

# Basiskennis requirements

IREB CPRE foundation examenstof  
uitgelegd en praktisch gemaakt

Jan Jaap Cannegieter  
Hans van Loenhoud  
Stefan Staal  
Johan Zandhuis



Eburon  
Utrecht 2021

ISBN 978-94-6301-348-2

Academische Uitgeverij Eburon, Utrecht  
[www.eburon.nl](http://www.eburon.nl)

Foto auteurs: Nicole Rijkers Fotografie  
Omslagontwerp: Textcetera, Den Haag  
Grafisch ontwerp: Studio Iris, Leende

© 2021 Alle rechten voorbehouden. Niets uit deze uitgave mag worden verveelvoudigd, opgeslagen in een geautomatiseerd gegevensbestand, of openbaar gemaakt, in enige vorm of op enige wijze, hetzij elektronisch, mechanisch, door fotokopieën, opnamen, of op enig andere manier, zonder voorafgaande schriftelijke toestemming van de rechthebbende.

# Inhoudsopgave

<b>Voorwoord</b>	<b>5</b>
<b>Inleiding</b>	<b>11</b>
<b>1 Inleiding en overzicht van requirements engineering</b>	<b>13</b>
1.1 Requirements: Wat	13
1.2 Requirements engineering: Waarom	18
1.3 Requirements engineering: Waar	21
1.4 Requirements engineering: Hoe	22
1.5 De rol en taken van een requirementsanalist	25
1.6 Wat te leren over requirements engineering	25
1.7 Samenvatting	26
1.8 Oefenvragen	27
<b>2 Basisprincipes van requirements engineering</b>	<b>29</b>
2.1 Overzicht van de principes	29
2.2 De principes uitgelegd	30
2.3 Samenvatting	54
2.4 Oefenvragen	56
<b>3 Werkproducten en manieren van documenteren</b>	<b>58</b>
3.1 Werkproducten in requirements engineering	60
3.2 Werkproducten op basis van natuurlijke taal	73
3.3 Werkproducten gebaseerd op sjablonen	79
3.4 Werkproducten gebaseerd op modellen	85
3.5 Verklarende woordenlijsten	111
3.6 Requirementsdocumentatiestructuren	114
3.7 Prototypes in requirements engineering	116
3.8 Kwaliteitscriteria voor werkproducten en requirements	118
3.9 Samenvatting	121
3.10 Oefenvragen	123

<b>4</b>	<b>Het uitwerken van requirements</b>	<b>129</b>
4.1	Requirementsbronnen	131
4.2	Het eliciteren van requirements	145
4.3	Conflictoplossing met betrekking tot requirements	163
4.4	Het valideren van requirements	175
4.5	Samenvatting	183
4.6	Oefenvragen	185
<b>5</b>	<b>Proces- en werkstructuur</b>	<b>189</b>
5.1	Beïnvloedende factoren	190
5.2	Requirements engineering procesaspecten	199
5.3	Het configureren van een requirements engineering-proces	203
5.4	Aanpak voor configureren van het requirements engineering-proces	208
5.5	Samenvatting	208
5.6	Oefenvragen	209
<b>6</b>	<b>Requirementsmanagement</b>	<b>212</b>
6.1	Wat is requirementsmanagement?	215
6.2	Levenscyclusmanagement	216
6.3	Versiebeheer	218
6.4	Configuraties en baselines	220
6.5	Attributen en perspectieven	222
6.6	Traceerbaarheid	225
6.7	Omgaan met veranderingen	228
6.8	Prioritering	231
6.9	Samenvatting	236
6.10	Oefenvragen	237
<b>7</b>	<b>Tool-ondersteuning</b>	<b>239</b>
7.1	Tools in requirements engineering	239
7.2	Het introduceren van tools	243
7.3	Nadere informatie	248
7.4	Samenvatting	248
7.5	Oefenvragen	249
<b>8</b>	<b>Antwoorden op de oefenvragen</b>	<b>251</b>
	Hoofdstuk 1	251
	Hoofdstuk 2	254
	Hoofdstuk 3	256
	Hoofdstuk 4	264
	Hoofdstuk 5	269
	Hoofdstuk 6	271
	Hoofdstuk 7	274

<b>Index</b>	<b>277</b>
<b>Verwijzingen</b>	<b>280</b>
<b>Auteurs</b>	<b>284</b>
Jan Jaap Cannegieter	284
Hans van Loenhoud	284
Stefan Staal	285
Johan Zandhuis	285

# Inleiding

Dit boek bevat de basiskennis die je nodig hebt voor het opstellen, vastleggen en gebruiken van requirements in projecten of teams. Zoals je in het boek zult zien, zijn requirements één van de belangrijkste, zo niet dé belangrijkste sleutel naar succesvolle systeemontwikkeling, ongeacht of jouw organisatie dat met een watervalproces of met een Agile aanpak doet. Als je niet weet wat je moet maken, dan is de kans dat je het goede maakt niet zo groot. In die zin is basiskennis van requirementsprocessen voor iedere betrokkene bij systeemontwikkeling van belang.

Het boek heeft twee doelstellingen. Allereerst kun jij je met behulp van dit boek, plus eventueel een bijbehorende cursus of opleiding, voorbereiden op het Certified Professional for Requirements Engineering (CPRE) Foundation Level examen van de International Requirements Engineering Board (IREB®). Ten tweede bevat het boek veel praktische ervaringen en tips om requirements succesvol te achterhalen, vast te leggen en te gebruiken in jouw organisatie. Als jij je alleen wilt voorbereiden op het CPRE-examen, dan volstaat het de niet-gemarkeerde delen te leren en de bijbehorende oefenvragen te maken. De praktische ervaringen en tips staan in de tekst met de grijze balk in de kantlijn. Die zijn dus niet van belang voor het examen. Wil je niet alleen een papiertje halen, maar ook een goede analist worden, dan zijn die stukjes juist de praktijktips die je boven de theorie laten uitstijgen.

