




2019-2020

ITIL - EEN INLEIDING

IT-ORGANISATIE

PCVO LIMBURG
HBO5 Informatica



HOOFDSTUK 1: ITIL?

Sinds de jaren '70 heeft IT zich ontwikkeld van een niche-product naar een onmisbaar onderdeel van de economie. In de moderne economie is Informatie Technologie niet meer weg te denken. Probeer je maar eens in te denken wat de economische schade zou zijn mochten voor een dag alle computers weigeren te werken.

Ondanks het belang is de hele IT-sector relatief gezien nog erg jong. Voor het grootste deel van zijn geschiedenis was de sector het domein van vooral technisch aangelegde enthousiastelingen die niet altijd even goed op de hoogte waren van hoe moderne dienstverlening moet georganiseerd worden. Enkele vooroordelen uit deze periode worden nog steeds door de sector meegesleept. De tijd van de bij wijze van spreken ongelimiteerde budgetten als het gaat om IT ligt ook al even achter de rug. De Dot-Com-bubbel van eind jaren '90 en de economische crisis eind jaren 2000 hebben er voor gezorgd dat ook de IT-sector niet ontsnapt aan besparingen en de drang naar grotere efficiëntie.

De organisatie van heel wat IT-bedrijven is echter niet mee gegroeid met deze ontwikkelingen. Enkele typische problemen in de sector: De helpdesk wordt nog steeds stiefmoederlijk behandeld, als een plaats om de minder getalenteerde IT-ers onder te brengen. Afdelingen communiceren onvoldoende met elkaar waardoor problemen niet snel genoeg worden opgelost, of het antwoord afhankelijk is van de persoon waarmee je contact hebt. Er is onvoldoende documentatie van de systemen waardoor enkel de mensen die het systeem mee hebben opgebouwd in staat zijn te antwoorden op bepaalde vragen wat uiteraard problemen met zich meebrengt als deze personen het bedrijf verlaten. Er is onvoldoende personeel op de Service Desk, en hun werk bestaat vooral uit het blussen van brandjes in plaats van er voor te zorgen dat toekomstige problemen vermeden worden.

Al vrij vroeg kwam men er achter dat als IT dezelfde kwaliteitsnormen moet halen als de rest van de dienstverlening, het nodig was om IT op dezelfde manier te managen als andere bedrijfstakken. **ITIL (Information Technology Infrastructure Library)** is ontstaan toen de Britse overheid een vaste werkwijze wou uitwerken om haar IT-projecten te organiseren. Het was de bedoeling om een reeks van richtlijnen uit te werken voor het managen van deze projecten, los van de technische inhoud. Deze richtlijnen hebben zich steeds verder ontwikkeld tot een reeks van **best practices**, gebaseerd op de praktijk die kunnen helpen om veel voorkomende problemen te vermijden. Ondertussen is ITIL meegegroeid met de IT-sector, de derde versie

van de verzameling best practices is in 2011 uitgegeven, en de principes worden wereldwijd toegepast in grote en kleine bedrijven uit de IT-sector.

1.1. Wat is ITIL?

Zoals gezegd is ITIL ontstaan in de schoot van de Britse overheid. Midden jaren '80 kwam men erachter dat het praktische onmogelijk was om de verschillende afdelingen, functies en producten op een efficiënt te beheren. Dit kostte de overheid handenvol geld en tijd. Om te proberen deze problemen op te lossen werd besloten om (in samenwerking met mensen van IBM) gesprekken te organiseren met een hele hoop bedrijven en organisaties binnen de IT-sector, publiek en privaat, groot en klein. De vraag was eenvoudig: "Hoe slagen jullie er in om jullie IT te organiseren?".

