

Software developer

Hans van Rheenen

Boom beroepsonderwijs
info@boomberoepsonderwijs.nl
www.boomberoepsonderwijs.nl

Auteur: Hans van Rheenen
Redactie en opmaak: Henk Pel
Titel: Software developer
ISBN 978 90 372 5744 1
Illustraties: Erik Eshuis
Eerste druk / eerste oplage
© Boom beroepsonderwijs 2020

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

Voor zover het maken van reprografische verveelvoudigingen uit deze uitgave is toegestaan op grond van artikel 16h Auteurswet dient men de daarvoor wettelijk verschuldigde vergoedingen te voldoen aan de Stichting Reprorecht (www.reprorecht.nl). Voor het overnemen van gedeelte(n) uit deze uitgave in compilatiewerken op grond van artikel 16 Auteurswet kan men zich wenden tot de Stichting PRO (www.stichting-pro.nl).

De uitgever heeft ernaar gestreefd de auteursrechten te regelen volgens de wettelijke bepalingen. Degenen die desondanks menen zekere rechten te kunnen doen gelden, kunnen zich alsnog tot de uitgever wenden.

Door het gebruik van deze uitgave verklaart u kennis te hebben genomen van en akkoord te gaan met de specifieke productvoorwaarden en algemene voorwaarden van Boom beroepsonderwijs, te vinden op www.boomberoepsonderwijs.nl.

Inhoud

Inleiding 9

B1-K1: Realiseert software 9

B1-K2: Werkt in een ontwikkelteam 10

1 Projectmatig werken 13

Project 13

1.2 Het projectplan 16

Taken en rollen 17 Fasering en planning 20

1.3 Opgaven 22

1.4 Projectomschrijving projectplan 23

2 Ontwikkelmethode 25

2.1 Watervalmethode 25

SDM 26 Fasen van SDM 26

2.2 Spiraalmethode 28

Scrum 29 | Hoe Scrum werkt – het begin 30 | Professionals in een Scrum 30 | Hoe Scrum werkt – het vervolg 32 | eXtreme Programming (XP) 35

2.3 Prototyping 38

Waarom prototyping? 38 | Wanneer prototyping? 39 | Hoe prototyping? 39

2.4 Opgaven 39

3 Veilig en betrouwbaar software ontwikkelen 43

3.1 Software Development Life Cycle (SDLC) 43

Requirements 43 | Design 44 | Coding 44 | Testing 44 | Deployment 44

3.2 Informatiebeveiliging 45

Beveiligingsprincipes 47

3.3 Secure Software Development Life Cycle (S-SDLC) 48

Requirements 49 | Design 52 | Coding 54 | Testing 55 | Deployment 55

3.4 Opgaven 57

4 Overleg voeren 61

4.1 De opdracht vaststellen 62

4.2 Luisteren, samenvatten en doorvragen (LSD) 62

Luisteren 63 | Samenvatten 64 | Doorvragen 64

4.3 Opgaven 67

- 4.4 Vragen die gesteld moeten worden 69
Algemene vragen 69 | Vragen over onderhoud 69
- 4.5 Programma van eisen 70
- 4.6 Opgaven 71
- 4.7 Rapport Programma van eisen 71
Eisen 72
- 5 Ontwerp 73**
 - 5.1 Functioneel ontwerp 73
Requirements 73 | Unified Modeling Language (UML) 74 | Structuur- en gedragsdiagrammen 75 | Use-case-diagram 75 | De use-case-tabel 79
 - 5.2 Opgaven 81
 - 5.3 Klasse en klassendiagram 83
Relaties in een klassendiagram 84 | Inheritance relationships (overerving) 85 | Compositie en aggregaties 86 | Ontwerpen van een klassendiagram 87 | Voorbeeld ontwerpen van een klassendiagram voor TopSpin 87 | Voorbeeld ontwerpen van klassendiagram Air-concepts 89
 - 5.4 Opgaven 95
 - 5.5 Functioneel-ontwerp-rapport 100
 - 5.6 Technisch ontwerp 103
 - 5.7 Activity diagram (activiteitendiagram) 104
 - 5.8 Opgaven 110
 - 5.9 Sequence diagram (sequentiediagram) 112
Voorbeeld snoepautomaat 112 | Voorbeeld aanmeldprocedure 114
 - 5.10 Opgaven 114
 - 5.11 Relatieve datamodel 116
Relationele database 116 | Normaliseren 117 | Normaliseren volgens Codd 119 | Diagrammen 126
 - 5.12 Opgaven 128
 - 5.13 Technisch-ontwerp-rapport 134
 - 5.14 Uitgewerkt voorbeeld 137
- 6 Realiseren van software 139**
 - 6.1 Voorbereiden van de realisatie 139
 - 6.2 De ontwikkelomgeving installeren en configureren 142
 - 6.3 De ontwikkelomgeving testen 143
 - 6.4 Instellingen en wijzigingen documenteren 143
 - 6.5 Versies en versie nummers 144
Compatibel 145
 - 6.6 Opgaven 146

