Certified Professional for Requirements Engineering

Foundation Level

Syllabus

Stan Bühne
Martin Glinz
Hans van Loenhoud
Stefan Staal

Gebruiksrecht

1. Individuen en opleidingsinstituten mogen deze syllabus gebruiken als basis voor trainingen, onder voorwaarde dat het copyright wordt erkend en is vermeld in het trainingsmateriaal. Voor het gebruiken van deze syllabus in advertenties is schriftelijke toestemming nodig van IREB e.V..
2. Elk individu of een groep van individuen mag deze syllabus gebruiken als basis voor artikelen, boeken of andere publicaties, onder voorwaarde dat het copyright van de auteurs en IREB e.V. als de bron en eigenaar van dit document wordt erkend in dergelijke publicaties.

Woord van dank

De eerste versie van deze syllabus werd in 2007 geschreven door Karol Frühauf, Emmerich Fuchs, Martin Glinz, Rainer Grau, Colin Hood, Frank Houdek, Peter Hruschka, Barbara Paech, Klaus Pohl en Chris Rupp. Zij werden hierbij gesteund door Ian Alexander, Joseph Bruder, Samuel Fricker, Günter Halmans, Peter Jaeschke, Sven Krause, Steffen Lentz, Urte Pautz, Suzanne Robertson, Dirk Schüpferling, Johannes Staub, Thorsten Weyer en Joy Beatty.

Versie 3 is een grote revisie, uitgevoerd door Stan Bühne, Martin Glinz, Hans van Loenhoud en Stefan Staal. Zij werden hierbij gesteund door Karol Frühauf, Rainer Grau, Kim Lauenroth, Chris Rupp en Camille Salinesi.

Tijdens deze revisie werd terugkoppeling gegeven door Xavier Franch, Karol Frühauf, Rainer Grau, Frank Houdek en Thorsten Weyer. Aanvullende terugkoppeling werd gegeven door Wim Decoutere en Hans-Jörg Steffe.

Reviews werden uitgevoerd door Christoph Ebert, Barbara Paech en Chris Rupp.

De syllabus werd goedgekeurd voor publicatie op 22 juli 2020 door de IREB Council op voorstel van Xavier Franch en Frank Houdek. Deze Nederlandse vertaling werd verzorgd door Jan Jaap Cannegieter, Wim Decoutere, Piet de Roo en Johan Zandhuis, met reviews van Hans van Loenhoud, Michiel van der Voort en Sven van der Zee.

We bedanken iedereen voor zijn of haar zeer gewaardeerde bijdrage.

Voorwoord

In de zomer van 2017 is door IREB een onderzoek gedaan naar de relevantie van de bestaande Certified Professional for Requirements Engineering (CPRE) Foundation Level certificering (versie 2.2). Het doel van het onderzoek was om een terugkoppeling te krijgen

over de praktische marktrelevantie van de certificering vanuit het perspectief van de aanbieders van opleidingen en vanuit het perspectief van gecertificeerde CPRE'ers die als Requirementsanalisten werken [MFeA2019]. Uit het onderzoek bleek dat de bestaande syllabus versie 2.2 van de CPRE Foundation nog steeds grotendeels voldoet aan de belangrijkste behoeften van de markt en dat de kandidaten de relevante RE-kennis leren. Desalnietemin kreeg IREB ook een terugkoppeling dat verschillende technieken in de praktijk niet meer worden toegepast, terwijl andere technieken – in de zoektocht naar een meer iteratieve en adaptieve ontwikkeling – ontbraken. Deze terugkoppeling was in lijn met de eigen perceptie van IREB met betrekking tot de veranderingen op het gebied van Requirements Engineering (RE). Daarom heeft IREB besloten om de syllabus van CPRE Foundation Level grondig te herzien, waarbij verouderde inhoud is verwijderd en nieuwe elementen zijn toegevoegd. Versie 3 van de syllabus sluit aan bij de stand van zaken in de hedendaagse Requirements Engineering praktijk met betrekking tot zowel een planmatige aanpak als een agile aanpak voor het specificeren en beheren van de requirements.

Van kandidaten die op basis van deze syllabus CPRE-certificering wensen, wordt verwacht dat zij enige basiskennis hebben van systeemontwikkeling met een planmatige en agile aanpak.

Doel van het document

Deze syllabus definieert het basisniveau (Foundation level) van het "Certified Professional for Requirements Engineering"-certificaat van de International Requirements Engineering Board (IREB). De syllabus biedt opleidingsinstituten een basis voor het ontwikkelen van trainingsmateriaal. Cursisten kunnen deze syllabus gebruiken om zich voor te bereiden op het examen.

Inhoud van de syllabus

Het Foundation Level is gericht op alle personen die bij Requirements Engineering betrokken zijn. Dit omvat personen in rollen zoals requirementsanalist, business- of systeemanalist, product owner of -manager, ontwikkelaar, project- of IT-manager of domeinexpert.

In deze syllabus en het bijbehorende handboek wordt de afkorting "RE" gebruikt voor Requirements Engineering.

Scope van de syllabus

Het CPRE Foundation Level reikt basisprincipes aan die voor elk type systeem (bijvoorbeeld mobiele apps, informatiesystemen of cyber-fysieke systemen) gelden. Bovendien gaat het CPRE Foundation Level niet uit van één specifiek ontwikkelproces en is het niet gericht op één specifiek toepassingsdomein. Opleidingsaanbieders mogen opleidingen aanbieden die gericht zijn op specifieke soorten systemen, processen of toepassingsdomeinen, zolang de leerdoelstellingen van deze syllabus maar volledig gedekt zijn.

Mate van detail

Het detailleringsniveau van deze syllabus maakt het mogelijk trainingen en examinering internationaal consistent te houden. Om dit doel te bereiken bevat deze syllabus:

- Algemene leerdoelen
- Inhoud die de leerdoelen verder beschrijft
- Referenties naar aanvullende literatuur (waar nodig)

Leerdoelstellingen/Cognitieve Niveaus

Alle modules en leerdoelen in deze syllabus zijn op een cognitief niveau ingedeeld. De volgende niveaus worden gebruikt:

- **L1: Weten** (beschrijven, opsommen, karakteriseren, herkennen, benoemen, onthouden, ...) – het onthouden of kunnen terughalen van eerder geleerde stof.
- **L2: Begrijpen** (verklaren, interpreteren, aanvullen, samenvatten, verantwoorden, classificeren, vergelijken, ...) – betekenis begrijpen of kunnen construeren met gegeven informatie of in een gegeven situatie.
- **L3: Toepassen** (specificeren, schrijven, ontwerpen, ontwikkelen, implementeren, ...) – kennis en vaardigheden kunnen toepassen in een gegeven situatie.

Hogere niveaus omvatten de lagere niveaus. Merk op dat alle termen in de verklarende woordenlijst die worden aangeduid als fundamentele termen bekend moeten zijn (L1), zelfs als ze niet expliciet worden vermeld in de leerdoelstellingen. De verklarende woordenlijst kan worden gedownload via de homepagina van IREB op het volgende adres https://www.ireb.org/downloads/#cpre-glossary

Structuur van de syllabus

De syllabus bestaat uit 7 hoofdstukken. Elk hoofdstuk behandelt één leereenheid (LE). Elke hoofdstuktitel bevat het cognitieve niveau van het hoofdstuk, dat wordt bepaald door het hoogst voorkomende niveau van de paragrafen. Daarnaast is de minimale trainingstijd aangegeven die men geacht wordt te besteden aan dit hoofdstuk. Opleidingsinstituten zijn vrij om meer tijd te besteden, maar moeten de verhoudingen tussen de leereenheden (LE's) handhaven. Belangrijke termen die in een hoofdstuk worden gebruikt zijn vermeld aan het begin van het hoofdstuk.

Voorbeeld

LE 4	Het uitwerken van requirements (L3)
Duur:	4 uur 30 minuten
Termen:	Requirementsbron, systeemgrens, systeemcontext, requirementselicitatie, requirementsonderhandeling, requirementsvalidatie, belanghebbende, Kano-model, conflict

Dit voorbeeld laat zien aan dat hoofdstuk 4 leerdoelen op niveau L3 bevat en dat vier en een half uur bedoeld zijn voor het onderwijzen van het materiaal in dit hoofdstuk.

Elk hoofdstuk bevat subhoofdstukken. In de titels van de paragrafen is ook het cognitieve niveau van de inhoud opgenomen.

Leerdoelen (LD) staan opgesomd voorafgaand aan de daadwerkelijke tekst. De nummering geeft aan bij welk subhoofdstuk ze horen. Zo wordt leerdoel LD 4.2.1 beschreven in subhoofdstuk 4.2.

Volgorde van de onderwerpen in de syllabus

De volgorde van de hoofdstukken in deze syllabus vormt een logische volgorde van onderwerpen. De onderwerpen hoeven echter niet in deze volgorde te worden behandeld. Het staat de aanbieders van opleidingen vrij om het materiaal in elke willekeurige volgorde te onderwijzen (inclusief het verweven van onderwerpen uit verschillende leereenheden) die zij passend achten in het kader van hun opleiding en die past bij hun didactische concepten.

Het examen

Deze syllabus vormt de basis voor examinering ten behoeve van het CPRE Foundation Level certificaat.

Een vraag in het examen kan betrekking hebben op verschillende hoofdstukken uit de syllabus. Alle hoofdstukken (LE 1 tot LE 7) van de syllabus kunnen worden geëxamineerd.

De vraagstelling van het examen is multiple-choice.

Examens kunnen direct na een training worden afgenomen, maar ook separaat (bijvoorbeeld bij een exameninstituut). Een lijst van de door het IREB erkende exameninstituten is te vinden op de website https://www.ireb.org.

Versiebeheer

Versie	Datum	Toelichting
3.0.0	Februari 2021	Deze tweede grote update sluit aan bij de stand van zaken in de hedendaagse Requirements Engineering praktijk met betrekking tot zowel een planmatige aanpak als een agile aanpak voor het specificeren en beheren van de requirements.
3.1.0	December 2022	Typefouten en referenties zijn verbeterd in het hele document om de leesbaarheid te verhogen. LE 1: ▪ Leerdoel LD 1.3.2 is verplaatst naar LD 1.1.2. ▪ LD 1.2.2 en 1.3.1 zijn bijgewerkt. LE 3: ▪ Leerdoel LD 3.1.3. is bijgewerkt. ▪ LD 3.1.2: paragraaf over abstractieniveaus is herschreven. ▪ LD 3.4: Alle modeltypes die niet toegepast moeten kunnen worden voor Foundation Level zijn naar een nieuw subhoofdstuk 3.4.6 verplaatst. ▪ LD 3.6: Naamgeving hoofdstuk aangepast naar "Requirementsdocumenten en Requirementsdocumentatiestructuren". LE 4: ▪ LD 4.1: De beschrijving van belanghebbenden en belanghebbendenrollen is aangescherpt. ▪ LD 4.2: De invoering en verantwoording voor ontwerp- en ideegenererende technieken is aangescherpt. ▪ LD 4.3: Naamgeving hoofdstuk aangepast naar "Het oplossen van requirementsconflicten" LE 6: ▪ LD 6.3.1 en LD 6.5.2 zijn beperkt aangepast.
3.1.1	Januari 2024	Nieuwe huisstijl

Content

1 Inleiding en overzicht van Requirements Engineering (L2)

Doel: Weten waar RE over gaat en de waarde van RE begrijpen

Duur: 1 uur

Termen: Requirement, requirementsspecificatie, Requirement Engineering (RE), belanghebbende, systeem, Requirementsanalist

Leerdoelen

LD 1.1.1 De basisterminologie kennen (L1)
LD 1.1.2 De verschillende soorten requirements begrijpen (L2)
LD 1.2.1 De waarde van RE kunnen uitleggen (L2)
LD 1.2.2 De symptomen van ontoereikende RE kunnen opsommen (L1)
LD 1.3.1 Weten waar RE kan worden toegepast en waar eisen worden gesteld (L1)
LD 1.4.1 De belangrijkste taken van RE kennen en weten dat een RE-proces op maat moet worden gemaakt om deze uit te voeren (L1)
LD 1.5.1 De kenmerken kunnen geven van de rol en taken van een Requirementsanalist (L1)
LD 1.6.1 Weten wat een Requirementsanalist moet leren (L1)

1.1 Requirements Engineering: Wat (L1)

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

Dingen die moeten worden gebouwd of doorontwikkeld kunnen 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

Al deze zaken kunnen worden beschouwd als *systemen*. In deze syllabus gebruiken we de term *systeem* om alle zaken aan te duiden waarvoor *belanghebbenden* requirements hebben. *Belanghebbenden* zijn personen of organisaties die invloed hebben op de requirements die aan een systeem worden gesteld of die door dat systeem worden beïnvloed.

Het doel van RE is om de requirements voor systemen zodanig te specificeren en te beheren dat de geïmplementeerde en opgeleverde systemen voldoen aan de wensen en behoeften van de belanghebbenden.