Het resultaat van deze gesprekken werd verzameld als een collectie van goed advies, of **best practices**, in 31 boeken die eind jaren '80 ook voor een breed publiek beschikbaar werden gemaakt. (Dit verklaart ook de Library in de afkorting). Deze boeken vormen samen wat later ITIL versie 1 (ITILv1) genoemd werd. Rond het jaar 2000 was het tijd voor een update. ITIL werd volledig herschreven, en ingekort, in 9 boeken. De focus bij ITILv2 lag vooral op een erg praktische benadering van de verschillende processen. In 2007 was het tijd voor versie 3. ITIL werd verder afgeslankt tot 5 boeken, opgebouwd rond de **IT-lifecycle**. In 2011 verscheen er een kleinere update, vooral om enkele onduidelijkheden op te helderen. Bovendien is men in 2011 afgestapt van de versie-nummers, en staat het framework gewoon bekend als ITIL zonder meer (Al zal je hier en daar ook ITIL2011 tegen komen.) Deze versie is voorlopig de laatste. Naast deze vijf boeken zijn er ook bijkomende uitgaves gericht op een specifieke sector, op bepaalde functies, op het integreren van ITIL met andere kwaliteitsprocessen, ... Deze bijkomende bronnen worden regelmatig ge-update, zodat de basisboeken niet elke paar jaar heruitgegeven moet worden.¹

¹ <https://www.axelos.com/best-practice-solutions/itil/what-is-itil> (De officiële site)



Figuur 1 - De ITIL boeken

1.2. De voordelen van ITIL

KOSTEN BESPAREN

Eén van de grote problemen binnen de IT-sector is dat het moeilijk is een bedrag te plakken op wat IT juist kost. Sommige kosten zijn onmiddellijk zichtbaar, zoals de aankoop van materiaal, de energie die nodig is om alles draaiende te houden,... een erg groot deel van de kosten binnen IT zijn echter **verborgen kosten**:

- **Downtime:** Een eerste en erg belangrijke verborgen kost is de kost van downtime. De tijd dat een werknemer niet productief kan zijn omdat de IT-infrastructuur niet bruikbaar is. In 2013 werd berekend dat bedrijven wereldwijd jaarlijks 3.2 miljard euro aan inkomsten misliepen door het uitliggen van ICT-infrastructuur. Uit Australisch onderzoek is gebleken dat werknemers per jaar gemiddeld twee werkweken niet productief kunnen zijn door problemen met IT. Het onderdeel dat hiervoor verantwoordelijk is **Availability Management**.

- **Onderhoud:** De aankoopprijs van een systeem blijkt in de praktijk maar een klein deel van de totale kostprijs te zijn. Buiten de aankoop zijn er kosten verbonden aan het beveiligen, updaten, herstellen van het systeem en uiteraard ook het personeel dat deze taken moet uitvoeren. Voor een gewone computer in een bedrijf wordt geschat dat 80% van de totale kosten gemaakt worden na de aankoop van de hardware en software
- **Training:** Het is uiteraard ook belangrijk dat de gebruikers met je nieuwe systeem kunnen werken. Deze kosten kunnen zowel formeel (de opleiding die je voorziet voor je werknemers) of informeel zijn (de tijd dat een werknemer niet aan het werk is omdat hij een collega aan het helpen is met een probleem).
- **De ervaring voor de eindgebruiker:** Een systeem dat is aangepast aan de workflow van de gemiddelde gebruiker zal een productiviteitsvoordeel opleveren omdat het werk sneller kan uitgevoerd worden, maar ook omdat het werk aangenamer is. (Een voorbeeld is bijvoorbeeld het vermijden van repetitieve handelingen).

Een belangrijk begrip binnen ITIL is **Total Cost of Ownership (TCO)**, de totale kostprijs van een systeem over de totale levensduur van een systeem. Zowel rechtstreekse kosten als onrechtstreekse kosten worden hierbij bekeken. Door te focussen op TCO is het voor het management makkelijker om de juiste beslissingen te nemen. Jammer genoeg wordt er in veel bedrijven nog steeds te veel aandacht besteed aan de aankoopprijs.