De niet-gemarkeerde delen zijn voor wat betreft inhoud, structuur en terminologie geheel conform de Nederlandse versie van de IREB® CPRE FL-syllabus, de Nederlandse vertalingen in de Engelstalige IREB® CPRE glossary en het Engelstalige IREB CPRE FL handbook; we zijn daar niet van afgeweken. De gemarkeerde delen zijn onze eigen ervaringen.

Even over IREB® en het CPRE Foundation Level examen. IREB®, de International Requirements Engineering Board, is een non-profitorganisatie en is de aanbieder van het Certified Professional for Requirements Engineering (CPRE) certificeringsprogramma. Het is een vereniging met vooraanstaande requirements engineering-vertegenwoordigers vanuit wetenschap en onderzoek, industrie en consultancy. De doelstelling van IREB® is het verbeteren van de praktijk met betrekking tot requirements. CPRE is een persoonlijk certificaat dat zich richt op professionals die zich bezighouden met requirements en daar hoge eisen aan stellen in termen van kwaliteit. Het certificaat is wereldwijd erkend en is levenslang geldig. Het certificeringsprogramma is ontwikkeld door IREB®, de inhoud wordt onderwezen door onafhankelijke opleidingsinstellingen en het examen wordt afgenomen door geaccrediteerde exameninstellingen. Naast het Foundation Level waar dit boek over gaat, biedt IREB® ook een aantal modules op Advanced Level en een Expert Level aan.

Gezien de vraag naar IREB®-cursussen en -examens, maar ook door de actieve IREB®-community, is er een groeiende aandacht voor requirements in het algemeen en IREB CPRE in het bijzonder. Dit boek draagt daaraan bij. Meer informatie over IREB® en de CPRE-examens vind je op <https://www.ireb.org/en/landingpage/nl/>.

Naast het boek kun je op de website [basiskennisrequirements.nl](http://basiskennisrequirements.nl) meer informatie vinden, zoals diverse sjablonen en artikelen.

# 1 | Inleiding en overzicht van requirements engineering

Dit boek gaat over het vakgebied requirements engineering. Het eerste hoofdstuk begint met het belang van requirements voor het succes van softwareontwikkelprojecten en de voornaamste redenen waarom requirements engineering soms niet goed wordt aangepakt. Ook komt het toepassingsgebied van requirements aan bod, evenals de meest voorkomende onderverdelingen in soorten requirements. Na introductie van de voornaamste begrippen volgt een beschrijving van de hoofdactiviteiten van requirements engineering. Deze worden gewoonlijk uitgevoerd door een requirementsanalist, die daarvoor over specifieke vaardigheden moet beschikken.

## 1.1 Requirements: Wat

Mensen en organisaties hebben wensen en behoeften om nieuwe dingen te ontwikkelen of bestaande zaken verder te ontwikkelen. We noemen dergelijke behoeften requirements.

### 1.1.1 Requirements

De term requirements reikt verder dan veel mensen aannemen; meer dan veel belanghebbenden in eerste instantie denken. Een goed en gemeenschappelijk begrip bij alle belanghebbenden van de term requirements is nodig voor een effectief requirementstraject.

#### *Vershil in beleving*

In de praktijk merken we dat de beleving bij de term requirement afhankelijk is van de rol die iemand heeft. Dit verschil in beleving is de eerste hindernis die je als requirementsanalist moet nemen:

- Zo kijkt een opdrachtgever bijvoorbeeld alleen naar globale requirements. Voor hem is het requirementstraject voltooid als het probleem dat opgelost moet worden en de globale eisen aan de oplossing helder zijn.
- Een toezichthouder denkt vooral vanuit de wetgeving. Het systeem moet aan bepaalde wettelijke regels voldoen.
- Gebruikers bekommeren zich veel meer over schermen en de gebruiksvriendelijkheid van het systeem.
- Degene die het moet bouwen, wil vooral precies weten waar het systeem uiteindelijk technisch aan moet voldoen.

Het te ontwikkelen systeem moet aan de requirements van alle belanghebbenden gaan voldoen.



**Definitie requirement:**

- (1) Een voorwaarde of mogelijkheid die een gebruiker nodig heeft voor het oplossen van een probleem of het bereiken van een doel.
- (2) Een voorwaarde of mogelijkheid die een systeem of systeemcomponent moet bieden om te voldoen aan een contract, standaardspecificatie of ander formeel opgelegd document.
- (3) Een gedocumenteerde weergave van een voorwaarde of mogelijkheid zoals is gesteld in (1) of (2).

Bovenstaande definitie is vertaald vanuit de internationale standaard IEEE Std.610.12-1990 (IEEE 610, 1990). Een moderne variant van deze definitie luidt:

1. Een behoefte waargenomen door een belanghebbende.
2. Een mogelijkheid of eigenschap die het systeem moet bieden of hebben.
3. Een vastlegging van een behoefte, mogelijkheid of eigenschap.

Een systematische representatie van requirements noemen we een requirementsspecificatie. De gedocumenteerde weergave van requirements kan zelf ook als requirement worden beschouwd. De requirementsspecificatie kan bijvoorbeeld als juridisch referentiekader worden gebruikt om bij contractuele discussies te beoordelen of aan de requirements is voldaan.

