

# Inhoud

---

Deel 1: de context van projectmatig werken .....	4
1. Projecten identificeren .....	4
1.1. Projecten in de bedrijfswereld .....	4
1.2. Projecten herkennen .....	4
2. Projecten aanpakken .....	5
2.1. Projectbouwsteen: deliverable .....	5
2.2. Pad 1: projecten uitvoeren .....	5
2.3. Pad 2: projecten leiden .....	6
3. Organiseren tot projectmatig werken .....	6
3.1. Projectsucces .....	6
3.2. Waarom projecten falen .....	7
3.3. Projecten in een organisatie .....	7
3.4. De uitgebreide projectomgeving: belangrijke rollen en verantwoordelijkheden .....	8
Deel 2: projecten opleveren .....	9
4. Process requirements .....	9
4.1. Beschrijving en doel .....	9
4.2. Rollen en verantwoordelijkheden .....	10
4.3. Deliverables .....	10
4.4. Methodologieën .....	10
5. Concept .....	10
5.1. Beschrijving en doel .....	10
5.2. Rollen en verantwoordelijkheden .....	11
5.3. Deliverables .....	11
5.4. Afhankelijkheden en aandachtspunten .....	12
6. Planning .....	12
6.1. Inleiding .....	12
6.2. Beschrijving en doel .....	12
6.3. Rollen en verantwoordelijkheden .....	13
6.4. Deliverables .....	13
6.5. Afhankelijkheden .....	14
7. Ontwerp .....	15
7.1. Inleiding .....	15
7.2. Beschrijving en doel .....	15
7.3. Rollen en verantwoordelijkheden .....	15

7.4.	Deliverables .....	15
7.5.	Methodologieën .....	16
10.	Ontwikkelstrategieën .....	16
10.1.	Inleiding, beschrijving en doel.....	16
10.2.	Build and fix.....	16
10.3.	Waterfall, lineair, sequentieel model.....	17
10.4.	Rapid prototyping.....	17
10.5.	Incrementeel ontwikkelen.....	18
10.6.	Iteratief ontwikkelen .....	18
10.7.	Agile principe – SCRUM .....	18
10.8.	Continuous delivery.....	20
10.9.	Kanban.....	<b>Fout! Bladwijzer niet gedefinieerd.</b>
11.	Turnover .....	21
11.1.	Inleiding.....	21
11.2.	Beschrijving en doel .....	21
11.3.	Rollen en verantwoordelijkheden .....	21
11.4.	Deliverables .....	21
11.5.	Methodologieën – ITIL.....	22
11.6.	SLA, SLO, SLI.....	23
Deel 3: projecten leiden .....		24
13.	Project plannen .....	24
13.1.	Inleiding en leerdoel.....	24
13.2.	Het belang van een degelijke planning .....	24
13.3.	Projectbaseline 1: scope.....	24
13.4.	Projectbaseline 2: time/schedule.....	25
13.5.	Projectbaseline 3: cost .....	25
13.6.	GOKIT.....	25
13.7.	Risicobeheer .....	25
13.8.	Team en informatie .....	27
13.9.	Overige planningsdomeinen .....	28
Extra's.....		29
ProjectManagementPlan.....		29
Programming practices (Cegeka eBook) .....		29
Scrum guide.....		29
Extra Agile termen.....		30

Legende:

**Begrippen**

Figuren

Concepten

# Deel 1: de context van projectmatig werken

Algemeen // identificeren, aanpakken, organiseren

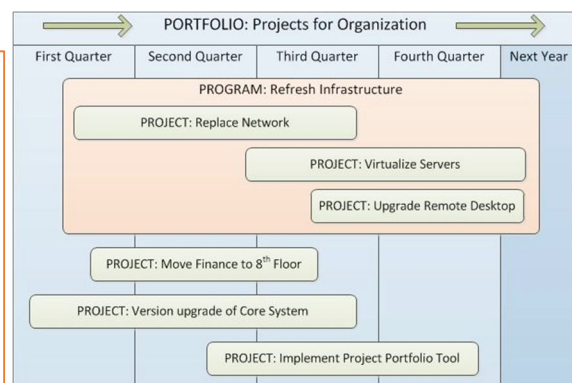
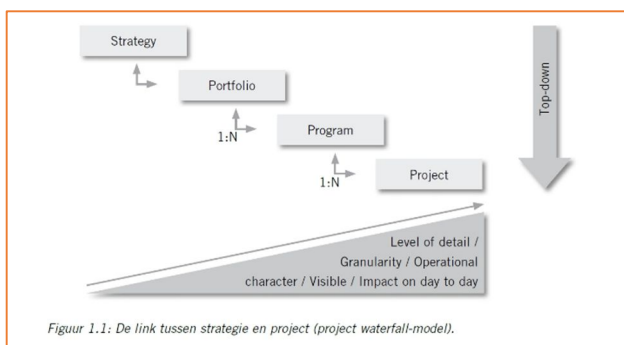
## 1. Projecten identificeren

### 1.1. Projecten in de bedrijfswereld

- **Change management** = goed begeleiden van een veranderingstraject
- **Product** = de kenmerken en functionaliteiten waaraan het resultaat van het project moet voldoen.
- **Project**
  - = de som van alle activiteiten die nodig zijn om een product uit te werken en op te leveren.
  - = een aantal mensen van verschillende disciplines die tijdelijk samenwerken met een beperkt budget om een vooraf bepaald doel te bereiken.
- Kenmerken project
  - Is altijd uniek, eindresultaat of de weg ernaartoe is steeds anders.
  - Heeft een afgeleijd eindproduct.
  - Zijn tijdelijk, met start- en einddatum.
    - Moet worden opgestart
    - Heeft een opdrachtgever
    - Is meestal multidisciplinair
    - Heeft een eigen (vast) budget
    - Heeft een eigen projectorganisatie
- **Programma** = een groep van individuele projecten, zo wordt de consistentie en prioriteit bewaakt.
- **Resource** = een element nodig om een bepaalde activiteit te voltooien (materieel, menselijk, financieel).
- **Operations** = dagelijks werk van een bedrijf of afdeling.

### 1.2. Projecten herkennen

- Oorsprong project (maw triggers die aanleiding geven om project te starten):
  - Extern (mandatory) zoals corona, een wijziging in wettelijk kader...
  - Intern zoals strategische beslissingen...
- Project waterfall-model:



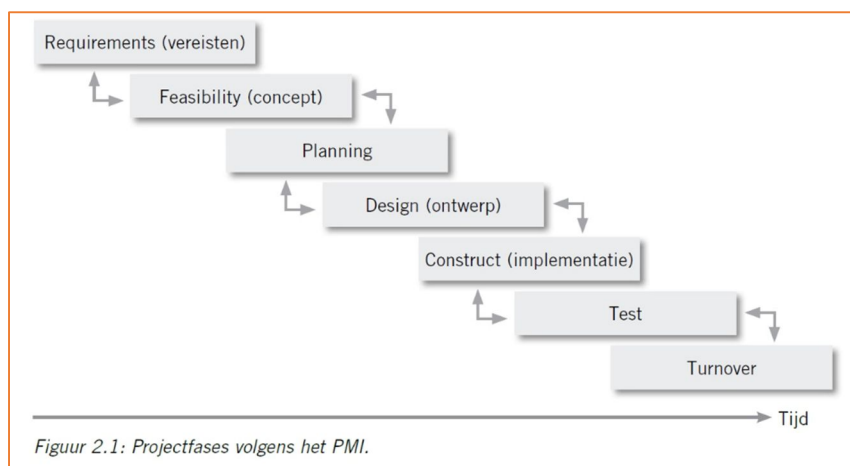
## 2. Projecten aanpakken

### 2.1. Projectbouwsteen: deliverable

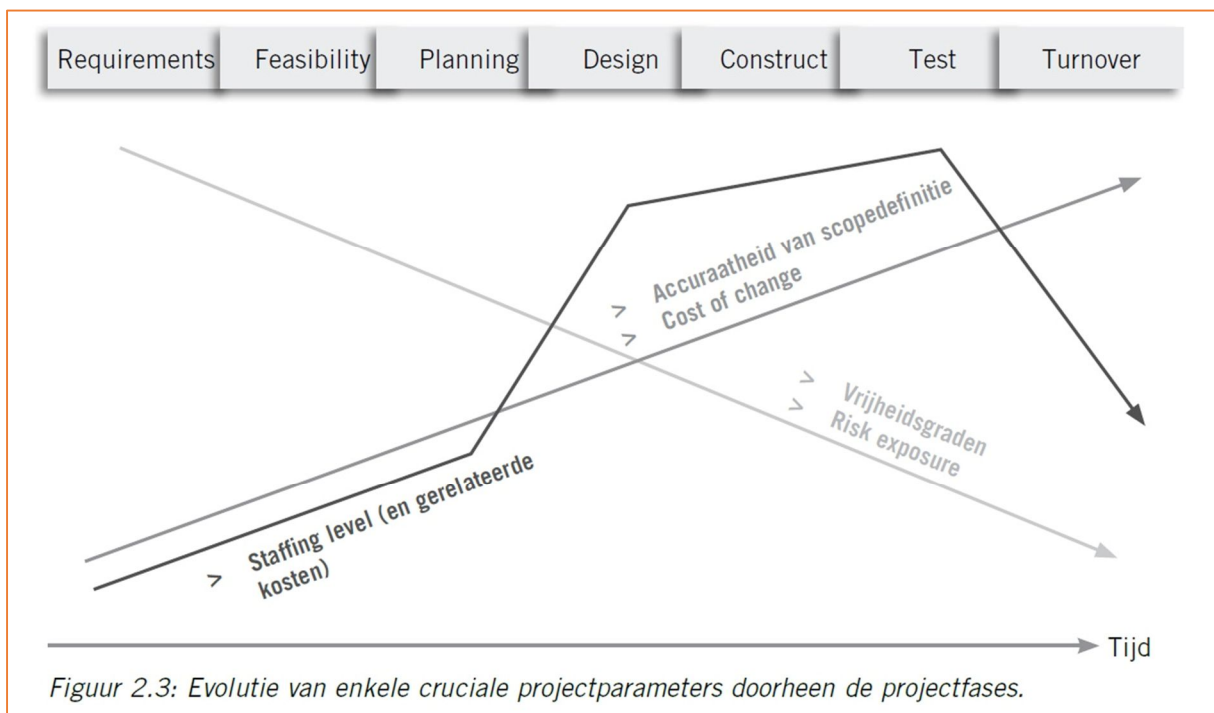
- **Scope** = de inhoudelijke grens van het project.
- **Deliverable** = finaal of tussentijds product dat wordt opgeleverd als resultaat van een projectactiviteit, draagt bij aan het eindproduct van het project.
- **Milestone** = project phase gates = momentopname op de projecttijdlijn dat wordt aangegrepen om de status van oplevering te monitoren, een checkpoint.
- **Projectgrens** = lengte en breedte van een project.