In RE maken we onderscheid tussen drie soorten requirements [Glin2020]:

- *Functionele requirements* hebben betrekking op een resultaat of gedrag dat door een functie van een systeem moet worden geleverd. Dit omvat requirements voor gegevens of de interactie van een systeem met zijn omgeving.

- *Kwaliteitsrequirements* hebben betrekking op kwaliteitskwesties die niet onder de functionele requirements vallen, zoals performance, beschikbaarheid, beveiliging of betrouwbaarheid.
- *Beperkingen* zijn requirements die de oplossingsruimte begrenzen binnen datgene wat nodig is om aan de gegeven functionele requirements en kwaliteitsrequirements te voldoen.

1.2 Requirements Engineering: Waarom (L2)

Adequate RE voegt *waarde* toe in het proces van het ontwikkelen en evolueren van een systeem:

- Vermindering van het risico om het verkeerde systeem te ontwikkelen
- Beter begrip van het probleem
- Basis voor de raming van de ontwikkelinspanning en -kosten
- Noodzakelijke randvoorwaarde voor het testen van het systeem

Typische symptomen van ontoereikende RE zijn ontbrekende, onduidelijke of onjuiste requirements. Dit is vooral te wijten aan:

- Haast om te beginnen met de bouw van het systeem
- Communicatieproblemen tussen betrokken partijen
- De veronderstelling dat de requirements voor zichzelf spreken
- Ontoereikende RE-opleiding en -vaardigheden

1.3 Requirements Engineering: Waar (L2)

RE kan worden toegepast op requirements voor elk soort systeem. Vandaag de dag wordt RE vooral toegepast voor systemen waarin software een grote rol speelt. Dergelijke systemen bestaan doorgaans uit softwarecomponenten, fysieke elementen en organisatorische elementen.

Requirements kunnen voorkomen als:

- *Systeemrequirements* – wat een systeem moet doen
- *Belanghebbendenrequirements* – wat de belanghebbenden willen vanuit hun perspectief
- *Gebruikersrequirements* – wat gebruikers willen vanuit hun perspectief
- *Domeinrequirements* – vereiste domeineigenschappen
- *Business requirements* – bedrijfsdoelen, doelstellingen en behoeften van een organisatie

1.4 Requirements Engineering: Hoe (L1)

De belangrijkste taken in RE zijn het eliciteren (4.2), documenteren (3), valideren (4.4) en beheren (6) van requirements. Tool-ondersteuning (7) kan helpen bij het uitvoeren van deze taken. Requirementsanalyse en -conflictoplossing (4.3) worden beschouwd als onderdeel

van de elicitatie. Om RE-activiteiten goed te kunnen uitvoeren, moet een geschikt RE-proces worden samengesteld uit een breed scala aan mogelijkheden (5).

1.5 De rol en taken van een Requirementsanalist (L1)

Requirementsanalist is meestal geen functieomschrijving, maar een *rol* die mensen vervullen, die:

- Requirements eliciteren, documenteren, valideren en/of beheren als onderdeel van hun taken.
- Diepgaande kennis van RE hebben.
- De kloof tussen het probleem en potentiële oplossingen kunnen overbruggen.

In de praktijk vervullen business analisten, applicatiespecialisten, product owners, systeemengineers en zelfs ontwikkelaars de rol van een Requirementsanalist.

1.6 Wat te leren over Requirements Engineering (L1)

Deze syllabus behandelt de basisvaardigheden die een Requirementsanalist onder de knie moet krijgen. Het omvat de basisprincipes van RE (2), het documenteren van requirements in verschillende vormen (3), de verschillende manieren om requirements uit te werken (4), het definiëren van en werken met geschikte RE-processen (5), het beheren van bestaande requirements (6) en het gebruik van ondersteunende tools (7).

2 Basisprincipes van Requirements Engineering (L2)

Doel: De principes van RE kennen en begrijpen

Duur: 1 uur 30 minuten

Termen: Context, requirement, Requirements Engineering (RE), belanghebbende, gedeeld begrip, validatie

Leerdoelen

LD 2.1.1 De principes van RE kunnen opsommen (L1)
LD 2.2.1 De voorwaarden kennen die verbonden zijn aan de principes (L1)
LD 2.2.2 De principes kunnen uitleggen en waarom zij belangrijk zijn (L2)

2.1 Overzicht van de principes (L1)

RE wordt beheerst door een aantal basisprincipes, die van toepassing zijn op alle taken, activiteiten en werkwijzen binnen RE. De volgende negen principes vormen de basis voor de toepassingen die in de volgende hoofdstukken van deze syllabus worden behandeld.

1. Waarde-oriëntatie: Requirements zijn een middel om een doel te bereiken, niet een doel op zich
2. Belanghebbenden: RE gaat over het voldoen aan de wensen en behoeften van de belanghebbenden
3. Gedeeld begrip: Succesvolle systeemontwikkeling is onmogelijk zonder een gemeenschappelijke basis
4. Context: Systemen kunnen niet los van hun omgeving worden gezien
5. Probleem – Requirement – Oplossing: Een onlosmakelijk met elkaar verweven trio
6. Validatie: Niet-gevalideerde requirements zijn nutteloos
7. Evolutie: Veranderende requirements zijn geen uitzondering, maar normaal
8. Innovatie: Meer van hetzelfde is niet genoeg
9. Systematisch en gedisciplineerd werken: We kunnen niet zonder in RE

2.2 De principes uitgelegd (L2)

Principe 1 - Waarde-oriëntatie: Requirements zijn een middel om een doel te bereiken, geen doel op zich

De waarde van een requirement is gelijk aan het voordeel ervan minus de kosten voor het eliciteren, documenteren, valideren en beheren van het requirement. Het voordeel van een requirement is de mate waarin het bijdraagt aan:

- Het bouwen van systemen die voldoen aan de wensen en behoeften van de belanghebbenden.
- Het verminderen van het risico op falen en dure aanpassingen tijdens de ontwikkeling van het systeem.

Principe 2 - Belanghebbenden: RE gaat over het voldoen aan de wensen en behoeften van de belanghebbenden

Aangezien het bij RE gaat om het begrijpen van de wensen en behoeften van de belanghebbenden, is een goede omgang met de belanghebbenden een kerntaak van RE. Elke belanghebbende heeft een rol in de context van het te bouwen systeem – bijvoorbeeld de gebruiker, de klant, de afnemer, de exploitant of de toezichthouder. Een belanghebbende kan ook meer dan één rol tegelijk hebben. Voor belanghebbendenrollen met te veel vertegenwoordigers of wanneer de vertegenwoordigers onbekend zijn, kunnen fictieve archetypische beschrijvingen, bekend als *persona's,* worden gedefinieerd ter vervanging. Het is niet voldoende om alleen rekening te houden met de requirements van eindgebruikers of klanten. Als we dit zouden doen, zouden we de kritische requirements van andere belanghebbenden kunnen missen. Gebruikers die een terugkoppeling geven over een in gebruik zijnde systeem moeten ook worden beschouwd als belanghebbenden.

Belanghebbenden kunnen verschillende behoeften en standpunten hebben, wat kan leiden tot tegenstrijdige requirements. Het is een taak van RE om dergelijke conflicten te identificeren en op te lossen.

Het betrekken van de juiste mensen in de relevante belanghebbendenrollen is cruciaal voor succesvolle RE. Toepassingen voor het identificeren, prioriteren en werken met belanghebbenden worden beschreven in 4.

Principe 3 - Gedeeld begrip: Succesvolle systeemontwikkeling is onmogelijk zonder een gemeenschappelijke basis

RE creëert, bevordert en verzekert een gedeeld begrip tussen en binnen de betrokken partijen: belanghebbenden, Requirementsanalisten en ontwikkelaars. Er zijn twee vormen van gedeeld begrip:

- *Expliciet gedeeld begrip* (bereikt door gedocumenteerde en overeengekomen requirements)
- *Impliciet gedeeld begrip* (gebaseerd op gedeelde kennis over requirements, visies, context, et cetera)

Domeinkennis, eerdere succesvolle samenwerking, gedeelde cultuur en waarden, en wederzijds vertrouwen maken gedeeld begrip mogelijk, terwijl geografische afstand, outsourcing of grote teams met een hoog verloop een obstakel vormen.

Bewezen werkwijzen om tot een gedeeld begrip te komen zijn onder meer het opstellen van verklarende woordenlijsten (3.5), het maken van prototypes (3.7) of het gebruik van een bestaand systeem als referentiepunt. Voor het beoordelen van gedeeld inzicht zijn voorbeelden van verwachte resultaten, het onderzoeken van prototypes of het schatten van de kosten van de implementatie van een requirement toepasbaar. De belangrijkste manier om de impact van misverstanden te verminderen is het gebruik van een proces met korte terugkoppelingslussen (5).

Principe 4 - Context: Systemen kunnen niet los hun omgeving worden gezien

Systemen zijn ingebed in een *context*. Zonder die context te begrijpen is het onmogelijk om een systeem goed te specificeren. In RE is de context van een systeem het deel van de omgeving van een systeem dat relevant is voor het begrijpen van het systeem en zijn requirements. De *systeemgrens* is de grens tussen een systeem en de omringende context. In eerste instantie is de systeemgrens vaak niet duidelijk en deze kan in de loop van de tijd zelfs veranderen.

Het verduidelijken van de systeemgrens en het definiëren van de externe interfaces tussen het systeem en de elementen van de context waarmee het in wisselwerking staat, zijn echte RE-taken. Tegelijkertijd moet de *beschouwingsgebied* van het systeem – dat wil zeggen het scala aan zaken dat bij de ontwikkeling van het systeem kan worden vormgegeven en ontworpen – worden bepaald. We moeten ook rekening houden met de zogenaamde *contextgrens* die het RE-relevante deel van de omgeving van een systeem scheidt van de rest van de wereld.

Bij het specificeren van een systeem is het niet voldoende om alleen rekening te houden met de requirements binnen de systeemgrens. RE moet ook beschouwen:

- Veranderingen in de context die van invloed kunnen zijn op de systeemrequirements.
- Requirements vanuit de praktijk die relevant zijn voor het systeem (en hoe deze af te beelden op de systeemrequirements).
- Aannames over de context die moeten opgaan om het systeem te laten werken en te voldoen aan de requirements vanuit de omgeving.

Principe 5 - Probleem - Requirement - Oplossing: Een onlosmakelijk met elkaar verweven trio

Een *probleem* doet zich voor wanneer belanghebbenden niet tevreden zijn met de situatie zoals die is. De *requirements* leggen vast wat de belanghebbenden nodig hebben om van het probleem af te komen of om het te vereenvoudigen. Een socio-technisch systeem dat aan deze requirements voldoet, vormt een *oplossing*.

Problemen, requirements en oplossingen komen niet noodzakelijkerwijs in deze volgorde voor. Oplossingsideeën kunnen gebruikersbehoeften creëren die als requirements moeten worden uitgewerkt en worden geïmplementeerd in een daadwerkelijke oplossing. Dit is typisch het geval bij innovatie.

- Problemen, requirements en oplossingen zijn nauw met elkaar verweven: ze kunnen niet los van elkaar worden aangepakt.
- Desalniettemin proberen Requirementsanalisten problemen, requirements en oplossingen zoveel mogelijk van elkaar te scheiden bij het denken, communiceren en documenteren. Deze scheiding van aandachtspunten maakt RE-taken beter beheersbaar.

Principe 6 - Validatie: Niet-gevalideerde requirements zijn nutteloos

Uiteindelijk moeten we controleren dat het ingezette systeem voldoet aan de wensen en behoeften van de belanghebbenden. Om het risico van ontevreden belanghebbenden vanaf het begin te beperken, moet de validatie van de requirements al tijdens RE beginnen. We moeten controleren of:

- Over de requirements overeenstemming is bereikt tussen de belanghebbenden,
- De wensen en behoeften van de belanghebbenden voldoende worden afgedekt door de requirements,
- De aannames over de context (zie Principe 4 hierboven) redelijk zijn.

Manieren om requirements te valideren worden besproken in 4.4.

Principe 7 - Evolutie: Veranderende requirements zijn geen toeval, maar normaal

Systemen en hun requirements zijn onderhevig aan *evolutie*. Dit betekent dat ze voortdurend veranderen. Verzoeken tot wijziging van een requirement of van een set van requirements voor een systeem kunnen bijvoorbeeld worden veroorzaakt door:

- Veranderde bedrijfsprocessen
- Concurrenten die nieuwe producten of diensten lanceren
- Klanten die hun prioriteiten of meningen veranderen
- Veranderingen in de technologie
- Veranderingen in wet- of regelgeving
- Terugkoppeling van systeemgebruikers die vragen om nieuwe of gewijzigde functies

Bovendien kunnen requirements veranderen als gevolg van terugkoppeling van belanghebbenden bij het valideren van requirements, als gevolg van het opsporen van fouten in eerder geëliciteerde requirements of als gevolg van gewijzigde behoeften.

Als gevolg daarvan moeten Requirementsanalisten twee ogenschijnlijk tegenstrijdige doelen nastreven:

- Toelaten dat requirements wijzigen
- Requirements stabiel houden

Details over hoe dit kan worden bereikt worden besproken in 6.7.