7	Testen van software	147
7.1	Versiebeheer	147
7.2	Functioneel testen	148
	<i>Testplan</i>	149 <i>Testdocumentatie</i>
	<i>149</i> <i>Beperkingen bij het testen</i>	151 <i>Manieren van testen</i>
	<i>152</i>	
7.3	Testtype	156
	<i>Regressietest</i>	156 <i>Acceptatietest</i>
	<i>157</i> <i>Functioneel testen versus niet-functioneel testen</i>	157 <i>Continu testen</i>
	<i>157</i> <i>Destructief testen</i>	157 <i>Softwareprestatietesten</i>
	<i>157</i> <i>Usability testing</i>	158 <i>Security-test</i>
	<i>158</i>	
7.4	Testproces	159
	<i>Agile- of extreme-ontwikkelingsmodel</i>	159 <i>Top-down en bottom-up</i>
	<i>160</i> <i>De testcyclus</i>	160
7.5	Opgaven	161
7.6	Testplan	162
7.7	Testrapport	163
8	Opleveren van software	165
8.1	Implementatie	165
	<i>'Big Bang' of schaduwdraaien</i>	167 <i>Acceptatietest</i>
	<i>168</i> <i>Een testplan opstellen</i>	170 <i>De testomgeving inrichten</i>
	<i>170</i> <i>Testscenario's of testcases maken</i>	170 <i>Testplan</i>
	<i>175</i> <i>Testformulier</i>	176 <i>Testrapport</i>
	<i>177</i>	
8.2	Opgaven	179
8.3	Het product presenteren	180
	<i>De presentatie maken</i>	181
	<i>Presenteren</i>	184
8.4	Opgaven	185
8.5	Het opgeleverde product evalueren	185
	<i>Verzamelen van gegevens</i>	185 <i>Evaluatierapport</i>
	<i>186</i>	
9	Onderhoud en beheer van software	189
9.1	Application Services Library 2 (ASL 2)	189
9.2	Processen op strategisch niveau	190
	<i>Applications Cycle Management</i>	190 <i>Organization Cycle Management</i>
	<i>190</i>	
9.3	Processen op tactisch niveau	191
	<i>Contractmanagement</i>	191 <i>Planning en control</i>
	<i>191</i> <i>Kwaliteitsmanagement</i>	192 <i>Financieel management</i>
	<i>192</i> <i>Leveranciersmanagement</i>	192
9.4	Processen op operationeel niveau	193
	<i>Gebruikersondersteuning</i>	194 <i>Configuratiebeheer</i>
	<i>194</i> <i>Operationele ICT-sturing</i>	194 <i>Continuïteitsbeheer</i>
	<i>194</i>	

- 9.5 Verbindende processen 195
 - Wijzigingsbeheer 195 | Programmabeheer en distributie 195*
- 9.6 Opgaven 195
- 9.7 Softwarelicenties 196
 - Commerciële software 196 | Software aanschaffen 198 | Shareware en freeware 198 | Licenties 199*
- 9.8 Opgaven 201

- 10 Wetgeving 203**
 - 10.1 Algemene verordening gegevensbeveiliging (AVG) 203
 - 10.2 AVG en opslag van data 204
 - Aan de volgende regels moet worden voldaan 204 | Rechten van betrokkenen 207*
 - 10.3 Geldigheid van de AVG 208
 - Wettelijke basis voor verwerking 209 | Datalekken 209*
 - 10.4 Opgaven 211

Inleiding

Om een diploma te kunnen krijgen moet je aan een hele reeks voorwaarden voldoen. Deze zijn opgesteld in samenwerking met het bedrijfsleven en vastgesteld door de minister van Onderwijs, Cultuur en Wetenschap. Hieronder staan de kennis en vaardigheden waaraan je moet voldoen om als beginnend beroepsbeoefenaar aan de slag te kunnen als Software developer (Crebonr. 25604). Het volledige kwalificatiedossier is te vinden op de site van de Stichting Samenwerking Beroeps- onderwijs Bedrijfsleven, oftewel S-BB (<https://www.s-bb.nl>).

Programmeren en coderen zijn de belangrijkste bezigheden van een developer. Dit boek gaat daar niet over. Dit boek gaat over alle zaken die om het maken van code heen van belang zijn. In weke omgeving je iets ontwikkelt zal sterk afhangen van het bedrijf en/of de toepassing. Op de site van Brinkman Uitgeverij kun je voor de verschillende omgevingen lesmateriaal vinden.