### 2.2. Pad 1: projecten uitvoeren

- **Projectuitvoering** = omvat alle activiteiten die direct bijdragen tot de oplevering van het product.
- Projectfases volgens het PMI:

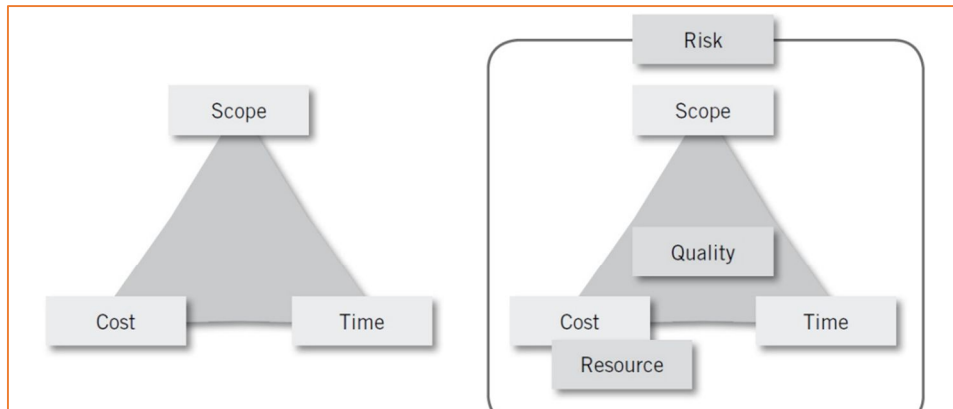


- Projectparameters evolueren tijdens het doorlopen van verschillende projectfases:



## 2.3. Pad 2: projecten leiden

- **Triple constraint:** scope, cost en time moeten nauwgezet in het oog gehouden worden, een verschuiving van een van deze drie componenten leidt tot een wijziging bij de andere twee. Maar ook risk, resource en quality moeten in het oog gehouden worden. (**devil's triangle**)



- If we choose to go for a low budget (cost) and fast paced project (time), then we will have to cut in features (quality).
- If we choose to increase or pace (time), we will have to sacrifice either quality or money (cost).

*Figuur 2.4: Evolutie van de constraints die de projectmanager dient te bewaken.*

- **Projecten leiden** = een discipline die noodzakelijk is om het actieplan uit te tekenen, uit te voeren en te bewaken opdat de projectobjectieven en het product volgens de gemaakte afspraken en op de meest efficiënte wijze wordt gerealiseerd.

## 3. Organiseren tot projectmatig werken

### 3.1. Projectsucces

- Succesvolle oplevering van een project = wordt gekenmerkt door een 100%-match met de vereisten van de klant. Daarin staat de validatie van de deliverables door de klant centraal.
- **Projectsucces/Succesfactoren:**
  - Perspectief van klant: overeenstemming tussen productomschrijving (project scope statement) en productoplevering (gevalideerde deliverables).
  - Perspectief van Projectmanager: bovenstaande + levert het juiste projectresultaat (scope/WAT), is op de afgesproken tijd afgerond (timeschedule/WANNEER), kost niet meer dan afgesproken (cost/KOSTEN).

Projectsucces  
=  
Overeenstemming tussen

Productomschrijving

&

Productoplevering

Project scope  
statement

Gevalideerde  
deliverables

*Figuur 3.1: Een succesvol project vanuit klantenperspectief.*

Projectsucces

=

Overeenstemming tussen

Productomschrijving

&

Productoplevering

+

Performance in lijn met

Projectmanagementplan

Project scope  
baseline

Project schedule  
baseline

Project cost  
baseline

Project  
quality  
management plan

Project risk  
management plan

Project  
contingency  
management plan

...

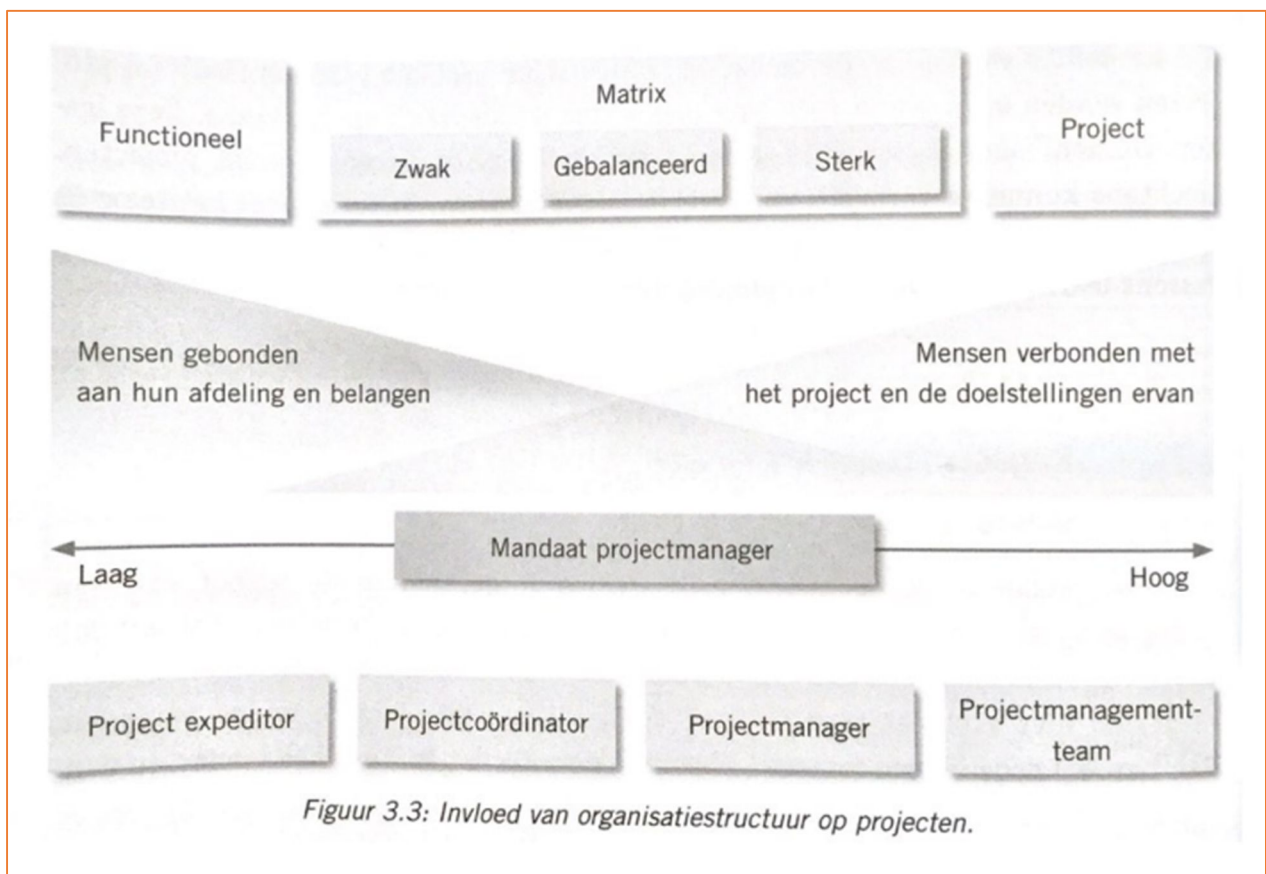
*Figuur 3.2: Een succesvol project vanuit het perspectief van de projectmanager.*

## 3.2. Waarom projecten falen

- **Faalfactoren** met kritieke impact
  - Gebrek aan formeel aanstellen van een projectmanager, projectdoelstellingen worden duidelijk geformuleerd, kennis en kunde van de projectmanager, samenstelling van het projectteam, mandaat van de projectmanager, ondersteuning management...
- **Faalfactoren** met grote impact
  - Gebrek aan ervaring van de projectmanager, goed communicatiemodel, skills in het projectteam, motivatie, risicobeheer, afwezigheid of wegwerken van knelpunten in informatie-uitwisseling, kwaliteit van de inschatting...

## 3.3. Projecten in een organisatie

- **Projectorganisatie** = de verdeling van de taken en verantwoordelijkheden binnen het projectteam en wijze van communiceren in het team en met de opdrachtgever.
- Organisatievormen:
  - **Functioneel** = afdelingen gebaseerd op kennis en vaardigheid. (?)
  - **Matrix** = alle medewerkers die hetzelfde soort werk doen in één afdeling, een werknemer rapporteert aan het hoofd van de afdeling en aan de projectleiders van projectteams. (?)
  - **Project** = werknemers gegroepeerd per project (?).



### 3.4. De uitgebreide projectomgeving: belangrijke rollen en verantwoordelijkheden

- Rollen en verantwoordelijkheden/Skills projectmanager/Projectrollen:

Persoon/projectentiteit	Rol en verantwoordelijkheid
Project Manager Office (PMO)	Fungeert als cement tussen verschillende projecten. Zorgt voor consistentie (bij tools, methodologie, templates...). Is verantwoordelijk voor inzet resources over projecten heen. Speelt een belangrijke rol in projectplanning en projectopvolging.
Functionele Manager	Leidt een afdeling binnen het bedrijf. Voorziet het project van input m.b.t. scope en vereisten.
Proceseigenaar	Beheert het businessproces volledig. Geeft input tijdens concept- en ontwerpfase.
Sleutelgebruiker	Bepaalt de vereisten mee tijdens concept- en ontwerpfase. Is een aanspreekpunt en het klankbord op de werkvloer. Reviewt deliverable en test applicatie als eerste.
Sponsor	Stelt middelen ter beschikking om het project uit te voeren. Initieert de projectvraag. Communiqueert met projectmanager. Volgt het project op van begin tot einde. Is meestal voorzitter van de stuurgroep.
Stuurgroep	Is een centraal rapporterings- en beslissingsorgaan binnen het project. Hebben maandelijkse bijeenkomst en op cruciale momenten.
Business analyst / functional consultant	Ondersteunt en adviseert m.b.t. optimalisatie van processen. <i>AS-IS</i> : inventariseert proactief de wensen, ideeën en problemen. <i>TO-BE</i> : definieert het te bereiken streefdoel. Helpt de interne functionele afdelingen bij evaluatie en verbetering.
Developer / ontwikkelaar / technical consultant	Schrijft technische specificatie uit. Ontwikkelt eigenlijke programmatuur. Test applicatie. Lost defecten/bugs op.
Architect	Bewaakt de big picture m.b.t. IT-architectuur. Is het meest betrokken bij feasibility- concept- en ontwerpfase.
Testmanager	Werkt testplan uit. Spoort aan om testscenario's uit te werken.
Tester	Staat in voor het uitvoeren van de testen. Is onafhankelijk en objectief. Rapporteert aan testmanager.



