

## H1 – Projecten identificeren

**Project:** Opgestart om duidelijk *doel* te realiseren -> (eind)product

= Som alle activiteiten op *product (beschreven in behoefte/vraag organisatie)* uit te werken en op te leveren. Brengt hiervoor juiste *kennis/expertise* samen zonder rekening te houden met functionele of organisatorische grenzen. -> Projectmanager (PM)

= In een project werkt een aantal *mensen van verschillende disciplines tijdelijk* samen met een beperkt *budget* om een *vooraf bepaald doel* te bereiken.

-> *Lifecycle*: geheel projectfasen

**Eign:** **UNIEK** – Voor de 1<sup>e</sup> keer; ingrijpend; ten ondersteuning grote verandering

Specifiek & haalbaar **eindresultaat (=product)**

**Tijdelijk** (meestal start -en einddatum)

+ moet worden opgestart; heeft opdrachtgever; meestal multidisciplinair, eigen (vast) budget, eigen projectorganisatie

=> *operations/dagelijks werk*

*Uitzondering: roll-out project / pilootproject met templateapplicatie* die later in andere afdelingen wordt uitgerold (niet uniek, wel project)

**(Eind)product:** kenmerken & functionaliteiten waaraan het resultaat van het project moet voldoen; reden om project op te starten

-> *Lifecycle*: 'idea-to-market', projectcyclus + oplevering + verbetering/ondersteuning

**Resource:** een element nodig om bepaalde activiteit te voltooien, types: materieel, menselijk, financieel, ... (MMM = Material, Mensen (met expertise), Money)

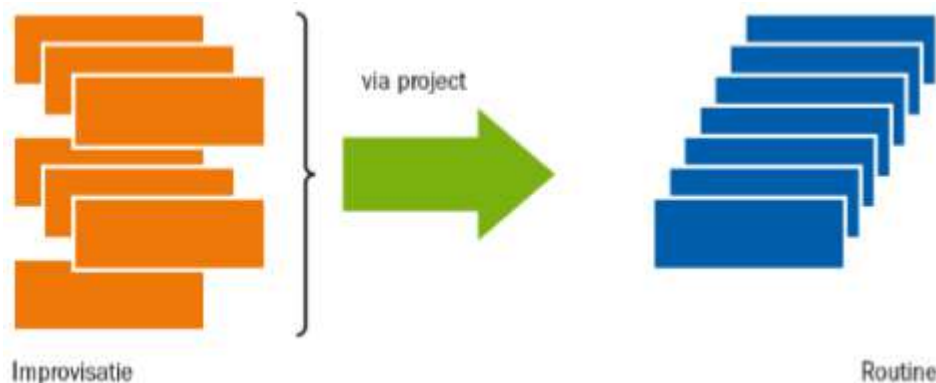
**Change management** = Goed begeleiden veranderingstraject

**Operations** = *Dagdagelijkse werk* bedrijf/afdeling; enkel gericht op realiseren bedrijfsdoelstellingen; operationele teams -> functionele/service delivery manager

Project => Routine (Projectmatig werken)

Door bij een **project** gebruik te maken van een plan van aanpak breng je een soort routine in **projectmatig werken**

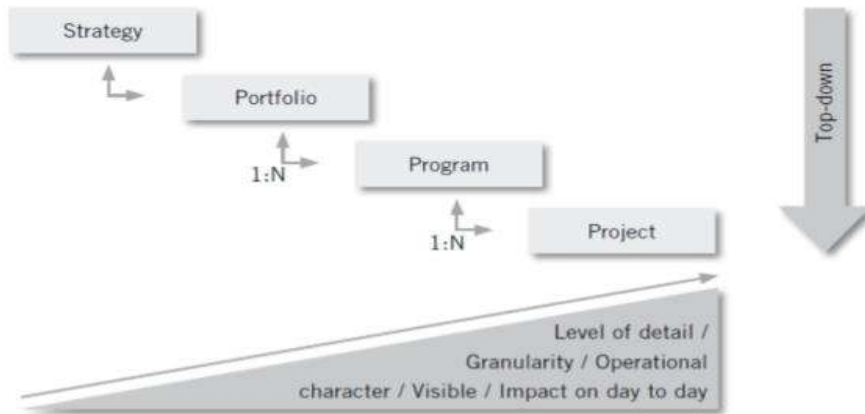
- Voorbereidingstijd vragen
- Overleg betrokkenen
- Grof -> Fijn; Top-Down; grote lijnen eerst, dan details
- Eerst denken, dan doen; project doordenken van voor naar achter en omgekeerd



Projecten herkennen

Oorsprong/reden project?

- **Externe/Mandatory** trigger (bedrijf weinig keuze)
  - Nieuwe wet; *Vb.: nieuwe btw-wetgeving; nieuwe privacy wet, ...*
  - Contractuele bepaling; *Vb.: grootbank stelt nieuwe leverancier aan met eis alle financiële communicatie digitaliseren (meetstanden, ...)*
- **Interne** trigger (strategische beslissing binnen bedrijf); 1 strategie -> N projecten  
 Hoog niveau: strategie -> portfolio (portfoliomanager) (1-1 link)  
 -> Stap voor stap opsplitsing in kleinere, beheersbare initiatieven (program) (1-N)  
 -> Elk program meerdere projecten mogelijk (1-N)

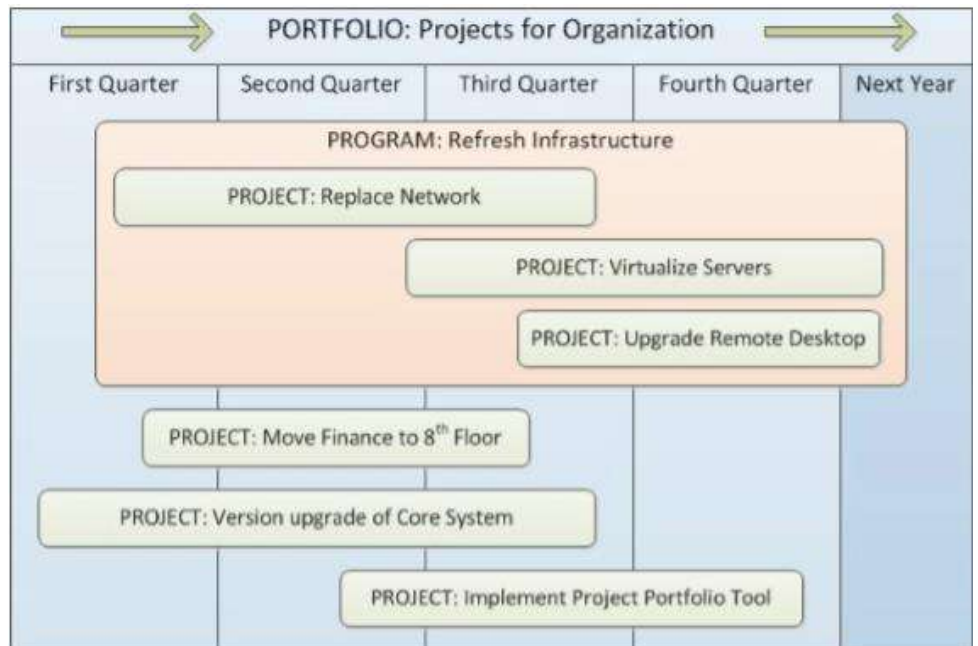


Figuur 1.1: De link tussen strategie en project (project waterfall-model).

### Programma vs. Projecten

**Programma** = groep individuele projecten (1-N) die we op gezamenlijke en gestructureerde manier leiden => programma management:

- Bewaren link strategie – projecten (**consistentie & prioriteiten** bewaken)
- *Shared resources* opvangen - > prioriteiten & **staffing critical resources** bewaken
- **Afhankelijkheden** activiteiten van individuele projecten identificeren en bewaken
- **Methodologie** binnen projecten bewaken (uniforme rapportering, overlegstructuur, ...)



## H2 – Projecten aanpakken

### Projectbouwsteen – Deliverable

**Scope** – Inhoud van het project in detail; bepaalt wat er door het project wordt opgeleverd; inhoudelijke *grens*

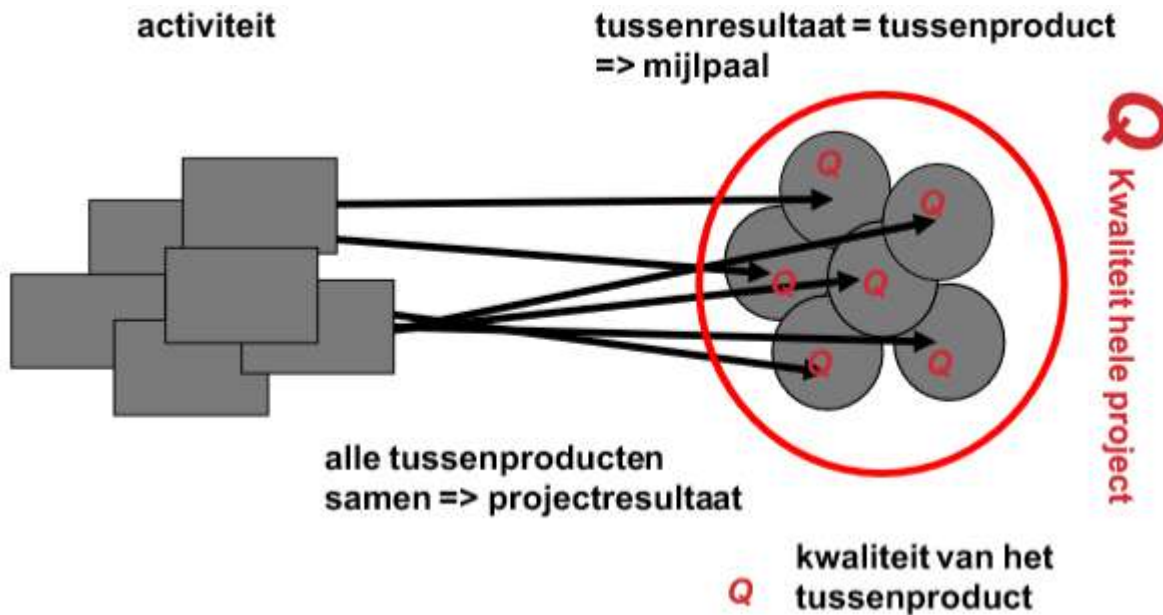
**Deliverable** – Finaal of tussentijds product, opgeleverd als *resultaat van* een projectactiviteit; *noodzakelijk* om activiteit/fase/project af te sluiten

- Niet noodzakelijk *tastbaar*

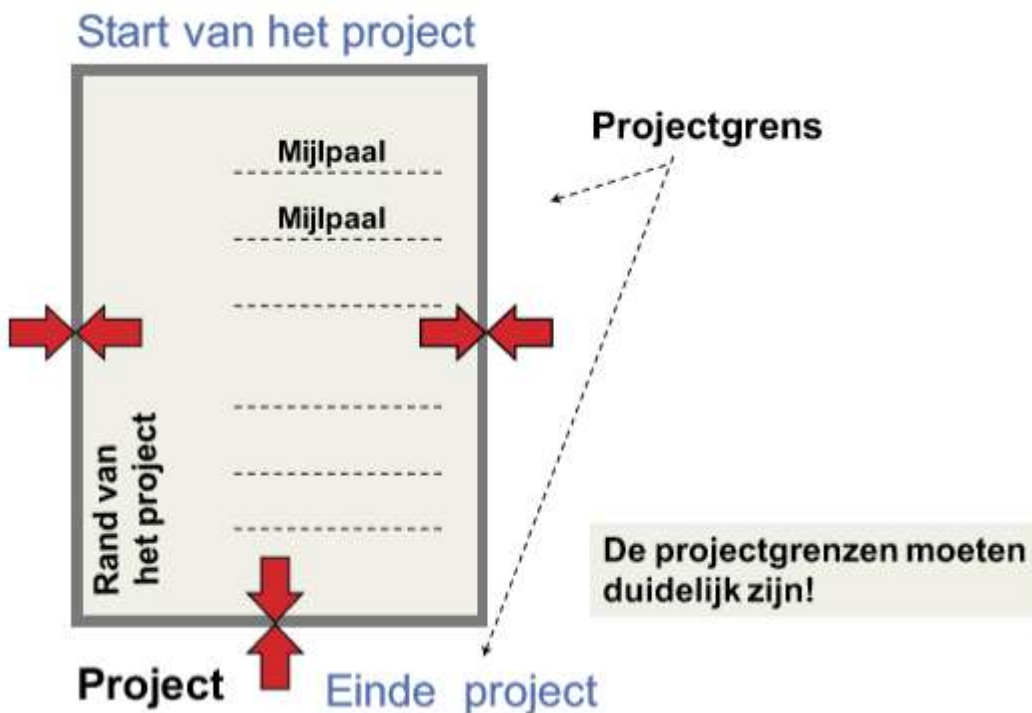
- PM gebruikt deliverables om doorlooptijd & kost te schatten -> projectmeetings, overleg partijen, resultaat van reviewactiviteiten, uitsturen communicatie, ... -> ook deliverable
- Alle deze activiteiten & hun output -> dragen bij tot eindproduct -> deliverables
- Deliverables linken met specifieke *vereisten*, met bepaald *stakeholder*, met bepaalde *kwaliteitsstandaard*, ...
- 1<sup>e</sup> keer beschreven in *project scope statement* -> belangrijke rol in grenzen project
- Laten uitwerken in: werkpakketten & activiteiten
- Hebben bepaalde doorlooptijd ->  $\neq$  *project milestone* (= *checkpoint* in tijdlijn)

**Milestone** – momentopname op projecttijdlijn dat wordt aangegrepen om status van oplevering te monitoren; veelal niet gelinkt aan individuele taak of deliverable -> Geen doorlooptijd (= *project phase gates*)

Activiteit -> Project



Projectgrenzen



*Projectgrenzen Lengte*

Hoe ver gaan we? -> *lengte* van project

Vb.: automatiseren financiële administratie -> alleen onderzoek welk pakket aanschaffen? Aanschaffen & installeren? Opleiden gebruikers? Aanpassen werkprocedures? ...