Een tweede manier waarop er in de sector veel geld verloren gaat is het niet afstemmen van de aanwezige IT-infrastructuur op de noden van het bedrijf. Dit kan in twee richtingen werken, ofwel is er onvoldoende capaciteit aanwezig, waardoor andere onderdelen van het bedrijf beknot worden in hun productiviteit, ofwel is er een teveel aan capaciteit waardoor de beperkte middelen niet efficiënt aangewend worden. Een heel onderdeel van ITIL (**Capacity Managment**) heeft als doel om de aanwezige infrastructuur zo kost-efficiënt mogelijk af te stemmen op wat nodig is voor een bedrijf.

PRODUCTIVITEIT

ITIL probeert binnen bedrijven die IT-diensten aanleveren er voor te zorgen dat de aanwezige middelen en mensen op een optimale wijze ingezet worden. Dit wordt onder meer mogelijk gemaakt door de rol en verantwoordelijkheden van de verschillende werknemers (of afdelingen) duidelijk af te lijnen zodat het voor iedereen duidelijk is wat zijn rol binnen het geheel is en dubbele verantwoordelijkheden vermeden worden.

Een onderdeel waar je de invloed van ITIL op de productiviteit binnen een bedrijf makkelijk zichtbaar kan maken is de Service Desk (Helpdesk). Eén bedrijf dat ITIL implementeerde binnen zijn organisatie meldde de volgende verbeteringen:

- Het aantal behandelde incidenten ging met 19% naar beneden.
- De behandeltijd van een probleem ging met 40% naar beneden.

In het geval van de Service Desk gaat het bijvoorbeeld over het toepassen van de aanbevelingen uit de onderdelen **Incident Management** en **Knowledge Management**.

IMPLEMENTATIE

ITIL is niet het enige framework dat probeert om IT-processen te stroomlijnen, maar het is ongetwijfeld wel het populairste. Eén van de grootste voordelen aan ITIL is dat het ondertussen al bijna drie decennia wordt toegepast in honderdduizenden bedrijven. Het systeem heeft zijn nut dus al bewezen. Een van de belangrijkste verwezenlijkingen van ITIL dat het gezorgd heeft voor een gemeenschappelijke woordenschat tussen al deze bedrijven wat de communicatie eenvoudiger maakt.

Eén van de redenen die er voor gezorgd heeft dat ITIL zo vlot verspreid werd is dat het volledig product-onafhankelijk is. ITIL is een set van boeken, geen software die geïmplementeerd moet worden. Of je bedrijf op Windows loopt, op Linux, of op oude Playstations, je kan ITIL implementeren. Daarnaast is ITIL niet opgevat al een set van wetten die je moet toepassen. ITIL is meer een set van aanbevelingen die je zonder problemen kan en mag aanpassen aan de dagelijkse realiteit van je bedrijf. Er is niemand die zal komen controleren of je alles correct hebt toegepast, ITIL is geen standaard, wel een framework. Je hoeft het ook niet volledig toe te passen, je kan a la carte de processen uitkiezen die voor jouw bedrijf zinnig zijn. ITIL is dus geen strak keurslijf,

1.3. De Valkuilen van ITIL

Ondanks alle voordelen is ITIL geen magische formule die van elk bedrijf een geoliede machine kan maken. Veel bedrijven hebben geprobeerd ITIL in de bedrijfsprocessen op te nemen, maar zijn op verschillende manieren mislukt in hun opzet. Enkele van de veel voorkomende problemen bij de implementatie

GEBREK AAN MOTIVATIE EN BETROKKENHEID VAN HET MANAGEMENT.

ITIL is niet iets dat enkel geïsoleerd binnen de IT-afdeling kan georganiseerd worden. Hoewel het zeker en vast mogelijk is om hier en daar winsten te boeken door zelf onderdelen toe te passen binnen een beperkte context vraagt ITIL een grote betrokkenheid van het management om alles te stroomlijnen.

ER WORDT TEVEEL TIJD BESTEED AAN HET MAKEN VAN THEORETISCHE MODELLEN EN DIAGRAMMEN.

De verleiding om ITIL te overdenken is groot, sommige bedrijven komen niet verder dan enkele mooie structuurdiagrammen en flowcharts. Sommige onderdelen van ITIL moeten echter meermaals per dag uitgevoerd worden, bijvoorbeeld Incident Management (het oplossen van problemen) zodat de rigide structuur eerder overkill is.

