



## Agile Manifesto, explained

We zijn er zo aan gewend en zo van overtuigd van de meerwaarde dat het bijna heiligschennis is om je af te vragen of het *Manifest voor Agile Software Ontwikkeling* wel correct is. En als het correct is, waarom het dan correct is. En als het niet correct is, waarom dan niet. Is het van belang om je dit af te vragen? Jazeker! Een tweetal redenen:

- Agile is teveel een IT feestje en sluit niet goed aan bij de problematiek van vandaag. Het doel was om geen dingen meer te ontwikkelen waar geen klant op zit te wachten, of inmiddels niet meer op zit te wachten (duurt te lang), sneller gewenste diensten in de markt te kunnen zetten, te stoppen met dingen die geen klantwaarde hebben, et cetera. Helaas is het zo dat de invoering van agile deze problemen slechts deels heeft opgelost.
- Zelfs als het lukt om de IT-afdeling volledig agile te laten opereren, wordt de time to market en fitness for use begrensd door het geheel van afdelingen (ook niet-IT) en de besturing daarvan. Het is in Nederland echter nog geen bedrijf of overheidsorganisatie gelukt om volledig agile te worden (zie bijvoorbeeld <https://www.linkedin.com/pulse/agile-de-boardroom-theo-janssen/>).

In mijn analyse kom ik tot verrassende conclusies, onder andere dat het agile manifest zélf bron is voor deze problemen waar we nu nog mee kampen. Een manifest waar velen zich bijna gedachteloos achter hebben geschaard blijkt voor een groot deel onjuist en onvolledig te zijn. Om deze analyse te begrijpen is er voorkennis nodig. Je moet in ieder geval iets van *Enterprise Engineering* en gedragsverandering weten om de analyse te begrijpen. Maar voordat we het manifest analyseren laten we eerst eens kijken wat agile nu eigenlijk is.

## Wat is Agile?

Agile is een term uit het Engels dat, in de context van het organiseren van werk, al behoorlijk lang bestaat. Het betekent zoiets als 'lenig, flexibel, wendbaar'. Zelf hoorde ik dit woord voor het eerst in 1996 toen ik betrokken was bij de invoering van snelle ontwikkelmethodes voor software bij een grote overheidsorganisatie. Maar de term bestond zelfs al voor die tijd in het bedrijfsleven.

Een wendbare organisatie die met veranderingen in de economie, markt en doelgroep meebeweegt, wie kan daar op tegen zijn? Het is de ultieme vorm die een organisatie kan aannemen. In de filosofie van de Lerende Organisatie (o.a. Peter Senge) komt wendbaarheid terug in de vorm van de vijf disciplines (gedeelde visie, mentale modellen, persoonlijk meesterschap, teamleren en systeemdenken). Zodra je organisatie zich deze vijf disciplines heeft eigen gemaakt, ben je er klaar voor om alle veranderingen in de buitenwereld snel door te vertalen naar veranderingen in je eigen organisatie. Maar het is ook de gedachte achter het Toyota Production System (TPS) en het later daarop gebaseerde Lean, holacratie en de celstructuur van BSO (Eckart Wintzen) en Semco (Ricardo Semmler). Als je maar blijft leren kun je je aanpassen aan de veranderingen buiten de organisatie, die zich ook nog eens steeds sneller aan het voltrekken zijn. *Adaptability*. Dat is het idee achter wendbaarheid. Er zit wel een uitgangspunt onder, namelijk dat je als onderliggend doel hebt om als bedrijf of als overheidsorganisatie lang te blijven bestaan, langer dan de oprichter(s) aanwezig zal/zullen zijn. Lang niet iedereen hanteert dit uitgangspunt overigens. De eerste publicaties over wendbare organisaties die ik heb kunnen vinden stammen al uit de jaren '70 van de vorige eeuw. En ik heb me laten vertellen dat zelfs al in de jaren '40 hierover ideeën zijn geponeerd.

In alle afdelingen van organisaties werd en wordt er nog steeds over nagedacht hoe je zo wendbaar mogelijk kunt zijn. Zo ook in de ICT-sector. In de jaren '80 en '90 van de vorige eeuw werd er druk geëxperimenteerd met snelle, flexibele methodes om software te ontwikkelen. Denk bijvoorbeeld aan RAD, IAD, DSDM, RUP, XP, et cetera. De methodes zijn allemaal verschillend maar onderhuids hebben ze een aantal gemeenschappelijke kenmerken.