*Projectgrenzen Breedte*

Hoe breed gaan we? -> breedte van project

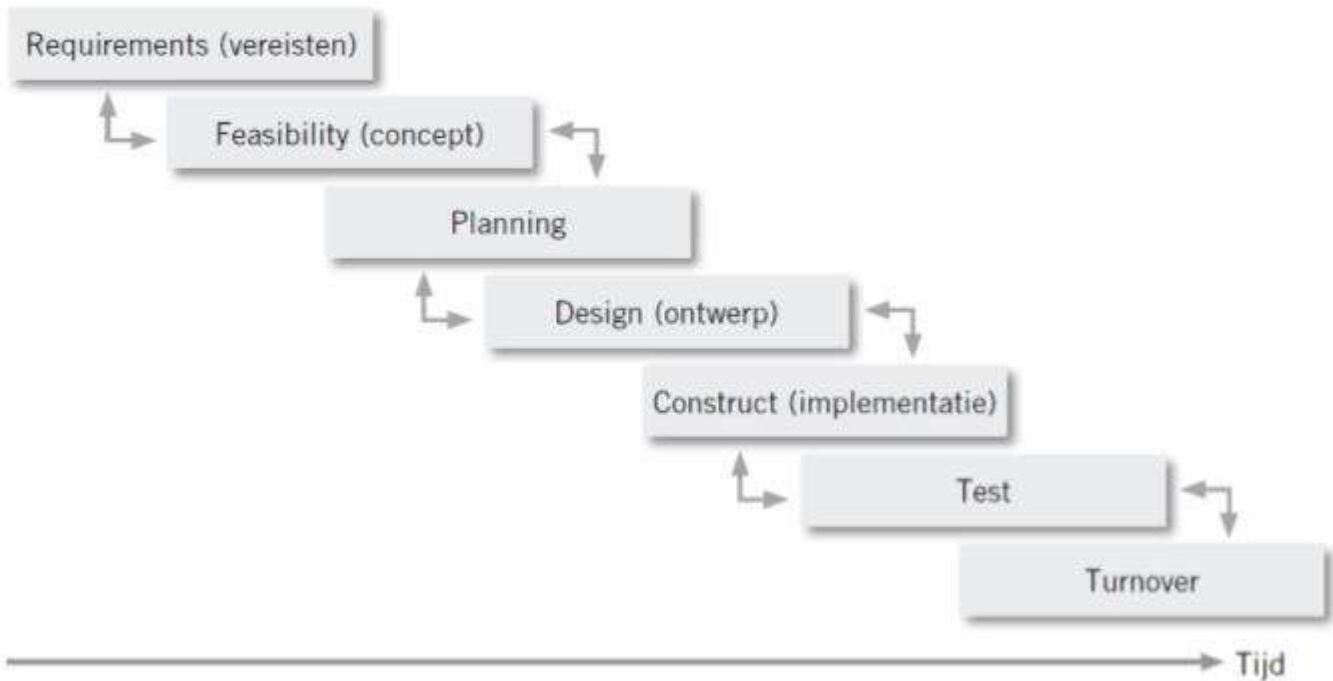
Vb.: automatiseren financiële administratie -> alleen verkoop & debiteuren (klanten)? Ook inkoop crediteuren (leveranciers)? Ook projectadministratie?

Pad 1: Projecten uitvoeren (fasering)

**Projectuitvoering** (=project execution/delivery) – omvat alle activiteiten die direct bijdragen tot de oplevering van het product (activiteiten uitgevoerd door projectteamleden)

Vb.: blueprint documenten, functionele & technische specificaties, programma's/modules, cutover-plan, ...

-> Gestructureerd vanuit tijdsbenadering -> **projectfases PMI (kennen!)**

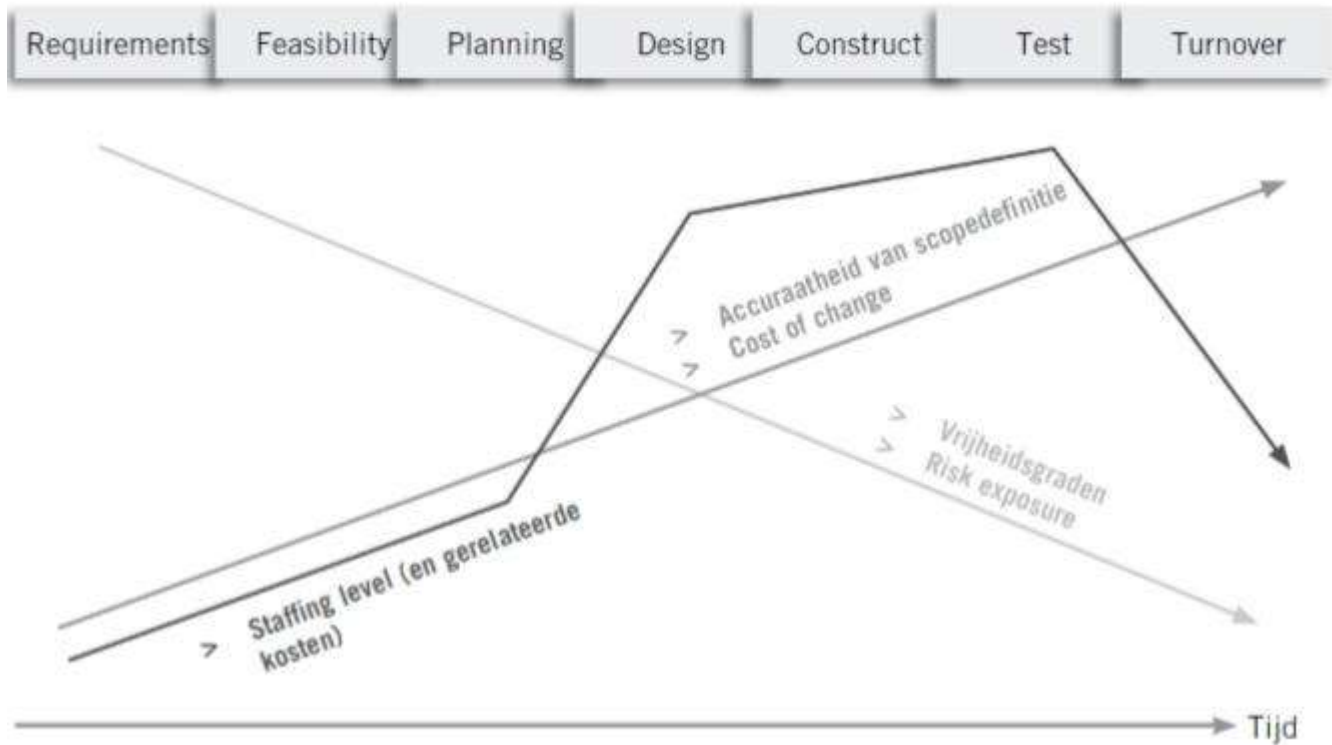


Figuur 2.1: Projectfases volgens het PMI.

⇒ Projectfases in realiteit vaak overlappend – Einde fase -> validering klant (rest team begint aan volgende fase)

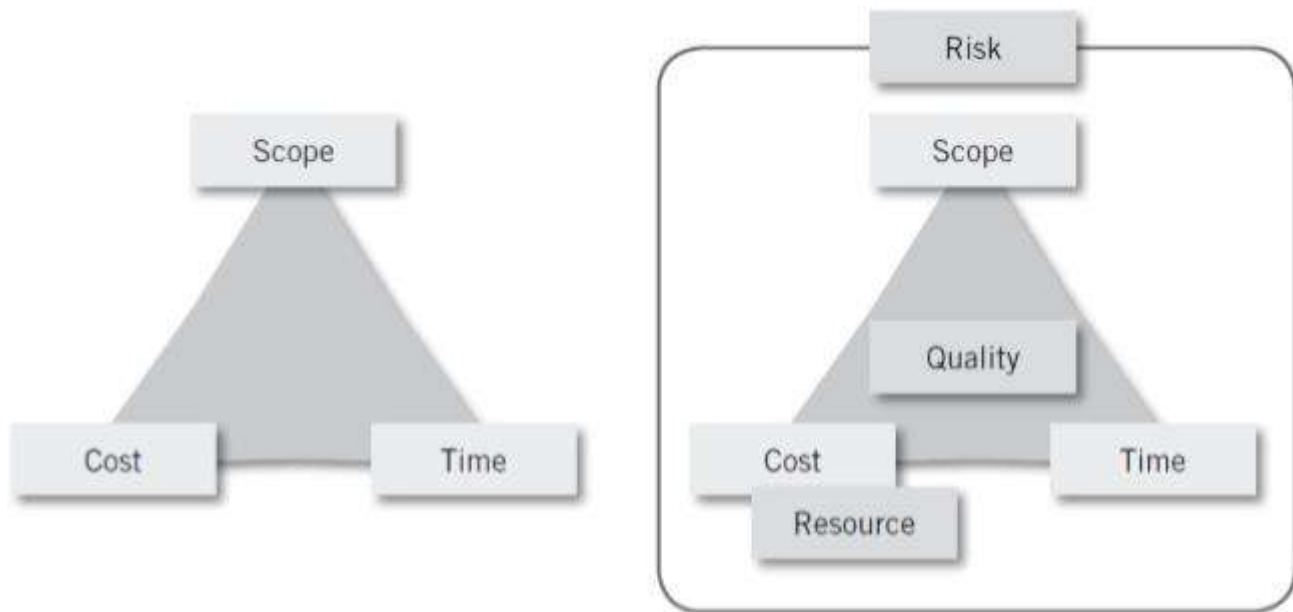
Doorlopen fases -> projectparameters evolueren mee:

**(Zelf hierin parameters en fases kunnen invullen)**



Figuur 2.3: Evolutie van enkele cruciale projectparameters doorheen de projectfases.

- **Staffing:** #resources/inspanningen neemt toe vanaf ontwerpfase, lichte groei in realisatie -> vanaf positieve testen: uitfasering team (!risico: kleiner team kan *critical issues* niet opvangen, PM dit bewaken)
- **Accuraatheid scope:** krijtlijnen steeds meer en meer vast -> noodzakelijk om *go-live* of oplever datum vastleggen; & kost/werk van wijzigingen stijgt (veel *rework*)
- **Vrijheidsgraad:** veranderingen worden steeds duurder ^, dus steeds minder vrijheid, MAAR risico's/onzekeerheid daalt ook (zowel in aantal, als impact)



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

**Triple constraint** (links): scope, cost, time > PM moet deze in het oog houden; verschuiving 1 vd 3 -> wijziging andere twee (**duivelsdriehoek**)

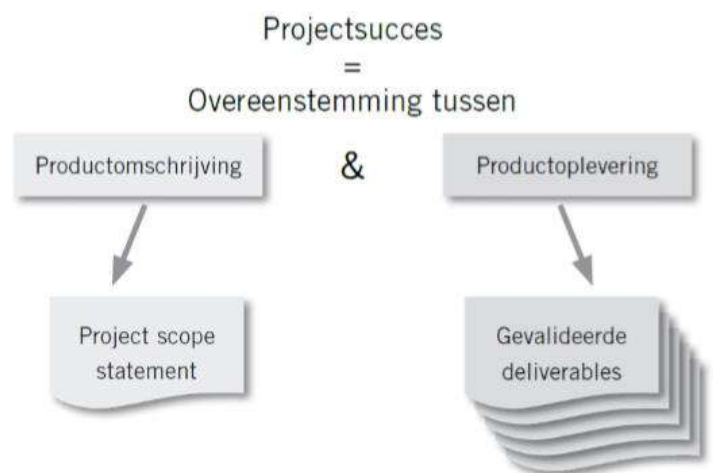
**6 constraints**: + Risk, Quality & Resource > Mag nooit inboeten op kwaliteit (daarom centraal); PM moet ook resources en risico's in het oog houden & managen

**Project leiden** = discipline noodzakelijk om actieplan uit te tekenen, uit te voeren en te bewaken opdat projectobjectieven en product volgens afspraken & zo efficiënt mogelijk worden gerealiseerd.