INSTRUCTIES WORDEN NIET NEERGESCHREVEN OF UP TO DATE GEHOUDEN.

ITIL heeft pas echt nut als iedereen weet wat juist zijn verantwoordelijkheden zijn, en wat er in een bepaalde situatie moet gebeuren. Je kan mooie schema's maken met verschillende escalatie-niveaus, maar als er voor de Service Desk geen richtlijnen zijn uitgeschreven om te bepalen welke situatie kritiek is, of eerder een onbelangrijk probleem hebben deze categorieën weinig nut.

HET BEDRIJF IS TE AMBITIEUS.

ITIL beslaat de volledige dienstverlening van een IT-bedrijf. Sommige bedrijven proberen dan ook in één keer alle processen volgens ITIL te organiseren, wat leidt tot verwarring en een te grote impact op de dienstverlening van het bedrijf tijdens de overgangsperiode. Het is beter om ITIL geleidelijk in te voeren, zodat je kan leren uit de ervaring.

HET BEDRIJF VERLIEST MOMENTUM

De gemiddelde implementatie van ITIL binnen een bedrijf duurt tussen de drie en de vijf jaar. Zeker als de grootste winsten in het begin van het proces zitten is het mogelijk dat het management het nut niet ziet in het volledige proces te doorlopen. Bovendien hebben mensen de neiging te hervallen in de oude slechte gewoonten eens het initiële enthousiasme wegzakt.

1.4. Certificaten

ITIL is eigendom van één bedrijf, AXELOS (deels eigendom van de Britse overheid) dat verantwoordelijk is voor het uitgeven van ITIL. Dit bedrijf stelt ook de regels op omtrent het uitreiken van ITIL-certificaten. Op dit moment zijn er vier niveaus

ITIL FOUNDATION

Het populairste certificaat is ITIL Foundation is het beginnerscertificaat. Dit certificaat is bedoeld voor iedereen die op de hoogte moet zijn van de basisprincipes van ITIL, bijvoorbeeld omdat het bedrijf waar hij werkt ITIL implementeert. Dit is het enige certificaat dat je in zelfstudie kan voorbereiden, en is toegankelijk voor iedereen. Het examen bestaat uit 40 vragen, om te slagen moet je 65% behalen. Je kan dit examen onder andere afleggen bij Oak3 op de Corda-campus in Hasselt

ITIL PRACTITIONER

ITIL Practitioner is een nieuw niveau tussen Foundation en Intermediate. Bedoeling is dat het vooral focust op de praktische implementatie van ITIL binnen een bedrijfsomgeving. ITIL Practitioner is geen vereist certificaat voor de ITIL Intermediate cursussen.

ITIL INTERMEDIATE

ITIL Intermediate zijn cursussen en certificaten die focussen op de verschillende deelaspecten van ITIL (1 voor elk boek), plus enkele meer implementatiegerichte cursussen Elke van deze cursussen levert drie studiepunten op (Net zoals Foundation twee studiepunten oplevert.) Deze examens worden enkel aangeboden na het volgen van een cursus.

ITIL EXPERT

Om deel te nemen aan het ITIL expert examen moet je minstens 17 studiepunten behaald hebben in eerdere examens. Met het ITIL-expert examen bewijs je dat je voldoende kennis hebt van de verschillende deelaspecten dat je een volledige implementatie kan toepassen, en dus dat je een grondige kennis hebt van het gehele proces.

ITIL MASTER

ITIL Master is het laatste, en het zeldzaamste, certificaat. Je kan het examen voor ITIL master best vergelijken met een eindwerk aan het einde van een opleiding. Het is de bedoeling dat je de toepassing van ITIL die je in je bedrijf hebt begeleid, volledig toelicht en verdedigt.

