Nederlandse praktijkrichtlijn

NPR 5326

(nl)

Risicobeheersing bij ontwikkeling en onderhoud van maatwerksoftware

Risk management during development and maintenance of custom software

Vervangt NPR 5326:2018 Ontw.

ICS 35.100.05

oktober 2019

Normcommissie 381007 'Software and systems engineering'

©2019 Koninklijk Nederlands Normalisatie-instituut
Postbus 5059, 2600 GB Delft
Telefoon (015) 2 690 390, www.nen.nl

Inhoud

Voorwoord **4**

1 Onderwerp en toepassingsgebied **5**

2 Verwijzingen **5**

3 Termen en definities **6**

4 Afkortingen **15**

5 Toelichting begrippen **15**
5.1 Het begrip risico 15
5.2 Het begrip beheersmaatregel 16

6 Risico's **17**
6.1 Algemeen 17
6.2 Productgerelateerde risico's 17
6.3 Projectgerelateerde risico's 18

7 Beheersmaatregelen **23**
7.1 Algemeen 23
7.2 Projectgerichte beheersmaatregelen 24
7.3 Organisatiegerichte beheersmaatregelen 35

Bijlage A Overzicht risico's en beheersmaatregelen **38**

Bijlage B Assessmentinstrument **40**

Bibliografie **42**

Voorwoord

In de serie Nederlandse praktijkrichtlijnen verschijnen publicaties van informatief karakter, zoals toelichtingen op normen, constructieve mogelijkheden, werkmethoden en productiegegevens. Aan deze publicaties mag geen normatieve waarde worden toegekend.

ICT-projecten kennen veel risico's. ICT-projecten kampen niet zelden met vertraging, budgetoverschrijdingen en een eindresultaat van te lage kwaliteit. Dit geldt onder andere voor ICT-projecten van de Nederlandse overheid. De commissie Elias concludeerde in 2014 in haar eindrapport[1] bijvoorbeeld: "De rijksoverheid heeft haar ICT-projecten niet onder controle". Maar ook het bedrijfsleven[2] en de zorg[3] kampen met dergelijke problemen.

ICT-projecten waarin maatwerksoftware wordt ontwikkeld en/of onderhouden, lopen veelal extra risico bovenop de risico's die toch al gemoeid zijn met ICT-projecten in het algemeen[4]. Dit lijkt onder meer te worden veroorzaakt door de grotere omvang en complexiteit van maatwerksoftwareprojecten en door het optreden van risico's die inherent zijn aan softwareontwikkeling en die door organisaties in onvoldoende mate worden gemitigeerd. Dat terwijl veel risico's bij de ontwikkeling van software op maat bekend zijn en er ook voor veel risico's passende beheersmaatregelen beschikbaar zijn.

NPR 5326 beschrijft beheersmaatregelen voor een deel van de risico's die inherent zijn aan softwareontwikkeling op maat. Het doel van deze NPR is dat belanghebbenden tijdens de ontwikkeling van maatwerksoftware kunnen beschikken over een verzameling passende beheersmaatregelen. Omdat de opgenomen beheersmaatregelen elk voor zich gangbaar zijn, vormt deze verzameling een logisch startpunt voor het borgen van de kwaliteit van maatwerksoftwareontwikkeling.

Doelgroepen die belang hebben bij het mitigeren van risico's tijdens het ontwikkelen van software op maat, zijn zowel de opdrachtgevers en opdrachtnemers van de softwareontwikkeling als de eindgebruikers en beheerders van de ontwikkelde software.

Deze NPR is opgesteld door de normcommissie 381 007 'Software and systems engineering', na voorbereiding door de 'NPR 5326-schrijfgroep'.

---

1 *Parlementair onderzoek naar ICT-projecten bij de overheid,* Tweede Kamer, vergaderjaar 2014–2015, 33 326, nr. 5 (https://zoek.officielebekendmakingen.nl/kst-33326-5.html).
2 *4 ICT-fiasco's in het bedrijfsleven,* Geert Wit, Computerworld, 13 oktober 2014.
3 *Falende IT serieus probleem in de zorg,* Thijs Doorenbosch, AG Connect, 23 oktober 2017.
4 *Delivering large-scale IT projects on time, on budget, and on value,* Michael Bloch et al, McKinsey, oktober 2012 (https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/delivering-large-scale-it-projects-on-time-on-budget-and-on-value).

Risicobeheersing bij ontwikkeling en onderhoud van maatwerksoftware

1 Onderwerp en toepassingsgebied

Het onderwerp van deze Nederlandse praktijklijn (NPR) betreft risicobeheersing bij de ontwikkeling en het onderhoud van maatwerksoftware en wordt als volgt afgebakend. NPR 5326:

1) omschrijft vaak voorkomende risico's bij de ontwikkeling en het onderhoud van maatwerksoftware;

2) omschrijft mogelijke beheersmaatregelen voor de vaak voorkomende risico's;

3) definieert de hierbij horende:

   a) termen;

   b) activiteiten, faciliteiten en rollen benodigd voor de beheersmaatregelen;

   c) eigenschappen van producten en processen;

   d) normen, metingen, testen en toetsen aan/van de eigenschappen van producten en processen.

2 Verwijzingen

Naar de volgende documenten wordt in de tekst zo verwezen dat de bepalingen ervan geheel of gedeeltelijk ook voor dit document gelden. Bij gedateerde verwijzingen is alleen de aangehaalde editie van toepassing. Bij ongedateerde verwijzingen is de laatste editie van het document (met inbegrip van eventuele wijzigingsbladen en correctiebladen) waarnaar is verwezen van toepassing.

NPR 5325, *Overdragen van software*

NPR-ISO Guide 73, *Risicomanagement – Verklarende woordenlijst*

NEN-ISO/IEC 25010, *Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models*

NEN-ISO/IEC/IEEE 12207, *Systems and software engineering – Software life cycle processes*

ISO/IEC 16085, *Systems and software engineering – Life cycle processes – Risk management*

ISO/IEC TR 29110-5-3, *Systems and software engineering — Lifecycle profiles for Very Small Entities (VSEs) — Part 5-3: Service delivery guidelines*

EN 301 549, V2.1.2 (2019-08), *Accessibility requirements for ICT products and services*

Carr, M. J., Konda, S. L., Monarch, I., Ulrich, F. C., & Walker, C. F., *Taxonomy-based risk identification* (No. CMU/SEI-93-TR-06). Carnegie-Mellon University, Software Engineering Institute, 1993. https://resources.sei.cmu.edu/asset_files/TechnicalReport/1993_005_001_16166.pdf

Ries, E., *The Lean Startup: How Relentless Change Creates Radically Successful Businesses*, Penguin Books Ltd., 2011.

Ken Schwaber, Jeff Sutherland, *The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game,* November 2017. https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf

Web Content Accessibility Guidelines (WCAG) 2.1, W3C Recommendation, 05 June 2018. https://www.w3.org/TR/WCAG21/

3 Termen en definities

Voor de toepassing van deze NPR gelden de volgende termen en definities.

3.1
acceptance test-driven development
ATDD
ontwikkelmethode waarbij teamleden met diverse achtergronden (ontwikkelaars, testers en business-analisten) gezamenlijk de acceptatietesten schrijven voorafgaand aan de ontwikkeling van de desbetreffende functionaliteit

3.2
Agile
manier van productontwikkeling die wordt gekenmerkt door directe communicatie tussen teamleden onderling en tussen teams en opdrachtgevers, klanten en eindgebruikers, en door het samenwerken in korte, overzichtelijke perioden (iteraties) waarbij aan het eind van elke periode een volgende, bruikbare versie van het product wordt opgeleverd

3.3
Algemene Verordening Gegevensbescherming
General Data Protection Regulation
Europese verordening die de regels voor de verwerking van persoonsgegevens door particuliere bedrijven en overheidsinstanties in de hele Europese Unie standaardiseert

3.4
backlog
lijst met taken, ook wel items genoemd, die moeten worden uitgevoerd tijdens de ontwikkeling van het product

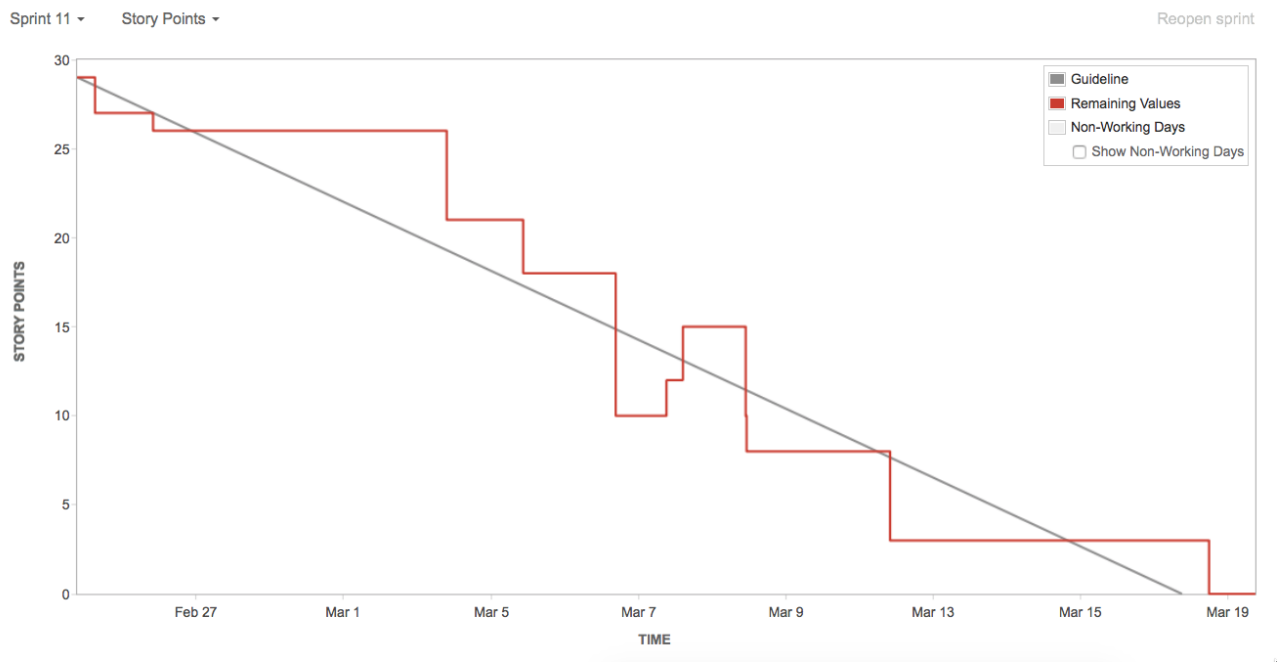
3.5
behavior-driven development
BDD
ontwikkelmethode waarbij teamleden met diverse achtergronden (ontwikkelaars, testers en business-analisten) gezamenlijk het gedrag van de beoogde functionaliteit beschrijven voorafgaand aan de ontwikkeling van de desbetreffende functionaliteit

3.6
beveiligingstest
penetratietest/pentest
test die handmatig en geautomatiseerd wordt uitgevoerd om de beveiliging van een applicatie of een systeem te toetsen

Opmerking 1 bij de term: Afhankelijk van de hoeveelheid informatie over het te testen systeem die de beveiligingstester heeft bij het uitvoeren van de test, wordt gesproken van blackbox- (minimale informatie), greybox- (dezelfde kennis en toegang als een gewone gebruiker) of whitebox-beveiligingstesten (toegang tot het gehele systeem, de documentatie en broncode).

3.7
broncode
source code
voor mensen leesbare tekst die door de programmeur in een programmeertaal is geschreven die kan worden vertaald naar uitvoerbare (executable) code zodat het programma door een computer kan worden uitgevoerd

3.8
burndown chart
grafische weergave voor het transparant maken van de omvang van het uitgevoerde werk versus de omvang van de nog uit voeren werkzaamheden



BRON: Niessink, F., ICTU, 's-Gravenhage

Figuur 1 — Voorbeeld van een burndown chart

Opmerking 1 bij de term: Dit kan per sprint worden weergegeven, maar ook per release of iteratie.

Opmerking 2 bij de term: Als omvangsmaat kunnen bijvoorbeeld user story-punten (3.51) of functiepunten (3.21) worden gebruikt.

3.9
business impact-analyse
BIA
wordt in het kader van het business continuity management (BCM) binnen een organisatie gebruikt om de kritieke processen van de niet-kritieke processen te scheiden

Opmerking 1 bij de term: Door de ICT-systemen, gebouwen, medewerkers, externe partijen, enz. die noodzakelijk zijn voor het uitvoeren van deze kritieke processen te bepalen, kunnen ook de kritieke middelen worden vastgesteld.

3.10
code review
activiteit waarbij één of meerdere ontwikkelaars de kwaliteit van (een deel van) de broncode vaststellen door deze door te nemen, vaak vlak voor het samenvoegen van gewijzigde broncode met de rest van de broncode of bij een incident

3.11
configuratiemanagementdatabase
CMDB
database die door een organisatie wordt gebruikt om informatie over de in gebruik zijnde hardware en software op te slaan

3.12
Data Protection Impact Assessment
DPIA
in de Algemene Verordening Gegevensbescherming (AVG) omschreven instrument om vooraf de privacyrisico's van een gegevensverwerking in kaart te brengen en om daarna maatregelen te kunnen nemen om de risico's te verkleinen

3.13
deliverable
(software)producten, diensten en resultaten die in een project worden geproduceerd en opgeleverd aan een of meer partijen buiten het project

3.14
Delphi-methode
onderzoeksmethode waarbij de meningen van meerdere experts wordt gevraagd ten aanzien van een onderwerp waar geen consensus over bestaat en waarbij door het (anoniem) terugkoppelen van de antwoorden van de andere experts wordt geprobeerd in een aantal rondes tot consensus te komen

3.15
design thinking
methodologie om (zeer complexe) problemen op te lossen waarbij deze vanuit de menselijke behoeften worden gedefinieerd

Opmerking 1 bij de term: In design thinking worden oplossingsrichtingen bepaald met brainstormsessies, waarbij prototypes worden vervaardigd om de beoogde oplossing te testen in de praktijk.