### H3 – Organiseren tot projectmatig werken

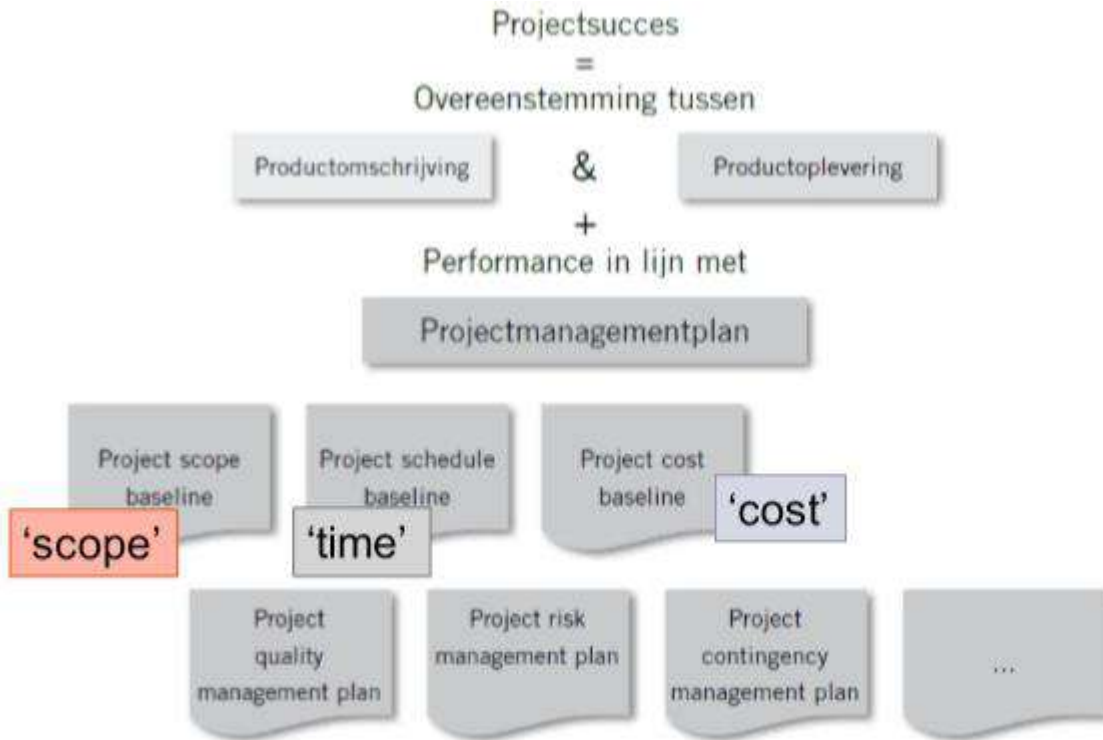
#### Projectsucces

- Perspectief klant: *customer satisfaction* centraal; klant verduidelijkt verwachtingen, beschrijft vereisten & stelt voorwaarden die kader schetsen waarin project gerealiseerd moet worden -> hoeverre verwachtingen ingelost = succesgraad project
  - Succesvolle oplevering**: 100% match met vereisten klant -> Validatie deliverables door klant centraal
    - o => Daarom: belangrijk behoeften klant gedetailleerd en formeer beschrijven



Figuur 3.1: Een succesvol project vanuit klantenperspectief.

- Perspectief PM: manier waarop project wordt uitgevoerd; om verwachtingen klant in te lossen -> PMP (PM Plan) dat scope, maar ook werkwijze beschrijft



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

### ! Verschil tussen fig. 3.1 & 3.2 kennen!

#### Succesvol project:

- Levert het juiste projectresultaat (doel, product) van afgesproken kwaliteit = **Scope**
- Is op de afgesproken tijd afgerond = **Tijd**
- Kost niet meer dan afgesproken = **Kost**

#### Waarom projecten falen

Hoofdbekommernis PM => beheren/beheersen verwachtingen klant = *managing expectations* -> faalt vaak

#### Veelvoorkomende redenen:

- Onrealistische schattingen & daarop gebaseerde tijdlijnen (ongefundeerde tijdlijn -> uitstel -> negatief)



The 2015 **CHAOS Report** has recently been released by the **Standish Group**. The CHAOS Reports have been published every year since 1994 and are a **snapshot of the state of the software development industry**. This year the report studied 50,000 projects around the world, ranging from tiny enhancements to massive systems re-engineering implementations.

MODERN RESOLUTION FOR ALL PROJECTS						
	2011	2012	2013	2014	2015	
SUCCESSFUL	29%	27%	31%	28%	29%	GESLAAGD
CHALLENGED	49%	56%	50%	55%	52%	GEWIJZIGD
FAILED	22%	17%	19%	17%	19%	GEFAALD

The Modern Resolution (OnTime, OnBudget, with a satisfactory result) of all software projects from FY2011-2015 within the new CHAOS database. Please note that for the rest of this report CHAOS Resolution will refer to the Modern Resolution definition not the Traditional Resolution definition.

Bepalende factoren

CHAOS FACTORS OF SUCCESS		
FACTORS OF SUCCESS	POINTS	INVESTMENT
Executive Sponsorship	15	15%
Emotional Maturity	15	15%
User Involvement	15	15%
Optimization	15	15%
Skilled Resources	10	10%
Standard Architecture	8	8%
Agile Process	7	7%
Modest Execution	6	6%
Project Management Expertise	5	5%
Clear Business Objectives	4	4%

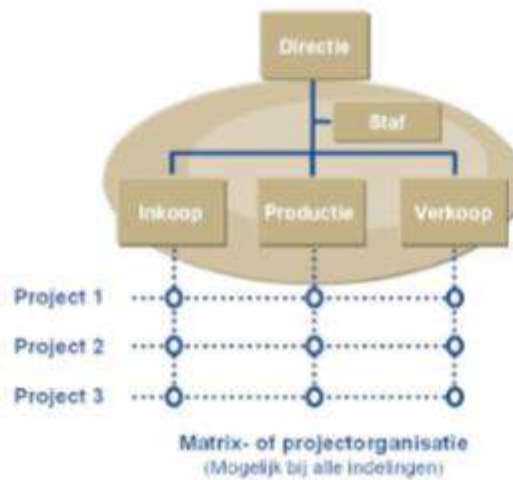


Projecten binnen een organisatie

Organisatievormen

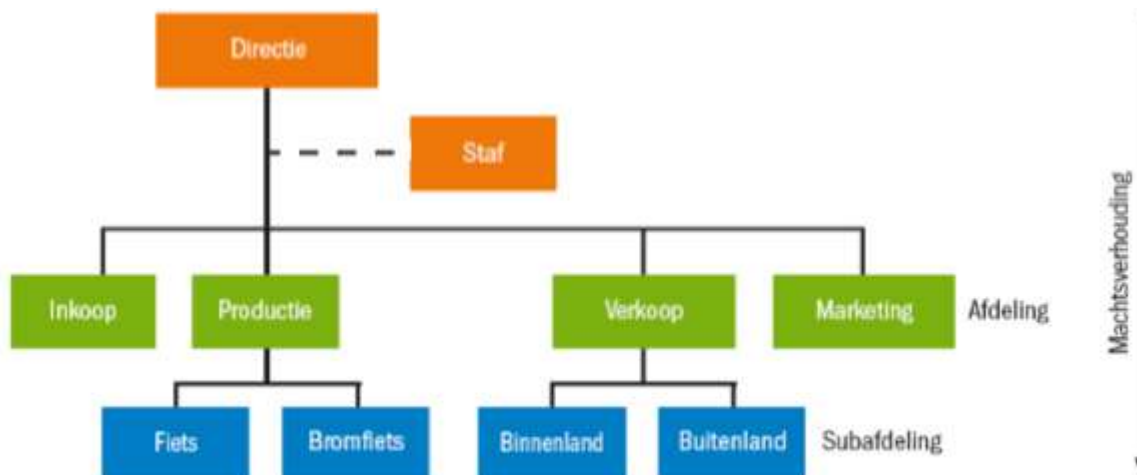


**Product**  
**Markt**  
**Geografisch**



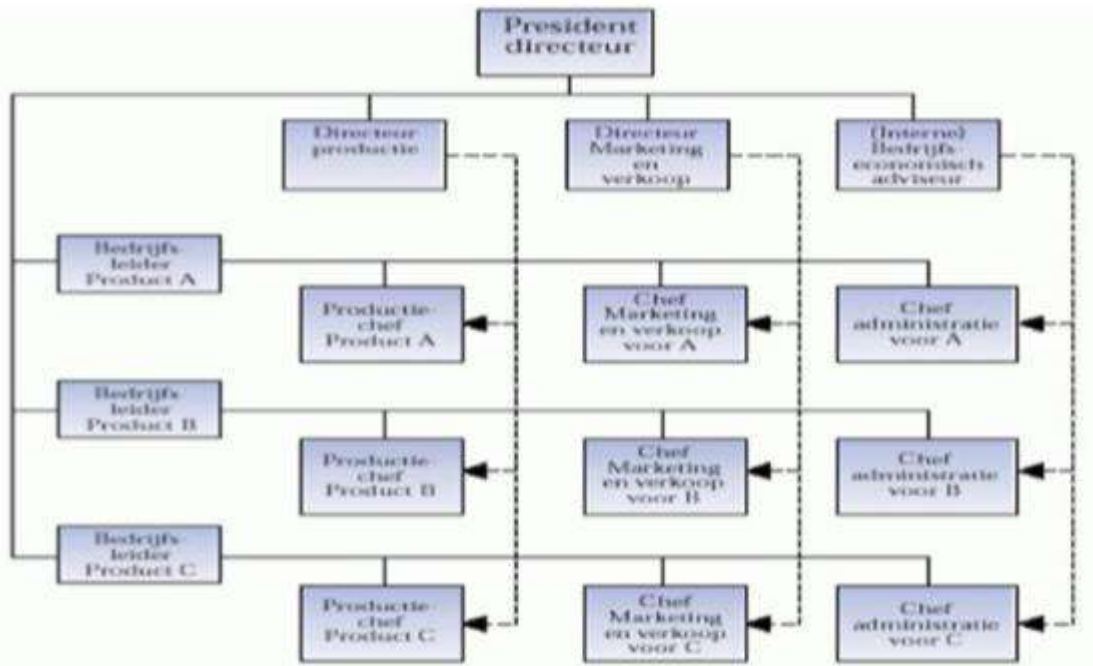
*Lijn-staf Organisatie*

Lijnorganisatie met stafafdeling (**functionele organisatie**)



*Matrixorganisatie*

**Voorbeeld matrixorganisatie (basisorganisatie)**



*Projectorganisatie*

**Projectorganisatie (tijdelijke matrixorganisatie)**

- Interdepartementaal
- Efficiënte toewijzing specialisten
- Organisatiedoelen gaan voor op departementale doelen
- Gebroken met principe: eenheid van gezag (soms nadelig en moeilijk)
- Tijdgebonden

*Tijdelijke projectorganisatie*

Tweebazen probleem:

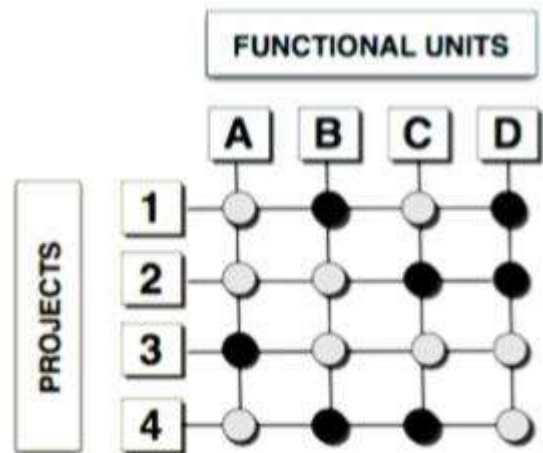
Projectmedewerker moet rapporteren 2 bazen: projectleider & afdelingshoofd

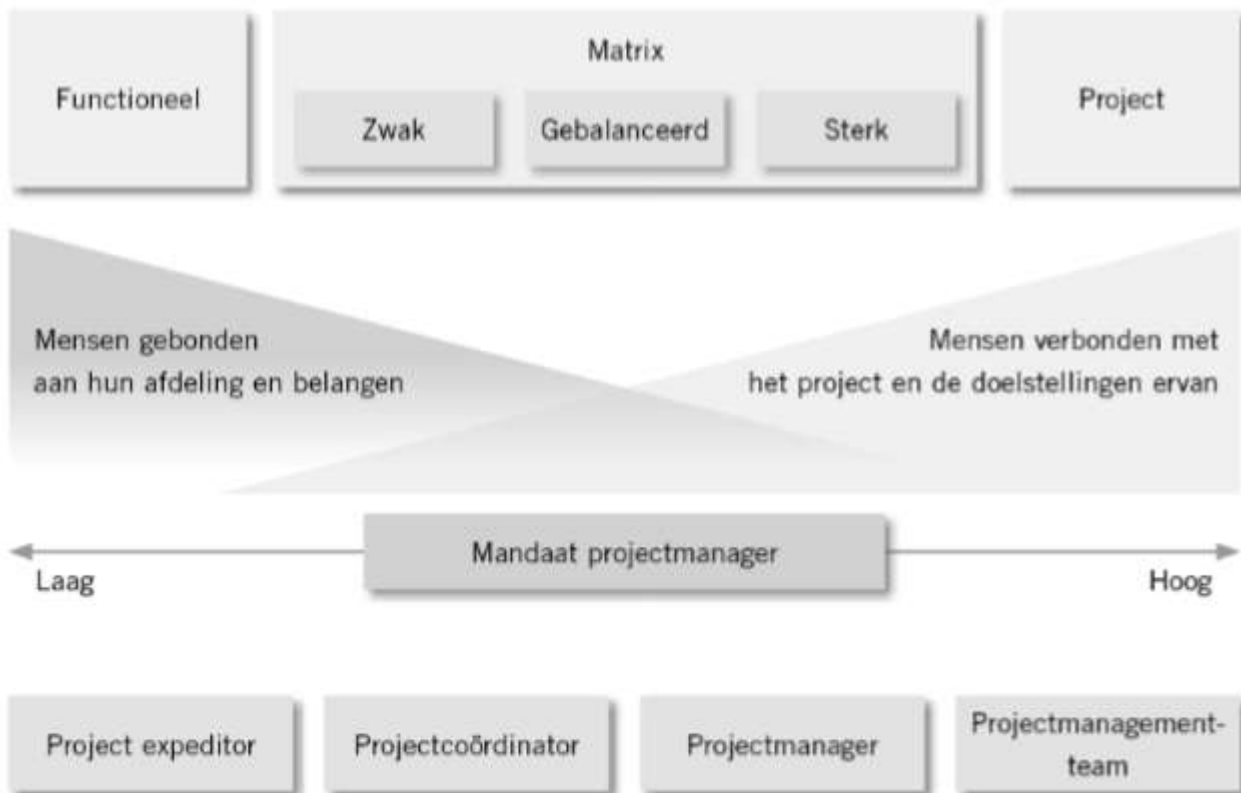
*Invloed van de organisatiestructuur*

PM niet mandaat om alle medewerkers aan te sturen of in te zetten; enkel als project rechtstreeks bijdraagt tot eigen afdeling.

