

Format voor artikel en onderzoeksrapport (t.b.v. ingenieurs)

In een artikel (of een (white) paper, een application note, een case study, een professionele blog-post of een degelijke tutorial) kan een lezer elk van de genummerde punten die hieronder zijn uitgewerkt op zijn minst terugvinden of herleiden. In een onderzoeksverslag dien je de genummerde punten over te nemen als hoofdstuknummers en -namen. Uitzondering hierop is een keuze voor een horizontale ordening van de uitwerkingen van de deelvragen (elk hoofdstuk behandelt telkens methoden - resultaten - conclusies per deelvraag).

1. Inleiding

Een inleiding bestaat in elk geval uit de volgende onderdelen, of deze nu expliciet worden benoemd of op organische, meer impliciete wijze in de opbouw van de inleiding worden verwerkt:

- **Onderwerp**

Het gaat hier om context en thema. Voor welke maatschappij, omgeving, sector, regio, sociale groep, organisatie of individu is er een probleem? In welke omstandigheden? Wat zijn de oorzaken, wat is de directe aanleiding en wat zijn de gevolgen (waarom is het een probleem)? Beschrijf de **functionele** kant van het probleem.

- **Probleem (of vraagstuk)**

Hier staat de afdaling naar het technisch domein centraal: isoleer uit het functionele probleem het **technische** (TI-)probleem of -vraagstuk en omschrijf dat. Je mag hier ook beschrijven welke *systemen* ‘last’ hebben van het probleem. *Geef duidelijk aan welk deel van dit probleem JIJ gaat oplossen!*

- **Relevantie**

Wie (functioneel) of wat (technisch) heeft er direct baat bij een **technische** oplossing van het probleem (d.w.z. wat is de impact van *jouw* bijdrage, d.i. de impact van de oplossing van het *technische* probleem)?

- **Doel**

Hier formuleer je de(/het) hoofdvraag(stuk) of je centrale probleemstelling in één zin. Je zult merken dat sommige ingenieurs in het werkveld niet de moeite nemen om in artikelen een onderzoeksvraag te articuleren. Toch is er in die gevallen wel op zijn minst een centrale probleemstelling of een ‘vraagstuk’, vaak in één zin, aan te wijzen. In grotere onderzoeksrapporten is de hoofdvraag meestal een samengestelde vraag, die voor de beantwoording rekent op het uitvoeren van meerdere deelonderzoeksstappen.

- **Methode**

Hier kijk je kort vooruit op je aanpak (zonder in de details te treden).

- **Overzicht**

Hier doe je een voorbespreking van de rode draad per hoofdstuk (of per sectie). Deze laatste twee punten (Methode en Overzicht) moet je zien als verwachtingsmanagement van je lezer.

2. Probleemstelling

Je probleemstelling is een gedetailleerde uitwerking van je centrale (TI-)vraagstuk, waarin je alle relevante TI-aspecten vooraf de revue laat passeren. Het technische probleem (en **niet** het functionele) staat daarom centraal in je probleemstelling. Het is hierbij bovendien heel belangrijk dat je laat zien in welke *probleemklasse* het informaticagerelateerde (ook wel: *het computationele*) deel van jouw vraagstuk valt.

Een voorbeeld: je *functionele probleem* is dat een mestschuifrobot alle hoekjes van een stal moet schoonvegen, waarvan jij de plattegrond als input aangeleverd krijgt. Het *technische deelprobleem* waar *jij* aan gaat werken is het geven van een garantie dat de robot alle ‘vakjes’ van een plattegrond heeft schoongeveegd. Dit probleem bestaat uit elektrotechnische / mechanische aspecten (transponders, het opstellen van een plattegrond, keuze van de rekenhardware, etc.), maar er is ook een duidelijke computationele component: het vinden van een zo efficiënt mogelijk *algemeen* (niet alle stallen zijn hetzelfde) algoritme dat je in staat stelt te garanderen dat je alle ‘vakjes’ van een plattegrond zult bezoeken met de robot.