Principe 8 - Innovatie: Meer van hetzelfde is niet genoeg

Door de belanghebbenden precies te geven wat ze willen, wordt de kans gemist om systemen te bouwen die nog beter voldoen aan de behoeften van de belanghebbenden dan ze zelf verwachten. Goede RE streeft er niet alleen naar om de belanghebbenden tevreden te stellen, maar ook om ze gelukkig te maken, te enthousiasmeren of zich veilig doen voelen. Dat is waar het uiteindelijk om gaat bij innovatie.

RE geeft vorm aan innovatieve systemen:

- Op kleine schaal door te streven naar pakkende nieuwe functies en gebruiksgemak.
- Op grote schaal door te streven naar disruptieve nieuwe ideeën.

In 4.2 worden verschillende technieken besproken om innovatie in RE te bevorderen.

Principe 9 - Systematisch en gedisciplineerd werken: We kunnen niet zonder in RE

We moeten geschikte processen en werkwijzen gebruiken om systematisch requirements te eliciteren, te documenteren, te valideren en te beheren, ongeacht het ontwikkelproces dat wordt gebruikt. Zelfs wanneer een systeem op een ad-hocmanier wordt ontwikkeld, verbetert een systematische en gedisciplineerde RE aanpak de kwaliteit van het uiteindelijke systeem.

Er bestaat geen enkel proces of werkwijze binnen RE dat bruikbaar is in iedere situatie of zelfs in de meeste situaties. Systematisch en gedisciplineerd werken betekent dat Requirementsanalisten:

- Hun processen en werkwijzen aanpassen aan het gegeven probleem, de context en de omgeving.
- Niet altijd hetzelfde proces en dezelfde werkwijzen gebruiken.
- Niet klakkeloos succesvolle processen en werkwijzen uit het verleden hergebruiken.

Voor elke RE-inspanning moeten processen, werkwijzen en werkproducten worden gekozen die het best passen bij de specifieke situatie. De details zijn uitgewerkt in 3, 4, 5 en 6.

3 Werkproducten en manieren van documenteren (L3)

Doel: De fundamentele rol van werkproducten in RE begrijpen en werkproducten kunnen creëren

Duur: 7 uur

Termen: Werkproduct, werkproducten op basis van natuurlijke taal, werkproducten op basis van sjablonen, werkproducten op basis van modellen, verklarende woordenlijst, kwaliteitscriteria, requirementsspecificatie

Leerdoelen

LD 3.1.1 De kenmerken kennen van RE-werkproducten en veelgebruikte soorten werkproducten kunnen opsommen (L1)

LD 3.1.2 Weten waarvoor elk werkproduct kan worden gebruikt en wat de levensduur van werkproducten is (L1)

LD 3.1.3 De verschillende abstractieniveaus voor requirements uit kunnen leggen, inclusief de wijze waarop passende abstractie- en detailniveaus kunnen worden gekozen (L2)

LD 3.1.4 De aspecten kennen die in aanmerking moeten worden genomen bij werkproducten en de onderlinge relaties tussen deze aspecten (L1)

LD 3.1.5 De algemene documentatierichtlijnen kunnen benoemen (L1)

LD 3.1.6 Beschrijven waarom het de moeite waard is om te plannen welke werkproducten gebruikt gaan worden (L1)

LD 3.2.1 De werkproducten op basis van natuurlijke taal kennen en hun voor- en nadelen kunnen geven (L1)

LD 3.2.2 De regels voor het schrijven van goede requirements op basis van natuurlijke taal kunnen uitleggen (L2)

LD 3.3.1 De categorieën kennen van de werkproducten op basis van sjablonen en hun voor- en nadelen (L1)

LD 3.3.2 Een individueel requirement en een user story specificeren met behulp van een zinsjabloon (L3)

LD 3.3.3 Een use case specificeren met behulp van een formuliersjabloon (L3)

LD 3.4.1 De rol, het doel en het gebruik van modellen in RE begrijpen (L2)

LD 3.4.2 De voordelen en beperkingen van het modelleren in RE begrijpen (L2)

LD 3.4.3 De volgende termen kennen: model, modelleertaal, activiteitenmodel, activiteitendiagram, klassenmodel, klassendiagram, contextmodel, contextdiagram, domeinmodel, doelmodel, interactiemodel, procesmodel, sequentiediagram, toestandsdiagram, toestandsmachine, toestandsmachinediagram, use case, use case diagram (L1)

LD 3.4.4 Begrijpen hoe een geschikt modeltype voor het specificeren van requirements in een bepaalde situatie wordt gekozen (L2)

LD 3.4.5 Begrijpen en interpreteren van eenvoudige modellen, indien van toepassing geschreven in UML, van de volgende typen: contextmodellen, use cases en use case diagrammen, domeinmodellen, klassenmodellen, activiteitenmodellen, procesmodellen en toestandsdiagrammen (L2)

LD 3.4.6 Een eenvoudig model van de gegevens van een systeem of de objecten in een domein met behulp van een UML-klassendiagram specificeren (L3)

LD 3.4.7 Een eenvoudige systeemfunctie of business proces met behulp van een UML-activiteitendiagram specificeren (L3)

LD 3.5.1 Uitleggen wat het doel is van verklarende woordenlijsten en hoe die moeten worden gecreëerd (L2)
LD 3.6.1 Veelgebruikte requirementsspecificatiedocumenten kennen (L1)
LD 3.6.2 Uitleggen welke documentstructuren welk doel dienen en wat de criteria zijn voor structurering (L2)
LD 3.7.1 De verschillende soorten prototypen kennen en weten waarvoor deze worden gebruikt (L1)
LD 3.8.1 De kwaliteitscriteria voor individuele requirements kennen (L1)
LD 3.8.2 De kwaliteitscriteria voor werkproducten kennen (L1)

3.1 Werkproducten in Requirements Engineering (L2)

Een werkproduct is een vastgelegd tussen- of eindresultaat dat in een werkproces is gegenereerd. Er bestaan verschillende werkproducten in RE, van, bijvoorbeeld, kortstondige schetsen en groeiende verzamelingen van user stories tot formeel gereleasede contractuele documenten met honderden pagina's requirements specificaties.

3.1.1 Kenmerken van werkproducten (L1)

Werkproducten worden gekenmerkt door hun doel, voorstellingswijze, grootte en levensduur. De volgende werkproducten komen vaak voor in de praktijk voor de gegeven doeleinden. Merk op dat een werkproduct andere werkproducten kan bevatten.

- Werkproducten voor één enkele requirement omvatten individuele requirements en user stories.
- Werkproducten voor een set van requirements omvatten use cases, grafische modellen van een bepaald type (3.4), taakbeschrijvingen, externe interfacebeschrijvingen en epics.
- Werkproducten die uitgebreide documenten of documentatiestructuren vormen, zijn onder meer specificaties van systeemrequirements, product- en sprintbacklogs, en story maps.
- Andere werkproducten zijn onder andere verklarende woordenlijsten, geschreven aantekeningen, schetsen en prototypes.

Werkproducten kunnen in verschillende vormen worden *weergegeven*:

- Op basis van natuurlijke taal (3.2)
- Op basis van sjablonen (3.3)
- Op basis van modellen (3.4)
- Andere voorstellingswijzen, zoals tekeningen of prototypes (3.7)

De meeste werkproducten worden elektronisch *opgeslagen* als bestanden, in databases, of in RE-tools. Informele, tijdelijke werkproducten kunnen ook op andere media worden opgeslagen – bijvoorbeeld op papier of op post-it notes op een Kanban-bord.

Als we kijken naar de levensduur van werkproducten, onderscheiden we drie categorieën:

- *Tijdelijke werkproducten:* gecreëerd ter ondersteuning van de communicatie en het creëren van gedeeld begrip.

- *Evoluerende werkproducten:* groeien in verschillende iteraties in de loop der tijd; hebben een beperkte set metagegevens nodig (6.5), wijzigingsbeheer kan van toepassing zijn.
- *Duurzame werkproducten:* zijn *gebaselined* of *gereleased*; hebben een volledige set metagegevens nodig (6.5); het wijzigingsproces moet worden gevolgd (6.3, 6.4).

Een tijdelijk werkproduct kan worden omgezet in een evoluerend werkproduct (door het te bewaren en er metagegevens aan toe te voegen). Ook kan een tijdelijk of evoluerend werkproduct duurzaam worden gemaakt door het te baselinen of te releasen.

3.1.2 Abstractieniveaus (L2)

Requirements bestaan meestal op veel *verschillende abstractieniveaus* – van, bijvoorbeeld, requirements op hoog niveau voor een nieuw business proces tot requirements op een zeer gedetailleerd niveau, zoals de reactie van een specifieke softwarecomponent op een uitzonderlijke gebeurtenis.

De keuze van het juiste abstractieniveau hangt af van het te specificeren onderwerp en van het doel van de specificatie. Het is echter belangrijk om geen requirements, die op verschillende abstractieniveaus liggen, door elkaar te gebruiken. In kleine en middelgrote werkproducten moeten de requirements op min of meer hetzelfde abstractieniveau liggen.

Bij grote werkproducten, zoals een systeemrequirementsspecificatie, moeten requirements op verschillende abstractieniveaus gescheiden worden gehouden door de specificatie dienovereenkomstig te structureren (3.6). Een requirement op een hoog abstractieniveau kan worden verfijnd in verscheidene gedetailleerde requirements op lagere, meer concrete niveaus.

Wanneer businessrequirements en belanghebbendenrequirements worden uitgedrukt in duurzame werkproducten, zoals businessrequirementsspecificaties, belanghebbendenrequirementsspecificaties of visiedocumenten, gaan deze vooraf aan de systeemrequirementsspecificaties. In andere omstandigheden kunnen businessrequirements, belanghebbendenrequirements en systeemrequirements samen evolueren.

3.1.3 Niveau van detaillering (L2)

Het niveau van *detaillering* van de requirements is afhankelijk van verschillende factoren, in het bijzonder:

- De probleem- en ontwikkelcontext
- De mate van gedeeld begrip van het probleem
- De mate van vrijheid van ontwerpers en programmeurs
- Beschikbaarheid van snelle terugkoppeling van belanghebbenden tijdens het ontwerp en de implementatie
- Kosten-baten van een gedetailleerde specificatie
- Opgelegde standaarden en beperkingen in de regelgeving

Hoe gedetailleerder de requirements, hoe kleiner het risico dat je uiteindelijk iets onverwachts of ongespecificeerds krijgt. De kosten voor de specificatie nemen echter toe naarmate het niveau van detaillering toeneemt.

3.1.4 Aspecten waarmee rekening moet worden gehouden voor werkproducten (L1)

Bij het specificeren van requirements in werkproducten moet rekening worden gehouden met verschillende aspecten.

1. Requirements worden ingedeeld volgens hun soort (1.1):
   a) Functionele requirements
   b) Kwaliteitsrequirements
   c) Beperkingen
2. Functionele requirements richten zich op verschillende aspecten van een systeem:
   a) Structuur en gegevens
   b) Functie en stroom
3. Toestand en gedrag
   a) Tot slot kunnen requirements alleen worden begrepen in hun context (Principe 4 in 2):
   b) *Systeemcontext*, inclusief externe actoren
   c) Systeem*grens* en externe interfaces

Er bestaan veel onderlinge verbanden en afhankelijkheden tussen de genoemde aspecten. Zo kan een verzoek van een gebruiker (context) een toestandsovergang (toestand en gedrag) veroorzaken die een actie in gang zet, gevolgd door een andere actie (functie en stroom) die gegevens (structuur en gegevens) nodig heeft om binnen een bepaald tijdsinterval (kwaliteit) een resultaat te leveren aan de gebruiker (context).

Sommige werkproducten richten zich op een specifiek aspect en abstraheren van andere aspecten. Dit is met name het geval voor requirementsmodellen (3.4). Andere werkproducten, zoals een systeemrequirementsspecificatie, omvatten al deze aspecten. Wanneer verschillende aspecten in aparte werkproducten of in aparte hoofdstukken van hetzelfde werkproduct worden gedocumenteerd, moeten deze werkproducten of hoofdstukken met elkaar in overeenstemming worden gehouden.

3.1.5 Algemene documentatierichtlijnen (L1)

Onafhankelijk van de gebruikte technieken zijn de volgende richtlijnen van toepassing bij het maken van werkproducten:

- Selecteer een werkproducttype dat past bij het *beoogde doel*.
- *Vermijd redundantie* door te verwijzen naar een bepaalde inhoud in plaats van dezelfde inhoud te herhalen.

- *Zorg ervoor dat er geen inconsistenties zijn* tussen de werkproducten, in het bijzonder wanneer deze verschillende aspecten bestrijken.
- *Gebruik termen consequent*, zoals gedefinieerd in de woordenlijst.
- *Structureer* de werkproducten op de juiste manier.

