Leuk dat je mijn boek hebt gedownload. Ik hoop dat het je waardevolle inzichten oplevert voor de praktijk.

Heeft het boek je verder geholpen? Dan wordt een vrijwillige bijdrage enorm gewaardeerd. Dat kan eenvoudig via de QR-code hieronder.



Of koop een fysiek boek via www.consultancyenzo.nl

DAT IS NIET WAT IK BEDOELDE!

Van vage wens naar de juiste oplossing

Michiel Lensink

© Michiel Lensink, 2026

Alle rechten voorbehouden. Gebruik de kennis uit dit boek naar eigen inzicht. Kopiëren van de letterlijke tekst is niet toegestaan zonder schriftelijke toestemming.

Eerste uitgave: September 2026

(Extra) boek kopen? www.consultancyenzo.nl

ISBN (fysiek boek): 9789465468570

Inhoudsopgave

Waarom ik dit boek geschreven heb **7**

Over informatieanalyse **10**

Wat is informatieanalyse? 11
Waarom projecten mislukken 13
Informatieanalyse is een vak 16
Is informatieanalyse altijd nodig? 18

Informatieanalyse in 3 fasen **20**

Fase 1: Van behoefte naar richting 23
Een verkennend gesprek 25
Weet voor wie je het doet 26
De wens achter de wens 30
(Visueel) ontwerpen op hoofdlijnen 33
Opdelen en prioriteren 37

Fase 2: Uitwerken van de details 41
Bouwblokken uitwerken 41
De gezamenlijke toets 46

Fase 3: Uitvoeren en bijsturen 51
Van analyse naar werkelijkheid 51
Feedback ophalen 53
Nazorg hoort er ook bij 56

Een uitgewerkt voorbeeld 58

Gesprekstechnieken **64**

Tekenen 65
Gedragsgericht doorvragen 68
5x waarom 70
Focus op emoties 73
Toekomstbeeld schetsen 76

Vraag naar eerdere oplossingen 79
Stel, het is mislukt 82
Samenvatten en toetsen 86

Technieken die je werk structureren **90**

Het stroomdiagram 92
MoSCoW: Kiezen als alles belangrijk lijkt 98
Persona's: Omdat 'de gebruiker' niet bestaat 101
INVEST: De kwaliteitscheck voor bouwblokken 104
Artificial Intelligence: Je assistent 109

Basiskennis techniek **112**

API's: Hoe systemen met elkaar praten 114
Databases: Het geheugen van het systeem 116
Omgevingen (OTAP) 117
Logging: De zwarte doos 118
Frontend, Backend en microservices 118
En dan heb je ook nog… 119

Nog wat laatste tips **120**

De vergeten eisen (Het werkt wel, maar...) 121
De kunst van 'Nee' zeggen 122
Waar laat je de waarheid? 122

Het is nu aan jou **124**

Bronnen & leestips **126**

Waarom ik dit boek geschreven heb

Ooit in projecten gewerkt? Je hebt hard gewerkt en alles verliep (grotendeels) volgens plan. En toch is de uitkomst niet wat er bedoelt was. De oplossing sluit net niet aan bij de verwachting, de samenwerking liep stroef of er zijn onderweg zoveel aanpassingen gedaan dat het oorspronkelijke doel nauwelijks nog zichtbaar is.

Dat is precies waar analyse het verschil maakt. Als je blind vertrouwt op wat iemand zegt te willen, of te snel aanneemt dat je de vraag wel snapt, eindig je vaak met iets dat goed bedoeld is, maar weinig oplost. Dit boek is er voor iedereen die dat wil voorkomen. Of je nu inhoudelijk verantwoordelijk bent, gesprekken voert met klanten, of gewoon vaker denkt: “Hadden we dit niet slimmer kunnen aanpakken?”

Zelf begon ik begin 2000 in de softwareontwikkeling. Ik was vooral bezig met programmeren. Ik kreeg opdrachten en plannen op mijn bureau, maar ik had weinig zicht op het waarom. Als het resultaat niet werkte, of de klant toch iets anders wilde, dan pasten we het aan. En nog eens…

Toen ik de overstap maakte naar een analyserende rol, dacht ik: nu ga ik het beter doen. Maar in het begin maakte ik precies dezelfde fout als mijn voorgangers. Ik schreef netjes alles op wat de klant zei, knikte tijdens overleggen maar zat daarna achter mijn bureau alsnog met vragen.

Het kwartje viel toen een klant tegen me zei: “Jij bent geen analist, jij bent een notulist.”

Dat was niet leuk om te horen, maar hij had wel gelijk. Ik noteerde wat er gezegd werd, maar ik stelde geen kritische vragen. Ik begreep onvoldoende waarom iets belangrijk was en vertaalde wensen naar documenten, in plaats van naar oplossingen.

Vanaf dat moment gooide ik het roer om. Ik dook in de theorie, maar merkte dat veel boeken onnodig ingewikkeld of technisch waren. Daarom ontwikkelde ik mijn eigen aanpak: eenvoud, visueel werken en doorvragen tot je de kern hebt. Dat bleek veel effectiever.

Wat je van dit boek kunt verwachten
In dit boek deel ik wat ik geleerd heb. Geen dikke, theoretische handboeken, maar praktische tips en simpele technieken die echt werken. Of je nu van nature analytisch bent of niet, dit boek helpt je om gerichter te denken. Dat begint met luisteren, doorvragen en begrijpen wat er speelt. Als dat lukt, wordt samenwerken ineens een stuk leuker en het resultaat beter.

De focus ligt op analyse in een flexibele omgeving, waar je altijd kunt bijsturen. De inzichten zijn echter breder toepasbaar, overal waar mensen samenwerken is dit boek toepasbaar. Omdat mijn achtergrond in de ICT ligt, komen veel voorbeelden uit de softwarehoek. Maar laat je daardoor niet afschrikken: ik heb het zo geschreven dat je de inhoud makkelijk kan vertalen naar jouw eigen praktijk, of je nu software bouwt, een proces in je werk wil verbeteren of een evenement organiseert.

Dit boek is bewust eenvoudig geschreven
Het zal je opvallen dat ik kies voor korte zinnen en moeilijke woorden heb geprobeerd te vermijden. Dat is geen toeval, maar een bewuste keuze. In een vak als informatieanalyse worden dingen al snel onnodig ingewikkeld gemaakt.

Vergelijk maar eens:

"Binnen de context van agile ontwikkeling is het cruciaal dat requirements iteratief geëvalueerd worden om adaptief in te kunnen spelen op veranderende businessbehoeften."

Of gewoon:

"Blijf kritisch op wensen, want wat de klant vandaag zegt, kan morgen veranderen."

Ik kies voor het tweede.

Over informatieanalyse



Over informatieanalyse

Er bestaan veel misverstanden over informatieanalyse. Dat het alleen voor IT'ers is, dat het dagen werk kost en dat het overbodig is in moderne manieren van werken.

In dit hoofdstuk wil ik deze misverstanden wegnemen. Informatieanalyse is overal toepasbaar, of je nu IT'er bent of officemanager die een evenement moet organiseren. Dankzij analyse begrijp je waarom en voor wie je iets doet en worden de wat en wanneer altijd beter.

In dit hoofdstuk kijken we wat het vak precies inhoudt, waarom projecten zonder analyse zo vaak mislukken of langer duren dan noodzakelijk en wanneer je het inzet.

Wat is informatieanalyse?

Als je het heel simpel zegt: informatieanalyse is verduidelijken.

Je pakt iets wat nog vaag, onduidelijk of half uitgesproken is, en je maakt het bruikbaar. Er komt een wens, een idee of een klacht op tafel:

- "Het moet allemaal wat makkelijker."
- "We willen dit voortaan zelf kunnen regelen."
- "Er moet gewoon meer overzicht komen."

Iedereen knikt, maar jij denkt: "Waarom? Wat bedoelen we hier precies mee? En wat moet er dan concreet gebeuren om dat te bereiken?"

Precies daar begint jouw werk.

Geen lijstjes, maar betekenis
Een goede analist is geen notulist die simpelweg opschrijft wat iemand zegt. Je luistert, je vraagt door en je ontrafelt wat er echt bedoeld wordt.

- Je vertaalt vage uitspraken naar concrete doelen.
- Je maakt van woorden gedrag.
- Je zet een gevoel om in actie.

Iemand kan zijn verhaal perfect vertellen, zonder dat het idee in de praktijk werkt. Wat logisch klinkt voor de één, kan rampzalig zijn voor het werkproces van de ander. Daar ligt jouw vakmanschap: de vertaalslag maken van mooie woorden naar een werkend systeem of proces.

Voorbeeld: "Het moet klantvriendelijker"

Een salesmanager zegt: “Het systeem moet klantvriendelijker, want klanten raken de weg kwijt.”

De beginnende analist noteert: "Eis: Systeem moet klantvriendelijk zijn." De bouwer kan hier niets mee. Als ervaren analist vraag je door:

- “Wat gebeurt er nu waardoor je zegt dat het niet vriendelijk is?”
- “Wanneer loopt de klant vast?”
- “Wat wil je dat ze wel kunnen doen?”

Na een paar vragen blijkt dat nieuwe klanten hun offerte niet kunnen terugvinden op de website en dan afhaken of gaan bellen.

De echte wens is dus niet “klantvriendelijkheid”, maar: “Een klant moet na inloggen binnen drie klikken zijn offerte kunnen openen, zonder hulp van support.”

Maar is dat geen businessanalyse?
De termen businessanalyse en informatieanalyse worden vaak door elkaar gebruikt. Toch zit er verschil in focus.

Simpel gezegd, businessanalyse kijkt naar het 'waarom', en informatie-analyse kijkt naar het 'hoe'.

	Businessanalyse	**Informatieanalyse**
Focus	Doelen, processen, strategie, geld	Eisen, bronnen, data, oplossingen
Resultaat	Kansen, businesscases	Heldere en toetsbare oplossingen
Kernvraag	"Waarom doen we dit en wat levert het op?"	"Wat moet de oplossing gaan doen?"

Mijn definitie: Je doet het allebei. In dit boek (en in de praktijk) vervaagt die grens. Als goede informatieanalist moet je namelijk ook snappen waarom iets nodig is voordat je kunt bedenken hoe het moet werken. Je hoeft geen businesscases te schrijven, maar je moet wel snappen hoe iets past binnen iets bestaands, zoals een manier van werken.

Daarom gebruik ik in dit boek de term informatieanalist voor het hele spectrum: van het snappen van de waarde tot het uitwerken van het gedrag.

Waarom projecten mislukken

Alles liep volgens plan. Je had de wensen opgehaald, het team draaide lekker, de voortgang werd netjes bijgehouden en de deadline werd gehaald. Elke stap leverde zichtbaar resultaat op.

En toch… niemand is echt blij met het eindresultaat. Niet de gebruiker, niet de organisatie, en jijzelf eigenlijk ook niet.
Gefrustreerd roep je: "Maar dit was toch precies wat jullie vroegen?"

De reacties:

- "Ja, we dachten dat dit handig zou zijn, maar nu het er is gebruiken we het eigenlijk niet."
- "Het doet precies wat we hebben afgesproken, maar het sluit toch niet aan op ons proces."

Voor je het weet zit je in 'fase 2', wat eigenlijk gewoon een dure herstelronde is met meer frustratie en minder budget voor het echt leveren van waarde.

Wat gaat hier mis?
Of je nu flexibel werkt of met een in beton gegoten projectplan: het probleem zit vaak niet in hoe je bouwt, maar in waar je van uitgaat. Vaak wordt er gebouwd op basis van een aanname die niet genoeg onderzocht is, of een vraag die te letterlijk genomen is.

Dit is precies waarom informatieanalyse geen 'moetje' is, maar de basis van succes. Analyse begint niet bij de oplossing, maar bij de vragen: *Waarom?* en *Voor wie?* Pas als je die antwoorden hebt, ga je nadenken over het *Wat* en *Wanneer*.

Goede analyse levert drie dingen op:

1. **Het voorkomt verspilling:** Je bouwt meteen het juiste. Verwachtingen worden vooraf afgestemd in plaats van achteraf gecorrigeerd (wat altijd duurder is).
2. **Concrete resultaten:** Het levert eisen op die je direct kunt bouwen en testen.
3. **Vertrouwen:** Misschien wel het belangrijkste. Vertrouwen bij je klant dat hij begrepen wordt, en vertrouwen bij je team dat ze het juiste aan het doen zijn.

Voorbeeld: De reispas

Een paar jaar geleden schoof ik aan bij een lopend project waar de sfeer wat stroef was. De startvraag leek glashelder:

"We willen dat medewerkers snel en eenvoudig zelf nieuwe reispassen kunnen aanvragen."

Klonk logisch. Het team bouwde het, het werkte en het werd opgeleverd. Maar een week later stroomden de klachten binnen vanuit de administratie:

- "Collega's die al een pas hebben, vragen er nu voor de zekerheid nog een aan. Dat mag niet, dus wij moeten nu alles handmatig controleren op dubbelingen."
- "Mensen kiezen nu zelf tussen 'vrij reizen' en een 'trajectabonnement', maar ze kiezen altijd de duurste optie, ook als dat niet bij hun functie past."

De oplossing werkte prima (je kon snel een pas aanvragen), maar de administratieve chaos was verdubbeld.

Had iemand aan het begin doorgevraagd:

- "Voor wie is dit bedoeld?" → Alleen mensen zonder pas.
- "Mogen die alles kiezen?" → Nee, dat hangt af van hun functie.

Dan was de behoefte heel anders geformuleerd: "We willen medewerkers zonder reispas een abonnementsvorm aanbieden die past bij hun functie." Dat had direct een beter systeem opgeleverd en fouten in de administratie gescheeld.

Informatieanalyse is een vak

Het was halverwege de ochtend. Ik zat net achter mijn laptop toen er een mailtje binnenkwam over een nieuw project.
Het begon hoopvol: er was al een duidelijk doel en we konden gebruikmaken van een bestaand proces. “De wensen zijn wel helder”, stond er. En toen kwam die ene zin: “Of jij daarnaast even wilde ophalen wat er nog meer nodig was.”

Ik las die zin nog een keer. “Wel helder... daarnaast... even...” Vooral dat woordje ‘even’ bleef hangen. Het klinkt alsof analyse iets is wat je er wel even tussendoor doet, als je toch al aan het meeschrijven bent.

Analyse doe je er niet ‘even bij’
In veel teams gebeurt analyse dus tussen de bedrijven door. Er worden wat vragen gesteld, er ontstaat een globaal idee, er wordt een wensenlijstje gemaakt en voor je het weet ben je gestart. Zonder verdieping.

Totdat iemand halverwege de bouw vraagt:

- “Wat bedoelen ze hier nou precies mee?”
- “Is dit wel het belangrijkste?”
- “Moet dit nu, of kan het later?”
- “Waarom doen we dit eigenlijk?”

Als niemand echt eigenaarschap pakt over de analyse, eindig je met herstelwerk dat je vooraf had kunnen voorkomen. Iedereen kan analyseren, daar heb je echt geen visitekaartje met ‘Informatieanalist’ voor nodig. Maar iemand moet wel de tijd nemen en de vaardigheden hebben om het goed te doen. Analyse vraagt aandacht, vooraf en tijdens de uitvoering. Het gaat over scherp blijven op de bedoeling.

Kan de projectleider dat niet gewoon doen?
Een rol als projectleider, product owner of manager betekent dat je op waarde, richting en prioriteit van het eindresultaat stuurt. Dit betekent alleen niet automatisch dat de tijd, kennis of focus aanwezig is om alle achterliggende belangen, afhankelijkheden en uitzonderingen boven water te krijgen.

Daarom maakt een goede analist het verschil. Niet om de rol van de projectleider over te nemen, maar om de diepte in te gaan: wat moet er precies gebeuren, waarom en voor wie? Dus ja, zij kunnen dit zeker ook doen, maar vervullen dan dus ook de rol van informatieanalist.

Dus samenvattend, wat maakt het een vak?
Het klinkt makkelijk: "gewoon even vragen wat ze willen". Maar goede analyse is een vak apart. Als analist moet je namelijk:

- Luisteren zonder meteen in oplossingen te denken (en dat is moeilijker dan je denkt).
- Vragen durven stellen tot het ongemakkelijk wordt.
- Doorhebben wat mensen bedoelen, ook als ze iets heel anders zeggen.
- Botsende belangen zichtbaar durven maken.
- Gedrag en behoeften vertalen naar concrete acties.
- Vage ideeën omzetten in duidelijke, werkbare stappen.
- Toetsen of wat je denkt te begrijpen, ook echt klopt.

Dat leer je niet door het “een keertje te doen” maar door het trainen van drie fundamentele eigenschappen: nieuwsgierigheid, verbeeldingskracht en communicatiekracht.

Is informatieanalyse altijd nodig?

Je kunt een project gewoon starten en onderweg wel zien waar je uitkomt. Zeker in de softwarewereld is dat al jaren de trend: “We doen het lekker Agile.”

Die gedachte is verleidelijk: niks vastleggen, gewoon beginnen met bouwen, elke paar weken even samen kijken of je goed bezig bent, wat aanpassen en weer door. Deze gedachte is hardnekkig maar toch echt fout.

Projecten werken beter met een goede analyse vooraf omdat je dan weet waar je je flexibiliteit op inzet. Flexibel zijn is waardevol en experimenteren mag, maar je moet wel weten *waarom* je iets doet, *voor wie* het bedoeld is, *wat* het moet opleveren en *wanneer* je iets doet.

Analyse is je basis. Het zorgt dat je de juiste vragen stelt voordat je de verkeerde kant op rent. Het voelt misschien als vertraging aan de start, maar het voorkomt kostbaar gedoe achteraf. Een denkfout die je ontdekt tijdens een gesprek aan het bureau kost je vijf minuten en een streep door een tekening op je whiteboard, diezelfde fout pas herstellen nadat je klaar was? Dat kost je een veelvoud aan weggegooide uren en vertrouwen.

Uit onderzoeken is gebleken dat gemiddeld 64% van uitgewerkte ideeën zelden of nooit worden gebruikt. Dat is pure verspilling. Als jij door scherpe vragen te stellen al het overbodige er vooraf uit filtert, heb je je eigen salaris al terugverdiend voordat er begonnen wordt.

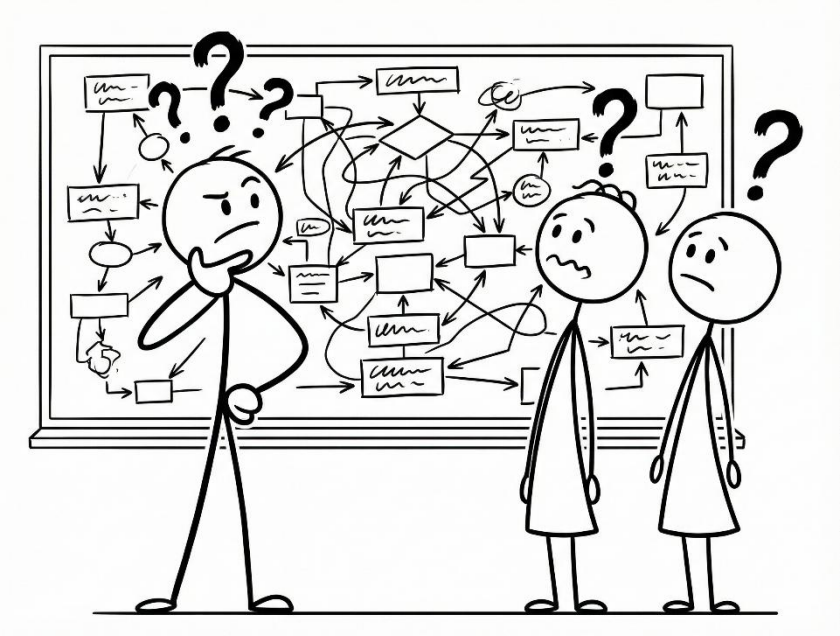
Dus ja, informatieanalyse is altijd nodig, alleen niet altijd even uitgebreid. Soms weet je binnen een paar uur genoeg, andere keren is het een fulltime baan.

Wat is Agile eigenlijk?
Over dat woord 'Agile' dat ik op de vorige pagina gebruikte. Ik noem het specifiek omdat dit vaak als excuus gebruikt wordt om geen analyse te doen. En dat is zonde, want Agile is een manier van werken waarbij je stap voor stap waarde levert, inspeelt op verandering en continu samenwerkt om het resultaat te verbeteren. Het is geen methode, maar een denkwijze die draait om:

- **Snelle deelopleveringen**: Je levert werk in kleine stukjes op, zodat je snel kunt bijsturen.
- **Samenwerking**: Je betrekt actief de mensen voor wie je het maakt.
- **Flexibiliteit**: Plannen mogen veranderen als de werkelijkheid daarom vraagt.
- **Waarde centraal**: Je werkt niet aan wat 'ooit bedacht was', maar aan wat nu het meeste oplevert.

Zoals je ziet: daar past informatieanalyse prima bij.

Informatieanalyse in 3 fasen



Informatieanalyse in 3 fasen

Misschien denk je inmiddels: "Nu weet ik het wel, informatieanalyse is belangrijk." Dat ik dit zo vaak herhaal, heeft een reden. Ik wil dat niemand meer denkt: Mwah, soms nodig", maar voelt: "Dit is onmisbaar". Als je daardoor gemotiveerd bent om er beter in te worden, is mijn missie geslaagd.

Nu gaan we de diepte in. Want zonder gestructureerde analyse raak je onderweg ongemerkt de belangrijkste dingen kwijt:

- Je verliest uit het oog wie je klant nu precies is.
- De 'waarom' achter de vraag verwatert.
- De samenhang tussen de verschillende werkzaamheden verdwijnt.
- Je raakt simpelweg je eigen overzicht kwijt.

Mijn 'denklijn'
Om dat te voorkomen, werk ik al jaren met een vaste aanpak. Geen ingewikkelde methode en zeker geen strak model dat je slaafs moet volgen. Zie het als een praktische denklijn. Een rode draad die voorkomt dat je iets belangrijks overslaat en die ervoor zorgt dat wat je oplevert ook echt hout snijdt.

Het is geen stappenplan om af te vinken. Het is een hulpmiddel om:

- Overzicht te houden.
- De juiste keuzes te maken.
- Gesprekken te sturen.
- Vage ideeën concreet te maken.

In de praktijk loopt analyse zelden in een rechte lijn. Soms ga je vooruit, soms moet je even terug, en soms leg je ineens een verband dat pas later logisch blijkt. Juist door een vaste structuur in je hoofd te hebben, kun je die flexibele bewegingen maken zonder dat je de weg kwijtraakt.