3.16
DevOps
samentrekking van 'development' en 'operations', verwijst naar werkwijzen die de nadruk leggen op samenwerking en communicatie tussen softwareontwikkelaars en IT-beheerders bij het automatiseren van het proces van het opleveren en beheren van software en het aanpassen van de IT-infrastructuur

[BRON: ISO/IEC TR 29110-5-3:2018, 7.1]

3.17
distributed denial of service attack
DDoS-aanval
poging om een computersysteem onbruikbaar te maken voor gebruikers door het systeem vanaf verschillende andere computers zodanig zwaar te belasten met netwerkverkeer dat het systeem traag of onbereikbaar wordt voor gewone gebruikers

3.18
doelstelling
vooraf vastgesteld te bereiken resultaat

Opmerking 1 bij de term: Om een doelstelling te behalen zijn in het algemeen meerdere middelen en activiteiten nodig waarvan maatwerksoftware er één kan zijn.

3.19
Earned Value Analysis
projectbeheersingsmethodiek die realisatie van projectresultaten, voortgang in tijd en kosten met elkaar in verband brengt

3.20
Extreme Programming
XP
vorm van Agile-softwareontwikkeling waarbij werkwijzen voor iteratief plannen en werken zijn gecombineerd met technische werkwijzen, zoals test-gedreven ontwerp en pair programming

3.21
functiepunt
objectieve maat voor de omvang van een systeem gebaseerd op het tellen van voor gebruikers relevante functies en (logische) gegevensverzamelingen

3.22
functiepuntanalyse
FPA
methode om de functionele omvang van een informatiesysteem te meten waarbij een functiepunt (3.21) als eenheid wordt gebruikt

3.23
gebeurtenis
optreden van of wijzigen van een bepaalde combinatie van omstandigheden

Opmerking 1 bij de term: Een gebeurtenis kan een- of meerledig zijn en kan diverse oorzaken hebben.

Opmerking 2 bij de term: Een gebeurtenis kan er ook uit bestaan dat iets niet gebeurt.

Opmerking 3 bij de term: Een gebeurtenis kan soms ook worden aangeduid als een ‘incident’ of ‘ongeval’.

Opmerking 4 bij de term: Een gebeurtenis die geen gevolgen (3.24) heeft, kan ook als een ‘bijna-ongeluk’ (‘near miss’, ‘near hit’, ‘close call’) of ‘incident’ worden omschreven.

[BRON: NPR-ISO Guide 73:2009, 3.5.1.3, gewijzigd – (3.24) is toegevoegd.]

3.24
gevolg
uitkomst van een gebeurtenis (3.23) waardoor doelstellingen (3.18) worden beïnvloed

Opmerking 1 bij de term: Een gebeurtenis kan een reeks gevolgen hebben.

Opmerking 2 bij de term: Een gevolg kan zeker of onzeker zijn en kan het behalen van doelstellingen positief of negatief beïnvloeden.

Opmerking 3 bij de term: Gevolgen kunnen kwantitatief of kwalitatief worden uitgedrukt.

[BRON: NPR-ISO Guide 73:2009, 3.6.1.3, gewijzigd – (3.23) en (3.18) zijn toegevoegd.]

3.25
integratietest
test waarbij functionaliteit wordt getoetst die afhankelijk is van meerdere softwarecomponenten en waarbij wordt getoetst of deze componenten goed met elkaar integreren

3.26
kwaliteit
mate waarin iets goed is; gesteldheid, hoedanigheid, aard

[BRON: Van Dale online woordenboek, https://www.vandale.nl, geraadpleegd 9 augustus 2019]

Opmerking 1 bij de term: Voor de kwaliteit van software en IT-systemen biedt NEN-ISO/IEC 25010 een onderverdeling in aspecten, zoals:

— betrouwbaarheid;

— beveiligbaarheid;

— bruikbaarheid;

— functionele geschiktheid;

— onderhoudbaarheid;

— overdraagbaarheid;

— prestatie-efficiëntie;

— uitwisselbaarheid.

3.27
lean startup
verzameling werkwijzen om ondernemers te helpen de kansen op een succesvolle startup te vergroten

[BRON: Ries, 2011, p. 27]

Opmerking 1 bij de term: De kern van lean startup is snel te leren wat werkt en wat niet. Eén van de werkwijzen is bijvoorbeeld het bouwen van een simpel prototype in plaats van met een uitgewerkt idee van je nieuwe product of dienst. Daarmee wordt uitvoerig getest of het in de smaak valt (meten). Het testen vindt plaats direct bij de klant, wat zorgt voor feedback en helpt bij de doorontwikkeling van je product (leren). Telkens wordt 'bouwen-meten-leren' toegepast. Dit versnelt de innovatie en beperkt de kosten.

3.28
minimum viable product
MVP
minimale functionele en niet-functionele eigenschappen die een product moet hebben om in gebruik te kunnen worden genomen

3.29
ontwikkelaar
persoon of organisatie die activiteiten uitvoert die direct bijdragen aan het ontwikkelen van software op maat

Opmerking 1 bij de term: Activiteiten van de ontwikkelaar van software op maat zijn onder andere het opstellen en analyseren van functionele en niet-functionele systeem- en software-eisen, ontwerpen, programmeren en testen.

Opmerking 2 bij de term: Ontwerpers, programmeurs en testers zijn dus allemaal ontwikkelaars.

3.30
ontwikkelpijplijn
bouwstraat
set aan tools gericht op het beheerst en gecontroleerd bouwen van een pakket waarmee een applicatie in gebruik kan worden genomen

Opmerking 1 bij de term: Als onderdeel hiervan worden diverse testen uitgevoerd om de kwaliteit vast te stellen en te beoordelen.

3.31
ontwikkelteam
team voor de ontwikkeling en/of het onderhoud van software

Opmerking 1 bij de term: De opdrachtgever, producteigenaar en toekomstige beheerder kunnen onderdeel uitmaken van het ontwikkelteam.

Opmerking 2 bij de term: Ontwikkelteams kunnen functioneel zijn ingedeeld (bijvoorbeeld een team met ontwerpers, een team met programmeurs, een team met testers) of multidisciplinair (elk team heeft bijvoorbeeld ontwerp-, programmeer- én testexpertise).

3.32
opdrachtgever
natuurlijk persoon, organisatie of deel van een organisatie die maatwerksoftware laat ontwikkelen en/of onderhouden

Opmerking 1 bij de term: Een opdrachtgever kan een interne of een externe partij zijn ten opzichte van de organisatie van de opdrachtnemer.

3.33
opdrachtnemer
natuurlijk persoon, organisatie of deel van een organisatie die maatwerksoftware ontwikkelt en/of onderhoudt in opdracht van een opdrachtgever (3.32)

Opmerking 1 bij de term: Een opdrachtnemer kan een interne of een externe partij zijn ten opzichte van de organisatie van de opdrachtgever (3.32).

3.34
performancetest
test gericht op het vaststellen van de performance van de applicatie en de omgeving waarop de applicatie zijn werk doet

Opmerking 1 bij de term: Veelal wordt onderscheid gemaakt tussen load-, stress- en duurtesten. Loadtesten simuleren een normale belasting op het systeem. Stresstesten laten de belasting toenemen om de maximale belasting vast te stellen waarbij het systeem nog functioneert. Duurtesten belasten het systeem voor langere tijd om zo geheugenlekken of andere problemen te ontdekken die zich pas na enige tijd manifesteren.

3.35
product breakdown structure
PBS
hulpmiddel dat de componenten van een bepaald product of systeem beschrijft in de vorm van een hiërarchie

Opmerking 1 bij de term: De PBS begint met het eindproduct bovenaan de hiërarchie en daaronder de hoofdcomponenten, die elk ook weer kunnen zijn onderverdeeld in componenten, enz.

3.36
product owner
medewerker van de opdrachtgever die verantwoordelijk is voor het opstellen en prioriteren van eisen aan en acceptatie van de ontwikkelde maatwerksoftware

3.37
proof of concept
PoC
realisatie van een bepaalde methode of idee om de uitvoerbaarheid ervan aan te tonen, of door een demonstratie na te gaan of een concept of theorie praktisch potentieel heeft

Opmerking 1 bij de term: Een PoC is meestal klein en kan al dan niet compleet zijn.

3.38
prototype
model van een product dat meestal slechts een paar aspecten ervan simuleert en compleet kan verschillen van het uiteindelijke product

Opmerking 1 bij de term: Het maken van prototypen heeft verschillende voordelen: de softwareontwerper en de uitvoerder kunnen in het begin van het project waardevolle feedback krijgen van de gebruikers. Het geeft de ontwikkelaars ook enig inzicht in de juistheid van initiële projectschattingen en of de voorgestelde deadlines en mijlpalen met succes kunnen worden gehaald.

3.39
regressietest
test waarbij wordt gevalideerd of alle (functionele) onderdelen nog goed functioneren nadat er wijzigingen in de software zijn aangebracht

3.40
risico
effect van onzekerheid op het behalen van doelstellingen

[BRON: NPR-ISO Guide 73:2009, 1.1]

3.41
risicobehandeling
proces waarmee een risico (3.40) wordt gewijzigd

Opmerking 1 bij de term: Risicobehandeling kan het volgende omvatten:

— het vermijden van het risico door te besluiten de activiteit waardoor het risico wordt veroorzaakt, niet uit te voeren of voort te zetten;

— het nemen of verhogen van het risico teneinde een kans te benutten;

— het wegnemen van de risicobron (zie 3.5.1.2 van NPR-ISO Guide 73);

— het veranderen van de waarschijnlijkheid (zie 3.6.1.1 van NPR-ISO Guide 73);

— het veranderen van de gevolgen (zie 3.6.1.3 van NPR-ISO Guide 73);

— het delen van het risico met een of meer andere partijen (met inbegrip van contracten en risicofinanciering (zie 3.8.1.4 van NPR-ISO Guide 73);

— het behouden van het risico op basis van een onderbouwde keuze.

Opmerking 2 bij de term: Een risicobehandeling die is gericht op negatieve gevolgen, wordt soms aangeduid met risicovermindering, risico-eliminatie, risicopreventie of risicoreductie.

Opmerking 3 bij de term: Door risicobehandeling kunnen nieuwe risico's ontstaan of bestaande risico's worden gewijzigd.

[BRON: NPR-ISO Guide 73:2009, 3.8.1, gewijzigd – (3.40) is toegevoegd.]

3.42
risicobeheersmaatregel
maatregel waarmee een risico (3.40) wordt gewijzigd

Opmerking 1 bij de term: Een beheersmaatregel kan elke vorm van proces of beleid, een voorziening, werkwijze of andere maatregel zijn waarmee het risico wordt gewijzigd.

Opmerking 2 bij de term: Een beheersmaatregel heeft mogelijk niet altijd het beoogde of veronderstelde effect.

[BRON: NPR-ISO Guide 73:2009, 3.8.1.1, gewijzigd – (3.40) is toegevoegd.]

Opmerking 3 bij de term: In deze NPR worden de termen risicobeheersmaatregel en beheersmaatregel beide gebruikt in dezelfde betekenis.

3.43
Scrum
framework om op een flexibele manier (software)producten te maken waarbij wordt gewerkt in multidisciplinaire teams die aan de hand van een geprioriteerde product backlog in korte sprints, met een vaste lengte van een tot vier weken, werkende (software)producten opleveren

Opmerking 1 bij de term: Scrum wordt vaak gebruikt bij producten waarvan de klant resp. gebruiker nog niet goed weet wat hij/zij wil en waarbij men al doende leert om de eisen en wensen beter te beschrijven en in bruikbare producten om te zetten. Vaak weet de klant resp. gebruiker pas wat hij/zij wil als het eerste product, of het prototype, wordt opgeleverd en dan worden alsnog de eisen aangepast. Scrum heeft de flexibiliteit om met laat wijzigende eisen en wensen om te gaan.

Opmerking 2 bij de term: Scrum is een vorm van Agile-softwareontwikkeling.

3.44
sprint
tijdsperiode met een vaste lengte van maximaal een maand waarin een gereed, bruikbaar en potentieel uit te brengen productincrement wordt gemaakt

[BRON: Schwaber en Sutherland, 2017]

Opmerking 1 bij de term: Sprints zijn iteraties binnen Scrum die meestal een week tot een maand duren. Sprints zijn vaste tijdblokken ('time box'): van tevoren staat vast hoe lang een sprint maximaal duurt en wanneer deze is afgelopen. Aan het begin van een sprint worden de user stories voor die sprint bepaald en vastgelegd in de 'sprint backlog'.

Opmerking 2 bij de term: Sprints resulteren in zo tastbaar mogelijke resultaten. Bij softwareontwikkeling heeft dat betrekking op bruikbare code, inclusief integratie, tests en documentatie, en liefst toepasbaar voor de klant of eindgebruiker. Aan het eind van een sprint vindt een sprint review plaats, waarbij het resultaat wordt getoond aan de belanghebbenden. Daarnaast vindt een evaluatie binnen het team plaats.

3.45
story map
tweedimensionale weergave van een (deel van de) backlog dat een visueel hulpmiddel is om de user stories op de backlog logisch op te knippen, te groeperen en te ordenen waarmee een overzicht ontstaat van de relaties tussen alle user stories

3.46
technische schuld
expliciete of impliciete keuze die op korte termijn het ontwikkelen en/of onderhouden van maatwerksoftware versnelt of vereenvoudigt maar op langere termijn een technische context creëert waarin het ontwikkelen en/of onderhoud van die software meer tijd zal kosten

Opmerking 1 bij de term: Het lukt opdrachtnemers vaak niet om opdrachtgevers goed uit te leggen waarom het bestrijden en voorkomen van technische schuld een belangrijk aandachtspunt is bij de ontwikkeling van software op maat, wat het voorkomen en aanpakken van technische schuld weer kan bemoeilijken.

