

## **2.F.12 CHECKLIST REQUIREMENTS ENGINEERING**

### **DOEL**

Het doel van deze checklist is het bepalen van de sterke en zwakke punten van de huidige werkwijze ten aanzien van requirements engineering. Met behulp van deze checklist kunnen processen zoals het vinden, het vastleggen, het beoordelen en het beheren van requirements worden beoordeeld. Op basis van de resultaten van het checklistonderzoek kunnen concrete verbeterpunten met betrekking tot requirements engineering worden vastgesteld.

### **TOEPASSINGSGEBIED**

Oprachtgevers nemen steeds minder genoegen met systemen die niet aan hun eisen voldoen. Ook de groei van outsourcing in de IT-industrie noodzaakt tot een professionelere omgang met requirements engineering. Het op tijd uitvoeren van IT-projecten met de juiste kwaliteit, functionaliteit en binnen budget begint met goede en eenduidige vastlegging van requirementsspecificaties. In de praktijk blijkt dat projecten veelal niet geheel succesvol zijn of zelfs volledig falen. In diverse onderzoeken is vastgesteld dat het niet goed uitvoeren van requirements engineering één van de belangrijkste faalfactoren van IT-projecten is. Deze checklist biedt concrete handvatten om het proces van requirements engineering te analyseren en verbeterstappen te maken.

Met behulp van deze checklist kan binnen een organisatie of project worden bepaald op welke aspecten het proces van requirements engineering adequaat is ingericht en op welke onderdelen verbeteringen mogelijk c.q. noodzakelijk zijn. Bij de ontwikkeling van deze checklist zijn onder andere de requirementsmethode Volere (Robertson en Robertson, 1999), het boek *Succes met de requirements!* (Arendsens, et al., 2010) en de syllabus van de IREB Foundation (IREB, 2009) als referentiekader gebruikt.

### **AUTEURSGEGEVENS**

Drs. Erik van Veenendaal CISA is sinds 1987 als consultant en manager werkzaam binnen de ICT-industrie. Na een carrière binnen softwareontwikkeling, is hij overgestapt naar het vakgebied softwarekwaliteit. Hij heeft daarbinnen een specialisatie ontwikkeld op het gebied van testen en requirements engineering. Als manager en adviseur is hij betrokken geweest bij een groot aantal automatiseringsprojecten. Tevens heeft hij bij een aantal omvangrijke organisaties in verschillende domeinen meegewerkt aan teststructurering, de implementatie van requirementsprocessen en formele reviews, en testprocesverbeteren.

Erik is de oprichter van Improve Quality Services BV (Improve), een dienstverlenende organisatie op het gebied van testen en requirements engineering ([www.improveqs.nl](http://www.improveqs.nl)). Improve levert adviesdiensten, detachert professionals en verzorgt opleidingen. Naast testen legt Improve zich nadrukkelijk toe op requirements engineering en -management. Improve is aangesloten bij de International Requirements Engineering Board (IREB).

Als universitair docent was Erik bijna tien jaar parttime verbonden aan de faculteit Technology Management van de TU Eindhoven. Van 2005 tot 2009 was hij vice-president van de International Software Testing Qualification Board (ISTQB). Sinds de oprichting is

hij voorzitter van de ISTQB Glossary-werkgroep en editor van de *ISTQB Standard Glossary of terms used in Software Testing*. Erik is de initiatiefnemer, en momenteel vicevoorzitter, van de Belgium Netherlands Testing Qualifications Board (BNTQB). Ten slotte is Erik ook vicevoorzitter van de TMMi Foundation en editor/auteur van het TMMi-model. Voor zijn bijdrage aan het vakgebied testen heeft hij in 2007 de European Testing Excellence Award ontvangen. Specifiek binnen het vakgebied requirements engineering is Erik actief betrokken in diverse werkgroepen bij de International Requirements Engineering Board (IREB), een internationale non-profitorganisatie die zich richt op het verder professionaliseren van het requirementsvakgebied. Erik is te bereiken via [eve@improveqs.nl](mailto:eve@improveqs.nl) en [www.erikvanveenendaal.nl](http://www.erikvanveenendaal.nl).

## INLEIDING

### *Terminologie en definitie*

Het begrip requirements is een van oorsprong Engels begrip. In het Nederlands betekent dit iets als behoeften, eisen, specificatie van eisen, vereisten of wensen, maar geen van deze begrippen dekt volledig de lading. In dit artikel is beschreven wat requirements zijn en wat in softwareontwikkeling het belang is van requirements die van kwalitatief voldoende niveau zijn. In dit artikel wordt de term requirements in plaats van eisen gebruikt, omdat:

- voor ‘requirements engineering’ als overkoepelende term geen Nederlands equivalent beschikbaar is (mocht een Nederlandse term nodig zijn, dan kan wellicht eisenbeheer & -ontwikkeling worden gebruikt);
- veel bedrijven deze (Engelstalige) term hanteren;
- het aantal hits bij een zoekopdracht op internet (alleen Nederlandse sites) overwegend het gebruik van de term requirements in plaats van eisen oplevert.