Let op: Dit is geen watervalmethode
De fasen die ik hierna beschrijf, lijken op papier misschien een klassiek 'waterval'-model: eerst alles uitzoeken, dan alles uitwerken, dan pas bouwen. Maar dat is het niet.

In mijn aanpak werk je wel gestructureerd, maar je doet niet alles tegelijk en zeker niet alles vooraf.

- Fase 1 (De basis): Dit breng je aan het begin in grote lijnen in kaart. Je moet tenslotte weten waar je uiteindelijk naartoe werkt.
- Fase 2 & 3 (Uitwerken & doen): Deze lopen dwars door elkaar heen.

In fase 2 & 3 analyseer je precies op het juiste moment. Je pakt een gedeelte van je totaal op, analyseert dat tot het scherp genoeg is, en geeft het dan direct door aan de uitvoerders. Terwijl zij bouwen (Fase 3), begin jij aan de analyse van het volgende blok (Fase 2).

Zo hou je snelheid en flexibiliteit, maar verlies je nooit de grip.

Fase 1: Van behoefte naar richting

Hier leg je het fundament voor je werk. In deze fase ontdek je wat de echte vraag is, wie er allemaal bij betrokken zijn en hoe de oplossing er in grote lijnen uit moet zien.

Je gebruikt deze fase volledig bij nieuwe projecten of bij vraagstukken waar de wensen nog vaag of abstract zijn. Bij bestaande projecten is dit werk vaak (deels) al gedaan; zonder behoefte was er immers geen project gestart.

Maar let op als je later instroomt! Stap je als analist in een project dat al loopt? Sla deze fase dan niet zomaar over. Je wilt altijd begrijpen wat de oorspronkelijke behoefte was.

In de loop van een project sluipen er namelijk vaak ongemerkt aannames in, of blijkt dat het 'waarom' nooit echt scherp is vastgesteld. Durf die aannames te bevragen, want vaak is daar wel op verder gebouwd. Jouw frisse blik is dan precies wat het team nodig heeft om terug te keren naar de kern: wat was ook alweer de bedoeling?

De vijf kernactiviteiten
In deze fase doe je vijf dingen. In dit boek werken we ze één voor één uit, maar in de dagelijkse praktijk lopen ze vaak door elkaar heen:

1. **Start met een verkennend gesprek**: Wat is de wens eigenlijk? Ga in gesprek met de opdrachtgever om de eerste, essentiële informatie op te halen. Tast af waar de pijn of de behoefte zit.
2. **Breng in kaart voor wie je het doet**: Wie zijn je 'klanten' of belanghebbenden? Vaak zijn dat er veel meer dan je in eerste instantie denkt. Ze hebben allemaal hun eigen belangen, wensen en prioriteiten. Het is van groot belang om dit vroeg te weten.
3. **Vind de wens achter de wens**: Wat is het achterliggende doel? Een klant vraagt vaak om een middel (bijv. "Ik wil een app"), maar wat hopen ze daarmee te bereiken? Denk in deze stap nog niet in

oplossingen of techniek, maar focus puur op het waarom. Waarom is dit belangrijk?

4. **Werk het uit op hoofdlijnen (liefst visueel)**: Maak je ideeën tastbaar. Of je het nu beschrijft of tekent: zorg dat het begrijpelijk is. Geef mensen iets om op te reageren. Een schets of een globaal model werkt vaak beter dan een lap tekst. Zo toets je direct of je de goede richting te pakken hebt.
5. **Knip het op in waardevolle stukken**: Een groot project is lastig te overzien. Maak het geheel kleiner door het op te delen in losse bouwblokken die elk op zich iets opleveren. Hierdoor creëer je overzicht en kun je alvast een grove volgorde en prioriteit bepalen.

Voorbeeld: Het gevaar van "natuurlijk"

Ik schoof ooit halverwege aan bij een project voor een interne applicatie. Alles leek op rolletjes te lopen en op schema te liggen.

Tijdens een gesprek met een gebruiker liet iemand echter terloops vallen: “Ja, de applicatie is natuurlijk ook voor onze externe partners bedoeld. Die zien we tenslotte ook als onderdeel van ons bedrijf.”

Het woordje ‘natuurlijk’ zette mij aan het denken.

Ik zocht het uit en wat bleek? In de technische uitwerking was helemaal geen rekening gehouden met toegang van buitenaf. Partners konden er dus niet in.

Omdat ik dit vrij snel scherp kreeg, konden we het ontwerp direct aanpassen. Was dit pas bij de oplevering ontdekt, dan had dat weken aan herstelwerk gekost.

Een verkennend gesprek

Voordat je ergens aan begint, heb je de basis nodig. Start je blanco aan een nieuw project? Dan haal je die informatie het snelst op in een kort gesprek met de opdrachtgever of de persoon die het probleem heeft aangekaart.

Dit gesprek is nog geen uitgebreide analyse. Het is een intake, bedoeld om:

- Het doel en de aanleiding te horen in de eigen woorden van de opdrachtgever.
- Gevoel te krijgen bij de context, de historie en de betrokkenen.
- Te ontdekken welke afdelingen of processen waarschijnlijk geraakt worden.
- Een startlijstje te maken van mensen die je hierna moet spreken.

Wat vraag je in zo'n gesprek?
Je wilt snel tot de kern komen. Gebruik bijvoorbeeld deze vragen als leidraad:

- **De aanleiding**: "Waarom starten we dit project nu? Wat is er gebeurd?"
- **Het doel**: "Wanneer is dit voor jou geslaagd? Wat moet er aan het eind staan?"
- **De spelers**: "Wie heeft hier het meeste belang bij, of wie gaat hier het meeste van merken?"
- **De beslisser**: "Wie besluit uiteindelijk wat het eindresultaat wordt?"
- **De historie**: "Ligt er al iets? Is er al eerder iets onderzocht of gebouwd?"
- **De kaders**: "Zijn er keuzes die al vaststaan en waar we niet meer aan mogen tornen?"
- **De risico's**: "Zijn er gevoeligheden of risico's waar ik rekening mee moet houden?"

Waarom deze stap onmisbaar is
Het lijkt verleidelijk om meteen de diepte in te duiken, maar dit verkennende gesprek bespaart je later veel tijd:

- Je voorkomt een valse start: Je weet zeker dat je vertrekt vanuit de juiste probleemstelling.
- Je leert de taal: Je hoort welke termen en voorbeelden de opdrachtgever gebruikt.
- Je herkent tegenstrijdigheden: Doordat je het oorspronkelijke verhaal kent, merk je het later direct als de praktijk of andere collega's iets heel anders zeggen.
- Je bouwt je netwerk: Je weet direct wie je hierna moet interviewen.

Dit gesprek is de opstap naar de volgende fase, waarin je echt gaat graven naar de details.

Tip bij lopende projecten
Stap je in een project dat al loopt? Check dan even of dit gesprek al gevoerd is. Is de basis helder en liggen de afspraken ook ergens vast? Prima. Is het toch een beetje vaag? Vraag dan alsnog dat halfuurtje met de opdrachtgever aan. Het is beter om nu even te checken, dan later te gokken.

Weet voor wie je het doet

Projecten waarbij niemand precies weet voor wie de oplossing eigenlijk bedoeld is, komen vaker voor dan je denkt.

De opdrachtgever denkt dat het voor zijn personeel is. Het personeel denkt dat het voor hun klanten is. En jij? Jij zit in een overleg vol meningen, aannames en gele plakbriefjes. Alles klinkt logisch, maar niets sluit op elkaar aan.

Waarschijnlijk is het project al goedgekeurd, dus er ligt vast ergens een opdrachtomschrijving. Dat is je eerste bron. Soms haal je daar veel uit,

soms staat er nauwelijks iets op papier. Maar voordat je ook maar één stap zet, is het essentieel om te weten wie je nodig gaat hebben.

Stel daarom altijd de vraag: **"Voor wie doen we dit eigenlijk?"**

Als je niet scherp hebt voor wie je het doet, weet je namelijk ook niet:

- Wat echt belangrijk is (en wat niet).
- Wat slechts randzaken zijn.
- Wat mag blijven zoals het is, en wat absoluut anders moet.

Zonder een helder beeld van je doelgroep loop je het risico dat je een perfecte technische oplossing bouwt voor het verkeerde probleem.

Je werkt voor je klant, maar wie is dat?
Bij het woord 'klant' denken we vaak aan degene die de factuur betaalt, of de eindgebruiker die het product gebruikt. In ons vakgebied ligt dat genuanceerder. In de analyse is iedereen die een belang heeft bij het resultaat een klant: van de eindgebruiker op de werkvloer tot de directeur die beslist, en van de inhoudelijk expert tot de collega die het systeem moet beheren.

Voor jou als analist is een 'klant' iedereen die:

- Betrokken is bij wat jij onderzoekt of maakt.
- Geraakt wordt door het resultaat (positief of negatief).
- Nodig is om jou van informatie te voorzien.

Als iemand belangrijk is voor het slagen van je analyse, behandel je diegene als klant. Je aanpak is daarbij voor iedereen hetzelfde: je wilt begrijpen wie het is, wat diegene nodig heeft en wat hij jou kan vertellen. Om te voorkomen dat je werkt op basis van aannames of alleen luistert naar de persoon met de grootste mond, gebruik je de volgende technieken:

1. Breng je klanten in kaart
Wie heeft er belang bij het resultaat? Wie mag er beslissen? Wie is er afhankelijk van jouw werk? Wie moet er toestemming geven?

Om overzicht te krijgen, kun je dit uittekenen. Een handige methode is om betrokkenen op twee assen te plaatsen:

- Invloed: Wie heeft de macht om besluiten te nemen of het project te sturen?
- Impact: Wie wordt er in zijn dagelijkse werk geraakt door de verandering?

Dit helpt je te focussen. Het voorkomt dat je alleen rent voor de mensen die het hardst roepen (vaak veel invloed, weinig impact), terwijl je de stille sleutelpersonen (weinig invloed, veel impact) vergeet. Juist die laatste groep heeft vaak de meest waardevolle informatie.

2. Ga in gesprek met alle groepen
Spreek met de mensen die het werk daadwerkelijk doen, niet alleen met hun managers. Managers vertellen je hoe het proces zou moeten gaan; medewerkers vertellen je hoe het echt gaat.

Gebruik in deze gesprekken de technieken die we verderop in het boek bespreken: vraag naar gedrag, vat samen wat je hoort, toets je aannames en graaf altijd naar de vraag achter de vraag. Zo ontdek je niet alleen wat mensen nodig hebben, maar kom je vaak ook nieuwe betrokkenen op het spoor die je eerder over het hoofd zag.

3. Maak persona's van de gebruikers
Soms is de groep betrokkenen zo groot of divers dat je het overzicht verliest. Dan helpt het om persona's te maken.
Een persona is een hulpmiddel om het gedrag, de frustraties en de behoeften van een groep gebruikers tastbaar te maken. Het is een korte, herkenbare beschrijving van een typische gebruiker, vaak met een naam, een rol, persoonlijke doelen en knelpunten. Let op: dit is geen verzonnen

marketingpraatje, maar een samenvatting van feiten die je echt hebt opgehaald uit je gesprekken en observaties.

Een persona helpt je om:

- Scherp te houden wat je gaat maken of veranderen.
- Te begrijpen wat voor die specifieke groep waardevol is.
- Inzicht te krijgen in hoe de oplossing straks gebruikt wordt.
- Discussies te beslechten ("We bouwen deze functie niet voor jou persoonlijk, maar voor persona 'Medewerker X'").

Het maken van persona's kost tijd. Het is vooral aan te raden bij projecten waar veel verschillende rollen of type gebruikers bij betrokken zijn. Ze helpen je dan om grip te houden op de uiteenlopende belangen.

Heb je een klein project of maar te maken met één of twee soorten gebruikers? Dan is een uitgebreide persona niet per se nodig. Zolang je maar scherp op je netvlies hebt wie je voor je hebt en wat diegene nodig heeft.

Meer over persona's lees je op pagina 101.

Wat als je bepaalde mensen niet mag spreken?
Soms krijg je te horen: "Je hoeft de gebruikers niet te storen, praat maar met mij." In dat geval verandert je doel niet, alleen je route.

Idealiter spreek je de mensen die daadwerkelijk met de oplossing gaan werken. Probeer dat altijd voor elkaar te krijgen. Lukt dat echt niet? Dan moet je het doen met een vertegenwoordiger; iemand die spreekt namens de groep, maar er zelf geen onderdeel van is.

In zo'n situatie gebruik je dezelfde interviewtechnieken, maar ben je extra kritisch:

- **Stel veel verduidelijkende vragen.** Ga niet af op vage termen.
- **Vraag door.** Hoe weet deze persoon dit?
- **Toets aannames.** Is dit een feit of een mening?
- **Vraag om concrete voorbeelden.** "Kun je een situatie schetsen waarin dit gebeurde?"

Bij twijfel kun je de vertegenwoordiger vragen om jouw samenvatting te toetsen bij de achterban. Zo blijf je, ondanks de beperking, toch zo dicht mogelijk bij de werkelijkheid.

De wens achter de wens

Dit is de stap voordat je in oplossingen gaat denken. Het is het moment waarop je de diepte in gaat om de echte behoefte boven water te halen. Niet met een checklist in je hand, maar met gezonde nieuwsgierigheid en de vraag: "Waarom willen ze dit eigenlijk?"

Mensen vragen om oplossingen
Als iemand tegen je zegt: "We willen een dashboard", dan zegt dat op zichzelf nog helemaal niets over het doel. Het zegt alleen dat diegene een bepaalde vorm (een dashboard) in zijn hoofd heeft omdat dat tastbaar voelt.

Voor je het weet ben je bezig met grafieken, ververs-knoppen en gebruikersrechten. Maar waarom? Misschien wil de klant sneller kunnen samenwerken, fouten signaleren, of gewoon het gevoel van controle hebben.
De grootste fout in analyse is de aanname dat de eerste vraag van de klant ook meteen de juiste is. Jouw taak is om terug te gaan naar het begin. *Waarom* die wens er is, is belangrijker dan *wat* ze precies vragen.

Technieken om de echte vraag te vinden
Hoe krijg je die onderste steen boven? Niet met een standaard vragenlijstje, maar door de juiste gesprekstools in te zetten.

Zie de onderstaande acht technieken als je arsenaal. Je hoeft ze niet allemaal tegelijk te gebruiken. Soms werkt een emotionele vraag, de andere keer een tekening. Jij voelt aan wat past.

- **Teken de werksituatie:** (of speel het na) Laat iemand tekenen of beschrijven hoe een dag verloopt. Wat gebeurt er precies? Waar loopt het vast en wanneer gaat het juist soepel?

  Tekenen maakt gedrag concreet. Je ziet meteen waar de processen stokken. Dit is misschien wel de krachtigste techniek.

- **Vraag naar gedrag**: Vraag niet naar meningen, maar naar feiten. "Wanneer gaat het nu fout?", "Wat probeer je op dat moment te doen?" en "Wat lukt er dan niet?"

  Mensen kunnen hun gedrag vaak makkelijker beschrijven dan hun abstracte wensen. Dit geeft jou zicht op het echte knelpunt.

- **De 'waarom'-vraag** (zonder te zeuren): Vraag door. "Waarom is dit belangrijk?", "Wat gebeurt er als we dit niet doen?" Meestal heb je drie tot vijf vragen nodig om tot de kern te komen.

  Je pelt de lagen af tot je bij de echte waarde of motivatie komt.

- **Focus op emoties**: Vraag waar iemand gefrustreerd van raakt, of juist blij van wordt. "Wat maakt je moedeloos in dit proces?"

  Emotie is een enorm sterk signaal. Waar frustratie zit, zit vaak het grootste probleem (en dus de meeste waarde om het op te lossen).

- **Schets het toekomstbeeld**: "Stel: over drie maanden werkt dit perfect. Wat kun je dan wel, wat je nu niet kunt?"

  Door mensen even te laten dromen, stappen ze uit de beperkingen van vandaag en krijg je zicht op hun doelen.

- **Vraag naar het verleden**: "Wat heb je eerder al geprobeerd? Waarom werkte dat niet?"

  Je ontdekt patronen en valkuilen waar mensen zich vaak niet eens bewust van zijn. Het voorkomt dat je het wiel opnieuw uitvindt.

- **Het doemscenario** (“Stel, het is mislukt”): "Stel, dit project is over een half jaar mislukt. Wat is er dan misgegaan?"

  Deze omkering haalt risico’s en zorgen naar boven die mensen in een positief gesprek vaak verzwijgen.

- **Samenvatten en toetsen**: Vat samen wat je denkt dat de ander bedoelt en check het direct: "Dus als ik het goed begrijp, wil je dat iemand zonder uitleg direct ziet of er nog taken open staan?"

  Je neemt aannames weg en weet zeker dat jullie op één lijn zitten.

Je hoeft deze technieken niet altijd alle 8 toe te passen. Gebruik ze los of in combinatie. De ene keer werkt emotie, de andere keer juist een schets, maar hoe vaker je ze gebruikt, hoe scherper je wordt. Je hoeft niet alles perfect te doen, maar als je echt luistert en via deze technieken door durft te vragen, ontdek je wat er werkelijk speelt.

Wil je dieper ingaan op deze technieken? Vanaf pagina 64 vind je per techniek extra uitleg en voorbeelden.

Wat levert dit op?
Als je dit goed doet, bouw je niet op drijfzand. Je krijgt geen vage meningen, maar harde uitgangspunten:

- Doel: Je weet waarom we dit doen.
- Kader: Je weet wat wel en wat niet bij de oplossing hoort.
- Afstemming: Iedereen snapt waar de reis naartoe gaat.
- Toetsbaar: Je kan te allen tijde toetsen of je oplossingen aansluiten bij de wens
- Prioritering: Je weet wat hoofdzaak is en wat bijzaak.

En niet onbelangrijk: je werk wordt er leuker van. Bouwen aan een oplossing waarvan je weet dat hij echt nodig is, geeft veel meer voldoening dan gokken wat de klant wil.

Tip: Schrijf alles op wat je hoort. Je hoeft geen roman te schrijven, maar noteer wel wat je hoort. Juist die ene zin waarvan je nu denkt: "Ach, dat is een detail", blijkt later vaak de sleutel tot de oplossing te zijn.

(Visueel) ontwerpen op hoofdlijnen

Informatieanalyse draait om één ding: snappen wat er echt speelt. Dat lukt niet met alleen woorden, maar ook niet met alleen plaatjes. Je hebt beide nodig.

Tegelijkertijd wil je voorkomen dat je een dik boekwerk produceert. Hoe meer tekst, hoe groter de kans dat niemand het leest. Hanteer daarom de stelregel: **wat je kunt tekenen, teken je**.

Zeker in deze fase van het project, waarin het nog niet om de details gaat maar om begrip en richting, werkt een schets vele malen sneller dan een lap tekst. Een proces, een scherm of een informatiestroom laat zich perfect tekenen. Door het visueel te maken, zie je namelijk waar het wel en niet werkt.

Betekent dit dat je alles moet tekenen? Nee. Er zijn uitspraken die je juist letterlijk moet vangen in woorden, omdat ze de kern van het probleem blootleggen. Denk aan zinnen als:

- “Ik reken het altijd nog even na in Excel, want ik vertrouw de uitkomst niet.”
- “Ik durf niet op die knop te drukken want de vorige keer raakte ik alles kwijt.”
- “Ik moet zoveel invullen dat het overbodig complex voelt.”

Dit soort zinnen zijn waardevol. Schrijf ze letterlijk op en verwerk ze in je analyse.

Het effect van tekenen: weg met de interpretatie
Onze hersenen zijn visueel ingesteld. Een simpele tekening op een whiteboard of kladblok zet mensen direct aan het denken. Ze zien ineens wat ze eerder alleen maar hoorden.

Visueel werken is geen extra stap, het is een versneller. Zolang iets alleen in woorden bestaat, blijft er ruimte voor interpretatie. Iedereen maakt zijn eigen plaatje in zijn hoofd. Maar zodra je samen naar een schets kijkt, verdwijnt die ruis en worden mensen scherp:

- "Hé, dit klopt niet."
- "Hier mist nog een stap."
- "Is dit echt wat je bedoelt?"

Voorbeeld: "Volgens mij bedoel ik dat niet. "

Ik vertelde aan een klant welke oplossing we bedacht hadden. Op papier klopte alles: ik gebruikte de woorden van de klant, de eisen waren netjes geformuleerd en een collega had het gecheckt. Toch zag ik verwarring in de ogen van de opdrachtgever en hij zei "volgens mij bedoel ik dat niet…"

Ik pakte een stift en maakte een snelle schets op het whiteboard: dit is wat je ziet, hier klik je op en dat gebeurt er dit en dat…

Hij knikte meteen. "Ah, ja dat bedoelde ik inderdaad."

Wat teken je wel (en wat niet)?
Je hoeft geen ontwerper te zijn. Sterker nog: je hoeft helemaal niet mooi te kunnen tekenen. Visueel ontwerpen is hier geen kunstvorm, maar een communicatiemiddel. Je tekent niet voor een systeem, maar om elkaar te begrijpen.

Vormen die vaak goed werken:

- **Schermschetsen** (wireframes): Wat staat waar en wat doet de gebruiker?
- **Stroomschema's**: Wanneer gebeurt er wat? In welke volgorde? Wat gaat handmatig en wat automatisch?
- **Wie-doet-wat**: Wie doet wat, wanneer en wie wacht op wie?
- **Informatiestromen**: Waar komt de informatie vandaan en waar gaat het heen?
- **Voor-en-na**: Hoe gaat het nu, en hoe ziet de wereld er straks uit?

Tekenen leidt tot betere vragen
Tijdens het tekenen dwing je jezelf om logisch na te denken. Je komt vanzelf tot vragen die je anders misschien was vergeten:

- "Moet deze stap eigenlijk wel?"
- "Wat gebeurt er als dit niet ingevuld wordt?"
- "Wie krijgt dit eigenlijk te zien?"

Het helpt je om verder te denken dan "klinkt goed" en maakt impliciete aannames pijnlijk zichtbaar.

Nog een voorbeeld: De goedkeuring

De wens is simpel: “We willen het makkelijker maken om een aanvraag goed te keuren.”

Als je vraagt wat dat betekent, krijg je vaak vage antwoorden. Maar als je het tekent:
📄 Document → 🖱️ Klik op knop → ✅ Akkoord → 📩 Verzenden

...dan zegt iemand ineens: “Ja, maar ze moeten wel eerst de voorwaarden hebben geaccepteerd.”

En ineens praat je niet meer over aannames, maar over concreet gedrag en regels.