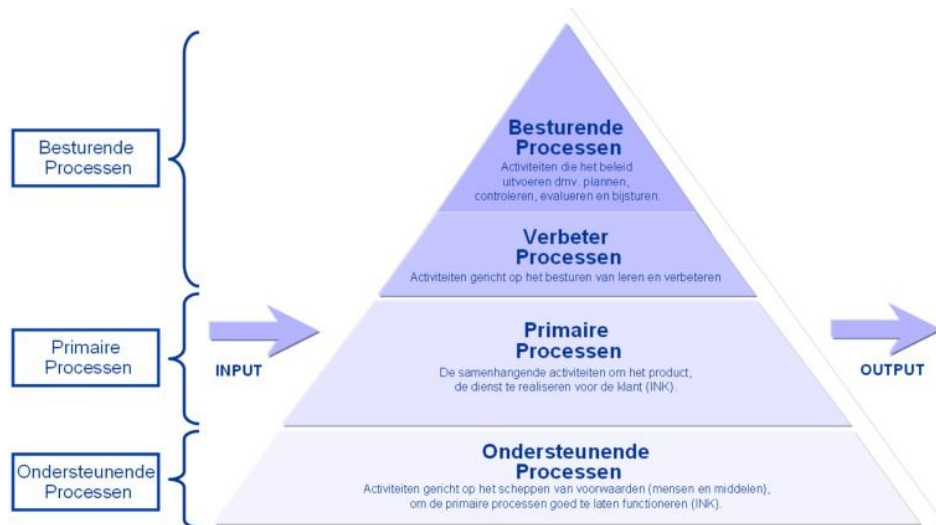
## Deel 2: projecten opleveren

Standpunt teamlid // requirements, concept, planning, ontwerp, implementatie, test, ontwikkelingstrategieën, turnover

### 4. Process requirements

#### 4.1. Beschrijving en doel

- **Process requirements** = de vereisten die een bedrijfsproces heeft ten aanzien van de ondersteunende IT-applicatie. Door processen eerst te beschrijven en daarna te optimaliseren.
- **Bedrijfsproces** = een zich herhalende keten van activiteiten. Deze keten is gericht op de klant en afgestemd op het bedrijf. Bedrijfsprocessen kan je indelen in drie hoofdsoorten:
  - **Besturend** = activiteiten die het beleid uitvoeren dmv plannen, controleren, evalueren en bijsturen.  
=> zijn gericht op het besturen van de organisatie en het inrichten van processen.
  - **Primair** = de samenhangende activiteiten om het product/de dienst te realiseren voor de klant.  
=> hebben direct te maken met het bedrijfsdoel, bv productie of verkoop.
  - **Ondersteunend** = activiteiten gericht op het scheppen van voorwaarden om de primaire processen goed te laten functioneren.  
=> ondersteunen primaire processen, bv administratie of financiële afdeling.



- Het bestuderen van de bestaande huidige bedrijfsprocessen brengt voordelen mee: input voor..
  - Mogelijke business case – de kosten en baten bespreken van het project en aantonen wat de winst is die het project op middellange en lange termijn zal genereren.
  - Blueprint – de uiteindelijke resultaten van het project gedetailleerd beschrijven.
  - Testfase – baseren op de scenario's die werden uitgeschreven in de procesbeschrijvingen. Zo wordt vermeden dat er delen van het proces vergeten worden tijdens het testen.
  - Change management – duidelijk maken welke rollen in het proces beïnvloed worden.

## 4.2. Rollen en verantwoordelijkheden

- **Best practice** = een bepaalde manier van werken of een bepaalde opzet van een applicatie die op basis van verschillende eerdere implementaties tot stand is gekomen en de beste manier van werken identificeert binnen een bepaald domein, een bepaalde taak of applicatie

Persoon/projectentiteit	Rol en verantwoordelijkheid
Business analyst	Stelt de procesbeschrijving op basis van input eind- en sleutelgebr. op. Identificeert mogelijke optimalisaties.
<b>Proceseigenaar</b>	Is verantwoordelijk voor een bepaald (deel)proces. Optimaliseert het proces, lost problemen op. Neemt finale beslissingen.
<b>Eindgebruiker</b>	Zal in de toekomst met de nieuwe applicatie werken.
<b>Sleutelgebruiker</b>	Is een eindgebruiker met een zeer goede kennis van het huidige proces.

## 4.3. Deliverables

- Het eindresultaat van het opstellen van de process requirements is een procesbeschrijving aan de hand van gedetailleerde process flowcharts.
- **Process flowchart** = beschrijft de verschillende stappen van een proces in een specifieke volgorde en geeft aan welke gegevens er worden doorgegeven en welke documenten of andere output er gegenereerd wordt.

## 4.4. Methodologieën

- **BPMN** = is een standaard in business process modeling, tracht de kloof tussen het ontwerpen van bedrijfsprocessen en het implementeren ervan te overbruggen. (Voorbeeld [figuur 4.1](#))
  - Legt businessprocessen vast en visualiseert ze.
  - Duidelijk voor de business analyst en onderhoudsvriendelijk voor eindgebruikers.

## 5. Concept

- **Concept** = het uitgewerkte idee van een project. De business requirements die het project initialiseren zijn geïdentificeerd. In deze fase wordt, onder andere op basis van een business case, nagegaan of een project een voldoende grote opbrengst heeft en of er voldoende ondersteuning kan gevonden worden binnen de onderneming.

### 5.1. Beschrijving en doel

- **Drie soorten projecten**: projecten gedreven door...
  - Factoren van buitenaf – noodzakelijk om het voortbestaan van de onderneming te garanderen. Aandacht aan aanpassingen maken.  
Kenmerken: Beperkte conceptfase, eenvoudige business case, zo snel mogelijke overgaan tot actie.
  - Optimalisatietrajecten – de nood aan een proces efficiënter uit te voeren. Aandacht aan het verzekeren  
Kenmerken: Weerstand verminderen, voldoende gedragen, engagement vanaf conceptfase.
  - Verbeteringstrajecten – geïnitieerd vanuit de eindgebruikers.  
Kenmerken: Veel aandacht voor business case, verkrijgen van engagement lukt beter.
  - ~~Een nood aan onderscheidend vermogen – klanten verwachten een breed gamma van IT-functionaliteiten, bedrijven moeten blijven innoveren, bedrijven moeten niet langer reactief maar net proactief zijn.~~

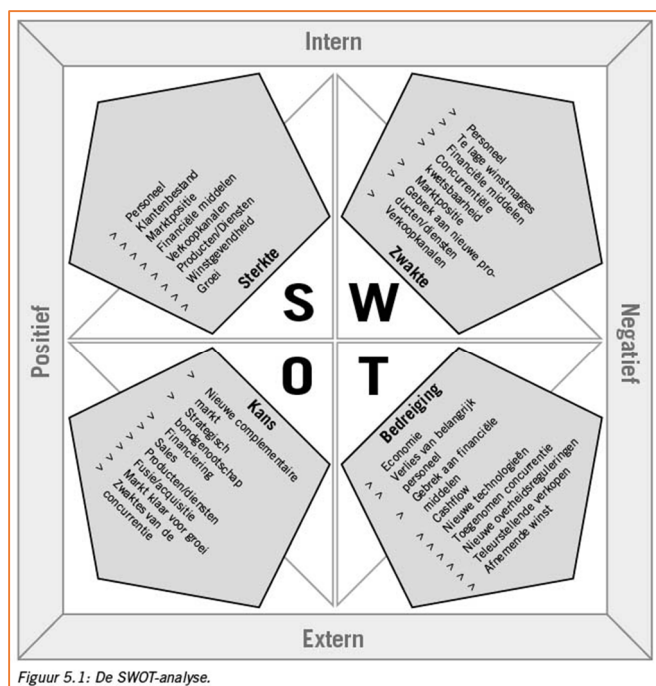
- Algemene activiteiten:
  - Opstellen van business case – een kosten-batenanalyse waarbij er gekeken wordt naar de mogelijke opbrengst van het project op middellange en lange termijn.
  - Voorstelling en goedkeuring aan het management
  - Commerciële fase – een lastenboek uitgegeven waarop externe spelers proposals kunnen schrijven. Op basis daarvan wordt software en leverancier gekozen.
  - Aanstelling projectmanager

## 5.2. Rollen en verantwoordelijkheden

Persoon/projectentiteit	Rol en verantwoordelijkheid
Business analyst	Verantwoordelijk voor het opstellen van de business case.
Ontwikkelaars	Worden betrokken bij het opmaken van de proposal, advies geven over technische haalbaarheid
Eindgebruiker	Leveren de inhoud voor het lastenboek en selecteren software en leveranciers mee.
Projectmanager	Bewaart overzicht van alles wat er zich afspeelt binnen en rond het project.

## 5.3. Deliverables

- **Business case** = geeft een antwoord op de vraag of een project winstgevend zal zijn op de middellange of lange termijn. Op basis van kosten-batenanalyse in de business case zal een project al dan niet opgestart worden.
- Componenten business case:
  - Beschrijft business requirements – waarom een bepaald project dient plaats te vinden, kwalitatieve en kwantitatieve gevolgen, voor- en nadelen dmv een SWOT-analyse...
  - SWOT-analyse – Sterkte, Zwakte, Kans en Bedreiging. Opgesteld door de business analyst of projectmanager in samenwerking met sponsor en eventuele sleutelgebruikers.
  - Economische assessment – er wordt berekend wat de mogelijke return of investment (ROI) van het project zal zijn.
  - Aanbeveling van beslissing.
- **SWOT-analyse** = weegt de sterktes en zwaktes van een project af tegen de kansen en bedreigingen uit de omgeving. Op deze manier krijgt men een beter overzicht van de mogelijke risico's en de gebieden die speciale aandacht nodig zullen hebben.



Figuur 5.1: De SWOT-analyse.

- **ROI** = return of investment = geeft het rendement op de investering aan.
- **Project proposal** = bevat een beschrijving van de oplossing met een kostenraming en een voorlopige tijdlijn.
  - Wanneer het project is goedgekeurd door het management, wordt er een lastenboek uitgeschreven waarop interne of externe leveranciers kunnen antwoorden met een offerte.
  - Componenten project proposal:
    - Context van het project en de mogelijke impact op de organisatie
    - Doelen van het project (via het SMART-principe)
    - De oplossing
    - Kosten (Fixed price, Time & material)
    - Voorwaarden en assumpties
    - Voorlopige tijdlijn
- **SMART** = Specifiek Meetbaar Acceptabel Realistisch Tijdsgebonden = Als je doelstellingen met deze criteria opstelt, worden ze duidelijk en concreet
- Scope statement = samenvatting van alle op te leveren onderdelen en de soort en het aantal resources die daarvoor nodig zijn. Een eenvoudige afspraak rond het eindresultaat.
  - **Change request** = een vraag tot wijziging die niet was opgenomen in de oorspronkelijke offerte.
  - **Scope creep** = het steeds uitbreiden van de oorspronkelijke scope zonder dat er formele wijzigingen gebeuren aan de oorspronkelijke scope met aanpassing van budget en planning waardoor ook de formele goedkeuring van deze wijzigingen ontbreekt.
- **Risicoanalyse** = analyseert een mogelijke gebeurtenis die, wanneer ze zich voordoet, een negatieve of positieve impact zal hebben op het eindresultaat van het project.