### 1.1.2 Soorten requirements

Er bestaan verschillende soorten requirements (Glinz, 2020):

- *Functionele requirements* hebben betrekking op een resultaat of het gedrag dat door een functie van het systeem moet worden geleverd c.q. getoond. Dit omvat onder andere requirements voor gegevens en de interactie van een systeem met zijn omgeving.
- *Kwaliteitsrequirements* hebben betrekking op kwaliteitsaspecten die niet onder de functionele requirements vallen, zoals performance, beschikbaarheid, beveiligbaarheid en betrouwbaarheid.
- *Beperkingen* zijn requirements die de oplossingsruimte begrenzen binnen datgene wat nodig is om aan de gegeven functionele requirements en kwaliteitsrequirements te voldoen.

Een geautomatiseerd systeem moet aan al deze soorten requirements voldoen. Vaak wordt daarbij overigens geen onderscheid gemaakt tussen kwaliteitsrequirements en beperkingen; ze worden dan samengevat als *niet-functionele requirements*.

*Functionele requirements* gaan over de gewenste resultaten, functies, interacties en het gedrag van het systeem. Ze geven aan welke functionaliteit ofwel welke mogelijkheden het te ontwikkelen systeem moet bieden. Bijvoorbeeld: het systeem moet het totaalbedrag van de bestelde artikelen berekenen.

*Kwaliteitsrequirements* geven aan op welke wijze het beoogde systeem haar functionaliteit moet aanbieden. Dit wordt ook wel de verschijningsvorm genoemd. Bijvoorbeeld: het systeem moet tienduizend gebruikers tegelijkertijd aankunnen zonder dat de performance daalt, of: het systeem mag de functionaliteit alleen bieden aan daartoe geautoriseerde gebruikersgroepen.

Een kwaliteitsrequirement kan aanleiding geven tot het definiëren van nieuwe aanvullende functionele requirements. Dit is onder meer bij beveiligbaarheid dikwijls het geval. Bij telebankieren bijvoorbeeld is toegangsbeveiliging met alleen een wachtwoord niet voldoende. Afhankelijk van de vereiste mate van beveiliging is aanvullende functionaliteit nodig, bijvoorbeeld een 'tweestapsauthenticatie'. Kwaliteitsrequirements hebben meer impact op de softwarearchitectuur dan functionele requirements. Toch blijven ze te vaak onderbelicht. Hoofdstuk 3 gaat dieper in op de problemen met het documenteren van kwaliteitsrequirements en op welke manier een requirementsanalist dit kan oplossen.

#### *Invloed op de softwarearchitectuur*

We hebben meegemaakt dat een door een leverancier regelmatig toegepaste softwarearchitectuur tijdens het achterhalen en kwantificeren van de kwaliteitsrequirements volledig op de schop ging. De architectuur voldeed niet aan de verwachte prestatie-eisen. De leverancier had aangenomen dat er een modern communicatienetwerk met hoge bandbreedte beschikbaar zou zijn. In werkelijkheid moest er gecommuniceerd worden met een vestiging in een land waar een stabiele netwerk- of telefonieverbinding eerder uitzondering dan regel was. Kwaliteitsrequirements kunnen dus veel invloed hebben op systemen en dikwijls juist de architectuur.

*Beperkingen* limiteren de oplossingsruimte of het ontwikkelproces en verkleinen daarmee de bewegingsvrijheid van het ontwikkelteam. In tegenstelling tot functionele en kwaliteitsrequirements is het niet mogelijk om beperkingen te implementeren. Je implementeert de oplossing en de uiteindelijke oplossing houdt rekening met de beperking. Voorbeelden van beperkingen zijn: het systeem moet gebruikmaken van een Oracle-database, en: het systeem moet backward compatible zijn met vorige systeemversies.

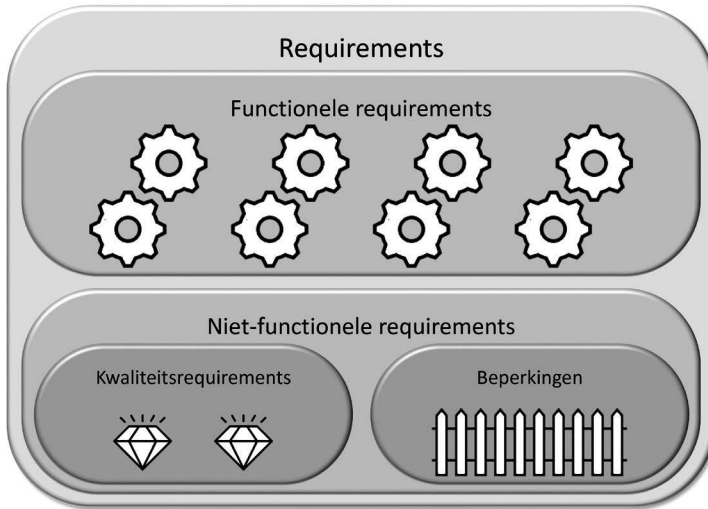
#### *Een zinvolle beperking*

Bij een grote organisatie was het de gewoonte om voor marketingcampagnes klantbestanden te gebruiken. De gegevens werden vlak voor de daadwerkelijke campagne gekopieerd vanuit een centraal relatiebestand of van een andere campagne hergebruikt. Dit had als voordeel dat het snel en goedkoop te realiseren was. Anders zou er een online koppeling naar het centrale relatiebestand gebouwd moeten worden. Een nadeel was dat het voor de campagne gebruikte klantenbestand niet de allerlaatste wijzigingen bevatte. Men realiseerde zich niet dat die laatste wijzigingen ook uitschrijvingen van overleden klanten konden bevatten. De vele klachten, pijnlijke discussies met nabestaanden en imagoschade leidden ertoe dat het topmanagement een rigide beperking voor het gebruik van relatiebestanden oplegde. Het werd verplicht om bij het benaderen van klanten alleen gebruik te maken van de online gegevens vanuit het centrale relatiebestand. Ontheffing van deze beperking moest schriftelijk bij de directie aangevraagd worden. Het werd al snel duidelijk dat je die toestemming niet kreeg.