Tools? Een bierviltje is genoeg
Digitale tools zijn handig, maar laat je niet tegenhouden als je geen laptop bij de hand hebt. Teken op papier, een whiteboard, een servetje of een bierviltje. Maak er een foto van en plak die in je documentatie.
Wees niet bang voor lelijke tekeningen. In analyse geldt: duidelijk gaat boven mooi.

Tot slot: een opstap naar de volgende fase
Visueel werken heeft nog een groot voordeel: het helpt je bij het opdelen van je werk. Zodra je een proces of scherm uittekent, zie je vaak vanzelf waar de logische 'knip' zit. Elk vakje of elke stap in je tekening kan het begin zijn van een los bouwblok dat je verder gaat analyseren.

Opdelen en prioriteren

Als het goed is, heb je door de vorige stappen een helder beeld van het geheel. Je begrijpt de vraag, de kaders zijn duidelijk en je hebt samen geschetst hoe de oplossing er ongeveer uit moet zien.

Je kent nog lang niet alle details (en dat hoeft ook niet), maar je hebt het grote plaatje te pakken. Maar dat hele plaatje bouw je niet in één keer. Daarom is de volgende stap: opdelen. Je knipt het grote project op in logische, functionele 'bouwblokken'.

Voor de IT'ers onder ons
In dit boek gebruik ik de term bouwblokken, omdat dit voor iedereen begrijpelijk is. Werk je in de softwareontwikkeling? Dan herken je dit waarschijnlijk als User Stories of Features. Het principe is exact hetzelfde: je maakt het werk klein en behapbaar.

Waarom opknippen?
Een groot doel is vaag. Kleine onderdelen zijn concreet. Hoe kleiner je het maakt, hoe beter je kunt:

- Inschatten: Hoeveel tijd kost dit specifieke stukje?
- Prioriteren: Wat is nu echt belangrijk en wat kan wachten?
- Plannen: Wie pakt welk onderdeel op?

Vergelijk het met het bouwen van een huis. Door de eerdere stappen weet je hoe het huis eruitziet en voor wie je het bouwt. Je kunt zeggen: "Over een jaar is het huis klaar." Maar hoe bepaal je dat? Je moet het uitsplitsen in werkzaamheden:

- Fundering storten
- Muren metselen
- Elektra aanleggen
- Dak plaatsen

Pas als je deze losse onderdelen ziet, snap je de afhankelijkheden (eerst de muren, dan het dak) en kun je inschatten wat daarbij komt kijken. Zo werkt het ook met jouw project: eerst opdelen, dan pas uitvoeren.

Let op: Nog geen details
We zitten in deze fase nog steeds op hoofdlijnen. Je maakt het werkbaar, maar je gaat nog niet elke stap of elke uitzondering uitwerken. Je beschrijft per blok vooral:

- Wat moet er gebeuren?
- Wat moet het resultaat zijn?
- Voor wie is het bedoeld?

Details als "Welke kleur heeft de knop?" of "Is er een uitzondering?" komen later. Het doel is dat je het werk opknipt in begrijpelijke blokken, dat je niks belangrijks over het hoofd ziet en dat je stukjes werk creëert die je kunt verfijnen, schatten en prioriteren.

Ontwerp vanuit gebruik, niet vanuit de oplossing
Een veelgemaakte fout is opdelen in oplossingen, zoals "Bouw een menu" of "Maak een dashboard". Dat zegt niets over de waarde voor de gebruiker.

Deel daarom altijd op basis van **gedrag**. Wat wil iemand kunnen doen? Bijvoorbeeld bij een klantportaal. De wens is: "We willen een selfserviceportaal voor klanten." Dit knip je niet op in schermen, maar in handelingen:

- Klant kan contactgegevens wijzigen.
- Klant kan openstaande facturen bekijken.
- Klant kan een verzoek indienen.
- Klant kan de status van een aanvraag inzien.
- Medewerker kan gebruikers toevoegen.

Elk blokje vertegenwoordigt nu waarde voor de gebruiker. Pas in de uitwerking van elk blok kijk je hoe de oplossing precies gaat werken.

Een structuur die altijd werkt
Om te zorgen dat je bouwblokken concreet blijven, kun je deze (voor IT'ers bekende) standaardzin gebruiken:

Als [rol] **wil ik** [actie], **zodat** [doel].

Bijvoorbeeld: "**Als** baliemedewerker **wil ik** direct op het startscherm zien met wie een klant een afspraak heeft, **zodat** ik hem bij binnenkomst meteen naar de juiste spreekkamer kan verwijzen."

Dit is duidelijk, meetbaar en werkbaar. Pas wel op voor nep-doelen. Zorg dat het stukje achter 'zodat' ook echt een doel is.

Fout: "...zodat ik mijn werk kan doen." (Dit zegt niks).
Fout: "...zodat ik het in het systeem kan zetten." (Dat is een handeling, geen doel).

Prioriteren met MoSCoW
Nu je een lijst met bouwblokken hebt, zul je merken dat je meer werk hebt dan tijd. Je moet kiezen. Een handige methode hiervoor is MoSCoW. Doe dit samen met je opdrachtgever of klant:

- **M - Must have**: Dit moet erin. Zonder deze onderdelen werkt de oplossing niet of faalt het project.
- **S - Should have**: Heel belangrijk, maar als het echt moet, kan het iets later of via een (tijdelijke) omweg.
- **C - Could have**: Leuke extra's ('nice to have'). We doen dit alleen als er tijd en budget over is.
- **W - Won't have**: Dit doen we nu niet. Misschien in de toekomst, dat bepalen we later nog wel.

Deze methode zorgt ervoor dat jij en je klanten duidelijke keuzes maken. Het voorkomt eindeloze discussies waarin alles 'belangrijk' wordt genoemd.

Meer over MoSCoW lees je op pagina 98.

Het hoeft niet perfect
De eerste lijst met bouwblokken hoeft niet perfect te zijn. Sterker nog, dat lukt je toch niet. Tijdens het project ontdek je vanzelf dat sommige blokken te groot zijn (die knip je op) of dat prioriteiten verschuiven. Dat hoort erbij. Zorg voor een logische startlijst, dan ben je al een heel eind.

Belangrijk verschil: Prioriteit vs. Volgorde
MoSCoW bepaalt wat belangrijk is (de waarde). Dat is niet altijd hetzelfde als de volgorde waarin je bouwt. Niet alle 'Must haves' hoeven als eerste klaar te zijn. Soms begin je met iets kleins om vertrouwen te winnen, of met het moeilijkste stuk techniek om een risico weg te nemen. Deze volgorde (de planning) bepaal je pas in de volgende stap.

Fase 2: Uitwerken van de details

In fase 1 heb je het grote plaatje scherp gekregen. Je weet voor wie je aan de slag gaat, wat de kernvraag is en je hebt de oplossing opgedeeld in logische brokken met een duidelijke prioriteit.

Toch ben je er dan nog niet. Een globaal idee is namelijk nog geen werkbaar plan. Je kunt niet tegen een team zeggen: "Bouw dit onderdeel maar", zonder verdere uitleg. Om van een plan naar uitvoering te gaan, moet je inzoomen.

Vanaf nu ga je per onderdeel de diepte in. Je maakt de vertaalslag van een abstract idee naar concrete actie. Het doel van deze fase is simpel: zorgen dat iedereen precies weet wat er moet gebeuren, en neem daar vooral ook de waarom in mee. Als men snapt waarom iets gedaan moet worden kunnen ze actief meedenken.

Bouwblokken uitwerken

Misschien herken je het wel: de hele groep zit klaar om knopen door te hakken, en dan zegt iemand: “Tja, dit moeten we eigenlijk nog even beter uitzoeken.”

Het gevolg? Dertig minuten discussie met vragen als:

- “Wat bedoelen ze hier precies mee?”
- “Is dit al definitief?”
- “Hoe zit dit technisch?”

Het gesprek loopt vast, er worden aannames gedaan en uiteindelijk ga je uit elkaar zonder echte beslissing. Zonde van de tijd en onnodig.

De oplossing: voorbereiding.
Je voorkomt dit door te zorgen dat een bouwblok al is uitgezocht voordat het de groep in gaat. Dat is jouw taak in deze fase. Je brengt vooraf de juiste scherpte aan, zodat het team straks snel en goed kan beslissen.

Je pakt dit hetzelfde aan als de grote analyse zoals beschreven in fase 1, maar dan op kleine schaal. Je zoomt in. Je activiteiten zijn:

- **Inhoud verdiepen**: Wat is precies de kern van de wens of het probleem?
- **Gesprekken voeren**: Praat met de mensen die direct met dit specifieke stukje te maken hebben.
- **Gedrag beschrijven**: Wat doet de gebruiker precies? Welke handelingen verricht hij?
- **Uitzonderingen zoeken**: Wat gebeurt er in 'lastige' situaties of als er iets misgaat?
- **Regels checken**: Zijn er wetten of interne afspraken waar we rekening mee moeten houden?
- **Schetsen**: Maak visueel hoe het eruit moet zien.

Je doel is helderheid. Geen vage ideeën meer, maar een scherp uitgewerkt stuk werk waar de bouwers direct mee aan de slag kunnen.

Voorbeeld: De douche

Eerder vergeleken we het project met het bouwen van een huis. We hebben het huis toen opgedeeld in kamers (de bouwblokken). Nu zoom je in op één specifiek onderdeel: de douche.

Om dit bouwblok uit te werken, stel je weer gerichte vragen. Niet meer over het hele huis, maar puur over dit detail:

- Waar komt de douche precies in de ruimte?
- Willen we een regendouche of een handdouche?
- Wordt het een inloopdouche of een cabine?
- Moet de vloer antislip zijn?

Hoe concreter je dit maakt, hoe kleiner de kans dat de aannemer straks iets bouwt wat je niet wilde.

Wat moet het opleveren?
Een goed uitgewerkt bouwblok voldoet aan deze eisen:

- Begrijpelijk: Iedereen snapt wat er staat.
- Realistisch: Het sluit aan op wat mensen echt doen.
- Concreet: Er zijn duidelijke keuzes te maken.
- Discussiewaardig: Er liggen scenario's klaar om over te praten.
- Getoetst: De mensen die ermee moeten werken, hebben er al naar gekeken.

Tip: Maak een vast sjabloon
Gebruik voor elk bouwblok een vaste indeling. Zo vergeet je niets en is het voor iedereen herkenbaar. Zet er in ieder geval in:

- Waarom dit nodig is.
- Wat het moet opleveren (het doel).
- Hoe het werkt (de inhoudelijke omschrijving en regels).
- Hoe het eruitziet (schetsen).
- Risico's en nog niet getoetste aannames.
- Afhankelijkheden (hebben we iets van een ander team nodig?).

Hoe pak je dit aan?
Hoe zorg je dat je dit efficiënt doet zonder dat je verzuipt in details? Hanteer deze werkwijze:

1. **Begin bij de hoogste waarde**: Pak de bouwblokken met de hoogste prioriteit als eerste op. Dit zijn de onderdelen die de meeste waarde leveren voor de gebruiker. Door hier te beginnen, zorg je dat het team (de uitvoerders) direct aan de slag kan met het belangrijkste werk.

   Let op: Jij moet voor de troepen uit lopen. Voorkom dat uitvoerders stilzitten omdat jij nog aan het analyseren bent. Zodra een blok scherp genoeg is, draag je het over en begin jij aan de volgende.

2. **Toets de volgorde**: Kijk kritisch of de volgorde die je eerder bedacht nog steeds logisch is. Overleg dit ook met de uitvoerders. Vraag hen: Wat is voor jullie een logisch startpunt?"

   Soms is het technisch handiger om met blok B te beginnen voordat blok A wordt gebouwd.

3. **Gebruik je interviewtechnieken** (maar dan ingezoomd): Bij het uitwerken gebruik je precies dezelfde gesprekstechnieken als in fase 1, zoals het vragen naar de 'wens achter de wens'. Het verschil is de scope: je lens staat nu ingezoomd.

Je vraagt niet meer: "Wat is het doel van het systeem?" Je vraagt nu: "Welke gegevens heb je hier nodig?" en "Wat moet er gebeuren als het misgaat?"

Hoe scherper jij dit beeld krijgt, hoe kleiner de kans op verrassingen en fouten tijdens de uitvoering.

Controleer of je bouwblok klaar is: INVEST
Een handig hulpmiddel om te checken of je bouwblok echt klaar is om opgepakt te worden, is de INVEST-checklist. Leg je uitwerking langs deze meetlat:

- **Independent** (Onafhankelijk): Het onderdeel staat zoveel mogelijk op zichzelf. Het hangt niet aan allerlei andere taken vast, waardoor je het los kunt bouwen en testen.
- **Negotiable** (Bespreekbaar): Het is geen dichtgetimmerd contract. Er is ruimte voor het team om mee te denken over de slimste technische oplossing.
- **Valuable** (Waardevol): Het levert direct iets op voor de gebruiker. Als je niet kunt uitleggen waarom dit waarde toevoegt, moet je terug naar de tekentafel.
- **Estimable** (Inschatbaar): Het is duidelijk genoeg om te zeggen hoeveel tijd het kost. Als het team zegt: "Geen idee hoe lang dit duurt", is je omschrijving waarschijnlijk nog te vaag.
- **Small** (Klein): Het is behapbaar. Is het te groot? Knip het dan op in kleinere stukken. Zo houd je snelheid en zie je sneller resultaat.
- **Testable** (Testbaar): Je kunt objectief vaststellen wanneer het 'af' is. Er zijn duidelijke criteria waar het eindresultaat aan moet voldoen.

Zie je gaten als je deze lijst afloopt? Corrigeer ze dan direct. Zo voorkom je dat je in de gezamenlijke sessie alsnog vastloopt.

Lees meer over INVEST op pagina 104.

De gezamenlijke toets

Analyseren doe je niet in je eentje. Als analist heb je inmiddels veel voorwerk gedaan: je hebt gesprekken gevoerd, informatie opgehaald, zaken uitgezocht en een concreet voorstel uitgewerkt. Waarschijnlijk heb je tussendoor al even met collega's gespard om je ideeën aan te scherpen. Maar voordat je echt van start gaat, moet je één ding zeker weten: klopt het wat er ligt? En misschien nog wel belangrijker: kan het ook echt?

Daarom bespreek je je voorstel altijd met de mensen die het project gaan uitvoeren. Zij helpen je om te controleren of het idee in de praktijk haalbaar is. Zij wijzen je op details die je misschien gemist hebt en komen vaak met slimme oplossingen waar je zelf nog niet aan had gedacht.

De uitvoerders zien vaak als eerste of iets echt werkt. Misschien blijkt er een technische beperking te zijn, of is jouw voorstel veel ingewikkelder om te bouwen dan het lijkt. Zulke dingen wil je ontdekken voordat de bouw begint, niet pas halverwege. Blijken er na het gesprek aanpassingen nodig? Dan verwerk je die direct of koppel je ze terug. Zo blijft iedereen aangehaakt en voorkom je verrassingen achteraf.

Wat gebeurt er tijdens zo'n sessie? Het is geen eenrichtingsverkeer, maar een wisselwerking:

- **Jij brengt de inhoud in**: Wat moet er gebeuren, voor wie doen we dit en waarom is het nodig?
- **Collega's toetsen de inhoud**: Klopt het verhaal inhoudelijk en technisch?
- **Specialisten vullen aan**: Zij voegen hun vakkennis toe op punten die jij niet weet.
- **Er worden kritische vragen gesteld**: Samen zoeken jullie naar blinde vlekken en dingen die vergeten zijn.
- **De volgorde wordt besproken**: Wat is logisch om eerst te doen en wat komt later?

- **Beperkingen worden duidelijk**: Zijn er randvoorwaarden of technische grenzen waar we rekening mee moeten houden?
- **Samen zoeken naar oplossingen**: Hoe kunnen we dit het slimste aanpakken?
- **Inschatting van het werk**: Hoeveel tijd en moeite gaat dit kosten?
- **Besluitvorming**: De beslisser geeft aan of dit haalbaar en waardevol genoeg is.
- **De eindconclusie**: Weten we genoeg om te kunnen starten?

Als je dit goed doet, is het geen saaie vergadering, maar een energieke werksessie. Het is de cruciale stap om fouten te voorkomen en met vertrouwen te beginnen.

Voorbeeld: Alles klopte… tot we er samen naar keken

De analist had zijn huiswerk grondig gedaan. De klant had een heldere wens: medewerkers moesten zelf hun pasfoto kunnen uploaden voor hun reispas. Deze foto zou dan automatisch op de pas worden gezet.

Het doel was duidelijk: minder administratief werk voor HR, een sneller proces, en medewerkers hebben zelf de controle. De analist had aan alles gedacht:

- Gesproken met HR en de eindgebruikers.
- De verschillende typen gebruikers (persona's) in kaart gebracht.
- Een schets gemaakt van het scherm en het proces.
- Nagedacht over uitzonderingen (wat als de foto het verkeerde formaat heeft?).
- De eisen waaraan het moet voldoen (acceptatiecriteria) opgeschreven.
- Alles netjes afgestemd en goedgekeurd door de opdrachtgever.

Kortom: op papier leek alles klaar voor de start, maar toen het team bij elkaar kwam om alles keer door te nemen kwamen er nieuwe vragen vanuit verschillende invalshoeken:

- De tester vroeg: “Wat gebeurt er als iemand voor de grap een ongepaste foto uploadt? Wie controleert dat?”
- De bouwer vroeg: “Waar slaan we die foto eigenlijk op? In het HR-systeem? Of ergens anders, want de koppeling met het HR-systeem is er niet zomaar.”
- De privacy-specialist merkte op: “Let op, we mogen deze persoonsgegevens niet zomaar opslaan zonder expliciete toestemming van de medewerker.”

Door dit gesprek werd duidelijk dat het plan nog niet compleet was:

- Er was een extra stap nodig voor een controle van de foto voordat deze op de pas komt.
- De opslaglocatie moest technisch worden afgestemd.
- Er moest een duidelijke melding komen waarin de medewerker toestemming geeft voor het gebruik van de foto.

Het resultaat
Er kwam een extra ‘bouwblok’ bij voor de fotocontrole, de analyse voor de opslag werd opnieuw bekeken en er werden eisen toegevoegd over privacy. Het kostte een kwartier extra overleg, maar het voorkwam werk achteraf.

Jouw rol als inhoudelijke gids
Je zit niet aan tafel om simpelweg je document voor te lezen of te vertellen wat iedereen moet doen. Je zit er om te zorgen dat het verhaal klopt en dat iedereen het snapt. Jij faciliteert het gesprek.

Dat betekent dat je:

- Mensen uitnodigt om hun vragen en twijfels uit te spreken.
- Verdiepende vragen stelt, zoals: "Wat missen we nog?" of "Is dit onderdeel essentieel?"
- Eerlijk benoemt wat je nog niet hebt uitgezocht en bespreekt of dat een probleem is.
- Blijft checken: "Begrijpen we dit echt allemaal hetzelfde?"

Hoe je het goed aanpakt (6 tips)

1. **Begin met gedrag, niet met oplossingen**: Lees je uitwerking niet van A tot Z voor. Start met het doel en het gedrag. Zeg bijvoorbeeld: "De gebruiker wil in situatie Y doel X bereiken." Zeg niet: "Ik wil dat hier een knop komt en daar een invulveld."

   Door te beginnen bij het gedrag, nodig je specialisten uit om mee te denken over de slimste oplossing. Misschien is die knop wel helemaal niet nodig.

2. **Gebruik beelden**: Eén plaatje zegt meer dan duizend woorden. Gebruik de schetsen die je gemaakt hebt, of teken tijdens het gesprek op een whiteboard. Stroomschema's of simpele tekeningen maken een abstract verhaal direct concreet. Laat zien wat je bedoelt.

3. **Laat ruimte voor verrassing**: Wees niet verliefd op je eigen oplossing. Misschien heeft een collega een veel slimmere of snellere aanpak bedacht dan jij. Vraag er actief naar en geef ruimte voor die creativiteit.

4. **Leg de risico's op tafel**: Doe niet alsof alles zeker is. Zijn er afhankelijkheden van andere teams? Zijn bepaalde gegevens

misschien niet beschikbaar? Externe factoren die invloed hebben? Benoem ze nu, zodat je er niet later over struikelt.

5. **Praat over toetsbaarheid**: Wanneer is het 'af'? Bespreek hoe jullie straks controleren of het geslaagd is. Wat moet er precies werken of meetbaar zijn?
6. **Bepaal samen de volgorde**: Je bepaalt in deze sessie in welke volgorde het werk wordt opgepakt. Waarschijnlijk heb je zelf al een idee (bijvoorbeeld via de MoSCoW-methode), maar dit is het moment om dat te toetsen.

   De volgorde hangt af van vier factoren: **prioriteit**, **afhankelijkheid**, **risico's**, of gewoon **snelle winst** (soms pak je iets kleins eerder voor een snel resultaat.)

Zodra een bouwblok geaccepteerd is door de uitvoerders kunnen zij direct aan de slag, terwijl jij alvast vooruitwerkt aan de volgende blokken.

Wanneer is iets echt 'klaar om te doen'?
Aan het einde van de sessie stel je de vraag: "Weten we genoeg om te starten?" Het antwoord is pas 'ja' als:

- Iedereen begrijpt wat de bedoeling is.
- De eisen helder zijn (we weten wat 'goed genoeg' is).
- De aanpak en werkwijze duidelijk zijn.
- Afwijkingen en randgevallen zijn besproken.
- Er een inschatting is van de hoeveelheid werk.
- Er overeenstemming is over de volgorde van werken.

Pas als iedereen het snapt, kan iemand het gaan maken.

Fase 3: Uitvoeren en bijsturen

Nu gaat het echt gebeuren: de plannen worden werkelijkheid. Het team gaat aan de slag met de afspraken die gemaakt zijn. Voor alle, ook de kleine, bouwblokken geldt dat jij hier noodzakelijk blijft. Er zullen altijd vragen opkomen die je niet had voorzien. Door beschikbaar te zijn, voorkom je dat kleine misverstanden uitgroeien tot grote fouten of verkeerde aannames.

Controleer regelmatig of het resultaat nog steeds klopt met de oorspronkelijke bedoeling. Vraag dit aan de mensen voor wie je het maakt. In de praktijk ontstaan vaak nieuwe inzichten die je achter een bureau nooit had kunnen verzinnen.

Wees niet bang om dingen aan te passen. Of het nu om een klein detail gaat of een grotere wijziging: bespreek het open met je team. Soms betekent dit dat de prioriteiten verschuiven. Dat is niet erg, zolang het uiteindelijk maar leidt tot een oplossing die werkt.

Van analyse naar werkelijkheid

De start is gemaakt. De afspraken staan vast, iedereen weet wat er moet gebeuren en de koers is duidelijk. Misschien leun je nu achterover en denk je: “Mijn werk zit erop, succes ermee.”