3.1.6 Plan de te gebruiken werkproducten (L1)

Elk project en elk domein is anders, dus het geheel van op te leveren werkproducten moet voor elke klus worden gedefinieerd. Daarom moeten de volgende punten worden overeengekomen:

- In welke werkproducten moeten de requirements worden vastgelegd en met welk doel?
- Welke abstractieniveaus moeten worden beschouwd?
- Tot welk detailniveau moeten de requirements op elk abstractieniveau worden gedocumenteerd?
- Hoe moeten de requirements in deze werkproducten worden weergegeven?

De te gebruiken werkproducten moeten in een vroeg stadium van een project worden gedefinieerd. Dit heeft verschillende voordelen, omdat het:

- Helpt bij het plannen van de vereiste inspanningen en middelen.
- Verzekert dat de juiste notaties worden gebruikt.
- Verzekert dat alle resultaten worden vastgelegd in de juiste werkproducten.
- Verzekert dat er geen grote herschikking van informatie en "eindredactie" nodig is.
- Helpt redundantie te voorkomen, wat resulteert in minder werk en een betere onderhoudbaarheid.

3.2 Werkproducten op basis van natuurlijke taal (L2)

Vanaf het ontstaan van systematische RE zijn requirements op basis van natuurlijke taal een kernmiddel geweest om requirements te specificeren.

Werkproducten op basis van natuurlijke taal hebben grote voordelen:

- Ongedwongen natuurlijke taal is uiterst expressief en flexibel.
- Bijna elk denkbare requirement in welk aspect dan ook kan in natuurlijke taal worden uitgedrukt.
- Natuurlijke taal wordt in het dagelijks leven gebruikt en op school onderwezen, dus er is geen specifieke training nodig om teksten in natuurlijke taal te lezen en te begrijpen.

Tegenover deze voordelen staat het feit dat teksten in natuurlijke taal vaak op verschillende manieren kunnen worden geïnterpreteerd, wat een probleem vormt bij het specificeren van de requirements. Bovendien is het opsporen van dubbelzinnigheden, weglatingen en inconsistenties in dergelijke teksten moeilijk en duur.

Het schrijven van goede requirements op basis van natuurlijke taal kan worden ondersteund door:

- Het schrijven van korte en goed gestructureerde zinnen.
- Het definiëren en consequent gebruiken van een uniforme terminologie (3.5).
- Het vermijden van vage of dubbelzinnige termen en zinnen.
- Het kennen van de onderstaande valkuilen van technisch schrijven.

Bij het schrijven van technische documenten in natuurlijke taal zijn er enkele welgekende valkuilen die vermeden of met voorzichtigheid [GoRu2003] gebruikt moeten worden.

Dingen om te vermijden:

- Onvolledige beschrijvingen
- Ongespecificeerde zelfstandige naamwoorden
- Onvolledige voorwaarden
- Onvolledige vergelijkingen

Dingen die met voorzichtigheid moeten worden gebruikt:

- Passief taalgebruik
- Universele kwantoren (zoals "alle" of "nooit")
- Nominalisaties (d.w.z. zelfstandige naamwoorden die zijn afgeleid van een werkwoord, bijvoorbeeld "authenticatie")

3.3 Werkproducten gebaseerd op sjablonen (L3)

Werkproducten gebaseerd op sjablonen worden gebruikt om enkele van de tekortkomingen van werkproducten gebaseerd op natuurlijke taal te verhelpen door vooraf gedefinieerde structuren voor de requirements aan te bieden.

- *Zinsjablonen* bieden een voorgedefinieerde syntactische structuur voor een zin die een requirement uitdrukt, in het bijzonder een individuele requirement of een user story.
- *Formuliersjablonen* voorzien een set van vooraf gedefinieerde velden in een in te vullen formulier, bijvoorbeeld voor het schrijven van een use case of een meetbare kwaliteitsrequirement.
- *Documentsjablonen* bieden een voorgedefinieerde structuur voor een requirementsdocument.

In de literatuur zijn verschillende sjablonen beschreven. [ISO29148], [MWHN2009] en [Rupp2014] bevatten zinsjablonen voor individuele requirements. [Cohn2004] definieert een veelgebruikt woordsjabloon voor user stories en [Cock2001] beschrijft de formuliersjablonen voor use cases. [Laue2002] heeft een sjabloon voorgesteld voor taakomschrijvingen. In [ISO29148] en [RoRo2012] staan documentsjablonen voor volledige specificaties. Bovendien zou een klant het gebruik van klantspecifieke sjablonen in een project kunnen opleggen.

Voordelen van werkproducten gebaseerd op sjablonen:

- Zorgen voor een duidelijke, herbruikbare structuur
- Helpen om de meest relevante informatie vast te leggen
- Zorgen dat requirements en specificaties er uniform uitzien

- Verbeteren de algemene kwaliteit van de requirements en requirementspecificaties

Nadelen en valkuilen van werkproducten gebaseerd op sjablonen:

- Mensen richten zich vaak op de formele voltooiing van het sjabloon in plaats van op de inhoud.
- Aspecten die niet in het sjabloon zijn opgenomen, worden eerder weggelaten.

3.4 Werkproducten gebaseerd op modellen (L3)

Requirements die in natuurlijke taal worden weergegeven, hebben beperkingen [Davi1993], in het bijzonder met betrekking tot het verkrijgen van een overzicht van een set van requirements en het begrijpen van de relaties tussen requirements. Modellering van requirements pakt deze beperkingen aan.

3.4.1 De rol van modellen in Requirements Engineering (L2)

Een *model* is een abstracte weergave van een bestaand deel van de werkelijkheid of een te creëren deel van de werkelijkheid. Het begrip "werkelijkheid" omvat alle denkbare elementen, fenomenen of concepten, inclusief andere modellen. Met betrekking tot een model wordt het gemodelleerde deel van de werkelijkheid het *origineel* genoemd. Een voorbeeld van een model buiten het domein van software engineering is een bouwinformatiemodel (BIM) [ISO19650], die de nodige elementen modelleert om gebouwen en andere bouwtechnische constructies te plannen, te bouwen en te beheren.

In RE helpen modellen om de relaties en onderlinge verbanden tussen requirements te begrijpen en geven ze een overzicht over een set van requirements. Dit wordt in de eerste plaats bereikt door te focussen op bepaalde aspecten – bijvoorbeeld gedrag – en alle andere aspecten weg te laten. Het verkrijgen van overzicht wordt ook ondersteund door het gebruik van een grafische weergave voor een model. Modellen kunnen echter ook op een niet-grafische manier worden voorgesteld, bijvoorbeeld met tabellen.

Requirementsmodellen hebben voordelen ten opzichte van requirements die in natuurlijke taal worden weergegeven:

- Relaties en verbanden tussen requirements zijn gemakkelijker te begrijpen met grafische modellen dan wanneer ze in natuurlijke taal worden gespecificeerd.
- Door te focussen op één enkel aspect vermindert dit de cognitieve belasting voor het begrijpen van de gemodelleerde requirements.
- Requirementsmodelleertalen hebben een beperkte syntax die mogelijke dubbelzinnigheden en omissies vermindert.

Modellen hebben ook beperkingen:

- Het is een uitdaging om modellen die zich op verschillende aspecten richten op elkaar af te stemmen.
- Informatie uit verschillende modellen moet worden geïntegreerd voor een algemeen begrip.

- Modellen zijn vooral gericht op functionele requirements; de meeste kwaliteitsrequirements en beperkingen kunnen niet of slechts met veel moeite in modellen worden uitgedrukt.
- De beperkte syntaxis van een grafische modelleertaal heeft tot gevolg dat niet elk relevant gegeven in een model kan worden uitgedrukt.

Daarom worden requirementsmodellen en requirements in natuurlijke taal vaak gecombineerd.

In RE kunnen modellen worden gebruikt voor:

- Het gedeeltelijk of zelfs volledig *specificeren van* (voornamelijk functionele) requirements als een manier om de tekstuele weergave van requirements te vervangen.
- *Het opdelen van* een complexe werkelijkheid in goed gedefinieerde en elkaar aanvullende aspecten; elk aspect wordt vertegenwoordigd door een specifiek model.
- *Het omschrijven van* tekstuele requirements om de begrijpelijkheid ervan te verbeteren, in het bijzonder met betrekking tot hun onderlinge relaties.
- *Het valideren van* tekstueel weergegeven requirements met als doel om omissies, dubbelzinnigheden en inconsistenties aan het licht te brengen.

Modelleertalen worden gebruikt om modellen uit te drukken. Verschillende modelleertalen, bijvoorbeeld UML [OMG2017] en BPMN [OMG2013], zijn gestandaardiseerd. Wanneer requirements worden gespecificeerd in een niet-standaard modelleertaal, is een legenda nodig die de syntax en de semantiek van de gebruikte modelleertaal verklaart.

Er zijn veel soorten modellen die in RE kunnen worden gebruikt. Een Requirementsanalist moet begrijpen welk type model het meest geschikt is om de requirements in een bepaalde situatie te specificeren.

In een vroeg stadium begint de Requirementsanalist vaak met het modelleren van de context (3.4.2) of de doelen van het beoogde systeem.

3.4.2 Modelleren van de context (L2)

Modellen die zich richten op het contextaspect specificeren de structurele inbedding van een systeem in zijn omgeving en de interactie tussen een systeem en de actoren in de systeemcontext.

Contextmodellen specificeren een systeem en de actoren in de systeemcontext die een interactie met het systeem hebben. Een contextmodel schetst ook de interfaces tussen een systeem en zijn context (bijvoorbeeld in termen van informatie die wordt uitgewisseld).

Contextdiagrammen worden gebruikt als grafische modelleertaal voor het uitdrukken van contextmodellen. Er is geen gestandaardiseerde notatie voor contextdiagrammen. Contextdiagrammen uit gestructureerde analyse [DeMa1978] of op maat gemaakte box-and-line diagrammen [Glin2019] kunnen worden gebruikt om contextmodellen uit te drukken.

In de modelleertaal UML [OMG2017] bieden *use case diagrammen* de mogelijkheid tot het modelleren van een systeem en zijn context in termen van de use cases van het systeem en de actoren in de systeemcontext die via deze use cases in verband staan met het systeem.

Use cases modelleren de dynamische interactie tussen een actor in de systeemcontext en een systeem vanuit het perspectief van de actor. Use cases worden meestal geschreven met behulp van formuliersjablonen in natuurlijke taal (3.3) of met behulp van UML-activiteitendiagrammen (3.4.4).

3.4.3 Modelleren van structuur en gegevens (L3)

Modellen die zich richten op structuur- en data-aspecten specificeren requirements voor statisch-structurele eigenschappen van een systeem of een domein.

Statische domeinmodellen specificeren de (business)objecten en hun relaties in een bepaald domein. Deze kunnen worden uitgedrukt met UML-klassendiagrammen [OMG2017].

Klassenmodellen specificeren op de eerste plaats de klassen van een systeem en hun attributen en relaties. Klassen vertegenwoordigen materiële en immateriële entiteiten in de echte wereld die het systeem moet kennen om zijn taken te kunnen uitvoeren. UML-klassendiagrammen worden doorgaans gebruikt als modelleertaal voor klassenmodellen.

3.4.4 Modelleren van functie en stroom (L3)

In modellen die zich richten op functie- en stroomaspecten worden eisen gesteld aan de volgorde van de acties die nodig zijn om de vereiste resultaten uit bepaalde invoer (input) te halen of aan de acties die nodig zijn om een (business)proces uit te voeren, met inbegrip van de besturings- en gegevensstroom tussen de acties en wie verantwoordelijk is voor welke actie.

Activiteitenmodellen worden gebruikt om de systeemfuncties te specificeren. In de modelleertaal UML [OMG2017] worden *activiteitendiagrammen* gebruikt om activiteitenmodellen uit te drukken. Ze bieden elementen voor het modelleren van acties en de besturingsstroom tussen acties. Activiteitendiagrammen kunnen ook aangeven wie verantwoordelijk is voor welke actie. Geavanceerde modelleringselementen (die niet onder het CPRE Foundation Level vallen) bieden de middelen voor het modelleren van de datastroom.

Procesmodellen worden gebruikt om bedrijfsprocessen of technische processen te beschrijven. Ze kunnen worden uitgedrukt met UML-activiteitendiagrammen of met specifieke procesmodelleertalen zoals BPMN [OMG2013]. Op het CPRE Foundation Level gebruiken we alleen UML-activiteitendiagrammen voor de procesmodellering.

3.4.5 Modelleren van toestand en gedrag (L2)

Modellen die zich richten op toestand en gedrag specificeren requirements voor het gedrag van een systeem of een domeincomponent in termen van toestandsafhankelijke reacties op gebeurtenissen of de dynamiek van componentinteracties.

Toestandsmachines modelleren gebeurtenissen die de overgang van de ene toestand naar de andere in gang zetten en de acties die moeten worden uitgevoerd wanneer een toestandsovergang plaatsvindt. *Toestandsdiagrammen* [Hare1988] zijn toestandsmachines met toestanden die hiërarchisch en/of orthogonaal zijn opgedeeld. Toestandsmachines, inclusief toestandsdiagrammen, kunnen worden uitgedrukt in de modelleertaal UML [OMG2017] met *toestandsmachinediagrammen* (ook wel *toestandsdiagrammen* genoemd).

