

Succes met de requirements!

derde herziene druk

Ontwikkeling, validatie en
beheer van requirements
voor informatiesystemen

Succes met de requirements!

Ontwikkeling, validatie en beheer van requirements voor informatiesystemen

Derde, herziene druk

Martin Arendsen

Jan Jaap Cannegieter

Arno Grund

Petra Heck

Serge de Klerk

Johan Zandhuis



Meer informatie over deze en andere uitgaven kunt u verkrijgen bij:

Sdu Klantenservice
Postbus 20014
2500 EA Den Haag
tel.: (070) 378 98 80
www.sdu.nl/service

© 2012 Sdu Uitgevers bv, Den Haag
Academic Service is een imprint van Sdu Uitgevers bv.

Eerste druk, eerste oplage september 2008
tweede oplage februari 2009
Tweede, herziene druk, eerste oplage mei 2010
tweede oplage september 2011

Redactie en zetwerk eerste druk: Bagas & partners, Nijmegen
Zetwerk tweede en derde druk: Peter Verwey Grafische Producties bv, Heemstede
Omslagontwerp: MMX, Bergambacht

ISBN 978 90 12 58488 3
NUR 982

Alle rechten voorbehouden. Alle auteursrechten en databankrechten ten aanzien van deze uitgave worden uitdrukkelijk voorbehouden. Deze rechten berusten bij Sdu Uitgevers bv.

Behoudens de in of krachtens de Auteurswet gestelde uitzonderingen, mag niets uit deze uitgave worden verveelvoudigd, opgeslagen in een geautomatiseerd gegevensbestand of openbaar gemaakt in enige vorm of op enige wijze, hetzij elektronisch, mechanisch, door fotokopieën, opnamen of enige andere manier, zonder voorafgaande schriftelijke toestemming van de uitgever.

Voor zover het maken van reprografische verveelvoudigingen uit deze uitgave is toegestaan op grond van artikel 16 h Auteurswet, dient men de daarvoor wettelijk verschuldigde vergoedingen te voldoen aan de Stichting Reprorecht (Postbus 3051, 2130 KB Hoofddorp, www.reprorecht.nl). Voor het overnemen van gedeelte(n) uit deze uitgave in bloemlezingen, readers en andere compilatiewerken (artikel 16 Auteurswet) dient men zich te wenden tot de Stichting PRO (Stichting Publicatie- en Reproductierechten Organisatie, Postbus 3060, 2130 KB Hoofddorp, www.cedar.nl/pro). Voor het overnemen van een gedeelte van deze uitgave ten behoeve van commerciële doeleinden dient men zich te wenden tot de uitgever.

Hoewel aan de totstandkoming van deze uitgave de uiterste zorg is besteed, kan voor de afwezigheid van eventuele (druk)fouten en onvolledigheden niet worden ingestaan en aanvaarden de auteur(s), redacteur(en) en uitgever deswege geen aansprakelijkheid voor de gevolgen van eventueel voorkomende fouten en onvolledigheden.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the publisher's prior consent.

While every effort has been made to ensure the reliability of the information presented in this publication, Sdu Uitgevers neither guarantees the accuracy of the data contained herein nor accepts responsibility for errors or omissions or their consequences.

Inhoud

Voorwoord	vii
Aanbeveling	ix
Dankwoord	x
1 Inleiding	1
1.1 Het belang van juiste en volledige requirements	1
1.2 Doelgroep	5
1.3 Opbouw van het boek	6
Deel 1 Requirements binnen systeemontwikkeltrajecten	7
2 Overzicht van het requirementsproces	9
2.1 Wat zijn requirements?	9
2.2 Systeem en systeemcontext	10
2.3 Requirements in het ontwikkeltraject	12
2.4 De soorten requirements	15
2.5 Het requirementsproces in hoofdlijnen	21
2.6 Requirements en verantwoordelijkheden	22
3 Requirementsontwikkeling	25
3.1 Het proces van requirementsontwikkeling	25
3.2 Management van belanghebbenden	30
3.3 De rol van de requirementsanalist	33
3.4 Requirementsontwikkeling in de praktijk	36
4 Technieken voor requirementsontwikkeling	39
4.1 Elicitatietechnieken	39
4.2 Analysetechnieken	45
4.3 Specificatietechnieken	48
4.4 Use cases	54
4.5 User stories	58
4.6 Technieken voor niet-functionele requirements	59
4.7 Prioriteringstechnieken	62
5 Validatie van requirements	65
5.1 Doel van de validatie van requirements	65
5.2 Eisen aan requirements	66
5.3 Validatietechnieken	71
5.4 Het organiseren van validatie	74
6 Requirementsmanagement	77
6.1 Activiteiten in het kader van requirementsmanagement	77
6.2 Intake en identificatie	78
6.3 Traceerbaarheid	80
6.4 Wijzigingsbeheer	83
6.5 Verificatie	86

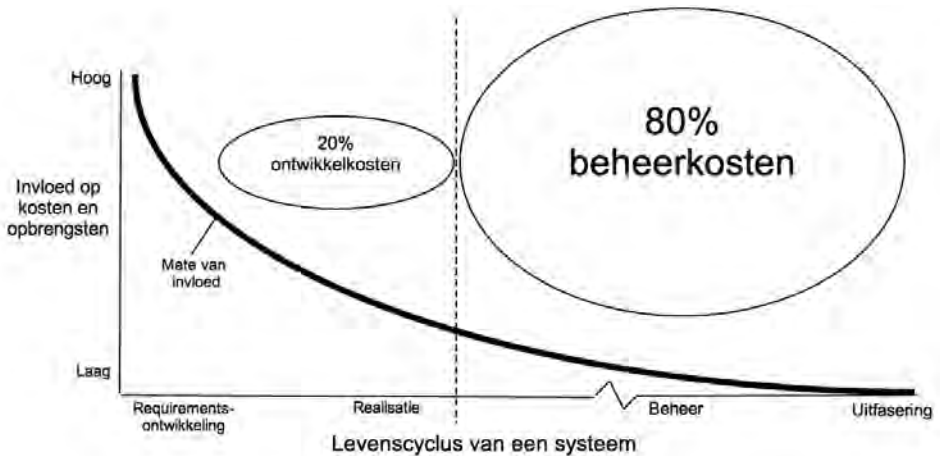
7	Requirements en agile	89
7.1	Het agile requirementsproces	89
7.2	Requirementsanalist en agile	94
7.3	Schatten	95
Deel 2 Requirements binnen organisaties		97
8	Requirementslevenscyclusbeheer	99
8.1	Requirementslevenscyclusbeheer	99
8.2	Het ontstaan van een requirement	99
8.3	Beheer van een requirement	101
8.4	Het verdwijnen van requirements	103
8.5	Statusregistratie van requirements	103
8.6	Requirementsattributen	105
8.7	Een voorbeeldproces	106
9	Verbetering van requirementsprocessen	109
9.1	Stap voor stap verbeteren	109
9.2	Voordat we gaan verbeteren	110
9.3	Implementatiemodel	111
9.4	Volgorde van implementatie	119
9.5	Verder verbeteren na implementatie	121
10	Schatting van de omvang	123
10.1	Relatie met requirementsontwikkeling	123
10.2	De verschillende schattingsmomenten	123
10.3	Nauwkeurigheid van de schattingen	124
10.4	Toelichting op de schattingsmethoden	126
10.5	Voortgangsmeting met behulp van requirements	131
11	Requirementsmanagementtooling	135
11.1	Tooling als ondersteuning	135
11.2	Beschikbare tools	137
11.3	De 'toolbox'	137
11.4	Het opstellen van de toolbox	138
Deel 3 Bijlagen		143
Bijlage A Requirements en standaarden		145
Bijlage B Voorbeeldsjablonen		162
Bijlage C Voorbeelden van procesbeschrijvingen		186
Bijlage D Verklarende woordenlijst		194
Bijlage E Referenties		197
Bijlage F Aanbevolen literatuur		200
Nawoord		201
Over de auteurs		202
Trefwoordenlijst		205

Inleiding

1.1 Het belang van juiste en volledige requirements

De kwaliteit van de requirements bepaalt in hoge mate het succes van ICT-projecten. Dit is niet alleen een algemeen aanvaarde stelling, het is ook logisch te beredeneren en het blijkt uit diverse onderzoeken. De logica is dat als de input van een proces slecht is, de output dat ook is. Met andere woorden: 'garbage in is garbage out'. En requirements zijn belangrijke input voor projecten. Zo maken de projectleider, de ontwerper en de tester gebruik van de requirements. Als de requirements slecht zijn, is het onwaarschijnlijk dat het werk van projectleider, ontwerper en tester goed is. Ongeacht hoe goed er gewerkt wordt in het vervolgtraject, als de requirements onjuist, onvolledig of niet consistent zijn, is de kans dat het eindproduct voldoet klein.

