseren van functionele eisen en wensen moet voldoende aandacht zijn voor de kwaliteit, in termen van bijvoorbeeld continuïteit en betrouwbaarheid. De backlog kan gedurende een project worden uitgebreid met nieuwe activiteiten. De product owner is ervoor verantwoordelijk dat deze passen bij de doelstelling.

Om de backlog goed te kunnen beheren, moet de product owner:

- regelmatig afstemmen met de business, om bijvoorbeeld noodzakelijke functionaliteit te bepalen, prioriteiten te bepalen, en baten, risico's en afhankelijkheden in kaart te brengen;
- de projectdoelen vertalen naar en opdelen in behapbare stukken werk (bijvoorbeeld user stories). Bepalen wat het MVP[2] is;
- samen met het ontwikkelteam beoordelen of de backlog voldoende concreet is, waardoor deze eenduidig kan worden geïnterpreteerd. Het team moet immers de eisen of wensen uit de backlog goed begrijpen om de omvang in te kunnen schatten en om te kunnen bouwen;
- voldoende ruimte bieden om invulling te geven aan kwaliteitsaspecten, zoals privacy, security, continuïteit en toekomstige onderhoudbaarheid. In onze onderzoeken constateren we dikwijls dat de focus vooral ligt op functionaliteit en dat werk dat betrekking heeft op kwaliteitsaspecten wordt vergeten, onvoldoende aandacht krijgt of onvoldoende specifiek is. Hierdoor wordt functionaliteit gebouwd die niet voldoet aan de eisen en moet worden aangepast, wat weer leidt tot meerwerk, vertraging en hogere kosten;
- beoordelen of geprioriteerd werk ook daadwerkelijk in productie kan worden gebracht in de kortcyclische periodes die zijn afgesproken (per sprint, per paar sprints of per release van maximaal een half jaar). Dat is een van de agile principes: steeds kleine stukken functionaliteit operationeel maken, om vroegtijdig te kunnen bijsturen en om van te leren in de verdere ontwikkeling.

[2] MVP staat voor *Minimum Viable Product*. De eerste versie van een applicatie (of ander product) waar genoeg functionaliteit in zit om waarde toe te voegen voor de gebruikers en in de praktijk te beproeven.

Bewaken van de voortgang

De opdrachtgever moet de voortgang van het project kunnen blijven volgen. Dat betekent dat de agile teams zich over de voortgang moeten verantwoorden door te rapporteren op basis van de doelen die de opdrachtgever heeft geformuleerd, inclusief het gebruikte deel van het toegekende budget. De product owner heeft een belangrijke rol als het gaat om inzicht te bieden in de voortgang van het team.

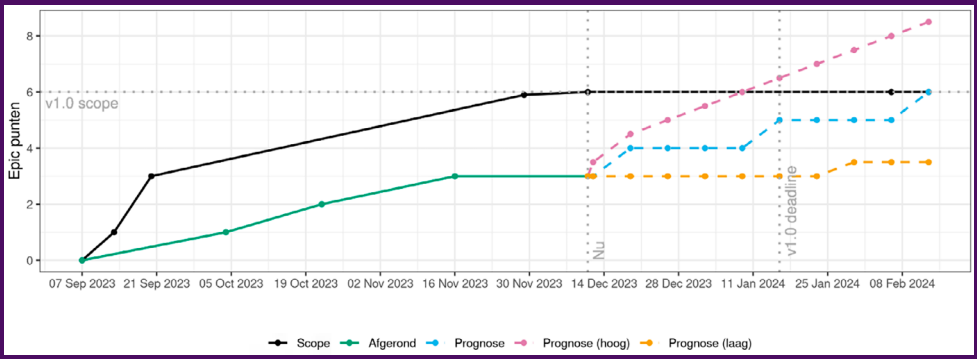
Een goede rapportage maakt duidelijk:

- wat de huidige voortgang is ten opzichte van de geplande voortgang;
- welke gerealiseerde doelen (of deeldoelen) wanneer operationeel zijn of worden gemaakt;
- welke kosten daaraan verbonden zijn en in welke mate deze afwijken van de raming;
- of de einddoelen nog kunnen worden bereikt op de vereiste momenten (bijvoorbeeld bij het in werking treden van wet- en regelgeving).

Een goede aanpak om te komen tot een dergelijke rapportage is het combineren van gegevens uit de voortgangsrapportages van de ontwikkeladministratie met die uit de financiële rapportage. Hierdoor kan beter worden gestuurd op de voortgang en kunnen vertragingen eerder worden ontdekt.