Helaas, dat is zelden zo. Sterker nog: juist nu wordt jouw analyse pas echt getest. Niet meer op papier of in theorie, maar in de weerbarstige praktijk.

Waarom jouw werk hier niet stopt
Op het moment dat mensen daadwerkelijk aan de slag gaan met jouw analyse, verandert het soort vragen dat ze stellen. In de voorbereiding ging het over de grote lijnen, nu gaat het over de details.

Je kunt vooraf nooit alles voor 100% dichttimmeren, en dat is ook niet erg. Maar bereid je voor op vragen als:

- “Wat moet het systeem doen als dit invoerveld leeg blijft?”
- “Wat doen we als een postcode ontbreekt in de gegevens?”

Als jij er op dat moment niet bent om antwoord te geven, gebeurt er iets risicovols: er wordt gegokt. Iemand maakt zelf een interpretatie of stelt de beslissing uit. Het resultaat is dan een oplossing die voor 90% klopt, maar net op de cruciale details de plank misslaat. Daarom blijf je beschikbaar. Niet om politieagent te spelen, maar om de oorspronkelijke visie te bewaken.

Wat doe je concreet tijdens de uitvoering?
Je rol verschuift van ‘bedenker’ naar ‘gids’. Dit zijn je belangrijkste taken:

- **Vragen van het team beantwoorden**: Zorg dat je bereikbaar bent voor inhoudelijke vragen. Geef kort en helder antwoord. Jij maakt op dit moment het verschil tussen twijfel en duidelijkheid. Weet je het zelf ook even niet? Toets het dan bij de opdrachtgever. Het team kan dit soms ook zelf, maar jij kent de achtergrond en het doel het beste, waardoor je sneller tot de kern komt.
- **De bedoeling achter de oplossing bewaken**: Tijdens de bouw of uitvoering komen mensen vaak met nieuwe, slimme ideeën. Dat is prima en moet je vooral aanmoedigen. Jouw taak is om te toetsen of die ‘slimme aanpak’ nog wel past bij het oorspronkelijke doel. Wordt het probleem er nog steeds mee opgelost? Zo ja: doen.
- **Afwijkingen en sluiproutes signaleren**: Soms kiest men voor de makkelijke weg omdat er haast is. Dat is niet altijd erg, maar jij bent er om te toetsen of zo’n ‘shortcut’ acceptabel is. Past dit nog bij de afspraken die we gemaakt hebben, of leveren we nu te veel kwaliteit in?
- **Herkennen wanneer de aanpak niet meer klopt**: Soms zie je pas tijdens de uitvoering dat wat jullie bedacht hadden, in de praktijk toch niet lekker werkt. Durf dan op de rem te trappen. Stel de vraag: “Moeten we stug doorgaan, of moeten we het plan aanpassen?”

Je analyse is niet heilig
Een belangrijke valkuil is verliefd worden op je eigen ideeën en uitwerkingen. Behandel je analyse niet als de heilige graal waar niet van afgeweken mag worden. Het is de basis, maar de praktijk is de ultieme toets.

Moet er bijgestuurd worden? Prima. Doe dat dan altijd vanuit de kernvragen die je aan het begin stelde: "Wat moet dit oplossen?" en "Wat wilden we hiermee bereiken?" Zolang het antwoord daarop overeind blijft, mag de vorm best veranderen.

Feedback ophalen

Je oplossing is klaar, getest en overgedragen. Misschien wordt er zelfs al mee gewerkt. En dan komt dat onvermijdelijke moment. Tijdens een presentatie of in een mailtje hoor je: "Ziet er goed uit… maar nu ik het zo voor me zie, wil ik het toch net iets anders..."

Dat is frustrerend. Je hebt je hele analyse goed gedaan en dacht dat je klaar was. Toch is dit vaak geen onwil van de ander, maar een teken dat de feedback te laat is opgehaald. De regel in ons vak is simpel: hoe later de feedback komt, hoe meer tijd, energie en geld het kost om iets aan te passen.

Waarom mensen pas laat iets zeggen
Mensen wachten vaak tot het allerlaatste moment met hun commentaar. Dat doen ze meestal niet om vervelend te zijn, maar door een van de volgende oorzaken:

- Ze willen niet 'zeuren' of ondankbaar lijken: Vaak zijn mensen allang blij dat er hard gewerkt is en dat er een resultaat ligt. Ze houden hun twijfels voor zich om de sfeer goed te houden of omdat ze bang zijn dat hun opmerking als kritiek op jou persoonlijk wordt opgevat.

- Ze nemen aan dat jij de expert bent: De gedachte is vaak: “Er zal wel goed over nagedacht zijn door de specialisten, dus wie ben ik om daar iets van te vinden?” Hierdoor worden onlogische stappen in een proces soms onbedoeld geaccepteerd.

- Ze merken het pas in de praktijk: Een ontwerp op papier of een beschrijving in een document ziet er heel anders uit dan de werkelijkheid. Pas als iemand daadwerkelijk met de oplossing aan de slag gaat, voelt men waar het klopt en waar niet.

- Ze kunnen het probleem niet goed verwoorden: Soms hebben mensen alleen een onderbuikgevoel dat er iets niet klopt, maar weten ze niet precies wat de oplossing is. Omdat ze geen concrete verbetering kunnen voorstellen, zeggen ze dan maar niks vooraf.

Het is daarom jouw taak om actief feedback te vragen. Wacht niet af tot ze naar jou toe komen, want dan ben je vaak al te laat.

Wanneer (en hoe) haal je feedback?
Het korte antwoord: zo snel mogelijk. Wacht absoluut niet tot alles ‘af’ is. Zodra er iets is dat mensen kunnen zien, voelen of ervaren, kun je feedback vragen. Vormen waarin je dit kunt doen zijn:

- **Een prototype of schets**: Een simpele visuele weergave van een scherm of proces, zelfs al werkt er nog niets echt.

- **Een deels werkende oplossing**: Een klein onderdeel dat al wel functioneert, zodat men de basis kan testen.

- **Een simulatie**: Het naspelen van het nieuwe werkproces, eventueel gewoon op papier of met een rollenspel. Het is een goed idee om dit al eerder, bij de uitwerking, te doen, maar het is nooit te laat.

Het sleutelwoord hierbij is ervaren. Iemand die iets zelf probeert, geeft veel rijkere feedback dan iemand die alleen passief naar een presentatie kijkt of een tekst leest.

Hoe krijg je goede feedback los?
Mensen hebben de neiging om sociaal wenselijke antwoorden te geven. Om te horen wat ze echt vinden, gebruik je deze drie technieken:

1. **Laat mensen het zelf doen**: Geef ze de ruimte om de oplossing te gebruiken. Ga er niet naast zitten om uit te leggen hoe het moet, maar observeer in stilte.

   Wat mensen doen zegt vaak meer dan wat ze zeggen. Als je ziet dat iemand drie keer op de verkeerde plek zoekt, is dat waardevolle informatie, zelfs als die persoon achteraf zegt: "Ja hoor, het was wel duidelijk."

2. **Stel open vragen**: Zorg dat het veilig voelt om kritisch te zijn. Vermijd gesloten vragen of vragen als “Snap je het?” (niemand wil dom overkomen, dus iedereen zegt "ja"). Stel liever vragen die uitnodigen tot vertellen:

   - “Wat zou je op dit punt doen als ik er niet bij zat?”
   - “Wat viel je op tijdens het gebruik?”
   - “Wat werkte anders dan je had verwacht?”

3. **Neem het serieus, maar beloof niets**: Laat merken dat je blij bent met de kritische punten. Schrijf alles op waar de ander bij is. Wees echter wel duidelijk in je verwachtingsmanagement: je noteert alles om het te onderzoeken, maar je kunt niet toezeggen dat alles direct wordt aangepast.

Wat doe je met de feedback?
Ren niet meteen naar je bureau om alles om te gooien zodra je commentaar krijgt. Analyseer de feedback eerst rustig aan de hand van de volgende punten:

- **Wat is het kernprobleem?** Probeer voorbij de directe opmerking te kijken. Als iemand zegt "Ik wil deze knop rood", is het probleem misschien niet de kleur, maar dat de knop niet genoeg opvalt.

- **Is het incidenteel of structureel?** Heeft één persoon hier last van, of loopt iedereen hier tegenaan?
- **Is het een blokkade of een wens?** Kan men echt niet verder werken (een 'showstopper'), of is het slechts een "zou leuk zijn als"-verzoek?
- **Wat is de urgentie?** Moet dit nu opgelost worden, kan het wachten tot een volgende versie, of doen we er niets mee?

Bespreek deze analyse met je team en je opdrachtgever. Samen bepaal je de vervolgstap: direct aanpassen, terug naar de tekentafel, op de lijst zetten voor later, of bewust niets doen. Dat laatste is ook een valide keuze: soms is de huidige oplossing simpelweg goed genoeg.

Nazorg hoort er ook bij

De oplossing is opgeleverd en werkt zoals gepland. De vlag kan uit. Misschien denk je nu: "Mooi, nu ziet mijn werk er echt op." Maar nee, nog steeds niet.

De nazorgfase is noodzakelijk. Dit is namelijk het moment van de waarheid. Nu wordt duidelijk of alles wat je bedacht, afgestemd en getest hebt, ook echt blijvend werkt in de dagelijkse praktijk. Hoe goed je de overdracht ook hebt geregeld, jij blijft vaak degene die het beste begrijpt waarom het werkt zoals het werkt. En dus ook waar de zwakke plekken zitten.

Je rol verandert, maar verdwijnt niet
Het gaat er niet alleen om dat het nu werkt, maar dat het blijft aansluiten bij het werk van de mensen. Een goede analist blijft daarom in deze fase betrokken. Niet om 'helpdeskje' te spelen, maar om drie redenen:

- Jij kent de historie: Je weet precies waarom bepaalde keuzes ooit gemaakt zijn.
- Jij bent de brug: Je snapt het verschil tussen de theorie (het plan) en de praktijk, bijvoorbeeld als iets anders of minder gebruikt wordt dan verwacht.

- Jij herkent afwijkingen: Als er iets misgaat, zie jij sneller of het een incident is of een structurele weeffout.

De vragen die op je afkomen
In deze fase fungeer je vaak als vraagbaak voor de gebruikers en beheerders. Je kijkt mee bij vragen als:

- "Is dit een fout, of hoort dit zo?"
- "Waarom reageert het systeem hier anders dan we gewend zijn?"
- "Kunnen we dit kleinigheidje alsnog even aanpassen?"
- "Hoe zat dit proces ook alweer precies in elkaar?"

Let op de stille signalen
Naast het beantwoorden van vragen, doe je in deze fase ook aan observatie. Soms zeggen mensen namelijk niks, maar zie je aan de data of het gedrag dat er iets niet lekker loopt.

Wees alert op:

- **Afwijkende gegevens**: Data die raar of onlogisch oogt, is vaak een teken dat mensen het systeem op een manier gebruiken die niet verwacht werd (dit zou natuurlijk niet moeten kunnen, dus een aanpassing is wenselijk) of er gewoon simpelweg ergens iets mis gaat.
- **Creatieve omwegen**: Gebruikers die 'officieel' alles goed doen, maar in de praktijk briefjes, Excel-lijstjes of andere trucs gebruiken om om het systeem heen te werken.

Het signaleren van dit soort gedrag is geen controle, maar pure informatieanalyse. Het vertelt je waar de oplossing nog niet naadloos aansluit op de werkelijkheid.

Een uitgewerkt voorbeeld

Tot nu toe heb je vooral over de aanpak gelezen. Mooie woorden, heldere stappen. Maar als jij net zo in elkaar zit als ik, weet je dat één praktijkvoorbeeld vaak meer zegt dan tien pagina's theorie. Dus: hoe gaat het nou echt, van een vaag idee naar iets dat werkt?

Analyse is geen kwestie van een boekje lezen en klaar. Je moet meters maken. Ervaring opdoen, doorvragen, afstemmen, uitwerken, opnieuw beginnen. Soms zit je goed, soms ga je de mist in. Dat hoort erbij.

Het mooie van de werkwijze in dit boek is dat je op ieder moment kunt bijsturen. Jij ontdekt dingen tijdens het uitwerken, je team ziet tijdens de uitvoering waar nog aandacht nodig is, en je klant merkt snel wat wel en niet werkt.

Ben je dus niet meteen volledig in je analyse? Geen probleem. Als je maar met de juiste dingen bent begonnen. Dat geeft ruimte om te verbeteren, uit te breiden of bij te stellen.

Dat gezegd hebbende, hieronder zie je een volledig uitgewerkt voorbeeld van een klein ICT-project, van idee tot concreet resultaat, inclusief momenten waarop is bijgestuurd. Ook als je niet in de ICT werkt: vooral lezen. De aanpak die je hier ziet, geldt voor vrijwel elk project, ICT of niet.

Fase 1: Van behoefte naar richting
Een teamleider van de buitendienst komt naar je toe. "Ik wil graag een mooie planningstool, zodat ik werkzaamheden van teams kan plannen."

Dat klinkt als een duidelijke opdracht. Maar je voelt meteen: hier zit meer achter. Dus je gaat doorvragen:

- "Wat is precies het probleem dat je nu ervaart?"
- "Wat hopen jullie hiermee op te lossen?"
- "Wie gaan dit straks nog meer gebruiken?"

Al snel ontdek je hoe het nu gaat: er wordt gewerkt in een 'gedeeld' Excelbestand. In theorie. In de praktijk zwerven er veel kopieën rond. Het gevolg: Het is niet altijd duidelijk wat de ander doet, afspraken worden gemist medewerkers worden dubbel ingepland en er sluipen fouten in de facturatie.

De echte behoefte is dus geen "mooie tool voor teamleiders", maar overzicht en afstemming voor iedereen die door met de planning te maken heeft. Minder dubbel werk, minder chaos. Je bent benieuwd naar de andere gebruikers in het gehele planningsproces.

Wie heb je nodig?
Je brengt in kaart wie je moet spreken. Je vraagt steeds: "Wie raakt dit nog meer?" Zo ontstaat dit plaatje:

- Teamleiders: Willen makkelijk plannen en bijsturen.
- Medewerkers: Willen op hun telefoon zien waar ze morgen moeten zijn.
- De manager: Wil rapportages zien over productiviteit.
- De administratie: Wil de uren makkelijk inlezen in hun factuursysteem.

Je spreekt mensen uit al deze groepen, checkt of ze zich herkennen in de problemen en wensen en maakt schetsen om je ideeën te visualiseren (een tekening van het proces, schetsen van een weekoverzicht voor de planner en een mobiel scherm voor de medewerker).

Na een paar gesprekken is het beeld helder: je weet wat men wil, waarom men het wil en wat de oplossing globaal moet bieden. Gewapend met een vol hoofd en je notulen ga je terug naar je bureau.

Opknippen in bouwblokken
Je definieert vier logische bouwblokken en geeft ze prioriteit (MoSCoW):

1. Must Have: Als teamleider wil ik activiteiten plannen voor mijn team, zodat ik ze efficiënt kan inzetten.
2. Must Have: Als medewerker wil ik mijn weekplanning inzien, zodat ik mijn werk kan voorbereiden.
3. Should Have: Als manager wil ik rapportages inzien, zodat ik de inzet kan sturen.
4. Could Have: Als administratie wil ik data exporteren, zodat ik sneller kan factureren.

De focus voor de eerste versie ligt op de teamleiders en de medewerkers. Daar zit de grootste pijn. De rest komt later.

Fase 2: Details invullen
Nu zoom je in op de belangrijkste bouwblokken en werkt deze verder uit.

Bouwblok 1: Als teamleider wil ik activiteiten plannen.

- Details: Een activiteit heeft een titel, starttijd, eindtijd, type werk, medewerker en een locatie.
- Regel: Bij overschrijding van 8 uur ingepland werk op een dag volgt een melding.
- Regel: Bij dubbele boeking volgt een waarschuwing (maar het blokkeert niet, want overwerk komt voor).

Bouwblok 2: Als medewerker wil ik mijn planning zien.

- Details: Toon activiteiten per dag.
- Details: Toon activiteiten per week.
- Regel: Medewerker moet zelf notities kunnen plaatsen per activiteit.

Bovenstaande bouwblokken zijn geen volledig uitgewerkt voorbeeld, in de praktijk zal je meer details toevoegen zoals ontwerpschetsen, logica en uitzonderingen.

Je concludeert op basis van deze bouwblokken dat de teamleiders ook een overzicht van de totaalplanning nodig hebben en dit overzicht moeten kunnen filteren op team, dus schrijf je hier een extra bouwblok voor.

Zodra je klaar bent, bespreek de uitwerking eerst met de gebruikersgroepen. Samen kijk je ernaar: klopt alles? Ontbreekt er nog iets? Is de bedoeling helder? Als je geen grote gaten of onduidelijkheden meer ziet, concludeer je: dit is stevig genoeg om te starten.

De gebruikers zijn akkoord en jij ook. Tijd voor de volgende stap: het ontwikkelteam. Dat heeft trouwens niet stilgezeten. Terwijl jij druk was met de analyse, zijn zij op basis van de eerste opdracht alvast begonnen met het leggen van de technische basis.

Tijdens het gesprek met het team nemen jullie de bouwblokken door. Er komen vragen en suggesties naar boven:

- "Als we filteren op team mogelijk maken, is filteren op medewerker daar ook makkelijk aan toe te voegen."
- "Zou het niet handig zijn om een 'kopieer afspraak knop' te hebben?"

Fijn, zo'n meedenkend team. Zowel jij als de klant hadden hier nog niet aan gedacht. De zoekfunctie neem je direct mee (weinig werk), de kopieerfunctie zet je als bouwblok op de lijst voor later (nice to have).

Omdat je het goed had voorbereid, is er ruimte voor dit soort verbeteringen. Iedereen kan aan de slag.

Fase 3: Uitvoeren en bijsturen
De bouwers gaan los. Jij zit niet stil, maar werkt alvast aan de analyse voor de volgende blokken. Tussendoor beantwoord je vragen van het team:

Zo komt er een vraag binnen: "Wat doen we als iemand iets plant op een nationale feestdag?"

Die situatie zat niet in je ontwerp. Je overlegt: het mag wel, maar het moet zichtbaar zijn. Toevoegen dus.

Na twee weken is er een simpele, deels werkende versie. Je laat een teamleider en een medewerker testen. Zij reageren:

- "Het is overzichtelijk, ik snap het meteen."
- "Maar… ik wil echt die 'kopieer afspraak knop'. Ik plan vaak hetzelfde."
- "Ik zie niet wie mijn rooster heeft aangepast. Dat vind ik vervelend."

Er ontstaan nieuwe inzichten. Niet omdat er iets mis is, maar omdat gebruik altijd nieuwe inzichten oplevert. De kopieerfunctie had je al omdat het ontwikkelteam dit al voorgesteld had. Het blijkt bij de klant hoog op het verlanglijstje te staan. Daarmee schuift het bouwblok op, van later bouwen naar nu direct. Het ontwikkelteam neemt het meteen mee in de planning.

Ook maak je een nieuw bouwblok:

Als teamleider wil ik kunnen zien wie de planning heeft aangepast, zodat ik wijzigingen kan herleiden bij fouten.

Je bespreekt de uitwerking van deze nieuwe wens met je klant en het team en bepaalt de prioriteit. Jullie komen tot de conclusie dat deze wens belangrijker is dan rapportage voor management en administratie, dus ook dit bouwblok wordt besproken en meegenomen in de ontwikkeling.

Het resultaat

Nog eens twee weken later is de eerste versie klaar. Hebben we alles gebouwd wat men in het begin riep? Nee. De manager en de administratie moeten nog even wachten en het systeem is nog niet heel erg verfijnd. Maar wat er staat, lost wel het kernprobleem op: teamleiders kunnen foutloos plannen en medewerkers weten waar ze aan toe zijn. De chaos is weg.

Nu de basis staat en de gebruikers tevreden zijn, starten jullie met de tweede ronde voor de manager, en daarna de administratie. En als dat klaar is? Verder met verfijning en bouwblokken die een lage prioriteit gekregen hadden.

Je hebt niet zomaar 'een tooltje' gebouwd. Je hebt een oplossing neergezet die past bij de praktijk. Iets waar mensen echt blij van worden, en hun werk beter en makkelijker maakt. En dat is waar goede analyse om draait.

Wat je hiervan moet onthouden:

- **Begin nooit direct met bouwen:** De eerste vraag is zelden het echte probleem. Ga doorvragen.
- **Betrek alle groepen:** Vraag steeds: Wie gebruikt dit? Wie moet er iets van vinden?
- **Werk in bouwblokken:** Knip het op en bepaal wat de prioriteit op wat nu de meeste waarde levert (en knip ze nog kleiner als ze niet concreet genoeg zijn).
- **Vul details pas later in:** Werk details pas uit als een blok aan de beurt is en toets ze met gebruikers.
- **De gezamenlijke toets:** Betrek het ontwikkelteam vroegtijdig; daar komen vaak de beste ideeën vandaan.
- **Blijf beschikbaar:** Veel inzichten komen pas tijdens het bouwen en testen en zelfs nog na oplevering.
- **Omarm feedback:** Gebruik nieuwe inzichten om te verbeteren en durf prioriteiten te schuiven.

Gesprekstechnieken



Gesprekstechnieken

Je kunt de strakste modellen volgen en de slimste tools gebruiken. Je kunt werken met templates tot je erbij neervalt. Maar zonder een goed gesprek begin je niets.

Als analist is jouw belangrijkste instrument niet je laptop, een model of een systeem. Natuurlijk helpen die dingen je om je werk te structureren, maar de echte inhoud haal je uit een gesprek. Een gesprek waarin je scherp doorvraagt, de ruis wegfiltert en de werkelijke behoefte op tafel krijgt.

In dit hoofdstuk geef ik je handvatten om zulke gesprekken te voeren. Verwacht geen standaard vragenlijstjes die je als een robot afwerkt. Ik geef je technieken waarmee je echt tot de kern komt. Technieken die je helpen om de wensen, zorgen en aannames van je klant bloot te leggen. Ook als ze die zelf nog niet helder hebben.

Tekenen

Tekenen is eigenlijk geen echte gesprekstechniek, maar wel is het een van de krachtigste dingen die je kan doen bij gesprekken. Daarom wil ik eerst dit onderwerp behandelen voor we verder gaan naar de echte gesprekstechnieken.

