

Mislukte automatisering

Jacques Honkoop

Veel automatiseringsprojecten mislukken. Daar is iedereen het over eens. Vooral grote maatwerkprojecten crashen en komen met regelmaat in het nieuws. De overheid loopt hierbij voorop. Gedegen onderzoek ter zake ontbreekt echter grotendeels. Wanneer dit toch plaatsvindt, gaat het doorgaans om nationale of sectorale rapportages, zoals bijvoorbeeld de Commissie Elias heeft laten zien met betrekking tot rijksoverheidsautomatisering. Slechts weinig onderzoeksbureaus doen op grote schaal wereldwijd onderzoek naar het falen van automatiseringsprojecten. Een uitzondering op deze regel vormt het Standish¹ Institute, dat vanaf 1985 de uitkomsten van ICT-projecten in verschillende landen onder de loep neemt. In recente publicaties tonen de Amerikaanse onderzoekers aan dat ontwikkelen met agile-methodieken leidt tot een sterke afname van het aantal mislukte projecten. Wij bekijken de onderzoeksresultaten kritisch.

Chaosrapporten

In 2016 staat bij het Standish Institute de teller op 85 duizend. Dat is het aantal ICT-projecten dat de onderzoekers afgelopen jaren hebben geanalyseerd. Geen enkel ander onderzoeksinstituut heeft een dergelijk grootschalig onderzoek uitgevoerd. De Standish rapporten verschijnen sinds 1994, aanvankelijk iedere twee jaar, maar sinds 2008 jaarlijks. Ze dragen de naam Chaosrapporten. De onderzochte projecten zijn ongeveer 55% Amerikaans en 25% Europees. Wat betreft de omvang van de ondernemingen is 50% groot (omzet hoger dan \$ 500 miljoen), 30% middelgroot (omzet \$ 200 miljoen tot \$ 500 miljoen) en 20% klein (omzet \$ 100 miljoen tot \$ 200 miljoen). Standish onderscheidt drie categorieën:

- Succes: opgeleverd conform initiële ramingen en functionaliteit
- Betwijfeld: uitloop in tijd en/of kosten en/of minder functionaliteit dan aan begin van het project is vastgelegd.
- Mislukt: niet geïmplementeerd.

Deze worden naar verschillende gezichtspunten gepresenteerd. Belangrijkste zijn breakdowns naar onder meer: projectomvang, nieuwbouw – verbouw – pakket, sectoren, faalfactoren. En sinds 2011 ook de gevolgde methodiek: *agile versus waternival*. Hoe omvangrijker het project, hoe groter de kans op mislukking. De belangrijkste faalfactoren

¹ Standish is opgericht door Jim Johnson met als doelstelling het verrichten van onafhankelijk onderzoek en het geven van adviezen. Hoewel men een lidmaatschap kent is het een private onderneming. Standish is gevestigd in Boston. Zie www.standishgroup.com

voor zowel de grote als de kleine projecten zijn van bestuurlijke, organisatorische en sociale aard. Gezamenlijk zijn deze voor 45% de belangrijkste oorzaak.

Bedroevend

De cijfers in de Chaosrapporten stemmen niet tot vreugde. In 1994 werd slechts 16% van de onderzochte projecten met succes afgerond, terwijl 53 % een twijfelachtig resultaat had en 31% mislukte. Ruim 20 jaar later, in 2015 bedraagt het succespercentage 29% tegen 19% mislukkingen. Het gaat dus heel langzaam beter.

Er is kritiek op de door Standish gehanteerde criteria.² Zo valt direct op dat 'succes' (alles conform ramingen uitgevoerd) en 'mislukt' (niet geïmplementeerd) nogal strak zijn gedefinieerd. Dat leidt onvermijdelijk tot een grote hoeveelheid projecten die in de categorie twijfelachtig vallen, ruim 50%. En natuurlijk zijn tijd, geld en functionaliteit niet uitwisselbaar. In 'The rise and fall of the Chaos Report Figures' hebben J. Laurens Eveleens / Chris Verhoef, VU, 2010 nog meer kritiek. Zij concluderen: "Resolution types Standish (Success/Challenged/Failed) are misleading, one-sided,..... resulting in meaningless figures".