Voorbeeld: visualiseren voortgang agile project

De burn-up chart geeft inzicht in de voortgang van gerealiseerde functionaliteit (Afgerond) ten opzichte van het doel (Scope) en geeft daarbij de haalbaarheid aan van de realisatie van versie 1.0 op basis van drie prognoses. Dat zijn: voortgang op huidige ontwikkelsnelheid (Prognose), een positieve schatting waarbij de ontwikkelsnelheid omhoog gaat ten opzichte van huidige snelheid (Prognose (hoog)) en een behoudende schatting waarbij de ontwikkelsnelheid lager is dan de huidige snelheid (Prognose (laag)). De chart geeft de bandbreedte van de verwachte haalbaarheid van de volledige scope op de geplande v1.0-datum (in de figuur is dat 17 januari 2024) en geeft vroegtijdig inzicht in de haalbaarheid en impact van scopewijzigingen. In dit voorbeeld is de huidige ontwikkelsnelheid net onvoldoende om de geplande v1.0-datum te halen.



Om de voortgang goed te kunnen monitoren is daarbij een overzicht op één A4-tje met belangrijkste kenmerken van de huidige en afgelopen periode onontbeerlijk. Voorbeelden van kenmerken zijn:

- aantal gerealiseerde features/epics;
- aantal (opgeloste en nog openstaande) fouten gevonden tijdens testen;
- aantal (opgeloste en nog openstaande) fouten sinds laatste productiegang;
- aantal keer dat er naar productie is gegaan;
- gemiddelde oplostijd van fouten en productieverstoringen;
- gemiddelde doorlooptijd van wijzigingen.

Advies 2 | Management, houd rekening met de inbedding van agile werken in de omgeving

Invoering van agile werken moet worden gezien als de start van een cultuurverandering: een organisatie die agile wil gaan werken, moet zich rekenschap geven van noodzakelijke veranderingen in de organisatiecultuur. Daarnaast moeten de technische randvoorwaarden in orde zijn en goede afspraken met leveranciers worden gemaakt. Op deze drie voorwaarden gaan we hieronder in.

Integreer het agile werken binnen de hele organisatie

Het is belangrijk dat de werkwijze en de achterliggende principes door de hele organisatie worden geadopteerd en niet alleen door de ontwikkelteams. Dit voorkomt een mismatch en/of spanning tussen organisatieonderdelen (bijvoorbeeld tussen de ICT en de business).

Houd bij het plannen rekening met de ervaring die de organisatie heeft met de toepassing van agile werken. Als agile net is ingevoerd, adviseren we, naast het opnemen van de agile transitie in het risicolog als potentiële bron van vertraging, rekening te houden met de extra benodigde tijd om inleerproblemen te overwinnen. Zorg behalve voor een ruimere planning, ook voor trainingen en sessies waarin ervaringen worden uitgewisseld.

Uit de praktijk: integrale aanpak voor meer efficiency

- In een project moest een omvangrijke bestaande applicatie vervangen worden. De oude applicatie kon in modules (functionele brokken) worden vervangen waarbij ervoor gekozen was om meerdere technologieën toe te passen; de teams waren per technologie ingedeeld. Dat was geen goede keuze. De modules uit het bestaande systeem moesten hierdoor door beide teams worden geanalyseerd. Er moest ook onderling afstemming plaatvinden over het opknippen in de verschillende technologieën. Had je gemengde teams gehad, dan had ieder team zelfstandig een aantal modules kunnen oppakken. Ons advies aan dit project was dan ook om teams te organiseren rondom een stuk functionaliteit met een behapbare complexiteit, dat binnen de invloedssfeer van het team lag.
- Bij een ander project was het ontwikkelteam sterk afhankelijk van een team buiten het project. Dat andere team werkte niet agile en was gebonden aan een strak releaseschema. Daardoor ontbrak – met name aan het einde van een kalenderjaar – de ruimte om het agile team bij te staan. Dit leidde bij het agile team tot vertraging en wachttijd waardoor zij niet meer in staat waren kortcyclisch op te leveren. Ook hier lag de oplossing in het verbeteren van de integratie tussen de teams door deze niet in te richten op basis van technologie maar multidisciplinair, gericht op een functioneel domein.

Faciliteer het kortcyclisch werken met de juiste technologie

Zonder de juiste technieken, tooling en technische kaders kan een agile organisatie chaotisch worden en ongewenste resultaten opleveren. De opdrachtgever van een agile project dient daarom eerst de benodigde technische randvoorwaarden en fundamentele kaders (zie advies 3) voor ontwikkelteams te scheppen.