PM hoopt op ruim mandaat, weinig rekening houden met rapporteringslijnen.

Hoe meer organisatie neigt naar projectorganisatie > hoe meer zekerheid mensen toegewijd aan project



> *Dedicated resourcing* (1 van de succesfactoren van project)

Figuur 3.3: Invloed van organisatiestructuur op projecten.

PM ook andere titel in functie van mandaat: invloed eerder beperkt > *project expeditor*

In projectorganisatie komt PM vaak recht in PMT (Project Management Team) met ruime scope van bevoegdheden.

### Projectmanager

#### Hard skills

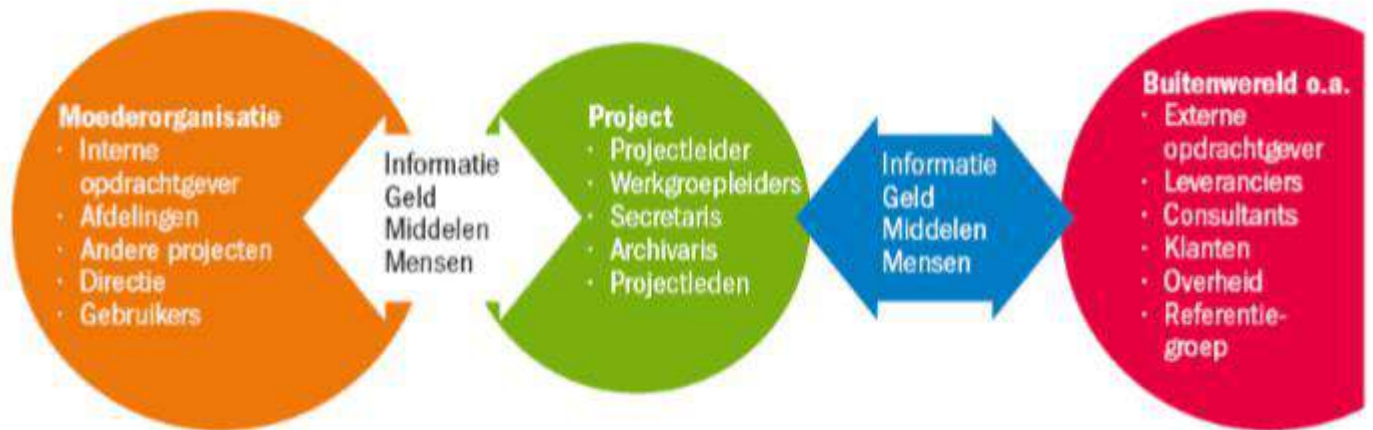
- Taakgericht
- Planner / controller: actieplan vastleggen & implementeren

#### Soft skills

- Mensgericht
- Communicatief
- Diplomaat / consensus (conflicten oplossen)
- *Stress management*
- *Creatief en oplossingsgericht zijn*

Uitgebreide projectomgeving: rollen & verantwoordelijkheden

Stakeholder?



Projectorganisatie

- Stuurgroep
- Opdrachtgever
- Projectleider
- Project/team-leden
- Toekomstige gebruikers

Rollen & verantwoordelijkheden

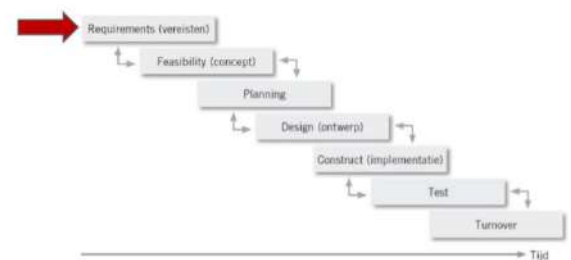
Persoon / projectentiteit	Rollen verantwoordelijkheid
<b>PMO</b>	Consistentie (std) Inzet resources over projecten heen Projectplanning Projectopvolging
<b>Functionele manager</b>	Opdracht afdeling vervullen <b>Input m.b.t. scope en vereisten</b>
<b>Proceseigenaar (process owner)</b>	Volledig beheren businessproces <b>Input tijdens concept -en ontwerpfase</b> 'Change'
<b>Sleutelgebruiker (key user)</b>	Vereisten mee bepalen (concept -en ontwerpfase) Mee reviewen deliverables Aanspreekpunt + klankbord op de werkvloer Verificatie applicatie (go-live en turnover) Testen applicatie als eerste
<b>Sponsor</b>	Initiatie projectvraag Eindverantwoordelijkheid product + beschikbaarheid finale middelen Directe communicatie met project manager Projectopvolging Voorzitter van stuurgroep
<b>Stuurgroep</b>	Centraal rapporterings- en <b>beslissingsorgaan</b> Maandelijks bijeenkomst + op cruciale momenten
<b>Businessanalist/ functional consultant</b>	Ondersteuning en advies bij optimalisatie van interne en externe processen (en procedures): AS-IS ⇔ TO-BE <b>Input voor 'business case'</b>

<b>Developer/ ontwikkelaar/ technical consultant</b>	Technische specificatie uitschrijven SW schrijven die nodig is om scope op te leveren (implementatiefase) Testen applicatie Oplossen defecten/bug
<b>Architect</b>	Bigpicture, keuzes moeten passen in bestaande architectuur (feasibility-, concept-en ontwerpfasen) Verschillende types SW -system -Enterprise Architect
<b>Testmanager</b>	Testplan uitwerken Motiveren teamleden en sleutelgebruikers om testscenario's uit te werken
<b>Tester</b>	Veelal onafhankelijk Uitvoeren testen en rapporteren aan testmanager (testfase)
<b>Transitie-/ Cutover-manager</b>	Vorbereiden, valideren en uitvoeren cutover-plan (oplevering) Overleg met projectmanager
<b>Change AdvisoryBoard (CAB)/ Change Control Board (CCB)</b>	Kanaliseren en beoordelen van 'change requests' (vragen tot verandering), change requests worden enkel na goedkeuring opgenomen in de project scope

## H4 – Process Requirements / Projecten identificeren

**Process Requirements** zijn de vereisten die een bedrijfsproces heeft ten aanzien van de ondersteunende IT-applicatie. Om die in kaart te brengen is het noodzakelijk de concrete bedrijfsprocessen uit te tekenen en te beschrijven zodat mogelijke optimalisaties in deze bedrijfsprocessen geïdentificeerd kunnen worden.

- Vereisten **bedrijfsproces** t.a.v. de **ondersteunende IT-applicatie**
- Processen eerst beschrijven, daarna optimaliseren (vooraleer te automatiseren)



Figuur 2.1: Projectfasen volgens het PM.

## Rollen & verantwoordelijkheden

**Best practice** = bepaalde manier van werken / 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/taak/applicatie.

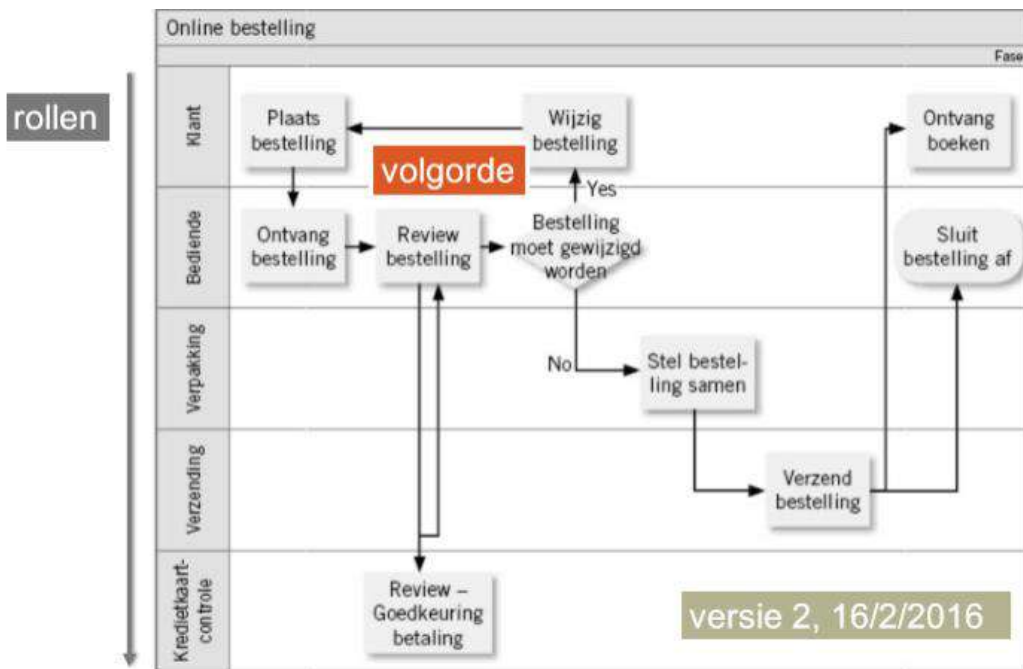
- **Business analyst** = stelt procesbeschrijvingen op gebaseerd op input van eind- en sleutelgebruikers, bestaande hadleidingen, documenten, ...; identificeert mogelijke optimalisaties
- **Process owner** = verantwoordelijk voor bepaald (deel)proces in onderneming; sleutelgebruiker die instaat voor identificatie/oplossen problemen; optimalisatie; geeft initiële input & finale beslissingen bij wijzigingen
- **Eindgebruikers** = huidige actoren in proces; voeren taken uit in proces; zullen met nieuwe applicatie werken
  - Leveren input voor opstelling van procesbeschrijving, vaak meer detail dan PO
  - Realiteit =/= theorie in uitvoeren taken werkvloer
- **Sleutelgebruikers** = specifieke groep eindgebruikers met goede kennis huidige proces, door ervaring
  - Nemen deel aan proces namens hele groep eindgebruikers

## Deliverables

Process requirements > **process flowcharts**

- Beschrijft verschillende stappen van proces, in specifieke volgorde, en geeft aan welke gegevens worden doorgegeven & welke document/output wordt gegenereerd

- Afhankelijkheden met andere processen
- Rollen = swimlanes
- Grote processen > opsplitsen in kleinere processen
- Niveau's: ondernemingsprocessen > afdelingsprocessen > deelprocessen > taken

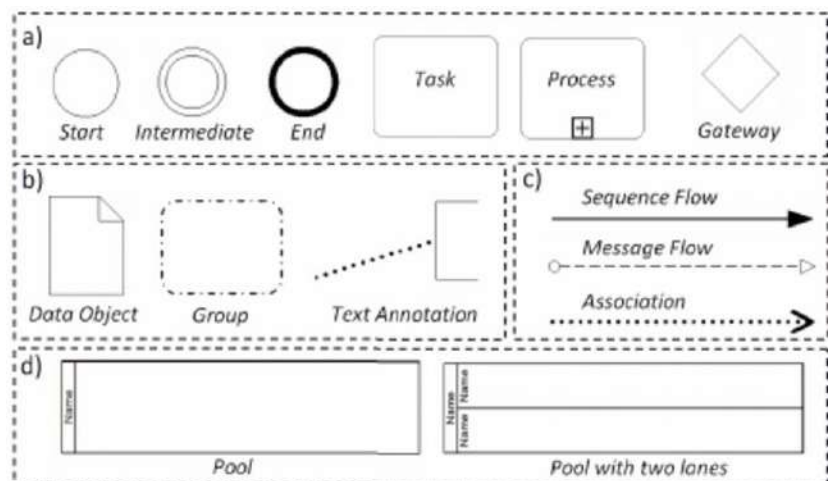


BPMN

Flow objects

Bouwstenen proces

- **Events** = gebeurtenis in proces, intern of extern
- **Activities/Tasks** = deelproces / taken binnen proces
- **Gateway** = punten waar proces wordt opgesplitst / samenkomt; vaak samen met beslissing



Connecting objects

Structuur proces, bouwstenen verbinden

- *Sequence flows* = volgorde activiteiten
  - o *Default flows* (pad wordt standaard gevolgd)
  - o *Conditional flows* (enkel wanneer bepaalde conditie voldaan)
  - o *Exception flows* (enkel wanneer uitzondering zich voordoet)
- *Message flows* = berichtenstroom tussen verschillende rollen
- *Associations* = afhankelijkheden tussen verschillende objecten

Swimlanes

Rollen / verantwoordelijkheden weergeven

Artifacts

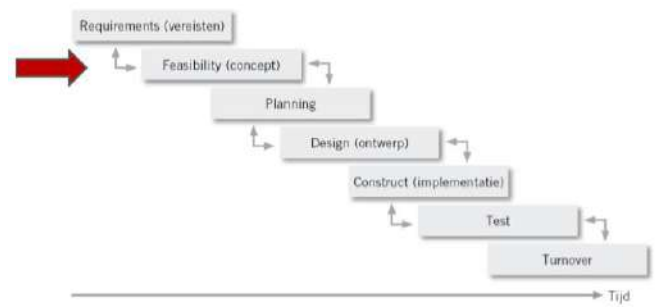
Extra info in procesdiagram op te slaan; bv.: dataobjecten (welke data nodig is om bepaalde activiteit uit te oefenen), annotaties, ...