Opmerking 2 bij de term: Voorbeelden van technische schuld zijn:

— kopiëren van bestaande functionaliteit om snel een nieuwe functie toe te voegen zonder de daarbij ontstane duplicatie van broncode op te lossen;

— niet opnemen van bepaalde testgevallen in een geautomatiseerde regressietest;

— niet upgraden van gebruikte bibliotheken of raamwerken naar een recentere versie;

— niet opzetten en uitvoeren van een ketenintegratietest met te koppelen systemen;

— niet bijwerken van documentatie;

— niet expliciet maken van gebruikte domeinconcepten in de broncode (bijvoorbeeld het doorgeven van ‘strings’ met straat en postcode in plaats van het maken van een klasse voor het concept ’adres’).

3.47
test-driven development
TDD
ontwikkelmethode waarbij de ontwikkelaars unit-testen schrijven voorafgaand aan de implementatie van de betreffende functionaliteit

3.48
T-shirt sizing
manier om relatieve inschattingen te maken door het vergelijken van user stories (3.51) en deze te verdelen in de categorieën extra-small, small, medium, large en extra large

Opmerking 1 bij de term: Met T-shirt sizing worden de onderlinge verhoudingen duidelijk zonder dat er tijd wordt verspild aan valse precisie.

3.49
unit-test
test geschreven door een ontwikkelaar om een specifiek gedeelte van de broncode te testen en waarbij wordt gekeken of een specifieke input altijd de verwachte output genereert

3.50
use case
beschrijving van een gedrag van een systeem dat reageert op een verzoek dat stamt van buiten het systeem: de use case beschrijft wie met het betreffende systeem wat kan doen

3.51
user story
korte beschrijving (story) van wat een opdrachtgever/gebruiker (user) beoogt te bereiken met een deel van de totaaloplossing

Opmerking 1 bij de term: Een user story bestaat gewoonlijk uit enkele zinnen gewone spreektaal van de opdrachtgever/gebruiker waarin staat wat de gebruiker doet of moet kunnen doen om een bepaalde doelstelling te behalen. Daarnaast beschrijft een user story welke acceptatiecriteria worden gehanteerd om de gerealiseerde functionaliteit als voltooid te beschouwen.

Opmerking 2 bij de term: Een veelgebruikt formaat voor user stories is: “Als <rol> kan ik <activiteit> zodat <doel>”. Bijvoorbeeld: “Als werknemer kan ik mijn onkosten specificeren zodat ik die vergoed krijg door mijn werkgever”.

3.52
user story-punt
relatieve maat voor de omvang van user stories, veel gebruikt in Agile-ontwikkelmethodes

4 Afkortingen

Hieronder staan de afkortingen die in dit document voorkomen zonder de volledige term. Afkortingen die alleen voorkomen in combinatie met de volledige term zijn niet opgenomen in deze lijst.

ISO	International Organization for Standardization
NPR	Nederlandse praktijkrichtlijn
SEI	Software Engineering Institute

5 Toelichting begrippen

5.1 Het begrip risico

Risico is het effect van onzekerheid op het behalen van doelstellingen. Een effect is een afwijking ten opzichte van de verwachting – positief en/of negatief. Een risico wordt vaak gekarakteriseerd door verwijzing naar mogelijke gebeurtenissen en gevolgen, of een combinatie daarvan.

In de context van deze NPR wordt het begrip risico als volgt geïnterpreteerd:

Onzekerheid wordt veroorzaakt door de kans dat er gebeurtenissen optreden. Er is alleen risico als er kans bestaat op het optreden van gebeurtenissen die effect hebben op het behalen van doelstellingen. Als de kans op het optreden van dergelijke gebeurtenissen nul is, is er geen sprake van een risico. Bijvoorbeeld, bij software die geen tot personen herleidbare gegevens bevat, is er ook geen kans dat dergelijke gegevens worden ontvreemd. Als de kans op het optreden van de desbetreffende gebeurtenis één is (ofwel 100 %), is er ook geen sprake van een risico. De gebeurtenis zal dan immers met zekerheid plaatsvinden.

Een voorbeeld van een gebeurtenis met effect op het behalen van een doelstelling is 'de te koppelen systemen zijn niet tijdig aangepast'. De kans op deze gebeurtenis kan op verschillende manieren worden bepaald. Eén manier is extrapolatie van historische gegevens. Stel dat van de 30 projecten die een organisatie heeft uitgevoerd, er 13 te maken hebben gekregen met te koppelen systemen die niet op tijd bleken te zijn aangepast. Een (naïeve) kansberekening levert dan een kans van $13/30 \cdot 100\ \% = 43\ \%$ op dat een project te maken krijgt met niet-tijdig aangepaste te koppelen systemen. Uiteraard zal een risico-inschatting die rekening houdt met het aantal te koppelen systemen nauwkeuriger zijn.

Naast de kans op het optreden van een gebeurtenis is minstens zo belangrijk te weten wat het effect is op het behalen van doelstellingen. Een gebeurtenis die geen effect heeft op het behalen van doelstellingen, vormt geen risico. Als het kabinet valt, maar dit geen effect heeft op het realiseren van de projectdoelstelling omdat het geen overheidsproject is, vormt de gebeurtenis geen risico. Een voorbeeld van een gebeurtenis die wel effect heeft op het behalen van doelstellingen, is het ongeoorloofd lezen van gegevens uit het ontwikkelde systeem. Een dergelijke gebeurtenis kan meerdere effecten hebben: de privacy van gebruikers is geschonden, maar ook de reputatie van de betrokken organisaties kan worden beschadigd en boetes kunnen worden opgelegd.

De effecten hangen samen met verschillende doelstellingen; in dit geval de privacy van gebruikers waarborgen en het hebben van een betrouwbare reputatie.

Vaak wordt alleen gesproken van een risico als er een gebeurtenis optreedt waarvan vaststaat dat deze een negatief gevolg heeft op het behalen van een doelstelling. In het geval van een gebeurtenis met een positief effect spreekt men over het algemeen van een kans. In complexe (samenwerkingsverbanden van) organisaties kan een ogenschijnlijk positief effect ook als een risico worden beschouwd. Denk hierbij aan het te vroeg opleveren van functionaliteit, waardoor de organisatie die dit product moet gaan gebruiken er nog niet klaar voor is.

5.2 Het begrip beheersmaatregel

Een beheersmaatregel is een maatregel waarmee een risico wordt gewijzigd.

In de context van deze NPR wordt het begrip beheersmaatregel als volgt geïnterpreteerd: Een beheersmaatregel wijzigt de kans dat een risico optreedt of wijzigt het effect van het optreden van een risico, of beide.

Een voorbeeld van een beheersmaatregel voor de gebeurtenis “er vindt een ‘distributed denial of service’ (DDoS)-aanval plaats op het systeem” is om in de IT-infrastructuur een ‘hardware appliance’ te plaatsen die het netwerkverkeer filtert en zo het effect van de aanval vermindert.

Een voorbeeld van een beheersmaatregel die de kans vermindert op een DDoS-aanval, is om het systeem niet via internet te ontsluiten, maar alleen via een intranet of virtual private netwerk (VPN).

Merk op dat een beheersmaatregel ook kan bestaan uit het nalaten van handelingen. Neem bijvoorbeeld het risico dat de efficiëntie van een ontwikkelteam negatief wordt beïnvloed doordat de opdrachtgever de werkzaamheden van het ontwikkelteam vaak onderbreekt met nieuwe, direct op te pakken wensen of te repareren fouten. Door af te spreken dat het ontwikkelteam de tijd verdeelt in tijdblokken (in sommige softwareontwikkelaanpakken sprints genoemd) en tevens af te spreken dat de opdrachtgever de lijst van taken waaraan het team werkt niet tijdens het lopende tijdblok mag aanpassen, kan dit risico worden verminderd.

Beheersmaatregelen kunnen risico’s op verschillende manieren wijzigen. Dit wordt ook wel risicobehandeling genoemd. Voorbeelden van risicobehandeling in het domein van maatwerksoftware zijn:

— Vermijden: door de activiteit die het risico met zich meebrengt niet uit te voeren wordt het risico vermeden. Door bijvoorbeeld gegevens niet via een koppelvlak op te halen uit een ander systeem maar alle gegevens in een eigen database op te slaan, wordt het risico vermeden dat het andere systeem niet bereikbaar is.

— Wegnemen: doordat de beheersmaatregel een risicobron wegneemt kan het risico niet optreden. Door bijvoorbeeld een functie niet te bouwen waarvoor geen duidelijke acceptatiecriteria zijn geformuleerd, wordt het risico weggenomen dat de functie niet voldoet aan de (impliciete) eisen en wensen van de opdrachtgever.

— Kans wijzigen: de beheersmaatregel beïnvloedt de kans dat het risico optreedt. Door bijvoorbeeld geautomatiseerde regressietesten te ontwikkelen en te draaien, wordt de kans op het niet tijdig detecteren van fouten in de software kleiner.

— Gevolgen wijzigen: de beheersmaatregel beïnvloedt de gevolgen van het optreden van het risico. Door bijvoorbeeld een nieuwe versie van de software uit te brengen voor een deel van de gebruikers in plaats van voor alle gebruikers tegelijk, worden de gevolgen van eventuele fouten in de software beperkt tot dat deel van de gebruikers.

— Delen: de beheersmaatregel zorgt ervoor dat het risico wordt gedeeld met een of meer andere partijen. Door bijvoorbeeld software samen met andere organisaties in een joint-venture te ontwikkelen, vermindert het risico voor elk van de deelnemende organisaties.

— Accepteren: de beheersmaatregel bestaat uit het onderbouwd behouden van het risico (accepteren). Het risico op performanceproblemen kan bijvoorbeeld worden geaccepteerd als het aantal beoogde gebruikers en de te verwerken hoeveelheid gegevens klein zijn.

— Overdragen: de beheersmaatregel bestaat uit het overdragen van het risico naar een andere partij. Het risico dat de ontwikkelomgeving niet beschikbaar is, kan bijvoorbeeld (deels) worden overgedragen naar een andere partij door de ontwikkelomgeving als een dienst af te nemen van de desbetreffende partij.

6 Risico's

6.1 Algemeen

Dit hoofdstuk beschrijft veelvoorkomende risico's die kunnen optreden bij de ontwikkeling en het onderhoud van maatwerksoftware. De in deze NPR beschreven risico's zijn gebaseerd op de risicotaxonomie uit *Taxonomy-based risk identification* (No. CMU/SEI-93-TR-06), de inbreng van ervaringen met de ontwikkeling en het onderhoud van maatwerksoftware van de 'NPR 5325-schrijfgroep' binnen de normcommissie, en input uit het veld in de vorm van reviewcommentaar op de ontwerpversie van deze NPR.

6.2 Productgerelateerde risico's

6.2.1 Algemeen

Productgerelateerde risico's komen voort uit de werkzaamheden die nodig zijn om de maatwerksoftware te ontwikkelen en/of te onderhouden.

6.2.2 Risico 01: De software wordt gewijzigd waardoor de kwaliteit van de software verslechtert

Het maken van wijzigingen aan software brengt het risico met zich mee op nieuwe fouten of op het opnieuw optreden van eerder gerepareerde fouten. Fouten kunnen ontstaan door verkeerde aannamen of vergissingen bij het aanpassen van de software, maar ook doordat een nieuwere versie van een softwarebibliotheek wordt gebruikt.

Fouten kunnen zich manifesteren in voor gebruikers direct merkbare verslechteringen, zoals niet-werkende functies, lagere prestatie-efficiëntie (performance) of slechtere bruikbaarheid, maar ook in voor gebruikers niet direct merkbare verslechteringen, zoals slechtere beveiliging of onderhoudbaarheid. Zie NEN-ISO/IEC 25010 voor een overzicht van verschillende kwaliteitsaspecten van software.

Het optreden van regressies leidt meestal tot ontevreden opdrachtgevers en gebruikers, en mogelijk tot allerlei vormen van schade (gezondheidsschade, financiële schade, reputatieschade, enz.). Maar het optreden van regressies heeft ook een potentieel secundair effect: het kan ertoe leiden dat partijen betrokken bij de ontwikkeling en/of het onderhoud van de software terughoudend worden om de software vaak uit te brengen en nieuwe versies in gebruik te nemen. Dat maakt het moeilijker een aantal beheersmaatregelen die deze NPR voor andere risico's aanbeveelt te implementeren. Het belang van het mitigeren van de risico's van regressies is dus nog groter dan al op het eerste gezicht duidelijk is.

Dit risico kan worden verminderd door de volgende beheersmaatregelen toe te passen:

— Maatregel 09: Geautomatiseerde ontwikkelpijplijn inrichten;

— Maatregel 10: Voortdurend voldoen aan de eisen met regressietests;

— Maatregel 13: Toepassen van een kwaliteitgedreven ontwikkelmethode;

— Maatregel 15: Deugdelijke overdracht.

6.2.3 Risico 02: De omgeving verandert waardoor de kwaliteit van de software verslechtert

Ook zonder dat software wijzigt kan de kwaliteit verslechteren of de functionaliteit niet meer goed werken. Zonder dat de broncode van de software verandert ontstaan er fouten of komen er fouten aan het licht. Dit fenomeen wordt ook wel 'software rot' genoemd en is te verklaren uit veranderingen in de omgeving van de software.

Mogelijke veranderingen in de omgeving die de kwaliteit van de software beïnvloeden, zijn bijvoorbeeld:

— nieuw ontdekte beveiligingskwetsbaarheden in gebruikte bibliotheken die de software onveiliger maken;

— koppelvlakken van andere systemen die veranderen waardoor de software geen gegevens meer kan ophalen bij die systemen;

— een nieuwe versie van het besturingssysteem;

— nieuwere versies van browsers op de computers van eindgebruikers;

— veranderd gebruik van de software waardoor fouten aan het licht komen die eerder niet zichtbaar waren;

— netwerkinstellingen die veranderen.

De meeste maatregelen die helpen voor het verminderen van de risico's bij het wijzigen van software, helpen ook hier, met dien verstande dat ze periodiek worden toegepast en niet alleen bij wijziging van de software.