Kortcyclisch werken houdt in dat er in hoge frequentie nieuwe functionaliteit voor een werkend product wordt opgeleverd. Om dit voorspelbaar te kunnen doen met minimale kans op fouten en om fouten snel te kunnen herstellen[3], dient een aantal randvoorwaarden te worden ingevuld. Zo moet de technologie er geschikt voor zijn en dienen de teams toegang te hebben tot het juiste instrumentarium. Dat kan door best practices te volgen op het gebied van softwareontwikkeling, zoals het werken met een moderne ontwikkelomgeving, testautomatisering en een ontwikkelstraat waarbij van het programmeren van de

[3] Dit zijn voorbeelden van DORA-metrieken (zie: https://dora.dev/research/) die een indicatie geven van de volwassenheid en prestatieniveau van een ontwikkelorganisatie.

eerste regels code tot aan het opleveren en in gebruik nemen van werkende software zoveel mogelijk geautomatiseerd is. Verbeteringen in de werkwijze zijn nooit klaar: agile werken gaat uit van continu verbeteren.

Een uitstapje: retrospectieve sessies om continu te verbeteren

In retrospectieve sessies wordt teruggeblikt op de afgelopen iteratie, en gekeken naar wat goed ging en wat beter kan. Op basis daarvan worden afspraken gemaakt met de bedoeling deze de komende iteratie op te volgen. In onze onderzoeken zien we vaak dat retrospectieve sessies (retrospectives) onvoldoende plaatsvinden. De sessies worden regelmatig gezien als een verplichte stap, maar de insteek om te verbeteren mist; acties worden voor de vorm geformuleerd en slecht opgevolgd. Toch zijn retrospectieve sessies van belang om het team continu te verbeteren. Ons advies aan organisaties is dan ook om retrospectieve sessies te omarmen. Gebruik ze als momenten om, op basis van statistiek en ervaren belemmeringen, te bepalen welke verbeteringen gewenst zijn in het proces. Evalueer die verbeteringen in een volgende retrospectieve sessie.

Maak passende afspraken met partners en leveranciers

Agile werken vraagt om de juiste afspraken met leveranciers en partners[4]. Zorg dat deze afspraken aansluiten bij de nieuwe werkwijze. Flexibiliteit en ruimte zijn cruciaal. Het faciliteren daarvan gaat goed samen met heldere zakelijke afspraken. Let bij het maken van afspraken op de volgende punten:

- Maak duidelijk op welke punten – zoals tijd, budget en resultaat – gestuurd gaat worden. Maak daarbij afspraken over de bijbehorende procedures en KPI's. Naast eisen aan functionaliteit moeten eisen op het gebied van security, privacy, documentatie en onderhoudbaarheid worden meegenomen. Zorg ook voor voldoende flexibiliteit in de uitvoering, bijvoorbeeld als het gaat om op- en afschaling van expertise en de mogelijkheid om – indien nodig – naar alternatieve producten over te stappen.
- Volg de best practice om leveranciers resultaatverantwoordelijk te maken. Daarvoor zijn afspraken met leveranciers nodig die goed aansluiten bij de verwachtingen en werkwijze van de eigen organisatie, bijvoorbeeld voor wat betreft de releasefrequentie, planningen en verwachte productiviteit van de ontwikkelteams.
- Kijk kritisch naar bestaande modelovereenkomsten en standaardcontracten. Ga na of die ook voor agile geschikt zijn.

Het bovenstaande luistert nog nauwer als er ook wordt samengewerkt met leveranciers/externe teams die niet agile werken.

[4] Lees ook onze handreiking gezamenlijke ICT-ambities (https://www.adviescollegeicttoetsing.nl/publicaties/documenten/publicaties/2023/05/15/handreiking-gezamenlijke-ict-ambities)

Advies 3 | Project, ga doordacht van start met agile softwareontwikkeling

Een goede product owner met voldoende mandaat (zie advies 1) en een goed ingerichte omgeving zijn randvoorwaarden om als project effectief software te ontwikkelen op een agile wijze. Daarnaast moet het project zelf de eigen zaken op orde hebben.

Een zorgvuldige beginfase waarin het toekomstige project vanuit meerdere invalshoeken en disciplines wordt bekeken, betaalt zich terug. Hierbij worden de benodigde specialisten uitgenodigd zodat het team goed voorbereid begint aan de bouw van de software. Als deze fase wordt overgeslagen en de eisen onvoldoende helder zijn, leidt dat onherroepelijk tot herbouw en/of vertraging. Eisen aan beveiliging, beschikbaarheid of bruikbaarheid van kernsystemen of missiekritieke systemen dienen van tevoren helder te zijn, omdat latere keuzes hiervan afhangen. Als eisen zich moeilijk laten specificeren, is het goed te beginnen met een proof of concept, om ze op die manier meer inzichtelijk te krijgen.