Het gaat daarbij om het werkproces *B1-K1-W3: Realiseert (onderdelen van) software* met de volgende eindtermen:

- heeft specialistische kennis van de principes van object oriented programming (OOP), waaronder encapsulation, modularity, inheritance, polymorphism;
- heeft specialistische kennis van één of meer programmeertalen (syntax en semantiek);
- kan diagrammen lezen, interpreteren en maken zoals UML;
- kan één of meerdere programmeertalen voor softwareontwikkeling toepassen (syntax en semantiek);
- kan één of meerdere softwareontwikkelingsmethodieken toepassen zoals Waterval, iteratief of incrementeel;
- kan één of meerdere softwareontwikkelingsprogramma's toepassen zoals IDE's;
- kan één of meerdere softwareontwikkelingstechnieken toepassen zoals objectgeoriënteerd programmeren, ECS, functioneel programmeren;
- kan gegevensverzamelingen omzetten in andere structuren;
- kan ontwerpeisen toepassen;
- kan technieken voor informatiebeveiliging toepassen.

B1-K1: Realiseert software

De beginnend beroepsbeoefenaar:

- Heeft brede kennis van cyber security en bedreigingen van netwerken en systemen.
- Heeft brede kennis van relevante zaken zoals wetgeving op het gebied van privacy, copyright en auteursrecht, computercriminaliteit.
- Heeft brede kennis van nieuwe ontwikkelingen op het gebied van software, zoals AI (artificial intelligence), machine learning en big data.

- Heeft kennis van ontwikkelingen op het vlak van ICT-infrastructuur en devices en welke consequenties deze op software development hebben.
- Heeft specialistische kennis van de IT-infrastructuur en/of ontwikkelingsplatforms waarop de software wordt toegepast.
- Heeft specialistische kennis van de principes van object oriented programming (OOP), waaronder encapsulation, modularity, inheritance, polymorphism.
- Heeft specialistische kennis van één of meer programmeertalen (syntax en semantiek).
- Heeft specialistische kennis van één of meerdere software-ontwikkelingsmethodieken zoals Waterval, iteratief of incrementeel.
- Heeft specialistische kennis van één of meerdere software-ontwikkelingsprogramma's zoals IDE's.
- Heeft specialistische kennis van één of meerdere testing tools en testtechnieken.
- Heeft specialistische kennis van één of meerdere software-ontwikkelingstechnieken zoals OOP, ECS of functioneel programmeren.
- Heeft specialistische kennis van licenties en gebruiksrechten.
- Kan diagrammen lezen, interpreteren en maken zoals UML.
- Kan één of meerdere programmeertalen voor software-ontwikkeling toepassen (syntax en semantiek).
- Kan één of meerdere software-ontwikkelingsmethodieken toepassen zoals Waterval, iteratief of incrementeel.
- Kan één of meerdere software-ontwikkelingsprogramma's toepassen zoals IDE's.
- Kan één of meerdere software-ontwikkelingstechnieken toepassen zoals objectgeoriënteerd programmeren, ECS, functioneel programmeren.
- Kan gegevensverzamelingen omzetten in andere structuren.
- Kan ontwerpeisen toepassen.
- Kan technieken voor informatiebeveiliging toepassen.
- Kan actief versiebeheer toepassen.
- Kan relevante wetgeving op het gebied van privacy, intellectueel eigendomsrecht, computercriminaliteit toepassen op software.
- Kan principes van Secure Software Development Life Cycle (SSDLC) toepassen.
- Kan controleren of een ontwerp van software voldoet aan gangbare beveiligingseisen en de bevindingen toelichten aan betrokkenen.

B1-K2: Werkt in een ontwikkelteam

De beginnend beroepsbeoefenaar:

- Heeft specialistische kennis van één of meerdere software-ontwikkelingsmethodieken zoals Waterval, iteratief of incrementeel.
- Heeft specialistische kennis van één of meerdere software-ontwikkelingsprogramma's zoals IDE's.
- Heeft specialistische kennis van één of meerdere testing tools en testtechnieken.
- Heeft specialistische kennis van één of meerdere software-ontwikkelingstechnieken zoals OOP, ECS of functioneel programmeren.
- Kan met betrokkenen communiceren over werkzaamheden.

- Kan gesprekstechnieken toepassen (zoals luisteren, samenvatten, doorvragen).
- Kan presentatietechnieken toepassen.
- Kan projectmatig werken.
- Kan relevante wetgeving op het gebied van privacy, intellectueel eigendomsrecht, computercriminaliteit toepassen op software.

1 Projectmatig werken

Als ontwikkelaar van software zul je bijna altijd aan projecten werken. Soms iets heel kleins, dat je alleen oplost, soms als onderdeel van een heel groot project, met veel deelprojecten. Maar meestal iets dat daartussenin zit qua omvang. Het overleg dat je moet voeren heeft dan ook vaak te maken met jouw rol binnen het project. Binnen een project zal er sprake zijn van een bepaalde vorm van projectorganisatie. Hoe en wat hangt van de gebruikte methodiek af. De meest voorkomende aspecten met betrekking tot een project worden hier behandeld.