Daarnaast is de kwaliteit en inhoud van requirements tijdens de systeemontwikkeling ook van invloed op de beheerkosten: beheer is volgend op ontwikkeling. De totale kosten gedurende de complete levenscyclus van een systeem bestaan eerst uit ontwikkeling van een systeem, de zogenaamde projectkosten, en daarna uit exploitatie en onderhoud, de beheerkosten. Uit onderzoek van Berghout blijkt dat de beheerkosten groter zijn dan de projectkosten, vaak een verhouding van 20% projectkosten en 80% beheerkosten. Maar de mate van beïnvloeding van de totale kosten is juist tijdens projecten groter dan tijdens beheer. Beheerrequirements die tijdens de ontwikkelfase worden gesteld bepalen het gemak, of moeite, waarmee een systeem tijdens de beheerfase is aan te passen. Denk hierbij aan onderhoudbaarheid, aanpasbaarheid, uitbreidbaarheid en testbaarheid van het systeem. Hetzelfde geldt voor de opbrengsten van ICT: deze worden gegenereerd tijdens de exploitatie maar al in hoge mate bepaald tijdens de systeemontwikkeling. Het feit dat zowel kosten als opbrengsten vooral tijdens de beheerfase worden gerealiseerd maar voornamelijk tijdens de projectfase zijn te beïnvloeden, is ook bekend als de beheerparadox (Berghout, 2001). In de praktijk sturen veel opdrachtgevers van projecten en projectmanagers juist op de projectkosten en niet op de totale levenscycluskosten. Dat de focus op louter projectkosten een explosie van beheerkosten veroorzaakt, is dan niet verwonderlijk. Dit inzicht in kosten pleit ervoor om tijdens de projectfase nadrukkelijk requirements te stellen aan opbrengsten van het systeem en aan de benodigde beheerinspanning. Requirements zijn een krachtig hulpmiddel om de beheerkosten, als het grootste deel van de systeemkosten, tijdens de systeemontwikkeling te beïnvloeden.



Figuur 1.1: Requirements en levenscycluskosten

Het belang van requirements voor het succes van het project blijkt ook uit onderzoek. Zo stelt Forrester dat 71 procent van de fouten in projecten het gevolg zijn van slechte requirements (Schwaber, 2006). Andere onderzoeken geven aan dat 50 procent van de fouten in projecten aan requirements gerelateerd zijn (Standish Group, 2001 en Wiegers, 2001). Ook het Software Engineering Institute van Carnegie Mellon University, een autoriteit op het gebied van systeemontwikkeling, geeft aan dat slechte requirementsprocessen de belangrijkste oorzaak van fouten vormen; 42 tot 64 procent van de fouten ontstaan door slechte requirements (Firesmith, 2003).

Slechte requirements hebben hoge herstelkosten en uitloop van projecten tot gevolg. Barry Boehm stelt dat het oplossen van fouten in requirements in de bouwfase 10 keer duurder is dan in de requirementsfase en in onderhoud 50 tot 200 keer duurder (Boehm, 1988). Steve McConnell geeft aan dat het 10 tot 100 keer duurder is een fout in de requirements na de requirementsfase te herstellen (McConnell, 2004). Dat bij veel projecten dit zogenoemde herstelwerk vaak voorkomt, blijkt onder andere uit onderzoek van Capers Jones. Zijn onderzoek toont aan dat 40 tot 50 procent van de projectinspanning besteed wordt aan het herstellen van fouten (Capers Jones, 2000). Wiegers stelt zelfs dat 80 procent van de projectinspanning wordt besteed aan het herstellen van fouten (Wiegers, 2001).

Uit onderzoek van Gartner blijkt dat projecten onbedoeld 70 procent groter worden dan oorspronkelijk gepland (Gartner, 2006). Deze zogenoemde 'Wedge-factor' is volgens Gartner te wijten aan falende requirementsprocessen.

Dit zijn nog lang niet alle onderzoeken op dit vlak. Al met al kan gesteld worden dat het onderschatten van het belang van juiste en volledige requirements leidt tot verlies van tijd, geld en kwaliteit.

Hoewel het belang van requirements niet genoeg kan worden benadrukt, blijkt het opstellen, valideren en beheren van requirements in de praktijk geen eenvoudige opgave. Uit de literatuur zijn onderzoeken bekend waarbij require-

ments zijn beoordeeld en requirementsproblemen zijn gegroepeerd. Voorbeelden zijn Boehm (Boehm, 1981) en Easterbrook (Easterbrook, 1998):

Boehm:		Easterbrook:	
Incorrect feit	49%	Ongedocumenteerde aannames	30%
Weglating	29%	Ontoereikendheid	27%
Inconsistentie	13%	Traceerbaarheid en inconsistentie	24%
Dubbelzinnigheid	5%	Onnauwkeurige terminologie	16%
Verkeerde plaats	2%	Logische fouten	3%
Anders	2%		
	100%		100%

De oorsprong van veel van deze problemen is te herleiden tot het requirementsproces. Hierbij moet gedacht worden aan problemen en situaties zoals:

- De business voelt zich niet verantwoordelijk voor de requirements en ziet dit als een ICT-aangelegenheid, terwijl het systeem toch voor de business wordt gerealiseerd. ICT gaat hierbij op de stoel van de business zitten.
- De rol van requirementsanalist wordt uitgevoerd door iemand die het domein niet kent of niet over de juiste vaardigheden beschikt.
- Er zijn bij het requirementsproces meerdere belanghebbenden betrokken. De betrokkenheid van meerdere belanghebbenden is een potentiële bron van communicatiestoringen. Ook kan het voorkomen dat enkele belanghebbenden niet bij het requirementsproces betrokken worden.
- De niet-functionele requirements worden niet, slechts beperkt of niet meetbaar beschreven, waardoor ontwerpbeslissingen bij de realisatie impliciet blijven.
- Requirements veranderen en hiervoor moet het wijzigingsproces goed in de organisatie zijn ingebed.

Er is ook een aantal actuele ontwikkelingen die meer aandacht voor requirements legitimeren. De eerste is het professionaliseren van de relatie tussen opdrachtgevers en (interne) ICT-afdelingen. Organisaties met een ICT-afdeling gaan steeds meer naar een demand-supplyorganisatie. Dat wil zeggen dat de business zich steeds meer als opdrachtgever gaat opstellen en professionaliteit verwacht van de ICT-afdeling. De ICT-afdeling gaat zich op haar beurt meer opstellen als opdrachtnemer en verwacht professionaliteit van de business. Bij deze beweging hoort dat requirements van de juiste kwaliteit zijn en dat het project doelgericht de requirements realiseert. In het kielzog van het professionaliseren van de verhoudingen tussen de business en de ICT-afdeling neemt de behoefte aan een professioneel requirementsproces toe.

Een tweede ontwikkeling die meer aandacht voor requirements rechtvaardigt, is uitbesteding, ongeacht of dit uitbesteding naar een lokale organisatie ('on shore'), uitbesteding binnen Europa ('near shore') of uitbesteding buiten Europa ('off shore') is. Bij uitbesteding gaat het principe 'garbage in is garbage out' nog meer op dan bij realisatie door een interne ICT-afdeling. De reden hiervoor is dat als het systeem wordt gerealiseerd door een organisatie die zich

fysiek en organisatorisch verder weg bevindt, deze organisatie minder kennis heeft van de business van de opdrachtgever en dat communicatie moeilijker is. Daarom dienen requirements van nog hogere kwaliteit te zijn in vergelijking met systeemrealisatie door een interne ICT-afdeling. Zo mag er bij outsourcing naar een buitenlandse systeemontwikkelformaat geen impliciete kennis verondersteld worden. Dat betekent dat gedetailleerdere requirements vereist zijn.

Een derde ontwikkeling die meer aandacht voor requirements rechtvaardigt, is de behoefte aan betere beheersing van ICT-projecten. Zoals uit de cijfers van tabel 1.1 blijkt, worden ICT-projecten in de afgelopen jaren weliswaar iets succesvoller, maar is slechts een klein deel van de projecten echt succesvol.