In de literatuur is geen definitie van requirements beschikbaar die volledig en door eenieder in de ICT-industrie is geaccepteerd. De Standard Glossary of Software Engineering Terminology (IEEE, 1990) geeft de volgende definitie voor een requirement:

- 1 *Een voorwaarde aan of een mogelijkheid van een systeem die een gebruiker nodig heeft om een probleem op te lossen of om een doel te bereiken.*
- 2 *Een voorwaarde aan of een mogelijkheid waarover een systeem of deelsysteem moet beschikken om te voldoen aan de eisen van een contract, standaard, specificatie of een ander formeel opgelegd document.*
- 3 *Een gedocumenteerde beschrijving van een conditie of vermogen zoals in 1 of 2 is verwoord.*

De definitie voor requirements die in dit artikel wordt gehanteerd en inmiddels veel wordt gebruikt, is afkomstig uit de Glossary van de International Software Testing Qualifications Board (Van Veenendaal, 2010):<sup>1</sup>

#### **Requirement**

*Een voorwaarde of voorziening van een gebruiker om een probleem op te lossen of een doel te bereiken, die in een systeem of subsysteem moet worden geïmplementeerd*

<sup>1</sup> De ISTQB Glossary is vertaald in het Nederlands en gedocumenteerd in het *Testwoordenboek* (Van Veenendaal en Posthuma, 2010).

*om aan een contract, standaard, specificatie of een ander formeel opgelegd document te voldoen.*

Een set van requirements wordt gebruikt als input voor de ontwerpfase van een systeemontwikkeltraject. De requirements beschrijven welke elementen en functies noodzakelijk zijn voor een specifiek project. In een requirement worden de benodigde capaciteiten, karakteristieken of kwaliteiten van een systeem geïdentificeerd, die bruikbaar zijn en meerwaarde bieden voor de business en/of de gebruiker. Feitelijk is de requirementsfase de eerste stap in het systeemontwikkelproces.

Alhoewel requirements vaak worden gezien als iets wat specifiek bij waterval- of V-model type ontwikkelmethoden hoort, hebben ook incrementele en interactieve ontwikkelmethoden requirements nodig. Immers, ook bij een increment of iteratie moet duidelijk zijn wat meerwaarde heeft vanuit business- en/of gebruikersoptiek en wat dient te worden geïmplementeerd en met welke prioriteit. Zelfs bij agile ontwikkelmethoden dienen requirements (hier vaak de backlog items genoemd) te worden opgesteld, alhoewel de vorm en het niveau van detaillering verschilt van hetgeen gebruikelijk is in meer traditionele omgevingen. Ook kan veel flexibeler worden omgegaan met de gedefinieerde scope in een agile ontwikkeltraject; wel dient de scope goed te worden gemanaged.

### ***Niveaus van requirements***

Volgens Karl Wieggers (1999) kunnen ten aanzien van requirements voor IT-systemen drie niveaus worden onderscheiden:

- businessrequirements;
- gebruikersrequirements (user requirements);
- systeemrequirements.

Uiteraard is de specifieke invulling in een organisatie of project altijd afhankelijk van de context en de toegevoegde waarde die het gebruik van meerdere requirementsniveaus in die context heeft.

*Businessrequirements* komen voort uit de doelstellingen van een organisatie of een klant die een nieuw systeem wil. Businessrequirements worden bepaald op een hoog abstractieniveau en de requirements die hieruit komen, worden vastgelegd in een visiedocument, scopedocument of een business case.

De *gebruikersrequirements* beschrijven voor een gebruiker de taken of processen die het systeem moet kunnen uitvoeren en worden afgeleid van de businessrequirements. Deze requirements worden veelal vastgelegd in de vorm van use cases (gebruikersscenario's).

#### **Use case**

*Een opeenvolging van transacties in een dialoog tussen een gebruiker en component of systeem met een tastbaar resultaat, waar een gebruiker een persoon kan zijn of iets wat informatie uitwisselt met het systeem (Van Veenendaal en Posthuma, 2010).*

*Systeemrequirements* beschrijven de functionaliteit en de niet-functionele kenmerken van het systeem dat gebouwd gaat worden, en zijn een vertaling vanuit de gebruikersrequirements.

Deze requirements (business-, gebruikers- en systeemrequirements) beschrijven uiteindelijk op enigerlei wijze, op een ander niveau van detail en vanuit een ander gezichtspunt de benodigde en gewenste functionaliteit van het systeem. Functionele requirements beschrijven specifiek gedrag of functies die het systeem dient te vervullen, bijvoorbeeld het opmaken van

een tekst of het moduleren van een signaal. Deze worden ook wel capaciteiten genoemd.

**Functionele requirement**

*Een eis die specificceert welke functionaliteit een component of systeem moet bieden (Van Veenendaal en Posthuma, 2010).*

Daarnaast moet ook rekening worden gehouden met requirements die voortkomen uit kwaliteitseisen (bijvoorbeeld op basis van ISO 9126) en beperkingen. *Kwaliteitseisen* bevatten requirements op het gebied van bijvoorbeeld gebruikersvriendelijkheid, portabiliteit, efficiëntie en betrouwbaarheid. Deze worden ook wel de niet-functionele requirements genoemd. *Beperkingen* (constraints) leggen beperkingen op aan de beschikbare keuzes die een ontwerper heeft.

**Niet-functionele requirement**

*Een eis die niet te maken heeft met functionaliteit, maar met kwaliteitsattributen zoals betrouwbaarheid, efficiëntie, bruikbaarheid, onderhoudbaarheid en portabiliteit (Van Veenendaal en Posthuma, 2010).*

Uiteindelijk dienen alle requirements dusdanig gestructureerd te zijn vastgelegd in een requirementspecificatie, zodat ze voor het ontwerp- en bouwproces voldoende input leveren om succesvol een kwalitatief systeem te kunnen bouwen. Dit impliceert dat de requirements onder andere realiseerbaar en testbaar moeten zijn, gerelateerd moeten zijn aan identificeerbare bedrijfsbehoefte of kansen, en gedefinieerd op een niveau dat voldoende gedetailleerd is voor het systeemontwerp.