Ervaringen

En dan is er ook nog de praktijk. Uit een eigen, kleine enquête, gehouden onder collegae-mediators, -arbiters en -gerechtelijk deskundigen trokken we de conclusie dat in ongeveer 40% van de ICT-projecten het afgewezen resultaat (nagenoeg) geheel conform de specificaties was ontwikkeld. Feitelijk was er – technisch gezien – dus geen sprake van een mislukt project. Waarom hebben deze projecten dan niet geleid tot implementatie? Uit de inventarisatie werden de volgende oorzaken het meest genoemd:

- Managementwisselingen. Dit komt veel voor. Nieuwe CEO's en CIO's gaan graag alles anders en beter doen en hebben vaak ook nog connecties uit eerdere betrekkingen.
- Noodzakelijke organisatorische en personele aanpassingen vinden niet plaats. Dit komt vooral voor bij pakketoplossingen.
- Te weinig aandacht voor ondernemingsraad en andere HRM-perikelen.
- Onkunde, desinteresse en sabotage van gebruikers.
- Discussies over kleine aanpassingen in de functionaliteit, enig meerwerk of geringe uitloop die escaleren.

Als zaken bij de rechter komen wordt in eerste instantie beoordeeld of is voldaan aan de juridische vereisten en pas daarna wordt het projectresultaat beschouwd. Het komt dan ook regelmatig voor dat men zijn gelijk niet krijgt omdat de overeenkomst hiaten vertoont of de tegenpartij niet op de in de wet voorgeschreven wijze in gebreke is gesteld. Andere vormen van geschillenbeslechting als bijvoorbeeld arbitrage verdienen vaak de voorkeur.³

² Met name 'succes' en 'mislukt', maar ook op andere punten.

³ Zie de bijdrage in deze bundel over conflictoplossing.

Agile?

In 1982 publiceerde Jaap van Rees zijn nog altijd veel geciteerde 'De methode doet het niet'.⁴ De bewuste ontwikkelmethode betrof SDM van Pandata. Na 1982 heeft de toepassing van nieuwe methodes en *frameworks* niet geleid tot een sterke afname van het aantal mislukte projecten. Vandaag doet Internet regelmatig kond dat de methode 'het nog steeds niet doet'. Dat betreft de agile-aanpak, met name het meest gebruikte framework SCRUM. Wat met Prince2, DSDM en ITIL niet lukt, gaat dat wel lukken met Agile/Scrum? Volgens Standish wel. Het onderzoeksbureau zet Scrum af tegen de waterval-aanpak en presenteert nu de volgende cijfers over de jaren 2011-2015:

<u>Omvang</u>	<u>Methode</u>	<u>Succes</u>	<u>Betwijfeld</u>	<u>Mislukt</u>
Alle projecten	Agile	39%	52%	24%
	Waterval	11%	60%	29%
Groot	Agile	18%	59%	23%
	Waterval	3%	55%	42%
Gemiddeld	Agile	27%	62%	11%
	Waterval	7%	68%	25%
Klein	Agile	58%	38%	4%
	Waterval	44%	45%	11%

'Scrum groot succes'

De vraag die direct opkomt, luidt: op welke wijze heeft Standish bij agile-projecten bepaald of het project succesvol was? Van een succesvolle implementatie is bij Standish immers slechts sprake als het project conform de initiële ramingen voor kosten en tijd én de vooraf bepaalde functionaliteit wordt opgeleverd. Bij een agile-aanpak wordt dat alles niet van tevoren vastgelegd, zeker niet voor wat betreft de functionaliteit. Dat is strijdig met het belangrijke uitgangspunt van de agile aanpak dat werkende software belangrijker is dan uitgebreide documentatie. En ook praktisch onmogelijk door de iteratieve oplevering in kleine systeemdelen, de sprints.

⁴ Maandblad Informatie, februari 1982.

Een *sprint* is een time-box van een maand of minder, waarin een werkend software increment⁵ wordt opgeleverd, dat voldoet aan de definitie van 'Done'. Elke sprint is dezelfde lengte gedurende de gehele ontwikkeltijd en volgen elkaar direct op. De *Definition of Done* beschrijft waar het resultaat van een sprint aan moet voldoen. Het is een hulpmiddel voor het team om de kwaliteit van het werk constant te houden. De Definition of Done wordt door het team zelf opgesteld en beschrijft dingen als testen, unittesten, documentatie enz.