	1994	1996	1998	2000	2002	2004	2009
Succes	16%	27%	26%	28%	34%	29%	32%
Falen	31%	40%	28%	24%	15%	18%	24%
Matig	53%	33%	46%	48%	51%	53%	44%

Tabel 1.1: Mate van succes van ICT-projecten (Standish Group, 1994, 1996, 1998, 2000, 2002, 2004 en 2009)

Uit deze cijfers komt naar voren dat de beheersing van projecten moet verbeteren. Als de projectsturing meer gebaseerd wordt op requirements, neemt de doelgerichtheid van projecten toe; hierdoor neemt ook de kans op een succesvol project toe.

De laatste actuele ontwikkeling die meer aandacht voor requirements legitimeert, is de toename van compliancy-eisen zoals SOX en SAS 70. SOX staat voor de Sarbanes-Oxley Act en is een Amerikaanse wet die in 2002 van kracht werd. SOX dient ervoor om deugdelijk bedrijfsbeheer te garanderen. Hiertoe moet elk bedrijf dat valt onder de SOX-wetgeving kunnen bewijzen dat het elk bedrijfsproces met mogelijk financiële implicaties volledig onder controle heeft. SAS 70 is een acroniem voor de verklaring over Auditing Standard 70. SAS 70 bevat professionele normen voor auditors en evaluatie van interne controles voor accountants. Effect van deze compliancy-eisen is dat systeemontwikkeling beter rekenschap moet kunnen geven van wijzigingen in systemen. Hierbij kan het requirementsproces een prominente rol spelen.

Voor alle belanghebbenden heeft het voordelen om projecten te starten met juiste en volledige requirements. De voordelen zijn:

- Goede requirements vormen een overeenstemming tussen opdrachtgever en opdrachtnemer over de vraag wat het systeem moet doen.
- Goede requirements vormen een goede basis voor het opstellen van de architectuur en het opstellen van het ontwerp.
- Goede requirements vormen een gedegen basis voor schattingen en planningen.
- Goede requirements beperken de ontwikkelkosten door minder herstelwerk.

- Goede requirements vormen een referentiekader voor reviews, inspecties en tests.
- Goede requirements vergemakkelijken het in gebruik nemen van het systeem.
- Goede requirements vormen buiten de context van projecten een basis voor het onderkennen van verbeteringen in systemen.

Juiste en volledige requirements zijn dus essentieel voor het succes van projecten en een aantal actuele ontwikkelingen legitimeert meer aandacht voor requirements. Dit boek levert een bijdrage aan het verbeteren van requirementsprocessen en aan de samenwerking tussen opdrachtgevers en opdrachtnemers in projecten. Daarmee levert dit boek een bijdrage aan het succes van ICT-projecten.

1.2 Doelgroep

Het doel van dit boek is het beschrijven van een toegankelijke en toepasbare aanpak van requirementsontwikkeling, -validatie en -management en een aantal verwante processen. In de praktijk bewezen toepassingen zijn het uitgangspunt van het boek, dus geen ongeteste nieuwigheden of theorieën, maar zaken die in de praktijk werken. Door het goed toepassen van deze 'best practices' kan bij veel organisaties genoeg winst behaald worden.

Bij requirementsprocessen zijn diverse belanghebbenden betrokken, zoals de opdrachtgever, verschillende gebruikersgroepen, informatiemanagers, functioneel beheerders, requirementsanalisten, architecten, ontwerpers, programmeurs en testers. Dit boek is voor al deze doelgroepen geschreven. Het boek richt zich dus niet specifiek op de businesskant of op de ICT-kant; beide partijen hebben even veel invloed op het succes van projecten. Daarom richt dit boek zich op beide groepen en wordt er een brug geslagen tussen deze werelden. Het boek vormt derhalve een gemeenschappelijke basis met behulp waarvan opdrachtgevers en opdrachtnemers hun samenwerking kunnen verbeteren.

Bij de lezer wordt enige ICT-kennis en kennis van de gang van zaken in ICT-projecten verondersteld. Kennis van en ervaring met requirementsprocessen wordt niet verondersteld.

Het boek richt zich niet specifiek op specialisten in requirementsontwikkeling, -validatie en -management. Deze specialisten kunnen hun kennis vergroten door de uitgebreidere, Engelstalige boeken te lezen. Hoewel we ervan overtuigd zijn dat dit boek ook voor specialisten nieuwe of aanvullende inzichten bevat, gaat het voor deze groep op enkele punten niet diep genoeg.

In dit boek worden de Engelse of 'verengelste' termen gebruikt waar deze ingeburgerd zijn of veel in de praktijk worden gebruikt. Af en toe zijn de termen in het Nederlands vertaald en is het Engelse woord tussen haakjes opgenomen. Daar waar in het boek 'hij' staat, dient 'hij/zij' gelezen te worden.

1.3 Opbouw van het boek

Dit boek is verdeeld in drie delen.

Hoofdstuk 2 tot en met 7 vormen het eerste deel *Requirements binnen systeemontwikkeltrajecten*. Dit deel gaat in op de basisbegrippen en requirementsprocessen die bij ieder individueel ontwikkeltraject van belang zijn.

Hoofdstuk 2 behandelt de basisbegrippen, de hoofdlijnen van het requirementsproces en de verdeling van verantwoordelijkheden rond requirements. Hoofdstuk 3 gaat over het ontwikkelen van requirements. Uit welke processtappen bestaat het ontwikkelen van requirements? Hoe worden requirements vastgelegd? Vervolgens behandelt hoofdstuk 4 de technieken voor het ontwikkelen van requirements.

Om te voorkomen dat requirements van onvoldoende kwaliteit zijn dienen ze gevalideerd te worden. Hoofdstuk 5 behandelt de eisen die aan requirements gesteld worden en hoe het proces van validatie van requirements dient plaats te vinden.

Tijdens het project moet zeker worden gesteld dat de requirements goed verwerkt worden in het systeem. Daarnaast moeten de requirements tijdens het project actueel worden gehouden. Deze activiteiten, aangeduid als requirementsmanagement, vormen het onderwerp van hoofdstuk 6.

Hoofdstuk 7 gaat nog apart in op de rol van requirements binnen agile ontwikkelmethoden en geeft aan hoe de materie uit de hoofdstukken 1 tot en met 6 in dergelijke projecten geïnterpreteerd moet worden.

Een requirement houdt niet op te bestaan na afloop van een project. Soms worden requirements in de loop van de tijd in andere systemen verwerkt of komen ze terug in andere projecten. Requirements dienen dus over projecten heen beheerd te worden. Dit is het onderwerp van hoofdstuk 8.

Hoofdstuk 9 gaat in op het verbeteren van requirementsprocessen. In dit hoofdstuk geven we aan wat een organisatie moet doen om haar requirementsprocessen blijvend te verbeteren.

Het onderwerp van hoofdstuk 10 is het begroten van projecten op basis van requirements. In dit hoofdstuk treft u ook de technieken aan die hiervoor gebruikt kunnen worden.

Hoofdstuk 11 besteedt aandacht aan requirementsmanagementtools. Wat voor soorten tools zijn er? Hoe kunnen we die tools in de requirementsprocessen gebruiken? Ook komt het selecteren van een tool aan bod.

In het derde deel staan een aantal bijlagen. Deze bevatten achtergrondmateriaal en hulpmiddelen die u in de praktijk kunt toepassen.

Bijlage A geeft beschrijvingen van de manier hoe in IEEE, CMMI, RUP, Prince2, ISO 25010 en TOGAF wordt omgegaan met requirements. Bijlage B bestaat uit een aantal sjablonen die tijdens het requirementsproces gebruikt kunnen worden. In bijlage C is een aantal voorbeeldbeschrijvingen van requirementsprocessen opgenomen. Bijlage D bevat een verklarende woordenlijst. Referenties en aanbevolen literatuur treft u aan in de bijlagen E en F.

DEEL **1**

**Requirements binnen
systeemontwikkeltrajecten**

Overzicht van het requirementsproces

Dit hoofdstuk geeft een overzicht van het requirementsproces. Wat is een requirement? Wat voor soorten requirements zijn er? Hoe ziet het requirementsproces er in hoofdlijnen uit? Wie heeft in welke fase van het requirementsproces welke verantwoordelijkheden? Op deze vragen wordt in dit hoofdstuk een antwoord gegeven. Daarmee zet dit hoofdstuk de kaders voor het vervolg van dit boek.

2.1 Wat zijn requirements?

Er bestaan in de literatuur meerdere definities van de term requirement. De IEEE Standard Glossary of Software Engineering Terminology definieert een requirement als (IEEE, 1998):

1. A condition or capability needed by a user to solve a problem or achieve an objective;
2. A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document;
3. A documented representation of a condition or capability as in 1 or 2.