## H5 – Concept / Projecten aanpakken

**Concept** = Uitgewerkt idee van project. Business requirements zijn geïdentificeerd. In deze fase wordt er o.a. op basis van een business case, nagegaan of een project voldoende opbrengst heeft & voldoende ondersteuning binnen onderneming.

- Uitgewerkt idee
- Voldoende 'opbrengst' en 'ondersteuning' => **business case** nodig



Figuur 2.1: Projectfasen volgens het PMI.

### Beschrijving & doel

Projecten gedreven door **factoren van buitenaf / extern**

Vb.: frank > euro, year 2000, fundamentele veranderingen BTW-wetgeving, security threats, ...

- Conceptfase beperkt
- Verwachte doel/resultaat duidelijk > business case eenvoudig
- Aanpassingen noodzakelijk > snel mogelijk overgaan tot actie (uitvoering!)

Projecten gedreven door **optimalisatietrajecten**

- Efficiëntie / Kostenbesparing
- Vraag van **management**
- Duidelijke nood aan verbetering > business case relatief eenvoudig
- Engagement organisatie nodig! (maatregelen niet altijd gewenst door eindgebruiker) > in conceptfase: verminderen weerstand & identificeren mogelijke sleutelspelers (change agents)

Projecten gedreven door **verbeteringstrajecten**

- Manier van werken
- Vraag van **eindgebruiker**
- Aandacht besteden aan **business case** > bewijzen wijzigingen winstgevend voor onderneming > management project autoriseren > veel engagement

Verloop conceptfase – deel 1

- |  |  |
|--|--|
| 1. Opstellen Business Case (BC)              | <ul style="list-style-type: none"> <li>- Kosten/Baten</li> <li>- Midden Termijn (MT) &amp; Lange Termijn (LT)</li> </ul> |
| 2. Voorstelling & goedkeuring aan management | Management beslist of project kan doorgaan op basis van business case en beschikbare budgetten                           |

1 & 2 herhalen indien nodig (niet voldoende resultaat, beperk budget, ...) > aanpassing voorstel, opsplitsing beheersbare / betaalbare deelprojecten

Verloop conceptfase – deel 2

- |                     |   |
|---------------------|---|
| 3. Commerciële fase | <ul style="list-style-type: none"> <li>- Lastenboek:                             <ul style="list-style-type: none"> <li>o Intern (proposal = definitievere kosten)</li> <li>o Extern</li> </ul> </li> <li>- Keuze Software en leverancier</li> <li>- <b>RFI versus RFP</b></li> </ul> |
| 4. Aanstelling PM   |   |

RFI – RFP

**RFI - Request for information**

- Niet bindend

**RFP - Request for proposal**

- Bindende offerte voor aanbieder (binnen bepaalde termijn)

- Algemene kostenraming	- Gedetailleerde kostenraming per activiteit
- Hulpmiddel opstellen BC	

## Rollen & verantwoordelijkheden m.b.t. conceptfase

Persoon / projectentiteit	Rol & verantwoordelijkheid
<b>Business analyst</b>	<ul style="list-style-type: none"> <li>- Opstellen BC <i>incl.</i> SWOT (op basis van projectbeschrijving – requirements)</li> <li>- Voorleggen BC aan mgm met sponsor</li> </ul>
<b>Eindgebruikers / Sleutelgebruikers</b>	<ul style="list-style-type: none"> <li>- Inhoud lastenboek</li> <li>- Mee software (SW) selecteren vanuit kennis van noden &amp; impact op (aangepast) bedrijfsproces</li> </ul>
<b>Ontwikkelaars</b>	<ul style="list-style-type: none"> <li>- Meewerken aan proposal (haalbaarheid &amp; indicatie kosten) -&gt; verkoop &amp; uitvoering hangen sterk samen bij SWprojecten</li> </ul>
<b>PM</b>	<ul style="list-style-type: none"> <li>- Overzicht</li> <li>- Neuzen in zelfde richting</li> <li>- Haalbaarheid bewaken ('cost' – 'time')</li> <li>- Risico-analyse (met BA)</li> <li>- SWOT (met BA)</li> </ul>

## Deliverables

### Business Case

- = Onderzoek naar de **levensvatbaarheid van een project**
- Te hernemen op verschillende tijdstippen in het projectverloop
- ⇔ feasibility / haalbaarheidsstudie = studie i.v.m. haalbaarheid en (technische degelijkheid van de oplossing)

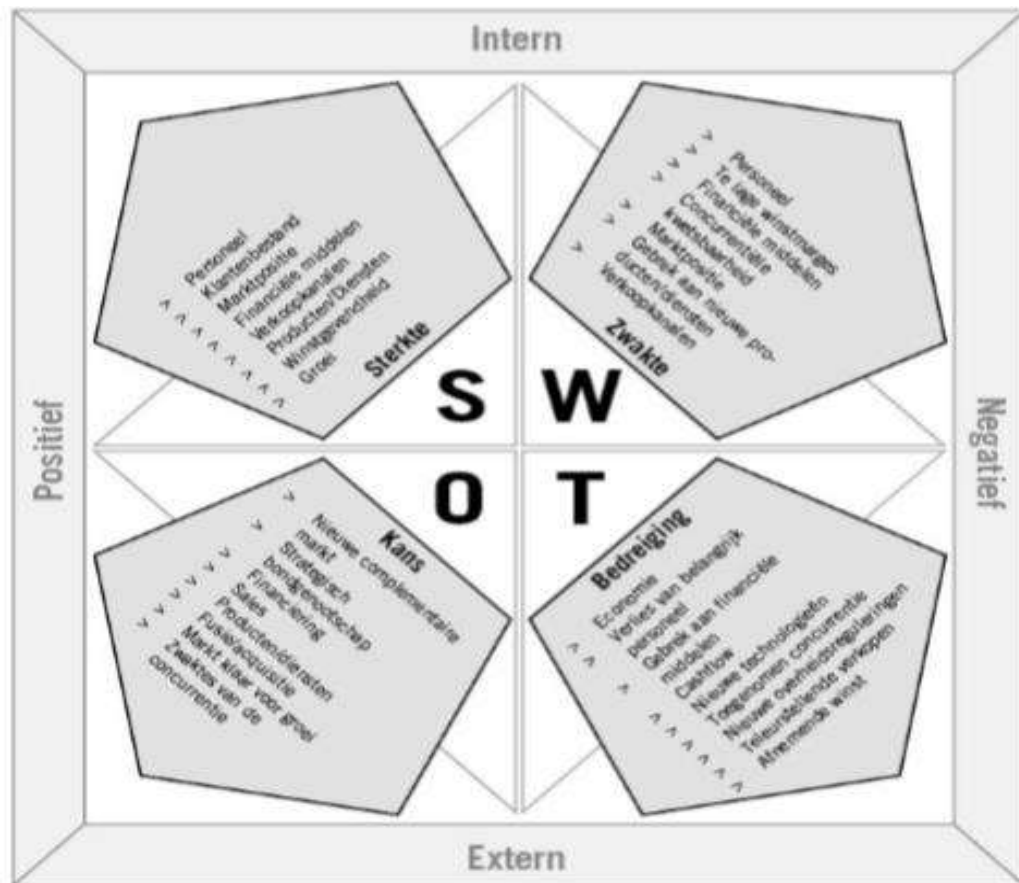
### Componenten BC

- Beschrijving business requirements, waarom?
- Kwalitatieve en kwantitatieve gevolgen
- Beschrijving betrokkenbedrijfsproces

- Welke oplossing (niet hoe)
  - SWOTanalyse
- Economische impact, ROI (Return On Investment)
  - MLT en LT (termijn)
- Aanbeveling tot beslissing

**SWOT-analyse** (Strengths, Weakness, Opportunities, Threats) = weegt SW project af tegen OT uit omgeving -> beter beeld risico's & gebieden die aandacht nodig hebben (**fig 5.1 kunnen tekenen!**)

- SW (**intern**): van project zelf, direct beïnvloedbaar door onderneming
- OT (**extern**): doen zich voor in omgeving van project; invloed op project; voordoen niet beïnvloed door onderneming of projectorganisatie



Figuur 5.1: De SWOT-analyse.

**ROI (Return On Investment):**

bekeken op middellange en lange termijn (MLT, LT); kosten project > ROI eerste periode direct na project altijd negatief, rekening houden met nieuwe kosten in de toekomst (onderhoud/ondersteuning, upgrades, wijzigingen, ...) > assumpties (niet exact te bepalen)

Project Proposal

6 delen

- Context project (belangrijk voor prijszetting) & impact
  - o Aanbieders moeten voldoende op de hoogte zijn van situatie waarin project zal verlopen > moeten rekening kunnen houden met WT om realistische kostenraming te geven
- Doelen (kwantitatief/kwalitatief) - **SMART** (Specific, Measurable, Achievable/Appointable, Realistic, Time-bound)
- Oplossing op basis van elementen lastenboek
- Kosten (**fixed price** ↔ **time & material**)
  - o Fixed: **vaste prijs**; risico leverancier; goed inschatten; **scope&change requests**
  - o Time/Material: **basis gepresteerde uren**; eindverantwoordelijkheid PM; risico verdeeld
- Voorwaarden en veronderstellingen/assumpties
- Voorlopige tijdslijn (doorlooptijden)

Scope Statement

- Samenvatting alle op te leveren onderdelen & soort; & aantal resources die nodig zijn
- Doel: eenduidige afspraak rond eindresultaat
- Change Request (CR): latere wijzigingen hierop
- **Scope creep** = steeds (organisch) uitbreiden van oorspronkelijke scope zonder formele wijzigingen aan oorspronkelijke scope met aanpassing van budget & planning waardoor ook formele goedkeuring van deze wijzigingen ontbreekt

Risico-analyse

- **Risico** = gebeurtenis die negatieve / positieve impact op eindresultaat. Afhankelijk van waarschijnlijkheid krijgen deze prioriteit score/gewicht.
- Analyse: om snel te herkennen & oplossen; of voorkomen

## H6 – Planning / Organiseren tot projectmatig werken

**Projectplanning** = dynamische verzameling van tools & documenten die de PM gebruikt om verloop project voor te stellen, op te volgen en bij te sturen

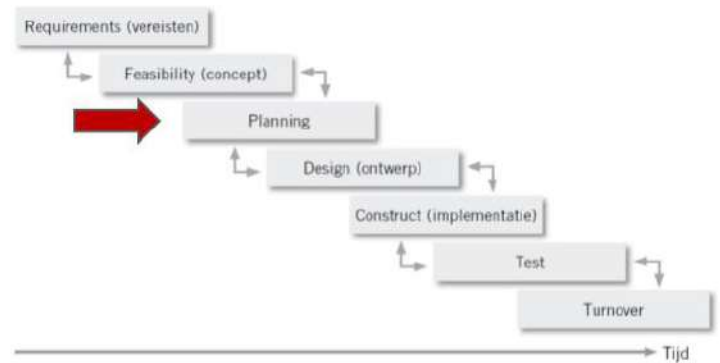
Beschrijving & doel

**Doel:**

- Organiseren van het werk van de projectmedewerkers
- Vaststellen totale projecttijd - **doorlooptijd**
- Vaststellen van de **gevolgen van vertraging**
- Vaststellen van de kosten van het project
- Bewaken van de voortgang en de kosten

**Praktisch: (Wie, Waar, Wanneer)**

- Welke activiteiten zijn er?
- Hoeveel tijd per activiteit – **inschatting per taak**
- Welke mensen nodig?
- Welke middelen nodig?
- Welke **afhankelijkheden** zijn er tussen activiteiten?
- 



Figuur 2.1: Projectfases volgens het PMI.

Rollen & Verantwoordelijkheden

Persoon/projectentiteit	Rol & verantwoordelijkheid
<b>Business analyst / technisch expert</b>	Inschattingen maken en corrigeren eerdere schattingen
<b>PM</b>	<ul style="list-style-type: none"> <li>- Uitwerken plan (SMART)</li> <li>- Concrete invullen resources (met <b>sponsors</b>)</li> </ul>

Deliverables

Kick-off meeting

- Projectplan voorstellen aan team (detail bekeken wie welke taken uitvoeren & wie verantwoordelijk): context, fundamenteen, objectieven
- Kennismaking betrokkenen: Sponsors & andere betrokken partijen ook aanwezig
- PM praktische afspraken: *naming conventions*, templates documenten, procedures, tijdsregistratie, ...

Projectplan

= Resultaat planning fase; gebruikt door PM, sponsors & team voor uitvoering project

= hiërarchische voorstelling van de deliverables die binnen project uitgevoerd moeten worden; opgebroken in werkpakketten

- Start met het **einddoel** dat je moet bereiken: *40 mensen hebben de cursus gevolgd*
- Deel verder op in **tussendoelen**:
  - o *cursusmateriaal klaar*
  - o *cursuslocatie geboekt*
  - o *draaiboek voor de cursus is klaar.*
- enz. ... tussendoelen **verder onderverdelen**.