3.4.6 Meer modeltypen in Requirements Engineering (L1)

Op het CPRE foundation-niveau blijft het begrip en de toepassing van modellen beperkt tot de geselecteerde, belangrijke modeltypes. Er bestaan nog andere soorten modellen die worden gebruikt bij Requirements Engineering. Op het niveau van de CPRE foundation volstaat het deze te kennen en te weten waarvoor ze worden gebruikt.

Doelmodellen vertegenwoordigen een reeks doelen, subdoelen en de relaties daartussen. Doelmodellen kunnen ook taken en middelen omvatten die nodig zijn om een doel te bereiken, actoren die een doel willen bereiken, en belemmeringen die het bereiken van een doel in de weg staan.

In SysML [OMG2019] kunnen *blokdefinitiediagrammen* worden gebruikt als contextdiagrammen met behulp van stereotype blokken voor het systeem en de actoren. Blokdefinitiediagrammen kunnen ook worden gebruikt om de structuur van een systeem te modelleren in termen van de conceptuele entiteiten van het systeem en de relaties tussen deze entiteiten.

Domeinverhaalmodellen kunnen worden gebruikt om functie en stroom te modelleren, door visuele verhalen te specificeren over hoe actoren interacteren met apparaten, artefacten en andere items in een domein, doorgaans met behulp van domeinspecifieke symbolen [HoSch2020]. Ze zijn een middel om het toepassingsdomein waarin een systeem voor bedoeld is te begrijpen.

Interactiemodellen modelleren de dynamische interacties tussen objecten of actoren. UML-*sequentiediagrammen* zijn een gebruikelijke manier om de interactie tussen objecten te specificeren.

3.5 Verklarende woordenlijsten (L2)

In alle RE werk waarbij meer dan één persoon betrokken is, bestaat het risico van een gebrek aan gedeeld begrip van de terminologie – dat wil zeggen dat sommige mensen dezelfde termen op verschillende manieren interpreteren. Om dit probleem te beperken, wordt het gedeelde begrip van de relevante termen vastgelegd in een verklarende woordenlijst.

Een *verklarende woordenlijst* is een centrale verzameling van definities voor context-specifieke termen, alledaagse termen met een speciale betekenis in de gegeven context, afkortingen en acroniemen. Synoniemen (verschillende termen die hetzelfde betekenen) moeten als zodanig worden aangemerkt. Homoniemen (waarbij dezelfde term voor verschillende dingen wordt gebruikt) moeten worden vermeden of als zodanig worden kenbaar gemaakt.

De volgende regels zijn van toepassing op de verklarende woordenlijsten:

- Beheer de woordenlijst centraal.
- Onderhoud de verklarende woordenlijst gedurende de gehele ontwikkeling van het systeem.
- Wijs een persoon of kleine groep mensen aan die verantwoordelijk is voor de verklarende woordenlijst.
- Gebruik een uniforme stijl en structuur voor de verklarende woordenlijst.
- Betrek de belanghebbenden en zoek overeenstemming over de terminologie.
- Maak de verklarende woordenlijst toegankelijk voor alle betrokkenen.
- Stel het gebruik van de verklarende woordenlijst verplicht.
- Controleer de werkproducten op het juiste gebruik van de verklarende woordenlijst.

3.6 Requirementsdocumenten en Requirementsdocumentatiestructuren (L2)

Requirementsspecificatiedocumenten (3.1.1) omvatten verschillende RE-werkproducten. Het is daarom belangrijk om dergelijke documenten in te bedden in een goed gedefinieerde structuur, om zo een consistente en onderhoudbare verzameling van requirements te creëren. Naast requirements kan een requirementsdocument ook bijkomende informatie en uitleg bevatten – bijvoorbeeld een verklarende woordenlijst, acceptatiecriteria, projectinformatie of kenmerken van de technische implementatie.

Requirements kunnen ook worden georganiseerd in andere documentatiestructuren dan klassieke documenten.

Veelgebruikte documenten zijn:

- Belanghebbendenrequirementsspecificatie
- Gebruikersrequirementsspecificatie (een deelverzameling van een belanghebbendenrequirementsspecificatie, die alleen gebruikersrequirements behandelt)
- Systeemrequirementsspecificatie
- Business requirementsspecificatie
- Visiedocument

Vaak gebruikte alternatieve documentatiestructuren zijn:

- Product backlog
- Sprint backlog
- Storymap

Zowel de selectie van een documentatiestructuur als de interne organisatie van de gekozen structuur zijn afhankelijk van:

- Het gekozen ontwikkelproces (5)
- De ontwikkelaanpak en het domein

- Het contract (een klant kan het gebruik van een bepaalde documentatiestructuur voorschrijven)
- De grootte van het document

Documentsjablonen kunnen helpen bij het structureren van een requirementsspecificatie. Sjablonen zijn beschikbaar in de literatuur [Vole2020], [RoRo2012] en in standaarden [ISO29148]. Sjablonen kunnen ook worden hergebruikt uit eerdere, soortgelijke projecten of kunnen worden opgelegd door een klant. Een organisatie kan ook besluiten om zelf een sjabloon aan te maken als interne standaard.

3.7 Prototypes in Requirements Engineering (L1)

In RE zijn *prototypes* een manier om requirements te specificeren aan de hand van een voorbeeld en om requirements te valideren. In het bijzonder kunnen prototypes worden gebruikt als de betrokken belanghebbenden geen werkproducten gebaseerd op natuurlijke taal, sjablonen, of modellen willen schrijven en reviewen.

Verkennende prototypes [LiSZ1994] worden gebruikt om gezamenlijk begrip te creëren, requirements te verduidelijken of requirements te valideren op verschillende niveaus van getrouwheid. Ze worden na gebruik weggegooid.

- *Wireframes* zijn prototypes van lage getrouwheid, opgesteld met eenvoudige materialen of schetstools, die vooral dienen om ontwerpideeën en gebruikersinterfaceconcepten te bespreken en te valideren.
- *Mock-ups* zijn prototypes van gemiddelde mate waarheidsgetrouwheid. Bij het specificeren van digitale systemen maken ze gebruik van echte schermen en klikbare voorbeelden, maar zonder echte functionaliteit. Ze dienen in de eerste plaats voor het specificeren en valideren van gebruikersinterfaces.
- *Native prototypes* zijn prototypes met een hoge mate van waarheidsgetrouwheid, die kritische onderdelen van een systeem zodanig implementeren dat belanghebbenden het prototype kunnen gebruiken om te ervaren of dat prototype deel van het systeem zal werken en zich zal gedragen zoals verwacht.

Naargelang de mate van waarheidsgetrouwheid kunnen verkennende prototypes duur zijn om te maken, zodat er altijd een balans moet worden gezocht tussen de kosten en de behaalde baten.

Evolutionaire prototypes [LiSZ1994] zijn proefsystemen die de kern vormen van een te ontwikkelen systeem. Het uiteindelijke systeem evolueert door het stapsgewijs uitbreiden en verbeteren van het systeem in verschillende iteraties.

3.8 Kwaliteitscriteria voor werkproducten en requirements (L1)

Een requirement moet aan specifieke kwaliteitscriteria voldoen om als een goed requirement te worden beschouwd. In de waardegerichte aanpak van moderne RE (Principe 1 in 2) moet de mate waarin aan een kwaliteitscriterium wordt voldaan, in evenwicht zijn met de waarde die door dit requirement wordt gecreëerd. Dit betekent dat de requirements niet volledig

hoeven te voldoen aan alle kwaliteitscriteria – maar hoe hoger de waarde van een requirement, hoe relevanter de kwaliteitscriteria, om het risico op mislukking te verkleinen.

Geschiktheid en begrijpelijkheid zijn de belangrijkste kwaliteitscriteria voor individuele requirements. Zonder deze criteria is een requirement nutteloos of zelfs schadelijk, ongeacht of aan alle andere criteria wordt voldaan.

Kwaliteitscriteria voor *individuele requirements*:

- Geschikt (beschrijft de werkelijke en overeengekomen behoeften van de belanghebbenden)
- Noodzakelijk
- Ondubbelzinnig
- Volledig (op zichzelf staand)
- Begrijpelijk
- Verifieerbaar

Zoals beschreven in 3.1.1 worden requirements meestal vastgelegd in verschillende werkproducten die individuele of meerdere requirements afdekken. De bovenstaande kwaliteitscriteria worden gebruikt om goede individuele requirements te creëren in een werkproduct. Voor werkproducten die meer dan één requirement beschrijven, moeten de volgende bijkomende kwaliteitscriteria in aanmerking worden genomen.

Kwaliteitscriteria voor werkproducten die *meerdere requirements* beschrijven:

- Consistent
- Niet-redundant
- Volledig (er ontbreken geen bekende en relevante requirements)
- Wijzigbaar
- Traceerbaar
- Conform standaards en sjablonen

4 Het uitwerken van requirements (L3)

Doel: Het gebruik begrijpen van werkwijzen om requirementsbronnen te identificeren, requirements te eliciteren, conflicten te identificeren en op te lossen, en requirements te valideren

Duur: 4 uur 30 minuten

Termen: Requirementsbron, systeemgrens, systeemcontext, requirementselicitatie, requirementsonderhandeling, requirementsvalidatie, belanghebbende, Kano-model, conflictsoplossing

Leerdoelen

LD 4.1.1 De grenzen van het systeem bepalen om te focussen op de relevante requirements (L3)
LD 4.1.2 De relevante bronnen voor het te ontwikkelen systeem kennen (L1)
LD 4.1.3 De belanghebbenden identificeren en creëren van een lijst van belanghebbenden (L3)
LD 4.1.4 De voordelen begrijpen van het beheren van belanghebbenden (L2)
LD 4.2.1 Begrijpen hoe het Kano-model kan helpen om de juiste requirements te eliciteren (L2)
LD 4.2.2 Het verschil begrijpen tussen verzameltechnieken en ontwerp- en ideegeneratietechnieken (L2)
LD 4.2.3 Begrijpen hoe de juiste elicitatietechniek voor een bepaalde situatie te kiezen (L2)
LD 4.3.1 De verschillende soorten conflicten kennen (L1)
LD 4.3.2 Begrijpen welke activiteiten nodig zijn om conflicten op te lossen (L2)
LD 4.3.3 Begrijpen hoe passende technieken voor het oplossen van conflicten toe te passen (L2)
LD 4.4.1 Begrijpen waarom requirementsdocumenten moeten worden gevalideerd (L2)
LD 4.4.2 De vier belangrijke aspecten kennen voor requirementsvalidatie (L1)
LD 4.4.3 Begrijpen hoe passende technieken voor requirementsvalidatie toe te passen (L2)

4.1 Requirementsbronnen (L3)

De kwaliteit en de volledigheid van de requirements zijn sterk afhankelijk van de betrokken requirementsbronnen. Het over het hoofd zien van een relevante bron leidt tot een onvolledig inzicht in de requirements of tot onvolledige requirements. De identificatie van requirementsbronnen is een iteratief en recursief proces dat voortdurend aandacht vereist.

Een gedeeld begrip (Principe 3 in 2) van de context van het te ontwikkelen systeem is een voorwaarde om de relevante requirementsbronnen te kunnen identificeren. Het gebied tussen de systeemgrens en de contextgrens wordt de (systeem)context genoemd (Principe 4 in 2). De (systeem)context is nodig om de aard van de te ontwikkelen requirements te begrijpen en zo de eigenlijke requirementsbronnen te identificeren.

Requirementsbronnen worden in drie types onderverdeeld:

- Belanghebbenden
- Documenten
- Systemen

De belanghebbenden van een systeem (zie [Glin2020] voor een definitie en zie ook Principe 2 in 2) zijn de belangrijkste requirementsbron. Gebruikelijke belanghebbendenrollen zijn [BiSp2003]:

- Gebruikers (ook eindgebruikers genoemd)
- Sponsors
- Managers
- Ontwikkelaars
- Autoriteiten
- Klanten

Voorts moeten mensen of organisaties die *de gevolgen* van een systeem *ondervinden*, als (indirecte) belanghebbenden worden beschouwd.

Aan het begin van een ontwikkelproject moeten de belanghebbenden systematisch worden geïdentificeerd en de resultaten moeten gedurende de gehele ontwikkeling als een continue activiteit worden beheerd. Dit omvat de identificatie van zowel de belanghebbendenrollen als de personen in deze rollen.

Voor alle systemen met een gebruikersinterface vormen de *eindgebruikers* van het systeem een belanghebbendengroep die van bijzonder belang is voor de Requirementsanalist. Eindgebruikers worden best gegroepeerd (bijvoorbeeld op basis van vergelijkbare rollen, taken of verantwoordelijkheden).

Wanneer eindgebruikers individueel kunnen worden geïdentificeerd, kies je best vertegenwoordigers van elke groep. In het andere geval kunnen persona's worden gedefinieerd als vertegenwoordiging van de relevante eindgebruikersgroepen [Coop2004].