## 5.4. Afhankelijkheden en aandachtspunten

- Grote afhankelijkheid tussen conceptfase en alle andere fases in het project.
- Aandachtspunten:
  - Maak duidelijke en eenduidige afspraken.
  - Maak realistische inschattingen.
  - Vergeet ook de randactiviteiten niet.
  - Bouw flexibiliteit in.

## 6. Planning

### 6.1. Inleiding

- **Projectplanning** = een dynamische verzameling van tools en documenten die de projectmanager gebruikt om het verloop van het project voor te stellen, op te volgen en bij te sturen wanneer nodig.

### 6.2. Beschrijving en doel

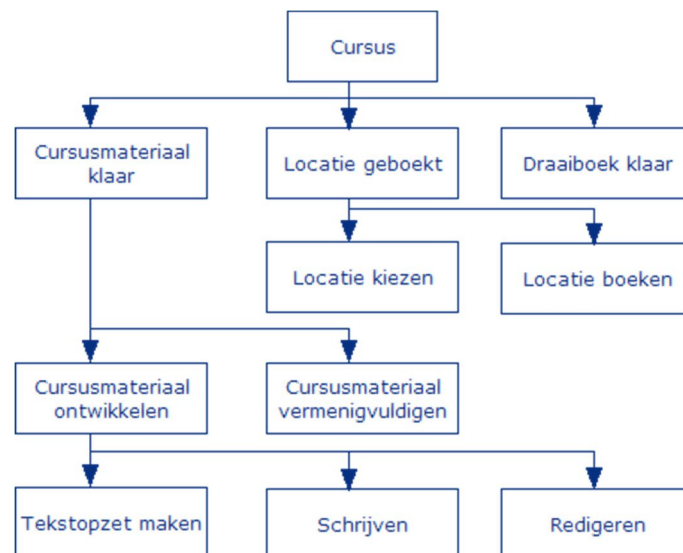
- **Wie**: de invulling van het projectteam en ondersteuning
- **Waar**: de praktische details
- **Wanneer**: de concrete tijdsplanning
- **Wat**: welke activiteiten, tijdschattingen per taak voor doorlooptijd, welke sprints, welke mensen, welke middelen, welke afhankelijkheden zijn er tussen de activiteiten

## 6.3. Rollen en verantwoordelijkheden

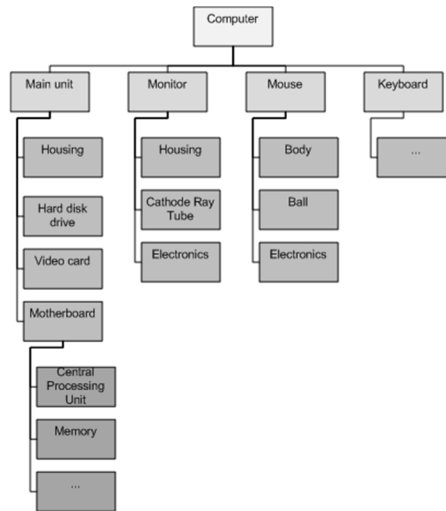
Persoon/projectentiteit	Rol en verantwoordelijkheid
Business analist en technische experts	Inschattingen maken voor verschillende taken.
Projectmanager	Werkt een realistisch plan uit (SMART) en zorgt voor concrete invulling van resources in samenspraak met sponsors.

## 6.4. Deliverables

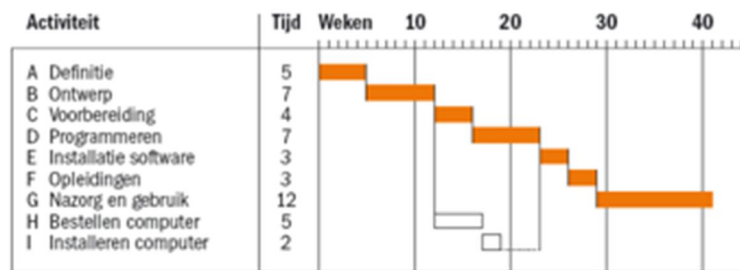
- **Kick-offmeeting** = waarbij het projectplan wordt voorgesteld aan alle teamleden, sponsors en betrokken partijen, er wordt in detail bekeken wie welke taken uitgevoerd moeten worden en wie verantwoordelijk is voor de taken, iedereen leert elkaar kennen, praktische afspraken worden toegelicht zoals het hanteren van naming conventions, templates voor documenten, te volgen procedures, de tijdsregistratie...  
(naming conventions = afspraken die gemaakt worden rond de naamgeving van documentatie zodat de juiste informatie later snel kan teruggevonden worden).
- **Projectplan** = een startdocument voor het omschrijven hoe en wanneer een bepaald tijdelijk doel wordt bereikt.
- Onderdelen van projectplan:
  - **WBS** = work breakdown structure, een hiërarchische voorstelling van de verschillende deliverables die binnen een project uitgevoerd moet worden. De deliverables worden opgebroken in werkpakketten die op hun beurt afzonderlijk ingeschat worden. De projectmanager is hier verantwoordelijk voor. => how to get there.
    - Project objectives (het einddoel) -> work items (tussendoelen) -> work packages (verzameling activiteiten) -> activiteiten



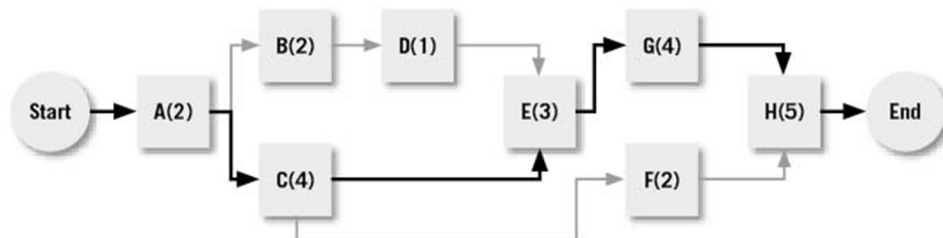
- **PBS** = product breakdown structure, om resultaten van een project te analyseren, te documenteren en mee te delen. => where you want to go.



- **GANTT-chart** = strokenplanning = een mogelijke techniek om een gedetailleerde tijdsplanning op te stellen.



- **Network diagram** = pert-chart = flowchart die aangeeft in welke volgorde activiteiten uitgevoerd dienen te worden.



Figuur 6.2: Kritiek pad.

- Begrippen
  - **Doorlooptijd** = de geschatte tijd die nodig is om de activiteit af te werken.
  - **Slack time** = de vertraging die een bepaalde activiteit mag oplopen zonder dat die een impact heeft op het project.
  - **Kritieke pad** = de reeks van activiteiten die de totale doorlooptijd van het project bepalen (het langst mogelijke pad).

## 6.5. Afhankelijkheden

- Beschikbaarheid van resources: vaak blijkt in praktijk dat de beschikbaarheid afwijkt van de resources waarmee rekening is gehouden.
- Omgevingsfactoren: economische situatie heeft invloed op het budget, nieuwe overheidsregels met impact op de planning...
- Organisatorische wijzigingen: resulteren in wijzigingen in procedures.

## 7. Ontwerp

### 7.1. Inleiding

- **Blueprint** = het document dat een gedetailleerde beschrijving bevat van de oplossing die de leverancier zal opleveren tegen de afgesproken prijs. Het document bevat de definitieve overeenkomst tussen leverancier en klant.

### 7.2. Beschrijving en doel

- Op basis van bestaande documentatie, procesbeschrijvingen en workshops met eindgebruikers en andere betrokkenen, wordt er een zo gedetailleerd mogelijk ontwerp gemaakt. Volgende onderdelen vinden we terug in de ontwerpfase:
  - **Workshops** – BA organiseert dit met eindgebruikers rond een bepaald thema, voorbereiding vooraf, gegroepeerd per thema...
  - **Opmaken van een proof of concept (prototype)** – demo of ontwikkelde light versie, mock-ups.
  - **Opstellen van de blueprint** – na de workshops zal de BA de blueprint opstellen, de definitieve kosten en de verplichte/optionele functionaliteiten worden opgesteld.
  - **Validatie van de blueprint** – wanneer alle betrokken partijen de blueprint gelezen en eventueel aangevuld hebben, komt er een vergadering om de blueprint te valideren. In detail wordt vastgelegd wat er wanneer opgeleverd zal worden en tegen welke kost.
  - **Vorbereiding van de implementatiefase** – finale projectteam vaststellen, wederkerende vergaderingen vastleggen, zorgen voor administratieve en logistieke ondersteuning...

### 7.3. Rollen en verantwoordelijkheden

Persoon/projectentiteit	Rol en verantwoordelijkheid
Business analist	Initiëren en sturen workshops, werken 'blueprint' document uit.
Eindgebruiker	Doen mee aan workshops, leveren input voor blueprint.
Ontwikkelaars	Nemen de consultatie- adviserende rol.
Sponsor	Nemen beslissingen, valideren de blueprint door een handtekening.
Project manager	Waakt over de haalbaarheid, treedt op als tussenpersoon, zorgt voor voorbereidingen van de implementatiefase.

### 7.4. Deliverables

- Volgende deliverables:
  - **Blueprint** – zie inleiding.
  - **Procesbeschrijvingen** via **AS IS – TO BE**
    - AS-IS-proces beschrijft de huidige situatie van een bedrijfsproces en geeft een realistisch beeld van de manier waarop er momenteel gewerkt wordt en welke stappen de gebruikers doorlopen.
    - TO-BE-proces beschrijft de situatie na de afloop van het project en geeft dus weer hoe de verschillende eindgebruikers zullen werken in de toekomst.
  - **Workflow** = de logische volgorde van activiteiten die bepaalde actoren moeten uitvoeren.
  - **Functioneel en technisch ontwerp**
    - Functioneel ontwerp beschrijft de oplossing op niveau van de eindgebruikers (use-cases).
    - Technisch ontwerp beschrijft de technische details van de applicatie zonder code op te nemen in het document (UML).