**Kwaliteit**

*De mate waarin een component, systeem of proces voldoet aan gespecificeerde eisen en/of gebruikers/klantbehoefte en -verwachtingen (Van Veenendaal en Posthuma, 2010).*

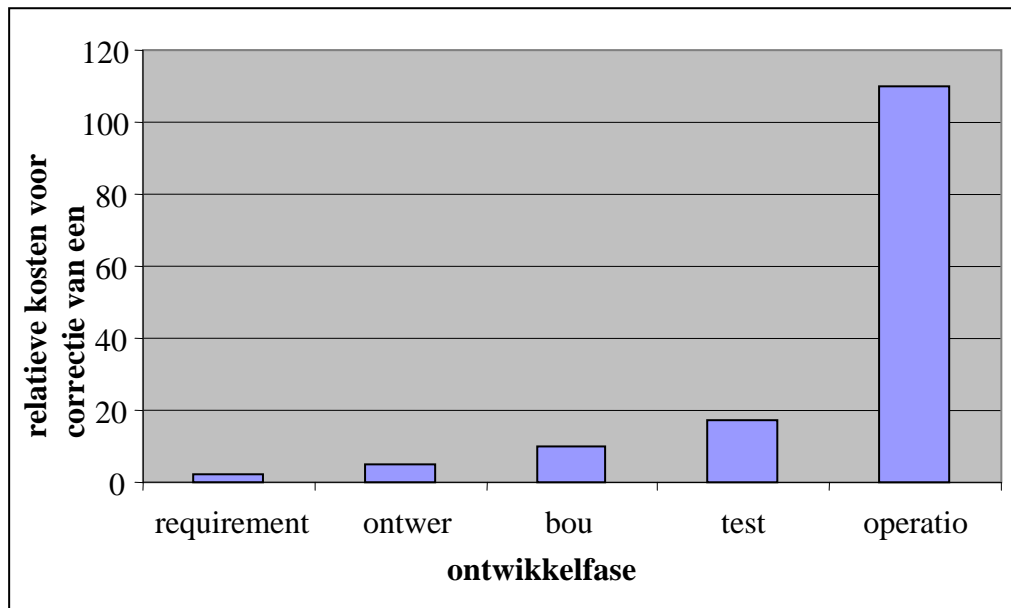
***Het belang van requirements***

Requirements leveren de input voor een softwareontwikkeltraject. Zonder goede input is de kans op goede output nagenoeg nihil: ‘garbage in’ is immers ‘garbage out’. Een groot risico bij het niet voldoen aan de eisen die gesteld mogen worden aan requirements is dat werk moet worden overgedaan. In figuur 1 zijn de relatieve kosten om een fout te herstellen weergegeven per ontwikkelfase. Deze grafiek is het resultaat van onderzoek verricht door Barry Boehm (1981) en staat ook bekend als ‘De wet van Boehm’. Deze wet toont aan dat fouten die pas laat in het ontwikkelproces worden gevonden, vele malen meer inspanning kosten om te herstellen dan wanneer dezelfde fouten gevonden worden in de requirementsfase.

**Requirementsfase**

*De fase in de softwareontwikkelingsfasering waarin de eisen voor een softwareproduct worden gedefinieerd en vastgelegd (Van Veenendaal en Posthuma, 2010).*

De wet toont tevens aan dat reviewinspanningen en expliciete beoordeling van requirementsdocumenten zeer lonende activiteiten zijn die het totale requirementsontwikkeltraject niet vertragen maar juist versnellen.



*Figuur 1. Herstelkosten per ontwikkelfase.*

Wanneer het proces van requirements engineering en -management goed is ingericht, kunnen in de praktijk de volgende voordelen worden behaald:

- minder fouten in requirements en daardoor minder fouten tijdens exploitatie;;
- minder herstelwerk gedurende het ontwikkeltraject;
- sneller ontwikkeltraject;
- minder miscommunicatie tijdens het project;
- minder veranderingen in de scope;
- beter beheerbaar en ordelijk verlopend project;
- efficiënter testtraject;
- hogere tevredenheid van ontwikkelaars, testers, gebruikers en overige belanghebbenden;
- lagere exploitatiekosten.

### ***Veelvoorkomende probleemsituaties***

De volgende situaties (in willekeurige volgorde) komen in de praktijk regelmatig voor in de requirementsfase van een ontwikkelproject.

#### *Onvoldoende betrokkenheid van de (eind)gebruikers*

Als eindgebruikers te weinig worden betrokken of betrokkenheid tonen bij het opstellen van requirements leidt dit tot het te laat en/of onvoldoende naar boven komen van gebruikersrequirements.

#### *Onvoldoende betrokkenheid van de ontwikkelaars*

Als ontwikkelaars te weinig worden betrokken of betrokkenheid tonen bij het opstellen van requirements leidt dit tot het te laat en/of onvoldoende naar boven komen van technische onhaalbaarheden.

#### *Veranderende requirements*

Gedurende het ontwikkelingsproces veranderen (groeien) gebruikerswensen en -eisen.

Meestal gebeurt dit op basis van voortschrijdend inzicht (gebruikers die gedurende het project wijzigingsverzoeken indienen). Dit kan onder andere ondervangen worden door aan het begin van het project duidelijke afspraken te maken over de scope, een gedegen review op het initiële requirementsdocument uit te voeren en het wijzigingsbeheer goed in te richten. Vervolgens moet de projectleider of degene die binnen het project verantwoordelijk is voor het beheer van de requirements tijdens het project er uiteraard wel op toezien dat de scope wordt gehandhaafd en het wijzigingsbeheer conform afspraak wordt uitgevoerd.

#### *Dubbelzinnige requirements*

Als requirements op meerdere manieren uitgelegd kunnen worden, ontstaan verschillen tussen de verwachting van belanghebbenden en de interpretatie van de ontwikkelaars. Dit kan (gedeeltelijk) worden voorkomen door vertegenwoordigers van belanghebbenden de requirements te laten reviewen en duidelijke regels op te stellen waaraan requirements dienen te voldoen.