Dit risico kan worden verminderd door de volgende beheersmaatregelen toe te passen:

— Maatregel 09: Geautomatiseerde ontwikkelpijplijn inrichten;

— Maatregel 10: Voortdurend voldoen aan de eisen met regressietests;

— Maatregel 13: Toepassen van een kwaliteitgedreven ontwikkelmethode.

6.3 Projectgerelateerde risico's

6.3.1 Algemeen

Projectgerelateerde risico's komen voort uit de omgeving waarin de ontwikkeling en het onderhoud van maatwerksoftware plaatsvindt.

6.3.2 Risico 03: De hoeveelheid werk is niet correct ingeschat waardoor de geplande functionaliteit niet op tijd af is

Bij veel maatwerksoftwareontwikkeling en -onderhoud heeft de opdrachtgever behoefte aan het gereed zijn van een bepaalde functionaliteit op een specifiek tijdstip, bijvoorbeeld omdat wetgeving of beleid verandert met ingang van een specifieke datum. Het inschatten van de hoeveelheid werk die het kost een bepaalde hoeveelheid functionaliteit te realiseren, is echter notoir lastig. Bij maatwerksoftware gaat het immers veelal (herbouw van een bestaand systeem vormt een uitzondering) om het maken van iets dat nog niet bestaat. Er is dus een risico dat de hoeveelheid werk te laag wordt ingeschat, met als gevolg dat de functionaliteit niet op tijd is gerealiseerd. Dit kan weer allerlei gevolgen met zich meebrengen, zoals teamleden die na de eerder geplande einddatum niet meer beschikbaar zijn, meerwerkkosten bij leveranciers of de perceptie van een mislukt project, met stopzetten als ultiem gevolg. Het niet op tijd realiseren van de functionaliteit kan zo leiden tot publiciteitsschade, politieke schade of gemiste omzet.

Hoe later wordt ontdekt dat de benodigde functionaliteit niet gereed is op het afgesproken tijdstip, des te lastiger het is voor een opdrachtgever om alternatieve oplossingen te realiseren.

Overigens is het te hoog inschatten van de hoeveelheid werk ook een risico. Dit kan leiden tot onnodig inzetten van resources en verspilling van tijd en middelen.

Dit risico kan worden verminderd door de volgende beheersmaatregelen toe te passen:

— Maatregel 01: Belanghebbenden identificeren en betrekken;

— Maatregel 04: Productdecompositie in incrementeel opleverbare delen met business-waarde;

— Maatregel 05: Technische schuld identificeren, inzichtelijk maken en planmatig oplossen;

— Maatregel 06: Oplossingsrichtingen verkennen;

— Maatregel 07: Incrementele oplevering van het product;

— Maatregel 08: Iteratieve ontwikkelaanpak;

— Maatregel 11: Voortgangsbewaking met burndown charts.

6.3.3 Risico 04: Er vinden scope-uitbreidingen plaats waardoor het product niet tijdig en binnen budget wordt opgeleverd

Gedurende de ontwikkeling van maatwerksoftware kunnen scope-uitbreidingen plaatsvinden. Hiervoor zijn verschillende redenen mogelijk. Externe gebeurtenissen hebben invloed op de gewenste functionaliteit en vereisen aanpassingen. Ook door incrementele oplevering en ingebruikname van de software kunnen betrokkenen tot nieuwe inzichten komen. Als er niet wordt gekozen om eerder bedachte functionaliteit te laten vervallen ten gunste van de extra bedachte functionaliteit, dan zal dit vaak leiden tot het niet tijdig en binnen budget opleveren van de software. Het niet laten vervallen van functionaliteit kan ook leiden tot hogere werkdruk binnen het ontwikkelteam, waardoor de kwaliteit onder druk komt te staan.

Dit risico kan worden verminderd door de volgende beheersmaatregelen toe te passen:

— Maatregel 04: Productdecompositie in incrementeel opleverbare delen met business-waarde;

— Maatregel 07: Incrementele oplevering van het product;

— Maatregel 12: Een officiële producteigenaar met mandaat.

6.3.4 Risico 05: Het team beschikt niet over de juiste expertise waardoor de software niet voldoet aan de gestelde eisen

Voor het ontwikkelen van software is vaak veel verschillende expertise nodig. Denk aan kennis van het toepassingsdomein, kennis van de gebruikte technologie, kennis van gerelateerde systemen en specifieke vakinhoudelijke kennis, zoals kennis van beveiliging, performance en gebruikskwaliteit.

Als de benodigde expertise niet tijdig aanwezig is binnen het team of als mensen met de benodigde expertise tussentijds het team verlaten, ontstaat het risico dat de opgeleverde software niet voldoet aan de gestelde kwaliteitseisen. Daarnaast bouwt een team gedurende de looptijd ook weer specifieke kennis op over de ontwikkelde maatwerksoftware. Hierdoor kan het lastig zijn kennis en expertise te borgen als er verloop plaatsvindt.

Als het team niet beschikt over de juiste technische kennis, bestaat het risico dat er een verkeerde aanpak of een verkeerde oplossing wordt gekozen en geen gebruik wordt gemaakt van geëigende patronen en raamwerken. Dit kan tot gevolg hebben dat de software duurder is om te ontwikkelen en onderhouden dan nodig.

Als het team niet beschikt over de benodigde domeinexpertise, ontstaat het risico dat er verkeerde aannames worden gemaakt over de gewenste ondersteuning van domeinspecifieke processen. Dit heeft als mogelijk gevolg dat de software niet of niet goed voldoet aan de eisen en wensen.

Dit risico kan worden verminderd door de volgende beheersmaatregelen toe te passen:

— Maatregel 01: Belanghebbenden identificeren en betrekken;

— Maatregel 02: Belangrijke niet-functionele eisen identificeren;

— Maatregel 03: Belangrijke functionele eisen identificeren;

— Maatregel 13: Toepassen van een kwaliteitgedreven ontwikkelmethode;

— Maatregel 15: Deugdelijke overdracht;

— Maatregel 16: Teams met specialistische kennis en hulpmiddelen ondersteunen.

6.3.5 Risico 06: Gebrekkige aansturing van het werk waardoor het product niet de juiste functionaliteit biedt

Bij de ontwikkeling en het onderhoud van maatwerksoftware zijn vaak verschillende partijen en mensen betrokken met verschillende wensen en eisen. Door het ontbreken van een officiële producteigenaar, onduidelijke overlegstructuren of het niet duidelijk hebben van de business case wordt een product gebouwd dat uiteindelijk niet de functionaliteit biedt die nodig is voor het realiseren van de doelstelling.

Een factor die kan bijdragen aan dit risico is onvoldoende kennis bij management van de gekozen ontwikkelmethodiek waardoor er op oneigenlijke momenten wordt ingegrepen in het proces. Denk aan een manager die tijdens een sprint ingrijpt in de werkzaamheden van een team of een teamlid andere taken geeft.

Als meerdere ontwikkelteams moeten samenwerken om het product te realiseren is dit risico groter. Ontwikkelteams die tot verschillende organisaties behoren, vergroten dit risico nog eens extra.

Dit risico kan worden verminderd door de volgende beheersmaatregelen toe te passen:

— Maatregel 01: Belanghebbenden identificeren en betrekken;

— Maatregel 12: Een officiële producteigenaar met mandaat.

6.3.6 Risico 07: Functionele eisen krijgen te veel prioriteit waardoor het product de juiste niet-functionele eigenschappen ontbeert

Bij de ontwikkeling van maatwerksoftware ligt de focus van (direct) betrokkenen op de functionaliteit van het te ontwikkelen softwareproduct. Hoewel dat op zichzelf logisch is, de functionaliteit is immers het primaire doel van het product, kan het gevolg zijn dat er te weinig prioriteit en dus tijd en aandacht wordt besteed aan andere producteigenschappen, zoals gebruikskwaliteit, performance, beveiliging, onderhoudbaarheid en beheerfunctionaliteit. Het niet, te weinig of te laat betrekken van rollen als eindgebruikers, functioneel beheerders en technisch beheerders is vaak debet aan dit risico.

De schade die kan ontstaan als dit risico optreedt, kan zich uiten in een noodzakelijke uitbreiding van de scope van het werk, met gevolgen voor doorlooptijd en/of kosten, onnodige kosten voor het uitvoeren van rework op reeds gerealiseerde software zodat die alsnog aan de eisen voldoet, of onvoorziene kosten na implementatie, zoals opschaling van beheerders of uitgebreide training van eindgebruikers.

De schade kan ook optreden in de vorm van technische schuld die zich op termijn manifesteert in dalende onderhoudbaarheid. Hierdoor kan op den duur zelfs een vicieuze cirkel ontstaan van meer technische schuld, dalende onderhoudbaarheid, meer nadruk op functionaliteit (want het kost zoveel tijd om de verandering te maken) en vervolgens nog meer technische schuld en een lagere onderhoudbaarheid.

Dit risico kan worden verminderd door de volgende beheersmaatregelen toe te passen:

— Maatregel 01: Belanghebbenden identificeren en betrekken;

— Maatregel 02: Belangrijke niet-functionele eisen identificeren;

— Maatregel 05: Technische schuld identificeren, inzichtelijk maken en planmatig oplossen;

— Maatregel 07: Incrementele oplevering van het product;

— Maatregel 08: Iteratieve ontwikkelaanpak;

— Maatregel 10: Voortdurend voldoen aan de eisen met regressietests.

6.3.7 Risico 08: De communicatie tussen belanghebbenden is suboptimaal waardoor misverstanden ontstaan

Omdat bij maatwerksoftwareontwikkeling veelal meerdere belanghebbenden moeten communiceren over de productvisie, eisen en wensen, vormt suboptimale communicatie een belangrijk risico. Een andere versterkende factor is fysieke afstand tussen het ontwikkelteam en de gebruikers. Als zij elkaar weinig of nooit in levenden lijve spreken, dan verhoogt dit ook de kans op misverstanden. Bij near- of offshoring van maatwerksoftwareontwikkeling kunnen culturele verschillen en onbekendheid met inhoudelijke materie dit probleem versterken.

De consequenties van dit risico bestaan vooral uit herstelwerk nodig omdat correct functionerende software toch niet doet of kan doen wat de belanghebbenden voor ogen hadden. Dit herstelwerk kan leiden tot oplopende kosten, zeker wanneer contractuele kosten zijn verbonden aan

wijzigingsvoorstellen en latere oplevering. In het uiterste geval kan een project zelfs compleet mislukken doordat de ontwikkelde software te ver van de visie van de belanghebbenden afstaat of omdat deze het vertrouwen in het product verliezen.

Traditioneel werd geprobeerd dit probleem op te lossen door visie, eisen en wensen zo goed mogelijk vooraf vast te leggen in documentatie, soms zelfs formeel gespecificeerd. In Agile-ontwikkelmethodes wordt meer nadruk gelegd op directe communicatie tussen de belanghebbenden op specifieke momenten tijdens de ontwikkeling, en het blijven verfijnen en bijstellen van de eisen en wensen.

Dit risico kan worden verminderd door de volgende beheersmaatregelen toe te passen:

— Maatregel 01: Belanghebbenden identificeren en betrekken;

— Maatregel 08: Iteratieve ontwikkelaanpak;

— Maatregel 12: Een officiële producteigenaar met mandaat.

6.3.8 Risico 09: Onvoldoende traceerbaarheid van ontwikkeling, gebruik en beheer van maatwerksoftware leidt tot het niet (aantoonbaar) voldoen aan verplichtingen

Bij de ontwikkeling en het onderhoud van maatwerksoftware is vaak sprake van specifieke verplichtingen met betrekking tot het gebruik en beheer van het systeem, zoals eisen van externe belanghebbenden, acceptatiecriteria van de beheerorganisatie, criteria voor het mogen aansluiten op een leverend systeem, contractuele verplichtingen en wettelijke verplichtingen, zoals toegankelijkheid en het voldoen aan de archiefwet. Als deze verplichtingen ter discussie staan, zoals bij een audit, formele uitvraag of conflict, moet aantoonbaar kunnen worden gemaakt dat het systeem hieraan voldoet. Dit wordt ook traceerbaarheid genoemd en kan bijvoorbeeld worden geborgd door audit trails van gebruiksacties, logging van systeemacties, documentatie en archivering van de staat van het systeem en wijzigingen erop, en het traceerbaar beheren van eisen en wensen.

Wanneer tijdens de ontwikkeling van maatwerksoftware hiervoor onvoldoende aandacht is geweest, ontstaat het risico dat de software of de beheerorganisatie niet is toegerust op het borgen van deze verplichtingen. Dat kan ertoe leiden dat niet bewijsbaar is wat de staat van het systeem is, of in het verleden is geweest, welke acties gebruikers wel of niet hebben gedaan met het systeem, welke informatie het systeem heeft geleverd aan andere systemen of aan welke eisen het systeem voldoet en op welke wijze deze eisen zijn gerealiseerd.

Een factor die de kans dat dit risico optreedt kan verhogen, is overdracht van taken aan andere partijen. Dit kan voorkomen wanneer ontwikkeling en/of onderhoud worden overgedragen aan een (andere) externe dienstverlener of wanneer data of functionaliteit in een maatwerksoftwaresysteem wordt gemigreerd naar een ander systeem. Ook de betrokkenheid van veel verschillende partijen bij de ontwikkeling en/of onderhoud van de software kan dit risico verhogen doordat er onduidelijkheid of misverstanden ontstaan over wie er verantwoordelijk is voor de realisatie van welke verplichtingen.

Dit risico kan worden verminderd door de volgende beheersmaatregelen toe te passen:

— Maatregel 12: Een officiële producteigenaar met mandaat;

— Maatregel 14: Archivering;

— Maatregel 15: Deugdelijke overdracht;

— Maatregel 16: Teams met specialistische kennis en hulpmiddelen ondersteunen.