Project

Anders dan een organisatie is een project eindig. Een project heeft een duidelijk eindpunt dat van tevoren is gedefinieerd. Een project is een eenmalige activiteit met een duidelijk eindresultaat. Het wordt meestal vastgelegd in een projectplan en geregeld door het projectmanagement. Binnen een project is de inzet van mensen geregeld en vastgelegd. Een van de definities van een project is:

een project is een unieke opgave, begrensd in tijd en middelen en afgesloten met een projectresultaat

De kenmerken van een project zijn: tijd, middelen, product en management.

Tijd

Een project is een tijdelijk gegeven. Een project kan een week duren of een aantal jaren, maar zal nooit oneindig doorlopen. Een project heeft een van tevoren beoogd eindpunt. Het komt voor dat bij de eerste planning van een project de einddatum nog niet is vastgesteld. Dit is het geval als de duur van de tussenliggende stappen nog onduidelijk is.

Middelen

Binnen een project spreekt men van *resources*. Resources zijn alle middelen die binnen het project ingezet worden. Dit zijn zowel menselijke inspanningen als materialen. Beide vormen van resources brengen kosten met zich mee en bepalen de totale kostprijs van het project. Men streeft ernaar om de in te zetten resources bij aanvang van het project vast te leggen en in te schatten.

Product

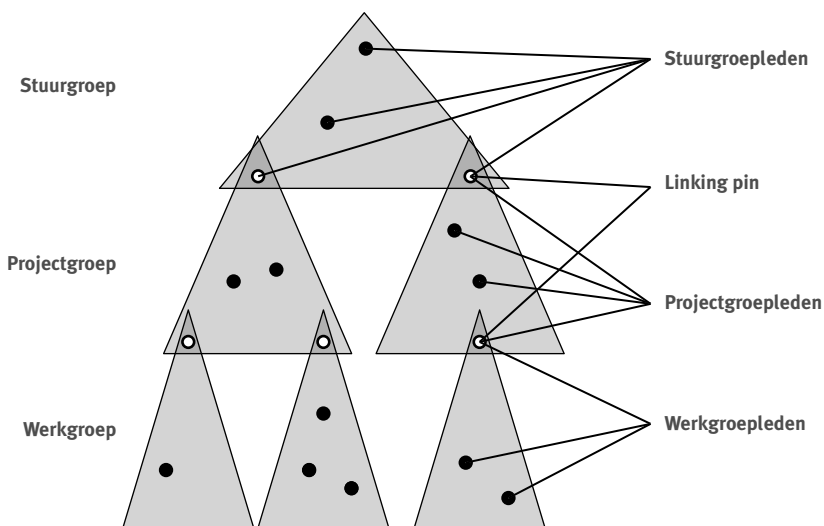
Op de einddatum moet het project uitmonden in een eindproduct. Er wordt een project opgezet om iets uitzonderlijks te realiseren. De normale organisatie gaat gewoon door.

Management

Een project heeft een eigen projectmanagement. Het projectmanagement moet ervoor zorgen dat het project met de beschikbare middelen binnen de afgesproken tijd tot een goed einde wordt gebracht.

Linking pin

Veelgebruikt is het linking-pin-principe. Er wordt een stuurgroep ingericht. Die stuurgroep richt één of meer projectgroepen in en die richten ieder weer één of meer werkgroepen in.



Er is een duidelijke taakverdeling binnen deze groepen.

Stuurgroep

De stuurgroep heeft de leiding over het gehele project. De stuurgroepleden houden zich niet bezig met de dagelijkse werkzaamheden voor het project. Zij komen af en toe bijeen, meestal één- of tweemaal per maand. Aan de stuurgroep wordt gerapporteerd door de linking-pin-leden van de projectgroep over de voortgang van elk project. De stuurgroep schept de voorwaarden wat betreft tijd en geld om deze voortgang te realiseren.

De stuurgroep bestaat uit slechts een klein aantal personen. De samenstelling kan bestaan uit een directielid, de accountant, de ICT-manager en de projectleider.

Projectgroep

De projectgroep heeft de dagelijkse leiding over het project. De projectgroep komt vaak, minimaal eens per week, bij elkaar. De voorzitter van de projectgroep is meestal de projectleider en in die hoedanigheid ook lid van de stuurgroep. De andere projectgroepleden zijn verantwoordelijk voor de producten die het project moet opleveren.

Naast de projectleider kunnen in de projectgroep de volgende functionarissen zitten:

- applicatiebeheerders
 - automatiseerders
 - controlefunctionarissen
 - materiedeskundigen (eindgebruikers)
 - organisatieadviseurs
 - personeelsfunctionarissen
 - technische systeembeheerders
- De samenstelling is afhankelijk van de aard van het project.