#### *Onbedoelde functionaliteit*

In een systeem wordt functionaliteit toegevoegd die niet voortkomt uit de requirements. Dit zal vertragend en kostenverhogend werken op het project. Door ervoor te zorgen dat alle gemaakte functionaliteiten terug te traceren zijn naar een afgesproken requirement, kan dit worden voorkomen.

#### *Vergeten belanghebbenden*

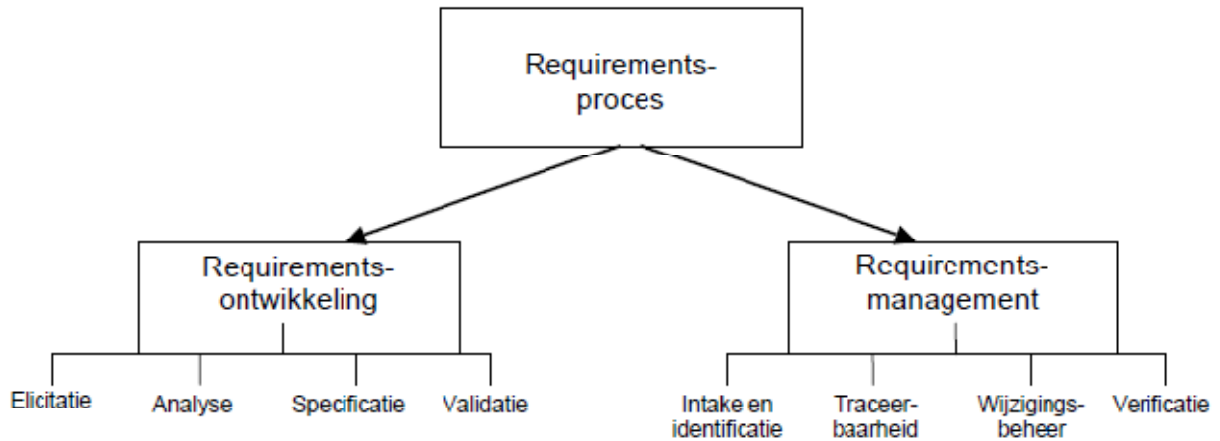
Het is belangrijk om tijdig alle belanghebbenden te identificeren. Wanneer een persoon of groep wordt vergeten, worden daarmee ook de samenhangende noodzakelijke requirements vergeten. Voor het identificeren van een volledige set van belanghebbenden is een gedegen stakeholderanalyse van groot belang.

#### *Onrealistische planning van de requirementsfase*

Requirements opstellen vergt tijd. Hoeveel tijd het vergt om uiteindelijk tot een set requirements te komen die volledig genoeg is om als input te dienen voor het ontwikkelproces, is op voorhand lastig te bepalen. Requirements opstellen is tot op zekere hoogte te vergelijken met een ontdekkingstocht: je weet onderweg pas wat je tegenkomt. Vaak is vanuit het project én de opdrachtgever veel druk om snel met ontwerpen of, nog liever, met bouwen te starten.

#### ***Requirementsprocessen***

De processen requirementsontwikkeling en requirementsmanagement vormen samen het proces requirements engineering.



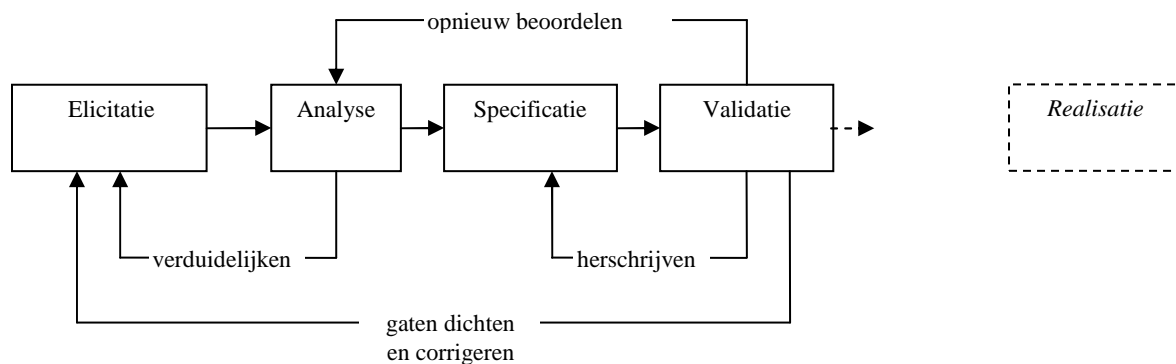
*Figuur 2. Requirementsprocessen.*

### *Requirementsontwikkeling*

Requirementsontwikkeling is een typisch iteratief proces, zelfs als bij de te hanteren ontwikkelmethodiek voor ‘waterval’ is gekozen. Nadere uitwerking van requirements die in het begin zijn gesteld, leidt vaak tot nieuw inzicht, uitbreiding van requirements, nadere detaillering en tot aanpassing van eerder geformuleerde requirements.

#### **Requirementsontwikkeling**

*Het proces gericht op het afleiden en analyseren van de behoeften van de belanghebbenden en het vertalen van deze behoeften naar gespecificeerde producteisen (Van Veenendaal en Cannegieter, 2010).*



*Figuur 3. Requirementsontwikkeling.*

In hoofdlijnen kan het proces als volgt worden geschetst (Wieggers, 1999):

- Requirementselicitering: het communiceren met klanten en gebruikers om te bepalen wat hun behoeften en vereisten zijn, en dit vertalen in requirements. Veelal wordt deze stap gestart met het opstellen van een business case, het bepalen van de scope en een gedegen stakeholderanalyse.
- Requirementsanalyse: bepalen in welke mate de opgestelde requirements onduidelijk, incompleet, onbepaald of tegenstrijdig zijn, en deze onvolkomenheden corrigeren.
- Requirementsspecificatie: het formeel documenteren van de requirements in een requirementsspecificatie. Requirements kunnen in verschillende vormen gedocumenteerd worden, zoals in natuurlijke taal, modellen of een combinatie van

- beide.
- Requirementsvalidatie: samen met klanten en gebruikers evalueren of de set van requirements juist en volledig is in relatie tot hun behoeften en vereisten.