6.3.9 Risico 10: Doordat veel tijd nodig is voor het invullen van randvoorwaarden voor de softwareontwikkeling, wordt het product niet tijdig opgeleverd

Net als bij vrijwel alle bedrijfsmatig uitgevoerde activiteiten moeten voor het ontwikkelen en onderhouden van maatwerksoftware bepaalde randvoorwaarden worden ingevuld. Voorbeelden hiervan zijn verwerving van de juiste resources, beschikbaarheid van de juiste hulpmiddelen en de inrichting van werkplekken. De ontwikkeling en onderhoud van maatwerksoftware stelt echter vrij specifieke eisen aan de invulling van deze randvoorwaarden. Als dat onvoldoende wordt ondersteund in de organisatie, kan dit negatieve gevolgen hebben voor het team.

Zo kan het uitdagend blijken om resources, hulpmiddelen of werkplekken te verstrekken via gestandaardiseerde processen, omdat de uitvoerders van deze processen vaak niet de specialistische kennis hebben van ontwikkel- of onderhoudsprocessen. Het gevolg kan zijn dat op deze wijze onvoldoende invulling wordt gegeven aan de specifieke behoeften van het team. Wanneer het team de eigen processen aan moet passen, de standaard processen vermijdt of deze processen zelf gaat uitvoeren, kan dat ten koste gaan van doorlooptijd en budget.

Contractuele afspraken met externe dienstverleners voor uitvoering van gestandaardiseerde processen kunnen van invloed zijn op de kans dat dit risico optreedt en op de ernst van de gevolgen. Contracten waarin procesafspraken te rigide zijn vastgelegd, kunnen een belemmering vormen voor de invulling van specifieke of veranderende behoeften. Ook kan het ontbreken van afspraken over uitzonderingen in de procesgang de mogelijkheden om de gevolgen op te vangen in het team beperken.

Dit risico kan worden verminderd door de volgende beheersmaatregel toe te passen:

— Maatregel 16: Teams met specialistische kennis en hulpmiddelen ondersteunen.

7 Beheersmaatregelen

7.1 Algemeen

Een organisatie die besluit zelf maatwerksoftware te ontwikkelen en/of te onderhouden, zal dat vanwege de kosten en benodigde expertise en hulpmiddelen vaak niet beperken tot één applicatie. Deze NPR veronderstelt dan ook dat de organisatie meerdere maatwerkapplicaties parallel ontwikkelt en/of onderhoudt en dat er binnen de organisatie meerdere organisatieonderdelen (teams, afdelingen en/of projecten) zijn die elk een of meerdere maatwerkapplicaties ontwikkelen en/of onderhouden. Deze organisatieonderdelen worden in deze NPR afhankelijk van de context aangeduid met de term project, projectteam of team.

OPMERKING De onderverdeling van beheersmaatregelen naar organisatiegerichte beheersmaatregelen en projectgerichte beheersmaatregelen is een vereenvoudigde versie van de onderverdeling van processen in NEN-ISO/IEC/IEEE 12207. ‘Organizational Project-Enabling processes’ heten in deze NPR organisatiegerichte beheersmaatregelen. ‘Agreement processes’, ‘Technical Management processes’ en ‘Technical processes’ heten in deze NPR projectgerichte beheersmaatregelen.

Sommige beheersmaatregelen zullen primair door de opdrachtgever van maatwerksoftware worden uitgevoerd, sommige primair door de opdrachtnemer en veel ook door opdrachtgever en opdrachtnemer samen. Voorbeelden van beheersmaatregelen die primair door de opdrachtgever worden uitgevoerd, zijn het identificeren van belanghebbenden (Maatregel 01) en het identificeren van de belangrijke niet-functionele en functionele eisen (Maatregel 02 en Maatregel 03). Voorbeelden van beheersmaatregelen die waarschijnlijk primair door de opdrachtnemer zullen worden toegepast, zijn het inrichten van een geautomatiseerde bouwpijplijn (Maatregel 09) en het inzichtelijk maken van technische schuld (Maatregel 05).

7.2 Projectgerichte beheersmaatregelen

7.2.1 Algemeen

Bij projectgerichte beheersmaatregelen wordt onderscheid gemaakt tussen maatregelen die bij de voorbereiding van de ontwikkel- en/of onderhoudswerkzaamheden worden uitgevoerd (de voorbereidingsfase), maatregelen die tijdens de looptijd worden uitgevoerd (de realisatiefase) en maatregelen die bij afronding van de ontwikkeling en/of het onderhoud worden uitgevoerd (de afrondingsfase).

Ook als het werk niet als project maar in de vorm van een doorlopende dienst wordt georganiseerd, zoals vaak het geval is bij onderhoud, zijn deze maatregelen van toepassing. Ook als een ontwikkelmethodiek wordt gebruikt waar deze fasen in elkaar overlopen of deels gelijktijdig plaatsvinden, zijn deze maatregelen van toepassing.

Tijdens de voorbereidingsfase werken vertegenwoordigers van de opdrachtgever, opdrachtnemer en beoogde beheerpartij nauw samen ten behoeve van een effectieve realisatiefase. Zij zijn bij voorkeur ook degenen die betrokken zullen zijn bij de uitvoering en realisatie van (een deel van) de op te leveren producten. Na de projectvoorbereiding beschikken de directbetrokkenen over voldoende kennis om de realisatiefase succesvol te kunnen uitvoeren. Tijdens de realisatiefase vindt de bouw en het onderhoud van de software plaats (zie 7.2.3). Tijdens de afrondingsfase vindt archivering en eventueel overdracht van het werk plaats (zie paragraaf 7.2.4).

Het doel van het splitsen in deze fasen is drieledig:

1) om de uitgangspunten, risico's en randvoorwaarden expliciet te maken voorafgaand aan de uitvoering van de werkzaamheden;

2) om te zorgen dat aan de randvoorwaarden wordt voldaan en voor zoveel mogelijk productspecifieke risico's maatregelen zijn genomen;

3) om te zorgen dat afsluiting en overdracht van de werkzaamheden voldoende aandacht krijgen.

7.2.2 Projectvoorbereiding

7.2.2.1 Algemeen

De projectvoorbereiding is de fase waarin de betrokken organisaties de uitvoering van de softwareontwikkeling en/of het onderhoud voorbereiden.

7.2.2.2 Maatregel 01: Belanghebbenden identificeren en betrekken

De identificatie van belanghebbenden (ook wel stakeholderanalyse genoemd) besteedt minimaal aandacht aan de belanghebbenden, hun belangen en de wijze waarop de ze worden betrokken. Dit wordt typisch vastgelegd in een projectplan of 'service delivery'-plan.

De volgende partijen zijn veelal belanghebbend bij de ontwikkeling en/of het onderhoud van maatwerksoftware:

— eindgebruikers;

— materiedeskundigen;

— beheerders van de software. Dit kunnen functioneel beheerders, applicatiebeheerders en technisch beheerders zijn;

— ontwikkelaars, met name als er specifieke expertise nodig is waar slechts een beperkt aantal ontwerpers, programmeurs of testers over beschikt;

— opdrachtgever(s);

— producteigenaar (product owner): de huidige eigenaar van het te onderhouden systeem of de beoogd eigenaar van een nieuw te ontwikkelen systeem. In Agile-methodes, zoals Scrum en Extreme Programming, is de producteigenaar de primair verantwoordelijke voor het maximaliseren van de waarde van het te ontwikkelen of beheren systeem. Dit is een rol en hoeft niet noodzakelijk ook de bedrijfsmatige producteigenaar te zijn;

— beheerders van te koppelen systemen;

— beheerders van de te gebruiken technische infrastructuur.

Een werkwijze om de belangen van gebruikers en andere belanghebbenden tastbaar te maken voor het ontwikkelteam is het gebruik van ‘persona’. Persona vormen een karakterisering van bepaalde types gebruiker, denk aan ‘sluiswachter Johanna van der Riet’ of ‘Jan Bakker, vader van twee jonge kinderen op de lagere school, te Edam’.

Naast het identificeren van belanghebbenden is het belangrijk om ze vroegtijdig te betrekken en op de hoogte te houden. Belanghebbenden kunnen bijvoorbeeld op de volgende wijzen worden betrokken bij de ontwikkeling en het onderhoud van de maatwerksoftware:

— Betrek de belanghebbenden bij het opstellen van eisen en acceptatiecriteria.

— Organiseer een startbijeenkomst. Bij deze bijeenkomst ontmoeten alle belanghebbenden en het ontwikkelteam elkaar. Hierdoor wordt iedereen op een gelijk kennisniveau gebracht en worden er contacten gelegd om later informatie uit te wisselen.

— Organiseer tijdens de looptijd van het werk regelmatig demo’s waarin het team de software demonstreert aan de belanghebbenden.

— Geef belanghebbenden toegang tot een demo-omgeving om de software uit te proberen en te testen.

— Breng gebruikers, beheerders en ontwikkelaars dichter bij elkaar zodat er een beter begrip ontstaat voor elkaars wensen en problemen. Een DevOps-aanpak kan daarbij helpen, waarbij een zogenoemd DevOps-team verantwoordelijk is voor zowel het ontwikkelen van de maatwerksoftware als het installeren en beheren van de software in de productieomgeving. Omdat het team verantwoordelijk is voor zowel wijzigen van de software als het beschikbaar houden ervan, voorkomt dit de traditionele spanning tussen ontwikkelteams (willen snel wijzigen) en beheerteams (willen stabiliteit).

Door vroegtijdig de belanghebbenden en hun belangen te identificeren kunnen de volgende risico’s worden verminderd;

— Risico 03: De hoeveelheid werk is niet correct ingeschat waardoor de geplande functionaliteit niet op tijd af is;

— Risico 05: Het team beschikt niet over de juiste expertise waardoor de software niet voldoet aan de gestelde eisen;

— Risico 06: Gebrekkige aansturing van het werk waardoor het product niet de juiste functionaliteit biedt;

— Risico 07: Functionele eisen krijgen te veel prioriteit waardoor het product de juiste niet-functionele eigenschappen ontbeert;

— Risico 08: De communicatie tussen belanghebbenden is suboptimaal waardoor misverstanden ontstaan.

7.2.2.3 Maatregel 02: Belangrijke niet-functionele eisen identificeren

Identificeer de belangrijke niet-functionele eisen tot een dusdanig niveau dat het mogelijk is de uitvoeringsfase te begroten met voldoende zekerheid over de benodigde expertise, doorlooptijd en kosten.

De belangrijke niet-functionele eisen kunnen worden geïdentificeerd door:

1) alle belanghebbenden vroegtijdig te betrekken bij het opstellen en goedkeuren van eisen;

2) een business impact-analyse (BIA) op te stellen;

   OPMERKING 1 In een BIA legt de opdrachtgevende organisatie vast hoe belangrijk informatiebeveiliging is voor de eigen bedrijfsvoering/processen. Naast de gevoeligheid voor incidenten wordt ook vastgelegd de mate waarin een organisatie bereid is risico's te nemen (de 'risk appetite'). Alleen de opdrachtgevende organisatie zelf kan hierover een uitspraak doen.

3) een Data Protection Impact Assessment (DPIA) op te stellen;

   OPMERKING 2 Een DPIA geeft inzicht in de risico's die de verwerking oplevert voor de betrokkenen, en in de maatregelen die de verantwoordelijke organisatie behoort te nemen om de risico's af te dekken. In veel gevallen wordt dit verplicht vanuit de Algemene Verordening Gegevensbescherming (AVG), dan wel de General Data Protection Regulation (GDPR). Zicht op privacygevoelige gegevens en het (laten) treffen van adequate en afdoende beschermingsmaatregelen is een wettelijke plicht die een organisatie niet aan een andere partij kan overdragen.

4) overige wettelijke verplichtingen te identificeren;

   OPMERKING 3 Denk aan toegankelijkheid en de archiefwet.

5) overige kaders als referentiearchitecturen, protocollen en samenwerkingsafspraken te identificeren;

6) kwaliteitsaspecten, zoals bijvoorbeeld door NEN-ISO/IEC 25010 beschreven, te prioriteren en de belangrijke uit te werken.

De niet-functionele eisen kunnen worden vastgelegd in een programma van eisen of als onderdeel van een projectstartarchitectuur. Indien van toepassing behoren ze ook in user stories te worden vervat en op een product backlog te worden geplaatst.

Niet-functionele eisen leiden vaak tot functionele eisen. Denk aan beveiligingseisen die vereisen dat er bepaalde logging plaatsvindt. De niet-functionele eisen en daarvan afgeleide functionele eisen vormen samen met de andere functionele eisen input voor de productdecompositie (zie Maatregel 04).

Door de belangrijke niet-functionele eisen vroegtijdig te identificeren kunnen de volgende risico's worden verminderd:

— Risico 05: Het team beschikt niet over de juiste expertise waardoor de software niet voldoet aan de gestelde eisen;

— Risico 07: Functionele eisen krijgen te veel prioriteit waardoor het product de juiste niet-functionele eigenschappen ontbeert.

7.2.2.4 Maatregel 03: Belangrijke functionele eisen identificeren

Identificeer de belangrijke functionele eisen tot een dusdanig niveau dat het mogelijk is de uitvoeringsfase met voldoende zekerheid te begroten. Dit kan worden gedaan door alle belanghebbenden vroegtijdig te betrekken bij het opstellen en goedkeuren van de eisen.

Het is verstandig ook de bron van elke eis vast te leggen, zodat de herkomst traceerbaar is. Afhankelijk van de werkafspraken (mogelijk vastgelegd in een 'definition of ready') behoren de functionele eisen ook meetbaar en testbaar te worden gemaakt door het toevoegen van acceptatiecriteria.

OPMERKING Functionele eisen kunnen bijvoorbeeld in de vorm van 'use cases' of 'user stories' worden vastgelegd.

De functionele eisen vormen samen met de niet-functionele eisen input voor de productdecompositie (zie Maatregel 04).

Structureel toepassen van een minimum viable product (MVP) en ondersteunende technieken als story mapping, design thinking en het lean startup-principe maken duidelijk wat minimaal moet worden opgeleverd op een bepaalde datum. Daar kan door alle partijen op worden gestuurd. Het MVP wordt in overleg met de opdrachtgever vastgesteld.