- **Interfaces**
  - Een connectie tussen twee systemen waarbij er data van het ene systeem doorgestuurd wordt naar het andere systeem.
- **Datamigratie**
  - Gegevens van een oude applicatie naar een nieuwe applicatie verplaatsen (welke gegevens, wie levert de gegevens in welk formaat, wanneer en hoe wordt de data overgezet, hoeveel testruns...?)
- **Systeemarchitectuur**
  - Beschrijft de technische opzet van de applicatie en de verschillende randapplicaties.
- **Testplan** – zie hoofdstuk 9.
- **Veronderstellingen**

## 7.5. Methodologieën

1. Bepaal je scope gebaseerd op de doelen die gesteld werden door de business.
2. Werk de specificaties uit in samenwerking met alle teamleden.
3. Beschrijf je vereisten aan de hand van voorbeelden.
4. Verfijn de specificaties.
5. Automatiseer de testen gebaseerd op de voorbeelden.
6. Valideer de software frequent met de automatische tests.
7. Ontwikkel een documentatiesysteem vanuit de specificaties met voorbeelden om het systeem te blijven ondersteunen.

## 10. Ontwikkelstrategieën

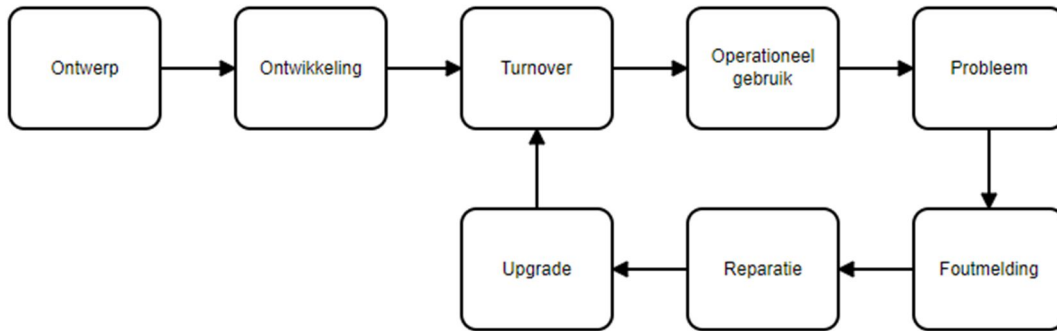
### 10.1. Inleiding, beschrijving en doel

- **Ontwikkelstrategie** = omvat de wijze waarop ontwerp, ontwikkeling en testen opgezet en ingepland zullen worden, afhankelijk van de ontwikkelstrategie verlopen de verschillende fases sequentieel dan wel iteratief.
- Basisstructuur met de verschillende fases (concept, planning, ontwerp, ontwikkeling, test en turnover) blijft geldig voor bijna elk project, maar het inplannen van deze fases kan sterk verschillen, door verschillende soorten ontwikkelstrategieën:
  - Build and fix
  - Waterfall
  - Rapid prototyping
  - Incrementeel
  - Iteratief
  - Agile
  - Continuous delivery

### 10.2. Build and fix

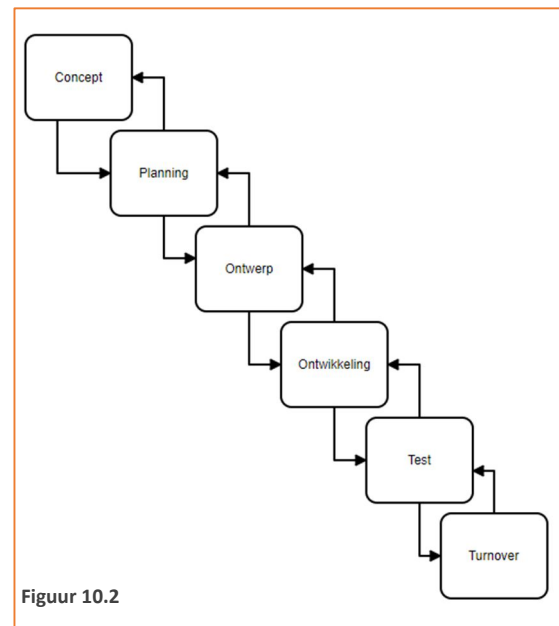
- **Feedback?** Geen feedback tussen verschillende fases, fouten komen pas aan het licht wanneer een klant dit opmerkt en meldt aan de maker van de software. De oplossing van zo'n fout wordt dan gemaakt en naar alle klanten doorgestuurd onder de vorm van een upgrade.
- **Bruikbaar?** Werkt enkel in een markt met heel weinig spelers waar klanten zeer geduldig zijn, tegenwoordig liggen verwachtingen ergens anders.
- **Testen?** Testen worden uitgevoerd door de klant die het product gaat gebruiken.





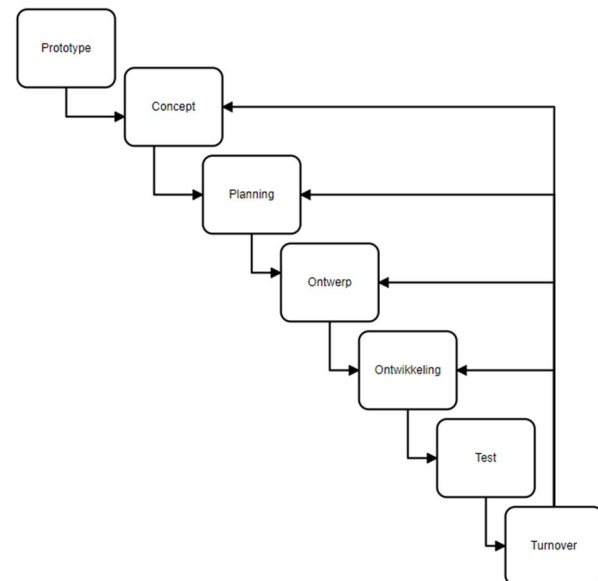
### 10.3. Waterfall, lineair, sequentieel model

- Gebaseerd op de principes van de constructie van grote bouwwerken zoals bruggen.
- **Feedback?** Wel feedback tussen verschillende fases, maar men kan niet meer teruggaan naar een vorige fase. Om overgang van de ene naar de andere fase vlot te laten verlopen, wordt er gebruik gemaakt van zeer uitgebreide documentatie.
- **Bruikbaar?** Bij zeer eenvoudige software-implementaties, geen complexe projecten. De eenvoud maakt het eenvoudiger om zeer langdurige projecten goed te plannen en op te volgen.
- **Testen?** Vaak pas op het einde van het project, kans is groot dat het ontwikkeld product dan niet volledig voldoet aan de vereisten.



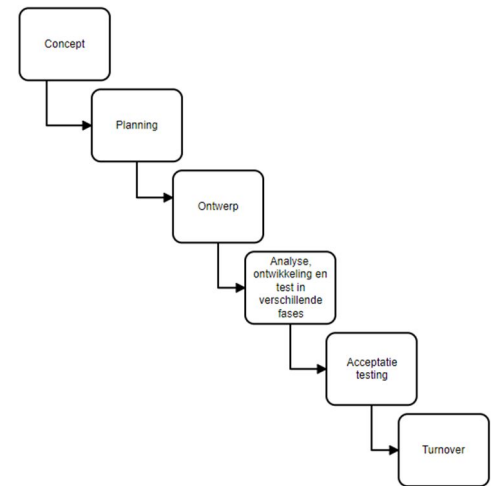
### 10.4. Rapid prototyping

- Eerst een prototype maken alvorens aan de definitieve projectcyclus te beginnen. Gelijkaardig aan waterfall-model, maar voor de aanvang van het uiteindelijke project zal men al een kleine projectcyclus doorlopen om het prototype te bouwen.
- **Feedback?** Op basis van het prototype kan de klant vereisten bepalen en feedback geven.
  - Throw-away: het prototype dient enkel om de klant te tonen hoe de uiteindelijke applicatie er zou kunnen uitzien. Het is niet de bedoeling dat dit prototype deel uitmaakt van de uiteindelijke oplossing.
  - Light implementation: het prototype bevat al een deel van de uiteindelijke oplossing en zal gebruikt worden als basis voor de verdere implementatie
- **Bruikbaar?** Klant krijgt al vroeg in het proces inzicht in wat er opgeleverd zal worden, maar prototypes bouwen kost tijd en geld.
- **Testen?** Het afgewerkt product zal de klant pas zien bij de uiteindelijke testfase.



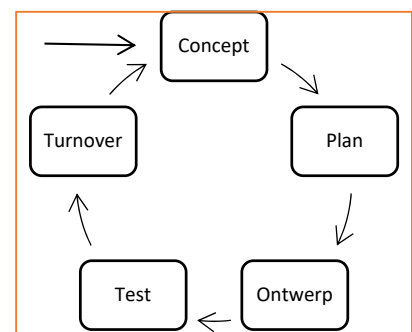
## 10.5. Incrementeel ontwikkelen

- Softwareontwikkeling is beter beheersbaar als het volledige product wordt opgedeeld in kleinere stukjes. Analyse, ontwikkeling en testen wordt per functionaliteit gedaan. Eerste release bevat meestal de basisfunctionaliteiten en de volgende releases bouwen daarop verder.
- **Feedback?** Tussen de verschillende fases binnen een release, maar dit is nog te weinig.
- **Testen?** Na elke release een acceptatietest, zodat volgende releases daar rekening mee kunnen houden.



## 10.6. Iteratief ontwikkelen

- Volledige project wordt opgedeeld in kleinere projecten. Er wordt telkens een nieuw volledig project opgedeeld in kleinere projecten. In de eerste iteratie zal er een basisoplossing opgeleverd worden, in de tweede iteratie worden nieuwe functionaliteiten toegevoegd, verbeterd en verfijnd op basis van ervaringen van de eerste oplossing. Om uiteindelijk tot een eindproduct te komen. Beter voor projectteam, meer focus werken aan minder complexe oplossingen, aangezien het zo opgedeeld is.
- **Feedback?** Eindgebruikers worden van begin tot einde betrokken bij het project. Hierdoor wordt vermeden dat de ontwikkelaars grote delen opnieuw moeten doen
- **Bruikbaar?** Na elke iteratie levert de leverancier een bruikbaar product op. Vertrouwen van klant stijgt en is budgetvriendelijk (na elke iteratie kan er gestopt worden).
- **Testen?** Bij elke iteratie, risico op ontevredenheid en op fouten wordt verkleind omdat er veel sneller getest wordt. Kwaliteit van product gaat omhoog door de vele validatiemomenten.
- Nadeel is dat er vooraf geen afgelijnde afspraken gemaakt kunnen worden over wat er opgeleverd zal worden op het einde van het project. Bij elke iteratie kunnen er vereisten bijkomen of wijzigen, een sluitend budget vaststellen is hierdoor moeilijk.