Requirements engineers kunnen verschillende technieken gebruiken om de requirements bij klanten vast te stellen. Veelal wordt gebruikgemaakt van interviews, het initiëren van gebruikersgroepen en of het opstellen van requirementslijsten. Andere technieken omvatten prototyping en use cases. In de praktijk worden de technieken veelal gecombineerd om de juiste requirements op tafel te krijgen.

Requirementsontwikkeling is dus het proces dat zich bezighoudt met het opstellen van de requirements, met als belangrijkste resultaat een overeengekomen set van requirements, vastgelegd in een requirementsbaseline. Deze requirementsbaseline, ook wel eisendossier genoemd, vormt de brug van requirementsontwikkeling naar requirementsmanagement. Bij het ontwikkelen van de requirements ligt de nadruk op analyse, bewustwording en het helder formuleren van de requirements voor de verschillende belanghebbenden.

#### **Baseline**

*Een specificatie die of een softwareproduct dat formeel is beoordeeld of overeengekomen en daarna dient als basis voor verdere ontwikkeling, en alleen gewijzigd kan worden middels een formeel wijzigingsproces (Van Veenendaal en Posthuma, 2010).*

#### *Requirementsmanagement*

Bij requirementsmanagement ligt de nadruk op het beheersen en beheren van de overeengekomen set requirements, het voorkómen van ongewenste scope-uitbreidingen en misverstanden over de inhoud van requirements gedurende het project. Het betreft het managen van alle requirements, ontvangen door of ontstaan binnen het project.

#### **Requirementsmanagement**

*Het proces gericht op het beheren en managen van individuele requirements en requirementsdocumenten, inclusief traceerbaarheid en het bewaken van consistentie tussen tussenproducten en requirements (Van Veenendaal en Cannegieter, 2010).*

Het doel van requirementsmanagement is het beheren van de requirements van het project en het identificeren van inconsistenties tussen die requirements en de projectplannen en werkproducten. Het toepassen van requirementsmanagement is erop gericht dat de set van overeengekomen requirements voor een systeem volledig en juist wordt geïmplementeerd. Een bijkomend voordeel van requirementsmanagement is dat de kans op het op tijd opleveren van het systeem sterk toeneemt doordat ongewenste scopefluctuaties en misverstanden bij acceptatie sterk worden verminderd. Bij aanvang van het project zal de set van dan overeengekomen requirements, de requirementsbaseline, dienen als startpunt voor wat in de ontwikkelfase moet worden gemaakt. Vervolgens wordt deze set van requirements gedurende het verdere ontwikkelingstraject gemanaged op aspecten als inhoud, samenhang, prioriteit, versie, status en traceerbaarheid

Om goed te kunnen managen op deze aspecten dient bij aanvang van requirementsmanagement te worden vastgesteld hoe dit wordt vormgegeven. Bijvoorbeeld: Welke procedures worden gehanteerd? Hoe liggen de verantwoordelijkheden? Welke organisatieonderdelen zijn betrokken? Welke methoden, technieken en tools worden gebruikt? Wat zijn eventuele eerdere ervaringen hiermee? Dit is al van belang bij het ontstaan van de eerste requirements; hoe eerder dit is geregeld hoe beter.



### **Requirementsmanagementtool**

*Een tool die ondersteunt bij het vastleggen van requirements, kenmerken van requirements (bijvoorbeeld prioriteit, verantwoordelijke en bron) en nadere toelichting. De tool ondersteunt bij de traceerbaarheid van requirements op verschillende niveaus en bij het wijzigingsbeheer van requirements. Sommige requirementsmanagementtools bieden ook functionaliteit voor statische analyse, zoals consistentiecontrole en controle op het overtreden van vooraf gedefinieerde regels betreffende het vastleggen van requirements (Van Venedaal en Posthuma, 2010).*

### **Prioriteitsbepaling van requirements**

Het toekennen van de juiste prioriteiten aan requirements zorgt ervoor dat het systeem zo snel mogelijk en zo veel mogelijk waarde oplevert voor de betreffende organisatie. Om deze voordelen te borgen, zal aan de belangrijkste requirements de hoogste prioriteit worden toegekend, zodat hieraan ook de eerste ontwikkelinspanning wordt besteed. Hoe logisch dit ook klinkt, duidelijke sturing hierop blijkt in de praktijk noodzakelijk om te voorkomen dat een te groot deel van ontwikkelcapaciteit wordt besteed aan relatief onbelangrijke requirements. Ter borging dient projectbreed en duidelijk het belang van iedere individuele requirement te worden vastgesteld. Samen met het vaststellen van het belang van requirements vanuit businessperspectief, dient ook te worden gekeken naar technische risico's met betrekking tot het daadwerkelijk kunnen realiseren van een (set van) requirements. Uitwerking van deze requirements moet zo vroeg mogelijk in het project gerealiseerd worden, omdat bij het niet voldoen aan deze requirements mogelijk de gehele business case voor het project in gevaar komt.