Dit bracht een zeventiental onafhankelijke software ontwikkelaars er toe om tijdens een uitje op 11, 12 en 13 februari 2001 in het skiresort The Lodge te Snowbird in Utah een manifest op te stellen voor Agile softwareontwikkeling: **Het Agile Manifest voor Agile Software Development**. Hoewel de deelnemers het niet erg eens waren met elkaar kwam er toch een document tot stand. De ontwikkelaars, die zichzelf de Agile Alliance noemden, waren op zoek naar een herziening van de softwareontwikkelingsprocessen die ze zagen als omslachtig, niet-reagerend en te gefocust op documenteren. In eerste instantie werd overigens als naam 'Adaptability Manifest' gekozen, maar omdat één van de deelnemers voor het bedrijf 'Adaptability' werkte werd gekozen voor een andere naam: Agile Manifest. Ook saillant is dat het alleen ontwikkelaars waren die hierbij betrokken waren. Geen testers, geen beheerders, geen analisten, geen managers, geen ... et cetera. Dus het is ook helemaal niet raar dat er niet is gefocust op een holistisch beeld van een volledige organisatie. Al van begin af aan is het Agile Manifest dé bron voor discussie, ruzie, vetes tussen de adepten van het heilige manifest en mensen die manieren bedenken om agile in de gehele organisatie werkbaar te krijgen. Scrum versus SAFe, wel/geen managers nodig, LeSS is eigenlijk het beste of toch niet, Agile projectmanagement slaat nergens op of is juist goed dat het er is, et cetera. **De tweedeling zat er van begin af aan in omdat het manifest expliciet alleen maar gaat over Software Ontwikkeling en alleen door software ontwikkelaars is opgesteld**. Dus als je het manifest probeert te extrapoleren vanuit software ontwikkeling naar een totale organisatie dan is het vragen om moeilijkheden. Maakt dat het manifest onbruikbaar? Nee, er staan wel degelijk goeie dingen in. Laten we eens naar de inhoud van het manifest kijken.

## Het Agile Manifest



Hoewel het behoorlijk duidelijk is wat er in bovenstaand figuur staat zijn er toch veel mensen die uit het bovenstaande concluderen dat er geen processen meer nodig zijn, dat je niets meer hoeft te documenteren, dat contracten er ook niet meer toe doen en dat er geen plannen gemaakt hoeven te worden. Dat staat er helemaal niet. Het linker rijtje is belangrijker dan het rechter rijtje. Maar dat wil niet zeggen dat je moet stoppen met het rechter rijtje. Mensen leggen het manifest graag in hun eigen voordeel uit...

Het is heel zinvol om dat linker rijtje belangrijker te achten dan het rechterrijtje. Dat voelt iedereen direct aan. Bijvoorbeeld: natuurlijk is het zo dat de manier hoe mensen met elkaar samenwerken belangrijker is dan processen en hulpmiddelen. En natuurlijk is werkende software belangrijker dan documentatie. Met de software gaan namelijk klanten geholpen worden, zonder dat zij ooit die documentatie te zien krijgen. Maar als je wilt dat in de toekomst klanten ook nog geholpen zijn (en er dus onderhoud op de software moet worden gepleegd) dan is het toch wel verstandig om ook te documenteren. Uit de software zelf kun je helaas vaak niet deduceren hoe deze werkt.

Naast de 4 waarden van het agile manifest zijn er ook nog een twaalfstal principes opgesteld die gehanteerd zouden moeten worden:

## 12 AGILE PRINCIPLES

<b>01</b> Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	<b>02</b> Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	<b>03</b> Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
<b>04</b> Business people and developers must work together daily throughout the project.	<b>05</b> Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	<b>06</b> Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
<b>07</b> Working software is the primary measure of progress.	<b>08</b> The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	<b>09</b> Continuous attention to technical excellence and good design enhances agility.
<b>10</b> Simplicity – the art of maximizing the amount of work not done – is essential.	<b>11</b> The best architectures, requirements, and designs emerge from self-organizing teams.	<b>12</b> At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Het klinkt allemaal wel logisch wat hier staat, maar laten we de verdieping eens zoeken.