Soms (vaak) kom je er met praten alleen niet uit. Iemand legt iets uit, maar het kwartje valt niet. Of de uitleg blijft vaag: “Ja, en dan loopt het vast bij de controle,” of “Daar wordt het proces een beetje onduidelijk.” Wat bedoelen ze dan precies? Waar gaat het mis? Hoe werkt dat dan? In zulke gevallen is er één oplossing die bijna altijd werkt: laat het tekenen.

Tekenen is ‘denken met een pen’. Door een situatie visueel te maken, maak je het abstracte concreet. Het dwingt iemand om logisch na te denken over de volgorde en de inhoud.

Voor jou als analist wordt meteen duidelijk waar het probleem zit. Aannames komen naar boven (“O, doen we dat altijd zo?”) en je ziet direct

waar afdelingen, systemen of mensen op elkaar botsen. Het hoeft geen kunstwerk te zijn; een simpele schets op een flipover of bierviltje zegt vaak meer dan twintig vragen.

Hoe doe je het?
Je hoeft geen kunstenaar te zijn. Je gesprekspartner ook niet. Pak een stift en papier (of loop naar het whiteboard) en begin. Vragen die je hierbij helpen:

- "Kun je eens tekenen hoe dat nu werkt?"
- "Teken eens uit wat je wanneer doet, en met wie?"
- "Zet eens een kruisje waar het precies misgaat?"

Kies een vorm die past bij het onderwerp:

- **Een stroomdiagram**: De vorm die je voor bijna alles kan gebruiken. Het mooie van deze vorm is dat je elke 'als-dan' eenvoudig inzichtelijk maakt. *(zie pagina 92 voor verdieping rondom dit type tekening).*
- **Een tijdlijn**: Voor processen die in de tijd verlopen.
- **Een blokkenschema**: Voor de route langs afdelingen of systemen.
- **Poppetjes**: Om te laten zien wie met wie samenwerkt.
- **Whatever works**: Als je geen idee hebt welke vorm past, combineer vormen, het gaat om de uitkomst.

Tip: Laat stiltes vallen als iemand tekent. Vaak is dat het moment waarop ze zelf ontdekken hoe hun proces eigenlijk werkt of eigenlijk niet klopt.

Alternatief: Naspelen
Sommige mensen tekenen niet graag of denken niet visueel. Geen probleem. Laat ze het dan naspelen of stap-voor-stap vertellen:

- "Wat doe jij als eerste?"
- "En wat doet je collega dan?"
- "En wat doe jij daarna?" Jij tekent ondertussen mee op papier om het alsnog visueel te maken.

Wanneer zet je dit in?
Zo vaak mogelijk, maar vooral als het gesprek blijft hangen in vage termen als "Gewoon, zoals het altijd gaat" of "Het loopt spaak tussen afdeling X en Y". Vooral bij overdrachtsmomenten (tussen mensen of systemen) is tekenen onmisbaar om de gaten in het verhaal te vinden.

Voorbeeld: Het verdwenen mailtje
Een medewerker vertelt dat leveringen soms vertragen tussen het magazijn en transport . Jij vraagt: "Teken dat eens?"

Ze schetst: Productie → Magazijn → Transport. Bij de pijl naar transport schrijft ze: "Via mailtje".

Jij wijst naar de pijl: "En wat gebeurt er als Transport dat mailtje mist?" Het antwoord: "Tja, dan blijft de levering staan."

Door die simpele tekening zag iedereen het probleem: het proces leunt op een onbetrouwbaar mailtje. De oplossing was niet 'beter opletten', maar een systeem dat een openstaande taak toont aan Transport.

Tips

- Moedig gekrabbel aan: Het gaat om inzicht, niet om kunst.
- Stel vragen tijdens het tekenen: “En wat gebeurt er daarna?”
- Laat momenten van stilte gebeuren: Dit is vaak denktijd
- Begin grof: Zoom pas in op details als dat nodig is.
- Vraag naar uitzonderingen: “Wat gebeurt er als dit niet volgens plan verloopt?”
- Leg het vast: Maak altijd een foto van de tekening. Ook die schets tussen neus en lippen door komt later nog van pas.

Niet doen

- Te veel detail: Als je elke muisklik tekent, raak je het overzicht kwijt.
- Te mooi willen doen: Dan gaat de aandacht naar de vorm in plaats van de inhoud.
- Gaten zelf invullen: Teken alleen wat de ander zegt, anders verlies je hun perspectief.

Gedragsgericht doorvragen

Mensen vertellen graag wat ze willen: “Het moet makkelijker, sneller en gebruiksvriendelijker.” Als jij als analist meteen roept: “Oké, gaan we bouwen!”, mis je cruciale informatie. Want wat betekent ‘gebruiksvriendelijk’ voor deze persoon?

Gedragsgericht doorvragen betekent dat je een stap terugzet: van wens naar werkelijkheid. Je begint niet bij de gedroomde oplossing, maar bij wat mensen nu doen. Hoe werken ze? Wat gaat er mis? Wat proberen ze te bereiken? Mensen zijn vaak zo gewend aan hun eigen ‘workarounds’ (omwegen) dat ze die niet meer als probleem zien, tot jij ernaar vraagt.

Hoe doe je het?
Je focust niet op de mening, maar op het feitelijke gedrag.

- "Wanneer gaat het nu fout?" (Zoek de oorzaak, niet het drama).
- "Wat probeer je op dat moment te doen?" (Zoek de intentie/het doel).
- "Wat lukt er dan niet?" (Het verschil tussen wens en realiteit).
- "Wat zou je willen dat er gebeurt?" (De gewenste situatie in gedrag, niet in oplossingen).

Wanneer zet je dit in?
Zodra mensen abstracte termen gebruiken:

- "Het moet makkelijker."
- "We willen een dashboard."
- "Er moet een knop komen."

Zodra je een oplossing hoort zonder context ("daar moeten we dan een dashboard voor hebben"), moet er een alarmbelletje afgaan: Terug naar het gedrag!

Voorbeeld: De terugbelverzoeken
Een helpdeskmedewerker zegt: "Ik wil snel zien welke terugbel-verzoeken voor mij openstaan."

Jij vraagt door:
- "Wanneer doe je dat nu?" → "Elke ochtend bij de start."
- "Wat doe je dan precies?" → "Ik moet elk verzoek openklikken om te zien of een collega het al heeft opgepakt."
- "Wat gaat daar fout?" → "Het kost me een veel te veel tijd om te zien wie ik moet bellen."
- "Wat wil je dat er gebeurt?" → "Gewoon een lijstje met namen die nog aan niemand gekoppeld zijn."

Nu heb je geen vage wens meer, maar een concreet doel.

Tips

- Luister echt: Heb je antwoord niet al klaar.
- Laat het voordoen: Laat iemand zijn scherm delen en hardop denken terwijl hij werkt.
- Check: Vat samen wat je hoort en ziet ("Dus je klikt nu vijf keer?").

Niet doen

- Meteen oplossingen aandragen: Dan stop je met luisteren.
- Gesloten vragen stellen: "Is het nu lastig?" (Ja/Nee helpt je niet).
- Aannames doen: "Ik snap het proces wel." (De praktijk is vaak anders dan de theorie).

5x waarom

Stel: iemand wil "een bevestigingsknop". Klinkt logisch. Maar waarom? Omdat ze vergeten iets te doen? Omdat ze onzeker zijn? Omdat hun manager bewijs wil zien? Zonder het 'waarom' weet je niet of die knop het echte probleem oplost.

Het concept van de 5x waarom is simpel: je vraagt net zo lang door naar de reden achter de reden, tot je bij de kern bent. De theorie zegt '5 keer', maar soms ben je er al na 3 keer. Het gaat niet om het tellen, maar om het afpellen van de lagen.

Lijkt op gedragsgericht doorvragen?
Ja, ze hebben veel gemeen. In beide gevallen zoek je naar wat er echt speelt, niet naar het eerste antwoord. Maar er is een belangrijk verschil:

- Gedragsgericht: Wat gebeurt er concreet? (De feiten).

- 5x Waarom: Waarom is dit belangrijk? (De motivatie).

Samen zijn ze sterk. Gebruik gedrag om het probleem zichtbaar te maken, en 'waarom' om de reden erachter te begrijpen.

Hoe doe je het?
Waarschuwing: als je als een peuter vijf keer letterlijk "Waarom?" vraagt, word je irritant. Doe dat dus niet.

Deze techniek vraagt geduld en een nieuwsgierige houding. Je krijgt vaak eerst een oppervlakkig antwoord of een oplossing, maar dat is zelden het echte motief. Door rustig en vriendelijk verder te vragen kom je stap voor stap dichter bij de kern

Varieer je vragen:

- "Wat maakt dit belangrijk voor jou?"
- "Wat gebeurt er als we dit niet doen?"
- "Wat is het gevolg daarvan?"
- "En waarom is dat een probleem?"

Met dit soort vragen dwing je de ander om verder te denken dan de eerste ingeving. Richt je op het motief achter de wens, niet op de wens zelf.
Wanneer zet je dit in?

- Als iemand met een kant-en-klare oplossing komt ("We willen een rapport").
- Als termen vaag blijven ("We willen meer controle").

Voorbeeld: Het dashboard

Een manager van een callcenter zegt: "Ik wil een nieuw dashboard."

1. "Wat wil je daarop zien?" → "De voortgang van de tickets."
2. "Waarom is dat belangrijk?" → "Zodat ik zie of we op schema liggen."
3. "Wat gebeurt er als je dat niet ziet?" → "Dan hoor ik pas achteraf dat klanten te laat zijn geholpen."
4. "En wat is daar het gevolg van?" → "Dan verliezen we het vertrouwen van grote klanten."

Conclusie: Het gaat niet om een mooi dashboard met grafiekjes. Het gaat om het voorkomen van vertraging om klantvertrouwen te behouden. Misschien is een proactieve melding bij vertraging wel veel beter dan een dashboard waar hij zelf op moet kijken.

Tips

- Wees oprecht nieuwsgierig: Toon en houding maken het verschil tussen interesse en een kruisverhoor.
- Niet zelf invullen: Soms is het antwoord heel anders dan je zelf denkt.
- Benoem de kern: "Aha, dus het gaat eigenlijk om klantvertrouwen."
- Combineer met gedragsgericht doorvragen: gedrag maakt het probleem zichtbaar, waarom geeft de reden erachter.

Niet doen

- Letterlijk 5x waarom vragen: Dat werkt alleen bij kleuters.
- Te vroeg stoppen: Tevreden zijn met het eerste antwoord.
- Oordelend klinken: "Waarom doe je dat nou?" klinkt als een verwijt.

Focus op emoties

In zakelijke gesprekken negeren we emoties vaak. We willen professioneel zijn. Maar frustratie, irritatie, trots of juist moedeloosheid zijn geen ruis. Als je wilt weten wat mensen echt belangrijk vinden zijn dit juist belangrijke aanwijzers.

Emotiegericht doorvragen is het oppikken van signalen. Een diepe zucht, een oog-rol, een sarcastisch grapje. Het zijn geen verstoringen in het gesprek, het is (vaak onbewust) waardevolle informatie waar je op door kan vragen.

Hoe doe je het?
Let op meer dan alleen woorden. Merk je iets aan de toon, ondertoon of lichaamshouding? Dit zijn allemaal emoties. Maak ze bespreekbaar en koppel ze direct aan het werk.

- Vraag waar emotie vandaan komt: “Wat maakt dit zo irritant?”
- Vraag ook naar positieve emoties: “Wanneer voelt je werk waardevol?”
- Koppel de emotie terug aan het proces: “Wat betekent dit voor je werkdag?”
- Blijf professioneel: hou het bij werk en blijf uit iemands privéproblemen.

Let vooral op de non-verbale signalen en hou het gerelateerd aan het werk.

Wanneer zet je dit in?
Als je merkt dat er irritatie of frustratie zit. Achter elke frustratie zit een verlangen naar hoe het beter kan. Emoties geven je inzicht in werkdruk, inefficiëntie en dingen die mensen normaal niet hardop uitspreken.

Emotie geeft je:

- Direct zicht op wat echt belangrijk is voor iemand
- Inzicht in werkdruk, inefficiëntie of gebrekkige samenwerking
- Aanknopingspunten voor wat beter moet, maar vaak niet wordt uitgesproken
- De mogelijkheid om de gebruiker echt centraal te zetten

Voorbeeld: De maandagochtend

Een medewerker zegt tussen neus en lippen door: "Ik weet vaak pas maandagochtend wat ik die week bij klanten moet doen. Daar word ik zo moe van."
Jij vraagt door:

- "Wat doet dat met je?" → "De ochtend ben ik kwijt aan het maken van mijn weekschema. Het voelt alsof mijn tijd niet serieus genomen wordt."

Je analyse verschuift nu van 'informatievoorziening' naar 'werkdruk en waardering'. Dat is een essentiële nuance voor de oplossing.

Let op: Je bent geen therapeut
Het kan verleidelijk zijn om, zodra iemand emotie toont, de rol van hulpverlener of coach aan te nemen. Dat is niet jouw taak. Je hoeft geen therapiegesprek te voeren of iemands privéleven boven tafel te krijgen. Het gaat je uitsluitend om de beleving in het werk.

Ga dus niet (teveel) mee in persoonlijke zijpaden. Begint iemand over een scheiding of ziekte? Erken het kort en respectvol ("Wat vervelend dat je dat meemaakt"), maar stuur het gesprek daarna vriendelijk terug naar het werkproces. Vraag niet naar trauma's, maar naar hoe het werk voelt. Daarmee blijf je professioneel en bewaak je je eigen grenzen.

Tips
Als je emoties serieus neemt, breng je vaak de diepere behoefte boven tafel.

- Benoem wat je ziet: “Ik merk dat dit onderwerp je frustreert.”
- Wees respectvol: Geef ruimte aan het gevoel, maar hou de link met het werk vast.
- Vraag naar beide kanten: Focus niet alleen op wat energie kost, maar ook op wat energie geeft.
- Gebruik stilte: Als iemand even stilvalt, is dat vaak een teken dat ze nadenken. Vul die stilte niet te snel in.
- Check je aanname: “Ik heb het idee dat je hier boos om wordt, klopt dat?” Ga niet gissen.

Niet doen
Emoties zijn krachtig, maar verkeerd omgaan met signalen kan het gesprek schaden.

- Emoties wegwuiven: “Ach, stel je niet aan” of “Wat een dramaqueen” is makkelijk gedacht, maar dan mis je wat echt belangrijk is voor die persoon.
- Therapeut spelen: Nogmaals: blijf bij de procesanalyse, niet het privéleven.
- Frustratie alleen noteren: “Piet is boos” is geen analyse. Vertaal het naar proces of gedrag (“Het proces faalt bij stap X, wat frustratie oplevert”).
- Alleen negatieve emoties uitvragen: Positieve ervaringen (trots, blijdschap) zijn net zo waardevol om te weten wat je moet behouden.
- Non-verbale signalen negeren: Iemand die "ja" zegt maar heel hard "nee" schudt of kijkt, bedoelt waarschijnlijk nee.

Toekomstbeeld schetsen

Soms blijft een gesprek hangen in frustratie. Mensen weten haarfijn te vertellen wat er allemaal niet werkt, maar vinden het moeilijk om te formuleren wat ze wel willen. De 'toekomstvraag' is de breekijzer-techniek die helpt om even los te komen van de dagelijkse rompslomp en vooruit te dromen.

Door het schetsen van de toekomst nodig je de ander uit om zich voor te stellen dat het probleem al is opgelost. Dit dwingt iemand om afstand te nemen van de huidige beperkingen en te denken in effect, impact en gevoel. Je vraagt niet: "Wat wil je gebouwd hebben?" (de techniek), maar: "Stel dat het straks perfect werkt, wat merk je dan?" (het effect).

Voor jou als analist is dit heel waardevol. Vanuit dat perfecte plaatje kun je namelijk veel makkelijker terug redeneren:

- Wat is er nodig om dat gevoel te bereiken?
- Welke functies dragen daar echt aan bij?
- En wat moet het systeem vooral niet meer doen?

Hoe doe je het?
Je zorgt ervoor dat mensen uitspreken wat ze in hun hoofd hebben. Geef ze de ruimte om te fantaseren, maar probeer het wel weer concreet te maken. Gebruik deze vragen als leidraad:

- **De magische vraag**: "Stel, over drie maanden werkt dit perfect. Wat kun je dan wel, wat je nu niet kunt?" (Zo kom je bij concrete wensen).
- **De impact**: "Wat merk je dan concreet aan je werkdag?" (Dit brengt de waarde in beeld).
- **De visualisatie**: "Wat zie je dan op je scherm, zonder dat je hoeft te zoeken?" (Dit maakt de gewenste situatie tastbaar).
- **De ballast**: "Wat is er dan niet meer nodig?" (Hiermee ontdek je welke onzin er uit het proces gesloopt mag worden).

Wanneer zet je dit in?

- Gebruik deze techniek als:
- Mensen blijven hangen in klagen of vage oplossingen (“Ik wil een knop die dat doet”), maar geen heldere wensen formuleren.
- Je het gevoel hebt dat de werkelijke behoefte nog steeds niet op tafel ligt.
- Je richting zoekt voor je prioritering of visueel ontwerp.

Voorbeeld: De declaratie

Een financieel medewerker klaagt steen en been: “Voor één declaratie moet ik door vier systemen heen. Dat is zo irritant en foutgevoelig.”

Jij stelt de toekomstvraag: “Stel dat het straks perfect werkt. Hoe ziet dat er dan uit?”

Het antwoord: “Dan open ik één scherm, sleep het bonnetje erin, en is het meteen verwerkt. Geen losse Excelletjes meer, geen dubbele controles.”

De analyse: Je hebt nu een helder toekomstbeeld dat richting geeft. Het doel is niet ‘betere koppelingen bouwen’, het doel is: één scherm, slepen, klaar. Dat staat voor eenvoud, snelheid en minder fouten.

Tips
Met een toekomstvraag ontdek je vaak de diepere wens achter alle klachten.

- Help mensen op weg: Mensen vinden deze vraag soms wat zweverig. Geef zelf een voorbeeld om het ijs te breken.
- Stel vervolgvragen: Neem geen genoegen met “Dan is het fijn”. Vraag door: “Wat maakt dat belangrijk?” of “Wat levert dat op?”
- Gebruik het als toets: Gebruik het toekomstbeeld om later te checken of je ontwerp klopt. (“Helpt dit ontwerp om dat gevoel te bereiken?”).
- Benoem de waarde: Zeg hardop wat je hoort: “Je zoekt dus vooral snelheid en overzicht.”
- Houd het concreet: Breng de droom altijd terug naar gedrag of de werkdag.

Niet doen
Toekomstvragen zijn krachtig, maar verkeerd gesteld leveren ze alleen maar lucht op.

- Te vaag vragen: “Stel dat alles perfect is...” is te groot zonder een concreet onderwerp.
- Vage antwoorden accepteren: “Dan werkt het meteen” zegt niets over wat er nodig is. Vraag: “Hoe zie je dat dan?”
- Alleen dromen verzamelen: Zonder vertaling naar acties en eisen blijft het een droom. Jij moet de vertaalslag maken.
- Te snel tevreden zijn: Het eerste antwoord is vaak oppervlakkig. Graaf dieper.
- Geen houvast bieden: Sommige mensen blokkeren bij ‘stel je voor’. Geef ze concrete haakjes of scenario’s.

Vraag naar eerdere oplossingen

In bijna elk project hoor je dezelfde uitspraken: “Dit werkt echt niet meer” of “We hebben echt een systeem nodig dat dit oplost.” Dat klinkt logisch en dringend. Maar voordat je direct in de oplossingen duikt, is er één vraag die je altijd moet stellen: “Wat hebben jullie hiervoor al geprobeerd?”

Het antwoord daarop is vaak verrassend waardevol. Misschien ontdek je dat er al eens een vergelijkbaar systeem is ingevoerd, maar dat niemand het gebruikte. Of dat er prima ideeën waren die strandden omdat de organisatie er nog niet klaar voor was. Soms bleek iets technisch onhaalbaar, soms liep het spaak op weerstand of gebrek aan draagvlak.

Deze techniek draait om het ophalen van de geschiedenis. Je duikt in het verleden van eerdere oplossingen. Niet om mensen af te rekenen op wat misging, maar om te leren van die pogingen.

Als je weet wat mensen al geprobeerd hebben, ontdek je namelijk niet alleen mislukkingen, maar ook:

- Wat ze belangrijk vinden: Want dat hebben ze geprobeerd te regelen.
- Wat ze irritant vonden: Want dat werkte niet lekker of werd niet gebruikt.
- De beperkingen: Waar liggen de technische of organisatorische grenzen?

En misschien wel het allerbelangrijkste: je voorkomt dat jij met een “briljant idee” komt, dat iemand een jaar geleden al geprobeerd heeft en dat toen grandioos geflopt is.

Hoe doe je het?
Je vraagt er expliciet naar en verkent samen wat er gebeurde. Begin simpel:

- "Hebben jullie dit probleem al eerder geprobeerd te tackelen?" en bouw daarna uit met vervolgvragen:
- De betrokkenen: "Wie waren er toen bij betrokken?"
- De reden: "Waarom is het gestopt of mislukt?"
- De nuance: "Wat werkte er toen wel, en wat niet?"
- De beleving: "Wat vonden de gebruikers ervan?"
- De oplossing: "Wat zou het volgens jou wel werkbaar maken?"

Belangrijk is de sfeer: geef ruimte om mislukkingen te delen zonder oordeel. Veel mensen vinden het een beetje ongemakkelijk om over oude, mislukte projecten, te praten als er een nieuwe kans is op een oplossing, terwijl die informatie voor jou juist heel waardevol is.

Wanneer zet je dit in?
Het is goed om altijd te vragen naar eerdere oplossingen, maar gebruik deze techniek vooral:

- Als iemand een wens noemt voor iets dat er al was ("Vroeger hadden we…"). Waarom is dat er nu niet meer dan?
- Bij onderwerpen waar veel weerstand of scepsis leeft ("Daar gaan we weer…"); vaak komt dat door eerdere slechte ervaringen.

Voorbeeld: Nog een SharePoint?

Je analyseert een intern kennisdelingssysteem. Een medewerker zegt: "We hebben echt een platform nodig waar mensen hun kennis delen."

Je vraagt: "Hebben jullie dat eerder geprobeerd?"
Medewerker: "Ja, we hadden SharePoint. Maar niemand gebruikte het."
Je vraagt door: "Waarom niet?"
Medewerker: "Je moest zelf notificaties aanzetten om updates te krijgen. Niemand wist dat. En als je iets zocht, vond je het nooit."

De analyse: Nu weet je dat de wens niet simpelweg "een platform" is (want dat hadden ze al). De echte eisen zijn:

- Automatische notificaties bij nieuwe updates
- Een ijzersterke zoekfunctie.