Door de belangrijke functionele eisen vroegtijdig te identificeren kan het volgende risico worden verminderd:

— Risico 05: Het team beschikt niet over de juiste expertise waardoor de software niet voldoet aan de gestelde eisen.

7.2.2.5 Maatregel 04: Productdecompositie in incrementeel opleverbare delen met business-waarde

Maak een productdecompositie in los opleverbare delen waarbij elke oplevering waarde voor de opdrachtgever en/of gebruikers kan realiseren.

Het maken van een productdecompositie bestaat uit de volgende activiteiten:

1) Maak een product breakdown structure (PBS) van het te realiseren product met onderdelen die los kunnen worden opgeleverd en die business-waarde hebben.

2) Wijs de functionele en niet-functionele eisen toe aan de onderdelen.

3) Orden de onderdelen op volgorde van business-waarde, rekening houdend met technische afhankelijkheden (product backlog).

4) Maak objectieve of intersubjectieve omvangschattingen van de onderdelen.

   OPMERKING 1 Voor het bepalen van de functionele omvang bestaan methoden als functiepuntanalyse, T-shirt sizing en user story-punten. Voor niet-functionele eisen bestaan dergelijke methoden niet of staan ze nog in de kinderschoenen.

   OPMERKING 2 Een intersubjectieve omvangschatting is een omvangschatting die is gebaseerd op meerdere subjectieve schattingen. Door middel van een iteratieve aanpak worden meerdere subjectieve schattingen omgezet in één gezamenlijke schatting. Voorbeelden van een dergelijke aanpak zijn de Delphi-methode en het 'pokeren' van user story-punten als onderdeel van Scrum.

5) Definieer de kleinste set van onderdelen die in gebruik kan worden genomen (minimum viable product).

Een productdecompositie in incrementeel opleverbare delen maakt het mogelijk de volgende risico's te verminderen:

— Risico 03: De hoeveelheid werk is niet correct ingeschat waardoor de geplande functionaliteit niet op tijd af is;

— Risico 04: Er vinden scope-uitbreidingen plaats waardoor het product niet tijdig en binnen budget wordt opgeleverd.

7.2.2.6 Maatregel 05: Technische schuld identificeren, inzichtelijk maken en planmatig oplossen

De aanwezigheid van technische schuld heeft een nadelige invloed op de kwaliteit van de eindproducten. Het ontstaan van technische schuld gedurende de levensduur van de software is in de praktijk echter onvermijdelijk. Technische schuld kan zelfs een nuttig hulpmiddel zijn indien bewust toegepast: door bijvoorbeeld een stuk bestaande functionaliteit te kopiëren en aan te passen is het soms mogelijk snel een nieuwe functie te realiseren. De technische schuld in de vorm van duplicatie tussen de originele en de gekopieerde broncode kan dan later worden opgelost.

Als bestaande maatwerksoftware moet worden afgebouwd, onderhouden, herbouwd en/of hergebruikt, dan is het van belang de kwaliteit te bepalen van die software en om eventueel aanwezige technische schuld te identificeren. Ook als later blijkt dat nog niet eerder onderzochte softwareproducten moeten worden toegevoegd aan de scope van het werk, vindt eerst aanvullend onderzoek plaats naar deze softwareproducten.

In alle gevallen is het verstandig om te weten welke technische schuld er bestaat. Om te voorkomen dat technische schuld niet wordt opgelost en alleen maar toeneemt, is het zaak om het verminderen van technische schuld planmatig aan te pakken.

Volg hiervoor de volgende stappen.

1) Onderzoek de compleetheid en kwaliteit van de bestaande softwareproducten. Het onderzoek resulteert in een overzicht van de kwaliteit en eventueel aanwezige technische schuld.

2) Bepaal tijdens de ontwikkeling en het onderhoud van de maatwerksoftware voortdurend de kwaliteit van de broncode en documentatie:

   a) Meet de kwaliteit van broncode door deze door geautomatiseerde tools te laten vergelijken met best practices op het gebied van programmeerrichtlijnen, architectuur en beveiliging.

   b) Meet de kwaliteit van de broncode en documentatie door teamleden elkaars wijzigingen te laten reviewen.

3) Maak inzichtelijk welke versies van derdepartijcomponenten en raamwerken worden gebruikt in de software en in hoeverre deze versies achterlopen.

4) Houd een issue log bij van technische schuld-issues.

5) Plan het oplossen van technische schuld-issues. Neem tijd voor het oplossen van technische schuld mee in de schatting van aanpassingen.

6) Los technische schuld-issues op.

Deze beheersmaatregel helpt de volgende risico's te verminderen.

— Risico 03: De hoeveelheid werk is niet correct ingeschat waardoor de geplande functionaliteit niet op tijd af is;

— Risico 07: Functionele eisen krijgen te veel prioriteit waardoor het product de juiste niet-functionele eigenschappen ontbeert.

7.2.2.7 Maatregel 06: Oplossingsrichtingen verkennen

Als er vooraf onduidelijkheden of onzekerheden bestaan over het gewenste product, dan kunnen oplossingsrichtingen worden verkend om tot een betere inschatting van de hoeveelheid werk te komen. Door een beperkte versie van (een deel van) de oplossing te realiseren kan relatief snel feedback worden verkregen over de haalbaarheid van een oplossingsrichting.

Door een of meerdere prototypes te ontwikkelen kan vroegtijdig terugkoppeling van gebruikers worden verkregen op de gebruikersinterface en de werking van de software. Als het primaire doel van een prototype is om te bepalen of een bepaald ontwerp haalbaar is, dan wordt dit ook wel een 'proof of concept' genoemd. In alle gevallen bevat een prototype slechts een deel van de uiteindelijk te realiseren functionaliteit of helemaal geen functionaliteit in het geval een 'papieren prototype', waar de userinterface op papier (of in een presentatie) wordt gesimuleerd.

OPMERKING Het gebruik van prototypes brengt een specifiek risico met zich mee: prototypes worden ontwikkeld als 'throw away'-software, maar er kan toch druk ontstaan om de prototypesoftware op te nemen in de uiteindelijke software, zonder dat er tijd wordt genomen de kwaliteit op voldoende niveau te brengen.

Ook tijdens de ontwikkeling van software kan soms extra onderzoek nodig zijn om de beste oplossingsrichting voor een deel van het werk te bepalen. In Extreme Programming wordt dit een 'spike' genoemd.

Deze beheersmaatregel helpt het volgende risico te verminderen:

— Risico 03: De hoeveelheid werk is niet correct ingeschat waardoor de geplande functionaliteit niet op tijd af is.

7.2.3 Projectuitvoering

7.2.3.1 Maatregel 07: Incrementele oplevering van het product

Lever het product op in incrementen aan de hand van de gemaakte opdeling (zie Maatregel 04). Doordat de opdeling geordend is naar business-waarde voor de opdrachtgever wordt het risico dat er onvoldoende functionaliteit op het gewenste tijdstip beschikbaar is sterk verminderd. Een increment van de software kan in principe in gebruik worden genomen en wordt ten minste door de opdrachtgever geaccepteerd. Onderdeel van de oplevering kan bestaan uit een productdemonstratie aan de opdrachtgever en gebruikers. Hierbij wordt opgemerkt dat de software pas echt af is als deze in gebruik is genomen en dat er tot dat moment nog allerlei problemen in verscholen kunnen zitten.

Deze beheersmaatregel helpt de volgende risico's te verminderen:

— Risico 03: De hoeveelheid werk is niet correct ingeschat waardoor de geplande functionaliteit niet op tijd af is;

— Risico 04: Er vinden scope-uitbreidingen plaats waardoor het product niet tijdig en binnen budget wordt opgeleverd;

— Risico 07: Functionele eisen krijgen te veel prioriteit waardoor het product de juiste niet-functionele eigenschappen ontbeert.

7.2.3.2 Maatregel 08: Iteratieve ontwikkelaanpak

Bij een iteratieve ontwikkelaanpak werkt het team aan de software in iteraties waarbinnen telkens ongeveer dezelfde werkzaamheden plaatsvinden. Denk hierbij aan het ontwerpen van een wijziging aan de software, het programmeren van deze wijziging, het uitvoeren van een code review op de gewijzigde broncode, het testen van de wijziging, het uitvoeren van regressietesten en het uitvoeren van beveiligingstesten.

Iteraties kunnen een vaste tijdslengte hebben waarbij een wisselend aantal wijzigingen per iteratie wordt gerealiseerd, of ze kunnen een wisselende tijdslengte hebben omdat per iteratie een vast aantal wijzigingen wordt gerealiseerd.

Omdat tijdens iteraties het hele team min of meer gelijktijdig aan een beperkt aantal wijzigingen werkt, is er veel gelegenheid tot mondelinge communicatie. Er is minder noodzaak om via documenten te communiceren, zeker vergeleken met een situatie waarbij een programmeur een ontwerp implementeert dat een ontwerper weken of maanden eerder heeft gemaakt.

Het werk wordt over iteraties verdeeld op volgorde van prioriteit zodat wijzigingen met een hogere prioriteit eerder worden gerealiseerd. Elke iteratie leidt tot een versie van de software die opleverbaar is (Maatregel 07) en voldoet aan de functionele en niet-functionele eisen (Maatregel 10).

OPMERKING Voorbeelden van een aanpak met iteraties met een vaste tijdslengte zijn Scrum en Extreme Programming. Een voorbeeld van een aanpak met iteraties zonder vaste lengte is Kanban.

Deze beheersmaatregel helpt de volgende risico's te verminderen:

— Risico 03: De hoeveelheid werk is niet correct ingeschat waardoor de geplande functionaliteit niet op tijd af is;

— Risico 07: Functionele eisen krijgen te veel prioriteit waardoor het product de juiste niet-functionele eigenschappen ontbeert;

— Risico 08: De communicatie tussen belanghebbenden is suboptimaal waardoor misverstanden ontstaan.

7.2.3.3 Maatregel 09: Geautomatiseerde ontwikkelpijplijn inrichten

Door een geautomatiseerde ontwikkelpijplijn in te richten die bij elke wijziging van de software automatisch de software bouwt, testen en kwaliteitscontroles uitvoert en hierover rapporteert aan de ontwikkelaars, wordt de kans op niet-tijdig gedetecteerde regressies kleiner.

Vanwege de doorlooptijden van testen (met name van duurtesten) en licenties van testtools is het niet altijd haalbaar de hele ontwikkelpijplijn te automatiseren. Idealiter voert de ontwikkelpijplijn de volgende activiteiten uit:

— bouw van de software;

— unit-testen;

— integratietesten;

— kwaliteitscontroles;

— functionele testen;

— performancetesten;

— beveiligingstesten;

— installatie van de software;

— oplevering van het totale product, dus inclusief alle deliverables, in de vorm zoals bruikbaar voor en afgesproken met de opdrachtgever.

De organisatie voorziet teams van ondersteuning en hulpmiddelen zodat zij deze pijplijn kunnen toepassen. De teams zijn zelf verantwoordelijk voor de correcte werking van de ontwikkelpijplijn.

Deze beheersmaatregel helpt de volgende risico's te verminderen:

— Risico 01: De software wordt gewijzigd waardoor de kwaliteit van de software verslechtert;

— Risico 02: De omgeving verandert waardoor de kwaliteit van de software verslechtert.

7.2.3.4 Maatregel 10: Voortdurend voldoen aan de eisen met regressietests

Regressietests zijn tests die verifiëren of eerder ontwikkelde software nog steeds correct werkt na wijzigingen in de desbetreffende software of in de omgeving van de software.

Regressietests testen de functionaliteit van de software, maar kunnen ook niet-functionele eigenschappen van de software testen. Denk aan het testen van performance, het testen van beveiliging en het testen van toegankelijkheid.

OPMERKING 1: Denk bij performancetesten aan load-, duur- en stresstesten. Vaak kan hetzelfde testscript worden gebruikt voor alle drie de performancetestsoorten. Bij de loadtest wordt het testscript met normale productiebelasting gedraaid om te meten of de responstijden en verwerkingstijden voldoen aan de eisen. Bij de duurtest wordt het testscript met normale productiebelasting gedraaid, maar veel langer om mogelijke geheugenlekken en andere resource-uitputting te ontdekken. Bij stresstesten wordt het testscript met toenemende belasting gedraaid, tot voorbij het moment waarop de software de belasting niet meer goed kan verwerken en de responstijden toenemen en/of er fouten optreden.

OPMERKING 2: Denk bij beveiligingstools in elk geval aan een tool voor het statisch analyseren van broncode op beveiligingsrisico's, een tool voor het dynamisch analyseren van de software op beveiligingsrisico's en een tool voor het rapporteren over bekende kwetsbaarheiden in gebruikte bibliotheken.

OPMERKING 3: Denk bij toegankelijkheid aan tools die (een deel van) de webrichtlijnen [WCAG, 2018] controleren.

Om de kwaliteit van de regressietests te bewaken kan de codedekking van de regressietests worden gemeten. Codedekkingsrapportages maken inzichtelijk welke onderdelen (modules, klassen, functies en/of regels) van de broncode wel of niet worden uitgevoerd bij het draaien van een regressietest. Een regressietest kan fouten in niet-gedekte onderdelen niet aan het licht brengen.

Essentieel voor de effectiviteit van deze maatregel is dat de regressietests worden onderhouden. Wanneer nieuwe functionaliteit wordt geïntroduceerd, zullen hiervoor gebruikte testscripts behoren te worden toegevoegd aan de regressietests. Bij gewijzigde functionaliteit behoort te worden onderzocht of ook de regressietests moeten worden gewijzigd.

Niet alle testen kunnen eenvoudig worden geautomatiseerd. Functionaliteit en kwaliteitsaspecten die niet geautomatiseerd kunnen worden getest, worden periodiek handmatig getest. Denk hierbij aan het

periodiek uitvoeren van een blackbox-beveiligingstest door een gespecialiseerde beveiligingstester en het uitvoeren van gebruiksvriendelijkheidsonderzoek met gebruikers. Echter, handmatig uitgevoerde regressietests zijn arbeidsintensief, foutgevoelig en veelal afhankelijk van de aanwezigheid van bepaalde medewerkers. De voorkeur is dus om regressietests waar mogelijk te automatiseren zodat ze herhaalbaar zijn en onderdeel kunnen uitmaken van de geautomatiseerde ontwikkelpijplijn (zie Maatregel 09).