Wiegiers combineert de punten 1 en 2 in de volgende definitie: 'A statement of a customer need or objective, or of a condition or capability that a product must possess to satisfy such a need or objective' (Wiegiers, 2003).

Uit de definities komt naar voren dat een requirement kan worden gezien als:

- een behoefte van een belanghebbende;
- een doelstelling van een belanghebbende;
- een voorwaarde aan een product;
- een eigenschap van een product.

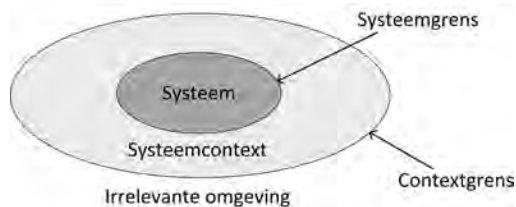
Dit boek definieert een requirement als een behoefte of doelstelling van een belanghebbende, danwel een eis, wens of beperking waaraan een systeem dient te voldoen om tegemoet te komen aan die behoefte of doelstelling.

De definitie spreekt over belanghebbenden in plaats van over eindgebruikers of klanten. Belanghebbenden is een bredere term en omvat alle actoren die met het systeem te maken hebben.

De requirements die aan een systeem worden gesteld, worden er ook requirements gesteld aan het project dat ervoor zorgt dat het systeem wordt gerealiseerd. Een voorbeeld is de gewenste einddatum van het project. Deze projectrequirements worden in dit boek niet afzonderlijk behandeld. De beschreven werkwijzen en technieken voor requirementsontwikkeling, -validatie en -management zijn ook van toepassing op projectrequirements, met het verschil dat projectrequirements alleen voor de duur van het project gelden.

2.2 Systeem en systeemcontext

Een nieuw te ontwikkelen systeem staat nooit op zichzelf. Het dient te passen in een bestaande omgeving. Daarnaast heeft de bestaande omgeving effect op de requirements waar het systeem aan dient te voldoen. Voor het vaststellen van de scope van het gehele ontwikkeltraject is het van belang de grenzen van het te ontwikkelen systeem eenduidig te definiëren. Ook is het van belang helder te communiceren met welke factoren buiten het te ontwikkelen systeem wel of juist geen rekening moet worden gehouden. Het transparant vaststellen van de scope van het systeem en de systeemcontext is daarvoor nodig. In figuur 2.1 zijn de verschillende begrippen die hierbij een rol spelen in samenhang met elkaar afgebeeld.



Figuur 2.1: Begrenzing van een systeem en systeemcontext

Zodra bepaalde oplossingsrichtingen in beeld komen, wordt ook duidelijk welke zaken door een bepaalde oplossingsrichting te beïnvloeden zijn en welke niet. Andersom wordt ook duidelijk waar een bepaalde oplossingsrichting rekening mee moet houden en welke omgevingsfactoren een oplossingsrichting mogelijk kunnen maken, kunnen bemoeilijken of zelfs kunnen verhinderen. Dit speelveld per oplossingsrichting wordt in kaart gebracht door het definiëren van de zogenoemde systeemgrens en contextgrens.

De systeemgrens en de contextgrens dienen gelijktijdig met het in kaart brengen van oplossingsrichtingen initieel te worden vastgesteld. Houd er rekening mee dat gedurende verdere uitwerking van requirements deze grenzen dikwijls nog wijzigen door toegenomen inzicht.

De systeemcontext wordt bepaald door twee grenzen, namelijk de systeemgrens en de systeemcontextgrens. Het gebied tussen die twee grenzen vormt

de systeemcontext. De systeemcontext is dat deel van de omgeving van een systeem dat voor het begrijpen en verder definiëren van de requirements relevant is (Glinz, 2012).

Doorlopend voorbeeld: systeemcontext

Huis	Datawarehouse
De verdere inrichting en bebouwing van de straat waar het huis wordt gebouwd. Bestemmingsplan. Ligging van het perceel ten opzichte van windrichtingen, achtertuin op het zuiden. De bewoners.	Opzet en inhoud van bronsystemen die data aanleveren. Wetgeving die eisen stelt aan op te leveren rapportages. Gebruikers, interne medewerkers en externe organisaties.

De systeemgrens scheidt het geplande systeem van zijn omgeving. Het onderscheidt de zaken die binnen de kaders van het systeemontwikkeltraject vormgegeven en veranderd kunnen worden van zaken in de omgeving die niet door het ontwikkelproces vormgegeven of veranderd kunnen worden (Glinz, 2012).

Doorlopend voorbeeld: systeemgrens

Huis	Datawarehouse
Aansluiting op riolering, energie, water, telecommunicatie. Erfafscheiding met de burens. De brievenbus.	Definitie van Extract-Transfer-Load (ETL) processen uit bronsystemen. Standaard dataleveringen. Aansluiting op bestaande infrastructuur.

Uit de vastgelegde systeemgrens blijkt welke grensoverschrijdende interactie met de context plaatsvindt. Indien het te ontwikkelen systeem interactie heeft met andere systemen in de context, dan definiëren zogenoemde interfacerequirements helder waaraan deze interactie dient te voldoen. Bestaande systemen kunnen op die manier beperkingen opleggen aan het te ontwikkelen systeem. Andersom legt het te ontwikkelen systeem beperkingen op aan systemen in haar context.

De contextgrens scheidt het relevante deel van de omgeving van een beoogd systeem van het irrelevante deel van de omgeving. Met het irrelevante deel wordt bedoeld: dat deel van de omgeving dat geen invloed heeft op het beoogde systeem en daarmee ook geen invloed heeft op de requirements van dat systeem (Glinz, 2012). Of iets wel of niet als relevant wordt beschouwd, hangt samen met keuzes en risico-inschattingen.

Doorlopend voorbeeld: contextgrens

Huis	Datawarehouse
Binnen scope: voorgenomen wijzigingen in het bestemmingsplan. Buiten scope: algemene economische ontwikkelingen, overstromingsrisico's.	Binnen scope: voorgenomen wetswijziging, concreet geplande wijzigingen in andere systemen. Buiten scope: reorganisaties, systeem aanpassingen die nog in onderzoek zijn.

Paragraaf 4.3.1. gaat dieper in op technieken voor het definiëren en modelleren van de systeemcontext.

Het vaststellen van de contextgrens zorgt ervoor dat het requirementsproces doelgericht en systematisch kan plaatsvinden. Tijdens het requirementsproces moet steeds bewaakt worden dat de requirements binnen de contextgrens blijven. Wijziging van een eerder vastgestelde en goedgekeurde systeemgrens of systeemcontext is van grote invloed op de requirements.

2.3 Requirements in het ontwikkeltraject

De systeemcontext geeft aan welke belanghebbenden in het requirementsproces moeten worden betrokken. Alle belanghebbenden hebben ieder hun eigen referentiekader. Deze belanghebbenden brengen hun requirements tijdens het proces in. Afhankelijk van de belanghebbende omvatten deze requirements onder meer de doelstellingen van de organisatie, een beschrijving van de taken die met het systeem uitgevoerd moeten worden, de informatiestromen die het systeem in en uit gaan, de integratie met andere systemen, enzovoort. Deze requirements moeten gedetailleerd genoeg zijn voor de personen die het systeem moeten maken of configureren. Dit betekent dat er meerdere niveaus van requirements te onderkennen zijn, waarbij ieder niveau van requirements bedoeld is voor een groep belanghebbenden. We bespreken nu de onderverdeling van de requirements in verschillende niveaus.

2.3.1 Businessrequirements

De doelstellingen van de business. Deze requirements bepalen de scope en de richting van de overige requirements. De businessrequirements geven antwoord op de waarom-vraag van het systeem: waarom moet dit systeem gemaakt worden?

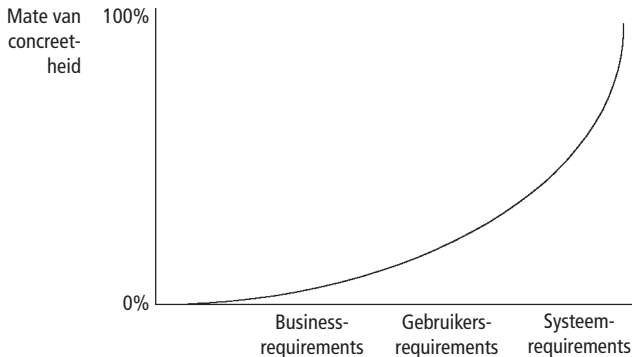
2.3.2 Gebruikersrequirements

De doelen en taken die de eindgebruikers van het systeem moeten kunnen uitvoeren. De gebruikersrequirements geven antwoord op de wat-vraag van het systeem: wat moet het systeem doen?