In dit voorbeeld valt het informaticagerelateerde deel van je technische probleem in de klasse ‘optimalisatieproblemen’, subklasse ‘zoekproblemen’, subsubklasse ‘padzoekproblemen’ en heel specifiek daarbinnen de categorie ‘Complete Coverage Path Planning’. Omdat deze problemen heel erg veel lijken op een andere subklasse van padzoekproblemen (de zogenaamde kortste-pad-algoritmen), moet je het contrast *heel* duidelijk maken. Deze diagnose maakt zowel *het schrijven van een vooronderzoek* (theoretisch kader) als *het componeren van een oplossing ongelooflijk veel eenvoudiger*.

De meest voorkomende hoofdklassen van informaticaproblemen (die gelijk iets zeggen over de **computationele moeilijkheidsgraad**) zijn decision problems, function problems en optimization problems.

Ons voorbeeld betreft een optimalisatieprobleem: wat is de ‘beste’ route, zodanig dat je het gehele oppervlak hebt afgedekt, terwijl de afgelegde afstand minimaal blijft. Het komt regelmatig voor dat je meerdere computationele deelproblemen moet herkennen in je opdracht. Je geeft van allemaal aan tot welke klasse ze behoren *in deze probleemstelling*.

Denk erom: als je een *computationeel probleem* niet kunt classificeren kan één van twee zaken aan de hand zijn in de ogen van je *peers*:

- a) Je hebt onvoldoende kennis van zaken en kunt een bekend beestje nog geen naam geven;
- b) Je bent op zo’n radicaal uniek probleem gestuit dat er fundamenteel wetenschappelijk onderzoek nodig is om tot een oplossing te komen (de oplossing voor het probleem is nieuw voor de gehele mensheid).

Beide gevallen zijn voor ingenieurs onwenselijk: in het eerste geval sla je een flater en in het tweede geval ben je niet bezig met werk dat jij als ingenieur zou

moeten doen. Noch ben je gekwalificeerd om zo'n studie uit te voeren. Dat is het werk van wetenschappelijk onderzoekers. Kortom: computationele problemen **MOET je ALTIJD** kunnen identificeren als een variant van een **BESTAAND** en **BEKEND** probleem.

Er zijn ook gevallen waarin je in een deel van je onderzoek **geen** computationeel probleem oplost. Bijvoorbeeld als je in je onderzoek een printplaat moet ontwerpen, of een softwarepakket moet selecteren volgens bepaalde eisen.

Niet-computationele technische problemen kun je nog steeds categoriseren, alleen niet meer volgens de computationele indeling (decision problems, function problems of optimization problems). Het proces (en de analyse die hoort bij) het kiezen van bijvoorbeeld een beste kandidaat wordt in het werkveld **pakketselectie** genoemd. Er is een geijkte aanpak voor dit soort problemen, dus het probleem in een vroeg stadium als zodanig herkennen is *essentieel*. Ook voor zaken als het ontwerpen van printplaten bestaat in het beroepenveld een 'koninklijke aanpak', die veelal raakvlakken heeft met industriestandaarden. In het geval van printplaten bijvoorbeeld IEC-60617 (zoek maar eens op!). Het proces dat moet leiden tot een ontwerp, zou je bijvoorbeeld kunnen categoriseren als 'watervalmethode' of 'scrum-aanpak'.

Je moet in de probleemstelling vanuit de wetenschappelijke literatuur duidelijk maken hoe je tot je diagnose / categorisering bent gekomen. Op die manier sluit jij met de manier waarop je over je probleem praat aan op het beroepenveld. Medingenieurs weten nu waar je het over hebt. Vragen die je dus voor jezelf moet beantwoorden zijn: Aan welke bronnen heb je je classificatie te danken? Welke bronnen helpen je in te zien wat voor werk er op dit vlak al gedaan is? Welke oplossingsrichtingen zijn voor jouw situatie bruikbaar en welke niet (en waarom wel/niet)? Wat word je beoogde oplossingsrichting? Welk vooronderzoek heeft de opdrachtgever al gedaan m.b.t. jouw oplossingsrichting? Welke standaarden zul je gaan gebruiken om je oplossing vorm te geven en te beschrijven?