Uit de praktijk: ondoordachte start leidt tot vertraging

Meerdere bouwteams startten met de vervanging van een groot systeem dat gebruikt werd door verschillende organisaties, nog voordat goed was nagedacht over wat gebouwd moest worden. Er was slechts op hoog niveau beschreven welke functionaliteit de verschillende teams zouden bouwen. De teams kwamen daardoor niet op stoom; na het inrichten van de ontwikkelomgeving werd al snel duidelijk dat er geen bruikbare items op de backlog stonden, alles riep vragen op en er werd geen software ontwikkeld. We hebben de projectmanager geadviseerd in te grijpen en de ontwikkelteams te koppelen aan een (nog aan te nemen) businessanalist die de PO ondersteunt om de backlog op orde te krijgen.

Agile teams zijn zelfsturend, maar dit betekent niet dat ze kunnen werken zonder doordachte architectuur, waarin de technische kaders staan beschreven. Richtlijnen over bijvoorbeeld acceptabele technologieën, standaarden en koppelvlakken creëren duidelijkheid zonder dat die de autonomie van de teams teveel beperken. Daarnaast is het van belang om vanaf de start van een project rekening te houden met de implementatie in de organisatie en (transitie naar) beheer. Hoe gaan we communiceren over de aankomende veranderingen, waar gaat het systeem draaien, wie moet het onderhouden, welke documentatie is daarvoor nodig, welke datamigratie is noodzakelijk en welke eisen worden daaraan gesteld? Als dit soort kaders ontbreekt, kan dat leiden tot herstelwerk of tot duurder onderhoud en beheer.

Denk dus goed na over (niet-)functionele eisen, architectuur en ontwerp, en kwaliteitsmaatregelen. Er zit een zekere spanning tussen hoeveel of hoe gedetailleerd een en ander van tevoren beschreven kan of moet zijn, en wat tijdens de ontwikkeling uitgewerkt kan worden. Dit hangt bijvoorbeeld af van kenmerken van het product: vervangen we (delen) van een bestaand systeem, hoeveel gebruikersinteractie is er, welke wet- en regelgeving is van toepassing, etc. Beschrijf bijvoorbeeld[5]:

- de technische kaders in architectuurbeschrijvingen;
- de functionele eisen in use cases, epics en/of stories;
- de niet-functionele eisen specifiek, meetbaar, acceptabel, realistisch en tijdgebonden;
- de maatregelen die het project gaat treffen om de eisen te implementeren en hoe kan worden vastgesteld of aan de eisen wordt voldaan;
- een testplan met daarin onder andere de aanpak van testen en testautomatisering.

Het toekomstige systeem kan met voldoende detailniveau worden uitgewerkt in een paar voorbereidende iteraties. Voordat echt met de ontwikkeling wordt gestart, moet dit uitgedacht zijn. Breng de gebruikersinteractie vroeg in beeld met behulp van interactieve schetsen (mock-ups/ prototypes). Deze kunnen goedkoper en sneller getoetst worden met eindgebruikers dan wanneer een ontwikkelteam de software eerst helemaal bouwt.

[5] Zie hoofdstuk 5: producten van de voorfase, in de ICTU Kwaliteitsaanpak (https://ictu.github.io/Kwaliteitsaanpak/) voor een voorbeeld van producten die voor de start van de softwareontwikkeling belangrijk zijn.

Wij adviseren ten slotte om, samen met de product owner, vroegtijdig goed na te denken over de juiste prioritering van de backlog in relatie tot het eindproduct: wat gaat de gebruikers het meest helpen? En hoe kunnen we zo prioriteren dat de MVP een versie is die gebruikers laat zien hoe de verwachte functionaliteit uitpakt en hoe het systeem zich in de praktijk gedraagt?

Uit de praktijk: juiste kaders voorkomen duur onderhoud en beheer

Door juiste technologische kaders mee te geven aan de teams kunnen beheeruitdagingen voorkomen worden. In een project waren de technische kaders niet vooraf vastgelegd. De drie teams die samen aan één kleine applicatie werkten, mochten zelf hun technologie kiezen. Dat leidde in ieder team tot een andere technologie voor de opbouw van de webpagina's. Voorzien was dat na afronding van het project maar één onderhoud- en beheerteam nodig zou zijn. Maar in de praktijk is in de beheerfase een ontwikkelaar nodig die alle drie de technologieën goed beheerst en daar zijn er minder van beschikbaar. Of er moet worden gekozen voor een groter team dat alle expertise in huis heeft.

Deze publicatie is een uitgave van:

Adviescollege ICT-toetsing
Postbus 16292 | 2500 BG Den Haag
T 070 426 63 23

december 2023