2.3.3 Systeemrequirements

Een systeemrequirement is een eis of beperking waaraan het systeem dient te voldoen om de business- en gebruikersrequirements te realiseren. De systeemrequirements geven antwoord op de hoe-vraag van het systeem: hoe moet het systeem werken om aan de bovenliggende requirements te voldoen?

De requirements worden tijdens het requirementsproces steeds concreter. Dit is afgebeeld in figuur 2.2.



Figuur 2.2: Concreetheid van requirements in de tijd

De requirements worden gerealiseerd in een ontwikkelproject. In de praktijk blijkt dat het per organisatie verschilt vanaf wanneer de requirementsontwikkeling en systeemontwikkelactiviteiten onder een project vallen, onderdeel zijn van een programma of gezien worden als lijnactiviteit. Algemeen kan gesteld worden dat de voor het project geldende businessrequirements en, afhankelijk van de systeemontwikkelmethode, de bijbehorende gebruikers-requirements bij het begin van een project bekend dienen te zijn. Het vereiste niveau van uitwerking van de requirements is afhankelijk van de gekozen systeemontwikkelmethode.

Er zijn diverse systeemontwikkelmethoden, die verschillende activiteiten en tussenproducten kennen. Drie belangrijke soorten systeemontwikkelmethoden zijn:

- watervalmethoden, bijvoorbeeld SDM en LAD;
- iteratieve methoden, bijvoorbeeld DSDM, RUP en IAD;
- agile methoden, bijvoorbeeld XP, Scrum en EVO.

Watervalmethoden worden gekenmerkt door een strikte fasering, waarbij een volgende fase niet begint voordat alle activiteiten van de vorige fase zijn afgerond en alle tussenproducten uit de vorige fase compleet en goedgekeurd zijn. Als een watervalmethode wordt gehanteerd, worden aan het begin van het project alle requirements tot op systeemniveau gespecificeerd. Pas als alle requirements zijn goedgekeurd, wordt gestart met de volgende fase.

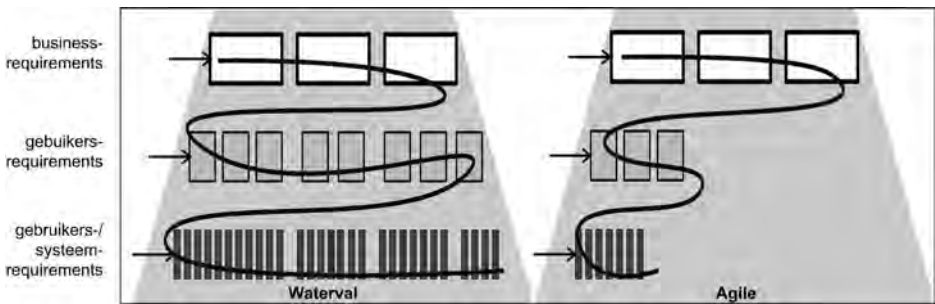
Bij iteratieve methoden wordt niet het hele systeem als geheel van tevoren gespecificeerd. Het systeem wordt onderverdeeld in kleine stukken, incrementen, die in korte cycli worden ontworpen en gebouwd. Eén cyclus wordt een iteratie genoemd. In principe worden alle iteraties aan het begin van het project vastgesteld. Een iteratie begint met requirements die veelal niet tot op het laagste detailniveau uitgewerkt zijn.

2.3.4 Agile methoden

Steeds meer organisaties ontwikkelen software volgens een agile methode. Agile (letterlijk vertaald als 'vlug en lenig') is geen specifieke methode, maar de

naam die is gegeven aan een manier van denken over softwareontwikkeling. SCRUM is een bekende methode waarin het agile gedachtengoed tot uiting komt.

Kenmerkend voor agile is dat er een korte ontwikkelcyclus is: in zogenoemde timeboxes (ook 'sprints' genoemd) van (doorgaans) twee tot vier weken wordt telkens een beperkt stuk functionaliteit volledig (vanaf ontwerp tot en met implementatie) gerealiseerd. Net als in traditionele (waterval)trajecten verloopt de requirementsontwikkeling van grof naar fijn. Anders dan bij waterval worden niet alle requirements vooraf tot in detail uitgewerkt, maar worden steeds alleen die requirements verder gedetailleerd die als eerste gerealiseerd gaan worden. Dit voorkomt dat al in een vroeg stadium tijd wordt besteed aan het ontwikkelen van requirements voor informatiebehoeften die al weer gewijzigd zijn wanneer ze gerealiseerd worden, of die uiteindelijk zelfs helemaal niet gerealiseerd worden. Ter illustratie is in figuur 2.3 het slingerpad uit figuur 3.4 (waterval) vergeleken met de agile denkwijze.



Figuur 2.3: Waterval vergeleken met agile methoden

Een ander kenmerk van agile is de nauwe samenwerking tussen het ontwikkelteam en de klant, met daarbij een sterke voorkeur voor visualisatie en directe communicatie van gebruikerswensen. Dit betekent dat niet alle gebruikerswensen tot in het kleinste detail op papier worden uitgewerkt.

Binnen agile projecten wordt gewerkt met multidisciplinaire teams waarin ook de klant vertegenwoordigd is. Hierbij bestaat een sterke voorkeur voor visualisatie en directe communicatie van gebruikerswensen. Dit betekent dat alle gebruikerswensen wel gedetailleerd in samenwerking tussen gebruiker en ontwikkelaar worden uitgewerkt, maar niet tot in het diepste detail op papier worden vastgelegd. De interactie tussen gebruiker en ontwikkelaar vervangt uitgebreide documentatie en maakt deze voor het ontwikkelen van werkende software onnodig. Een valkuil is dat er helemaal geen documentatie gemaakt wordt. Een bekend risico bij agile ontwikkeling is dat er aan het eind van het project te weinig gedegen documentatie ligt. Wanneer het projectteam is ontbonden, is een basale documentatie van requirements, user stories (zie hoofdstuk 4), architectuur, testen en dergelijke van cruciaal belang om het systeem te kunnen onderhouden.

In dit boek is geen uitgebreide behandeling van agile opgenomen. Wel wordt in hoofdstuk 7 aangegeven welke raakvlakken bestaan tussen agile en requirementsontwikkeling en requirementsmanagement zoals die in dit boek beschreven zijn. Voor een meer diepgaande bestudering van agile in het algemeen, en requirements binnen agile in het bijzonder, worden verschillende titels aanbevolen (Leffingwell, 2011; Beck, 2001; Cohn 2004).

In dit boek wordt geen voorkeur uitgesproken voor een bepaalde systeem-ontwikkelmethode. De voorgestelde aanpak is ook niet gebaseerd op of bedoeld voor een bepaalde systeemontwikkelmethode.

2.4 De soorten requirements

In paragraaf 2.3 is naar voren gekomen dat requirements worden onderverdeeld in:

- businessrequirements;
- gebruikersrequirements;
- systeemrequirements.

Op elk niveau kunnen we nog verder onderscheid maken tussen functionele en niet-functionele requirements. Functionele requirements hebben betrekking op de functies die het systeem voor de belanghebbenden moet vervullen. Niet-functionele requirements zijn eigenschappen of karakteristieken waaraan het systeem moet voldoen. De verschillende soorten requirements en hun samenhang zijn samengevat in figuur 2.4.

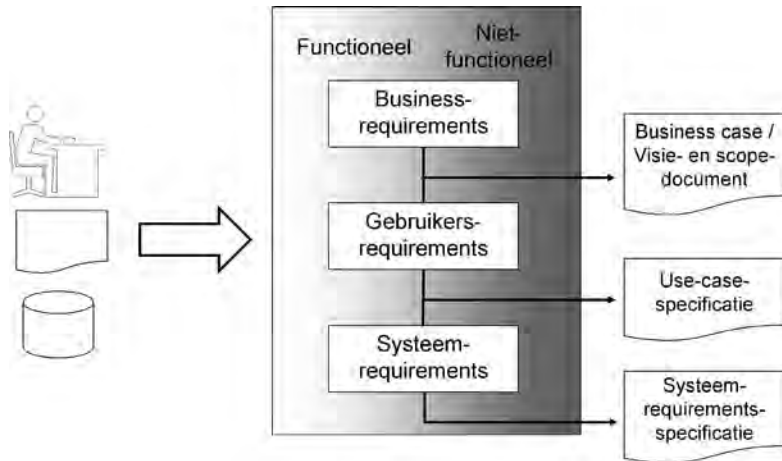
In figuur 2.4 is aan de linkerkant de input voor het requirementsproces opgenomen. Aan de rechterkant staan de documenten waarin de verschillende requirements beschreven worden.