Deze vragen zul je in de probleemstelling allemaal moeten beantwoorden. Je kunt dat het beste indelen volgens het volgende format:

- Theoretisch kader (ook wel: vooronderzoek genoemd)
 - Beschrijf de componenten van je technische probleem kort (die zal je helpen om later de deelvragen op te stellen).
 - **[ESSENTIEEL]** Benoem de probleemklasse waarin het computationele deel van jouw technische probleem valt, als dat deel van je onderzoek zal zijn.
 - Vul het **kennisgat** tussen wat een Technisch Informaticus (Computer Engineer, ook wel CE) hoort te weten en wat er nodig is om je onderzoek te kunnen begrijpen.
 - Bespreek bekende **oplossingsrichtingen** waarvan je al weet dat jij ze *niet* zult inslaan, benoem daarbij de nadelen van de betreffende oplossing.

- Bespreek kort de oplossingsrichting die jij *wel* verwacht in te slaan, om het contrast aan te geven met andere bekende oplossingsrichtingen. Denk erom: ‘bestaande oplossingsrichtingen’ bevatten referenties naar eerder uitgevoerd (toegepast) wetenschappelijk werk!
- Introduceer een eigen (of leun op een bestaande) **vocabulaire** (incl. definities) om je probleemdomain zo eenduidig en effectief mogelijk te beschrijven (parametriseren).
- Bespreek eventueel **eerder gedaan onderzoek** van directe collega’s, die al gewerkt hebben aan hetzelfde *functionele probleem*, of voorwerk hebben gedaan voor jouw onderzoek. Valideren hun werk als input voor jouw aankomende onderzoek.
- Opmerking: je theoretisch kader / vooronderzoek gebruikt de onderzoeksmethode ‘literatuurstudie’. Dat betekent niet dat je bij latere deelvragen geen literatuurstudie meer zult uitvoeren. De literatuurstudie die je later nog zult verrichten, zal echter ALTIJD onderdeel zijn van een analyse die nodig is om bijvoorbeeld een gemaakt besluit te beargumenteren, een gebruikte methode te verantwoorden, etc.
- Vraagstelling

Als je een kort artikel schrijft met maar één enkelvoudig onderzoeksvraagstuk, en maar één uit te voeren deelonderzoek, dan hoef je geen deelvragen op te stellen. Als je een onderzoek uitvoert met een vraagstuk dat meerdere onderzoekscomponenten heeft, dan werk je in dit deel van je probleemstelling je centrale vraagstuk uit in *deelvragen* / *deelproblemen*.

Een onderzoeksverslag met deelvragen is niets anders dan een geordende verzameling van samenhangende en vaak onderling afhankelijke artikelen. Tezamen beantwoorden de onderzoeken in deze artikelen een hoofdvraag. Het is een uitstekend idee om schematisch weer te geven op welke wijze je deelvragen zich tot elkaar verhouden (volgordelijkheid, afhankelijkheid, terugkoppeling/iteratie, gelijktijdigheid, etc. kun je allemaal kwijt in dit schema).

Je deelvragen moeten aan de volgende voorwaarden voldoen:

- Elke deelvraag moet resulteren in een onderzoeksproduct. Een onderzoeksproduct is niet hetzelfde als een beroepsproduct (bijv. een printplaat of een stuk code). Een onderzoeksproduct is het naslagwerk dat ingezien kan worden door mede-ingenieurs, waaruit blijkt dat jij je besluiten die geleid hebben tot je beroepsproduct systematisch hebt vastgelegd.

Een onderzoeksproduct moet daarom het resultaat zijn van het uitvoeren van een zeker onderzoeksprotocol. Een onderzoeksproduct kan verschillende vormen hebben, voorbeelden zijn een analyse (een formele redenering of afleiding), een verzameling metingen / observaties, een formeel ontwerp of specificatie, een welonderbouwd advies, een beschrijving van een testopstelling / experiment, een interview, etc. We zullen in de bespreking van het Methode-hoofdstuk hierna zien dat onderzoeksproducten heel

vastomlijmd zijn.