Werkgroep

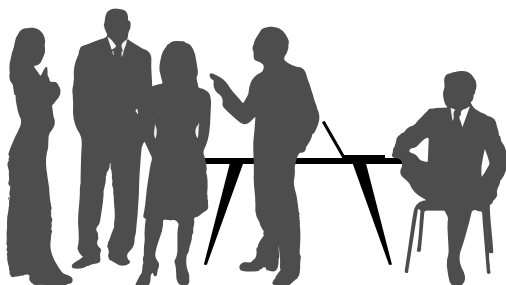
Om ervoor te zorgen dat ieders talenten goed tot hun recht komen en niet iedereen zich met alles gaat of hoeft te bemoeien, worden er werkgroepen ingericht. Een werkgroep heeft een specifieke taak binnen het project. Elke werkgroep is belast met één aspect van het project.

Deze werkgroepen zijn verantwoordelijk voor hun eigen resultaten. De werkgroep-leider is de linking pin naar de projectgroep en is verantwoording verschuldigd aan de projectgroep.

Mogelijke werkgroepen:

- Administratieve organisatie
- Conversie
- Invoering
- Juridische aspecten (Wet persoonsregistratie)
- Nazorg
- Ontwikkeling van de verschillende deelsystemen
- Opleiding
- Testwerkgroep

Gebruikers zijn in de werkgroep een belangrijk onderdeel. Zij weten waar het over gaat en kunnen vanuit hun professie een belangrijke bijdrage leveren aan een werkgroep.



Waar je ook in een project wordt ingezet, vaak zal je deelname aan het project

maar een deel van je tijd beslaan. Daarom is het belangrijk dat er duidelijke afspraken worden gemaakt, ook over jouw inzet. Wanneer, waar en met welke bevoegdheid neem je deel aan het project? Zelfs als alles goed is afgesproken, kun je nog in moeilijke situaties terechtkomen. Je hebt namelijk te maken met twee verschillende leidinggevers. Je hebt een functioneel leidinggevende in de vorm van je directe chef. Maar in de werkgroep heb je te maken met een operationeel leidinggevende in de vorm van de werkgroepleider of de projectgroepleider. Wat ga je doen als je baas zegt: 'We hebben een spoedklus, vandaag mag je even niet aan het project werken'?

Als je het goed doet, laat je dat door de twee leidinggevers zelf uitzoeken. Breng ze met elkaar in contact en laat ze een oplossing bedenken. Wijs naar de afspraken zoals ze zijn gemaakt bij de inrichting van het project.

1.2 Het projectplan

Software kan op basis van veel verschillende methodieken worden ontwikkeld. Maar in bijna alle gevallen ligt er een vraag van een klant aan ten grondslag en op basis daarvan een projectplan. De vraag van de klant zal vaak resulteren in een Pakket van Eisen (PvE) waaraan de ontwikkelaar moet voldoen. Het projectplan zal in veel gevallen een Plan van Aanpak (PvA) bevatten. We onderscheiden drie 'hoofdmanieren' op het gebied van software-ontwikkeling: de Watervalmethode, prototyping of de iteratieve methode.

In alle gevallen zal er sprake zijn van de behoefte van een klant, waaraan voldaan moet worden. Een klant wil graag weten waar hij aan toe is: 'Wanneer beginnen we?', 'Wanneer is het klaar?', 'Wat gaat het kosten?', 'Wat wordt er van mij verwacht?', 'Wie zijn er nog meer bij betrokken?' – en mogelijk zijn er nog meer vragen. Door een projectplan te schrijven wordt het voor een klant tastbaarder en krijgt hij een belangrijk deel van de antwoorden op zijn vragen.

Wanneer met de klant overeengekomen is dat er iets gemaakt gaat worden moet er een plan gemaakt worden van de aanpak. Een projectplan wordt gebruikt om de uitvoering van het project te begeleiden. Het belangrijkste doel van het projectplan is de plannings-aannames en beslissingen te documenteren. Op basis van het projectplan wordt dan gemonitord of mijlpalen gehaald worden en of er moet worden bijgesteld. Het projectplan kan bestaan uit de volgende onderdelen:

- Inleiding
- Projectomschrijving
 - Aanleiding
 - Doelen
 - Resultaat
 - Afbakening
 - Risico's
 - Effecten
 - Randvoorwaarden

- Fasering en planning
- Projectbeheersing (van tijd, geld, kwaliteit, informatie en organisatie)

Taken en rollen

Taken

In een goed projectteam zijn de taken slim verdeeld. Het team bestaat uit specialisten met kennis en ervaring. Deze kennis en ervaring moeten zoveel mogelijk benut worden. De webdeveloper kan best een site-ontwerp maken, maar is er wellicht een mediadeveloper die het nog beter kan?