*Achterliggende reden*

We adviseren requirementsanalisten altijd kritisch te zijn op beperkingen en ze niet te snel als voldongen feiten te beschouwen. Onze ervaring is namelijk dat ze nogal eens op onjuiste veronderstellingen berusten of eigenlijk verkapte oplossingen zijn. Vraag daarom goed door naar de achterliggende reden van een beperking. Daarmee zeggen we niet dat je alle beperkingen van tafel moet zien te krijgen. Een organisatie kan ook goede redenen hebben om een beperking aan het project op te leggen, bijvoorbeeld vanuit het IT-beleid of vanuit concurrentieoverwegingen.

Figuur 1.1 geeft de soorten requirements schematisch weer.



Figuur 1.1. Soorten requirements

*Requirementstypering*

Er bestaan vele indelingen van requirements die allemaal op een andere wijze onderscheid maken in typen requirements.

Niet-functionele requirements gaan, zoals de naam aangeeft, niet over de specifieke functionaliteit van het systeem maar over alle andere typen requirements. Het gaat dan om kwaliteitsrequirements en beperkingen.

Naast de indeling in functionele en niet-functionele requirements kom je ook de onderverdeling in perspectief in de praktijk tegen. Bekijk je de requirements vanuit het perspectief van de systeemontwikkelaar, vanuit de gebruiker of vanuit een bedrijfsdoel?

Deze onderverdeling komt in paragraaf 1.3 verder aan de orde.

### 1.1.3 Requirements engineering

Het vakgebied dat zich bezighoudt met het opstellen en beheren van requirements heet requirements engineering. Het moet ervoor zorgen dat de te ontwikkelen software voldoet aan de behoeften van de belanghebbenden. Requirementsanalyse is een veelgebruikt synoniem voor requirements engineering.

Door requirements engineering systematisch en gedisciplineerd uit te voeren en daarbij best practices toe te passen, is de kans op succes het grootst.

**Definitie requirements engineering (soms afgekort als RE):**

Een systematische en gedisciplineerde benadering voor het specificeren en beheren van requirements met als doel het begrijpen van de wensen en behoeften van belanghebbenden en het minimaliseren van het risico op een opgeleverd systeem dat niet aan de wensen en behoeften van belanghebbenden voldoet.

*Creativiteit*

Uit deze definitie zou je kunnen opmaken dat requirements engineering een mechanische benadering is, waarin geen ruimte is voor creativiteit. Dat is onjuist. Als requirementsanalist moet je bijvoorbeeld creatief zijn bij het achterhalen van de requirements, bij de wijze waarop je informatie overbrengt aan de belanghebbenden en bij het bewerkstelligen van overeenstemming over de requirements en de gesprekken die daarvoor nodig zijn. Sommige requirements zullen door jou als requirementsanalist zelf worden bedacht en – doordat belanghebbenden ze herkennen en omarmen – realiteit worden. Meer hierover lees je in hoofdstuk 4.

### 1.1.4 Belanghebbende

In de voorgaande alinea komt het begrip *belanghebbende* naar voren. Deze term is essentieel binnen requirements engineering.

**Definitie belanghebbende:**

Een persoon of organisatie die (direct of indirect) invloed heeft op de requirements van het systeem of door dat systeem wordt beïnvloed.

De belanghebbenden vormen de belangrijkste bron van requirements. Het voldoen aan behoeften en wensen van belanghebbenden komt ook als een van de basisprincipes terug in hoofdstuk 2. Voorbeelden van belanghebbenden zijn de eindgebruikers van het systeem, afdelingen of organisaties die gebruikmaken van informatie uit het systeem, de helpdesk, brancheorganisaties en de projectteamleden. Hoofdstuk 4 gaat dieper in op de wijze waarop je als requirementsanalist belanghebbenden kunt benutten als bron voor requirements en hoe je dit op gestructureerde wijze kunt doen.

### 1.1.5 Systeem

In requirements engineering volgens IREB® geldt een brede definitie voor de term systeem. Een systeem kan van alles zijn wat moet worden gebouwd of doorontwikkeld. Voorbeelden van systemen zijn:

- Producten die aan klanten worden geleverd.
- Diensten die aan klanten ter beschikking worden gesteld.
- Alle andere op te leveren producten, zoals apparaten, procedures of hulpmiddelen die mensen en organisaties helpen een specifiek doel te bereiken.
- Samenstellingen of onderdelen van producten, diensten of andere op te leveren producten.

**Definitie systeem:**

- (1) Algemeen: een principe voor ordening en structurering.
- (2) In engineering: Een coherente afgebakende set van elementen die – bij een gecoördineerde actie – een bepaald doel bereikt.

Let op dat een systeem andere systemen of componenten als subsystemen mag bevatten.

Volgens de toelichting op de definitie (Glinz, 2020) mag een systeem het doel op verschillende manieren bereiken:

- Aflevering van het systeem op de plaats waar het wordt gebruikt.
- Verkopen of leveren van het systeem aan gebruikers, als een product.
- Ter beschikking stellen van de systeemfuncties door dienstverleners als een dienst voor gebruikers.

De term systeem geldt als een paraplueterm die zowel producten, diensten, apparaten, apps en componenten betreft.

Hoofdstuk 2 behandelt begrip van het systeem als een van de essentiële principes voor het goed kunnen specificeren van de requirements voor dat systeem. Vervolgens gaat hoofdstuk 3 gaat dieper in op systeemmodellering.