Om de verschillende definities toe te lichten zijn twee doorlopende voorbeelden toegevoegd, een van een huis en een van een datawarehouseapplicatie.

Voor de requirements zijn er meerdere vormen van input, te weten de bedrijfsdoelstellingen, business rules, kwaliteitseisen en beperkingen. In de bedrijfsdoelstellingen is opgenomen wat de organisatie wil bereiken.

Doorlopend voorbeeld: bedrijfsdoelstellingen

Huis	Datawarehouse
Volgend jaar 5% geld (be)sparen.	Het aantal klanten groeit met 5% per jaar.



Figuur 2.4: Soorten requirements en hun samenhang

De business rules zijn beleidsaspecten, leidraden, standaarden of regels waaraan bij de uitvoering van het bedrijfsproces voldaan moet worden. De business rules bevatten de inhoudelijke eisen waar het systeem uiteindelijk aan moet voldoen.

Doorlopend voorbeeld: business rules

Huis	Datawarehouse
Zakelijke transacties: wij doen waar mogelijk zaken met een leverancier uit de regio (zelfde gemeente of direct aanliggende gemeenten).	Medewerkers hebben alleen inzage in de bedrijfsgegevens na toestemming van de directeur.

De kwaliteitseisen zijn de eisen aan de werking van het systeem. Voorbeelden van kwaliteitseisen zijn snelheid, beveiliging en gebruikersvriendelijkheid van het systeem.

Doorlopend voorbeeld: kwaliteitseisen

Huis	Datawarehouse
(Businessniveau) Brandveiligheid – de woning valt in de normale brandverzekering, geen verhoogd brandrisicotarief.	(Businessniveau) Prospectinfo komt snel, juist en volledig beschikbaar.
(Gebruikersniveau) Brandveiligheid – niet bang hoeven zijn voor brand bij het stoken van het houtvuur.	(Gebruikersniveau) Performance – de maximale tijd tussen opvragen van een overzicht en het tonen van de resultaten is maximaal tien seconden.
(Systeemniveau) Brandveiligheid – toegepaste dakbedekking is brandveilig of brandvertragend.	(Systeemniveau) Beveiligbaarheid – het systeem dient de autorisatie te controleren op basis van een gebruikersnaam en een wachtwoord.

De beperkingen limiteren de keuzes die de ontwikkelaars hebben in het ontwerp of de realisatie van het systeem. Deze kunnen bijvoorbeeld technisch van aard zijn; denk hierbij aan de beperkingen in ontwikkelplatformen of communicatieprotocollen. Beperkingen kunnen ook organisatorisch van aard zijn;

zo wordt de werking van het systeem in sommige gevallen beperkt door de functies en rollen zoals die in de organisatie bekend zijn.

Doorlopend voorbeeld: beperkingen

Huis	Datawarehouse
Voor het huis mag niet meer dan driemaal het bruto jaarsalaris worden geleend.	Voor databases gebruikt de organisatie type 'x'. Motivatie: dit beperkt onze licentiekosten en er is slechtst expertise van één type database nodig. Deze expertise bereikt ook een hoog niveau door het intensieve gebruik. De rapportgenerator dient gecertificeerd compatibel te zijn met het Operating System vastgelegd in de voor de organisatie gedefinieerde 'werkomgevingsstandaard'.

Aan de rechterkant in figuur 2.4 zijn voor de verschillende typen requirements de documenten afgebeeld. De hier gebruikte benamingen voor de documenten komen veel voor, maar zijn indicatief. Er zijn ontwikkelmethoden en documentatiestandaarden die sjablonen voorschrijven. Ook hebben sommige organisaties op basis van ervaring zelf sjablonen ontwikkeld.

2.4.1 Businessrequirements

De businessrequirements beschrijven wat een organisatie met een systeem wil bereiken. We spreken bewust van businessrequirements en niet van business case; de term business case wordt in de praktijk vaak verward met de term business-casedocument (Van der Molen, 2010). Businessrequirements geven antwoord op de waaromvraag: 'Waarom doen we dit?' De businessrequirements leggen een nadrukkelijke relatie met de bedrijfsdoelstellingen en de bedrijfsprocessen waaraan het systeem een bijdrage moet leveren. Alle veranderingen in het IT-domein komen voort uit de business en businessprocessen. Uiteindelijk dienen alle IT-projecten waarde voor de business te creëren. Deze benadering wordt ook wel 'business driven requirements engineering' genoemd.

De businessrequirements zijn in het algemeen afkomstig van het (top)management of de sponsor van het project. Bij complexe organisaties is vaak een seniorgebruiker op hiërarchisch hoog niveau aangewezen die vanuit de organisatie het mandaat heeft om businessrequirements te mogen bepalen. Of er zijn functionele boards ingericht, die de samenhang tussen te ontwikkelen of te beheren systemen en businessdoelstellingen formuleren en bewaken.

Tijdens het opstellen van de businessrequirements komen globale oplossingsrichtingen in beeld. Een oplossingsrichting geeft in grote lijnen de door te voeren aanpassingen in operationele processen en/of veranderingen in IT-systemen weer. Daarbij kan nog onderscheid gemaakt worden in hergebruik, koop of maak (maatwerk). De oplossingsrichtingen worden getoetst aan het geldende beleid ten aanzien van systeemarchitectuur. Ook worden de risico's en financiële aspecten per oplossingsrichting in kaart gebracht. Uiteindelijk wordt het businessdocument opgesteld, met daarin de vastlegging van businessrequirements, belanghebbenden, overwogen oplossingsrichtingen, risico's,

financiële aspecten en de uiteindelijk geselecteerde oplossingsrichting. Voor de gekozen oplossingsrichting worden vervolgens in het project de requirements op gebruikers en systeemniveau uitgewerkt.

Doorlopend voorbeeld: businessrequirements

Huis	Datawarehouse
Met een eigen huis besparen we 20% belasting om op die manier de bedrijfsdoelstellingen te realiseren. We moeten leven in gezamenlijkheid, waarbij ieder individu zich kan terugtrekken in een eigen omgeving.	Met een informatiesysteem dat voorziet in informatie- en rapportagebehoefte kan de organisatie 3 FTE op jaarbasis aan 'lijstjesmakerij' besparen. Met een ondersteunend informatiesysteem kan de organisatie 50% meer prospects opsporen om de bedrijfsdoelstelling te halen.

Bij verschillende standaarden voor project- en programmamanagement zoals PRINCE2 (PRINCE2, 2009) en PMBoK (PMI, 2009) geldt dat de rechtvaardiging voor het project transparant wordt vastgelegd in een business-casedocument. Daarnaast zijn er systeemontwikkelmethoden zoals RUP, die de doelstellingen van het ontwikkeltraject vastleggen in een visie- en scopedocument. De daarin gestelde business case of businessrequirements sturen de besluitvorming, waardoor wordt geborgd dat het project zich blijft focussen op de gestelde doelstellingen en de na te streven baten.

Bijlage B bevat voorbeeldsjablonen van een business-casedocument en een visie- en scopedocument.

2.4.2 Gebruikersrequirements

De gebruikersrequirements beschrijven de doelen en taken die gebruikers met het systeem moeten uitvoeren. Hierbij dient wel opgemerkt te worden dat de term 'gebruikers' breed gedefinieerd wordt; dit zijn zowel belanghebbenden aan de gebruikerskant als belanghebbenden aan de ICT-kant.

Voor het vastleggen van gebruikersrequirements zijn meerdere technieken beschikbaar. Een veel toegepaste techniek is het opstellen van use cases en scenariobeschrijvingen; zie hiervoor hoofdstuk 4. Basis voor de gebruikersrequirements zijn in eerste instantie de businessrequirements. Deze geven, samen met eventueel aanwezige procesbeschrijvingen¹, een goede indicatie van de gewenste functionaliteit van het systeem. In de gebruikersrequirements wordt deze functionaliteit verder geconcretiseerd. Hierbij wordt rekening gehouden met de geldende business rules.

1 Bij de ontwikkeling van standaardsoftware is er geen sprake van één organisatie. In deze gevallen wordt veelal gebruikgemaakt van een branchemodel voor de processen.