**PBS (Product Breakdown Structure)** Vb.: computer  
 – main unit, monitor, mouse, keyboard

PBS ↔ WBS

Work BS	Product BS
- tool to identify the <b>tasks &amp; activities</b> required to deliver those outputs	- tool for analysing, documenting & communicating the <b>outcomes</b> of a project
- <b>how</b> to get there	- <b>where</b> you want to go

**Doorlooptijd** – geschatte tijd om activiteit af te werken

**Slack time** – vertraging die bepaalde activiteit mag oplopen, zonder dat die een impact heeft op project (flexibiliteit om activiteit uit te stellen)

- Maak overzicht alle mogelijke paden begin-eind > isoleer per pad activiteiten die niet deel van CP zijn > bereken slack door doorlooptijd desbetreffende pad van doorlooptijd CP te trekken

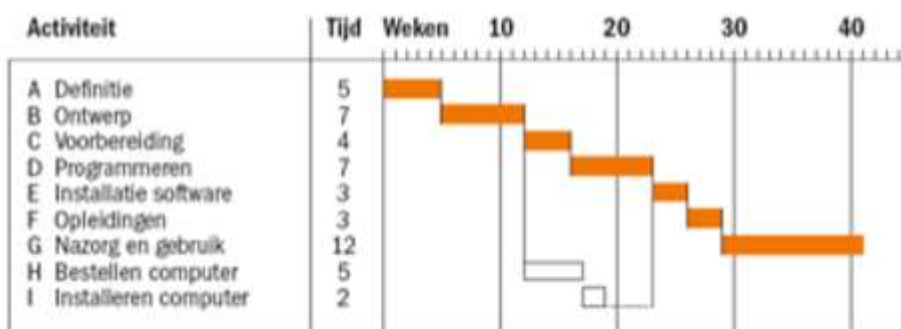
**Kritieke pad** – reeks activiteiten die totale doorlooptijd van project bepalen; vertraging hierop, vertraagd altijd project (geen slack time)

- Maak overzicht alle mogelijk paden begin-eind met doorlooptijd > langste tijd is CP

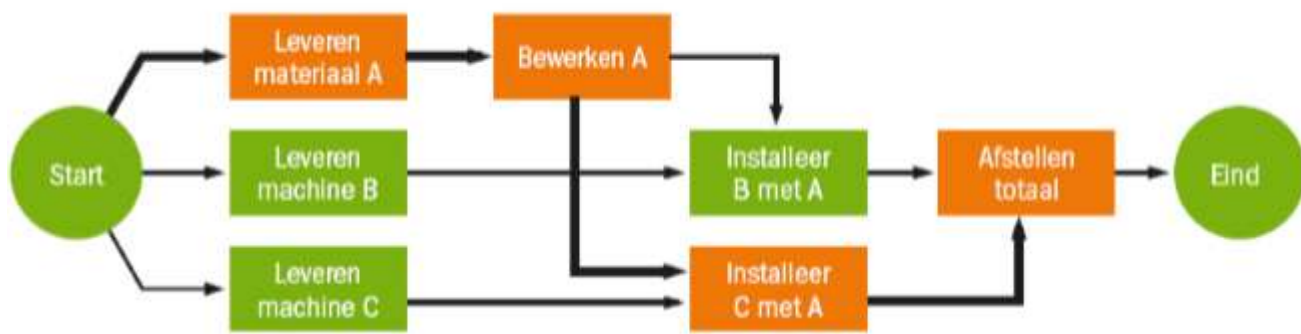
**Early – latest finish**: eerst mogelijke / laatst mogelijke einde van activiteit (vanuit gaande dat voorgaande activiteiten in netwerk ook eerst / laatst mogelijk geëindigd zijn)

**Early – latest start**: eerst mogelijke / laatst mogelijke start van activiteit (vanuit gaande dat voorgaande activiteiten in netwerk ook eerst / laatst mogelijk gestart zijn)

GANTT-chart (bar chart) / Strokenplanning



Afhankelijkheden tussen activiteit duidelijk te herkennen



GANTT ⇔ Network

GANTT-Chart	Network diagram
- volgorde/sequentie duidelijk	- volgorde niet zo duidelijk
- start- en eindpunten duidelijk	
- afhankelijkheden tss activiteiten niet zo duidelijk	- afhankelijkheden duidelijk
- 1 wijziging -> impact aanpassen Gannt	- 1 wijziging -> impact aanpassen diagram

### Afhankelijkheden

Naast invloeden vanuit project zelf, ook externe factoren:

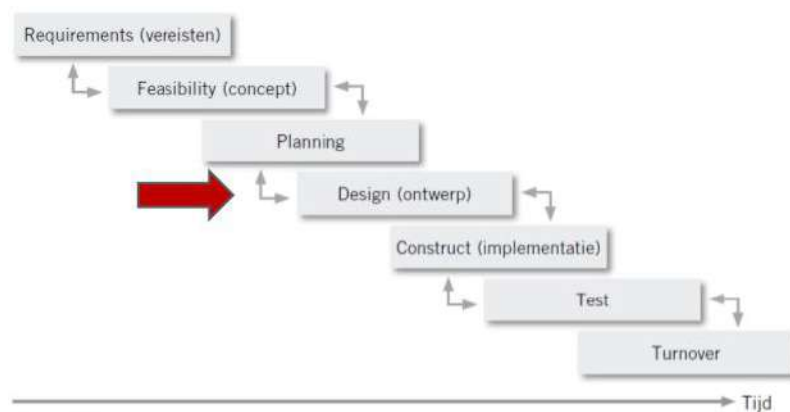
- Beschikbaarheid nodige resources (*levertermijnen, beschikbaarheid experts, ...*)
- Omgevingsfactoren (*economische conjunctuur die budget beïnvloed, ...*)
- Organisatorische wijzigingen (*wijziging organisatie klant/leverancier*)

## H7 – Ontwerp

**Blueprint** = document dat gedetailleerde beschrijving bevat van de oplossing die de leverancier zal opleveren tegen afgesproken prijs; bevat dus definitieve overeenkomst tussen (interne) leverancier & klant

### Beschrijving & Doel

Van: documentatie, procesbeschrijvingen & workshops  
 Naar: gedetailleerd ontwerp (**blauwdruk**) van nieuwe applicatie of wijzigingen bestand applicatie



Figuur 2.1: Projectfases volgens het PMI.



Workshops

Wie?

- **business analyst** + eindgebruikers

Hoe?

- voorbereiding vooraf (bestaande info meenemen)
- gegroepeerd per thema – functionaliteiten, deelprocessen, organisatiestructuur, locatie

Proof of concept – Prototype

Voor wie?

- eindgebruikers

Doel?

- begrijpen eindapplicatie d.m.v.
  - standaard demo of
  - ontwikkelde 'light versie'
  - mock-ups

Blueprint opstellen & valideren

**INHOUD**

- gelezen en aangevuld
- wat, wanneer opleveren tegen welke kost
- hoe omgaan met afwijkingen t.o.v. de blueprint

**VALIDEREN**

- handtekenen, beide partijen

Rollen & verantwoordelijkheden m.b.t. ontwerpfase

Persoon / projectentiteit	Rol & verantwoordelijkheid
<b>Business analyst</b>	<ul style="list-style-type: none"> <li>- Initiëren en sturen workshops</li> <li>- Uitwerken 'blueprint' document (verantwoordelijk)</li> </ul>
<b>Eindgebruikers / sleutelgebruikers</b>	<ul style="list-style-type: none"> <li>- Meedoen aan workshops</li> <li>- Input leveren voor blauwdruk</li> </ul>
Ontwikkelaars	<ul style="list-style-type: none"> <li>- Consultatie- / adviserende rol (technische specificaties)</li> </ul>
Sponsor(s)	<ul style="list-style-type: none"> <li>- Beslissen</li> <li>- Blueprint handtekenen = bestellen</li> </ul>
PM	<ul style="list-style-type: none"> <li>- Haalbaarheid bewaken</li> <li>- Tussenpersoon</li> <li>- Voorbereiden implementatie</li> </ul>

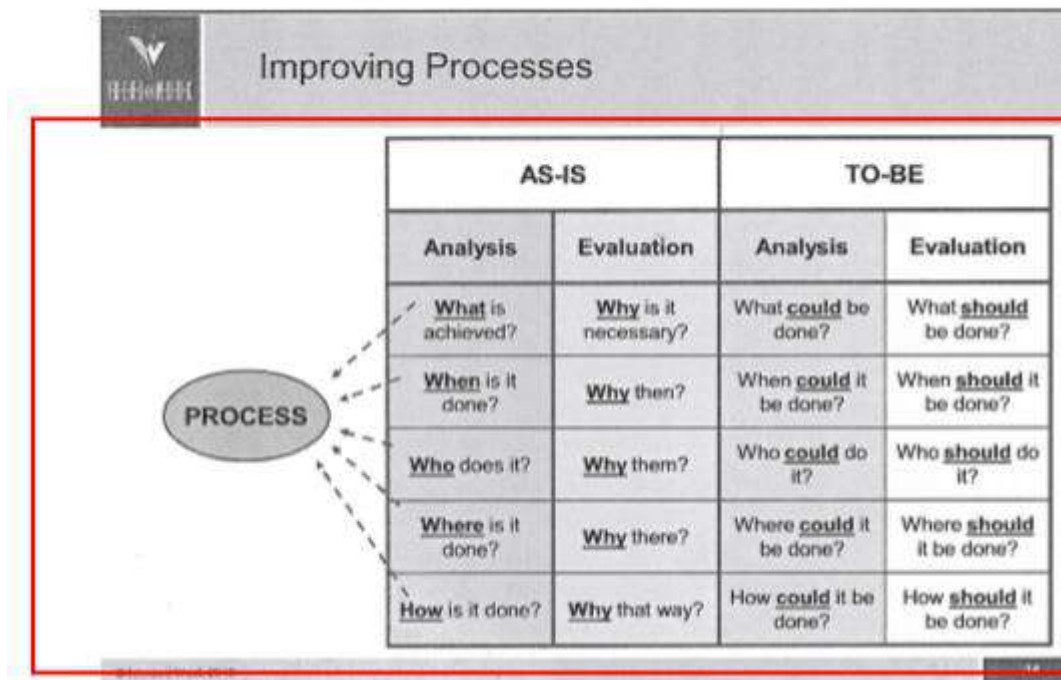
Deliverables

Blueprint

Procesbeschrijvingen **AS-IS & TO-BE**

Procesbeschrijvingen uit H4, opgesplitst in AS-IS & TO-BE:

- **AS-IS:** huidige situatie van een bedrijfsproces; realistische beeld van hoe momenteel gewerkt wordt & welke stappen gebruikers doorlopen
- **TO-BE:** situatie na afloop project; hoe verschillende eindgebruikers zullen werken in de toekomst
- > In workflows



Functioneel ontwerp

- Use case (OSA)

Technisch ontwerp

- UML: klassendiagram / sequence diagram (OSA)

Security, autorisaties & rollen

Interfaces

Datamigratie

Systeemarchitectuur

Testplan

Veronderstellingen / Assumpties

## H8 – Implementatie

**Implementatie- of bouwfase** = Volledige applicatie wordt gebouwd, volgens specificatie blueprint, langste fase.

Wijze afhankelijk van ontwikkelingsstrategie: sequentieel, incrementeel of iteratief (zie H10).

Naast functionaliteiten, ook volgende activiteiten

- Intallatie, upgrade & testen nodige hard -en software
- Configuratie systeem of software
- Bij implementatie sw, niveau's standaardisatie:
  - o Standaard / 'vanille' oplossing (wel config, maar geen code aanpassing)
  - o Standaardoplossing met klantspecifieke aanpassingen/uitbreidingen (vroeger: rechtstreeks in broncode, nu sw pakketten om programma's uit te breiden zonder broncode aan te passen, minder risicovol)
  - o Compleet nieuwe oplossing: 100% op maat, minder stabiel, meer werk
- Aanmaken testgegevens
- Aanmaak documentatie
- Opzetten rollen & autorisaties
- Opzetten interfaces

## Rollen & verantwoordelijkheden

Persoon / projectentiteit	Rol & verantwoordelijkheid
<b>Ontwikkelaars</b>	<ul style="list-style-type: none"> <li>- Ontwikkelen applicatie</li> <li>- Zie boven</li> </ul>
<b>PM</b>	<ul style="list-style-type: none"> <li>- Opvolging activiteiten (nauwgezet, meest gevoelig aan vertragingen)</li> </ul>
<b>Eindgebruikers</b>	<ul style="list-style-type: none"> <li>- Feedback</li> <li>- Beschikbaar voor vragen ontwikkelaars</li> </ul>

### Deliverables

Werkende oplossing die voldoet aan vereisten van de klant; sterk afhankelijk van ontwikkelingsstrategie (zie H10)

## H9 - Test

**Testen** = reeks handelingen uitgevoerd om te bepalen of de geboden oplossing voldoet aan de vereisten en verwachtingen van de uiteindelijke gebruiker.

### Beschrijving & Doel

Belang testen:

- Complexiteit projecten neemt toe; globale context, complexe technologieën, doorgedreven integratie
- Concurrentie sw markt groter, budgetten bedrijven kleiner; bedrijven sneller naar andere leverancier als ze denken dat investering niet opbrengt (goede naam belangrijk; testen > kwaliteitscontrole)
- Controles op sw nemen toe (strengere eisen auditeurs)
- SW meer geïntegreerd dagelijks leven; software faalt > leven verstoord (*treinen op tijd, lonen uitbetalen, ...*)

Vroeger: ongestructureerd

Nu: testmanagers & testers, gestructureerd, kwaliteitsproces, deel van risicomanagement > nodige kosten

Wat testen/nakijken?

- Hardware (*machines krachtig genoeg, genoeg ruimte server, ...*)
- Organisatie & procedures (*geautomatiseerde verlofaanvragen, maar manager moet tijdig goedkeuren*)
- Documentatie (*handleiding duidelijk nieuwe gebruikers*)
- Implementatie (*trial cutover – deel data al testen op testsysteem*)
- Conversie (*conversieregels doen conversie correct?*)

### Testtypes

#### Functionaliteitstesten

Eindgebruiker test of functionaliteiten correct werken (verificatie) en voldoen aan verwachtingen/vereisten (validatie).