Het vaststellen van de prioriteit is typisch een aangelegenheid van alle belanghebbenden. Verschillende technieken bestaan voor het vaststellen van het belang van requirements. Een eenvoudige techniek is 'stikkeren'. Iedere deelnemer ontvangt een beperkt aantal stickers en plakt deze op een requirement. Het is zelfs toegestaan alle stickers op één specifieke requirement te plakken. Deze techniek zorgt ervoor dat volgens vooraf vastgestelde spelregels door alle belanghebbenden tot een rechtvaardige waarde van het belang van requirements kan worden gekomen. Door de spelregels los te koppelen van de discussie over de requirements zelf, wordt voorkomen dat de hardste schreeuwer of degene met de grootste overredingskracht de prioriteit bepaald. De toe te passen techniek en spelregels zullen uiteraard per organisatie verschillen.

De eerste prioriteitsbepaling vindt al plaats tijdens de fase van requirementsontwikkeling. Gedurende het project kan de prioriteit wijzigen. Het requirementsmanagementproces dient dusdanig te zijn ingericht dat het voorziet in het snel en adequaat kunnen doorvoeren van dit type wijzigingen.

### **Certificatie (IREB)**

Inmiddels wordt het vakgebied requirements engineering steeds volwassener. In dit kader is onder andere de International Requirements Engineering Board (IREB) actief. IREB is een internationale non-profitorganisatie die zich richt op het verder professionaliseren van het vakgebied requirements. (Vergelijkbaar met ISTQB op het gebied van testen.) Binnen IREB zijn onder andere experts op het terrein requirements engineering zoals Chris Rupp en Suzanne Robertson lid van de internationale board. Inmiddels heeft IREB een internationaal certificatieprogramma ontwikkeld voor requirements engineering. Het certificatieprogramma mag zich verheugen in een sterk toenemende belangstelling, ook in Nederland en België. Het

certificatieprogramma is onderverdeeld in een drietal niveaus: foundation, advanced en expert. Meer informatie over dit initiatief in te vinden op <http://certified-re.de/en/>.

## **DE CHECKLIST**

Zoals eerder aangegeven is het doel van deze checklist het bepalen van de sterke en zwakke punten van de huidige requirementsprocessen. Indien men de requirementsprocessen wenst te implementeren of verbeteren, dient allereerst te worden bepaald wat de huidige status van requirements engineering en requirementsmanagement is. Om dit te bepalen, wordt een kort onderzoek uitgevoerd. De hierna beschreven checklist kan worden gebruikt bij het uitvoeren van dit onderzoek.

### ***Requirements engineering***

#### *Algemeen*

- 1 Wordt er in de ontwikkelfasering een duidelijk onderscheid gemaakt in een requirementsfase en een systeemontwerpfase?
- 2 Wordt er in de organisatie een onderscheid gemaakt tussen requirements engineering en requirementsmanagement en is dit onderscheid ook voor belanghebbenden duidelijk?
- 3 Wordt er binnen het requirementsproces een onderscheid gemaakt in verschillende niveaus van requirements, bijvoorbeeld requirements voor gebruikers en voor het systeem?
- 4 Wordt er binnen het requirementsproces een duidelijk onderscheid gemaakt in functionele, niet-functionele en randvoorwaarden als typen requirements?

### ***Requirementselicitation en -analyse***

#### *Doelstelling*

- 5 Is de doelstelling (inclusief rationale) van het product gedocumenteerd?
- 6 Is de doelstelling gedocumenteerd in de vorm van een business case?
- 7 Omvat de business case een korte omschrijving van de probleemsituatie die heeft geleid tot het te ontwikkelen product c.q. de gewenste aanpassing?
- 8 Omvat de business case een korte omschrijving van de werkzaamheden die dienen te worden verbeterd met behulp van het product?
- 9 Is het bedrijfsbelang duidelijk beschreven?
- 10 Zijn de voordelen gekwantificeerd in meetbare eenheden?
- 11 Is de business case (inclusief de doelstelling) gereviewed met belanghebbenden?

- 12** Is er een duidelijk commitment ten aanzien van de business case (inclusief de doelstelling) bij de belanghebbenden?

*Belanghebbenden (stakeholders)*

- 13** Heeft een gedegen stakeholderanalyse plaatsgevonden?
- 14** Zijn de klanten en andere stakeholders eenduidig vastgesteld?
- 15** Zijn de ontwikkelaars, de gebruikers en degenen die op enigerlei wijze beïnvloed worden door het product als stakeholders onderkend?
- 16** Is per onderkende stakeholder vastgelegd welke kennisgebieden deze vertegenwoordigt?

*Gebruikers*

- 17** Zijn de gebruikers(groepen) van het product bekend?
- 18** Heeft een contextanalyse plaatsgevonden ten aanzien van de verschillende gebruikers(groepen)?
- 19** Bieden de resultaten van de contextanalyse inzicht in de kenmerken van de gebruikers (onder andere kennis en kunde, opleidingsniveau en fysieke eigenschappen)?
- 20** Bieden de resultaten van de contextanalyse inzicht in de taken van de gebruikers (onder andere frequentie, doelstelling, resultaat en afhankelijkheden)?
- 21** Bieden de resultaten van de contextanalyse inzicht in de werkomgeving van de gebruikers (onder andere werkplekontwerp, softwaresystemen en omgevingsomstandigheden)?

*Beschouwingsgebied (scope)*

- 22** Is de scope van het product vastgesteld en gedocumenteerd, bijvoorbeeld in een contextdiagram?
- 23** Zijn de uit te voeren werkzaamheden op een eenduidige wijze afgebakend?
- 24** Zijn de kernfunctionaliteiten geïdentificeerd?
- 25** Zijn de resultaten (leveringen) van het product aan zijn omgeving vastgelegd?
- 26** Zijn op het niveau van de gebruikersrequirements de aangrenzende processen gedefinieerd?
- 27** Zijn op het niveau van de systeemrequirements de aangrenzende producten en systemen gedefinieerd?