Doorlopend voorbeeld: gebruikersrequirements

Huis	Datawarehouse
De bewoners willen 's avonds gezellig in huis een houtvuur kunnen stoken.	De functioneel beheerder van het DWH moet kunnen controleren of het laden van gegevens uit bronsystemen goed is gegaan.
In de garage moeten vier fietsen en een auto passen.	Gebruikers in het Marketing-domein en in het Operations-domein dienen zelf een rapportage te kunnen samenstellen.
De bewoners dienen eenvoudig van openbaar vervoer gebruik te kunnen maken.	Ten behoeve van het MT dient de functioneel beheerder van het DWH de vaste maandelijkse prospectrapportages te kunnen uitdraaien op basis van het DWH.
De twee kinderen slapen ieder in een eigen ruimte.	

De gebruikersrequirements worden vastgelegd in een use-casespecificatie (zie bijlage B).

2.4.3 Systemrequirements

Systemrequirements zijn eisen of beperkingen waaraan het systeem dient te voldoen om de business- en gebruikersrequirements te realiseren. Indien een systeem uit meerdere deelsystemen bestaat, wordt ieder systeemrequirement toegewezen aan een of meer deelsystemen.

Doorlopend voorbeeld: systeemrequirements

Huis	Datawarehouse
De woning is voorzien van een open haard die geschikt is voor een houtvuur en die voldoet aan de Praktijkrichtlijn gasinstallaties – Deel 20: Blokkenvuurstellingen.	Iedere keer dat gegevens in het DWH worden geladen, voegt het DWH een loggingbericht toe aan de logging.
Om de fietsen en auto te stallen is een garage van 5 bij 4 meter nodig.	Het loggingbericht bestaat uit de datum en het tijdstip van het laden van de gegevens, om welke gegevens het gaat, wat het bronsysteem is en of het laden goed is gegaan.
	Het standaardoverzicht van nieuwe prospects bevat de organisatiename en de branchcode van de door de gebruiker opgegeven periode. Een periode bestaat uit een begindatum en tijdstip, en einddatum en tijdstip.

De systeemrequirements worden vastgelegd in systeemrequirementspecificatie (SRS). In bijlage B treft u een voorbeeldsjabloon van een SRS aan.

Acceptatiecriteria zijn meetbare criteria waaraan het eindproduct moet voldoen voordat de belanghebbenden het eindproduct accepteren (PRINCE2, 2009).

Doorlopend voorbeeld: acceptatiecriteria

Huis	Datawarehouse
Aan de hypotheekvoorwaarde dient minimaal te worden voldaan en aan 95% van alle gebruikers-requirements dient te zijn voldaan.	Aan 95% van de gebruikersrequirements met prioriteit 'hoog' is voldaan.
Tijdsgebondenheid: alle bovenstaande voorwaarden gelden bij ingebruikname van het huis.	Er zijn geen openstaande bevindingen van de categorie 'ernstig'.
	Tijdsgebondenheid: alle bovenstaande voorwaarden gelden vanaf de eerste release van het DWH.

2.4.4 Functionele en niet-functionele requirements

Business-, gebruikers- en systeemrequirements bestaan uit functionele en niet-functionele requirements. Functionele requirements beschrijven de functies die het systeem voor de belanghebbenden moet vervullen. Functionele requirements hebben betrekking op de volgende aspecten:

- gedrag;
- gegevens;
- foutafhandeling;
- dynamiek;
- presentatie;
- interfaces.

Niet-functionele requirements beschrijven een eigenschap of karakteristiek waar het systeem aan moet voldoen. Deze hebben onder andere betrekking op de volgende aspecten zoals genoemd in ISO 25010 (ISO, 2011):

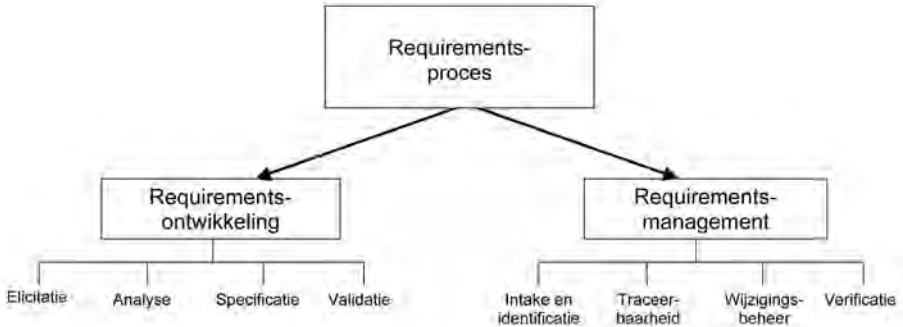
- functionele geschiktheid;
- beveiligbaarheid;
- betrouwbaarheid;
- bruikbaarheid;
- prestatie-efficiëntie;
- uitwisselbaarheid;
- onderhoudbaarheid;
- overdraagbaarheid.

25010 is verder uitgewerkt in hoofdstuk 4 en in bijlage A van dit boek.

Net als functionele requirements worden ook niet-functionele requirements tijdens het requirementsproces concreter. Op businessniveau kunnen de requirements geïntroduceerd worden door uitspraken van belanghebbenden als: 'Een goldcardlid moet sneller worden geholpen dan een gewoon lid'. Deze regels zijn nog niet meetbaar, maar worden tijdens het proces verder uitgewerkt totdat ze meetbaar zijn.

2.5 Het requirementsproces in hoofdlijnen

Het requirementsproces bestaat uit twee fasen: requirementsontwikkeling en requirementsmanagement. Iedere fase bestaat weer uit een aantal activiteiten.



Figuur 2.5: Onderverdeling van het requirementsproces

Het doel van requirementsontwikkeling is het juist en volledig vastleggen van de requirements. Requirementsontwikkeling omvat het eliciteren, analyseren en specificeren van de business-, gebruikers- en systeemrequirements. Deze activiteiten beginnen met het in kaart brengen van de belanghebbenden van het project. Vervolgens worden de behoefte, doelstelling, eisen, wensen en beperkingen geëliciteerd en geanalyseerd. Hierbij wordt ‘van boven naar beneden’ gewerkt: eerst worden de businessrequirements opgesteld, waarna deze worden uitgewerkt en geconcretiseerd in de gebruikersrequirements. Op basis hiervan worden de systeemrequirements opgesteld. In hoofdstuk 3 gaan we uitgebreid in op requirementsontwikkeling.

Op het moment dat de requirements beschikbaar zijn, wordt op ieder niveau van requirements een validatie uitgevoerd. Tijdens de validatie kijken de verschillende belanghebbenden gezamenlijk vanuit verschillende gezichtspunten, zoals correctheid, compleetheid en consistentie, naar de requirements. Hiervoor zijn meerdere technieken beschikbaar, zoals de individuele review, inspectie, walkthrough, prototyping en simulatie. Validatie is geen eenmalige activiteit. Het ontwikkelen van requirements heeft een iteratief karakter. Daarnaast veranderen requirements tijdens de ontwikkeling van het systeem regelmatig. Na iedere iteratie of wijziging dienen de requirements gevalideerd te worden. In hoofdstuk 5 gaan we uitgebreid in op requirementsvalidatie.

Requirementsmanagement heeft als doel dat de requirements juist en volledig worden verwerkt in het systeem. Om dit zeker te stellen dient in de eerste plaats de traceerbaarheid tussen requirements en tussenproducten bewaakt te worden. Daarnaast dienen wijzigingen in de requirements beheerst te worden. Tot slot dient geverifieerd te worden of de requirements juist en volledig zijn verwerkt in de tussenproducten die tijdens de realisatie worden gemaakt. Om traceerbaarheid, wijziging en verificatie te kunnen uitvoeren is het noodzakelijk om een intake op de requirements uit te voeren en ze uniek identificeerbaar te maken. In hoofdstuk 6 gaan we uitgebreid in op requirementsmanagement.