Mogelijke bronnen voor het identificeren van relevante belanghebbenden en belanghebbendenrollen zijn:

- Checklists van typische belanghebbendengroepen en -rollen
- Organisatiestructuren
- Bedrijfsprocesdocumentatie
- Marktrapporten
- Initiële belanghebbenden voor het identificeren van *bijkomende* belanghebbenden

Belanghebbenden moeten worden gedocumenteerd in een up-to-date belanghebbendenlijst met (minstens) de volgende informatie:

- Naam
- Functie (rol)
- Bijkomende persoonlijke gegevens en contactgegevens
- Beschikbaarheid (tijd en locatie) gedurende de looptijd van het project
- Relevantie
- Gebied en omvang van de expertise
- Doelstellingen en belangen in termen van het project

Er kunnen zich problemen voordoen met belanghebbenden als de rechten en plichten van een belanghebbende niet duidelijk zijn of als de behoeften van de belanghebbende niet

voldoende aan bod komen. Relatiebeheer van belanghebbenden [Bour2009] is een effectieve manier om problemen met belanghebbenden tegen te gaan.

In de meeste systeemcontexten zijn er nog meer bronnen beschikbaar. Die moeten ook worden beschouwd voor een succesvol nieuw systeem, aangezien de meeste belanghebbenden niet spreken over het voor de hand liggende: hun "onderbewuste" requirements (4.2).

Bijkomende requirementsbronnen zijn onder andere:

- Bestaande en legacy systemen
- Procesdocumenten
- Wettelijke of regelgevende documenten
- Bedrijfsspecifieke regelgeving
- (Marketing-)informatie over mogelijke toekomstige gebruikers

Een andere requirementsbron kan worden gevonden door te kijken naar vergelijkbare situaties in totaal verschillende domeinen.

4.2 Het eliciteren van requirements (L2)

Binnen de elicitatie is het de taak van de Requirementsanalist om de wensen en behoeften van de belanghebbenden te begrijpen en er tegelijkertijd voor te zorgen dat de requirements van alle relevante requirementsbronnen zijn verzameld door passende technieken toe te passen. Een belangrijk punt in elicitatie is het omzetten van impliciete behoeftes, wensen en verwachtingen in expliciete requirements.

Om requirements te eliciteren is het cruciaal om de aard en het belang van elke requirement te kennen. Deze kunnen van project tot project en ook in de loop van de tijd veranderen. Het Kano-model [KaeA1984] deelt requirements in drie relevante categorieën in:

- Delighters (synoniemen: WOW-factoren, onbewuste requirements)
- Satisfiers (synoniemen: prestatiefactoren, bewuste requirements)
- Dissatisfiers (synoniemen: basisfactoren, onderbewuste requirements)

Er bestaan allerhande verschillende technieken om deze categorieën van requirements te eliciteren. We maken een onderscheid tussen:

- Verzameltechnieken
- Ontwerp- en ideegeneratietechnieken

Verzameltechnieken zijn bekende requirementselicitatietechnieken [BaCC2015] die helpen om satisfiers en dissatisfiers te eliciteren door verschillende bronnen te onderzoeken.

Er zijn vier hoofdcategorieën te onderscheiden:

- Uitvraagtechnieken
- Samenwerkingstechnieken
- Observatietechnieken
- Technieken gebaseerd op artefacten

Ontwerp- en ideegeneratietechnieken zijn bedoeld om de creativiteit te stimuleren tijdens de requirementselicitatie. Ze zijn gericht op het bedenken van manieren om een probleem op te lossen en op het uitpluizen van ontwerpideeën [Kuma2013]. Dit kan leiden tot nieuwe of innovatieve requirements die vaak delighters zijn. Populaire voorbeelden van dergelijke technieken zijn brainstorming [Osbo1979], analogieën, prototyping (bv. mock-ups), scenario's, en storyboards.

Een breder concept met betrekking tot ontwerp- en ideegeneratie is *design thinking*. Er bestaan verschillende benaderingen, zoals *d.school* [Sdsc2012] en *Designing for Growth* [LiOg2011]. Beiden bieden een groot aanbod van technieken die kunnen worden gebruikt voor requirementselicitatie.

Elicitatietechnieken moeten in staat zijn om alle soorten requirements – zowel functionele en kwaliteitsrequirements als beperkingen – te ontdekken. In de praktijk krijgen kwaliteitsrequirements en beperkingen vaak minder aandacht.

Om *kwaliteitsrequirements* te eliciteren, kan een kwaliteitsmodel zoals de ISO 25010-norm [ISO25010] als checklist worden gebruikt. Dit model kan ook nuttig zijn bij het kwantificeren van de requirements.

Beperkingen kunnen worden gevonden door rekening te houden met mogelijke beperkingen van de potentiële oplossingsruimte – bijvoorbeeld technische, juridische, organisatorische, culturele of milieukwesties.

Het kiezen van de juiste elicitatietechnieken is een kritische sleutelcompetentie die afhankelijk is van veel verschillende factoren, zoals:

- Type systeem
- Software development life cycle model
- Betrokken personen
- Inrichting van de organisatie

De beste resultaten worden gewoonlijk bereikt met een combinatie van verschillende elicitatietechnieken. [CaDJ2014] presenteert een systematische aanpak om technieken te selecteren.

4.3 Het oplossen van requirementsconflicten (L2)

Elicitatietechnieken op zich zorgen er niet voor dat de resulterende set van requirements als geheel consistent, volledig, conformerend, enzovoorts is. (3.8). Echter, voor de definitieve set moeten alle belanghebbenden alle voor hen relevante requirements begrijpen en ermee instemmen. Als sommige belanghebbenden niet akkoord zijn, moet deze situatie worden onderkend als een conflict dat dienovereenkomstig moet worden opgelost. Op basis van het soort conflict en de contextuele informatie moeten geschikte technieken om het conflict op te lossen worden geselecteerd. Dit vereist een diepgaand begrip van de aard van het requirementsconflict en van de houding van de betrokken belanghebbenden.

De stappen om conflicten vast te stellen en op te lossen zijn:

- Conflictidentificatie
- Conflictanalyse
- Conflictoplossing
- Documentatie van de conflictoplossing (genomen besluiten)

Het is nuttig om een onderscheid te maken tussen verschillende soorten [Moor2014] conflicten. De volgende soorten conflicten vereisen vaak de aandacht van de Requirementsanalist:

- Inhoudelijk conflict
- Gegevensconflict
- Belangenconflict
- Waardenconflict
- Relatieconflict
- Structureel conflict

Om conflicten succesvol op te lossen kunnen algemeen bekende technieken worden toegepast:

- Overeenstemming bereiken
- Compromis sluiten
- Stemmen
- Overrulen
- Definiëren van varianten

Daarnaast zijn er verschillende hulptechnieken, bijvoorbeeld:

- Consider-All-Facts
- Plus-Minus-Interesting
- Beslissingsmatrix

4.4 Het valideren van requirements (L2)

Het valideren van requirements is een belangrijke stap in de richting van een succesvol systeem (Principe 6 in 2). Het vroeg in het proces borgen van de kwaliteit van requirements zal verspilde moeite in latere fases beperken. Valideren van requirements betekent het controleren van de kwaliteit van zowel het werkproduct in kwestie als de individuele requirements die erin vervat zitten (zie 3.8 voor details).

Belangrijke aspecten waarmee rekening moet worden gehouden bij de validatie van requirements zijn:

- Betrek de juiste belanghebbenden
- Houd de detectie en de verbetering van fouten gescheiden
- Valideer vanuit verschillende invalshoeken
- Valideer herhaaldelijk

Er bestaan verschillende technieken voor validatie (bijvoorbeeld [GiGr1993], [OleA2018]). Deze validatietechnieken worden vaak ingedeeld in:

- *Reviewtechnieken*, waaronder:
  - Walkthrough
  - Inspectie
- *Verkennende technieken*, bijvoorbeeld:
  - Toepassen van prototypes
  - Alfa- en bètatesten
  - A/B-test [KoTh2017]
  - Het bouwen van een Minimum Viable Product (MVP)
- *Ontwikkeling van een proefstuk*

Deze technieken verschillen in formaliteit en vereiste inspanning. Welke techniek moet worden geselecteerd, hangt af van factoren zoals het software development life cycle model, de volwassenheid van het ontwikkelproces, de complexiteit en het risiconiveau van het systeem, diverse wettelijke en/of reglementaire requirements, en/of de noodzaak van een audit trail.

5 Proces- en werkstructuur (L3)

Doel: Uitleggen van de concepten van RE-processen en de juiste procesconfiguraties toepassen

Duur: 1 uur 15 minuten

Termen: Proces, RE-proces

Leerdoelen

LD 5.1.1 De belangrijke factoren kennen die van invloed zijn op een RE-proces (L1)
LD 5.1.2 Begrijpen hoe en waarom deze factoren van invloed zijn (L2)
LD 5.2.1 De facetten die beschouwd moeten worden voor de configuratie van een RE-proces begrijpen (L2)
LD 5.3.1 Typische RE-procesconfiguraties kennen (L1)
LD 5.3.2 De stappen voor het configureren van een RE-proces begrijpen (L2)
LD 5.3.3 Voor eenvoudige systeem- en ontwikkelomgevingen een geschikte RE-procesconfiguratie selecteren en die toepassen (L3)

Er is een proces nodig om het te verrichten RE-werk in een bepaalde context vorm te geven en te structureren. Aangezien er geen one-size-fits-all RE-proces (1.4) bestaat, moet er een op maat gemaakt RE-proces worden geconfigureerd dat past bij de gegeven ontwikkel- en systeemcontext.

Het RE-proces geeft vorm aan de informatiestroom en het communicatiemodel tussen verschillende deelnemers (bijvoorbeeld klanten, gebruikers, Requirementsanalisten, ontwikkelaars, testers) en definieert ook de te gebruiken of te produceren werkproducten. Het RE-proces biedt dus het kader voor het eliciteren, documenteren, valideren en beheren van requirements.

5.1 Beïnvloedende factoren (L2)

Vele factoren beïnvloeden de configuratie van een RE-proces. De belangrijkste factoren zijn:

- Algehele procesaansluiting: het RE-proces moet aansluiten op het totale systeemontwikkelproces.
- Ontwikkelcontext
- Bekwaamheid en beschikbaarheid van belanghebbenden
- Gedeeld begrip
- Complexiteit en kriticiteit van het te ontwikkelen systeem
- Beperkingen
- Beschikbare tijd en budget
- Volatiliteit van de requirements
- Ervaring van de Requirementsanalisten

Een analyse van de beïnvloedende factoren geeft informatie over de juiste manier om het RE-proces te configureren. De beïnvloedende factoren beperken ook de ruimte van mogelijke procesconfiguraties. Wanneer bijvoorbeeld de belanghebbenden alleen aan het

begin van het project beschikbaar zijn, kan er geen proces worden gekozen dat uitgaat van continue feedback van de belanghebbenden.

5.2 Requirements Engineering procesaspecten (L2)

Er zijn drie doorslaggevende aspecten waarmee rekening moet worden gehouden bij het configureren van een RE-proces [Glin2019].

Tijdsaspect: Lineair versus iteratief

In een lineair proces worden de requirements vooraf in één enkele fase van het proces gespecificeerd. In een iteratief proces worden de requirements stapsgewijs gespecificeerd, te beginnen met algemene doelen en enkele initiële requirements, en vervolgens worden in elke iteratie requirements toegevoegd of gewijzigd.

Criteria voor de keuze van een *lineair* RE-proces:

- Het ontwikkelproces voor het systeem is planmatig en grotendeels lineair.
- De belanghebbenden kennen hun behoeften en kunnen deze vooraf specificeren.
- Als contractuele basis voor het uitbesteden van het ontwerp en de implementatie van het systeem is een uitgebreide requirementsspecificatie vereist.
- De regelgevende instanties eisen in een vroeg stadium van de ontwikkeling een uitgebreide, formeel gereleasede requirementsspecificatie.

Criteria voor de keuze van een *iteratief* RE-proces:

- Het ontwikkelproces van het systeem is iteratief en agile.
- Veel requirements zijn niet van tevoren bekend, maar zullen tijdens de ontwikkeling van het systeem naar voren komen en evolueren.
- Belanghebbenden zijn beschikbaar zodat er korte feedback-lussen kunnen worden opgezet als middel om het risico te beperken dat het verkeerde systeem wordt ontwikkeld.
- De duur van de ontwikkeling laat meer dan slechts één of twee iteraties toe.
- De mogelijkheid om requirements gemakkelijk te veranderen is belangrijk.

Doelaspect: Prescriptief versus verkennend

In een prescriptief RE-proces vormt de requirementsspecificatie een contract: alle requirements zijn bindend en moeten worden geïmplementeerd. In een verkennend RE-proces zijn alleen de doelen a priori bekend, terwijl de concrete requirements moeten worden verkend.

Criteria voor het kiezen van een *prescriptief* RE-proces:

- De klant eist een vast contract voor de systeemontwikkeling.
- Functionaliteit en beschouwingsgebied (scope) zijn belangrijker dan kosten en deadlines.
- De ontwikkeling van het gespecificeerde systeem kan worden aan- of uitbesteed.