## 1.2 Requirements engineering: Waarom

De kwaliteit van requirements is in belangrijke mate bepalend voor het succes van een project. Requirements geven aan wat het te ontwikkelen systeem moet kunnen om aan de behoefte van de belanghebbenden te voldoen. Ze dienen als vertrekpunt voor de overige ontwikkelactiviteiten en als input voor degenen die het moeten maken: wat moet er precies gemaakt worden, waaraan moet dat voldoen? Zelfs wanneer requirements niet expliciet worden geuit, bestaan ze toch.

Wanneer er fouten in de requirements zitten, werken die door in alle vervolgfasen van het project. De uitdrukking *'garbage in is garbage out'* is daarbij van toepassing. Het risico bestaat dan, dat het opgeleverde systeem niet aan de behoeften van de gebruikers voldoet en dat de doelstel-

lingen niet worden gehaald. Bovendien heeft het herstellen van fouten in requirements verderop in het ontwikkeltraject hoge kosten en uitloop tot gevolg. Ondanks het feit dat je met requirements engineering niets ontwerpt of bouwt, voegt het wel degelijk waarde toe aan het proces van het ontwikkelen en evolueren van een systeem. Adequate requirements engineering leidt tot:

- Vermindering van het risico om het verkeerde systeem te ontwikkelen.
- Een beter begrip van het probleem.
- Een basis voor een raming van de ontwikkelinspanning en -kosten.
- Invulling van een noodzakelijke randvoorwaarde voor het kunnen testen van het systeem.

### *Belang van requirements*

Er zijn veel onderzoeken gedaan naar het belang van requirements voor het succes van softwareontwikkelpromen. Deze onderzoeken tonen de noodzaak aan voor zorgvuldige requirements engineering. We noemen een aantal conclusies die uit deze onderzoeken te trekken zijn:

- **Toename herstelkosten**  
Barry Boehm concludeert dat het herstellen van fouten in latere fasen van het systeemontwikkelpromen 50 tot 200 keer duurder is dan in vroege ontwikkelfasen (Boehm, 1981). Dit staat sinds die tijd bekend als de ‘Wet van Boehm’.
- **Toename projectinspanning**  
Naar aanleiding van een grootschalig benchmarkonderzoek onder softwareontwikkelpromen, komt Capers Jones tot de conclusie dat 40 tot 50% van de projectinspanning wordt besteed aan het herstellen van fouten in de requirements (Jones, 2000). Foute requirements leiden tot het maken van producten waar niemand op zit te wachten.
- **Oorzaak van problemen**  
Forrester Research stelt in een onderzoek dat bij 71% van de falende softwareontwikkelpromen de oorzaak herleidbaar is naar fouten in de requirements (Schwaber, Leganza, & Daniels, 2006). Onderzoek van het Software Engineering Institute toont aan dat 42 tot 64% van de fouten in projecten ontstaan door slechte requirements (Firesmith, 2003). De percentages verschillen, maar beide onderzoeken geven aan dat veel van de problemen in projecten veroorzaakt worden door fouten in de requirements.

Op basis van onze ervaringen in projecten durven wij te stellen dat goede en volledige requirements één van de belangrijkste succesfactoren van succesvolle projecten of teams zijn. Veel, zo niet alle mislukte projecten of niet-succesvolle teams die we hebben gezien, hadden geen, slechte of onvolledige requirements. Alle succesvolle projecten of teams hadden goede en volledige requirements.

#### 1.2.1 Symptomen van ontoereikende requirements engineering

Menselijk gedrag werkt het onzorgvuldig opstellen van requirements in de hand. Onze communicatie met anderen is niet zuiver rationeel, maar wordt beïnvloed door externe omstandigheden en persoonlijke kenmerken. De voornaamste oorzaken van ontoereikende requirements door menselijk gedrag zijn:

- *Tijdsdruk, haast* om te beginnen met de bouw van het systeem  
Opdrachtgevers, maar ook projectleiders, product owners en andere belanghebbenden, willen graag het nieuwe systeem zo snel mogelijk tot hun beschikking hebben. De requirements-fase zien ze doorgaans als tijdverspilling omdat er dan nog niets wordt gebouwd. Onbegrip en onwetendheid over het belang van goede requirements zorgen ervoor dat requirementsanalisten onvoldoende tijd krijgen om de requirements op te stellen. Met name opdrachtgevers voeren de druk nogal eens hoog op.
- *Communicatieproblemen* tussen betrokken partijen  
Verschillen in kennis en ervaring zorgen ervoor dat wat voor de één vanzelfsprekend en logisch is, voor een ander nieuw kan zijn. Het referentiekader verschilt van persoon tot persoon. Termen kunnen daardoor anders worden uitgelegd. Bovendien ligt het doen van aannames op de loer. Hoofdstuk 3 gaat onder andere in op de problemen die ontstaan bij het formuleren van requirements in natuurlijke taal en geeft technieken om het risico op miscommunicatie te verkleinen.
- *Niet expliciet gemaakte requirements*  
Belanghebbenden verkeren vaak ten onrechte in de veronderstelling dat bepaalde requirements voor zich spreken, dat ze toch vanzelfsprekend zijn. Hierdoor maken ze niet al hun wensen expliciet en blijven er requirements onbenoemd. Dit gebeurt vaak onbewust. Meer hierover lees je in hoofdstuk 4. Belanghebbenden leggen de nadruk op de functionaliteit die ze in de huidige situatie missen of waaraan ze door de geplande veranderingen behoefte krijgen. De functionaliteit waarover ze al beschikken noemen ze gewoonlijk niet. Dit leidt tot misverstanden later in het ontwikkeltraject. Wanneer ontwerpers of ontwikkelaars niet op de hoogte zijn van bepaalde behoeften, kunnen ze daar immers ook geen rekening mee houden.
- *Ontoereikende requirements engineering-opleiding en vaardigheden*  
Binnen projecten is de rol van requirementsanalist niet altijd expliciet belegd. Soms wordt deze impliciet verwacht bij een projectmanager of product owner, soms bij een contractmanager of bij een architect. Op zich allemaal mogelijk, mits ze voldoende begrip hebben van het vak van requirements engineering en daarbij inderdaad een gestructureerde en gedisciplineerde benadering hanteren. Onderzoek (PMI, 2014) toont aan dat dit slechts in minder dan de helft van de projecten het geval is.