#### Cross-platform-testen

Software functioneert op verschillende platformen (browsers, mobiel, ...); bestaan tools voor; best meerdere keren

#### Performantietesten

Bepaalt acceptatie van de oplossing; eindgebruikers verwachten snelle sw; moet ook voldoende gebruikers tegelijkertijd kunnen bedienen (*online ticketverkoop weinig performant*)

#### Gebruiksvriendelijkheidstesten

Met minimaal aantal handelingen en intuïtief alles terugvinden; testen door eindgebruikers

#### Veiligheidstesten

Gedigitaliseerde gegevens/data moeten goed beveiligd zijn van buitenaf; rollen/autorisaties en hun toekenningen moeten werken (dit vaak deel van functionele testen)

*Regressietesten*

Wijziging bepaalde functionaliteit > impact andere functionaliteiten; sw tegenwoordig modulair om herbruikbaarheid te garanderen > 1 wijziging, meerdere gevolgen

Regressietesten > na wijziging test alle andere geïmplementeerde functionaliteiten

Testlevels

*Unittesten*

Technische correctheid bouwsteen, geen functionele testen, door ontwikkelaars

*Integratietesten*

Samenwerking tussen componenten

*Systeemtesten*

Oplossing in zijn geheel. Software en al zijn mogelijk componenten (zie boven)

*Acceptatietesten*

Oplossing testen door eindgebruikers, nagaan functioneel werken & voldoen vereisten/verwachtingen

Rollen & Verantwoordelijkheden m.b.t. testen

Persoon / projectentiteit	Rol & verantwoordelijkheid
<b>Ontwikkelaars</b>	<ul style="list-style-type: none"> <li>- Unit- en integratietesten</li> <li>- Verbeteren software wanneer eindgebruikers fouten vinden</li> </ul>
<b>Testers</b>	<ul style="list-style-type: none"> <li>- Op basis van testscenario's: integratie- en systeemtesten</li> </ul>
<b>Eind-/Sleutelgebruikers</b>	<ul style="list-style-type: none"> <li>- Acceptatietesten (testen functionaliteiten zoals ze deze zullen gebruiken in dagelijkse activiteit)</li> </ul>
<b>Testmanager (met PM)</b>	<ul style="list-style-type: none"> <li>- Testactiviteiten coördineren</li> <li>- Testplan opstellen</li> <li>- Waken resultaten correct gedocumenteerd</li> <li>- Opstellen &amp; valideren testscenario's (met BA en sleutelgebruikers)</li> </ul>

Deliverables

Testplan (begin project), Testscenario's (basis: bedrijfsprocessen), Testdata, Verantwoordelijkheden & Planning, Testtools, Afspraken rond acceptatie,

- Testresultaten

Oorzaken fouten:

- Technische oorzaak > ontwikkelaar fout herstellen
- Fout functionaliteit > beslist in blueprint?
  - Functionaliteit anders in blueprint > ontwikkelaar aanpassingen aan sw component
  - Functionaliteit zoals in blueprint & eindgebruiker niet akkoord > change request procedure

H10 – Ontwikkelstrategieën

**Ontwikkelstrategie** = wijze waarop ontwerp, ontwikkeling en testen opgezet en ingepland worden. Afhankelijk van de ontwikkelstrategie verlopen deze fases sequentieel, incrementeel of iteratief.

Build & Fix

Testen uitgevoerd door klanten die product gebruiken; geen feedback tussen fases en fouten in sw komen pas aan het licht wanneer klanten dit melden > wordt opgelost > naar alle klanten gestuurd (*support package / update*)

Reactieve strategie > enkel schade beperken wanneer deze zich voordoet > vroeger sw minder belangrijk, dus toen acceptabel

Nu: enkel wanneer startende bedrijven goedkoop op markt brengen, in ruil voor testen product > door lage prijs verwachtingen minder hoog (vaak: technologie die door maar 1 aanbieder gemaakt kan worden)

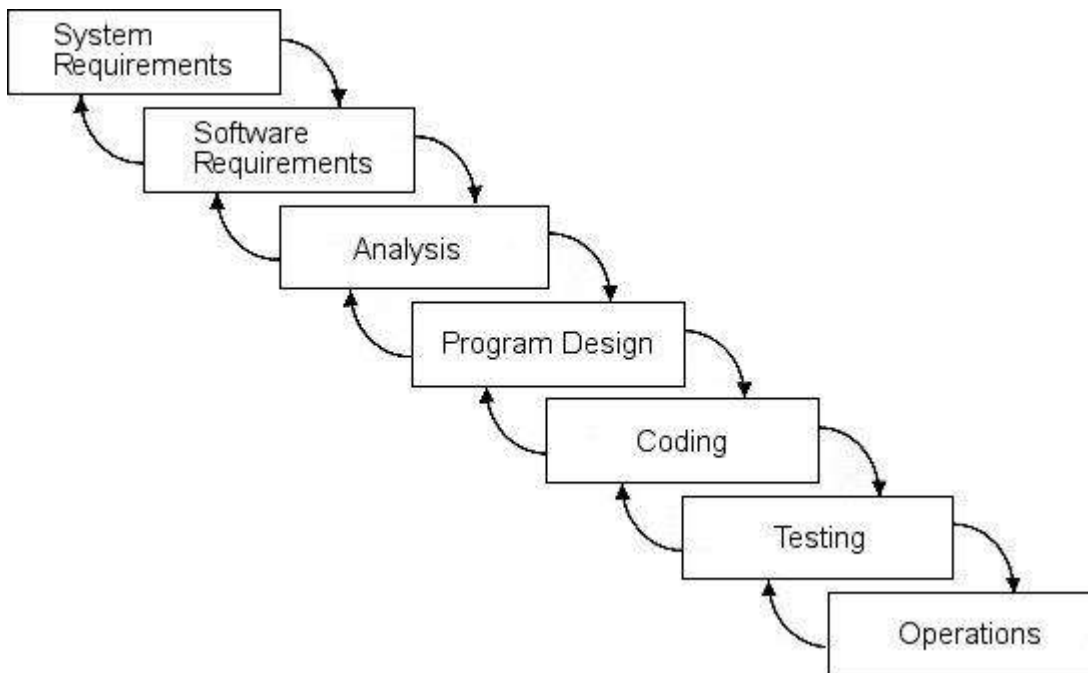
## Waterfall

= lineair, sequentieel model = traditionele ontwikkelingsstrategie, outdated

Feedback tussen 2 opeenvolgende fases, maar niet meer teruggaan; overgang fases vlot door uitgebreide documentatie

Enkel bruikbaar bij zeer eenvoudige sw implementaties, moeilijk bij complexere

Grote gevaar: testen op einde > grote fouten of niet volgens eisen klant > veel herwerk doen



## Rapid prototyping

Last probleem van late betrokkenheid klant op door prototypes te maken, 2 types:

- *Throw-away*: dient enkel om klant te tonen hoe er zou uitzien, niet deel van uiteindelijke opl
- *Light implementation*: prototype bevat al deel van uiteindelijke opl & zal gebruikt worden als basis verdere impl

Fases nog altijd sequentieel, maar kleine projectcyclus voor prototype

Voordeel: klant vroeg in proces zicht op resultaat; feedback mogelijk

Maar: nog altijd wachten tot na testfase voor afgewerkt product -> risico hoge kosten indien niet voldaan vereisten

## Incrementeel ontwikkelen

Volledig product > opgedeeld in kleinere stukjes > analyse, ontwikkeling & testen opgedeeld in stukjes functionaliteit  
Concept & volledig ontwerp wel echter vooraf al op punt gezet

1<sup>e</sup> release: basisfunctionaliteiten > daarop verder bouwen

Soms: incrementeel intern om ontwikkelingsproces te vereenvoudigen > nog steeds volledig afgewerkt naar klant

Pos: helpt bij beheersen zeer complexe sw projecten; wanneer na elke release ook acceptatietesten > feedback klant snel mee te nemen & risico's verminderen

Neg: contact met klant soms nog beperkt tot ontwerp- en testfase, nog te weinig feedback

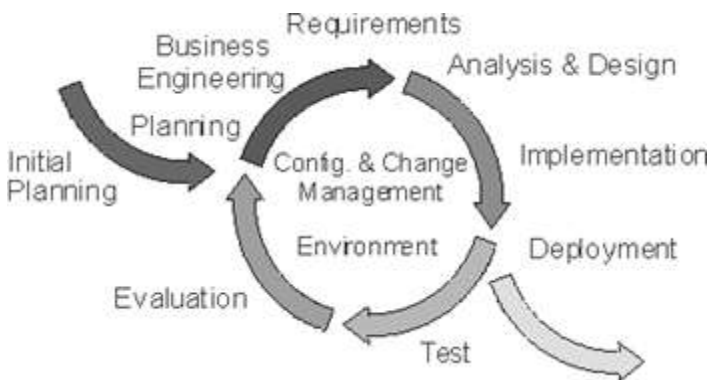
## Iteratief ontwikkelen

Volledige project (niet alleen ontwikkeling) opdelen in kleinere projecten > steeds nieuw deelprojectje (1 iteratie) uitwerken waarbij eindgebruiker van begin tot einde betrokken wordt > opgeleverde (tussen)oplossingen steeds volledig werkende oplossingen

Voordelen:

- Minimalisering risico ontevreden klant of falend project (risico fouten/problemen kleiner),
- Zeer flexibel op wijzigingen,
- Elke iteratie bruikbaar product > vertrouwen klant neemt toe & optie van te stoppen,
- Beter voor projectteam (complexiteit vermindert),
- Betere kwaliteit oplossing (meer validatiemomenten)

Nadeel: geen afgelijnde afspraken over wat er opgeleverd zal worden op einde project > sluitend budget vaststellen moeilijk (bij aanvang enkel initiële scope mogelijk)



## Agile ontwikkelen

Variante iteratief mode: aandacht aan directe inzetbaarheid software na elke iteratie, bewust zeer kort > klant ziet vooruitgang; geen concrete planning, continu aanpassen aan vereisten klant

4 vuistregels:

- Personen & interacties boven processen & tools
- Software die werkt boven lijvige documentatie
- Samenwerking met klant boven onderhandeling over contract
- Omgaan met verandering boven volgen van een plan

## SCRUM

- ⇒ Meest populaire Agile principe strategie
- Iteratie = Sprint (2-4 weken)
- *Product backlog* = lijst functionaliteiten die geïmplementeerd moeten worden (onderhevig aan verandering)
- *Sprint backlog* = ^ die tijdens huidige sprint moeten ontwikkeld worden
- Klant bepaald prioriteit van deze functionaliteiten > sprint-planning op basis hiervan
- Tijdens sprint geen wijzigingen mogelijk
- PM = *Scrum Master*, team max 9p (geen wijzigingen tijdens sprint hierin)

Communicatiemomenten:

- *Daily standup meeting (scrum meeting)*: ci. 15min., elk teamlid zegt wat hij heeft gedaan, vandaag gaat doen, eventuele problemen
- *Sprint review (einde sprint)*: Getoond wat er tijdens sprint is gerealiseerd
- *Sprint retrospective*: Met team & klant kijken wat wel & niet werkt



- *SCRUM-board & burndown chart*: post-its per story/functionaliiteit/bug > burndown geeft weer in planning, uitvoer, of klaar > goede scrum burndown lineair dalen > **kanban board**

Zie ook: Cegeka (!)

## H11 – Turnover

**Turnover** betekent de overdracht van het afgewerkte product naar de klant. Daarbij dient niet enkel software te worden overgedragen, maar ook documentatie, training & kennisoverdracht voorzien moeten worden zodat de klant met de applicatie kan blijven werken nadat de leverancier vertrokken is.

### Beschrijving & Doel

Vrijgeven sw applicatie in de operationele omgeving = 'go-live', maar ook:

#### Documentatie

Technische & functionele documentatie

Technische: beschrijft implementatie in detail, met beschrijvingen van geïmplementeerde sw, parameters & klant specifieke code (+ goede docu in code zelf voor toekomstige ontwikkelaars) > Vaak ook voor IT-afdeling

kennisoverdracht, waarbij leverancier opleiding voorziet rond nieuwe sw en aangeeft waar docu is

Functionele: = gebruikershandleiding + FAQ

#### Training

Alle eindgebruikers moeten training krijgen in nieuwe app.

Aantal: al geleerd voor testen

Moet best vóór go-live

Door leverancier aan alle eindgebruikers of aan trainers/sleutelgebruikers (*train-the-trainer*) > minder weerstand eindgebruikers als collega de training geeft i.p.v. externe

#### Support

Ondersteuning voor eindgebruiker in geval van fouten/problemen na go-live

Periode van 'hypercare': meestal hoogst 6-8 weken, door leverancier, fouten & functionele vragen, nadien IT of via supportcontract, opgezet volgens ITIL (zie later)

**SLA = Service Level Agreement:** contract tussen klant en leverancier dat in detail beschrijft welke service de klant van de leverancier mag verwachten. Beschrijft o.a. helpdesk, reactiesnelheden & boetes wanneer voorwaarden niet bereikt worden. (~OLA tussen 2 afdelingen zelfde organisatie)

Ook: installeren upgrades/support packs, implementeren change requests & routine operaties (*aanmaken nieuwe users*) > kunnen intern of extern (laatste jaren: tendens outsourcen)

*Support pack* = verzameling verbeteringen van bestaande software, dat meestal een oplossing biedt voor gekende problemen. Het is geen nieuwe versie van de bestaande sw.

*Upgrade* = nieuwe versie van bestaande sw, die naast de verbeteringen uit vorige support packs, vaak ook nieuwe functionaliteiten bevat die door meerdere klanten werden gevraagd.