### Het Agile Manifest onder de loep

- Waarom zijn er 4 waarden en niet meer? En waarom deze 4? En wat is het verschil tussen een gebod en een principe? Geen idee en ook op het internet ga je hier geen antwoord op vinden.
- Waarom zijn er 12 principes en niet meer of minder? Wat is eigenlijk een *Agile principe* precies? Ook hier ga je geen antwoord op vinden.
- Waarde 1: **Mensen en hun onderlinge interactie** boven processen en hulpmiddelen. In eerdere artikelen (bijvoorbeeld <https://www.linkedin.com/pulse/een-organisatie-nooit-verdraaid-theo-janssen/>) introduceer ik begrippen als ontologie, infologie en datalogie. Deze kennis heb je nodig om in te zien dat 'mensen en hun onderlinge interactie' onderdeel is van de ontologie van een organisatie. En dat het proces(model) volgt uit deze interactie tussen mensen. En hulpmiddelen zijn datalogisch en datalogie volgt na ontologie en infologie. Dus het gebod is correct, alleen niet volledig. Waarom is infologie niet aangehaald? Waarom worden processen expliciet genoemd (als onderdeel van de ontologie)? Het eerste gebod is wel juist maar niet volledig.
- Waarde 2: **Werkende software** boven allesomvattende documentatie. Bij de eerste waarde zaten we op een heel ander level (ontologie) naar de organisatie te kijken dan bij de tweede waarde. Werkende software en documentatie zijn beiden aspecten van de datalogie van een organisatie. Blijkbaar moeten we daar een hiërarchie in onderkennen, volgens de auteurs. Er wordt nergens aangegeven waarom dat zo zou moeten zijn. En het is ook niet correct. Een klant wordt NU misschien wel geholpen door werkende software, maar over een paar maanden wil ie wat anders en dan is het toch wel handig dat je nog kunt deduceren wat je gedaan hebt om een aanpassing correct uit te voeren. Dit gebod zorgt voor de 'veelgeprezen' houding van veel software ontwikkelaars: het moet quick and dirty. Snel de klant tevreden maken. En dan later met de gebakken peren zitten. Een foute gedachte. Er is geen hiërarchische verhouding tussen datalogische elementen!