Gelet op de definities is het niet verwonderlijk dat er binnen de agile-gemeenschap veel discussie is ontstaan over de vraag wanneer een sprint succesvol of mislukt is. Hierdoor wordt een uitspraak over het totale project nogal diffuus. In recente artikelen wordt beweerd dat 40% tot 70 % van de Scrum projecten falen, gelijk aan de ontwikkeling volgens de waterval aanpak.⁶ Naar mijn mening levert SCRUM nu al minder mislukkingen op. Vooral door de oplevering in sprints die permanent bijsturen tijdens het traject mogelijk maken. Maar voor echt betere resultaten moet er nog veel gebeuren. Dat komt vooral omdat de transformatie naar Scrum wordt onderschat. Het vraagt een andere denk- en werkwijze als in de waterval. Mede om deze reden wordt vaak voor een gedeeltelijke adaptatie van Scrum gekozen.⁷

Conclusies

Doorgaans betreffen de butts de frequentie van afstemmingsbijeenkomsten (als *daily scrums* die in plaats van dagelijks wekelijks worden gehouden). De scrumbutts maken van het raamwerk al snel een zaak van IT'ers, waarin de rol van de gebruiker wordt gereduceerd. Daardoor blijft het gebrek aan gebruikersparticipatie, net als in de waterval, de grootste faalfactor. Dat wordt mede veroorzaakt doordat gebruikers het belang van hun rol en met name die van de *product owner* nog niet inzien.⁸ Ook voortraject (offerteaanvraag en overeenkomst) is nog een leerproces gaande, immers het eindresultaat kan vooraf niet duidelijk worden vastgelegd.

Algemeen kan worden gesteld dat veel ICT-projecten mislukken, maar voor zowel waterval als agile ontbreken bruikbare criteria om dat te beoordelen. Hierdoor zijn beschikbare cijfers onbetrouwbaar. Sociaal organisatorische factoren leiden er in veel gevallen toe dat de systeemontwikkeling wordt afgebroken of dat het systeem niet wordt geïmplementeerd terwijl er technisch gezien geen sprake is van een mislukt project. De Agile/Scrum aanpak zal op termijn leiden tot minder mislukkingen en minder conflicten en juridische procedures.

⁵ Onder 'increment' wordt doorgaans verstaan: aanwas, uitbreiding, aanvulling. Het systeem 'groeit' dus bij de Scrum aanpak met iedere sprint.

⁶ Het watervalmodel is afgeleid van de traditionele manier van werken in grote projecten in de constructiebouw. De bedoeling van deze manier van werken is dat het project in verschillende fasen wordt opgedeeld. Men begint niet eerder met fase 2 dan wanneer fase 1 is afgesloten. En wanneer in een van de fasen een fout wordt ontdekt gaat men terug om die fase te corrigeren en de daaropvolgende stappen opnieuw uitvoeren. Bij de ontwikkeling van software zijn de fasen: definitiestudie/analyse, basisontwerp, technisch ontwerp/detailontwerp, bouw, testen, integratie en beheer.

⁷ Scrum-adapten noemen dat 'scrumbutts', afgeleid van 'we are using scrum, but...'.⁷

⁸ *Product owner* is een persoon die verantwoordelijk is voor het maximaliseren van de waarde van hetgeen het ontwikkelteam oplevert. Hij vertegenwoordigt de belangen van alle belanghebbenden.

Adviezen

- Schakel in omvangrijke projecten *direct vanaf de start* een onafhankelijke externe deskundige in (op afroep), teneinde mede te kunnen beoordelen of de afspraken correct worden nagekomen, om de kwaliteit van het proces *en* het softwareproduct te bewaken en om kleine geschillen op te lossen.
- Wanneer een geschil dreigt te escaleren: vraag zo snel mogelijk een deskundig advies. Oordeel vervolgens over het opgeleverde product en ga pas daarna – als het dan nog nodig is - discussiëren over ingebrekestelling, fatale data, wanprestatie, meerwerk enz.
- Voorkom escalatie en juridificering van het geschil. Te veel projecten ‘mislukken’ onterecht. Met alle gevolgen van dien. Ga dus niet te snel naar de rechter. Voordat deze zich over de inhoud buigt, wordt nagegaan of aan allerlei juridische vereisten is voldaan. Bovendien kunnen alternatieve wijze van geschillenbeslechting uitkomst bieden.