Trefwoordenlijst

A

acceptatiecriteria 19, 195
acceptatietestgevallen 72
actor 55, 195
adoptie 116
adoptiemeting 117
agile 58, 89
 ~ fasering 89
 ~ methoden 13
 ~ validatie 93
alternatief scenario 55
analyse 26, 27, 195
analyse bestaand systeem 40
analysetechnieken 45
atomair 69
attribuut 195
audit 87
automatische
 traceerbaarheidsrelaties 83

B

baseline 136
bedrijfsdoelstellingen 15
belanghebbenden 9, 30
beperkingen 16
bidirectionele traceerbaarheid 81, 195
boehm 112
brainstormen 42
business case 17, 101, 107
business-casedocument 17
businessrequirement 12, 17, 195
business rule 16, 195

C

CAB (Change Advisory Board) 195
capability Maturity Model
 Integration (CMMI) 147
CASE-tools 136
CFFP (COSMIC Full Function
 Points) 195
change advisory board 85
checklist 45
classificatie 46
CMMI (Capability Maturity Model
 Integration) 114, 147, 195

commercial-off-the-shelf (COTS)
 107
Concept of Operations (ConOps)
 146
correctheid 71
cosmic-functiepunt (CFP) 130
COSMIC-methode 130

D

databaserecordidentificatie 78
demand-supplyorganisatie 3
documentstudie 39
dynamische nummering 78

E

earned value 131
eenduidig 69
elicitatie 26, 27, 195
 ~technieken 39
entiteitrelatiediagram 50
ERD (Entiteitrelatiediagram) 50, 195

F

facilitator 41
foutsценario 55
FPA (Functiepunctanalyse) 196
FPAi 126
functiepunctanalyse 123
functioneel beheer 101
functionele requirements 15, 20, 196

G

gebruikersrequirements 12, 18, 196
gedetailleerde functiepunctentelling (FPA_d) 124, 127
geplande waarde 131
gerealiseerde waarde 131
gesloten interview 40
globale functiepunctentelling (FPA_g) 127
glossary 151
grammaticaal correct 70

H

hoofdsценario 55
horizontale prototype 43

I

IDEAL 109
identificatie 78
IEEE (Institute of Electrical and
 Electronics Engineers) 145, 196
identificatie 68
implementatiedetails 70
inspectie 72
implementatie 119
implementatiemodel 111
indicatieve functiepunctanalyse
 (FPA_i) 124, 126
indicatoren 61
inspectie 87, 196
intake 78
interactiematrix 46
interviews 40
ISO 9126 154
ISO 25010 20, 60
iteratieve methoden 13

K

kwaliteitsbewustwording 118
kwaliteitseigenschappen 60
kwaliteitseisen 16

M

managementreview 86
meetvoorschriften 61
metamodel 135
moderator 87

N

NESMA (NEderlandse Software
 Metrieken Associatie) 123, 196
niet-functionele requirements 15, 20, 59, 196

O

observaties 42
omvangsschatting 123

open interview 40
opstellen 140

P

planned value 131
PM-BOK 131
PRINCE2 152, 196
prioriteit 71
prioriteringstechnieken 61
problempatronen 47
problem frames 47
procesbeschrijvingen 187
procesverbetering 109
product backlog 90
project 196
projectrequirement 10, 196
proof of concept (PoC) 44, 141
prototypen 43, 73

Q

quality attribute workshop 61

R

RACI-matrix 187
Rational Unified Process (RUP) 149
Request for Change (RfC) 107
Request for Delivery (RfD) 107
Request for Information (RfI) 106
Request for Proposal (RfP) 107
requirement 9, 196
requirementsanalist 28, 33, 196
requirementsattributen 46, 105
requirementscategorieën 46
requirementslevenscyclusbeheer 99
requirementsmanagement 77, 196
requirementsmanagementplan (RMP) 150, 196
requirementsmanagementtooling 135
requirementsmetamodel 136
requirementsontwikkeling 25, 36
review 72
RfC (Request for Change) 197
RfD (Request for Delivery) 197
RfI (Request for Information) 197
RfP (Request for Proposal) 197
RUP (Rational Unified Process) 149, 197

S

SAS 70 4
SCD (Systeemcontextdiagram) 197
scenario's 43, 55
schattingsmethoden 123
schrijver 41
scope creep 136

S-curve 132
selectietraject 139
shortlist 140
simulatie 73
sjabloon 53, 163
sjabloon Systeemrequirements-specificatie (SRS) 177
sjabloon use-casespecificatie 173
sjabloon verbeterplan 181, 183
sjabloon visie en scope 164, 169
SMART 114, 197
Software Requirements Specification 146
SOX 4
specificatie 26, 29, 197
specificatietechnieken 48
SRS (Systeemrequirements-specificatie) 19, 146, 197
scenario's 73
standaardzinsamenstelling 53
statusregistratie 103
story board 59
storyboard prototyping 44
stroomschema 52
successscenarios 55
symbolische identificatie 78
systeemcontext 10
systeemcontextdiagram 49
systeemontwikkelmethoden 13
systeemrequirements 12, 19, 197
systeemrequirementsspecificatie (SRS) 19
System Requirements Specification (SyRS) 146

T

taakanalyse 43
technisch beheer 107
technische review 87
template 53
testbaar/verifieerbaar 70
toestandtransitiediagram (TTD) 51
TOGAF 160
toolbox 135
tools 110
traceerbaar 70
traceerbaarheid 80, 136
traceerbaarheidslijsten 83
traceerbaarheidsmatrix 82
traceerbaarheidstabellen 82
traceerbaarheid (Traceability) 197
TTD (Toestandtransitiediagram) 197

U

UCP (Use Case Point) 129, 197
UC (Use case) 197

uitbesteding 3
UML 51, 53
UML (Unified Modeling Language) 198
Unified Modelling Language (UML) 58
universum van discussie (UvD) 50
use-casediagram 57
use-casemodel 198
use casepunten 128
use cases 54, 59, 73
use-casespecificatie 19
user story 58

V

validatie 26, 29, 65, 198
validatietechnieken 71
verbeterplan 115
verboden woorden 74
verificatie 86, 198
versietraceerbaarheid 80, 102
verticale prototype 43
visie- en scopedocument 18
visueel brainstormen 43
voortgangsmeting 131

W

walkthrough 87, 73
watervalmethoden 13
Wedge-factor 2
wijzigingsbeheer 83, 136
workshops 41

'Voor wie met requirements wil beginnen is dit boek een mooi startpunt.'

*Meile Postuma – Bestuurslid Testnet,
Testadviseur Logica
Boekrecensie in Testnet Nieuws*

'Het is een gezamenlijk vertrekpunt voor opdrachtgevers en opdrachtnemers voor een succesvolle uitvoering van een project.'

*Simon Bos – redacteur IT-Infra en IT Interim-
manager Bos+Cohen IT Management
Recensie in IT Beheer*

'Het helder en uiterst deskundig geschreven boek gaat in op gangbare (technische) standaarden in de IT-wereld en geeft talloze tips en aandachtspunten voor het managen van de werkprocessen rondom specificaties.'

*Mr. Peter van Schelven
Recensent namens NBD Bibliotheek*



Martin, Jan Jaap, Arno, Serge, Petra en Johan.

De kwaliteit van requirements is essentieel voor het succes van een project. Onderschatting ervan leidt juist tot verlies van tijd, geld en kwaliteit. Actuele ontwikkelingen zoals de trend naar een demand-supplyorganisatie, outsourcing, offshoring en hogere eisen aan compliance, rechtvaardigen meer aandacht voor requirements.

Het belang van requirements en een gemeenschappelijk begrip van basisprincipes en requirementsprocessen wordt gelukkig meer en meer onderkend, aan zowel business- als ICT-kant. *Succes met de requirements!* levert hierin een belangrijke bijdrage en in de Nederlandstalige markt is dit boek inmiddels uitgegroeid tot standaardwerk over het thema. Het boek helpt bij het overbruggen van de kloof tussen business en ICT en bevordert de noodzakelijke aandacht vanuit beheerafdelingen voor requirements. Voldoende reden voor een derde druk dus!

Ook deze derde druk van *Succes met de requirements!* beschrijft een toegankelijke en toepasbare aanpak voor het ontwikkelen en beheren van requirements en hun levenscyclus. Daarnaast komen gerelateerde onderwerpen aan de orde zoals het verbeteren van requirementsprocessen, omvangsschatting met behulp van requirements, requirementstooling en prioriteringstechnieken. Uiteraard hebben de auteurs deze editie geheel aan de veranderingen in de praktijk aangepast, zoals agile systeemontwikkeling. Ook hebben ze alle normen en referenties bijgewerkt aan de laatste stand van zaken.

Dit boek levert een gemeenschappelijke basis voor het succesvol uitvoeren van projecten, voor iedereen die vanuit business of ICT betrokken is bij projecten, zoals opdrachtgevers, informatiemanagers, functioneel beheerders, requirementsanalisten, projectmanagers, operationele managers, ontwikkelaars, testers en QA-medewerkers.

sysQa
proud of it



9 789012 584883

ISBN 978 90 12 58488 3

NUR 982



ACADEMIC
SERVICE

www.academic-service.nl