Criteria voor het kiezen van een *verkennend* RE-proces:

- Belanghebbenden hebben in eerste instantie slechts een vaag idee over hun requirements.
- Belanghebbenden zijn sterk betrokken en geven continu een terugkoppeling.
- Deadlines en kosten zijn belangrijker dan functionaliteit en beschouwingsgebied (scope).
- Het is niet a priori duidelijk welke requirements daadwerkelijk zullen worden uitgevoerd en in welke volgorde ze zullen worden uitgevoerd.

Doelwitaspect: Klantspecifiek versus marktgericht

In een klantspecifiek RE-proces wordt het systeem door een klant besteld en door een leverancier ontwikkeld. In een marktgericht RE-proces wordt het systeem ontwikkeld als een product of dienst voor een markt, gericht op specifieke doelgroepen.

Criteria voor de keuze van een *klantspecifiek* RE-proces:

- Het systeem zal voornamelijk worden gebruikt door de organisatie die het systeem heeft besteld en betaalt voor de ontwikkeling ervan.
- De belangrijkere belanghebbenden zijn vooral verbonden met de organisatie van de klant.
- Er kunnen individuele personen worden geïdentificeerd voor de rollen van de belanghebbenden.
- De klant wil een requirementsspecificatie die als contract kan dienen.

Criteria voor de keuze van een *marktgericht* RE-proces:

- De ontwikkelende organisatie is van plan om het systeem in een bepaald marktsegment als product of dienst te verkopen.
- Potentiële gebruikers zijn niet individueel identificeerbaar.
- De Requirementsanalisten moeten de requirements zodanig ontwerpen dat deze overeenkomen met de voorziene behoeften van de beoogde gebruikers.
- Product owners, marketeers, digitale ontwerpers en systeemarchitecten zijn de belangrijkste belanghebbenden.

Hints en kanttekeningen

- De hierboven gepresenteerde criteria zijn eerder *heuristieken* dan vaste regels. Zo gebeurt bijvoorbeeld het uitbesteden van de ontwikkeling van het systeem bij voorkeur met een prescriptief RE-proces in plaats van met een verkennend proces, omdat het contract tussen de klant en de leverancier typisch gebaseerd is op een uitgebreide requirementsspecificatie. Het is echter ook mogelijk om een uitbestedingscontract af te spreken op basis van een verkennend RE-proces.
- Lineaire RE-processen werken alleen als er een geavanceerd proces voor het veranderen van requirements is ingericht.
- Lineaire RE-processen impliceren lange terugkoppelingslussen: om het risico op het ontwikkelen van het verkeerde systeem te beperken, moeten de requirements intensief worden gevalideerd.

- Bij het definiëren van een RE-proces worden *lineair* en *prescriptief* vaak samen gekozen.
- Verkennende RE-processen zijn typisch ook iteratieve processen (en omgekeerd).
- In een marktgeoriënteerd proces is terugkoppeling van gebruikers het enige middel om te valideren of het product daadwerkelijk voldoet aan de behoeften van het beoogde doelgroep.
- Het marktgerichte aspect gaat niet goed samen met de lineaire en prescriptieve aspecten.

5.3 Het configureren van een Requirements Engineering proces (L3)

In een concrete systeemontwikkelingscontext moeten degenen die voor RE verantwoordelijk zijn het toe te passen RE-proces configureren. Op basis van een analyse van de beïnvloedende factoren zoals beschreven in 5.1, 5.2 kan een geschikte combinatie van de beschreven procesaspecten worden gebruikt [Glin2019]. Hieronder worden drie typische combinaties beschreven.

Participatief RE-proces: Iteratief & verkennend & klantspecifiek

Belangrijkste toepassing:	Leverancier en klant werken nauw samen; belanghebbenden zijn sterk betrokken bij zowel de RE als de ontwikkelprocessen
Typische werkproducten:	Product backlog met user stories en/of taakomschrijvingen, prototypes
Typische informatiestroom:	Continue interactie tussen belanghebbenden, product owners, Requirementsanalisten en ontwikkelaars; kan ook een terugkoppeling van gebruikers bevatten

Contractueel RE-proces: Typisch lineair (soms iteratief) & prescriptief & klantspecifiek

Belangrijkste toepassing:	De requirementsspecificatie vormt de contractuele basis voor de ontwikkeling van een systeem door mensen die niet betrokken zijn bij de specificatie en met weinig interactie met de belanghebbenden na de requirementsfase.
Typische werkproducten:	Klassieke systeemrequirementsspecificatie, bestaande uit requirements gebaseerd op natuurlijke taal en modellen
Typische informatiestroom:	Voornamelijk van belanghebbenden naar Requirementsanalisten

Product-georiënteerd RE-proces: Iteratief & verkennend & marktgericht

Belangrijkste toepassing:	Een organisatie specificeert en ontwikkelt software om deze als product of dienst te verkopen of te distribueren
Typische werkproducten:	Product backlog, prototypes
Typische informatiestroom:	Interactie tussen product owner, marketing, Requirementsanalisten, digital designers, ontwikkelaars en (misschien) snelle terugkoppeling van klanten/gebruikers

Bedenk dat er systeem- en ontwikkelingscontexten kunnen zijn waarin geen van de bovengenoemde configuraties past. Zo kunnen wettelijke beperkingen het gebruik van een proces dat voldoet aan bepaalde normen zoals ISO/IEC/IEEE 29148 [ISO29148] opleggen.

Bij het configureren van een RE-proces raden wij aan om een vijfstapsprocedure te gebruiken:

1. Analyseer de beïnvloedende factoren (5.1)
2. Beoordeel de aspectcriteria (5.2)
3. Configureer het proces (5.3)
4. Bepaal de werkproducten (3)
5. Selecteer de juiste werkwijze

6 Het managen van requirements (L2)

Doel: De behoefte aan en het voordeel van requirementsmanagement begrijpen

Duur: 2 uur

Termen: requirementsmanagement, wijzigingsbeheer, traceerbaarheid, requirementsattributen, requirementslevenscyclus, prioritering

Leerdoelen

LD 6.1.1 Weten wat requirementsmanagement is en waarom het nodig is (L1)
LD 6.2.1 Uitleggen waarom requirements(tussen)producten een status/levenscyclusmodel nodig hebben (L2)
LD 6.3.1 Uitleggen hoe een concept voor versiebeheer van eisen er in een gegeven projectsituatie uitziet (L2)
LD 6.4.1 Het gebruik kennen van de requirementsconfiguraties en baselines (L1)
LD 6.5.1 Weten wat het nut is van requirementsattributen (L1)
LD 6.5.2 Uitleggen hoe een passende set attributen voor eisen eruit ziet in een gegeven projectsituatie (L2)
LD 6.5.3 Het nut van perspectieven verklaren en de verschillende perspectieven van requirements benoemen (L2)
LD 6.6.1 Redenen voor traceerbaarheid van requirements benoemen (L1)
LD 6.6.2 De verschillen tussen impliciete en expliciete traceerbaarheid samenvatten (L1)
LD 6.6.3 Weten hoe expliciete traceerbaarheid kan worden gedocumenteerd (L1)
LD 6.7.1 Weten hoe om te gaan met veranderingen in lineaire (planmatige) en agile benaderingen (L1)
LD 6.8.1 De reden voor prioritering kennen en bruikbare beoordelingscriteria kennen (L1)
LD 6.8.2 De stappen benoemen om requirements te prioriteren (L1)
LD 6.8.3 Verschillende categorieën van prioriteringstechnieken benoemen (L1)

6.1 Wat is requirementsmanagement? (L1)

Requirementsmanagement is het proces van het managen van bestaande requirements die in verschillende werkproducten zijn vastgelegd. Dit omvat in het bijzonder het opslaan, wijzigen en traceren van requirements [Glin2020]. Requirementsmanagement kan op verschillende manieren en op verschillende niveaus gebeuren, afhankelijk van het gekozen ontwikkelproces en de context – zie bijvoorbeeld [Leff2011], [Rupp2014] en [WiBe2013]. Ongeacht de omstandigheden is het de taak van requirementsmanagement om de requirements zo te managen, dat alle rollen in een project effectief en efficiënt kunnen werken.

6.2 Levenscyclusmanagement (L2)

Levenscyclusmanagement heeft betrekking op het proces van het bijhouden van alle werkproducten met betrekking tot de status in hun levenscyclus. Elke gedocumenteerd requirement en elk werkproduct heeft zijn eigen levenscyclus: het wordt gecreëerd, vervolgens geëvalueerd en verfijnd voordat het wordt gereviewed, aangepast, geconsolideerd, geaccordeerd, enzovoorts. Om te kunnen bepalen welk werkproduct in welke toestand verkeert, is een levenscyclusmodel nodig dat elke toegestane

levenscyclusstatus en statusovergang definieert. De werkelijke status van een werkproduct moet altijd duidelijk zijn, inclusief (normaliter) de geschiedenis van de wijzigingen.

6.3 Versiebeheer (L2)

Versiebeheer van requirements heeft betrekking op het proces van het bijhouden van alle werkproducten tijdens hun evolutie. Elke verandering in een werkproduct moet leiden tot een nieuwe versie. Met behulp van versiebeheer kan de geschiedenis van een werkproduct worden getraceerd tot zijn oorspronkelijke versie en kan een werkproduct worden hersteld naar een eerdere versie. Om dit te bewerkstelligen, heb je binnen versiebeheer drie dingen nodig:

- Een versienummer om de versie van een werkproduct uniek te bepalen.
- Een historiek van wat er veranderd is.
- Een concept voor het opslaan van werkproducten.

Versiebeheer moet worden toegepast op alle werkproducten [WiBe2013]. Een versienummer bestaat doorgaans uit minstens twee delen: de versie en het increment.

6.4 Configuraties en baselines (L1)

Een *requirementsconfiguratie* is een consistente set van werkproducten die requirements bevatten. Elke configuratie is gedefinieerd voor een specifiek doel en bevat maximaal één versie van elk werkproduct [Glin2020]. Het doel van configuraties is bijvoorbeeld om een set van werkproducten te beoordelen of om een inschatting van de ontwikkelinspanning te vergemakkelijken.

Een *baseline* is een stabiele, beheerde *configuratie* van werkproducten, die wordt gebruikt voor de releaseplanning of andere opleveringsmijlpalen in een project [Glin2020].

Configuraties hebben de volgende eigenschappen:

1. Logisch verband
2. Consistentie
3. Uniek
4. Onveranderlijkheid
5. Basis voor reset

6.5 Attributen en perspectieven (L2)

Attributen zijn nodig om belangrijke metadata voor een werkproduct te documenteren en worden meestal gebruikt om een aantal belangrijke vragen te beantwoorden tijdens de project- of productlevenscyclus.

Het doel van requirementsattributen is om teamleden en andere belanghebbenden in staat te stellen de informatie over de requirements, eender wanneer zij deze tijdens het project nodig hebben, te krijgen.

Het bepalen welke set van attributen relevant is, is afhankelijk van de informatiebehoefte van de verschillende belanghebbenden in het project. Bestaande standaarden, bijvoorbeeld [ISO29148], geven een overzicht van de meest relevante attributen.

De *perspectieven* zijn een selectie van de totale set van requirements, die alleen die inhoud bevatten die op dat moment van belang is. Technisch gezien is een perspectief een combinatie van filter- en sorteerinstellingen die beschikbaar of herbruikbaar is voor andere deelnemers door de geselecteerde combinatie op te slaan.

We maken een onderscheid tussen drie soorten perspectieven:

- *Selectieve perspectieven*
- *Projectieve perspectieven*
- *Aggregerende perspectieven*

In de meeste gevallen zijn requirementsperspectieven combinaties van selectieve, projectieve en aggregerende perspectieven voor het maken van rapporten.

6.6 Traceerbaarheid (L1)

Traceerbaarheid [GoFi1994] is de mogelijkheid om een requirement *terug* te leiden *tot zijn oorsprong* (d.w.z. belanghebbenden, documenten, rechtvaardigingen, et cetera) en *voorwaarts naar latere werkproducten* (bijvoorbeeld testgevallen), evenals *naar andere requirements* waarvan deze requirement afhankelijk is.

Traceerbaarheid is een voorwaarde voor requirementsbeheer en wordt vaak expliciet vereist in standaarden, wet- en regelgeving. Het implementeren van traceerbaarheid betekent in wezen het onderhouden van afhankelijkheden tussen verschillende werkproducten (3.1) op verschillende abstractieniveaus (3.1.2), detailniveaus (3.1.3) en naar alle relevante voorgangers en opvolgers om analyse, respecteren van normering en informatieverstrekking mogelijk te maken.

Traceerbaarheid kan *impliciet* worden gedocumenteerd, door werkproducten te structureren en te standaardiseren, of *expliciet,* door werkproducten aan elkaar te linken via hun unieke identifiers in verschillende vormen [HuJD2011]. Veel voorkomende representatievormen zijn hyperlinks, verwijzingen, matrices, tabellen of grafieken.