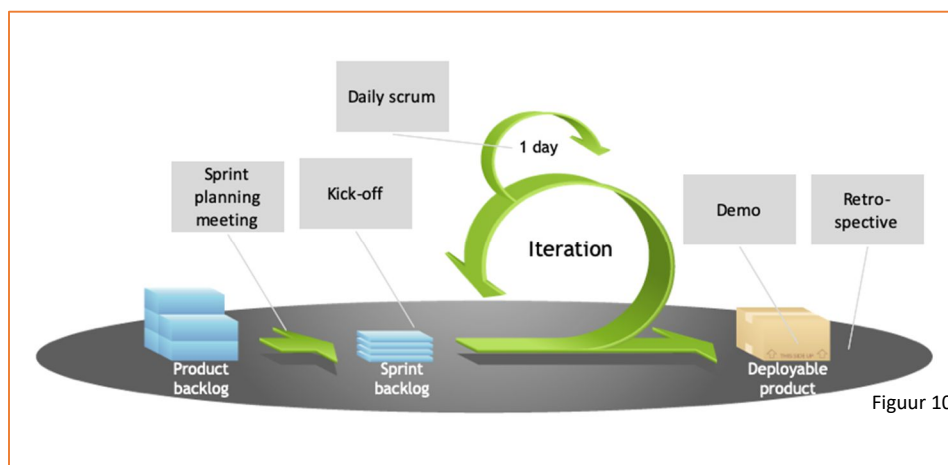
Figuur 10.5

## 10.7. Agile principe – SCRUM

- Agile ontwikkelen heeft vier vuistregels
  1. Personen en interacties boven processen en tools.
  2. Software die werkt boven lijvige documentatie.
  3. Samenwerking met de klant boven onderhandeling over het contract.
  4. Omgaan met verandering boven het volgen van een plan.
- Agile ontwikkelen is een variant op het iteratief model waarbij veel aandacht wordt besteed aan de directe inzetbaarheid van de software na elke iteratie. Iteraties worden bewust kort gehouden zodat de klant snel vooruitgang ziet. Daarnaast heeft agile ook sterke directe communicatie, naar klant toe en binnen het team. Geen concrete planning, continu aanpassen aan veranderende omgeving en vereisten van de klant.

## Scrum

- SCRUM-methode is de meest populaire ontwikkelstrategie die werken via het agile principe.
- **Doel** van SCRUM is om zo veel mogelijk toegevoegde waarde te creëren op de kortst mogelijke termijn.
- Elke **sprint**, iteratie, duurt twee tot maximaal vier weken. Na deze periode levert het team een stukje werkende software af en kan de klant beslissen om al dan niet verder te gaan in een volgende sprint.
- Start met vastleggen van een **product backlog**, een verzameling van alle functionaliteiten met inschatting (story points), maar kan nog aangepast worden.
- Aan begin van elke sprint wordt een **sprint backlog** opgesteld, een verzameling van alle functionaliteiten die ontwikkeld moeten worden in die sprint.
- Tijdens een sprint wordt er een **daily standup meeting** georganiseerd, elk teamlid vertelt kort wat hij heeft gedaan, wat hij zal doen en waar hij problemen mee ondervindt.
- Na de sprint houdt men de **sprint review**, waarin wordt getoond wat er tijdens de sprint gerealiseerd werd.
- Na de sprint wordt er ook een **sprint retrospective** gehouden, waarin samen met het team en de klant bekeken wordt wat er wel en niet werkt.
- De **SCRUM-master** is degene die waakt over de regels en zorgt dat het team in optimale omstandigheden kan samenwerken.
- Een **SCRUM-team** bestaat maximaal uit negen personen die voltijds aan het project worden toegewezen.
- **Fases van SCRUM:**
  - Analyse – opgedeeld in twee stukken; functionele analyse, dus opmaken van blueprint en overeenkomst met klant, en technische analyse.
  - Implementatie – het bouwen van software zal worden opgesplitst in groepjes functionaliteiten. In elke sprint zullen er extra functionaliteiten worden toegevoegd aan een werkende toepassing.
  - Testen – ontwikkelen en testen gebeurt continu, testen gebeuren vaak ook automatisch, testfase is dus niet echt apart maar samen in de implementatie.
  - Turnover – de werkende software zal na validatie in productie gezet worden en opgenomen worden door het operations team van de klant. Hypercare is kort.



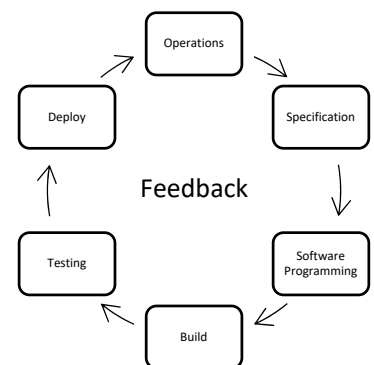
Figuur 10.6

## SAFe SCRUM

- Scale Agile Framework SCRUM is gebaseerd op de principes van lean management, dat er op gericht is om alle overbodige zaken uit een proces te verwijderen.
- Ontstond vanuit de uitdaging van een program manager om overzicht te bewaren tussen honderden software developers die elk aan hun eigen toepassing op een zeer flexibele manier werken.
- Zorgt ervoor dat SCRUM ook naar grote organisaties geschaald kan worden.
- Gebaseerd op negen principes
  1. Economisch inzicht
  2. Pas systeemdenken toe
  3. Ga uit van verandering en houd opties open
  4. Bouw incrementeel met snelle geïntegreerde cycli
  5. Baseer milestones op een objectieve evaluatie van het werkend systeem
  6. Zorg voor een lean proces
  7. Zorg voor een ritme en doe aan planning over domeinen heen
  8. Zorg voor intrinsieke motivatie bij het team
  9. Zorg voor gedecentraliseerde beslissingen

## 10.8. Continuous delivery

- Klanten verwachten steeds snellere updates van softwareproducten, IT heeft steeds een groter aandeel in de business waardoor er steeds vaker naar IT gekeken wordt om te besparen, fouten zijn steeds moeilijker te verantwoorden. Hierdoor werken steeds meer bedrijven volgens de methode van continuous delivery.
- Het hele proces van specificatie tot operations verloopt in zeer korte cycli, zodat werkende software meerdere keren per maand, per week of zelfs per dag in productie kan worden opgeleverd.
- Belangrijk om een multidisciplinair team samen te stellen die samen eindverantwoordelijkheid draagt. Het betrekken van alle rollen in één team.
- Kosten zullen verlagen, kwaliteit verhoogt, omdat fouten worden opgespoord nog voor de software in productie wordt gezet. Productiviteit van softwareontwikkeling gaat omhoog. Door snellere doorlooptijd kan er veel sneller worden ingespeeld op de vragen van de business.



## 11. Turnover

### 11.1. Inleiding

- **Turnover** = de overdracht van het afgewerkte product naar de klant. De software, documentatie, trainen en kennisoverdracht zodat de klant met de applicatie kan blijven werken nadat de leverancier vertrokken is.

### 11.2. Beschrijving en doel

- Documentatie
  - Technisch = implementatie van de applicatie in detail, met beschrijvingen van de geïmplementeerde software, parameters en klantspecifieke code. Documentatie over de code zelf.
  - Functioneel = gebruikershandleiding.
- Training – sleutelgebruikers opleiden zodat die op hun beurt de rest kan trainen.
- Support – ondersteuningsnood is het hoogst in de eerste zes tot acht weken.
  - Service Level Agreement (SLA) = een contract tussen klant en leverancier dat in detail beschrijft welke service de klant van de leverancier mag verwachten. Beschrijft onder andere de beschikbaarheid van de helpdesk, de reactiesnelheden en de boete wanneer de voorwaarden niet bereikt worden.
  - Support pack = verzameling van verbeteringen van bestaande software, dat meestal een oplossing biedt voor gekende problemen.
  - Upgrade = nieuwe versie van een bestaande software die, naast verbeteringen, vaak ook nieuwe functionaliteiten bevat.
- **IT service management** – houdt zich bezig met implementatie en kwaliteitsbewaking van verschillende IT-services die een organisatie nodig heeft ter ondersteuning van de business.
- Doel = er voor zorgen dat de ontwikkelde applicatie optimaal gebruikt kan worden door de eindgebruikers en dat de kwaliteit van de toepassing gegarandeerd blijft in de toekomst.

### 11.3. Rollen en verantwoordelijkheden

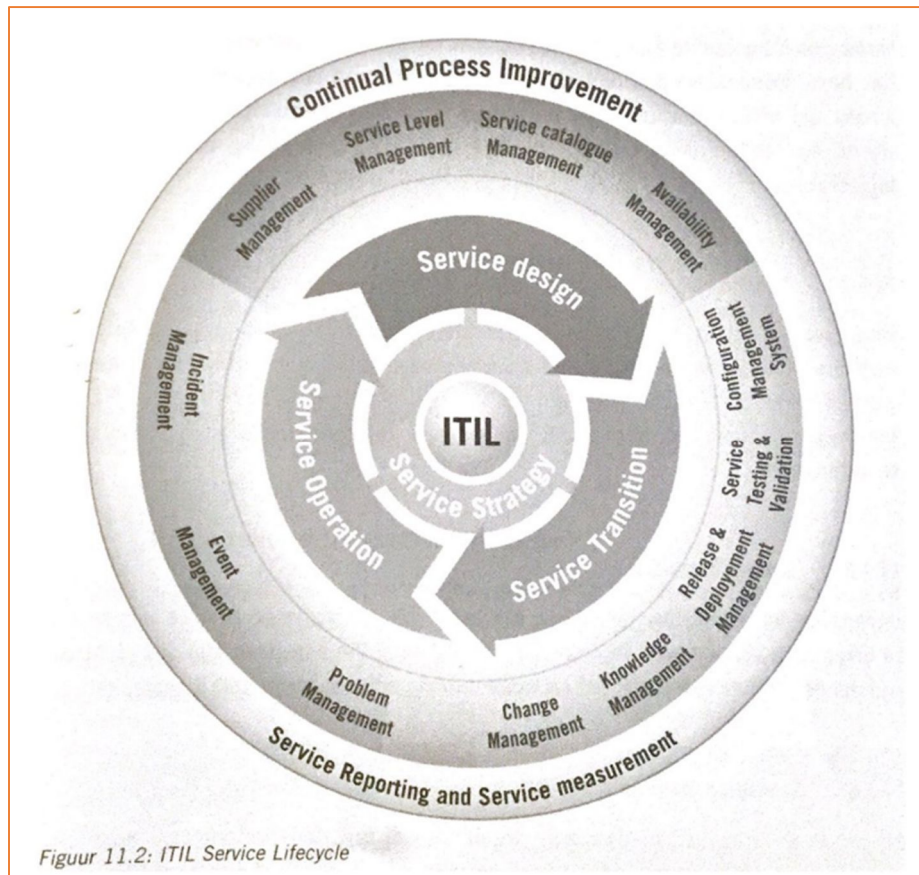
Persoon/projectentiteit	Rol en verantwoordelijkheid
Business analyst	Antwoorden functionele vragen van eindgebruikers, verantwoordelijk voor user manuals.
Ontwikkelaars	Verantwoordelijk voor de hypercare en voor kennisoverdracht naar de IT-afdeling van de klant.
Trainer	Zorgen voor training van eindgebruikers.
Project manager	Verantwoordelijk voor opvolging van de verschillende trainingsactiviteiten en documentatie. Stelt Cut-over plan op.
Helpdesk	Verzorgt ondersteuning van eindgebruikers.