- Waarde 3: **Samenwerking met de klant** boven contractonderhandelingen. Samenwerken met de klant is een goed idee. En dat geldt zowel voor het systeem van samenwerken als het bijbehorende gedrag. De systeemkant is ingebed in de ontologie van een organisatie, en het gedrag wat erbij past is maatwerk: iedere klant heeft een andere cultuur en hoe beter de culturen matchen des te prettiger de samenwerking is. Contractonderhandelingen gaan deels over infologische- en deels over datalogische aspecten. Dus het klopt dat samenwerken belangrijker is, want dat is iets ontologisch. Echter, het feit dat er een contract ligt op basis waarvan wordt samengewerkt komt voort uit een ontologische beslissing om afspraken te maken over de samenwerking. Dus formeel klopt het dat samenwerking met de klant belangrijker is dan contractonderhandelingen, maar samenwerken met de klant is NIET belangrijker dan het feit dat er besloten wordt een contract met de klant op te stellen. Dat is namelijk ook iets ontologisch. Er is geen onderscheid in waarden voor beide ontologische aspecten. Waarde 3 is dus deels waar.
- Waarde 4: **Inspelen op verandering** boven het volgen van een plan. Dat is een heel goed idee. We kunnen namelijk niet plannen, we hebben een planningsbias (zie [https://www.businezz.nl/zoeken/100-8304\\_50-veranderprincipes](https://www.businezz.nl/zoeken/100-8304_50-veranderprincipes)). Dus laten we die planningscapaciteiten van ons mensen niet al te serieus nemen. Een week, twee weken van te voren iets plannen dat lukt nog wel enigszins. Maar dan heb je het wel gehad. Je moet vooral goed zijn in het kunnen omgaan met veranderde omstandigheden. Dat is letterlijk de betekenis van wendbaarheid. Eigenlijk worden bij dit gebod appels met peren vergeleken. Inspelen op verandering is namelijk wendbaarheid zelve, is agile. De definitie dus. En het volgen van een plan is gedrag waar we heel slecht in zijn. Het kan niet anders dan dat waarde 4 juist is, alleen is het een tautologie: het was al waar door de manier hoe je het opschrijft. Het is hetzelfde als zeggen dat  $1 + 1 = 2$  (per definitie correct) is belangrijker dan niet-lineaire vergelijkingen oplossen (zijn we niet goed in). Tja, wat zegt deze waarde nu eigenlijk?
- **Principe 1: Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.** In dit principe staan twee dingen genoemd: het wat en het hoe. Om de klant blij te maken als hoogste prioriteit te stellen is een keuze. Treacy en Wiersema noemen dat Customer Intimacy. Dat in de principes een keuze wordt gemaakt is normaal. Dit is waar ze van uit gaan, Customer Intimacy zou in een agile organisatie de hoogste prioriteit moeten hebben. Om vervolgens aan te geven dat dat moet gebeuren via vroege, continue oplevering van software dat ook nog eens waardevol is. Wat is vroeg? Kan het ook té vroeg zijn? Is het dan nog steeds goed voor de klant? Wat is continu? Toch niet iedere seconde een app die verandert mag ik hopen? En wat is waardevol? Daar kun je hele bibliotheken over schrijven. En waarom zijn nu juist deze aspecten van belang? Ik kan nog wel meer dingen benoemen: service verlenen, aandacht schenken, een lach op het gezicht, software die eenvoudig aanpasbaar is, een mooie *look and feel*, een klein detail wat het extra mooi maakt, et cetera. Het is een compleet willekeurige invulling van Customer Intimacy. Een vakgebied waar nog veel meer te ontdekken valt dan alleen dit. Beetje mager dus, dit principe. Hoewel het zo goed begon: klanten tevreden stellen zou in een agile organisatie inderdaad prioriteit 1 moeten zijn. Dat betekent overigens niet: geven wat de klant vraagt. Henri Ford zij ooit: als ik mijn klanten had gevraagd wat ze het liefste zouden willen, dan was het antwoord 'snellere paarden' geweest. Het moet bij Customer Intimacy vooral gaan om het bevredigen van de latente behoefte van de klant, iets waar Steve Jobs meester in was. Zie voor meer informatie hierover in <https://www.linkedin.com/pulse/what-bleep-do-we-know-quality-theo-janssen/>.
- **Principe 2: Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.** In feite zegt dit principe hetzelfde als het vorige principe. Als je continu waardevolle software aan de klant wilt leveren dan zul je inderdaad in staat moeten zijn om mee te deinen met de veranderingen in de klantwensen. Het is goed om dit expliciet te benoemen maar is geen apart principe.