HOOFDSTUK 2: DE WOORDENSCHAT VAN ITIL

Een van de grote voordelen van ITIL is dat het zorgt voor een gedeelde woordenschat die het mogelijk maakt om te communiceren over IT-dienstverlening. Door termen consequent te gebruiken en exact te definiëren wordt het voor IT-ers makkelijker om over de dienstverlening te praten zodat het voor iedereen die op de hoogte is van ITIL duidelijk is wat er bedoeld wordt.

Zeker in het begin kan het wat weg hebben van een stortvloed aan termen. Sommige termen zijn misschien nieuw voor jou, andere ken je misschien maar worden binnen ITIL anders ingevuld.

2.1. Een Service (Een Dienst)

Volgens ITIL:

Een manier om waarde te leveren aan klanten door het faciliteren van uitkomsten die klanten willen realiseren zonder dat zij de specifieke kosten en risico's moeten dragen die voortkomen uit de bedrijfsvoering van de service verlening.

Een service is de dienst die je een klant (extern of intern) aanbiedt. Of het nu gaat om iets tastbaar (bvb. computers, software) of niet-tastbaar (bvb. internettoegang). De klant wil een bepaald product, en is niet geïnteresseerd hoe jij dat mogelijk maakt. Ze zijn geïnteresseerd in het resultaat, niet het proces.

Om het met een voorbeeld van buiten de IT-wereld te illustreren: Als ik op restaurant ga dan krijg ik een menu met daarop de prijs voor een pizza. Op die manier is er een duidelijke afspraak. Ik betaal 15 euro en in ruil daarvoor krijg ik een pizza. Om van mijn pizza te genieten hoef ik niet te weten welke contracten het restaurant heeft met leveranciers, op welke temperatuur de oven draait, net zomin als een afnemer van een IT-dienst wil weten welke routers je gebruikt om de dienst tot bij hem te krijgen. Uiteraard zijn de kosten van het personeel, de brandverzekering en dergelijke in de prijs van mijn pizza opgenomen, maar ik hoef de details hiervan niet te weten.

ITIL onderscheidt drie soorten diensten:

Een **kerndienst (core service)** is de basisfunctionaliteit die een klant vraagt. Een klant wil bijvoorbeeld werken met e-mail.

Een **enabling service**² is een dienst die het leveren van een kerndienst mogelijk maakt. Mogelijk is deze dienst onzichtbaar voor de eindgebruiker. Wanneer het gaat over e-mail is dit bijvoorbeeld alles wat te maken heeft met beveiliging, of het programma dat gebruikt wordt om de e-mails te openen.

Een **enhancing service** is een dienst die toegevoegd wordt aan de kerndienst om die aantrekkelijker te maken voor de eindgebruiker. Enhancing services zijn niet noodzakelijk, maar worden toegevoegd bijvoorbeeld om een onderscheid te maken met eenzelfde kerndienst van een concurrent. In het geval van e-mail zou het bijvoorbeeld kunnen gaan over een archieffunctie binnen de cloud.

Om het onderscheid duidelijker te maken een voorbeeld van buiten de IT: Een hotel heeft als kerntaak het leveren van een bed en een badkamer. Om dit op een goede manier te kunnen leveren is er een kuisploeg nodig (een enabling service). Om er voor te zorgen dat je voor dit hotel kiest worden extra's aangeboden: gratis Wifi of een ontbijt (de enhancing services)

2.2 Service Management (dienstenbeheer)

Volgens ITIL

Het geheel van gespecialiseerde organisatorische vaardigheden voor het leveren van waarde aan klanten in de vorm van diensten.

Als services zijn wat je aan klanten levert dan is Service Management het geheel van kennis, infrastructuur, organisatie die in je bedrijf of in je IT-afdeling voorhanden zijn om het mogelijk te maken deze diensten aan je klanten aan te bieden. Het doel van Service Management is een goede verstandhouding garanderen tussen leverancier en klant door het **beheren van de verwachtingen** van de klant en het leveren en ondersteunen van de gewenste resultaten.