### 11.4. Deliverables

- Volgende deliverables:
  - Documentatie
  - Procedures – voor het melden van problemen, voor het aanvragen van change requests, voor de aanvraag van gebruikers en rollen...
  - Cutover-plan – als draaiboek voor de go-live.

## 11.5. Methodologieën – ITIL

- **Information Technology Infrastructure Library (ITIL)** = een van de meest gebruikte IT service management frameworks, benadert IT service op een klantgerichte manier en dekt op een end-to-end manier de volledige IT service lifecycle af.
- Overzicht van de vijf fases:
  1. Service strategy
    - Financial Management
    - Service Portfolio Management
    - Demand Management
    - Strategy Management for IT services
    - Business Relationship Management
  2. Service design
    - Service Catalogue Management
    - Service Level Management
    - Capacity Management
    - Availability Management
    - IT Service Continuity Management
    - Information Security Management
    - Supplier Management
    - Design Coordination
  3. Service transition
    - Transition Planning and Support
    - Change Management
    - Service Asset and Configuration Management
    - Release and Deployment Management
    - Service Validation and Testing
    - Evaluation
    - Knowledge Management
  4. Service operation
    - Event Management
    - Incident Management
    - Request Fulfilment
    - Problem Management
    - Access Management
  5. Continual Service Improvement
    - Seven step Improvement Process
- Meer uitleg:
- **Service strategy** = strategie voor de hele service lifecycle wordt bepaald. Een uitgebreide analyse van de klanten wordt gemaakt en er wordt gekeken naar hoe ze het beste kunnen worden ondersteund.
- **Service design** = de volledige service-organisatie wordt uitgetekend: welke services worden aangeboden, hoe zullen de processen er gaan uitzien, de SLA's worden opgesteld, er wordt bepaald wat er nodig is van hardware, software en andere resources om de nodige capaciteit, beschikbaarheid en veiligheid te garanderen.
- **Service transition** = omvat alle processen die nodig zijn om nieuwe software op te leveren en op te nemen in de service-organisatie.



## 11.6. SLA, SLO, SLI

	SLI	SLO	SLA
Afkorting	Service Level Indicators	Service Level Objectives	Service Level Agreements
Definitie	Een kwantificeerbare maatstaf van betrouwbaarheid	Een nagestreefd betrouwbaarheidsniveau van een service	Een wettelijk contract dat, indien het wordt geschonden, financiële sancties met zich meebrengt
Voorbeeld	The ratio of valid request loaded in < 400 ms	99% of requests served in < 400 ms over a 28-day rolling window	Monthly average round-trip transmission time of 500 ms or less for 98% of requests
Wie	Site reliability engineers	PO, SRE, Ops Team, developers and customers	Business Development and legal team alongside IT and DevOps teams

# Deel 3: projecten leiden

Standpunt project manager // initiëren, plannen, uitvoeren, opvolgen, afsluiten

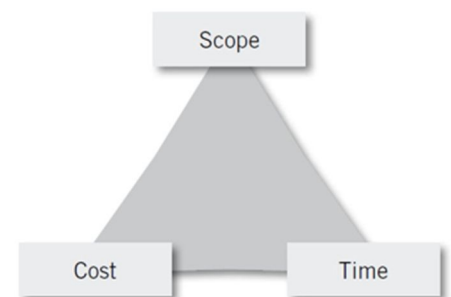
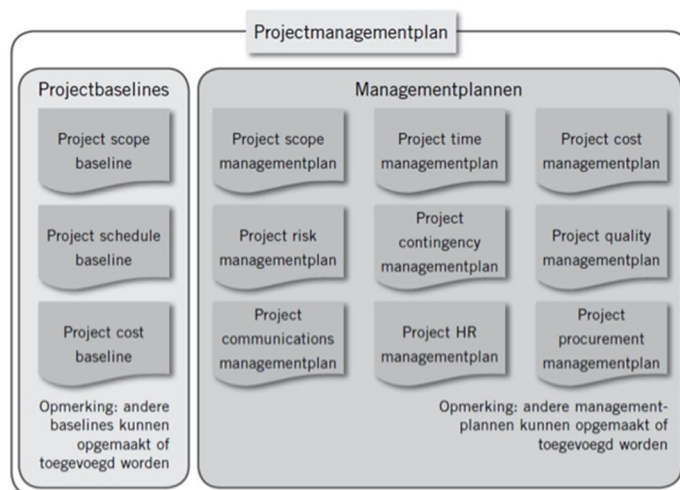
## 13. Project plannen

### 13.1. Inleiding en leerdoel

- **Planning** = bepalen hoe we het project sturen en hoe we trachten het op de rails te houden. Door project scope af te bakenen, het verfijnen van de projectobjectieven en het vastleggen van de te volgen koers door goede afspraken te maken.
- **Projectmanagementplan** = eindresultaat van het plannen van een project, de handleiding die het team informeert over wat er te gebeuren staat, hoe dat zal gebeuren en, samenvattend, wat de projectobjectieven zijn. Dit stappenplan is het resultaat van projectplanningsactiviteiten en vormt de leidraad doorheen het volledige project.

### 13.2. Het belang van een degelijke planning

- Projectmanagementplan is opgebouwd uit twee belangrijke blokken: projectbaselines en managementplannen.



Figuur 13.1: Overzicht van de bestanddelen van een projectmanagementplan.

- Elk domein heeft zijn eigen specificaties, daarom wordt een PMP voor een groot project nooit door slechts één persoon uitgewerkt.
- **Primary driver** = de beperking die vanuit het oogpunt van de klant het belangrijkste is om door de projectmanager bewaakt te worden.

### 13.3. Projectbaseline 1: scope

- Verschil product scope en project scope:
  - Product scope = beschrijft wat de karakteristieken en functionaliteiten zijn van het product
  - Project scope = beschrijft welke activiteiten uitgevoerd moeten worden om dit product te realiseren.
- Scope baseline bestaat uit drie elementen:
  - Project scope statement = geeft aanzet tot het omlijnen van de project scope.
  - **Work Breakdown Structure (WBS)** = Een grafische weergave die de project scope opsplijst in deelcomponenten (de deliverables van het project).



- WBS dictionary = vult de WBS aan, eerste kolom bevat alle deliverables, andere kolommen bevatten nuttige informatie.

### 13.4. Projectbaseline 2: time/schedule

- Geeft aan wanneer in de tijd bepaalde activiteiten starten en eindigen, wat de minimale doorlooptijd van het project is.
- Schedule baseline in vijf stappen:
  1. Definieer de activiteiten
  2. Bepaal de volgorde van de activiteiten
  3. Bepaal de resourcenoden
  4. Schat de doorlooptijd van de activiteiten
  5. Ontwikkel de projecttijdlijn
- Critical Path Method (CPM) = bepalen van de minimale doorlooptijd van het project.
  - ES = Earliest start
  - EF = Earliest Finish
  - LS = Latest Start
  - LF = Latest Finish
  - Planningsoefening maken, voorbeeld pg 194

### 13.5. Projectbaseline 3: cost

- Uitwerken van totale budget.
- De prijs waarmee we moeten rekenen om kwaliteit op te leveren valt uit elkaar in twee componenten:
  - Cost of conformance – de investering om conform de kwaliteitseisen te produceren. Die kosten maken we om te vermijden dat we een product van slechte kwaliteit produceren.
  - Cost of non-conformance – de kost om kwaliteitsdefecten op te vangen. De kosten maken we om deliverables van slechte kwaliteit te corrigeren.

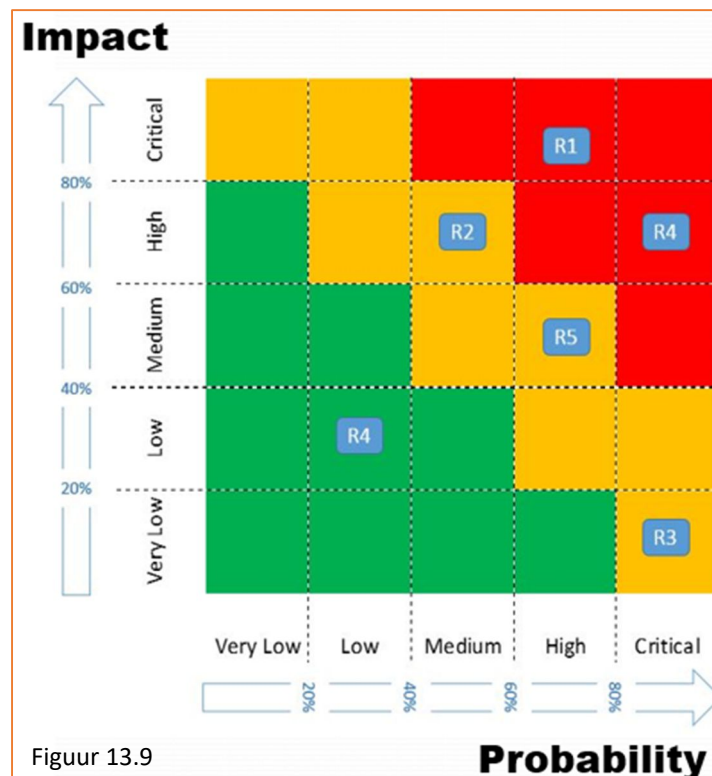
### 13.6. GOKIT

- De GOKIT-methode is een methode voor het beheersen van benoemde aspecten in het projectmanagement
  - Geld = het totaal aan financiële middelen nodig om het project te financieren.
  - Organisatie = het totaal van personen, bedrijven, partijen en andere entiteiten die activiteiten kunnen ondernemen binnen in maar ook van buiten af voor het project.
  - Kwaliteit = het totaal aan kwaliteitskenmerken en kwaliteitswaarborgen, nodig om het prospect succesvol te laten verlopen en op te leveren.
  - Informatie = alle beschikbare informatie die voorafgaand aan het project in de ontwikkelingsfase, tijdens de uitvoeringsfase en achteraf in de nazorgfase aanwezig was, er van buiten af bij komt en binnen het project wordt geproduceerd.
  - Tijd = het tijdsaspect van het project op zich en alle aspecten van tijd rond om het project.

### 13.7. Risicobeheer

- **Risico** = onzekerheden die een invloed hebben op de projectobjectieven
  - Negatieve risico's = bedreigingen
  - Positieve risico's = opportuniteiten
- Projectrisicobeheer = managen van onzekerheden die de projectobjectieven mogelijk beïnvloeden.