Bevindingen uit de handmatige regressietesten worden vastgelegd als onderdeel van de werkvoorraad voor het ontwikkelproces.

Deze beheersmaatregel helpt de volgende risico's te verminderen:

— Risico 01: De software wordt gewijzigd waardoor de kwaliteit van de software verslechtert;

— Risico 02: De omgeving verandert waardoor de kwaliteit van de software verslechtert;

— Risico 07: Functionele eisen krijgen te veel prioriteit waardoor het product de juiste niet-functionele eigenschappen ontbeert.

7.2.3.5 Maatregel 11: Voortgangsbewaking met burndown charts

Door een productdecompositie te maken en de omvang van de onderdelen te schatten (Maatregel 04), incrementeel op te leveren (Maatregel 07) en iteratief te werken (Maatregel 08) ontstaat de mogelijkheid de voortgang te bewaken in termen van nog op te leveren werk. Op elk moment is er een overzicht van nog op te leveren onderdelen van de productdecompositie (vaak 'backlog' genoemd) en de omvang van die onderdelen. Nadat de eerste incrementen zijn opgeleverd, is er bovendien empirische informatie beschikbaar over de snelheid waarmee het team onderdelen oplevert en kunnen er projecties worden gemaakt van de momenten waarop bepaalde onderdelen (of het geheel) zullen worden opgeleverd. Door het delen of zichtbaar ophangen van de burndown chart ontstaat transparantie over de voortgang.

Door de omvang van de nog op te leveren onderdelen op de Y-as van een grafiek te plaatsen en de tijd op de X-as ontstaat een burndown chart (zie figuur 1).

Op basis van de burndown chart kan de hoeveelheid werk opnieuw worden geschat. Indien blijkt dat de hoeveelheid werk niet correct is geschat, kan de planning worden aangepast.

OPMERKING 1 Burndown charts kunnen ook worden gebruikt om de voortgang binnen een iteratie te bewaken.

OPMERKING 2 Indien een waardebepaling kan worden gemaakt van gerealiseerde items, kan er ook een zogenoemde Earned Value Analysis worden uitgevoerd. Bij deze analyse wordt de waarde van gerealiseerde items afgezet tegen de tijd.

Deze beheersmaatregel helpt het volgende risico te verminderen:

— Risico 03: De hoeveelheid werk is niet correct ingeschat waardoor de geplande functionaliteit niet op tijd af is.

7.2.3.6 Maatregel 12: Een officiële producteigenaar met mandaat

Stel een producteigenaar met domeinexpertise vast. De producteigenaar is een uit Scrum afkomstige rol die ervoor zorgt dat het ontwikkelteam aan de juiste user stories werkt, op volgorde van prioriteit. De producteigenaar verzamelt continu wensen en eisen, prioriteert deze en communiceert deze aan

het ontwikkelteam. De producteigenaar moet voldoende mandaat hebben om te beslissen wat het ontwikkelteam uitvoert.

Deze beheersmaatregel helpt de volgende risico's te verminderen:

— Risico 04: Er vinden scope-uitbreidingen plaats waardoor het product niet tijdig en binnen budget wordt opgeleverd;

— Risico 06: Gebrekkige aansturing van het werk waardoor het product niet de juiste functionaliteit biedt;

— Risico 08: De communicatie tussen belanghebbenden is suboptimaal waardoor misverstanden ontstaan;

— Risico 09: Onvoldoende traceerbaarheid van ontwikkeling, gebruik en beheer van maatwerksoftware leidt tot het niet (aantoonbaar) voldoen aan verplichtingen.

7.2.3.7 Maatregel 13: Toepassen van een kwaliteitgedreven ontwikkelmethode

Een kwaliteitgedreven ontwikkelaanpak begint met het specificeren van testen of acceptatiecriteria, zodat voorafgaand aan het programmeren van incrementen of stories reeds gespecificeerd is aan welke criteria deze getoetst gaan worden. Dit maakt het voor het team duidelijk wanneer het werk aan een increment of story gereed is. Bovendien helpt het om de vraagstelling en de oplossing concreter te maken. Ten slotte kunnen de specificaties worden gebruikt om feedback te krijgen van de opdrachtgever, voordat er software gaat worden gebouwd.

Enkele bekende kwaliteitgedreven ontwikkelmethoden zijn test-driven development (TDD), acceptance test-driven development (ATDD) en behavior-driven development (BDD).

Deze beheersmaatregel helpt de volgende risico's te verminderen:

— Risico 01: De software wordt gewijzigd waardoor de kwaliteit van de software verslechtert;

— Risico 02: De omgeving verandert waardoor de kwaliteit van de software verslechtert;

— Risico 05: Het team beschikt niet over de juiste expertise waardoor de software niet voldoet aan de gestelde eisen.

7.2.4 Afsluiting van de ontwikkeling en/of het onderhoud

7.2.4.1 Maatregel 14: Archivering

Na afronding van de ontwikkeling en/of het onderhoud van de software wordt het werk expliciet afgesloten. Alle documentatie, broncode, referentiedata en 'credentials' die tijdens de werkzaamheden nodig waren of zijn opgeleverd, worden gearchiveerd en van werkstations van medewerkers verwijderd. Archiveren faciliteert het eventueel herstarten van het werk of overdragen van het product op een later tijdstip. Verwijderen neemt een onnodig risico op inbreuk op vertrouwelijkheid weg en vrijwaart medewerkers en organisatie van verdenking en aansprakelijkheid wanneer een incident optreedt.

Deze beheersmaatregel helpt het volgende risico te verminderen:

— Risico 09: Onvoldoende traceerbaarheid van ontwikkeling, gebruik en beheer van maatwerksoftware leidt tot het niet (aantoonbaar) voldoen aan verplichtingen.

7.2.4.2 Maatregel 15: Deugdelijke overdracht

Als de maatwerksoftware door een andere organisatie zal worden onderhouden, behoort er deugdelijke overdracht plaats te vinden van de software, documentatie en testmiddelen. Voor de overdracht zorgen betrokken partijen dat:

— de documentatie een correcte afspiegeling is van de ontwikkel- en testomgeving die is toegepast;

— er documentatie aanwezig is over gegevensmodellen, de functionele indeling, de beschrijving van koppelingen en berichtdefinities en (werk)processen;

— er documentatie aanwezig is over back-up/recovery, procedures bij calamiteiten, regelmatig terugkerende beheeractiviteiten, opstart- en afsluitprocedures;

— er een lijst van bekende tekortkomingen aanwezig is;

— de wijzigingsgeschiedenis van broncode en bijbehorende documentatie aanwezig is;

— de broncode een gezonde balans kent tussen isolatie, cohesie en koppeling;

— de mate van duplicatie in de broncode beperkt dan wel onderbouwd is;

— het aantal mogelijk te doorlopen paden in individuele onderdelen van de broncode dusdanig beperkt is dat hun werking te testen valt;

— sporen van niet-afgeronde werkzaamheden afwezig of beperkt in aantal of in omvang zijn;

— een afdoende deel van de broncode tijdens de testuitvoering wordt geraakt;

— niet-functionele eisen door testgevallen worden geraakt;

— alle onderkende productrisico's door een testplan en testgevallen zijn gedekt;

— er een regressietest (handmatig of geautomatiseerd) beschikbaar is die de werking van de software afdoende waarborgt;

— er een traceerbare koppeling is tussen testgevallen en eisen;

— de opbouw van de testset goed is gestructureerd;

— versiebeheer en versiegeschiedenis worden overgedragen.

Zie verder NPR 5325.

Deze beheersmaatregel helpt de volgende risico’s te verminderen:

— Risico 01: De software wordt gewijzigd waardoor de kwaliteit van de software verslechtert;

— Risico 05: Het team beschikt niet over de juiste expertise waardoor de software niet voldoet aan de gestelde eisen;

— Risico 09: Onvoldoende traceerbaarheid van ontwikkeling, gebruik en beheer van maatwerksoftware leidt tot het niet (aantoonbaar) voldoen aan verplichtingen.

7.3 Organisatiegerichte beheersmaatregelen

7.3.1 Maatregel 16: Teams met specialistische kennis en hulpmiddelen ondersteunen

Veel van de beheersmaatregelen veronderstellen of vereisen bepaalde specialistische kennis en hulpmiddelen. Vaak geldt dat het inefficiënt is als elk team die zelf verwerft, implementeert en onderhoudt. De organisatie stelt kennis en hulpmiddelen waarvoor dat geldt ter beschikking aan de teams. Het gaat daarbij om de volgende hulpmiddelen:

1) templates voor documenten, zoals projectstartarchitectuurdocument, softwarearchitectuurdocument, kwaliteitseisen, projectplan en dergelijke;

2) een tool dat iteratief en incrementeel werken ondersteunt. Een dergelijk tool voorziet in het opvoeren van eisen, opvoeren van logische testgevallen en koppelen van logische testgevallen aan eisen, bijhouden van een werkvoorraad (backlog), plannen van iteraties en toewijzen van eisen aan iteraties;

3) een tool dat het inrichten en uitvoeren van een geautomatiseerde ontwikkelpijplijn ondersteunt;

4) een tool dat het monitoren van de kwaliteit van broncode ondersteunt;

5) een tool dat het releasen van software ondersteunt;

6) een tool dat het maken van testrapportages ondersteunt;

7) een tool dat het maken van kwaliteitsrapportages ondersteunt;

8) een tool voor versiebeheer en indien nodig een configuratiemanagementdatabase (CMDB);

9) beveiligingssoftware die de configuratie van de applicatie, en de omgeving waarbinnen die applicatie draait, controleert op bekende en veelvoorkomende kwetsbaarheden;

10) beveiligingssoftware die de door de applicatie gebruikte versies van externe bibliotheken, raamwerken of andersoortige bouwblokken scant op bekende kwetsbaarheden;

11) beveiligingssoftware die de broncode geautomatiseerd controleert op het voorkomen van bekende onveilige constructies;

12) een tool dat de broncode geautomatiseerd beoordeelt op het hanteren van coding-standaarden.

De organisatie onderhoudt de genoemde hulpmiddelen en ondersteunt teams bij de configuratie en toepassing ervan.

Naast hulpmiddelen stelt de organisatie specialistische kennis ter beschikking op het vlak van kwaliteitsaspecten die van groot belang zijn voor de meeste softwareproducten die de organisatie ontwikkelt en onderhoudt, zoals:

— beveiliging;

— performance;

— gebruiksvriendelijkheid en toegankelijkheid (toegankelijkheid is een wettelijke verplichting voor de publieke sector, zie EN 301 549);

— privacy.

Verder ondersteunt de organisatie de teams bij het inhuren en/of aannemen van nieuwe medewerkers en het inwerken van deze nieuwe medewerkers. Medewerkers zijn nieuw als ze van buiten de organisatie instromen, maar in grotere organisaties kan het zinvol zijn medewerkers die vanuit andere afdelingen instromen ook als nieuw te beschouwen.

De ondersteuning bestaat uit hulpmiddelen, zoals standaard profielen, een proces voor intakes/sollicitatietrajecten en screening, een onboarding-traject, opleidingstrajecten in de vorm van traineeships, e-learning-voorzieningen, enz.

De ondersteuning bestaat ook uit richtlijnen voor het selecteren van medewerkers zodat bij het inhuren/aannemen van nieuwe medewerkers een consistente afweging kan worden gemaakt tussen criteria als kwaliteit, ervaring, opleidingsniveau en termijn waarop een medewerker beschikbaar zal zijn.

De ondersteuning bestaat verder uit richtlijnen voor het bemensen van nieuwe teams. Denk aan richtlijnen voor de gewenste mix van medewerkers voor wat betreft diversiteit, ervaringsniveau en de mate waarin ze bekend zijn met de werkwijze binnen de organisatie.

Deze beheersmaatregel helpt de volgende risico's te verminderen:

— Risico 05: Het team beschikt niet over de juiste expertise waardoor de software niet voldoet aan de gestelde eisen;

— Risico 09: Onvoldoende traceerbaarheid van ontwikkeling, gebruik en beheer van maatwerksoftware leidt tot het niet (aantoonbaar) voldoen aan verplichtingen;

— Risico 10: Doordat veel tijd nodig is voor het invullen van randvoorwaarden voor de softwareontwikkeling wordt het product niet tijdig opgeleverd.

7.3.2 Maatregel 17: Continu risicomanagement

Deze NPR beoogt beheersmaatregelen te geven voor veelvoorkomende risico's bij maatwerksoftwareontwikkeling en -onderhoud, maar pretendeert geen compleetheid. Bovendien loopt elke organisatie die maatwerksoftwareontwikkeling en -onderhoud doet, naast de in hoofdstuk 6 genoemde generieke risico's, ook project- en organisatie-specifieke risico's. Om deze risico's vroegtijdig te onderkennen en te mitigeren is het aan te bevelen continu risicomanagement uit te voeren om zowel project- als organisatie-specifieke risico's te blijven identificeren en mitigeren.

Continu risicomanagement is een proces en bestaat uit de volgende activiteiten, die voor zowel organisatie, als project, als team kunnen worden toegepast:

1) identificatie van risico's;

2) inschatting van de gevolgen. Bepaal onder andere kosten, planning, gevaren, technische performance, geschiktheid, functionaliteit;

3) analyse en prioritering van de risico's van meest kritisch naar minst kritisch;

4) risicomitigatie. Bepaal voor welke risico's risicoreductie moet worden toegepast, waarom voor deze risico's, door middel van welke beheersmaatregel en welke risico's worden geaccepteerd en waarom;

5) implementatie. Plan de implementatie van de risicobeheersmaatregelen en monitor de behaalde risicoreductieresultaten;

6) monitoring. Voeg alle nieuw geïdentificeerde risico's toe aan een lijst en houd deze actueel door deze periodiek te reviewen en te updaten met de behaalde risicoreductieresultaten.