**28** Zijn de belangrijkste gegevensverzamelingen geïdentificeerd?

*Conventies en definities*

**29** Zijn naamgevingsconventies vastgelegd?

**30** Is er een lijst met afkortingen vastgelegd?

**31** Is er een lijst met termen en definities beschikbaar?

**32** Zijn de belangrijkste termen die worden gebruikt door de belanghebbenden en gebruikersgroepen er in opgenomen?

**33** Komen alle gedefinieerde termen voor in minimaal één requirement?

*Bestuderen huidige situatie*

**34** Is gebruikgemaakt van het bestuderen van de huidige situatie voor het verzamelen van requirements?

**35** Is duidelijk wat de doelstelling is van het huidige systeem en het daaraan gekoppelde bedrijfsbeleid?

**36** Is duidelijk wat de sterke punten zijn van het huidige systeem?

**37** Is een model opgesteld c.q. beschikbaar van het huidige systeem?

**38** Zijn gebruikers geobserveerd tijdens het uitvoeren van hun werkzaamheden met het huidige systeem?

*Interviews*

**39** Is gebruikgemaakt van interviews voor het verzamelen van requirements?

**40** Is bij ieder interview duidelijk wat de doelstelling is c.q. wat dient te worden bereikt?

**41** Wordt tijdens interviews gebruikgemaakt van open vragen?

**42** Zijn de interviews voorbereid door middel van een checklist?

**43** Is tijdens de interviews flexibel omgegaan met de checklist en hebben antwoorden geleid tot nieuwe aanvullende vragen?

**44** Wordt tijdens de interviews gebruikgemaakt van mindmaps of een soortgelijke modelleertechniek om de verkregen informatie te structureren?

**45** Wordt gebruikgemaakt van een (standaard) checklist in gebruikersterminologie om de niet-functionele requirements te bepalen (Pinkster, et al., 2004; Trienekens en Van Veenendaal, 1997)?

### *Brainstormen*

- 46 Is gebruikgemaakt van brainstormen voor het verzamelen van requirements?
- 47 Is de brainstormgroep breed samengesteld zowel qua domein als kennis?
- 48 Wordt de brainstormsessie gestart met een duidelijke probleemstelling, eventueel ondersteund door een contextdiagram?
- 49 Worden alle ideeën en suggesties tijdens de brainstorm als positieve bijdrage gewaardeerd?
- 50 Wordt tijdens een brainstormsessie gebruikgemaakt van een onafhankelijke facilitator?
- 51 Worden de ideeën en resultaten van de brainstormsessie gedocumenteerd?
- 52 Wordt per idee tevens vastgelegd wiens idee het was, de zogenaamde bron?
- 53 Krijgt ieder idee een uniek (requirements)nummer?
- 54 Worden de ideeën pas in een latere fase uitgewerkt tot formele requirements en vervolgens geverifieerd met de persoon die het idee heeft aangeleverd?
- 55 Maken deelnemers tijdens de brainstorm gebruik van elkaar ideeën?
- 56 Worden eventueel ideeën ten aanzien van het project of het ontwerp in een afzonderlijk document vastgelegd?

### *Use case workshop*

- 57 Vindt de use cases workshop plaats op basis van een contextdiagram?
- 58 Worden de use cases vastgelegd in een beperkt aantal stappen, bijvoorbeeld zes?
- 59 Wordt tevens gebruikgemaakt van mis-use cases in het kader van onder andere beveiligingseisen?
- 60 Worden bij de use cases de actor (gebruiker, ander systeem), de uitgangssituatie alsmede het resultaat vastgelegd?
- 61 Is het resultaat dusdanig concreet gedefinieerd dat dit testbaar is?
- 62 Wordt in de use case workshop voldoende aandacht gegeven aan niet-functionele requirements?

### *Requirementsspecificatie*

#### *Functionele requirements*

- 63 Zijn de requirements gekoppeld aan de use cases?
- 64 Zijn er templates voor het documenteren van individuele requirements?
- 65 Wordt gebruikgemaakt van enkelvoudige en eenvoudige zinnen?
- 66 Wordt gebruikgemaakt van direct taalgebruik om requirements te specificeren?
- 67 Zijn de requirements vrij van oplossingen en ontwerpen?
- 68 Heeft elke requirement een uniek nummer?
- 69 Is bij elke requirement de rationale vastgelegd?
- 70 Is bij elke requirement de bron vastgelegd?
- 71 Zijn de requirements voorzien van een prioriteitsstelling?
- 72 Wordt gebruikgemaakt van zogenoemde fit-criteria om de requirement meetbaar en testbaar te maken?
- 73 Zijn er specifieke regels ('rule set') opgesteld waaraan individuele requirements moeten voldoen? Bijvoorbeeld:
- traceerbaarheid naar een hoger liggende requirement of brondocument;
  - extern consistent met de hoger liggende requirements;
  - annotatie (toelichtende tekst) wordt duidelijk afwijkend gedocumenteerd van de requirements;
  - intern consistent met requirements van hetzelfde niveau;
  - er is geen sprake van samengestelde requirements;
  - kort en bondig gespecificeerd;
  - voorzien van een prioriteit;
  - testbaar en meetbaar;
  - eenduidig (is er een lijst met niet te gebruiken, zogenoemde fuzzy, termen?);
  - technisch haalbaar.
- 74 Zijn de regels specifiek en concreet gemaakt per requirementsniveau?