Het beheren van de verwachtingen is in dit proces een cruciaal onderdeel. Het is erg belangrijk dat zowel de klant als de leverancier duidelijk zijn over wat er verwacht wordt voor de afgesproken prijs. Als het bijvoorbeeld niet duidelijk is voor de klant of er al dan niet

² Deze term is door ITIL niet vertaald

ondersteuning inbegrepen is bij de levering van een dienst kan dat heel wat irritatie veroorzaken. Een goed voorbeeld hiervan zijn kleine zelfstandigen die op persoonlijke naam een internetlijn hebben. De meeste providers bieden enkel een herstelling binnen de 24 uur aan voor professionele klanten met een duurder abonnement. Het is dan ook belangrijk dat een klant goed geïnformeerd wordt bij het afsluiten van een contract, iets wat al eens over het hoofd gezien wordt.

In zijn meest elementaire vorm is Service Management eigenlijk heel eenvoudig, en als elk IT-bedrijf deze eenvoudige stappen zou volgen zou er al heel wat minder frustratie bestaan in de sector:

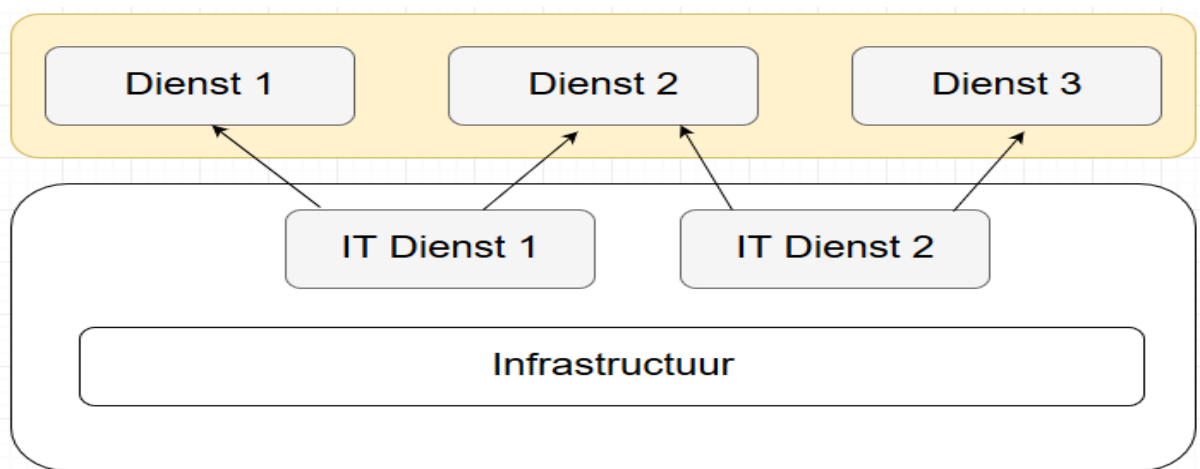
1. **Spreek af wat** je gaat doen
2. **Lever** wat je hebt afgesproken
3. **Bewijs** dat je hebt geleverd wat je hebt afgesproken.

Eigenlijk is alles wat in deze cursus volgt terug te brengen op één van deze drie stappen.

2.3. IT-Service Management

Volgens ITIL

Implementatie en beheer van kwalitatieve IT dienstverlening die voldoet aan de behoeften van de bedrijfsvoering. IT-servicemanagement wordt uitgevoerd door IT-service providers in een juiste combinatie van mensen, processen en IT



Om de rol van IT-service management te begrijpen is het handig om te kijken naar de plaats van IT binnen een bedrijf. Een bedrijf (geel) biedt verschillende diensten aan, deze diensten zijn de basistaak van het bedrijf. Een bank bijvoorbeeld biedt leningen aan. Hiervoor wordt kredietwaardigheid van een klant gecontroleerd (Dienst 1), de lening wordt goedgekeurd en geregistreerd (Dienst 2), de betaling van de lening wordt opgevolgd (Dienst 3). De bank doet dit al jaren voor de opkomst van de IT, en het proces is sinds de komst van de IT niet spectaculair gewijzigd. Om de taken van de bank te ondersteunen biedt de IT-afdeling bepaalde diensten aan de werknemers van de bank, bijvoorbeeld internettoegang, of een databank met klantgegevens. Opnieuw heeft het personeel geen interesse in hoe deze diensten worden aangeboden, enkel dat ze er zijn en werken.