Naast het specialistische werk moeten in een project ook dingen gebeuren die niet zijn voorbehouden aan specialisten. De projectleider zal in de meeste gevallen de vergaderingen voorzitten. Maar wat als de projectleider een beroerde gespreksleider is? Dat doe je elkaar toch niet aan? Kijk eens hoe een ander het als gespreksleiderschap ervan af brengt. Zo zijn er veel zaken die moeten gebeuren. In alle gevallen is het belangrijk dat de persoon die het geschiktst is voor een taak deze ook krijgt toegewezen. Je moet hierbij denken aan:

- Bijhouden van het projectarchief
- Documenteren
- Eindredactie van rapporten
- Maken van de verslagen
- Onderhandelen met opdrachtgevers
- Uittesten van eerste versies

Je wilt graag in het projectteam spelvreugde en creativiteit ontwikkelen en ervoor zorgen dat iedereen goed functioneert.

Rollen

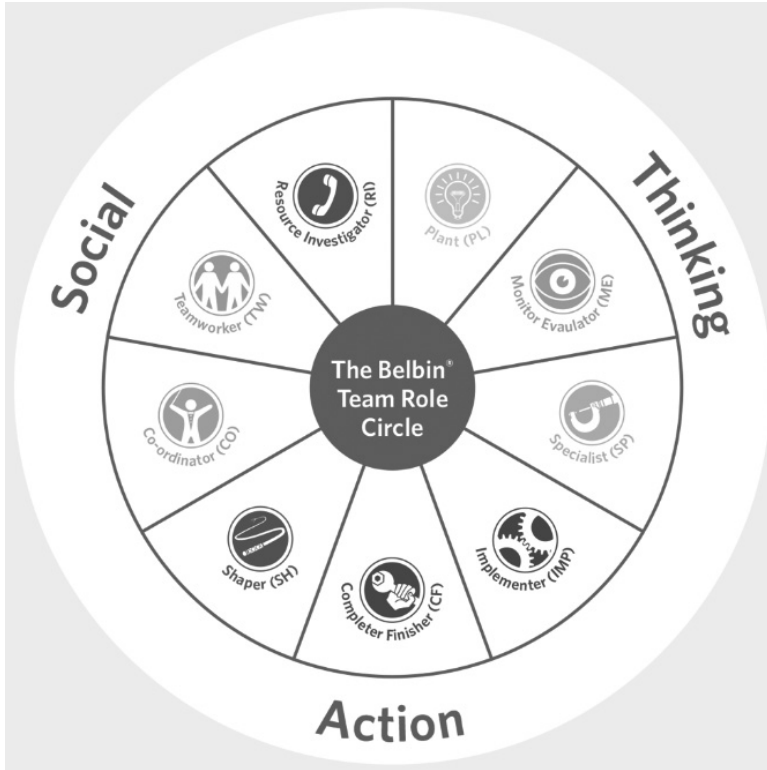
In een team zijn er niet alleen taken, maar ook rollen. Een rol is een afspraak over welk gedrag men per situatie vertoont. Het is niet het gedrag wat men altijd vertoont, maar gedrag dat is afgesproken. Je weet precies wat je aan elkaar hebt: 'Als jij een voorstel hebt om af te wijken van het PvA dan ga ik zeuren dat het toch niet kan. Samen zoeken we dan een haalbaar compromis.' 'Als Hans weer eens begint met 'Maar...' dan vraag jij hem niet alleen zijn bezwaar te noemen, maar ook een oplossing voor dat bezwaar.'

Een rolverdeling kan ook dodelijk zijn voor de creativiteit en het elan binnen het team. Toch is een afspraak over de rolverdeling belangrijk: duidelijkheid gaat boven alles! Mogelijk dat je afspraken kunt maken over verschillende rollen die iemand kan spelen.

De belangrijkste teamrollen zijn door de Britse onderzoeker Meredith Belbin onderzocht en als volgt omschreven:

- **Afronder (Completer Finisher)**
 - Consciëntieus
 - Regelt het planmatig verloop
 - Werkt achter de schermen
 - Zorgelijk en bezorgd
- **Brononderzoeker (Resource Investigator)**
 - Communicatief
 - Enthousiast
 - Extravert
 - Gaat op zoek naar ideeën
 - Nieuwsgierig
- **Coördinator (Coordinator)**
 - Doelgericht
 - Kalm
 - Laat ieder teamlid tot zijn recht komen
 - Realistisch en nuchter
 - Zelfvertrouwen
- **Groepswerker (Teamworker)**
 - Gevoelig
 - Kan goed luisteren
 - Mild
 - Sociaal gericht
 - Stimuleert en integreert
- **Monitor (Monitor Evaluator)**
 - Analyseert en bekritiseert
 - Nuchter
 - Weinig emoties
 - Zakelijk
- **Plant (Plant)**
 - Fantasie
 - Individualistisch
 - Intellect, kennis
 - Onorthodoxe ideeën
- **Specialist (Specialist)**
 - Plichtsgetrouw
 - Praktisch
 - Zelfdiscipline
 - Zet beslissingen om in werkzaamheden
- **Uitvoerder (Implementer)**
 - Ordelijk
 - Plichtsgetrouw
 - Taakgericht
 - Weinig flexibel
- **Vormer (Shaper)**

- Dynamisch
- Energiek
- Extravert
- Ongeduldig



Als een team uit een groot aantal personen bestaat, moet om te beginnen worden bepaald welke rollen er van nature aanwezig zijn. De sterkte en/of zwakte van een team kun je met behulp van de Belbin Team Circle in beeld brengen. Als je een grote groep hebt, deel je deelnemers in 'teams' van ongeveer vier tot zes in. Als je met een kleinere groep werkt, hoeft je de groep natuurlijk niet op te splitsen.