#### *Ruimte voor verbetering*

Tegenwoordig wordt er bij werving van personeel voor bepaalde rollen vaak om relevante certificaten gevraagd. Vrijwel elke zichzelf respecterende projectmanager heeft minimaal een Prince2 of PMI-certificaat; zonder dat kom je nauwelijks meer binnen. Voor inzet van softwaretesters is inmiddels het beschikken over een TMAP- of iSTQB-certificaat een minimumvereiste. Maar hoe vaak kom je bij invulling van de requirementsrol binnen een project een specifieke opleidingseis over requirementsvaardigheden tegen? Toch raar eigenlijk, als je beseft dat de requirements aan de basis staan van een succesvol project: *garbage in is garbage out*.

Wanneer de requirementsfase niet naar behoren is uitgevoerd, leidt dat in eerste instantie tot ontbrekende en onduidelijk geformuleerde requirements. Dit zijn symptomen van een gebrekkige requirements engineering.

#### *Harde feiten*

De eerdergenoemde Barry Boehm (Boehm, 1981) en ook Steve Easterbrook (Easterbrook, et al., 1998) hebben onderzocht wat voor soort fouten er in requirements voorkomen. Hoewel ze andere woorden gebruiken en op verschillende percentages uitkomen, zijn de onderzoeksresultaten veelzeggend.

<u>Boehm</u>		<u>Easterbrook</u>	
Incorrect feit	49%	Ongedocumenteerde aannames	30%
Weglatting	29%	Ontoereikendheid	27%
Inconsistentie	13%	Traceerbaarheid en inconsistentie	24%
Dubbelzinnigheid	5%	Onnauwkeurige terminologie	16%
Verkeerde plaats	2%	Logische fouten	3%
Anders	2%		

Dit zijn weliswaar oude onderzoeken maar de huidige praktijk wijst uit dat de situatie sindsdien niet substantieel verbeterd is. De onderzoeken impliceren dat er nog veel te verbeteren valt in ons vakgebied. Door ons meer te richten op het voorkomen van de genoemde type fouten, worden veel problemen later in het ontwikkeltraject ondervangen.

Bijvoorbeeld:

- Als we alle aannames gaan verifiëren, daalt het aantal fouten in de requirements met ruim een derde.
- Als we meer aandacht besteden aan het volledig maken van de requirements, is daar veel winst mee te behalen, namelijk bijna een derde van de fouten in requirements.
- Als we alert zijn op tegenstrijdigheden in requirements, kunnen we daarmee ongeveer een vijfde van de fouten in requirements elimineren.
- Als we de terminologie consequent toepassen, scheelt dat ongeveer een tiende van de fouten in requirements.

### 1.3 Requirements engineering: Waar

Requirements engineering kan worden toegepast op requirements voor elk soort systeem. Vandaag de dag wordt requirements engineering echter vooral toegepast voor systemen waarin software een grote rol speelt. Dergelijke systemen bestaan doorgaans uit softwarecomponenten, fysieke elementen (hardware zoals computers, sensoren, schermen en andere apparaten) en organisatorische elementen (personen, rollen of functies, bedrijfsprocessen, wet- en regelgeving). Systemen die zowel software als hardware bevatten, worden wel *cyber-fysieke systemen* genoemd, systemen die zowel software, hardware als organisatorische aspecten omvatten *socio-technische* systemen.



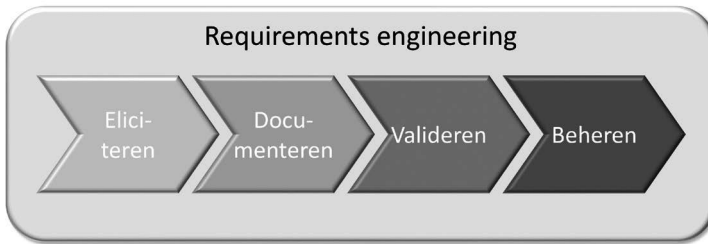
We onderscheiden vijf perspectieven op requirements:

- *Belanghebbendenrequirements* – Dit zijn de wensen en behoeften van belanghebbenden waaraan het te bouwen systeem moet voldoen, bekeken vanuit het perspectief van belanghebbenden.
- *Businessrequirements* – Dit zijn de met het systeem te behalen doelen. De businessrequirements representeren de toegevoegde waarde van het systeem voor de organisatie en geven aan waarom het belangrijk is voor de organisatie. Businessrequirements zijn geformuleerd vanuit het perspectief van de bedrijfsdoelstellingen.
- *Gebruikersrequirements* – Dit zijn de activiteiten, processen of taken die een gebruiker met behulp van het systeem wil uitvoeren. De gebruikersrequirements beschrijven op welke manier de businessrequirements behaald worden en welke werkzaamheden en acties door het systeem uitgevoerd of ondersteund worden. De gebruikersrequirements worden geformuleerd vanuit het perspectief van de bedrijfsvoering en de gebruikers.
- *Systeemrequirements* – Dit is het gewenste gedrag of de gewenste kwaliteit van het systeem. In de systeemrequirements ligt vast aan welke eisen het systeem moet voldoen. Ze geven aan welke acties het systeem in welke situaties moet uitvoeren. De systeemrequirements worden geformuleerd vanuit het perspectief van het systeem. Daar waar het systeem een softwaresysteem betreft, noemen we dit *softwarerequirements*.
- *Domeinrequirements* – Dit zijn vereiste domeineigenschappen van een socio-technisch of een cyber-fysiek systeem, die voortkomen uit het specifieke toepassingsdomein van een systeem en daarmee gelden binnen die specifieke (domein) context van dat systeem.