*Niet-functionele requirements*

- 75 Wordt voldoende aandacht gegeven aan niet-functionele requirements?
- 76 Wordt gebruikgemaakt van een raamwerk voor de terminologie van niet-functionele kwaliteitsattributen, bijvoorbeeld ISO 9126 (2001)?
- 77 Zijn de niet-functionele requirements gekoppeld aan de functionele requirements en/of aan specifieke use cases?
- 78 Zijn de niet-functionele requirements specifiek, meetbaar en testbaar gemaakt door middel van fit-criteria?

### *Requirementspecificatie*

- 79** Is gebruikgemaakt van een template voor het opstellen van het requirementsspecificatiedocument?
- 80** Bestaat de template uit een drietal hoofdonderdelen:
- introductie (doelstelling, stakeholders, gebruikers, conventies en definities);
  - algemene beschrijving (contextdiagram, kernfunctionaliteiten, randvoorwaarden, acceptatiecriteria);
  - requirements (use cases, functionele requirements, niet-functionele requirements)?
- 81** Zijn de randvoorwaarden in voldoende mate beschreven, bijvoorbeeld ten aanzien van werkomgeving, doorlooptijd, interfaces, technische omgevingen, ontwerprestrictie?
- 82** Wordt gebruikgemaakt van modellen om requirements te verhelderen of als middel om requirements te documenteren, bijvoorbeeld DataFlowDiagram, EntityRelationDiagram, Class diagram, Use case diagram, of Sequence charts?
- 83** Zijn acceptatiecriteria opgesteld op systeemniveau?
- 84** Zijn de requirements geprioriteerd?
- Is dit gebeurd in samenwerking met alle belanghebbenden?

### *Requirementsverificatie en validation*

- 85** Zijn de requirements gevalideerd en geverifieerd door middel van formele reviews?
- 86** Is ten behoeve van de validatie gebruikgemaakt van een walkthrough met de belangrijkste stakeholders?
- 87** Wordt bij de walkthrough gebruikgemaakt van prototyping om requirements te verduidelijken of duidelijk te krijgen?
- 88** Wordt tijdens de walkthrough expliciet aandacht gegeven aan de begripsvorming en de betekenis van de requirements?
- 89** Is ten behoeve van de verificatie gebruikgemaakt van een inspectie?
- 90** Wordt bij de inspectie gereviewed ten opzichte van de formele set van requirementsregels ('rule set')?
- 91** Vindt bij zowel de walkthrough als de inspectie een gedegen opvolging plaats zodat er zekerheid bestaat dat de geïdentificeerde fouten en opmerkingen correct en volledig zijn verwerkt?
- 92** Wordt aan het einde van de reviews expliciet commitment verkregen bij de deelnemers met betrekking tot de requirements en de implicaties ervan?
- 93** Vindt een follow-upcontrole plaats dat de bevindingen goed en correct zijn verwerkt?

### **Requirementsmanagement**

- 94 Worden de requirements pas in beheer genomen als ze met goed resultaat zijn gereviewed en derhalve voldoen aan de regels?
- 95 Worden de reviews gebruikt als een formele kwaliteitspoort alvorens (een set van) requirements naar een volgende fase kunnen gaan?
- 96 Worden de wijzigingen aan de requirements gedurende het project conform een vastgestelde procedure uitgevoerd?
- 97 Vindt een gedegen impactanalyse plaats alvorens wijzigingen op requirements te accepteren en door te voeren?
- 98 Wordt traceerbaarheid in twee richtingen onderhouden tussen de requirements enerzijds en de projectplannen en (tussen)producten anderzijds?
- 99 Worden inconsistenties tussen projectplannen en de (tussen)producten enerzijds en de requirements anderzijds geïdentificeerd?
- 100 Is ter ondersteuning van de activiteiten van het requirementsmanagement tooling beschikbaar?
- 101 Wordt de status van de individuele requirements bewaakt?
- 102 Vindt versiebeheer plaats op de individuele requirements?
- 102 Vindt versiebeheer plaats op de requirementsdocumenten?

### **Literatuur**

- Arendsen, M., H.J.J. Cannegieter, A. Grund, P. Heck, S. de Klerk & J. Zandhuis (2010), *Succes met de requirements!*, 2e herziene druk, Sdu, Den Haag.
- Boehm B.W. (1981), *Software Engineering Economics*, Prentice-Hall.
- Chrissis M.B., M. Konrad & S. Shrum (2004), *CMMI, Guidelines for Process Integration and Product Improvement*, Addison Wesley.
- IEEE (1990), IEEE 610.12, *Standard Glossary of Software Engineering Terminology*, IEEE Standards Board.
- IREB (2009), IREB Foundation syllabus, Certified professional for requirements engineering – Foundation level – Version 2.0, International Requirements Engineering Board.
- ISO (2001), ISO/IEC 9126-1, *Software Engineering – Software Product Quality – Part 1: Quality Characteristics and sub-characteristics*, International Organization of Standardization.
- Pinkster, I., B. van der Burgt, D. Janssen & E. van Veenendaal (2004), *Successful Test Management*, Springer.
- Robertson S. & J. Roberson (1999), *Mastering the Requirements Process*, Addison-Wesley.
- SYSQA (2010), *Requirements; een introductie*, interne publicatie SYSQA B.V.
- Trienekens J. & E. van Veenendaal (1997), *Software Quality from a Business Perspective*, Kluwer Bedrijfsinformatie.
- Veenendaal, E. van, ed. (2010), *Standard glossary of terms used in Software Testing – Version 2.1*, International Software Testing Qualifications Board.
- Veenendaal, E.P.W.M. van & H.J.J. Cannegieter (2010), *De kleine TMMi*, Sdu, Den Haag.
- Veenendaal, E.P.W.M. van & M. Posthuma (2010), *ISTQB Testwoordenboek – Engelse en Nederlandse definities*, Tutein Nolthenius, Den Bosch.
- Wieggers, K.E. (1999), *Software Requirements*, Microsoft Press.