- Elk onderzoeksproduct(/-resultaat), en daarmee elk antwoord op een deelvraag, levert een aanwijsbare bijdrage aan de beantwoording van de het centrale vraagstuk.
- De formulering van je deelvraag moet onthullen wat je van plan bent te gaan *doen*. D.w.z. dat een vraag die bijvoorbeeld gaat leiden tot het maken van een ontwerp ook op die manier gesteld moet zijn (“Welke software-architectuur sluit het beste aan bij de eisen die ...” is bijvoorbeeld een vraag die onthult dat het resultaat iets te maken heeft met een ontwerp).

LET OP: Deelvragen komen niet uit de ijle lucht naar beneden gedwarreld! Dat betekent dat je **voordat** je een vraag presenteert moet motiveren waarom dit een zinnige onderzoeksrichting is! Gebruik hiervoor je gezond verstand en grijp terug op je inleiding, op het overleg met diegene(n) voor wie jij het probleem oplost en op de uitleg in het theoretisch kader!

3. Methode

Afhankelijk van het (deel)probleem kies je voor elke (deel)vraag één of meer passende onderzoeksmethoden. Het komt soms voor dat je meer dan één methode nodig hebt om een (deel)vraag te beantwoorden. Stem de keuze van je methode in dat geval af op het product dat jij met de vraag voor ogen hebt. Als je moeite hebt met kiezen, bekijk dan eerst of het niet nodig is om nog een deelvraag op te nemen!

Een aantal veelvoorkomende onderzoeksmethoden voor Computer Engineers zijn:

- **interview**
Bijvoorbeeld voor het vaststellen van requirements in overleg met je opdrachtgever.
- **enquete**
Bijvoorbeeld voor het vaststellen van draagvlak voor een nieuw idee, of het inventariseren van bestaande ideeën, in de organisatie. Deze onderzoeksmethode wordt door computer engineers maar heel zelden toegepast.
- **steekproef / observatiestudie**
Bijvoorbeeld voor het verkrijgen van een dataset vanuit een bestaand systeem in de vorm van een meetrapport.
- **analyse**
Bijvoorbeeld een (formele) logische redenering of eigen wiskundige analyse (berekeningen) op basis van uitgangspunten, aannames of formele intuïties (heuristieken / vuistregels), die je in staat stellen om in een later stadium een ontwerp te maken. Voorbeelden:
 - de vereenvoudiging van een logische expressie;
 - de krachtenberekening voor een motorspecificatie;
 - een telkundig verantwoord besluit om een bep. algoritme te gebruiken;

- het uitdrukken van een datarepresentatie in de vorm van bijvoorbeeld een matrix of een graaf;
- het bewijs voor de correctheid of complexiteit v.e. computationele taak;
- een requirementsanalyse bij het bouwen van een geïntegreerd softwarepakket dat eindgebruikers kent;
- etc.

Sommige analyses kunnen schematisch worden weergegeven, bijvoorbeeld met UML. Je kunt een analyse ook gebruiken om een warboel aan richtlijnen, standaarden, specificaties en developer notes (in datasheets, RFCs, IEEE, NEN-normen, internationale regelgeving, etc.) overzichtelijk te presenteren. Naast je analytische vermogens steunt een analyse vrijwel altijd op relevante (toegepast-)wetenschappelijke bronnen.

Veel studenten hebben de neiging deze onderzoeksmethode een literatuurstudie te noemen. Die term vinden wij verwarrend met het oog op het theoretisch kader / vooronderzoek. Dat zou de eigenlijke literatuurstudie met daarin jouw vooronderzoek moeten bevatten. Techneten kennen daarom geen aparte (deelvraag)onderzoeksmethode ‘literatuurstudie’. Als je naar literatuur kijkt tijdens je eigenlijke onderzoek, dan is dat altijd met een bepaald doel. Dat doel benoemen we hier als de onderzoeksmethode (meestal analyse).