Grote bedrijven: gelaagde organisatie voor ondersteuning:

- front line support: helpdesk medewerkers die functionele vragen beantwoorden & eenvoudige routineoperaties
- advanced support: problemen analyseren, parameters wijzigen & controles uitvoeren
- experts, klachtenbehandeling, escalatie (experts enkel bezig met echte technische problemen): wijzigen code

IT Service Management

= houdt zich bezig met implementatie & kwaliteitsbewaking van de verschillende IT-services die een organisatie nodig heeft ter ondersteuning van de business.

Deliverables

Documentatie

Procedures

Procedures voor melden van problemen; aanvragen van change requests; aanvragen gebruikers & rollen; ...

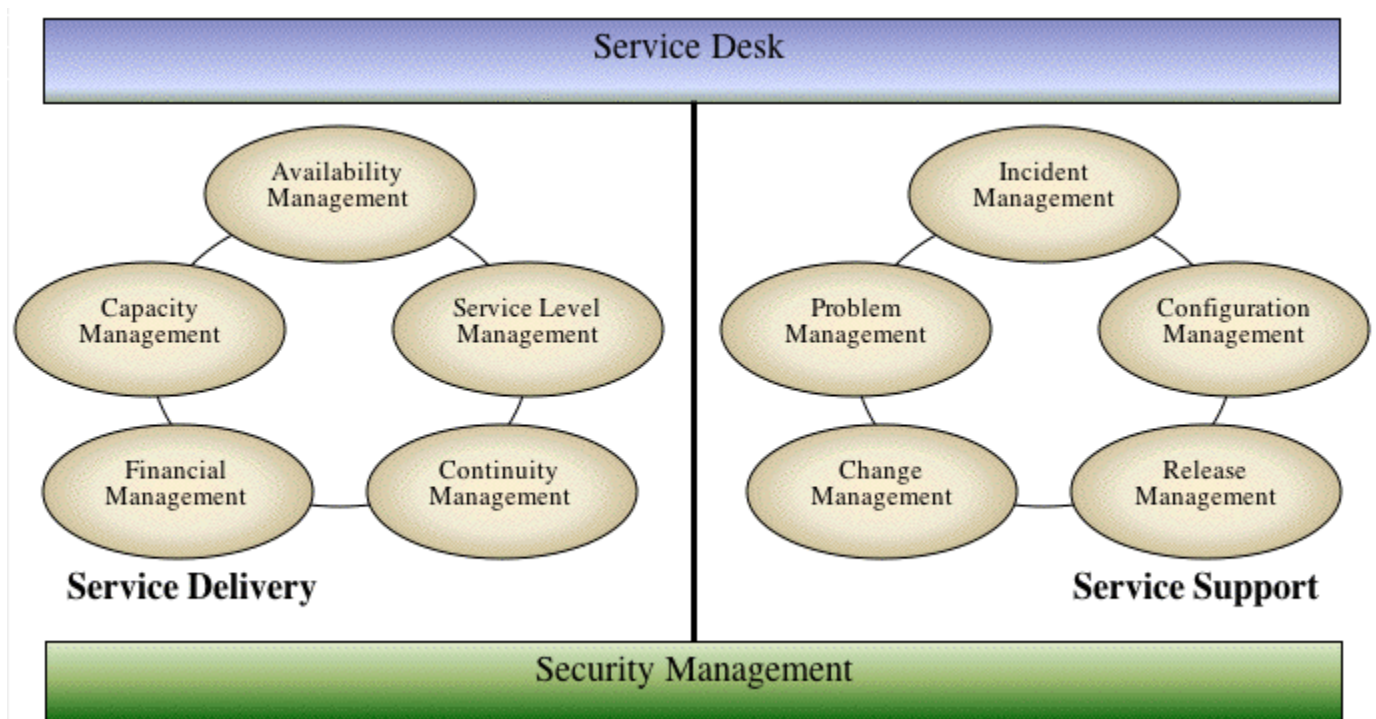
Cutover-plan

= Draaiboek voor go-live, afhankelijkheden in kaart gebracht, monitoring tool go-live dag voor PM

Methodologieën – ITIL

= Information Technology Infrastructure Library

- **Service support** = 5 processen die ondersteuning bieden voor de dagdagelijkse invulling van IT-services
- **Service delivery** = 5 processen die op ML Termijn proberen IT-service te plannen/verbeteren



**Doel:** efficiënte IT services afleveren die afgestemd zijn op noden eindgebruikers, stabiel & duurzaam zijn

Service support

*Incident management*

Dagdagelijks gebruik IT-systemen minimale hinder onderbrekingen.

**Incident** = gebeurtenis die zorgt voor onderbreking in, of vermindering van, de kwaliteit binnen bepaald IT-service

- Incidenten registreren, catalogiseren & eindgebruiker verder helpen zodat deze taken kan hervatten
- Prioriteiten aan incidenten toekennen (helpdesk)
- Reduceren impact incident op IT-service; analyseren oorzaak > niet bij helpdesk!

*Problem management*

Voorkomen dat bepaalde incidenten terugkeren

**Probleem** = onderliggende oorzaak die 1 of meer incidenten veroorzaakt; indien gekend: **gekende fout** > *workaround* of definitieve oplossing bekend

**Workaround** = alternatieve werkwijze om bepaalde taak uit te voeren, zonder dat het probleem in eerste instantie wordt opgelost > eindgebruiker kan taak hervatten (vaak manuele alternatieven voor geautomatiseerd proces)

- Incidenten analyseren om problemen te vinden
- Problemen registreren en prioriteit geven
- Indien mogelijk: probleem gewijzigd naar 'gekend probleem'
- Change request opstarten om probleem op te lossen

**Service request** = standaardaanvraag, waarvoor meestal procedure beschikbaar is (*user aanmaken, adreswijziging*)

#### *Change management*

Beheren alle veranderingen die worden aangevraagd; zorgt voor gestandaardiseerde methodes & procedures voor snelle & efficiënte afhandeling

**Wijziging** = oplossing voor een probleem of toevoeging nieuwe functionaliteit

- Accepteren, registreren, plannen en testen verschillende wijzigingen (meestal klein)

#### *Release management*

Beheer IT-diensten aangeboden door verschillende leveranciers; coördineert releases sw, hardware, upgrades, ...

**Release** = verzameling van voorziene wijzigingen in het systeem op een gecontroleerde manier operationeel zetten

- Plannen & monitoren implementatie van nieuwe hardware, sw & docu
- Coördineert met change & configuration management (voor update docu)
- Rekening houden verwachtingen gebruikers & management

#### *Configuration management*

Beschrijving van alle componenten van de IT-infrastructuur (sw, hw & docu)

**Configuration item** = specifiek deel van de IT-infrastructuur; **Config mgm db** = registratie alle config items

- Onderhoud configuration management database (CMDB)
- Integratie / Afhankelijkheid tussen items documenteren

#### *Service delivery*

##### *Service level management*

- Garandeert kwaliteit IT-services door afspraken te maken met verschillende partijen
- SLA's & OLA's
- Plan van aanpak om continuïteit van service te waarborgen wanneer problemen zich voordoen
- Dienstencatalogus opstellen (klanten zien lijst van opties om aan SLA/OLA toe te voegen)

##### *Availability management*

- Garandeert op kosteneffectieve en duurzame manier de beschikbaarheid van alle IT-services noodzakelijk voor eindgebruikers
  - Reliability (voortdurende beschikbaarheid)
  - Maintainability (snel terug operationeel krijgen bij problemen/storingen)
  - Serviceability (garantie door (externe) dienstverlener dat IT-infrastructuur altijd beschikbaar zal zijn)
  - Resilience (methodes om betrouwbaarheid IT-infrastructuur te verhogen)
  - Security (veiligheid IT-infrastructuur)

##### *Capacity management*

- Zorgt voor beschikbaarheid van capaciteit

- Garandeert performantie van systemen op kosteneffectieve manier

#### Financial management

- Beste IT-service tegen beste prijs > op zoek naar betaalbare opl op de markt & begeleid aankoopproces van nieuwe diensten en applicaties

#### IT service continuity management

- Zorgt dat IT-services binnen afgesproken tijd terug beschikbaar zijn wanneer onderbreking service zich voordoet
- Bewaakt kwaliteit van de garantie op continuïteit (niet zelf acties ondernemen bij onderbreking, dat is deel service support)
- Risicoanalyse waarin grootste risico's onderbreking van service zijn
  - Bekijkt herstellingsopties
  - Maakt plan van aanpak
  - Test plan van aanpak geregeld om relevantie ervan te verzekeren

(Zie ook ITIL lezing)

## Extra 1 – Release Management

**Release** = Een release is een nieuwe versie van de applicatie waarin een verzameling in het proces change management vastgestelde wijzigingsverzoeken als een samenhangend geheel wordt ontworpen, gerealiseerd, getest en ingevoerd.

#### Naar MoSCoW

##### M - must have:

- deze requirements moeten in het eindresultaat terugkomen
- zonder is het product niet bruikbaar

##### S - should have:

- zeer gewenste requirements
- zonder is het product wel bruikbaar

##### C - could have:

- deze requirements komen enkel aan bod als er tijd genoeg is

##### W - won't have (= would have):

- deze requirements komen niet aan bod in dit project
- kunnen in de toekomst, bij een vervolgproject, interessant zijn

#### Check gebruik

Na een release is er nieuwe functionaliteit in gebruik zijn genomen

Hoe controleer je of dit (juist) wordt gebruikt?

- **Kengetallen** (vb.: *percentage studenten ingelogd vorige maand*)
  - Objectbepaling: welk onderdeel van bedrijfsvoering door kengetal bekijken?
  - Operationalisatie: hoe druk ik wat ik van de bedrijfsvoering wil bekijken uit in kengetal?
  - Welke waarden aanvaardbaar / niet?
  - Hoe kom ik aan gegevens en bereken ik kengetal
  - Hoe waarde te beschikken stellen?
  - Uit systeem halen: SQL/Rapport
  - **Startpunt** analyse -> redenen om stuk niet te gebruiken

• **Meekijken met de gebruiker**

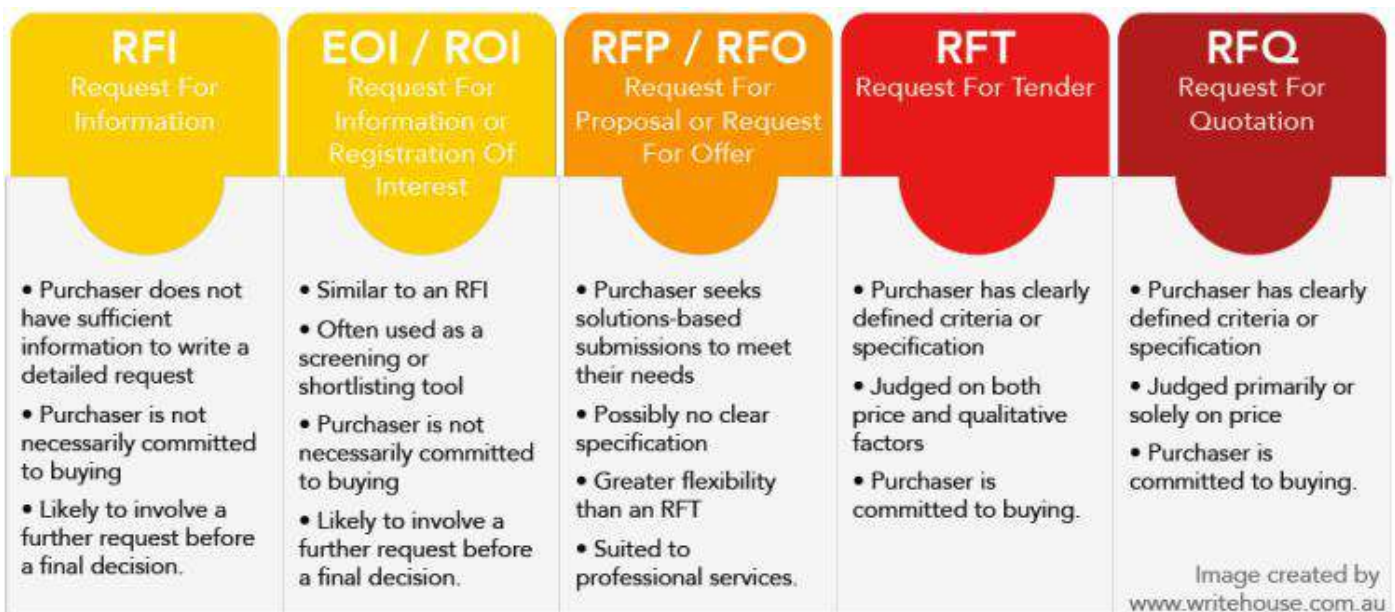
- Kijken naar handelingen in systeem
- Kijken of/hoe gebruik nieuwe/aangepaste functionaliteiten beter kan
- Onderscheiden van nieuwe wensen

Grip!

- Betrek gebruikers/stakeholders
- Maak keuzes: ook als je functionaliteiten niet implementeert
- Continu proces
- Checken geeft inzicht
- Na go-live ben je nog niet klaar!

Extra 2 – Financial Aspects PM

RFI ⇔ RFP/RFO



Offerte inhoud

- Doel
- Voorstel XXX
  - Voorgestelde oplossing*
  - Specificaties*
  - XXX Configuratie*
- Documentatie
- Optionele producten
- Prijsopgave

RFP

Zeer detail georiënteerd, vb.:

Context, technical features, project implementation plan, service delivery plan, contract, bidder’s reply