Daarmee ga je verder. En je voorkomt dat je de fout maakt om "gewoon nog een SharePoint" neer te zetten.

Tips
Met deze vraag toon je respect voor wat er al gedaan is en leer je van de geschiedenis.

- Benoem je intentie: Zeg erbij dat je niet oordeelt, maar wilt begrijpen hoe het beter kan.
- Zoek de lichtpuntjes: Vraag ook naar successen. Wat was er wel goed aan de vorige poging?
- Kijk breder: Let op politieke of organisatorische redenen. Soms was de oplossing goed, maar de timing fout.
- Check de beeldvorming: Heeft de vorige mislukking gezorgd voor een negatief imago van dit soort projecten?

Niet doen
Het verleden ophalen is waardevol, maar als je het verkeerd doet, roept het weerstand op.

- Oordelen: Zeg nooit: "Ja, logisch dat dat niet werkte." Dan klappen mensen dicht.
- Alleen naar mislukkingen kijken: Dan mis je de onderdelen die wel waarde hadden en die je misschien kunt hergebruiken.
- Organisatorische oorzaken negeren: Soms sneuvelde iets door tijdgebrek of budget, terwijl het idee zelf prima was.
- Arrogant zijn: Te snel denken dat jouw idee beter is. Zonder check loop je het risico precies dezelfde fout te maken.

Stel, het is mislukt

Je zit met een klant aan tafel. Er is een idee, er ligt een plan en iedereen is enthousiast. De sfeer is goed. En dan stel jij die ene, onverwachte vraag: "Stel dat dit project over zes maanden faliekant mislukt is. Wat is er dan misgegaan?"

Het klinkt somber en misschien zelfs een beetje als een sfeerverpester. Maar vergis je niet: dit is een van de meest waardevolle vragen die je kunt stellen.

Deze techniek (ook wel een pre-mortem genoemd) draait om het bewust uitlokken van twijfel. Als mensen plannen maken, kijken ze meestal naar het beste scenario. Ze zien voor zich hoe het straks werkt en hoe mooi het eruitziet.

Daardoor blijven risico's, blinde vlekken of slechte ervaringen uit het verleden onder de radar. Door expliciet te vragen naar wat er mis kan gaan, maak je die schaduwkant bespreekbaar. Het geeft jou direct zicht op organisatorische, technische en menselijke risico's. Bovendien laat het zien dat je niet blind optimistisch bent, maar realistisch meedenkt over het succes.

Hoe doe je het?
Je gebruikt vragen die de ander uitnodigen om hun zorgen hardop uit te spreken. Je doet dat op een vriendelijke, open manier. Niet alsof je hoopt dat het fout gaat, maar omdat je wilt voorkomen dat het fout gaat.

Vragen die je kunt stellen:

- "Stel dat dit systeem straks live staat en toch niet werkt. Wat is er dan fout gegaan?"
- "Wat zouden we op dit moment over het hoofd kunnen zien?"
- "Waar maak jij je stiekem een beetje zorgen over?"
- "Wat is er eerder in soortgelijke trajecten misgegaan?"

Het doel is niet om iemand bang te maken, maar om te begrijpen welke risico's er leven in de hoofden van de mensen aan tafel.

Wanneer zet je dit in?
Timing is belangrijk bij deze techniek. Zet hem in zodra er een richting, plan of idee begint te ontstaan.

- Niet te vroeg: Tijdens een brainstorm werkt het verlammend en slaat het de creativiteit dood.
- Niet te laat: Als de keuzes al zijn gemaakt.

Het beste moment is als iedereen net enthousiast wordt en de plannen vorm krijgen. Zorg wel voor een rustige setting waarin mensen vrijuit durven praten. Doe dit niet even snel tussen de soep en de aardappels door of aan het einde van een vluchtige vergadering.

Voorbeeld: De selfservice-portal

Een organisatie werkt aan een nieuwe selfservice-portal voor medewerkers. Iedereen is blij, het ontwerp ziet er gelikt uit.
Jij stelt de vraag: "Stel dat dit straks live staat, maar het wordt toch niet gebruikt. Wat is er dan fout gegaan?"

De antwoorden volgen meteen:

- "Dan is het waarschijnlijk te ingewikkeld."
- "Veel van onze mensen zijn helemaal niet handig met computers."
- "Of mensen weten niet eens dat het bestaat, net als bij de vorige tool."

In één gesprekje van vijf minuten ontdek je drie cruciale risico's.

1. Als het niet intuïtief is, haken mensen af (UX-design).
2. Er is behoefte aan duidelijke uitleg of training (Educatie).
3. De interne communicatie moet op orde zijn (Marketing).

Met die inzichten kun je nu, nog voor de bouw, bijsturen.

Tips
Als je het goed inzet, levert deze techniek een lijst van reële aandachtspunten op die je analyse veel sterker maken.

- Blijf positief in je bewoording: Leg uit dat dit scenario's zijn waar niemand op zit te wachten, en dat je ze daarom nu bespreekt.
- Vraag door: "Wat maakt dat precies een risico?"
- Analyseer het realisme: Niet elke zorg is terecht. Filter de reële risico's eruit.
- Bespreek de oplossing: Kunnen we dit ondervangen? En hoe dan?
- Bepaal de impact: Raakt dit jouw analyse? Moet jij een extra eis of functie toevoegen?

Niet doen
De kracht van deze techniek zit in de uitvoering.

- Te vroeg gebruiken: Dan sla je elk enthousiasme meteen dood.
- Negatief blijven hangen: Het gesprek moet realistisch zijn, niet somber of cynisch.
- Risico's wel ophalen maar er niets mee doen: Dat is dodelijk voor het vertrouwen. Mensen voelen zich dan niet serieus genomen.
- Alles even belangrijk vinden: Maak onderscheid tussen 'kleine hobbels' en 'showstoppers'.
- Te vluchtig bespreken: In een snelle meeting durven mensen hun echte zorgen vaak niet te delen. Neem er de tijd voor.

Samenvatten en toetsen

Je denkt dat je elkaar snapt maar een week later blijkt dat jullie totaal iets anders bedoelden. Zonde van de tijd, en nog erger: zonde van het vertrouwen. Precies daarom is dit een van de krachtigste technieken die je als analist kunt inzetten.

Samenvatten en toetsen betekent dat je de kern van wat iemand vertelt in je eigen woorden teruggeeft, en direct checkt of dat klopt. Het lijkt simpel, maar het effect is groot. Mensen denken vaak dat ze helder communiceren, terwijl taal altijd vaag en contextgevoelig is. Door te parafraseren maak je impliciete aannames expliciet. Bovendien voelt de ander zich gehoord, en dat versterkt de samenwerking enorm.

Waarom dit werkt:

- Scherpte: Je dwingt jezelf om actief te luisteren.
- Controle: Je checkt direct jezelf of je de boodschap goed hebt begrepen. Als je het niet in je eigen woorden kan navertellen heb je het nog niet begrepen.
- Preventie: Je voorkomt misverstanden achteraf.
- Verbinding: Je laat de ander voelen: “Hé, die luistert echt naar mij.”

Hoe doe je het?
Je luistert actief en vat dan samen. Gebruik de woorden van de ander, maar niet letterlijk. Vertaal het naar jouw interpretatie en maak het concreet.

Zeg bijvoorbeeld:

- “Als ik het goed begrijp, wil je dat je op één scherm kunt zien wie er nog actie moet ondernemen?”
- “Dus eigenlijk wil je dat het systeem jou waarschuwt als een stap te lang blijft liggen?”

- “Je bedoelt dus: de klant krijgt wel een bevestiging, maar het interne team weet dat nog niet automatisch?”

Daarna komt de belangrijkste stap: de toets. Vraag expliciet:

- “Klopt dat?”
- “Mis ik nog iets?”
- “Is dat wat je bedoelt?”

Help mensen structureren
Let op: mensen vinden het soms lastig om iets uit te leggen. Ook dan is deze techniek handig om het scherper te krijgen. Je helpt ze hun gedachten te ordenen. Zeg dan: “Dus als we dit stap voor stap bekijken: er gebeurt eerst X, dan Y, en dan loopt het vast bij Z. Klopt dat?”

Let op de stilte
Soms blijft het na jouw samenvatting even stil. Laat dat gebeuren. Stilte is vaak het moment waarop iemand nadenkt of jouw samenvatting echt de lading dekt. Zie je aarzeling in hun ogen? Grijp die kans:

- “Ik zie dat je twijfelt?”
- “Wat zit er nog niet helemaal lekker?”

Wanneer pas je dit toe?
Gebruik samenvatten en toetsen op deze momenten:

- Aan het eind van een uitleg.
- Als je merkt dat iemand afdwaalt of vaag blijft.
- Als je zelf begint te twijfelen of je het goed snapt.
- Als je voelt dat jullie elkaar níet helemaal snappen, maar niemand het hardop zegt.

Het lijkt misschien eenvoudig, maar het is een cruciale vaardigheid. Het geeft helderheid, voorkomt misverstanden en levert vaak nieuwe inzichten op. Liever te vaak gedaan dan te weinig.

Voorbeeld: actieve waarschuwing of niet

Een klant vertelt een lang verhaal over een nieuwe wens. Jij vat samen: "Dus als ik het goed begrijp, wil je vooral dat het systeem je actief waarschuwt zodra de goedkeuring van de manager te lang blijft liggen?"

Scenario A: De klant knikt enthousiast. "Ja, precies dat!" (Check, je zit goed).

Scenario B: De klant zegt: "Nee, dat is niet helemaal wat ik bedoel. Ik wil het wel zien, maar ik hoef geen waarschuwing." (Check, je hebt een misverstand voorkomen).

Door actief te toetsen voorkom je overbodig werk.

Tips

Gebruik deze techniek zo veel mogelijk. Als je dit consequent doet, versterkt het zowel de inhoud als de relatie.

- Parafraseer: Gebruik je eigen woorden, niet die van de ander. Alleen zo check je of jij het snapt.
- Vraag altijd expliciet of het klopt: Ga er niet vanuit.
- Gebruik stiltes: Die geven de ander ruimte om na te denken.
- Help zoeken: Vat ook stap-voor-stap samen als iemand zelf nog zoekende is.
- Frequentie: Doe het liever te vaak dan te weinig om misverstanden te voorkomen.

Niet doen
Samenvatten lijkt makkelijk, maar verkeerd gedaan levert het weinig op.

- Papegaaien: Letterlijk herhalen wat de ander zei. Dat voegt niets toe en kan irritant zijn.
- Overslaan: Denken “Ik snap het wel”. Juist dan ontstaan de grootste fouten.
- Geen check doen: Zonder de vraag “Klopt dat?” is je samenvatting waardeloos.
- Te oppervlakkig samenvatten: Zorg dat je de diepte pakt, anders gaat de nuance verloren.
- Het als formaliteit zien: Het is een gesprekstechniek om te begrijpen, geen afvinklijstje.

Technieken die je werk structureren



Technieken die je werk structureren

Gesprekken voeren is de basis van je vak. Met goede vragen, doorvragen en samenvatten kom je een heel eind. Maar met alleen een goed gesprek ben je er nog niet.

Informatie ophalen is één ding, er iets zinnigs mee doen is een tweede. Een gesprek alleen levert namelijk nog geen strakke prioriteitenlijst op. Het geeft nog geen gedeeld beeld van wie de gebruikers zijn. En het maakt nog geen harde keuzes tussen "leuk om te hebben" en "absoluut noodzakelijk".

Daar heb je andere technieken voor nodig. Zie dit als de inhoud van je gereedschapskist. Ik heb het hier niet over zware, theoretische modellen die je invult en vervolgens in een lade laat verstoffen. Ik heb het over praktische manieren om structuur te geven aan de chaos die je hebt opgehaald. Het doel is niet om dikke rapporten te schrijven, maar om snel helderheid te scheppen.

Denk aan situaties zoals:

- Visualiseren: Hoe teken je een ingewikkeld proces zo uit, dat iedereen meteen ziet waar het vastloopt?
- Prioriteren: Wat doe je als de klant roept dat alles prioriteit 1 heeft?
- Belanghebbenden: Hoe breng je in kaart wie er echt toe doet (ook als ze zichzelf niet of juist wel zo belangrijk vinden)?
- Bouwblokken klein maken: Hoe knip je grote wensen op in blokjes die direct waarde leveren?.

In dit deel bespreken we de technieken die je hierbij helpen. Zie ze niet als verplichte nummertjes, maar als handvatten die je flexibel inzet. Je moet alleen weten welk gereedschap je uit de kist pakt om de klus te klaren.

Het stroomdiagram

Bij het hoofdstuk over gesprekstechnieken geef ik een aantal voorbeelden maar stroomdiagrammen zullen vrij zeker je favoriet zijn. Deze techniek kan je namelijk altijd inzetten, ongeacht het probleem of waar je bent in je analyse. Of je nog aan het sparren bent of bezig met je verdieping, dit diagram geeft altijd overzicht.

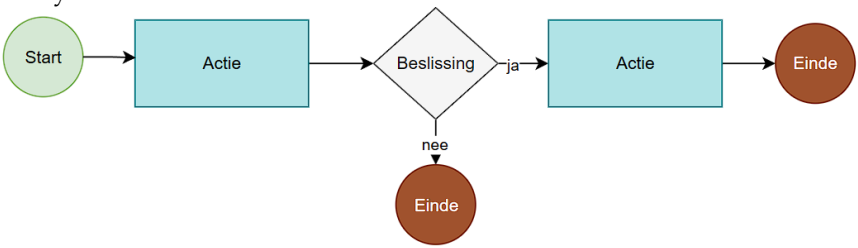
Een stroomdiagram is een visueel overzicht van een proces. Het laat zien welke stappen je zet, welke keuzes je onderweg maakt en waar je uiteindelijk uitkomt. Het dwingt je om logisch na te denken: na stap A moet er een pijl lopen naar stap B. Als de pijl nergens heen gaat, klopt je proces niet.

Waarom werkt dit voor bijna alles?
Mensen vertellen verhalen vaak chaotisch. Ze springen van de hak op de tak: "Ja, als de aanvraag binnenkomt, dan kijken we ernaar, tenzij hij incompleet is, oh ja, en Piet moet hem ook nog goedkeuren, maar dat hoeft niet als het bedrag onder de 500 euro is."

Als je dit uittekent in een stroomdiagram, zie je meteen de gaten.

- Wat gebeurt er als Piet ziek is?
- Waar gaat de incomplete aanvraag heen? Terug naar de klant? De prullenbak in?

Hoe doe je het?
Je kunt stroomdiagrammen zo ingewikkeld maken als je wilt (Google ze...), maar in de praktijk heb je voor 95% van je werk genoeg aan deze vier symbolen:





De Ovaal/Cirkel (Start/Einde)**:** Hier begint en stopt het proces. Elk schema heeft er minimaal twee.



De Rechthoek (De Actie): Iets wat gedaan wordt. "Vul formulier in", "Bereken totaal", "Stuur e-mail".



De Ruit (De Beslissing): Het kruispunt. Hier wordt een vraag gesteld die je met Ja of Nee kunt beantwoorden. "Is de voorraad op?"

De Pijl (De Richting): Waar gaan we heen?

Een goede gewoonte in stroomdiagrammen is om ze simpel te houden. Het is aanlokkelijk om alles in één schema te tekenen, maar je zal merken dat volledige processen met al hun logica en uitzonderingen erg groot worden. Ook is het lastig om alles wat je later ontdekt er nog 'in te proppen'.

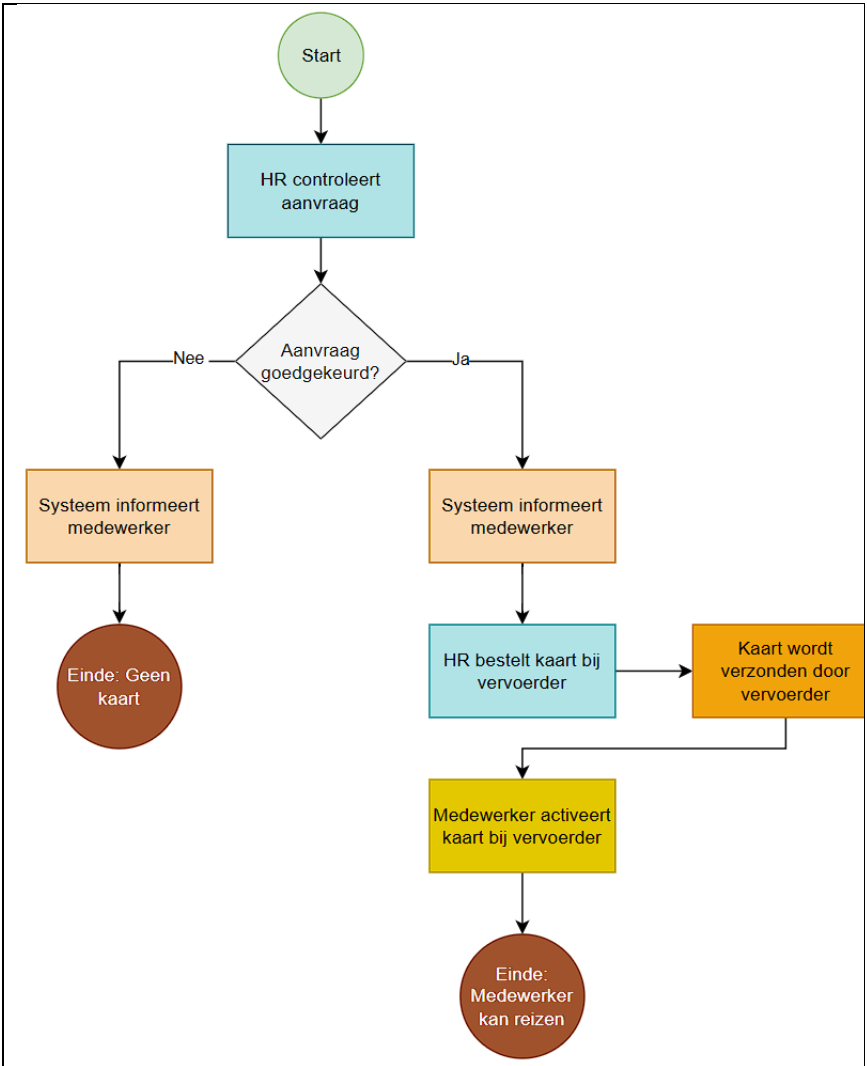
De oplossing hiervoor is **verdiepen**. Elke actie en beslissing is uit te tekenen in een apart stroomdiagram, waarin je de diepere logica in dat blokje uitwerkt. Zo zou je een nieuw stroomschema voor een actie 'goedkeuren' kunnen tekenen dat de logica achter de goedkeuring beschrijft terwijl je dit achterwege laat in je hoofdschema. Een voorbeeld hiervoor vind je in het voorbeeldscenario van dit hoofdstuk.

Voorbeeld: De OV-kaart goedkeuren

HR moet OV-kaart aanvragen voor medewerkers goedkeuren: Jij tekent het uit:

- Start: Aanvraag komt binnen
- Actie: Controleer aanvraag
- Beslissing (Ruit): Is de aanvraag goedgekeurd?
    - Nee → Informeer medewerker.
    - Ja → Informeer medewerker, bestellen, verzenden, activeren door medewerker.
- Einde: Geen kaart of wel een kaart.

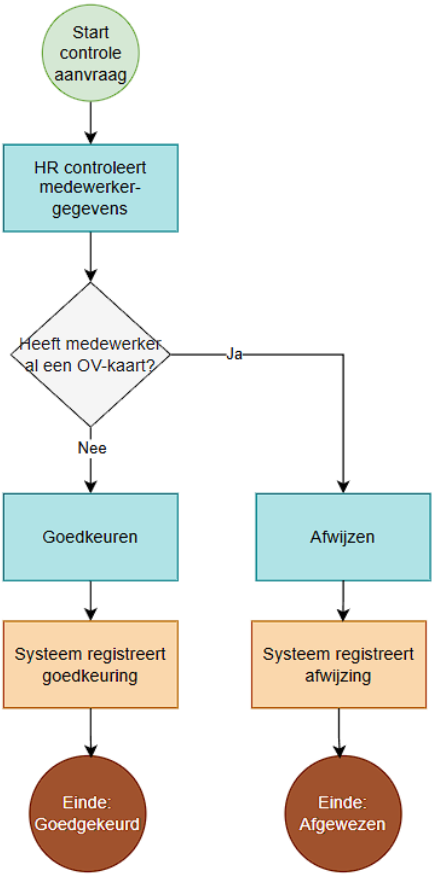
Op papier klinkt dit duidelijk en eenvoudig, maar als we het uittekenen dan verraadt het een misser die (waarschijnlijk) over het hoofd gezien gaat worden.



Zie je de misser in dit schema? We leggen nergens vast wat de uitslag is. De medewerker kan dus blijven aanvragen en HR blijft controleren.

Je eerste neiging is misschien om er blokjes bij te tekenen, maar doe dat niet, tenzij je zeker weet dat je tekening klein blijft. Je zal echter merken dat je tekening met alle mitsen en maren al snel te groot wordt.
De oplossing is verdiepen. Je maakt een aparte uitwerking voor het blokje 'Controleer aanvraag'. In dat sub-proces leg je vast dat de status ('Goedgekeurd' of 'Afgewezen') wordt opgeslagen.

Probleem opgelost, zonder dat je hoofdtekening een chaos wordt.



Stel jezelf bij het inzoomen steeds de vraag: heb ik nu alles afgedekt? Bovenstaand voorbeeld is expres simpel gehouden, maar de werkelijkheid zit vaak vol met mitsen en maren. Want geldt dit voor alle loonschalen? Zijn er verschillende abonnementen? Mag een medewerker tussentijds wisselen? Wat bij functiewijzigingen? Moet er niet geregistreerd worden per wanneer de kaart in gebruik is?

Juist in je diepere uitwerking moet je deze details vangen. Zo blijft je hoofdtekening leesbaar, maar is je analyse wel compleet.

Tips

- Stroomt het lekker? Een goed schema lees je van boven naar beneden of van links naar rechts. Ga niet kriskras over het papier.
- Ja/Nee is heilig: Zorg dat er uit een Beslissing (Ruit) altijd duidelijke lijnen komen. Meestal is Ja en Nee voldoende. Heb je meer opties? Gebruik meerdere ruitjes achter elkaar.
- Geen doodlopende wegen: Elk lijntje moet ergens uitkomen. Een proces stopt nooit zomaar in het luchtledige (tenzij het Einde is).
- Houd het overzichtelijk: Werk eerst het hoofdproces uit, en focus je daarna pas op de logica achter elk blokje.
- Werk met kleuren: In een stroomdiagram doen vaak meerdere mensen of systemen hun acties, door deze een eigen kleur te geven is snel duidelijk wie wat doet.