- **(multi)criteria-analyse**

Bijvoorbeeld voor het selecteren van de best passende oplossing uit een inventaris van meerdere mogelijkheden of een andere vorm van vergelijkend waren-onderzoek (w.o. pakketselectie en prestatievergelijkingen). Dit is een variant op de hierboven genoemde methode ‘analyse’, die een vast protocol kent (kandidaten inventariseren, kanshebbers isoleren (short list opstellen), criteria vaststellen, wegingen per criterium vaststellen, criteria per kandidaat beoordelen, resultaten uitzetten in tabelvorm).

- **ontwerp**

Bijvoorbeeld een ontwerp voor een elektronisch circuit, een database-ontwerp (ERD), een algoritme (PSD), een 3D CAD-tekening, of een objectgeoriënteerd softwaremodel (UML). Wat je ook nodig hebt, je resultaat wordt *altijd* uitgedrukt in een **formele taal**.

Een belangrijk onderdeel van een ontwerp is de verantwoording van je ontwerpkeuzes. Door je ontwerpen in formele talen uit te drukken maak je aan je lezer duidelijk op welke wijze jij de architectuur van je oplossing voor je ziet. De onderzoeksmethode is erop gericht om je ontwerpkeuzes toe te lichten op basis van criteria die uit een eerdere (requirements-, multicriteria-)analyse komen.

- **experiment**

Bijvoorbeeld voor het testen van een ontwerp, de onderlinge vergelijking van meerdere realisatie-opties, de verantwoording van de realisatie, de validatie van een ontwerp (*proof of concept*) of de vergelijking met andere implementaties (d.w.z. als onderdeel van een grotere meetopstelling).

- **anders, namelijk . . .**

Het kan natuurlijk voorkomen dat je een exotisch of maatwerk onderzoeksinstrument ontwikkelt dat zich niet laat categoriseren in deze lijst – uiteraard worden dezelfde eisen gesteld aan je bespreking en invulling ervan als die gelden voor de hierboven genoemde categorieën!

Elk onderzoeksresultaat, en daarmee elk antwoord op een deelvraag, valt te bereiken door minstens één, maar vaak meerdere, onderzoeksmethoden uit te werken in een (meet)instrument dat een kader voor interpretatie geeft. In je methodehoofdstuk beantwoord je daarom voor elke deelvraag (en in een artikel dus alleen voor je centrale onderzoeksvraag) de volgende vragen (**STRIKT NOODZAKELIJK!!**):

- Welke methode kies je (WAT ga je doen)?
- Waarom kies je die methode (motivatie van het WAT)?
- Hoe ga je de methode toepassen (concreet plan van uitvoering)?
- Waarom doe je het op deze manier (motivatie van het HOE)?
- Wanneer heb je voldoende resultaten geboekt om je vraag te kunnen beantwoorden? Dit is een lastige vraag. Je zult moeten nadenken over de criteria waaraan je moet voldoen om ‘klaar’ te zijn met het toepassen van je onderzoeksmethode. Kortom, je zult een *instrument* moeten ontwikkelen. Een instrument is een protocol, een set van afspraken, waar jij je als onderzoeker aan belooft te houden tijdens het meten, redeneren en proberen. Een lat waarlangs je al je observaties, denkstappen en uitkomsten legt.

Voorbeelden van *instrumenten* per onderzoeksmethode zijn:

- Het format van je meetrapport
Welke posten komen er op je meetrapport en waarom precies deze?
- Het format van je interview
Welke vragen komen er in je interview en waarom precies deze?
- De wiskundige of logische formules waarmee je een analyse bouwt
Welke wiskundige / logische formules en principes zul je gaan toepassen en waarom precies deze?
- De tabel met gewichten-per-criterium uitgezet tegen de lijst van kandidaten (voor een multicriteria-analyse)
Welke criteria worden met welke weging gebruikt om de kandidaten te vergelijken? Waarom deze criteria en deze weging?
- De internationale standaard of specificatie als basis voor je ontwerp
Welke analytische uitkomsten moeten in het ontwerp zijn verwerkt om te

kunnen zeggen dat het alle nodige functionaliteit bevat? Welke ontwerptechnieken, architectuur en ‘best practices’ ga je toepassen? Waarom deze? Hoe kun je controleren wanneer je analytische uitkomsten volledig zijn opgenomen in je ontwerp?