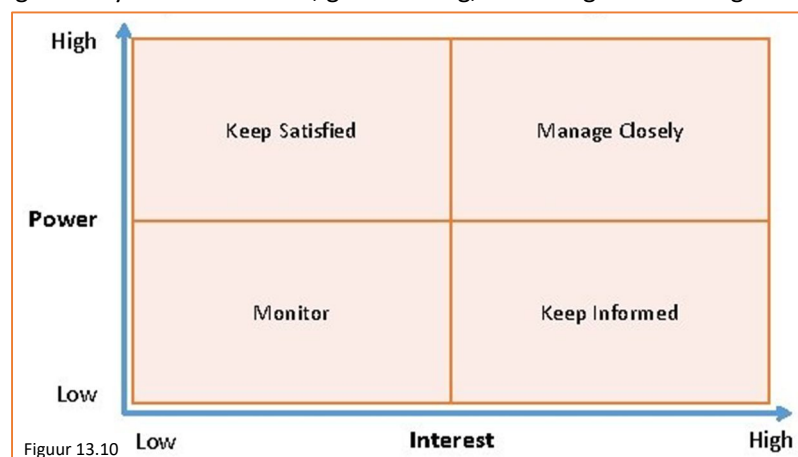
- **Risicobeheer plannen**
  - Identificatie – met welke technieken risico's identificeren, welke tijdlijn er zal gevolgd worden, hoe risico's bijhouden in risicoregister.
  - Evaluatie – regels vastleggen om prioriteiten te stellen, vastleggen hoe probabiliteit en impact zullen ingeschat worden, probabiliteit-impact-matrix.
  - Organisatie – rollen en verantwoordelijkheden vastleggen, wie risico-manager wordt.
  - Budgettering – hoe risicobeheer geïntegreerd wordt in budgetbepaling.
  - Rapportering – welke elementen uit het risk register we zullen rapporteren en met welke frequentie.
- **Risico's identificeren**
  - Organiseren van risk identification sessions – brainstorm met grotendeel van team.
  - Assumption & constraint- analyse – veronderstellingen en beperkingen evalueren.
  - SWOT-analyse – strengths, weaknesses, opportunitites en threats in kaart brengen.  
=> Met als resultaat een **risk register** = een lijst van risico's, omschrijving van de onzekere gebeurtenis en niet de oorzaak of het gevolg.
- **Prioriteiten stellen**
  - Bepaald door drie parameters:
    - Wat is de kans dat het risico zich voordoet?
    - Wat is de impact van het mogelijke risico op de projectobjectieven?
    - Wanneer kan het risico zich voordoen?
  - Uitzetten in Probability/impact grid, risico's opsplitsen in drie categorieën:
    - Rode zone – vraagt onmiddellijke aandacht, toewijzen aan een individu
    - Oranje zone – belangrijk genoeg om op te volgen, toewijzen aan PM team
    - Groene zone – niet cruciaal, niet actief beheren



- **Omgaan met risico's**
  - Bedreiging
    - Avoid - Volledig vermijden, niets doen of alternatieve benadering kiezen.
    - Transfer – Volledige impact doorgeven aan derde partij.
    - Mitigate – Impact inperken, terugdringen tot bepaald niveau.
    - Accept – Risico aanvaarden en mee leven?
  - Opportuniteit
    - Exploit – Kans dat het zich voordoet maximaliseren.
    - Share – Kans verhogen, maar bereid om stuk te delen, ownership gedeeltelijk.
    - Enhance – Verhogen de kans en/of impact van het risico.
    - Accept – Bestaan bevestigen, maar geen actieve stappen zetten om te verhogen.
- Valkuilen risicobeheer
  - Verkeerde positionering (als toezichtsfunctie in plaats van beheersfunctie positioneren)
  - Te weinig communiceren
  - Bij één persoon beleggen
  - Standaard checklists
  - Nadruk op harde aspecten
  - Bagatelliseren/risico's onderschatten
  - Het-ligt-nooit-aan-mij-syndroom
  - Schuldigen aanwijzen ipv op zoek gaan naar oorzaken en oplossingen.
  - Te weinig tijd
  - Verslapping van aandacht

## 13.8. Team en informatie

- **Stakeholders** = betrokkenen die beïnvloed kunnen of zullen worden door het project, het project kunnen of zullen beïnvloeden.
- **Categorieën:**
  - Monitor stakeholder – weinig belang, passief op de radar – Low interest + Low power
  - Keep informed – lage invloed, hoge interesse, informeren – High interest + Low power
  - Keep satisfied – veel invloed, lage interesse, proactief samenzitten – Low int + High power
  - Manage closely – sterke invloed, groot belang, actief begeleiden – High int + High power



- Project governance = de structuur en relaties tussen teamleden en stakeholders die noodzakelijk zijn voor een vlotte informatiedoorstroming.
- Projectcommunicatieplan = communicatie in kaart brengen, communicatie plannen; welke frequentie, welke medium, push- of pull-communicatie, templates...

## 13.9. Overige planningsdomeinen

- Kwaliteit opleveren
  - Kwaliteitstandaarden definiëren, nodige maatregelen rond kwaliteit plannen en uitvoeren, resultaat monitoren, plan herbekijken.
  - Kwaliteit van het product en kwaliteit binnen het proces.
  - Welke meetpunten/quality metrics opvolgen?
- Aankoopprocessen

# Extra's

---

## ProjectManagementPlan

- Achtergronden
- Doelstelling, projectopdracht en business case
- Projectgrenzen en randvoorwaarden
- Projectorganisatie en -communicatie
- Projectactiviteiten
- Tussen- en eindproducten
- Planning
- Kwaliteit
- Risico's

## Programming practices (Cegeka eBook)

- **Extreme programming (XP)** = een methodologie voor softwareontwikkeling die tot doel heeft de kwaliteit van de software te verbeteren en beter in te spelen op veranderende eisen van de klant.
  1. **Pair programming** = programmeren in paar waarbij iemand de 'driver' is en de andere de 'navigator'. Dit resulteert in betere kwaliteit van code aangezien de code constant beoordeeld wordt en de kennis gedeeld wordt.
  2. **Collective code ownership** = niemand bezit de code, elke developer kan elk stukje code aanpassen. Consistentie en leesbare code helpt.
  3. **Test-driven development** = testen schrijven die falen om daarna code te schrijven zodat de testen slagen. Focus op één klein probleem en focus op kwaliteit.
  4. **Behaviour-driven development** = focus op verwachtingen van de klant over de code en niet de focus op de code zelf.
  5. **Domain Driven Design** = developer moet de taal van de klant spreken.
  6. **Incremental Design** = houdt design eenvoudig, gelimiteerde designtijd.
  7. **Refactoring** = stapsgewijs verbeteren en onderhouden van het ontwerp.
  8. **Customer collaboration** = dialoog aangaan met klant om requirements vast te stellen, misverstanden onmiddellijk uit de weg ruimen.
  9. **Constant feedback** = klant en team werken nauw samen, prioriteit stellen met de feedback.
  10. **Minimise time between stages** = tijd tussen development, testing en bug fixing minimaliseren.
  11. **Continuous build and integration** = code gebouwd en automatisch getest, build scripts maken een geheel van nieuwe en bestaande code, developers 'check-in' hun code meermaals per dag.
  12. **Automation** = fix kapotte build onmiddellijk.
  13. **Quality constraints** = niet-functionele eisen worden, indien mogelijk, ook beschreven in de vorm van user stories.
  14. **Managing trade-offs** = prioriteiten stellen bij Duivels driehoek.
  15. **Continuous improvement** = via retrospectives en trainingen mensen voortdurend verbeteren.

## Scrum guide

- **Scrum team** = bestaat uit één Scrum Master, één Product Owner en Developers. Het team is multidisciplinair, klein genoeg om wendbaar te blijven en groot genoeg om werk van betekenis te voltooien binnen een sprint.
  - **Developer** = diegene die iedere Sprint gecommitteerd zijn aan het maken van elk aspect van een bruikbaar Increment. Is verantwoordelijk voor de sprint Backlog, garanderen van

- kwaliteit door vast te houden aan definition of done en hun plan aan te passen richting Sprint doel.
- **Product owner** = is verantwoordelijk voor het maximaliseren van de waarde van het product. Hij ontwikkelt Product Backlog items en zorgt voor transparantie.
  - **Scrum master** = verantwoordelijk voor het opzetten van Scrum, zorgt voor effectiviteit en leiderschap, hij verwijdert hindernissen, plant en organiseert Scrum implementaties.
- **Gebeurtenissen**
    - **Sprint** = hartslag van Scrum met vaste lengte, Al het noodzakelijke werk wat nodig is om het Product Doel te bereiken vindt plaats binnen Sprints.
    - **Sprint planning** = start de Sprint door het uit te voeren werk voor de Sprint uit te stippelen. Behandelt volgende onderwerpen:
      - Waarom is de sprint waardevol?
      - Wat kan er tijdens deze sprint worden afgerond?
      - Hoe zal het gekozen werk gedaan worden?
    - **Daily scrum** = is om voortgang richting het Sprint doel te inspecteren en de Sprint Backlog als nodig aan te passen, waarbij het aankomend gepland werk wordt bijgesteld. Duurt zo'n 15 minuten.
    - **Sprint review** = om de uitkomst van de Sprint te inspecteren en toekomstige aanpassingen te bepalen.
    - **Sprint retrospective** = om manieren te bedenken om kwaliteit en effectiviteit te verhogen en in te plannen.
  - **Artefacten**
    - **Product backlog** = is een levende, geordende lijst van wat nodig is om het product te verbeteren. Het is de enige bron van het werk dat door het Scrum Team gedaan wordt.
      - **Product doel** = Doel beschrijft een toekomstige staat van het product, dat kan dienen als een doelwit voor het Scrum Team om tegen te plannen.
    - **Sprint backlog** = samengesteld uit het Sprint Doel (waarom), de set van Product Backlog items geselecteerd voor de Sprint (wat) en een uitvoerbaar plan voor het opleveren van het Increment (hoe).
      - **Sprint doel** = de enige doelstelling voor de Sprint.
    - **Increment** = een concrete stap in de richting van het Product Doel.
      - **Definition of done** = een lijst van eisen waaraan moet worden voldaan voordat een user story als compleet kan worden beschouwd.

## Extra Agile termen

- **Minimum Viable Product** = de meest simpele versie van je product waarmee je daadwerkelijk waarde kunt bieden aan je gebruiker, deze tevreden kunt stellen en aan je kunt binden.
- **Kanban** = een concept waarbij het wordt gebruikt als een planningssysteem dat je vertelt wat je moet produceren, wanneer je het moet produceren en hoeveel je moet produceren. (TO DO – DOING – DONE)
- **Work in progress (WIP) limits** = het beperken van de maximum aantal werkitems in de verschillende stadia van de workflow .
- **Spike** = een soort user-story dat gebruikt wordt om technische kennis die nodig is voor een bepaalde requirement onder de knie te hebben