Niet doen

- Alles in één tekening verwerken": Probeer niet alle stappen in detail te verwerken in één tekening. Dit wordt snel te groot. Teken het hoofdproces, en werk daarna de subprocessen los uit.
- Kruisende lijnen: Probeer te voorkomen dat pijlen elkaar kruisen, dat wordt onleesbaar. Verplaats je blokjes tot de lijnen vrij lopen.
- Te veel tekst: Zet geen hele alinea's in een blokje. Gebruik werkwoorden: "Keur goed", "Verzend mail".

MoSCoW: Kiezen als alles belangrijk lijkt

In bijna elk project kom je hetzelfde probleem tegen: alles is belangrijk. De ene belanghebbende wil kunnen invoeren, de ander automatisch verwerken, de derde een gelikt rapport. Voor je het weet heb je een wensenlijst waar achter elke regel "Prioriteit 1" staat. En als alles prioriteit heeft, heeft niets prioriteit.

MoSCoW is een simpele, maar krachtige techniek om samen harde keuzes te maken. Je verdeelt alle wensen in vier categorieën:

- **M - Must have**: Zonder dit werkt de oplossing simpelweg niet. Als dit ontbreekt, faalt het project. (De wetgeving vereist het, of het proces stopt zonder deze stap).
- **S - Should have**: Heel belangrijk. Als dit ontbreekt is het pijnlijk of lastig, maar er is een (tijdelijke) omweg mogelijk.
- **C - Could have**: Leuk om te hebben ('Nice to have'). Dit doen we alleen als er tijd en budget overblijft.
- **W - Won't have** (for now): Dit doen we bewust niet in deze fase. Misschien later, maar nu niet.

Het lijkt een simpel modelletje, maar het effect is groot: je haalt de emotie uit de discussie door samen te bepalen wat echt nodig is en wat slechts 'wenselijk' is.

Hoe pak je dit aan?
MoSCoW werkt alleen als je het gesprek goed begeleidt. Het is geen invuloefening in Excel, maar een onderhandeling. Zo doe je dat:

1. **Leg de spelregels uit**: Wees streng. Een 'Must' is echt een must. Als we alles op Must zetten, leveren we waarschijnlijk niks op.
2. **Laat iedereen eerst zelf nadenken**: Voorkom dat de hardste roeper de richting bepaalt.

3. **Bespreek de verschillen**: Waarom vindt de één iets een Must en de ander een Could? Daar zit vaak waardevolle informatie.
4. **Check de balans**: Als 80% van de wensen een Must is, heb je geen prioriteiten gesteld. Dwing de groep om te kiezen.
5. **Maak het zichtbaar**: Gebruik een whiteboard of een digitale muur met post-its. Als je een kaartje fysiek van 'Must' naar 'Should' verplaatst, voelt iedereen de keuze.

Wanneer zet je dit in?

- Als er te veel wensen zijn voor de beschikbare tijd of budget.
- Bij de start van een project, om verwachtingen te managen.
- Als spanningen oplopen tussen belanghebbenden (“Mijn onderdeel is belangrijker!”).
- Tijdens het verfijnen van werkvoorraad, als dreigt dat alles tegelijk gedaan moet worden.

Voorbeeld: Het klantportaal

Een projectteam werkt aan een nieuw klantportaal voor inzicht in energieverbruik. De stakeholders komen met een waslijst aan wensen: inloggen, chatfunctie, meertaligheid, energiecontract inzage, energie-analyse en een dashboard.

In de eerste ronde roept iedereen bij alles: "Must!"

Jij begeleidt het gesprek met de kritische vraag: "Kan het product zonder dit onderdeel live?"

- "Als er géén energieoverzicht voor huidig verbruik is, kan de klant het portaal dan gebruiken?"

- Antwoord: "Nee, want daar komen de meeste klanten voor."
- Conclusie: -> Must have

- "Is het inzien van het energiecontract inzien doel van de portal?"
  - Antwoord: "Het doel is inzicht in energie. Maar als energieleverancier willen we het huidige lopende contract wel inzichtelijk maken in het portaal.
  - Conclusie: Doel is inzicht in energie, niet in contract. -> Should have
- "Zijn alle talen belangrijk?"
  - Antwoord: "Engels en Nederlands zijn noodzakelijk. Duits zou mooi zijn voor mensen in het grensgebied."
  - Conclusie: NL/EN -> Must, Duits -> Could of Won't

Je gaat zo alle wensen- en eisen af. Binnen een uur heb je de discussie omgebogen van "Ik wil dit" naar "Wat is noodzakelijk?".

Tips

- De lakmoesproef: Stel bij twijfel altijd de vraag: "Als we dit weglaten, werkt het dan helemaal niet meer?" Is het antwoord "Jawel, maar...", dan is het geen Must.
- Wees streng: Durf de discussie aan te gaan. Niet alles kan even belangrijk zijn.
- Vergeet de 'Won't' niet: Deze bak is namelijk ook belangrijk voor verwachtingsmanagement. Door iets expliciet in 'Won't' te zetten, zeg je: "We hebben je gehoord, maar we doen het nu niet." Dat is beter dan vaagheid.
- Herhaal het: Prioriteiten verschuiven. Wat vandaag een 'Could' is, kan volgende maand een 'Must' zijn door nieuwe wetgeving.

Niet doen

- Alles in Must stoppen: Dan kun je het model net zo goed niet gebruiken.
- Geen gezamenlijke discussie voeren: Als jij het in je eentje invult, voelt niemand zich eigenaar van de keuzes.
- Verwarren met planning: MoSCoW gaat over belang (prioriteit), niet over volgorde (planning). Soms doe je een 'Should have' eerder dan een 'Must have' omdat het sneller of een latere afhankelijk wegneemt.

Persona's: Omdat 'de gebruiker' niet bestaat

Systemen worden gebouwd voor mensen. Maar "de gebruiker" bestaat niet. Een administratief medewerker die de hele dag data invoert, werkt totaal anders dan een manager die één keer per maand een rapport draait. En een digibeet heeft andere hulp nodig dan een power user die alle sneltoetsen kent.

Toch hoor je in projecten vaak: "De gebruiker wil dit." Dat is gevaarlijk vaag. Want over welke gebruiker hebben we het?

Persona's helpen om die vaagheid te doorbreken. Je maakt een aantal concrete, fictieve (maar realistische) profielen die je belangrijkste doelgroepen vertegenwoordigen.

Let op: dit zijn geen marketing-karikaturen ("Linda, 34, houdt van cappuccino's, yoga en de Libelle"). Voor informatieanalyse is dat nutteloze info. Het gaat jou om een hulpmiddel om gedrag, doelen en frustraties scherp te krijgen.

Hoe pak je dit aan?
Een goede persona is gebaseerd op observaties uit de praktijk, niet op wat jij achter je bureau verzint. Je hoeft geen roman te schrijven; een half A4 per profiel is genoeg.

1. **Verzamel input**: Praat met verschillende gebruikers. Noteer hun doelen, hun dagelijkse taken, maar vooral hun frustraties en digitale vaardigheid.
1. **Clusteren**: Zoek patronen. Welke groepen lijken op elkaar in gedrag of behoefte? (Soms lijkt een manager in gedrag op een nieuwe medewerker omdat ze beiden het systeem nauwelijks kennen).
2. **Maak er profielen van**: Geef elke groep een naam en een kort profiel. Beschrijf hun context, hun doel en hun grootste pijnpunt.
3. **Gebruik beeld**: Plak er een foto of pictogram bij. Dat klinkt kinderachtig, maar het maakt een persona menselijk in discussies.
4. **Hou het relevant**: Schrijf alleen op wat relevant is voor het ontwerp of de prioritering.

Wanneer zet je dit in?

- Bij de start: Om scherp te krijgen voor wie je nu eigenlijk bezig bent.
- Tijdens werksessies: Zodra de discussie verzandt in "Ja, maar de gebruiker wil...", kap jij dat af met: "Welke gebruiker bedoel je?"
- Tegenwicht voor management: Als beslissingen alleen vanuit de directiekamer worden genomen, helpen persona's om de werkvloer weer in beeld te krijgen.
- Bij prioriteren: Een functie kan voor persona A noodzakelijk zijn, maar voor persona B ruis. Dat helpt bij het maken van keuzes.

Voorbeeld: Vernieuwen van het inventaris-systeem

Een organisatie ontwikkelt aan de vernieuwing van het bestaande inventaris-systeem. Iedereen praat over "de medewerker". Jij merkt dat dit te vaag is en maakt drie persona's:

1. Sanne: Is backoffice medewerker. Kort in dienst. Kent het programma niet (goed). Raakt snel de weg kwijt in het menu.
2. Kees: Is backoffice medewerker. Werkt hier al jaren. Zit dagelijks in het systeem. Kent alle codes uit zijn hoofd en ergert zich groen en geel aan elke extra muisklik of dubbele invoer.
3. Bob: Is manager. Ervaren met het systeem. Wil in één oogopslag zien hoe het met de voorraad gesteld is. Gebruikt het systeem weinig, maar als hij erin zit moet er direct overzicht zijn in de huidige voorraad en eventuele problemen.

Met deze persona's wordt de discussies direct concreet. Als je klant zegt: "De navigatie moet simpeler", vraag jij: "Bedoel je voor Sanne, die de weg niet weet? Of voor Kees, die gewoon snel wil kunnen werken?" Voor Sanne heb je uitleg nodig, voor Kees juist sneltoetsen.

Tips

Goed toegepaste persona's brengen gebruikers tot leven.

- Hou het kort: Een half A4 met bulletpoints is perfect.
- Maak het zichtbaar: Hang ze letterlijk aan de muur of zet ze op de eerste slide van je presentatie.
- Valideer ze: Laat echte gebruikers de persona's lezen. Herkennen ze zichzelf of hun collega's?
- Gebruik de namen: Verwijs in vergaderingen naar 'Sanne' en 'Kees', in plaats van 'de gebruiker'.
- Kies de juiste focus: Richt je op gedrag en doelen, niet op leeftijd of functietitel.

Niet doen
Persona's zijn alleen nuttig als je ze goed gebruikt. Trap niet in deze vallen:

- Marketingprofielen maken: Leeftijd, hobby's en favoriete tv-programma's zijn zelden relevant voor jouw oplossing.
- Te veel persona's: Meestal zijn 3 tot 5 duidelijke profielen genoeg om 90% van de gebruikers te dekken. En bij een heel eenvoudig project kan je ze ook overslaan.
- Verwarren met functies: Een persona is niet hetzelfde als een functietitel. 'De manager' kan digitaal heel onhandig zijn (gedrag), terwijl 'de stagiair' alles in twee seconden snapt. Cluster op dat gedrag, niet op de titel op het visitekaartje.

INVEST: De kwaliteitscheck voor bouwblokken

We hebben het in dit boek eerder gehad over bouwblokken, het klein maken van stukken uit te voeren werk. Niet elk bouwblok dat je bedenkt is meteen bruikbaar. Soms is het te groot, te vaag of hangt het te veel af van andere onderdelen.

Je wilt voorkomen dat je een vaag plan over de schutting gooit bij het uitvoerende team. Om te zorgen dat een bouwblok echt werkbaar is, gebruiken veel teams het INVEST-model.

INVEST is een handig ezelsbruggetje dat staat voor:

I - Independent (Onafhankelijk)

N - Negotiable (Bespreekbaar)

V - Valuable (Waardevol)

E - Estimable (Inschatbaar),

S - Small (Klein)

T - Testable (Toetsbaar)

Independent – Onafhankelijk
Een bouwblok moet bij voorkeur zoveel mogelijk op zichzelf staan. Als alles aan elkaar vastknoopt ("Ik kan pas beginnen als Pietje klaar is met stap 3"), loopt de planning vast.

- Het streven: Knip blokken zo dat ze zelfstandig waarde leveren of los gebouwd kunnen worden.
- De realiteit: Soms kan het niet anders (je kan meestal in een internetapplicatie niet werken zonder eerst in te loggen). Dat is niet erg, zolang je die afhankelijkheid maar expliciet benoemt.

Negotiable – Bespreekbaar
Een bouwblok is een startpunt voor een dialoog, geen dichtgetimmerd contract. Je beschrijft het wat en waarom, maar je laat ruimte voor het hoe.

- Het streven: Geef het team de vrijheid om de slimste technische oplossing te kiezen.
- De realiteit: Soms ligt het vast (door wetgeving of eerdere afspraken). Wees daar dan eerlijk over: "Dit is een harde eis, geen discussiepunt."

Valuable – Waardevol
Elk blok moet iets opleveren voor de gebruiker of de organisatie. Geen techniek om de techniek.

- Het streven: Maak altijd duidelijk wie er blij van wordt en waarom.
- De realiteit: Soms moet je iets technisch bouwen (zoals beveiliging). Vertaal dat dan naar waarde: "We bouwen dit om datalekken te voorkomen", in plaats van "We installeren een firewall".

Estimable – Inschatbaar
Het team moet kunnen zeggen: "Dit kost ons ongeveer drie dagen." Als ze zeggen: "Geen flauw idee", is je blok nog te vaag.

- Het streven: Concreet genoeg voor een realistische schatting.

- De realiteit: Soms is iets gloednieuw en complex. Dat mag, maar spreek dan af dat je eerst een klein onderzoekje doet om het scherper te krijgen.

Small – Klein genoeg
Hoe kleiner, hoe beter. Kleine blokjes geven snelheid, overzicht en sneller resultaat.

- Het streven: Blokken die in een paar dagen (of zelfs uren) af kunnen zijn.
- De realiteit: Sommige wensen zijn nu eenmaal groot. Probeer toch te knippen. Kan de eerste versie simpeler?

Testable – Toetsbaar
Wanneer is het 'af'? Je moet objectief kunnen vaststellen of het werkt.

- Het streven: Concrete criteria (Acceptatiecriteria). "Als ik dit doe, moet dat gebeuren."
- De realiteit: Niet alles is zwart-wit. Maak dan afspraken over wat 'goed genoeg' is, zodat je niet in eindeloze discussies belandt over wanneer iets goed genoeg werkt.

Een goed bouwblok hoeft niet op elk punt een 10 te scoren, maar hoe meer vinkjes je kunt zetten, hoe groter de kans dat je plan echt uitvoerbaar en waardevol is.

Hoe pak je dit aan?
INVEST is geen formulier dat je moet invullen. Zie het als een bril waardoor je naar je eigen werk kijkt.

1. **Schrijf eerst, check later**: Schrijf je bouwblok eerst gewoon uit. Kort en helder, alsof je het uitlegt aan een collega.
2. **Toets jezelf**: Lees je tekst terug en stel jezelf per letter de vraag. Is dit blok klein genoeg? Is het toetsbaar?
3. **Noteer twijfels**: Is het antwoord ergens 'nee' of 'misschien'? Schrijf erbij waarom.
4. **Verbeter direct**: Is het te groot? Knip het op. Is het vaag? Maak het concreet.
5. **Durf te schrappen**: Soms ontdek je dat een blok eigenlijk helemaal geen waarde toevoegt. Weg ermee.

Het resultaat: een bouwblok dat niet alleen jij snapt, maar waar het hele team mee aan de slag kan.

Wanneer zet je dit in?
- Tijdens je eigen analyse: Gebruik INVEST om je concepten te toetsen voordat je ze deelt.
- Tijdens werkoverleg: Als het team twijfelt, pak je INVEST erbij. "Jongens, is dit wel Small genoeg? Is dit Testable?"

Voorbeeld: Het profiel beheren

Een team bespreekt een bouwblok: “Als gebruiker van het internet portaal wil ik mijn profiel kunnen beheren.”

De INVEST-check:

- Independent - Nee, hangt samen met inloggen.
- Negotiable - Ja, ruimte voor discussie.
- Valuable - Ja, nuttig voor de gebruiker.
- Estimatable - Nee! Veel te vaag. Wat is 'beheren'?
- Small- Nee! Dit kan van alles zijn, dus waarschijnlijk groot.
- Testable - Nee! Wanneer is het af?

Er wordt besloten dat dit bouwblok teveel vaagheden bevat dus wordt het opgeknipt in drie kleine, heldere blokken:

1. Wachtwoord wijzigen.
2. Adresgegevens aanpassen.
3. Profielfoto uploaden.

Nu zijn ze wel Small, Estimable en Testable.

Tips

- Gebruik het als startpunt: Het moet een gesprek openen, geen discussie sluiten.
- Wees streng op ‘Small’: Te grote blokken zijn de nummer 1 oorzaak van vertraging.
- Maak ‘Negotiable concreet genoeg: Er moet wel een duidelijk doel zijn.

- Samen is beter: INVEST werkt het best als het hele team meedenkt.

Niet doen

- Afvinken: Het is geen checklist. Het gaat om het inzicht.
- Te streng zijn: 100% perfectie bestaat niet. Goed genoeg is goed genoeg.

Artificial Intelligence: Je assistent

Bewust had ik AI (Artificial Intelligence) tot nu toe niet genoemd in dit boek, omdat het belangrijk is dat jij begrijpt hoe informatieanalyse werkt en jij een aanpak leert, en niet alles uitbesteed aan AI.

Zie AI, zoals ChatGPT, Claude, Copilot en Gemini, niet als je vervanger maar als je assistent. Het is snel, heeft vrijwel overal (theoretische) kennis van en wordt nooit moe. Maar het heeft ook geen geweten, kent de context van jouw bedrijf niet en verzint soms dingen die gewoon niet waar zijn. Het is een fantastisch hulpmiddel, zolang jij de baas blijft.

Hoe AI je kan helpen
Een informatieanalyse traject starten kan lastig zijn, en ook met een aanpak zoals in dit boek beschreven is het vaak lastig om te bedenken waar je moet beginnen, welke aanpak het beste is of je wel volledig bent. Gebruik AI voor:

- **Brainstormen en voorbereiden**: Heb je morgen een interview met een klant over een nieuw logistiek proces? Vraag je AI-assistent om hulp:

  "Ik ben analist en ik moet een interview afnemen met een logistiek manager over het verbeteren van de voorraad. Geef me 10 kritische vragen die ik kan stellen om de knelpunten boven tafel te krijgen."

  Je hoeft ze niet alle 10 te gebruiken, maar er zitten er vast 3 tussen waar je zelf niet aan gedacht had.

- **Samenvatten en structureren**: Heb je een uur lang aantekeningen gemaakt waarbij iedereen door elkaar praatte?
  "Hier zijn mijn ruwe notities. Haal de 5 belangrijkste actiepunten eruit en vat de kern van het probleem samen in één alinea.". Jij was bij het gesprek en weet waar AI het juist heeft of juist de plank misslaat.

- **Concepten schrijven**: Moet je twintig bouwblokken uitwerken? Laat AI de eerste aanzet doen.
  "Schrijf een User Story voor een klant die zijn adres wil wijzigen, inclusief acceptatiecriteria en foutmeldingen."
  Het resultaat is zelden 100% perfect, maar je hebt binnen 10 seconden een basis die voor 80% staat. Jij doet de verfijning.

- **De advocaat van de duivel**: Gebruik AI om je eigen blinde vlekken te vinden.

  "Ik heb dit proces bedacht voor het aanvragen van verlof [plak proces]. Wat kan er misgaan? Verzin 5 scenario's waarbij dit proces vastloopt."

- **(Technische) vertaling**: Snap je een gebruikte terminologie van een software ontwikkelaar of je klant niet?

  "Leg uit wat een '…' is alsof ik 12 jaar ben, en gebruik een analogie van een restaurant."

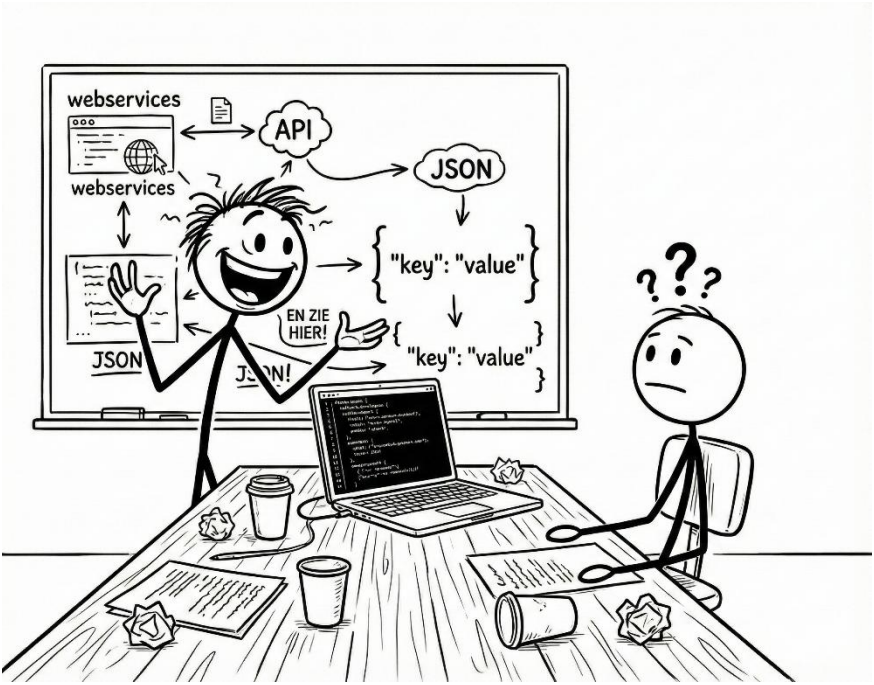
Waarom jij de baas moet blijven
Als AI zo slim is, waarom hebben we jou dan nog nodig? Omdat AI drie cruciale dingen mist die jij wel hebt.

1. Context en gevoel: AI weet niet dat Henk van de administratie vorige week boos is weggelopen uit een vergadering, dat er een politieke strijd woedt tussen Sales en Marketing of wat al eerder geprobeerd is. AI snapt nuance en sfeer niet. Jij wel. Jij weet wanneer een "ja" eigenlijk een "nee" betekent.

2. Waarheid: AI-modellen zijn getraind om een aannemelijk antwoord te geven, niet per se een waar antwoord. Dit noemen we hallucineren. Als de AI even het antwoord niet weet, verzint hij soms vol overtuiging een feit, een wetsartikel of een technische oplossing die niet bestaat. Neem nooit iets klakkeloos over. Jij bent verantwoordelijk voor de output, niet de chatbot.
3. Privacy en veiligheid: Let op met wat je invoert. Plak nooit, maar dan ook nooit, persoonsgegevens, wachtwoorden of bedrijfsgeheimen in een openbare AI-tool. Anonimiseer alles voordat je het vraagt.