- De beschrijving van een testopstelling / realisatie voor het uitvoeren van een experiment

Welke keuzes heb je gemaakt bij het schrijven van je code, het maken van je printplaat of het opzetten van je experiment? Waarom gebruik je specifiek deze componenten of technieken eigenlijk? Wijk je ergens af van het ontwerp? Zo ja, waarom? Hoe kun je controleren wanneer je ontwerp volledig is gerealiseerd? Wat zijn de verwachte uitkomsten van je experiment? Onder welke voorwaarden is het ‘geslaagd’ en naar welke parameters kijk je tijdens het testen? Waarom naar die parameters, en niet andere?

De data die je verzamelt door je instrumenten toe te passen, verwerk je meestal in de bijlagen van je artikel of onderzoeksrapport, tenzij je ze kunt presenteren in een afbeelding, tabel of broncode-opgave die op maximaal 1 pagina past. Bronnen en verwijzingen naar standaarden verwerk je in je bibliografie.

Als je een **artikel** schrijft (geldt dus NIET voor een onderzoeksscriptie), let dan in de methodeparagraaf op het volgende:

Het kan zijn dat één van de methoden centraal staat of van belang is voor je opdrachtgever (bijvoorbeeld een analyse), maar dat je ter validatie ook een ontwerp en realisatie maakt. *Dat betekent niet automatisch dat deze stappen ook onderdeel van je artikel uitmaken.* Een artikel heeft namelijk, zoals we eerder hebben afgesproken, een **smalle focus**. Niet alleen inhoudelijk, ook op het vlak van de aanpak moeten keuzes worden gemaakt. De nadruk ligt dus bewust op de belangrijkste onderdelen. Het kan zijn dat daardoor leuke implementatiedetails on(der)belicht blijven. Jammer dan – richt je op het waarmaken van je onderzoeksvoornemens en de gevolgen daarvan!

4. Resultaten

Communiceer hier de resultaten van de uitvoer van jouw onderzoeksmethode (specifiek: *de resultaten van de toepassing van je onderzoeksinstrument*) zo eerlijk en getrouw mogelijk.

Bij het opstellen van je hoofdvraag / deelvragen heb je nagedacht welk **product** je als *resultaat* van je {interview, enquête, steekproef, analyse, pakketselectie, ontwerp, experiment, ...} verwacht. Als het goed is heb je dat product in het vorige hoofdstuk ‘voorspeld’.

De manier van communiceren hangt af van de door jou gekozen onderzoeksmethode – een aantal voorbeelden:

- samenvatting van de antwoorden op interviewvragen in de vorm van een lijst of tabel;
- grafieken die de uitkomsten van een meetrapport weergeven;

- afleiding of redentie met daaronder de uitkomst van je analyse (inclusief samenvatting van de belangrijkste ontdekkingen uit de vakliteratuur, die relevant zijn om je analyse te kunnen uitvoeren);
- ingevulde kruistabel voor je multicriteria-analyse met daarin uitgezet: criteria, wegingen en keuze-opties met verwijzingen naar waar de informatie van elke keuze-optie te vinden is;
- een objectgeoriënteerd ontwerp van een softwaresysteem in UML of een elektrisch schema getekend volgens internationale standaard met een toelichting van je ontwerpkeuzes;
- screenshots van de testresultaten van de executie van door jou opgezette tests (m.b.v. het door jou geschreven computerprogramma) en een verwijzing naar de code in de bijlagen die hierachter schuilgaat;

In sommige gevallen spreekt een resultaat niet helemaal voor zichzelf en moet je het toelichten of zelfs een korte nabeschuiving schrijven. Je mag hier **NIET** opschrijven wat jij als professional vindt dat de resultaten betekenen (interpretatie)!