- **Principe 3: Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.** Tja, da's gewoon een herhaling van principe 1.
- **Principe 4: Business people and developers must work together daily throughout the project.** Blijkbaar gaat het over projecten en er zijn geen anderen bij betrokken dan alleen business mensen en ontwikkelaars. Dat is toch wel een erg beperkte kijk op software ontwikkeling. Er zijn veel meer mensen met andere functies bij betrokken en dat hoeft ook niet altijd in projectvorm te zijn.
- **Principe 5: Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.** Zo, hier staat ineens veel om op te reageren. Het gaat blijkbaar nog steeds alleen over projecten. En niet-gemotiveerde mensen, wat doen we daarmee? En hoe en wie bepaalt of iemand gemotiveerd is of niet? En moeten we gemotiveerde mensen dan maar carte blanche geven? In een eerder artikel (<https://www.linkedin.com/pulse/100-zelfsturing-bestaat-niet-theo-janssen/>) heb ik al eens aangetoond dat we teams zeker geen carte blanche moeten geven. Dat leidt tot wanorde en inteelt.
- **Principe 6: Agile processes promote sustainable development. The sponsors, developers and users should be able to maintain a constant pace indefinitely.** Er moet dus een soort oneindige heartbeat voor het opleveren van software zijn. Maar dat is niet hoe de wereld werkt. Meedeinen met de veranderende vraag van klanten betekent ook dat de snelheid van meedeinen verandert. Dát is pas agility.
- **Principe 7: Working software is the primary measure of progress.** Werkende software, wat dat ook moge zijn, bepaalt wat de primaire maat van voortgang is. Als je echt agile bent dan zou dat de mate van klanttevredenheid moeten zijn. Bovendien lijkt dit op wat bij principe 1 wordt beweerd – het is dubbelop.
- **Principe 8: The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.** Tja, inmiddels weten we door COVID-19 dat het wel anders kan, dat samenwerken. Het hybride werken (<https://www.deltacc.nl/kennis-delen/kennisgebieden/hybride-samenwerken-met-lego>) heeft definitief z'n weerslag gevonden in het hedendaagse werken. Soms is face-to-face effectiever en soms is online veel effectiever. Dus dit principe is niet altijd waar. Als de bedoeling achter dit principe is om te stoppen met eindeloze mailwisselingen met alle bijbehorende communicatiestoornissen dan zou dat het principe moeten zijn: stoppen met e-mail. Maar dat staat er niet. Helemaal stoppen met e-mail is overigens ook een utopie. *Traceability* is erg belangrijk, zowel om te kunnen leren van fouten uit het verleden als om aan te kunnen tonen wat je waarom gedaan hebt.
- **Principe 9: Continuous attention to technical excellence and good design enhances agility.** Een constante focus op verbeteren, dát zou het principe moeten zijn. Of dat nou om techniek of design gaat, of om het proces van samenwerken, de manier van omgaan met risico's, het ophalen van informatie van gebruikers et cetera gaat, dat maakt niet uit. Een agile organisatie is een lerende organisatie. Principe 9 is in principe correct maar moet naar een hoger level getrokken worden mijns inziens.
- **Principe 10: Simplicity – the art of maximizing the amount of work not done – is essential.** Wat tussen de – tekens staat is echt grote onzin. Laten we dan maar helemaal niks meer doen, dan zijn we blijkbaar perfect bezig. Als we die zin weglaten dan lijkt er iets te staan als 'eenvoudig te gebruiken software' oftewel gebruikersvriendelijke software. Dat is onderdeel van principe 1, dus redundantie. Als er wordt bedoeld 'alles zo simpel mogelijk aanvliegen' dan bestaat er het risico om ingewikkelde dingen te simpel aan te vliegen. Soms zijn dingen nu eenmaal ingewikkeld. Het platslaan in modellen en eenvoudige processen helpt om de snelheid te verhogen, maar de *devil* is nu eenmaal vaak *in the details*. Zeker als het gaat om

het tevreden stellen van klanten. Dit is een principe dat goed voelt, maar waarbij er wel enig risico bestaat tot over simplificatie. Oppassen dus!

- **Principe 11: The best architectures, requirements, and designs emerge from self-organizing teams.** Dat hoeft niet zo te zijn. In de praktijk gaan teams alleen nog maar doen wat ze zelf goed vinden zonder rekening te houden met teams die verderop iets aan het ontwikkelen zijn waar ze een verbinding mee hebben en ook niet aan het grotere belang van de gehele organisatie. Na een paar jaar een team z'n eigen gang te laten gaan komt een organisatie met een technical debt te zitten waar je U tegen zegt. Schrappen dit principe! Zie bijvoorbeeld <https://www.linkedin.com/pulse/100-zelfsturing-bestaat-niet-theo-janssen/>.
- **Principe 12: At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.** Ha, wederom iets over een lerende organisatie. Dat heeft overlap met principe 9. En er staat iets cruciaals: gedragsverandering. En dit is eigenlijk het belangrijkste: het creëren van een agile mindset. Een aspect waar vrijwel nooit expliciet aandacht naar uit gaat bij agile transformaties.

## GROWTH MINDSET    FIXED MINDSET



### Wat zou wél een goed manifest zijn?

Zoals gezegd is het niet vreemd dat het agile manifest incorrect en onvolledig is. Het zijn ontwikkelaars geweest die iets hebben opgeschreven over een wendbare manier van software ontwikkelen. En dan het liefst op een dusdanige manier zoals de developers het zelf prettig zouden vinden. Toch zitten er wel goeie dingen in, maar Stephen Denning is er m.i. veel beter in geslaagd met *The Age of Agile* om de essentie van agile vast te leggen. Nu volgt een korte samenvatting van dit boek.