6.7 Omgaan met veranderingen (L1)

Requirements zijn niet statisch. Veranderingen in requirements vinden om vele verschillende redenen plaats en moeten goed worden afgehandeld (Principe 7 in 2), bijvoorbeeld door een formeel *wijzigingsverzoek* aan te maken, of door een nieuw item toe te voegen aan de *product backlog*.

De besluitvorming, planning en beheersing van het doorvoeren van een verandering is afhankelijk van de ontwikkelaanpak en het tijdstip waarop de verandering plaatsvindt.

Bij een *lineaire* aanpak wordt de beslissing met betrekking tot een verandering vaak genomen door een Change Control Board (in projecten) of een Change Advisory Board (als

het systeem in productie is). In een meer *iteratieve* aanpak neemt de product owner de verandering op in de product backlog en geeft het nieuwe item de juiste prioriteit.

6.8 Prioritering (L1)

Niet alle requirements zijn even belangrijk [Davi2005]. Beoordeling en prioritering worden toegepast om te bepalen wat de meest relevante requirements voor de volgende productrelease of –increment zijn.

De *beoordeling* van de requirements is de basis voor hun prioritering, vaak bepaald op basis van meerdere beoordelingscriteria zoals business value, urgentie, vereiste inspanning, afhankelijkheden en andere.

De *prioriteit* van een requirement beschrijft het belang van één requirement in vergelijking tot andere requirements volgens bepaalde criteria [Glin2020]. De *prioritering* zelf wordt uitgevoerd op basis van één criterium of op basis van meerdere criteria; dit hangt vooral af van de gekozen prioriteringstechniek.

Stappen voor de prioritering:

- Bepaal de belangrijkste doelen en beperkingen voor de prioritering
- Bepaal de gewenste beoordelingscriteria
- Bepaal de belanghebbenden die moeten worden betrokken
- Bepaal de requirements die geprioriteerd moeten worden
- Selecteer de prioriteringstechniek
- Voer de prioritering uit

Prioriteringstechnieken kunnen worden ingedeeld in:

- *Ad–hoc* prioriteringstechnieken
- *Analytische* prioriteringstechnieken

7 Tool-ondersteuning (L2)

Doel: Een overzicht geven van de rol van RE-tools en aspecten voor de implementatie

Duur: 30 minuten

Termen: Tool, RE-tool

Leerdoelen

LD 7.1.1: De verschillende soorten RE-tools kennen (L1)
LD 7.2.1: Uitleggen waarmee rekening moet worden gehouden bij de implementatie van RE-tools (L2)

7.1 Tools in Requirements Engineering (L1)

Het RE-proces kan worden ondersteund met tools die specifieke taken en activiteiten ondersteunen. Aangezien RE-processen vrij individueel zijn (5), richten de bestaande RE-tools zich vaak alleen op bepaalde aspecten in RE en ondersteunen ze zelden alle activiteiten. Voordat een tool wordt geselecteerd, moeten Requirementsanalisten beslissen welke taken en activiteiten tijdens het RE-proces moeten worden ondersteund en hoe. We onderscheiden tools die het volgende ondersteunen:

- Requirementsbeheer:
  - Definiëren en opslaan van requirementsattributen
  - Prioriteren van requirements
  - Beheren van versies en configuraties
  - Bepalen en volgen van requirements
- Beheren van wijzigingen in de requirements
  - Beheer van het RE-proces:
  - Meten van en rapporteren over het RE-proces
  - Meten van en rapporteren over de productkwaliteit
- Beheren van de RE-workflow
  - Documenteren van de kennis over de requirements:
  - Delen van requirements
  - Creëren van een gedeeld begrip van de requirements
- Modelleren van requirements
- Samenwerking in RE
- Testen/simulatie van requirements

Tools bevatten vaak een mix van de bovengenoemde kenmerken. Om ervoor te zorgen dat alle RE-taken op de juiste wijze worden uitgevoerd, kunnen verschillende tools worden gecombineerd.

Soms worden andere soorten tools, bijvoorbeeld kantoorautomatisering- of bevindingenbeheertools, gebruikt om requirements te documenteren of te beheren. Deze tools hebben beperkingen en moeten alleen worden gebruikt wanneer er een gecontroleerd RE-proces is en requirements op elkaar afgestemd en behoorlijk stabiel zijn.

7.2 Toolintroductie (L2)

Het selecteren van een RE-tool verloopt niet anders dan het selecteren van een tool voor een ander doel. Het doel, de context en de requirements moeten worden beschreven alvorens de selectie succesvol kan zijn [Fugg1993].

Een geschikte tool kan pas worden gezocht als de juiste RE-procedures en -technieken zijn ingevoerd. Het introduceren van een tool vereist duidelijke RE-verantwoordelijkheden en -procedures. Bij de introductie van een RE-tool zijn de volgende aspecten relevant:

- Houd rekening met alle kosten tijdens de gehele levensduur, niet alleen de licentiekosten
- Houd rekening met de benodigde inzet van mensen en middelen
- Gebruik proefprojecten om risico's te vermijden
- Evalueer de tool met behulp van gedefinieerde criteria
- Leer medewerkers hoe de tool te gebruiken

Referenties

[AnPC1994] Annie I. Antón, W. Michael McCracken, Colin Potts: Goal Decomposition and Scenario Analysis in Business Process Reengineering. CAiSE (Conference on Advanced Information Systems Engineering), 1994, 94–104.

[BaCC2015] K. Baxter, C. Courage, K. Caine: Understanding Your Users: A Practical Guide to User Research Methods, 2nd edition. Morgan Kaufmann, Burlington, 2015.

[BiSp2003] K. Bittner, I. Spence: Use Case Modelling. Pearson Education, Boston, 2003.

[Bour2009] L. Bourne: Stakeholder Relationship Management: A Maturity Model for Organisational Implementation. Gower Publishing Ltd, Birlington, 2009.

[CaDJ2014] D. Carrizo, O. Dieste, N. Juristo: Systematizing requirements elicitation technique selection. Information and Software Technology 2014, 56(6): 644–669.

[Cock2001] A. Cockburn: Writing Effective Use Cases. Addison-Wesley, Boston 2001.

[Cohn2004] M. Cohn: User Stories Applied – For Agile Software Development. Addison-Wesley, Boston, 2004.

[Coop2004] A. Cooper: The Inmates Are Running the Asylum: Why High-Tech Products Drive Us Crazy and How to Restore the Sanity. Que, Indianapolis, 2004.

[Davi2005] A. M. Davis: Just Enough Requirements Management – Where Software Development Meets Marketing. Dorset House Publishing, New York, 2005.

[Davi1993] A. M. Davis: Software Requirements – Objects, Functions, & States, 2nd edition, Prentice Hall, New Jersey, 1993.

[DeMa1978] T. DeMarco: Structured Analysis and System Specification. Yourdon Press, New York, 1978.

[Fugg1993] A. Fuggetta: A classification of CASE technology. IEEE Computer 1993, 26 (12): 25–38.

[GiGr1993] T. Gilb, D. Graham: Software Inspectie. Addison Wesley, Boston, 1993.

[Glin2019] M. Glinz: Requirements Engineering I. Course Notes, University of Zurich, 2019. https://www.ifi.uzh.ch/en/rerg/courses/archives/hs19/re-i.html#resources. Laatst bezocht in juli 2020.

[Glin2020] M. Glinz: A Glossary of Requirements Engineering Terminology. Version 2.0. https://www.ireb.org/en/downloads/#cpre-glossary. Laatst bezocht in juli 2020.

[GoFi1994] O. Gotel, A. Finkelstein: An Analysis of the Requirements Traceability Problem. 1st International Conference on Requirements Engineering, Colorado Springs, 1994. 94–101.

[GoRu2003] R. Goetz, C. Rupp: Psychotherapy for System Requirements. 2nd IEEE International Conference on Cognitive Informatics (ICCI'03), London, 2003. 75–80.

[GRL2020] Goal oriented Requirement Language. University of Toronto, Canada https://www.cs.toronto.edu/km/GRL. Laatst bezocht in mei 2020.

[Hare1988] D. Harel: On Visual Formalisms. Communications of the ACM 1988, 31 (5): 514–530.

[HoSch2020] S. Hofer, H. Schwentner: Domain Storytelling — A Collaborative Modeling Method. Available from Leanpub, http://leanpub.com/domainstorytelling. Laatst bezocht in juli 2020.

[HuJD2011] E. Hull, K. Jackson, and J. Dick: Requirements Engineering. Springer, 3rd Ed, 2011.

[ISO29148] ISO/IEC/IEEE 29148: Systems and Software Engineering – Life Cycle Processes – Requirements Engineering, International Organization for Standardization, 2018.

[ISO19650] ISO 19650: Organization and Digitization of Information about Buildings and Civil Engineering Works, including Building Information Modelling (BIM)– Information Management Using Building Information Modelling – Part 1 and 2, International Organization for Standardization, 2018.

[ISO25010] ISO/IEC/IEEE25010:2011: Systems and software Quality Requirements and Evaluation (SQuaRE) –– System and software quality models. International Organization for Standardization, Geneva, 2011.

[Jack1995] M. A. Jackson: Software Requirements and Specifications: A Lexicon of Practice, Principles and Prejudices. Addison-Wesley, New York, 1995.

[Jack1995b] M. Jackson: The World and the Machine. 17th International Conference on Software Engineering 1995 (ICSE 1995). 287–292.

[KaeA1984] N. Kano et al.: Attractive quality and must-be quality. Journal of the Japanese Society for Quality Control 1984, 14(2): 39–48. (in het Japans)

[KoTh2017] R. Kohavi, S. Thomke: The Surprising Power of Online Experiments – Getting the most out of A/B and other controlled tests. Harvard Business Review, Sept–Oct 2017: 74–82.

[Kuma2013] V. Kumar: 101 Design Methods – A Structured Approach for Driving Innovation in Your Organization. John Wiley & Sons, Hoboken, 2013.

[Laue2002] S. Lauesen: Software requirements. Styles and Techniques. Addison-Wesley, Harlow, 2002.

[Leff2011] D. Leffingwell: Agile Software Requirements, Lean Requirements Practices for Teams, Programs, and the Enterprise. Addison-Wesley, Boston, 2011.

[LiOg2011] J. Liedtka, T. Ogilvie: Designing for Growth: A Design Thinking Tool Kit For Managers. Columbia University Press, 2011.

[LiSZ1994] H. Lichter, M. Schneider-Hufschmidt, H. Zullighoven: Prototyping in Industrial Software Projects – Bridging the Gap Between Theory and Practice. IEEE Transactions on Software Engineering 1994, 20 (11): 825–832.

[MFeA2019] D. Méndez Fernández, X. Franch, N. Seyff, M. Felderer, M. Glinz, M. Kalinowski, A. Volgelsang, S. Wagner, S. Bühne, K. Lauenroth: Do We Preach What We Practice? Investigating the Practical Relevance of Requirements Engineering Syllabi – The IREB Case. CIbSE 2019: 476–487.

[Moor2014] C. W. Moore: The Mediation Process – Practical Strategies for Resolving Conflicts, 4th edition. John Wiley & Sons, Hoboken, 2014.

[MWHN2009] A. Mavin, P. Wilkinson, A. Harwood, en M. Novak: Easy Approach to Requirements Syntax (EARS). 17th IEEE International Requirements Engineering Conference (RE'09), Atlanta, Georgia, 2009. 317–322.

[OleA2018] K. Olsen et al.: Certified Tester, Foundation Level Syllabus – Version 2018. International Software Testing Qualifications Board, 2018.

[OMG2013] Object Management Group: Business Process Model and Notation (BPMN), version 2.0.2. OMG document formal/2013-12-09 http://www.omg.org/spec/BPMN. Laatst bezocht in juli 2020.

[OMG2017] Object Management Group: OMG Unified Modeling Language (OMG UML), versie 2.5.1. OMG document formal/2017-12-05. https://www.omg.org/spec/UML/About-UML/. Laatst bezocht in juli 2020.

[OMG2019] Object Management Group: OMG Systems Modeling Language (OMG SysML™), Version 1.6. OMG Document formal/19-11-01. https://www.omg.org/spec/SysML/. Laatst bezocht in januari 2022.

[Osbo1979] A. F. Osborn: Applied Imagination, 3rd revised edition. Charles Scribner's Sons, New York, 1979.

[RoRo2012] S. Robertson en J. Robertson: Mastering the Requirements Process, 3rd edition. Addison-Wesley, Boston, 2012.

[Rupp2014] C. Rupp: Requirements-Engineering und Management, 6. Auflage. Hanser, München, 2014. (in het Duits).

[Sdsc2012] Stanford d.school: An Introduction to Design Thinking. Hasso Plattner Institute of Design, Stanford, 2012. https://dschool-old.stanford.edu/groups/designresources/wiki/36873. Laatst bezocht in juli 2020.

[vLam2009] Axel van Lamsweerde: Requirements Engineering: From System Goals to UML Models to Software Specifications. Chichester: John Wiley & Sons, 2009.

[Vole2020] Volere: Requirements Resources. https://www.volere.org. Laatst bezocht in juli 2020.

[WiBe2013] K. Wiegers en J. Beatty: Software Requirements, 3rd edition. Microsoft Press, Redmond, 2013.