5. Conclusions

Formuleer conclusies door de resultaten te interpreteren en te beschrijven wat de resultaten betekenen voor vervolgvragen in je onderzoek. Als je bij de vraagstelling een onderzoeksvraag hebt geformuleerd, dan **MOET** je die hier beantwoorden. Een veelgemaakte **FOUT** is dat studenten in de conclusie de methode en resultaten gaan **samenvatten**. De resultaten zouden zo bondig moeten zijn opgetekend, dat dit **NOOIT** nodig zou moeten zijn!

In het geval van een (formele) analyse leg je hier kort uit wat jij denkt dat je afleiding, bewijs, redentie, etc. betekent voor de opdracht door bijvoorbeeld een lijst te maken met zaken die in het ontwerp moeten terugkomen. In het geval van een multicriteria-analyse bevat dit hoofdstuk de daadwerkelijke analyse. Bespreek bijvoorbeeld de top 5 van je resultaten en verantwoord waarom je (toch) voor een bepaalde optie kiest. Bij een ontwerp kijk je vooruit naar de implementatie, bij een experiment heb je het over de validiteit van je werk (bespreek je of de realisatie geslaagd is).

Voor een ingenieur heeft de conclusie bovendien veelal te maken met het verantwoord en dichttimmeren van (technische) besluiten - een aantal gewetensvragen die je jezelf kunt stellen tijdens het schrijven van een conclusie:

- wordt het nieuwe idee op basis van je enquête breed gedragen door de medewerkers in de organisatie en is het een goed idee om de nieuwe koers te gaan varen?
- blijkt uit je meetrapport dat het systeem zo nauwkeurig is als je voor ogen had en kan het ingezet worden om een betrouwbare diagnose te krijgen over de toestand van een model in het veld?
- leidt je analyse tot voldoende kennisinjectie om de organisatie verder te helpen?

- leidt je ontwerp tot het kunnen opstapen van een validerend experiment? (een proof of concept)
- valideert het experiment (prototype) het eerder gemaakte ontwerp en kan de organisatie hiermee een productiemodel gaan fabriceren?

Analyseer in dit hoofdstuk dus de *gevolgen van* de resultaten!

6. Aanbevelingen

Uit de conclusie kan een nieuw probleem(pje) volgen. Als het probleem groot genoeg is voor vervolgonderzoek, dan volgt hier een aanbeveling voor een nieuwe onderzoeksrichting. Welke kant moet de organisatie in jouw ogen op? Geef adviezen waar nodig!

Kleine bijkomende problemen die snel oplosbaar zijn neem je natuurlijk mee tijdens je lopende onderzoek. Die maken dus al deel uit van de vorige hoofdstukken. Kleine wijzigingen die nog niet verkende alternatieven aan het licht brengen of oplossingen geven voor problemen die niks met het aangepakte probleem te maken hebben (de gevalletjes ‘gelukkig toeval’) kun je in deze aanbevelingen benoemen.

Naast vooruit kijken doe je hier ook een stap ‘naar boven’ (helikopterview) en kijk je naar het functionele probleem dat je in de inleiding hebt beschreven. Helpt jouw technische oplossing de organisatie in het (deels) oplossen van het oorspronkelijke **functionele** probleem? Zijn er ook andere probleemdomeneinen waarin jouw oplossing toegepast zou kunnen worden?

7. Voor- en nawerk

In de genoemde punten zijn de belangrijkste delen van de *inhoud* van een artikel besproken. Vanzelfsprekend is een artikel niet volledig zonder voorwerk (samenvatting, evt. dankwoord, inhoudsopgave, coverpage) en nawerk (lijsten, index, bibliografie, bijlagen, etc.). Zie voor deze elementen het paarse boekje, daarin staat dat duidelijk vermeld. Voor de volgorde voor de onderdelen van het voorwerk en nawerk houden we de voorgeschreven volgorde in het paarse boekje aan.