Er blijken drie wetten te zijn waaraan Agile organisaties voldoen:

- De Wet van het Kleine Team
- De Wet van de Klant
- De Wet van het Netwerk

### De Wet van het Kleine Team

Kleine, multidisciplinaire teams leveren regelmatig werk op in kleine hoeveelheden, dus geen grote bulk na een lange periode. De hoeveelheid werk die onderhanden is, is gelimiteerd zodat het de juiste hoeveelheid aandacht krijgt. De belangrijkste dingen eerst, en dan zien we daarna wel weer

wat d n belangrijk is. In autonome teams wordt de klus iedere keer geklaard en opgeleverd aan klanten. Die op hun beurt iedere cyclus opnieuw het team van feedback voorzien. Er zijn geen onderbrekingen meer in het team en geen wachtstanden. Er is continue focus op de belangrijkste issues die op d t moment spelen. In dagelijkse stand-ups vertelt het team elkaar wat ze aan het doen zijn, waar ze tegenaan lopen en wat ze willen gaan doen. Dit alles vereist onderling vertrouwen en radicale transparantie: iedereen weet wie waarmee bezig is en hoe de vorderingen zijn. Teamleden zijn niet bang om zo transparant te zijn want wat er ook gezegd wordt: het wordt nooit tegen ze gebruikt. Door middel van voortdurende evaluaties wordt teruggekeken op de gang van zaken in de afgelopen cyclus. Enkel met als doel om het voortaan n g beter te doen en n g meer klantwaarde te leveren.

Aspecten die van belang zijn voor de prestaties van zo'n team:

1. *Psychologische veiligheid*: kunnen we in dit team risico's nemen zonder dat we ons onzeker voelen of voor schut worden gezet?
2. *Betrouwbaarheid*: kunnen we van elkaar verwachten dat we op tijd werk van hoog niveau afleveren?
3. *Structuur en duidelijkheid*: zijn de doelstellingen, rollen en uitvoeringsplannen in ons team helder?
4. *Betekenis van het werk*: werken we aan iets wat voor ons allemaal persoonlijk belangrijk is?
5. *Impact van het werk*: zijn we ervan overtuigd dat ons werk ertoe doet?
6. *De samenstelling van het team*: er is niemand die  lle competenties heeft die nodig zijn om het team te laten presteren, teamleden moeten elkaar aanvullen op hard- en soft skills.



## De Wet van de Klant

In de 21e eeuw hebben kenniswerkers een centrale rol gekregen bij de creatie van de innovatie die nodig is om klanten aan te trekken en te behouden. En internet heeft alles veranderd. Er voltrok zich een historische revolutie in het commerci le zwaartepunt. De macht is verschoven van verkoper naar koper. Niet het bedrijf, maar de consument is voortaan het centrum van het commerci le universum. Nu de klant het op de markt voor het zeggen heeft, is het voor bedrijven heel riskant om zichzelf centraal te blijven stellen. Bedrijven moeten bedenken wat de problemen van de klanten zijn en oplossingen vinden waarmee ze de klant kunnen verrassen om hem een fantastische



gebruikerservaring te bieden. En als de klant geen voortreffelijke gebruikerservaring heeft moet je de interne systemen en processen aanpassen om een betere uitkomst voor de klant te genereren. Een slogan als 'de klant staat altijd centraal' mag geen slogan meer zijn. Tegenwoordig staat de klant écht centraal. De klant is de baas! En niet alleen de marketing- en salesafdeling heeft oog voor klanten. Ook de operationele afdelingen en zelfs alle backofficefuncties hebben helder zicht op de uiteindelijke klant; ze passen hun systemen en processen voortdurend aan met het oog op een grotere bijdrage aan de uitkomst voor die klant. Als systemen en processen niet meer bijdragen aan klantwaarde, moet de vraag worden gesteld waarom ze überhaupt bestaan. Bedrijven moeten wel voor ogen houden dat de klantwaarde uiteindelijk ook voor de organisatie te gelde moet worden gemaakt. Hoewel men zich primair toelegt op de rol van de klant, onderkent het bedrijf dat het geld moet verdienen om te overleven. Geld verdienen is echter het resultaat en niet het doel.