Daarnaast is het zinvol een mechanisme in te richten om risico's die in projecten en teams worden geïdentificeerd, maar eigenlijk project- of teamoverstijgend zijn, organisatiebreed te kunnen analyseren en mitigeren.

Zie ISO/IEC 16085.

Bijlage A

Overzicht risico's en beheersmaatregelen

Tabel A1 — overzicht van welke risico's door welke beheersmaatregelen worden verminderd

Maatregelen	Risico 01 [a]	Risico 02 [b]	Risico 03 [c]	Risico 04 [d]	Risico 05 [e]	Risico 06 [f]	Risico 07 [g]	Risico 08 [h]	Risico 09 [i]	Risico 10 [j]
Maatregel 01: Belanghebbenden identificeren en betrekken			x		x	x	x	x		
Maatregel 02: Belangrijke niet-functionele eisen identificeren					x		x			
Maatregel 03: Belangrijke functionele eisen identificeren					x					
Maatregel 04: Productdecompositie in incrementeel oplever-bare delen met business-waarde			x	x						
Maatregel 05: Technische schuld identificeren, inzichtelijk maken en planmatig oplossen			x				x			
Maatregel 06: Oplossingsrichtingen verkennen			x							
Maatregel 07: Incrementele oplevering van het product			x	x			x			
Maatregel 08: Iteratieve ontwikkel-aanpak			x				x	x		
Maatregel 09: Geautomatiseerde ontwikkelpijplijn inrichten	x	x								
Maatregel 10: Voortdurend voldoen aan de eisen met regressietests	x	x					x			
Maatregel 11: Voortgangsbewaking met burndown charts			x							

Maatregelen	**Risico 01** [a]	**Risico 02** [b]	**Risico 03** [c]	**Risico 04** [d]	**Risico 05** [e]	**Risico 06** [f]	**Risico 07** [g]	**Risico 08** [h]	**Risico 09** [i]	**Risico 10** [j]
Maatregel 12: Een officiële product-eigenaar met mandaat				x		x		x	x	
Maatregel 13: Toepassen van een kwaliteitgedreven ontwikkelmethode	x	x			x					
Maatregel 14: Archivering									x	
Maatregel 15: Deugdelijke overdracht	x				x				x	
Maatregel 16: Teams met specialistische kennis en hulpmiddelen ondersteunen					x				x	x
Maatregel 17: Continu risicomanagement										

[a] Risico 01: De software wordt gewijzigd waardoor de kwaliteit van de software verslechtert
[b] Risico 02: De omgeving verandert waardoor de kwaliteit van de software verslechtert
[c] Risico 03: De hoeveelheid werk is niet correct ingeschat waardoor de geplande functionaliteit niet op tijd af is
[d] Risico 04: Er vinden scope-uitbreidingen plaats waardoor het product niet tijdig en binnen budget wordt opgeleverd
[e] Risico 05: Het team beschikt niet over de juiste expertise waardoor de software niet voldoet aan de gestelde eisen
[f] Risico 06: Gebrekkige aansturing van het werk waardoor het product niet de juiste functionaliteit biedt
[g] Risico 07: Functionele eisen krijgen te veel prioriteit waardoor het product de juiste niet-functionele eigenschappen ontbeert
[h] Risico 08: De communicatie tussen belanghebbenden is suboptimaal waardoor misverstanden ontstaan
[i] Risico 09: Onvoldoende traceerbaarheid van ontwikkeling, gebruik en beheer van maatwerksoftware leidt tot het niet (aantoonbaar) voldoen aan verplichtingen
[j] Risico 10: Doordat veel tijd nodig is voor het invullen van randvoorwaarden voor de softwareontwikkeling, wordt het product niet tijdig opgeleverd

Bijlage B

Assessmentinstrument

B.1 Algemeen

Eén van de aanbevelingen in deze NPR is het initieel en regelmatig uitvoeren van een risico-inventarisatie of een risico-(self)assessment. Om het initiële en actuele risiconiveau te onderzoeken, vast te stellen en zorgvuldig te documenteren kan worden gebruikgemaakt van verschillende hulpmiddelen. Een voorbeeld van zo'n hulpmiddel is het in deze bijlage beschreven assessmentinstrument dat ook in de bijgevoegde zip-file is gevoegd.

Het assessmentinstrument is opgebouwd uit enerzijds de in deze NPR genoemde risico's en anderzijds de per risico aangegeven mogelijke maatregel(en). Het assessmentinstrument helpt bij het inzichtelijk maken van de mate waarin een bepaalde maatregel effect heeft op het verminderen van een risico. Het kan zijn dat het effect van een maatregel zodanig wordt ingeschat of blijkt te zijn, dat hiermee het risiconiveau op een acceptabel niveau komt. Het kan ook zijn dat er meerdere maatregelen nodig zijn om het risico op een acceptabel niveau te krijgen. Of het kan zijn dat er geen maatregelen uit deze NPR mogelijk zijn om dit specifieke risico op een acceptabel niveau te brengen. In dat geval kan alleen het risico als zodanig worden geaccepteerd en goed gemonitord, of er kan worden besloten het project niet uit te voeren. Dit alles wordt inzichtelijk gemaakt met dit instrument.

Belangrijk is wel dat elke waarde in het instrument is opgebouwd uit schattingen vanuit meerdere disciplines en er een gewogen oordeel is gevormd. Hierbij behoort te worden vermeden dat de mening van een bepaalde partij van doorslaggevende betekenis is, zonder dat dit uitvoerig is onderbouwd. Het documenteren van dergelijke standpunten en de onderbouwing is daarom van cruciaal belang. Ook hierbij kan het assessmentinstrument helpen.

Dit onderzoek kan met een vooraf vastgestelde regelmaat steeds opnieuw worden uitgevoerd. Hierbij wordt gekeken of het actuele risiconiveau, conform het verwachte effect door het toepassen van maatregelen, in stand is gebleven. Is dit niet het geval, dan vraagt dit uiteraard om nader onderzoek en nieuwe of andere maatregelen.

B.2 Werkwijze

De werkwijze bij het onderzoek is als volgt:

— Een multidisciplinair team, van bij voorkeur ervaren professionals die (inhoudelijk) betrokken zijn bij de opdracht, neemt alle mogelijke risico's door.

— Het team beoordeelt de toepasselijkheid van de in het instrument gegeven risico's voor het onderhanden project.

— Is het risico van toepassing, dan wordt dit aangegeven. Is het niet van toepassing, dan wordt een onderbouwing gegeven waarom dat niet het geval is. Onderbouwing met bewijzen verdient uiteraard de voorkeur.

— Risico's en effecten uit eerdere en/of soortgelijke projecten, bij voorkeur voorzien van kengetallen (urenregistraties, financiële verslaglegging, gebruikersfeedback en/of onderhoudservaringen), kunnen dienen als uitgangspunt.

— Bij gebrek aan bewijsvoering, bijvoorbeeld omdat wordt gebruikgemaakt van nieuwe techniek, nieuwe toepassing of een combinatie die nog niet eerder is voorgekomen, behoort de gezamenlijke mening van de betrokken experts als onderbouwing te worden genomen.

B.3 Invulinstructies assessmentinstrument

— Beoordeel op het tabblad 'Risico-inventarisatie' welke risico's van toepassing zijn (door het invullen van 'WAAR' of 'ONWAAR').

— Als alle risico's die een maatregel helpen te mitigeren in het tabblad 'Risico-inventarisatie' als niet van toepassing zijn beoordeeld, is de desbetreffende maatregel op het tabblad 'Maatregelbeoordeling' grijs. De maatregel kan immers niet bijdragen aan het verlagen van de geïdentificeerde risico's.

— Beoordeel op het tabblad 'Maatregelbeoordeling' de mate waarin elk van de maatregelen actueel, compleet, consistent en correct is uitgevoerd in de organisatie:

  — Actueel: is de maatregel recent en op de juiste momenten uitgevoerd?

  — Compleet: worden alle beschreven onderdelen afgedekt door de maatregel zoals daadwerkelijk uitgevoerd in de organisatie?

  — Consistent: is de uitvoering van de maatregel hetzelfde bij verschillende projecten, teams en softwareproducten?

  — Correct: wordt de maatregel uitgevoerd zoals bedoeld?

— Beoordeel de maatregelen met één van de volgende scores:

  -- : De maatregel wordt niet of niet correct toegepast of is te lang geleden uitgevoerd om de gerelateerde risico's significant te kunnen mitigeren.

  - : De maatregel wordt niet volledig, niet consistent, niet op de juiste momenten of onvoldoende correct toegepast om de gerelateerde risico's betrouwbaar te mitigeren.

  + : De maatregel wordt voldoende volledig, consistent, correct en tijdig toegepast om de gerelateerde risico's betrouwbaar te mitigeren.

  ++ : De maatregel wordt dusdanig volledig, consistent, correct en frequent toegepast dat er veel zekerheid is dat de maatregel de gerelateerde risico's significant mitigeert.

— Voor elk van de van toepassing zijnde risico's geeft het werkblad aan in welke mate het risico is gemitigeerd. De mate van mitigatie is hoger als meer maatregelen die bijdragen aan het verlagen van het risico zijn uitgevoerd en/of als de uitvoering een hogere score heeft gekregen op het werkblad 'Maatregelbeoordeling'.

Bibliografie

Kruchten, P., Nord, R.L., Ozkaya, I., & Falessi, D., *Technical Debt: Towards a Crisper Definition; Report on the 4th International Workshop on Managing Technical Debt*. ACM SIGSOFT Software Engineering Notes, September 2013, Volume 38, Number 5, 2013.
https://resources.sei.cmu.edu/asset_files/Article/2013_101_001_424860.pdf

Waarom betaalt u voor een norm?
Normen zijn afspraken voor en door de markt, zo ook deze norm. NEN begeleidt het gehele normalisatieproces. Van het bijeenbrengen van partijen, het maken en vastleggen van de afspraken en het bieden van hulp bij de toepassing van de normen. Om deze diensten te kunnen bekostigen betalen alle belanghebbende partijen die aan tafel zitten voor het normalisatieproces, en u als gebruiker voor normen en trainingen. NEN is een stichting en heeft geen winstoogmerk.

Wat is nu precies de toegevoegde waarde van normen?
Stelt u zich eens voor … u wilt in het buitenland geld pinnen, maar uw bankpas past niet. Of uw nieuwe telefoon herkent uw simkaart niet. De samenstelling van de benzine over de grens is anders, waardoor u niet kunt tanken. Het dagelijks leven zou zonder goede afspraken over producten, processen en diensten een stuk complexer zijn.

Het maken en vastleggen van afspraken door belanghebbende partijen noemen we het normalisatieproces. Normalisatie had vanouds betrekking op techniek en producten. Nu worden steeds vaker normen voor diensten ontwikkeld. Zo zijn er afspraken op het gebied van gezondheidszorg, schuldhulpverlening, kennisintensieve dienstverlening, externe veiligheid en MVO.

Normen zorgen voor verbetering van producten, diensten en processen; qua veiligheid, gezondheid, efficiëntie, kwaliteit en duurzaamheid. Dit ziet u op de werkvloer, in de omgang met elkaar en in de samenleving als geheel. Organisaties die normalisatie onderdeel van hun strategie maken, vergroten hun professionaliteit, betrouwbaarheid en concurrentiekracht.

Wat doet NEN?
NEN ondersteunt in Nederland het normalisatieproces. Als een partij zich tot NEN richt met de vraag om een afspraak tot stand te brengen, gaan wij aan de slag. We onderzoeken in hoeverre normalisatie mogelijk is en er interesse voor bestaat. Wij nodigen vervolgens alle belanghebbende partijen uit om deel te nemen. Een breed draagvlak is een randvoorwaarde. De afspraken komen op basis van consensus tot stand en worden vastgelegd in een document. Dit is meestal een norm. Afspraken die in een NEN-norm zijn vastgelegd mogen niet conflicteren met andere geldige NEN-normen. NEN-normen vormen samen een coherent geheel. Een belanghebbende partij kan een producent, ondernemer, dienstverlener, gebruiker, maar ook de overheid of een consumenten- of onderzoeksorganisatie zijn.

De vraag is niet altijd om een norm te ontwikkelen. Vanuit de overheid komt regelmatig het verzoek om te onderzoeken of er binnen een bepaalde sector of op een bepaald terrein normalisatie mogelijk is. NEN doet dan onderzoek en start afhankelijk van de uitkomsten een project. Deelname staat open voor alle belanghebbende partijen. NEN beheert ruim 30.000 normen. Dit zijn de in Nederland aanvaarde internationale (ISO, IEC), Europese (EN) en nationale normen (NEN). In totaal zijn er ruim 800 normcommissies actief met in totaal bijna 5.000 normcommissieleden. Een goed beheer van de omvangrijke normencollectie en de afstemming tussen nationale, Europese en internationale normcommissies vereisen dan ook een zeer goede infrastructuur.

Betalen kleine organisaties net zoveel als grote organisaties?
Het uitgangspunt is dat alle partijen die deelnemen aan het normalisatieproces een evenredig deel betalen. De normcommissieleden kunnen onderling andere afspraken maken. Zo worden er wel eens afspraken gemaakt dat de grote partijen een groter deel betalen dan de kleinere bedrijven. De prijzen voor normen zijn voor iedereen gelijk. De kosten voor licenties zijn afhankelijk van de omvang van een organisatie en het aantal gebruikers.

Voordelen van normalisatie en normen
Gegarandeerde kwaliteit | Veiligheid geborgd | Bevordert duurzaamheid | Opschalen en vermarkten van nieuwe innovatieve producten | Meer (internationale) handelsmogelijkheden | Verhoogde effectiviteit en efficiëntie | Onderscheidend in de markt.

Voordelen van deelname
Invloed op de (internationale en Europese) afspraken | Als eerste op de hoogte van veranderingen | Netwerk; ook op Europees en internationaal niveau | Kennisvergroting.