- 1 Vraag elke deelnemer zichzelf drie rollen toe te bedelen die het dichtst bij hem in de buurt zitten. Laat dit ook door twee collega's doen. Vergelijk de uitkomst en bepaal samen in welke volgorde de rollen het beste bij die persoon passen.
- 2 Elk team tekent een Belbin-cirkel met negen secties, een voor elk van de teamrollen van Belbin. Van elk teamlid worden de namen in de segmenten gezet die overeenkomen met hun bovenste twee rollen.
- 3 Laat het team vijf hoofdgebieden benoemen waarvan zij denken dat daar hun sterke en zwakte punten liggen.
- 4 Het team moet drie actiepunten bedenken op basis van de bevindingen. Dit moeten actiepunten zijn die ervoor zorgen dat het team beter gaat presteren.

Het kan zijn dat er wat meer detaillering nodig is in de planning. Als voorbeeld de mijlpaal Ontwerp, die uit twee subdelen bestaan, de mijlpaal Functioneel ontwerp en de mijlpaal Technisch ontwerp.

Ook kan het gebeuren dat er meerdere personen bij het project betrokken zijn die ieder hun eigen verantwoordelijkheden hebben. Mieke weet dat zij in week 4 het Functioneel ontwerp moet opleveren wil Ahmad in week 5 kunnen starten met het Technisch ontwerp. Per fase is er duidelijk wie er een rol heeft (actoren).

Voorbeeld 3

VOORBEELD 3			Week																								
Fase	Mijlpalen/activiteiten	Actoren	Verantwoordelijke	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	Pakket van eisen	Mieke/Jesse/Klant	Mieke																								
2	Plan van aanpak	Mieke	Mieke																								
3	Ontwerp																										
3.1	Functioneel ontwerp																										
3.1.1	Vaststellen requirements	Mieke/Jesse/Klant	Mieke																								
3.1.2	Prioriteit requirements	Jesse/Klant	Mieke																								
3.1.3	Maken use case	Jesse	Mieke																								
3.1.4	Opleveren functioneel ontwerp	Mieke/Klant	Mieke																								
3.2	Technisch ontwerp	Ahmed/Jesse	Ahmed																								
4	Realiseren	Ahmed/Jesse	Ahmed																								
5	Implementeren	Mieke/Ahmed/Klant	Mieke																								

Hoe groter een project wordt hoe nauwkeuriger de planning moet worden uitgewerkt. In Voorbeeld 3 zie je een voorbeeld van de uitwerking van de mijlpaal Functioneel ontwerp. Deze bestaat uit de activiteiten: Vaststellen requirements, Prioriteit requirements, Maken use case en Opleveren functioneel ontwerp.

Een nog verdere uitwerking van de planning zou kunnen bestaan uit het aantal uren dat nodig is. Voor de voortgang zou er bijgehouden kunnen worden hoeveel procent van een bepaalde activiteit of mijlpaal al gerealiseerd is.

Projectbeheersing

Onder projectbeheersing zie je voor een deel principes van PRINCE2 terugkomen in combinatie met de punten die zijn behandeld bij de projectomschrijving en de projectplanning. PRINCE2 is de opvolger van de methode PRINCE, die in de jaren tachtig is ontwikkeld voor ICT-projecten. Het bundelen van praktijkervaringen, zogenaamde best practices, is hierbij de kern. PRINCE is door de Britse semi-overheidsinstelling de Office of Government Commerce (OGC) ontwikkeld en wordt in de versie PRINCE2 in veel bedrijven in Europese landen gebruikt. Het gaat om zaken als:

- beheersing van het bestaansrecht
- beheersing projectscope
- beheersing tijd en geld
- beheersing kwaliteit
- beheersing overige randvoorwaarden
- beheersing van de projectrisico's
- beheersing van de projectaanpak

Van belang is dat je hier aangeeft hoe, door wie en wanneer de projectvoortgang in de gaten wordt gehouden. Het zou heel goed kunnen zijn dat de oplevering van elke mijlpaal of activiteit een moment is om even te kijken of en hoe de volgende stap moet worden gezet.