De requirementsperspectieven hierboven komen grotendeels overeen met de ISO-standaard (ISO/IEC JTC 1, 2018). Domeinrequirements zijn niet in deze norm opgenomen. Ze zijn echter wel van belang om hier te vermelden, aangezien domeinrequirements als een belangrijk onderdeel naar voren komen tijdens behandeling van principe 4 in hoofdstuk 2 over de systeemcontext.

## 1.4 Requirements engineering: Hoe

Requirements engineering is een veelomvattend vakgebied dat onder te verdelen is in vier hoofdactiviteiten. Om de requirements scherp te krijgen, moet een requirementsanalist ze eliciteren, documenteren en laten valideren. Verder moet hij de requirements beheren. Requirementsanalyse en conflictoplossing worden beschouwd als onderdeel van de elicitatie.



Figuur 1.2 Hoofdactiviteiten van requirements engineering.

Figuur 1.2 laat de hoofdactiviteiten van requirements engineering zien.

- Eliciteren*

Eliciteren is een woord voor het ontlokken en achterhalen van requirements. Het omvat meer dan alleen het ophalen of inventariseren van requirements. De belanghebbenden helpen bij het expliciet maken van hun requirements is juist een belangrijke taak van een requirementsanalist. Belanghebbenden maken daarbij een bewustwordingsproces door. Het is een actief proces waarbij een requirementsanalist aan de belanghebbenden requirements ontlokt. Dit doet hij door prikkelende vragen te stellen, creativiteit te bevorderen, alternatieven te suggereren en te reflecteren op de geuite behoeften. Een requirementsanalist maakt hiervoor gebruik van elicitatietechnieken zoals brainstormsessies, interviews en veldobservaties. Hoofdstuk 4 gaat uitvoerig in op het eliciteren van requirements. Aangezien de behoeften van de belanghebbenden zelden geheel synchroon lopen, komen we vaak tegenstrijdige requirements tegen. De requirementsanalist moet deze tegenstrijdigheden analyseren. Vervolgens moeten de belanghebbenden daarover in overleg treden, zodat er één gemeenschappelijke en gedragen set aan requirements ontstaat. Een requirementsanalist leidt dit proces en heeft daarvoor verschillende technieken voor conflictoplossing tot zijn beschikking.
- Documenteren*

Het documenteren van de geëliciteerde requirements is een belangrijke taak van een requirementsanalist. De uitdaging is om ze duidelijk, consistent en ondubbelzinnig te formuleren. Daarom gebruikt een requirementsanalist er zowel tekstuele beschrijvingen als conceptuele modellen voor. Hoofdstuk 3 gaat in op het documenteren van requirements en de verschillende documentatievormen. Een andere veelgebruikte benaming voor documenteren is *specificeren*. Goed gedocumenteerde requirements zijn van essentieel belang voor een IT-project dat een systeem wil opleveren dat voldoet aan de behoeften van de gebruikers. Documentatie is nodig om requirements (beter) overdraagbaar en onderhoudbaar te maken. Zonder documentatie zou er bij het verstrijken van de tijd veel informatie verloren gaan of anders uitgelegd worden.

- *Valideren*

Tijdens het valideren controleren de belanghebbenden de gedocumenteerde requirements. Daarbij kijken ze of de requirements hun eisen en wensen correct weergeven en of ze aan de gestelde kwaliteitscriteria voldoen. In hoofdstuk 4 wordt het validatieproces van requirements verder uitgewerkt.

- *Beheren*

Het beheren van requirements, ook wel requirementsmanagement genoemd, behelst het beheren en onderhouden van de geëliciteerde, gedocumenteerde en gevalideerde requirements en de requirementsdocumentatie als geheel. Een requirementsanalist zorgt er daarmee voor dat de requirements toegankelijk en goed bruikbaar blijven binnen het systeemontwikkeltraject. Requirementsbeheer omvat het prioriteren en traceren van requirements, wijzigingsbeheer, versiebeheer, het toekennen van attributen aan requirements en het maken van managementoverzichten. Hoofdstuk 6 gaat op al deze onderdelen in.

### *Geen vaste volgorde*

Het opsplitsen van requirements engineering in hoofdactiviteiten en de schematische weergave daarvan, roept wellicht het beeld op dat ze in een vaste volgorde en stap voor stap uitgevoerd worden. Dat beeld is onjuist. We willen benadrukken dat je deze activiteiten door elkaar en deels parallel uitvoert. De reden voor de opsplitsing is het hanteerbaar en bespreekbaar maken van een complex geheel. We gaan hier in hoofdstuk 4 dieper op in.

De activiteiten zijn tevens onafhankelijk van de gehanteerde systeemontwikkelaanpak. De hoofdactiviteiten zijn herkenbaar in ieder professioneel uitgevoerd systeemontwikkeltraject. Dat geldt zowel voor waterval- als voor Agile projecten. Je moet immers de behoeften en eisen van de belanghebbenden kennen om een waardevol systeem te kunnen opleveren.