Aspecten die van belang zijn bij het omgaan met klanten:

1. *Doelgroep*: je moet zorgvuldige keuzes maken op wie je je inspanningen richt
2. *Experimenteer voortdurend*: zoals in Lean Startup: vernieuw continu, leer en stel bij
3. *Werk samen*: focus op waar je goed in bent en werk samen met anderen die je product/dienst vervolmaken
4. *Maak je product kneedbaar*: rigide producten kunnen niet snel aangepast worden aan de veranderende wensen van klanten, dus zorg dat het product wél kneedbaar is
5. *Focus*: mik op simpele dingen om klanten gelukkig te maken
6. *Innoveer stapsgewijs*: lanceer je product of dienst met de belangrijkste features die je primaire klant wil hebben en voeg dan selectief features toe via upgrades
7. *Evalueer*: check dat elke toegevoegde feature klanten écht gelukkig maakt
8. *Je moet bereid zijn om mensen teleur te stellen*: Amazon maakt alleen kopers gelukkig, niet hun leveranciers ...
9. *Lever sneller waarde*: tijd is voor klanten vaak een belangrijke factor
10. *Houd rekening met individuele wensen*: op maat gemaakte producten en diensten is precies wat klanten tegenwoordig willen



## De Wet van het Netwerk

In grote netwerken samenwerken wordt vaak als onmogelijk gezien vanwege de beperkingen die voortvloeien uit wat sociologen het 'getal van Dunbar' noemen. Dit is het getal dat in de jaren negentig van de vorige eeuw door de Britse antropoloog Robin Dunbar werd geponeerd als de cognitieve limiet van het aantal mensen met wie een individu een stabiele, sociale relatie kan onderhouden. Dit gemiddelde maximumaantal is 148, vaak afgerond op 150, onder de voorwaarde dat de leden van die groep bij de groep willen horen. Door het idee dat grote groepen mensen maar moeilijk met elkaar kunnen samenwerken zijn organisaties op hiërarchische leest geschoeid. Zuilen van verantwoordelijkheden, verticaal maar soms ook horizontaal en verticaal tegelijk, geven een gevoel van controle maar weerhouden mensen feitelijk om goed samen te werken. Efficiënte netwerken zijn echter van eminent belang om goed en snel te kunnen samenwerken en de klant te leveren wat hij nodig heeft.

Aspecten die van belang zijn bij het netwerken:

1. *Het netwerk heeft een meeslepend doel:* het is het van nature inspirerende doel om waarde te leveren aan de klant
2. *Het netwerk bestaat uit kleine groepen:* succesvolle organisaties zijn georganiseerd rond kleine teams die samenwerken om in sterk van elkaar verschillende contexten impact te hebben, en deze teams worden verenigd door een losjes gekoppeld netwerk dat ze in staat stelt elkaar om hulp te vragen en te leren van de aanpak van andere teams
3. *De groepen zijn daadgericht:* de teams willen hun ideeën in de praktijk brengen en doen niet aan abstracte kennis, ze zijn gericht op het veranderen van de wereld die ze een nieuwe koers willen geven
4. *Het netwerk is de som van de kleine groepen:* de groepen zijn het netwerk
5. *Het juridische kader van het netwerk blijft op de achtergrond:* juridische processen zoals compliance, risk, toezichthouders, etc. blijven bestaan maar verdwijnen wel naar de achtergrond omdat de belangrijkste beslissingen worden genomen door het op competentie gebaseerde netwerk



Tot zover de korte samenvatting van het boek *The Agile of Agile*.

## Agile mindset

Naast de drie genoemde wetten is er een samenspel van een aantal gedragingen die gezamenlijk zorgen voor een zogenaamde agile mindset. Zonder te pretenderen dat ik volledig ben horen hier bijvoorbeeld bij:

- Focus op klantwaarde en ruim waste op: zorg dat zoveel mogelijk elementen uit je bedrijfsprocessen worden geschrapt. Dit heet het opruimen van *waste*. Alleen als er klantwaarde wordt gecreëerd (direct of indirect, zie ook <https://www.linkedin.com/pulse/what-bleep-do-we-know-quality-theo-janssen/>) dan is het een zinvolle activiteit. Anders niet. Denk bijvoorbeeld eens aan beoordelingsgesprekken: wat is daar de klantwaarde van? Of een procedure maken en hanteren voor het aanvragen van vakantiedagen. Wat heeft de klant daar aan? Dit kan behoorlijk ver gaan. Veel dingen waarvan we al tientallen/honderden jaren maar aannemen dat ze belangrijk zijn en dat ze gedaan moeten worden zijn dat eigenlijk niet (meer). Schrappen maar!
- Just in Time (JIT): het heeft geen enkele zin om een plan te maken voor langere periodes (bijvoorbeeld 2-3 jaar). Mensen kunnen helemaal niet plannen (al eerder aangehaald dat wij een planningsbias hebben) want er verandert gaandeweg zoveel dat je dat van tevoren toch niet kan bedenken. Wat je wel kunt doen is zo wendbaar mogelijk te zijn om te anticiperen op die veranderingen. Stop dus met het maken van (gedetailleerde) planningen. Maak iets op het moment dat je 't nodig hebt en doe iets op 't moment dat het moet.
- Doe de belangrijke dingen eerst: dat klinkt meer dan logisch, maar toch zijn we vaak alleen bezig om e-mails weg te werken, de acties uit de actielijst uit te voeren en soms nutteloze dingen die onze manager heeft gevraagd aan te leveren.
- Neem gezamenlijk verantwoordelijkheid voor het eindresultaat: Deze houding bevordert de samenwerking, en ook als iemand er een keer niet is en er moet iets opgeleverd worden of een besluit genomen worden dan gaan we niet allemaal zitten wachten totdat die ene persoon weer aanwezig is maar dan nemen anderen de taken over. Dat betekent overigens niet dat iedereen ook verantwoordelijk is voor het werk dat iemand anders doet. Gedeelde verantwoordelijkheid bestaat niet, weten psychologen al lang. Alleen voor het eindresultaat. Zie een eerdere post van mij: <https://www.linkedin.com/pulse/100-zelfsturing-bestaat-niet-theo-janssen/>.
- Monetaire beloning schrappen: een monetaire beloning voor geleverd werk is een afgeleide en geen doel. Ook hier heb ik al eens in een post over geschreven met als basis het werk van Dan Ariely (<https://www.amazon.nl/Predictably-Irrational-Hidden-Forces-Decisions/dp/0061353248>). Stop dus bijvoorbeeld met functiegebouwen (waste).
- Creëer een extreem klantgerichte mindset binnen de organisatie. Nogmaals: dat betekent niet om maar te doen wat de klant vraagt. Het gaat om het vinden van de latente behoefte van klanten.
- Verleg de aandacht van het bekende naar het onbekende, van bestaande producten naar nieuwe producten. Een agile organisatie leeft van innovatie, van vernieuwing, van continu leren. Een agile organisatie is een lerende organisatie.

## Vervolg

Het agile manifest is enorm succesvol geweest en heeft vele mensen geïnspireerd om software ontwikkeling anders te doen dan voorheen gebeurde. Alleen is die inspiratie, wat mogelijk het doel is geweest van het manifest, verworden tot een soort bijbel waaraan je je moet houden voor velen. We weten ook uit de praktijk dat het manifest niet langer houdbaar is omdat het te beperkt is opgesteld, namelijk alleen voor software ontwikkeling. Om als bedrijf/organisatie agile te zijn moet niet alleen de software afdeling agile zijn. De gehele organisatie zal er aan moeten geloven. Er moet

een nieuw manifest voor agile komen. Eén die alle aspecten van de bedrijfsvoering raakt, en niet alleen het ontwikkelen van software.

Ik ben er voorstander van om een nieuw manifest te starten met een aantal axioma's:

1. De mens is een goed wezen van nature (Rutger Bregman – De meeste mensen deugen)
2. Continu veranderen is de enige constante
3. Een organisatie is een sociaal systeem
4. Een verandering is pas een verandering als er ander gedrag waarneembaar is

Op basis van deze axioma's kan een manifest worden opgesteld waar de drie wetten van Denning en de aspecten van de agile mindset in terugkomen. Dit is slechts een startpunt. Wie wil er meeschrijven aan een nieuw manifest?