1.3 Opgaven

- 1 Wat is de belangrijkste input voor het projectplan?
- 2 Wat zijn de belangrijkste taken in een projectteam software-ontwikkeling?
- 3 Wat is een rol?
- 4 Over welke eigenschappen moet een Groepswerker beschikken?
- 5 Over welke eigenschappen moet een Uitvoerder beschikken?
- 6 Vul met een team van 4 tot 6 personen de Belbin Team Circle in. Bepaal eerst over welke eigenschappen elk teamlid beschikt. Volg de stappen zoals die zijn beschreven in paragraaf 1.2.
- 7 Uit welke onderdelen kan een projectplan bestaan?
- 8 Wat is het belangrijkste dat moet worden beschreven bij het onderdeel Resultaat van een projectomschrijving?
- 9 Wat is het belang van het benoemen van risico's in de projectomschrijving?
- 10 Geef een definitie van planning.
- 11 Wat is het belang van projectbeheersing?
- 12 Maak de planning in Voorbeeld 2 verder af. Gebruik hierbij de volgende gegevens.
 - Stappen Technisch ontwerp:
 - Vertaal het Functioneel ontwerp naar technische specificaties.
 - Ontwerp een grafisch ontwerp van de user interface.
 - Ontwerp een relationeel datamodel.
 - Lever het Technisch ontwerp.
 - Stappen Inrichten ontwikkelomgeving:
 - Inventariseer de benodigde onderdelen voor de realisatie.
 - Installeer en configureer de ontwikkelomgeving.
 - Test de ontwikkelomgeving.
 - Documenteer de instellingen en wijzigingen.
 - Stappen Realisatie:
 - Opleveren deelproduct 1.
 - Opleveren deelproduct 2.
 - Opleveren deelproduct 3.
 - Opleveren deelproduct 4.
 - Stappen Implementatie:
 - Stel een acceptatietest op voor de applicatie.
 - Begeleiden van de acceptatietest.

- Beschrijf eventuele aanpassingen.
- Werk de documentatie van de applicatie bij.
- Evalueer het product met de betrokkenen.
- Evalueer het proces met de betrokkenen.
- Leg dit vast in een evaluatieverslag.
- Laat het evaluatieverslag accorderen.

1.4 Projectomschrijving projectplan

De projectomschrijving kan bestaan uit de volgende onderdelen:

- aanleiding
- doelen
- resultaat
- afbakening
- planning
- risico's
- randvoorwaarden

Aanleiding

Onder dit kopje geef je aan wat de aanleiding is om het project te starten. Vaak is dat een probleem dat opgelost moet worden.

Doelen

Hier geef je aan wat het doel is dat met de uitvoering van het project moet worden bereikt. Omdat een project meestal bestaat uit een aantal elkaar opvolgende fasen zul je per fase de doelen omschrijven. Dit worden ook wel de mijlpalen genoemd.

Resultaat

Bij het resultaat moet exact worden aangegeven wat er opgeleverd moet worden. Hierbij moet je zo concreet mogelijk elk op te leveren product benoemen. Benoem de cruciale succesfactoren, de onderdelen waarop geconcludeerd kan worden of het project is geslaagd of juist mislukt. Het is verstandig om een duidelijke afbakening te maken door ook de zaken te benoemen die niet door het project worden opgeleverd of door de beperkingen te noemen. Dit noemen we ook wel de mijlpaalproducten.

Afbakening

Heel belangrijk is het managen van verwachtingen. Dat doe je door duidelijk aan te geven wat er wel en wat er niet wordt gedaan. Hoe duidelijker je dit doet, hoe minder 'gezeur' je hoeft te verwachten. De afbakening kan je halen uit de MoSCoW-methode (zie paragraaf 4.5) die in het Programma van eisen is opgesteld.

Planning

Een planning is het voor de toekomst systematisch voorbereiden. Dit doe je door het op elkaar afstemmen en het nemen van besluiten die noodzakelijk zijn om een bepaald doel te bereiken. De detaillering zal afhangen van de aard en de omvang van het project. Het toepassen van een strokenplanning is meestal behulpzaam.

Risico's

Aan elk project zijn risico's te koppelen die een succesvol afsluiten van het project in de weg kunnen staan. Vaak zijn die al in een vroeg stadium zichtbaar. Door ze duidelijk te benoemen, zijn ze goed in beeld en kan erop gestuurd worden om ze zoveel mogelijk te vermijden.

Een risico is de kans dat iets fout gaat. Maar dat is niet exact waar het hierom gaat. Het risico dat er een taalfout op de website staat is van een heel andere orde dan dat de website niet op tijd afkomt. Wat heb je aan een website met de aankondiging van een evenement als dat evenement al voorbij is? Risico's moeten ook gewogen worden. De mate van impact moet worden meegenomen. Daar kan de projectleider dan op sturen.

Randvoorwaarden

Met het benoemen van de randvoorwaarden of ook wel de uitgangspunten waaronder het project uitgevoerd moet worden, kun je een deel van de risico's verkleinen. De beschikbaarheid van mensen en middelen is hierbij een belangrijk onderdeel, zoals ook wie verantwoordelijk is voor de realisatie. Dit maakt het de projectleider makkelijk om mensen aan te spreken als er problemen dreigen te ontstaan.