Het requirements engineering-proces moet in iedere organisatie worden gefinetuned naar de specifieke projectsituatie. Er bestaat geen universele blauwdruk die je overal zomaar kunt toepassen. Er zijn wel belangrijke factoren te benoemen die het requirements engineering-proces sterk beïnvloeden.

Voorbeelden hiervan zijn:

- Het algehele systeemontwikkelproces (bijvoorbeeld Systems Engineering of lineaire ontwikkelmethoden versus iteratieve en Agile ontwikkelmethoden).
- De context waarbinnen het systeem moet worden ontwikkeld. Met name de relatie tussen de leverancier, klant en gebruikers is hierbij van belang.
- De beschikbaarheid en vaardigheden van de belanghebbenden.

Er is ook een wederzijdse relatie tussen het gekozen requirements engineering-proces en de requirementswerkproducten die gemaakt moeten worden. Hoofdstuk 4 gaat hier dieper op in.

*Specifieke werkproducten*

Als bijvoorbeeld een specifieke Agile aanpak zoals Scaled Agile Framework® (SAFe®) wordt gekozen, dan volgt daaruit ook het gebruik en opleveren van de daarbij behorende werkproducten, zoals strategic themes, epics en stories.

In een ontwikkelproject dat volgens Systems Engineering wordt opgepakt, is er vaak een functioneel technische specificatie (FTS), een specificatie met procesrequirements (PS), en andere (interface) specificaties waaraan het systeem of de subsystemen daarbinnen moeten voldoen. Daarbij hebben de specificaties ook een belangrijke contractuele juridische waarde.

## 1.5 De rol en taken van een requirementsanalist

Er zijn weinig mensen die ‘requirementsanalist’ als functieomschrijving hebben. Een requirementsanalist is iemand die het vak requirements engineering uitoefent.

Het is een rol die mensen oppakken die:

- requirements eliciteren, documenteren, valideren en/of beheren als onderdeel van hun taken;
- diepgaande kennis van requirements engineering hebben;
- de kloof tussen het probleem en de potentiële oplossingen kunnen overbruggen.

In de praktijk kom je de rol tegen in veel verschillende functies, zoals business analisten, applicatiespecialisten, product owners en system engineers. Soms vervullen zelfs ontwikkelaars de rol van requirementsanalist.

Ook voor andere rollen en functies die betrokken zijn bij systeemontwikkeling, zoals projectleiders, ontwerpers, testers en de CTO is het handig en raadzaam dat zij kennis hebben van requirements engineering. Op die manier kunnen zij het werk en de toegevoegde waarde van requirements engineering plaatsen en kunnen ze actief bijdragen aan de technieken die een requirementsanalist hanteert.

## 1.6 Wat te leren over requirements engineering

Voor het eliciteren, documenteren, laten valideren, onderhandelen over en managen van requirements moet een requirementsanalist over uiteenlopende vaardigheden beschikken. Dit boek behandelt in verschillende hoofdstukken de basiskennis en -vaardigheden die een requirementsanalist moet beheersen. Dat omvat naast het beheersen van technieken en analytische vaardigheden ook zogenaamde soft skills, zoals luisteren, bemiddelen, onderhandelen en empathisch vermogen.

Tot de basiskennis behoren de negen fundamentele principes van requirements engineering. Als requirementsanalist moet je deze principes kennen en je er bewust van zijn tijdens het uitvoeren van je taken. Deze principes vind je terug in hoofdstuk 2.

Het documenteren van requirements in verschillende vormen is onderwerp van hoofdstuk 3. Requirementsdocumenten kunnen variëren in omvang en mate van detail, afhankelijk van de aard van het systeem en de toegepaste requirements engineering-processen.

Hoofdstuk 4 behandelt het uitwerken van requirements voor verschillende toepassingen. Het geeft een overzicht van taken, activiteiten en technieken van het eliciteren tot en met het valideren van requirements.

Hoofdstuk 5 gaat dieper in op requirements engineering-processen en het afstemmen ervan op de specifieke situatie. Het behandelt de factoren die daarbij in overweging moeten worden genomen.

Beheer van requirements staat centraal in hoofdstuk 6. Het gaat daarbij om het beheer van individuele requirements, het beheer van requirementsdocumenten en om het beheren van de relatie naar het systeem dat gebaseerd is op deze requirements. Daarbij maakt de requirementsanalist gebruik van specifieke technieken om dit beheer goed uit te kunnen voeren.

Ten slotte behandelt hoofdstuk 7 het gebruik van ondersteunende tools voor requirements engineering. Zonder adequate tools blijkt het in de praktijk moeilijk of zelfs onmogelijk om een (vaak omvangrijke) set aan requirements in onderlinge samenhang goed te kunnen beheren. Het hoofdstuk geeft inzicht in hoe tools de requirementsanalist kunnen ondersteunen bij de uitvoering van requirements engineering-taken.

## 1.7 Samenvatting

Requirements engineering is een specifiek vakgebied:

- Houdt zich bezig met de requirements van belanghebbenden voor een systeem.
- Voegt waarde toe door het risico te beperken dat het geleverde systeem niet aan de wensen en behoeften van de belanghebbenden voldoet.
- Belanghebbenden zijn de belangrijkste bron voor requirements.
- Requirements engineering niet naar behoren uitgevoerd => ontbrekende en/of onduidelijk geformuleerde requirements => tijd- en kostenoverschrijding.

Soorten requirements:

- Functionele requirements
- Kwaliteitsrequirements
- Beperkingen

Indeling requirements naar verschillende perspectieven:

- Belanghebbendenrequirements
- Businessrequirements
- Gebruikersrequirements
- Systeemrequirements
- Domeinrequirements