Vroeger was de analist degene die alles uittypte. Met AI verandert je rol. Je wordt minder de schrijver en meer de regisseur. Jij bepaalt de richting, jij stelt de kritische vragen, jij checkt de kwaliteit en jij zorgt dat het landt bij de mensen.

Basiskennis techniek



Basiskennis techniek

Niet elk project is technisch. Niet elke analyse resulteert in code of datastromen. Maar als jij als analist werkt binnen een IT-omgeving, met ontwikkelaars en architecten, dan is dit hoofdstuk essentieel.

Stel je voor, jij en je team bespreken je analyse. Je vertelt enthousiast over een nieuwe functie die gegevens moet ophalen uit een ander systeem. Een developer fronst: "Weten we al of dat systeem een REST API heeft?"

Jij kijkt hem aan: "Een rest… wat?"

Een andere developer schuift ongemakkelijk op zijn stoel: "API. Tja, hoe vertel ik dat... Dat is, eh… zo'n koppelding."

Jij kan alleen maar bekennen: "Ja… nee… dat weet ik eigenlijk niet."

In IT-teams werk je niet voor de developers, maar met hen. Je zit in dezelfde meetings en zoekt samen naar oplossingen. Natuurlijk zijn zij de technisch specialisten, maar het helpt niet als je afhaakt zodra termen vallen als:

- "We missen de relatie tussen die twee tabellen."
- "Dit komt via een POST-request binnen."
- "Loggen we dit wel op acceptatie?"

Als je op dit soort momenten alleen maar wazig kunt kijken, wordt sparren lastig. Je stelt minder scherpe vragen, ziet risico's over het hoofd en snapt niet waarom iets de ene keer technisch onhaalbaar is en de andere keer een fluitje van een cent.

In dit hoofdstuk leg ik de termen uit die je dagelijks gaat horen. Niet om je te leren programmeren, maar om je te leren begrijpen.

API's: Hoe systemen met elkaar praten

Systemen praten continu met elkaar. Een webshop praat met het voorraadsysteem, de betalingsdienst en de transporteur. Maar welke taal spreken ze?

De taal: JSON en XML
Systemen praten niet in volzinnen, maar in gestructureerde berichtjes.
JSON (JavaScript Object Notation) is tegenwoordig de norm. Het is lichtgewicht en leesbaar. Een berichtje ziet er zo uit:

JSON

```
{
  "klantnummer": 123,
  "naam": "Jan Jansen",
  "actief": true,
  "bestellingen": ["AB123", "CD456"]
}
```

Onthoud vooral: het werkt met sleutels en de daarbij horende waarde.

Oudere systemen (vaak bij overheid of banken) gebruiken nog XML. Hoewel dit in moderne systemen niet vaak meer gebruikt wordt, is het handig om het verschil te herkennen.

XML

```
<klant>
  <klantnummer>123</klantnummer>
  <naam>Jan Jansen</naam>
  <actief>true</actief>
</klant>
```

Het loket: API's
De meeste systemen wisselen die JSON-berichtjes uit via een API. Zie een API als een loket. Je stopt er een verzoek in (Request) en krijgt een antwoord terug (Response).

Je gebruikt hierin vier methoden:

- GET: Gegevens opvragen (Mag ik de klantgegevens?)
- POST: Iets nieuws maken (Maak deze nieuwe klant aan)
- PUT: Iets wijzigen (Pas het adres van deze klant aan)
- DELETE: Iets verwijderen (Verwijder deze klant)

Statuscodes: Ging het goed?
Bij elk antwoord geeft de API een code terug. Als analist is het handig om de belangrijkste te herkennen, zodat je snapt wat er misgaat:

- 200 OK: Alles gelukt.
- 400 Bad Request: Je hebt foute data gestuurd (bijv. een datum in het verleden).
- 401/403 Forbidden: Je mag hier niet in (geen rechten).
- 404 Not Found: Hetgeen je zoekt bestaat niet.
- 500 Server Error: Het systeem is stuk (niet jouw fout).

De oude manier: SOAP
Naast de moderne API die hierboven beschreven is kom je bij grote instanties soms nog SOAP tegen. Dit is een oudere, formele standaard die veel strikter is. Het werkt met het oudere XML en strenge regels. Voor jou als analist betekent SOAP vaak: programmeurs hebben meer afstemming en minder flexibiliteit dan bij REST.

Waarom moet jij dit weten?
Jij bepaalt in je analyse welke gegevens nodig zijn. Als jij zegt "we hebben de geboortedatum nodig", moet die datum dus wel in zo'n JSON-berichtje zitten. Ontbreekt het daar? Dan werkt je hele functie niet.

Databases: Het geheugen van het systeem

Zonder database onthoudt een systeem niets. Geen klanten, geen bestellingen, geen historie. De meeste databases lijken in de basis enorm op Excel.

- Tabellen: Elk onderwerp heeft een eigen tabblad (bijv. 'Klanten', 'Producten').
- Kolommen: De eigenschappen (Naam, Adres, Prijs). Als je een kolom niet hebt, kun je die informatie nergens kwijt.
- Rijen: De daadwerkelijke items (Klant Jan Jansen is één rij).
- Relaties: De kracht zit in de koppeling. In de tabel 'Bestellingen' staat niet de hele naam van de klant, maar alleen het Klantnummer (bijv. 123). Daarmee verwijst hij naar de tabel 'Klanten'.

SQL: De vraagtaal
Ontwikkelaars gebruiken meestal SQL (Structured Query Language) om tegen de database te praten.

```
SELECT naam FROM klanten WHERE woonplaats = “Utrecht”
```

Waarom moet jij dit weten?
Zodat je weet wat er kan. Als jij vraagt: "Ik wil een overzicht van alle klanten die in 2023 rode schoenen kochten", moet je weten of die relatie wel te leggen is. Als de kleur niet los in een kolom van een tabel staat, is je wens technisch moeilijk uit te voeren zonder de database aan te passen.

Omgevingen (OTAP)

Je hoort het vaak als je met ontwikkelaars praat: "Staat het al op ACC?" of "In PROD werkt het wel."

Developers werken niet op één computer, maar in tal van omgevingen (vaak 'de OTAP-straat' genoemd).

- O - Ontwikkeling (DEV): De werkplaats. Hier wordt gebouwd. Het is rommelig, dingen zijn half af en de data is nep.
- T - Test (TEST): Hier wordt gecheckt of de techniek werkt.
- A - Acceptatie (ACC): De generale repetitie. Hier checken gebruikers (en jij!) of de oplossing klopt. Deze omgeving lijkt het meest op de werkelijkheid.
- P - Productie (PROD): Het echte systeem. Hier werken de echte klanten met echte data. Hier mag niets fout gaan.

Deze 'straat' kan ook DTAP genoemd worden, waar de D voor development staat.

Waarom moet jij dit weten?
Stel, iemand zegt: "Ik mis een knop." Jouw eerste vraag moet zijn: "In welke omgeving kijk je?" Misschien is de knop in Productie nog niet zichtbaar, maar in Acceptatie wel. Of misschien is de data in de Test-omgeving zo vervuild dat het lijkt alsof iets stuk is. Weet waar je naar kijkt.

Logging: De zwarte doos

Als een systeem raar doet, roept een developer: "Ik kijk wel even in de logs." Logs zijn als de zwarte doos in een vliegtuig. Alles wat het systeem doet (en elke fout die optreedt), wordt weggeschreven in tekstbestanden.

Waarom moet jij dit weten?
Je hoeft logs niet zelf te lezen, maar je moet weten dat ze bestaan.

- Is er een incident? Vraag: "Wat zeggen de logs?" (Vaak beter dan gissen).
- Privacy en veiligheid: Als analist moet jij scherp zijn. Worden er BSN-nummers of wachtwoorden in de logs geschreven? Dat mag absoluut niet wegens privacyregels. Daarnaast worden logbestanden vaak onderling door ontwikkelaars gedeeld voor foutanalyse, wat buiten de bestaande veiligheidssystemen om gaat, als er persoonlijke gegevens in staan is er een groter risico dat deze in verkeerde handen vallen. Wees kritisch op de inhoud van logs.

Frontend, Backend en microservices

Software bestaat uit lagen.

1. Front-end: Wat de gebruiker ziet (de app, de website).
2. Back-end (of Back-end for Frontend): De hersenen. Hier worden berekeningen gemaakt en data opgehaald.
3. Microservices: Kleine zelfstandige stukjes software die los van elkaar werken maar elkaar wel nodig hebben om samen één systeem te zijn.

Vroeger was software één groot blok (een monoliet). Daarna kregen we een losse front-end en een losse back-end. Tegenwoordig knippen we software meestal op in kleine, zelfstandige stukjes: de microservices.

En dan heb je ook nog...

Om je woordenschat compleet te maken, hier nog vier termen die je gaat horen:

- Authenticatie: Wie ben je?
- Autorisatie: Wat mag je? (Rollen en rechten). Als iemand een "403-melding" krijgt, weet jij nu: dat is een rechtenkwestie.
- Caching: Het tijdelijk opslaan van gegevens zodat het systeem sneller is. Zie je een oude waarde terwijl je net iets gewijzigd hebt? Grote kans dat je naar de cache kijkt.
- Events: Systemen die reageren op een signaal. "Als er een bestelling is (event), stuur dan een mail." De mail wordt dus niet direct verstuurd door de bestelknop, maar reageert op het event. Als jij die mail dus niet ontvangt, kan het best aan een niet-werkend event liggen.
- Versiebeheer: Team A gebruikt versie 1.2, Team B zit al op versie 1.8. Hier kunnen verschillen in zitten waardoor functies en data anders zijn dan verwacht of simpelweg nog missen. Als analist check je: “praten we over dezelfde versie?”

Je bent geen ontwikkelaar dus je hoeft echt niet alles tot in detail te kennen, maar je bent wel de brug tussen wens en uitvoer. Met deze basiskennis kun je kritisch doorvragen, begrijp je waarom dingen soms langer duren en ben je een volwaardig gesprekspartner voor je team.

Nog wat laatste tips



Nog wat laatste tips

Je bent er bijna doorheen. Je hebt nu de basis, het proces en de technieken geleerd. Theoretisch staat niets je meer in de weg om een topanalyse te leveren. Maar zoals we allemaal weten, de theorie en de praktijk zijn helaas niet altijd gelijk.

Daarom wil ik dit boek afsluiten met drie onderwerpen die in bijna elk project voor gedoe zorgen als je er niet op let.

De vergeten eisen (Het werkt wel, maar...)

Als analist focus je snel op de stappen van het proces: "Eerst vullen we het formulier in, dan keuren we het goed." Maar daarmee ben je er vaak nog niet. Weet je wel onder welke voorwaarden dit proces moet draaien?

Een proces kan op papier kloppen, maar in de praktijk volledig vastlopen omdat het niet bestand is tegen de werkelijkheid. Stel jezelf daarom deze vragen over de kwaliteit:

- Tijd en geduld: Hoe lang mag iets duren? Mag een klant een week wachten op een antwoord, of moet dat binnen een uur? (Een proces dat werkt, maar te lang duurt, is alsnog een slecht proces).
- Volume en drukte: Werkt jouw oplossing ook op een piekmoment? Een werkwijze waarbij je even iets handmatig opzoekt werkt prima bij drie klanten per dag, maar zorgt voor chaos bij dertig klanten per uur.
- Toegang en vertrouwelijkheid: Wie mag wat zien? Mag de stagiair alles zien? Heeft een medewerker dezelfde toegang als de manager?
- Foutgevoeligheid: Wat als iemand een fout maakt? Is je oplossing zo ingericht dat een fout (per ongeluk) onherstelbare schade aanricht, of is er een vangnet?

Als je deze randvoorwaarden niet bespreekt, wordt er vaak gekozen voor de makkelijkste weg. En dat is in de praktijk vaak de route die later voor vertraging, lekken of chaos zorgt.

De kunst van 'Nee' zeggen

Als analist ben jij de bewaker van de grens. Klanten willen vaak gouden bergen: "Alles moet kunnen, het moet morgen af en het mag niks kosten." Aan jou de ondankbare taak om soms op de rem te trappen. Hoe zeg je 'nee' zonder de relatie te verpesten?

- Zeg geen 'Nee', maar 'Ja, mits…': Als iemand een enorme nieuwe functie wil, zeg dan: "Ja, dat kunnen we bouwen, mits we iets anders laten vallen, want de tijd zit vol." Zo leg je de keuze terug bij de klant.
- Verwijs naar het doel: "Ik snap dat je dit wilt, maar we hadden afgesproken dat we eerst het planningsprobleem zouden oplossen. Dit leidt daarvan af."
- Wees de expert: Soms vraagt een klant iets dat technisch onhandig of onveilig is. Wees dan eerlijk: "Ik adviseer om dit niet te doen, want het maakt het systeem traag/onveilig. Mijn voorstel is..."

Onthoud: een analist die overal 'ja' op zegt, levert uiteindelijk een project op dat nooit afkomt of rammelt aan alle kanten. 'Nee' zeggen is ook een vorm van kwaliteit leveren.

Waar laat je de waarheid?

Je hebt prachtige analyses gemaakt, processen getekend en eisen uitgewerkt. Maar waar sla je dat op? Te vaak zie ik projecten waar belangrijke info in de hoofden van mensen zit, of verspreid staat over 20 mailtjes en drie versies van een Word-document genaamd Definitief_v3_EchtDefinitief.docx.

Dat is vragen om problemen. Want een uitvoerder kijkt in versie 2, de tester in versie 3 en jij hebt versie 4 in je hoofd.

Zorg voor één bron van waarheid. Het maakt niet uit waar dit is, in Word, Excel, een tool als Confluence of Sharepoint, zolang het maar aan twee regels voldoet:

- Het staat centraal: Iedereen kijkt naar dezelfde, live versie. Mail nooit documenten rond als bijlage.
- Als het er niet staat, bestaat het niet: Afspraak bij de koffieautomaat gemaakt? Prima, maar het geldt pas als het in je centrale waarheid staat.

Dwing jezelf en je team om die bron heilig te verklaren. Dat scheelt je later eindeloze discussies over "wat hadden we ook alweer afgesproken?".

Het is nu aan jou

Je hebt het boek uitgelezen. Je weet nu dat informatieanalyse geen kwestie is van dikke rapporten schrijven, maar van de juiste vragen stellen. Je hebt je gereedschapskist gevuld met een gefaseerde aanpak, gespreks-technieken en technieken om je werk te stuctureren. Je weet hoe je AI voor je karretje spant en hoe je een ontwikkelaar begrijpt die het over API's heeft.

Nu begint het echte werk pas. Informatieanalyse leer je namelijk niet uit een boek, maar door het te doen.

Begin klein
Misschien duizelt het je een beetje. "Moet ik nu bij elk project persona's maken, een klantreis tekenen en 5 keer waarom vragen?"
Nee. Absoluut niet.

Zie dit boek als een gereedschapskist waar je uit kan halen wat je op dat moment nodig hebt. Je pakt het gereedschap wat je op dat moment nodig hebt.

- Is het doel vaag? Teken een proces.
- Zijn er te veel wensen? Gebruik MoSCoW.
- Is de sfeer stroef? Vraag naar het verleden.
- Heb je geen inspiratie? Vraag het aan AI.

Je hoeft niet alles te gebruiken, en zeker niet allemaal tegelijk.

Durf de 'domste' te zijn
Als ik je één laatste advies mag geven, is het dit: wees niet bang om de 'domme' vragen te stellen.

In vergaderingen houden mensen vaak hun mond omdat ze bang zijn dat ze iets horen te weten. Ze knikken ja, maar denken nee. Jij bent degene die die stilte doorbreekt. Jij bent degene die durft te zeggen: "Sorry, maar ik snap het nog niet helemaal. Kun je dat tekenen?"

Dat voelt in het begin ongemakkelijk. Maar je zult zien: zodra jij die vraag stelt, slaken drie andere mensen een zucht van verlichting.

Je bent geen notulist
Onthoud waar we dit boek mee begonnen. Je zit er niet om op te schrijven wat de klant zegt, maar om te ontdekken wat de klant nodig heeft. Dat verschil maakt jou onmisbaar.

Of je nu 'Informatieanalist' op je visitekaartje hebt staan, of dat je projectleider, ontwikkelaar of officemanager bent: zodra jij stopt met aannemen en begint met vragen, ben je aan het analyseren.

Dus: boek dicht, stift pakken en vragen maar.

Veel succes!

Bronnen & leestips

De technieken en methoden die ik in dit boek gebruik zijn niet allemaal door mij bedacht. In veel gevallen gaat het om bestaande technieken waar ik een eigen, eenvoudigere, draai aan gegeven heb.

Voor wie de diepte in wil gaan of de officiële documentatie wil nalezen, heb ik hieronder de belangrijkste bronnen en mijn persoonlijke leestips verzameld.

De basis

- Agile Manifesto: De basis van het agile werken. In 2001 opgesteld door zeventien software-pioniers. De tekst is nog steeds verrassend actueel en vormt de kern van hoe we tegenwoordig naar waarde-levering kijken.
    - *Bron: agilemanifesto.org*

- De Scrum Guide: De 'bijbel' van Scrum, geschreven door de grondleggers Jeff Sutherland en Ken Schwaber. Hierin staan de officiële spelregels van het raamwerk beschreven.
    - *Bron: scrumguides.org*

- Scrum.org: Waar de Scrum Guide de regels geeft, is Scrum.org de plek voor de professionele invulling. Opgericht door Ken Schwaber, biedt dit platform artikelen, video's en blogs over de praktijk. Dit is ook dé plek voor wie een officieel certificaat (zoals PSPO of PSM) wil behalen.
    - *Bron: scrum.org*

- De 64%-regel: Dit cijfer komt uit het bekende Chaos Report van de Standish Group. Hun jarenlange onderzoek naar IT-projecten laat zien dat gemiddeld 64% van alle functies in maatwerksoftware zelden of nooit wordt gebruikt. Voor een analist is dit dé bron om de ‘waarom-vraag’ te blijven stellen en verspilling tegen te gaan.

Gesprekstechnieken

- Gedragsgericht doorvragen: De techniek om naar specifiek gedrag uit het verleden te vragen is gebaseerd op de STAR-methode (Situatie, Taak, Actie, Resultaat). Het is de beste manier om aannames te vervangen door feiten.

- 5x Waarom (5 Whys): Deze techniek komt uit de koker van Sakichi Toyoda (de oprichter van Toyota). Het is de simpelste manier om van een oppervlakkig symptoom naar de werkelijke wortel van een probleem te graven.

- Focus op emoties: Het begrijpen van de frustratie of hoop van een gebruiker is een kernonderdeel van Design Thinking. De Empathy Map, ontwikkeld door Dave Gray (XPLANE), helpt je systematisch in kaart te brengen wat een gebruiker echt ervaart.

- Toekomstbeeld schetsen en vragen naar eerdere oplossingen: Deze technieken komen voort uit de psychologie van Steve de Shazer en Insoo Kim Berg. Het dwingt de gesprekspartner om uit de probleem-modus te stappen.

- Stel, het is mislukt: Deze techniek, waarbij je vooraf doet alsof een project al geflopt is om zo de risico's te achterhalen, is bedacht door psycholoog Gary Klein. Het is dé manier om de 'olifant in de kamer' bespreekbaar te maken.

Technieken die je werk structureren

- Stroomdiagrammen (Flowcharts): De basis voor het visueel maken van processen werd al in 1921 gelegd door Frank en Lillian Gilbreth. In de moderne IT-wereld is dit verder geprofessionaliseerd in de BPMN-standaard (Business Process Model and Notation). Het helpt je om de 'logische weg' van informatie te volgen zonder te verdwalen.

- MoSCoW-methode: Deze manier van prioriteren is ontwikkeld door Dai Clegg tijdens zijn werk bij Oracle UK. Het is inmiddels de standaard om keuzes te maken die door iedereen begrepen worden.

- Persona's: Het concept om fictieve gebruikers te creëren om mensgericht te ontwerpen, werd gepopulariseerd door Alan Cooper. Het helpt je om te stoppen met denken in 'de gebruiker' en te gaan denken in 'vlees en bloed'.

- INVEST-criteria: Bill Wake bedacht dit acroniem om de kwaliteit van User Stories (onze bouwblokken) te toetsen. Als een brok informatie aan deze zes letters voldoet, weet je dat het klaar is voor de uitvoerders.

Mijn persoonlijke leestips

Als je na dit boek de smaak te pakken hebt, zijn dit de boeken die op mijn eigen nachtkastje hebben gelegen:

- "**The Back of the Napkin**" van Dan Roam. Omdat een tekening meer zegt dan duizend vergaderminuten.

- "**User Story Mapping**" van Jeff Patton. Hét boek als je wilt leren hoe je een grote berg werk visueel maakt zonder het overzicht te verliezen.

- "**Start with Why**" van Simon Sinek. Hoewel het een managementboek is, is de kern essentieel voor elke analist: begin altijd bij de bedoeling, pas daarna bij de uitvoering.

- "**Scrum: The Art of Doing Twice the Work in Half the Time**" van Jeff Sutherland. Een vlot geschreven boek dat de gedachte achter kort-cyclisch werken uitlegt.

OVER DE AUTEUR



Michiel Lensink is geen theoreticus die vanaf de zijlijn toekijkt, maar een man van de praktijk. Sinds begin 2000 loopt hij rond in de wereld van ICT en softwareontwikkeling.

Hij begon zijn carrière als programmeur, maar realiseerde zich al snel dat succes niet alleen afhangt van goede code. Tegenwoordig vervult hij daarom diverse rollen: van Product Owner en Informatieanalist tot Scrum Master en Agile Coach.

Die breedte is een bewuste keuze. Michiel laat zich niet graag in een hokje duwen, maar kiest de rol die op dat moment nodig is, vanuit de sterke overtuiging dat hij daar het verschil kan maken.

Daarbij heeft hij een speciale interesse ontwikkeld voor de menselijke kant van softwareontwikkeling. Want processen en techniek zijn belangrijk, maar uiteindelijk zijn het de mensen die het werk moeten doen.

Met zijn bedrijf Consultancy En Zo slaat hij de brug tussen 'de business' en 'de IT', altijd met een nuchtere blik en een hekel aan onnodig papierwerk.

Wil je samen eens nadenken over jouw project of Michiel inhuren? Kijk op www.consultancyenzo.nl of stuur een bericht via LinkedIn.

Leuk dat je mijn boek hebt gelezen. Ik hoop dat het je waardevolle inzichten heeft opgeleverd voor de praktijk.

Heeft het boek je verder geholpen? Dan wordt een vrijwillige bijdrage enorm gewaardeerd. Dat kan eenvoudig via de QR-code hieronder.



Of koop een fysiek boek via www.consultancyenzo.nl