IT Service management, en dus ook waar ITIL om draait, is de IT-afdeling zo organiseren dat het de taak van het leveren van IT-diensten aan de rest van het bedrijf zo goed mogelijk uitvoert rekening houdend met de vereisten van het bedrijf. Dat is het doel, ITIL is slechts een hulpmiddel om dit doel te bereiken.

2.4. Wie is Wie?

Wanneer het gaat om wie er betrokken is bij het leveren van een bepaalde dienst onderscheid ITIL de volgende personen:

Een eerste belangrijke persoon is **de klant (customer)**. In ITIL is de klant de persoon (of het bedrijf) dat betaald voor een dienst of product. Deze klant kan zowel intern zijn (een andere afdeling binnen hetzelfde bedrijf) of extern zijn (een ander bedrijf of een particulier). Dit zijn de mensen die de autoriteit hebben om beslissingen te nemen over contracten, financiën,...

De **gebruiker (user)** van een dienst is iemand die met de geleverde dienst werkt. Het is belangrijk om een duidelijk onderscheid te maken tussen de klant enerzijds en de eindgebruiker anderzijds. Om opnieuw naar het voorbeeld van een bank te gaan: De bank zelf is de klant: Zij hebben een contract afgesloten om bepaalde diensten te ontvangen in ruil voor een bepaalde prijs. De persoon die achter het loket met de diensten werkt is de gebruiker, hij heeft geen beslissingsrecht over de voorwaarden van de IT-dienst. Het is mogelijk dat de eindgebruikers betrokken worden bij het opstellen van de vereisten voor de dienst, maar ze hebben enkel een adviserende rol. Het is de directie die de uiteindelijke beslissing zal nemen. Klant en gebruiker worden vaak door elkaar gebruikt, maar binnen ITIL zijn het twee heel verschillende concepten.

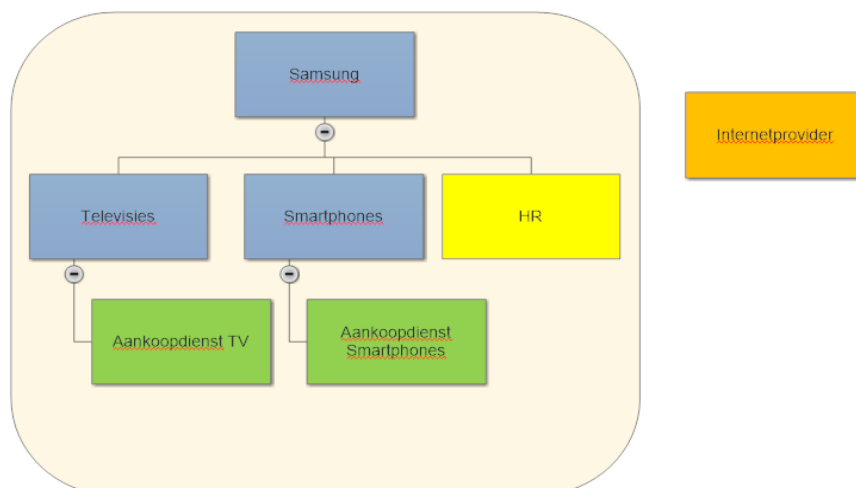
Naast klanten en gebruikers moet je ook rekening houden met **leveranciers (suppliers) of dienstverleners**. Dit zijn derden verantwoordelijk voor het leveren van goederen of diensten, dingen die je nodig hebt om een dienst te kunnen leveren.

Er zijn drie verschillende soorten leveranciers:

Een eerste type zijn de **interne dienstverleners**. Deze zijn nauw verbonden met een bepaalde afdeling van een bedrijf. In het volgende schema zijn de aankoopdiensten voor de afdelingen Televisies en Smartphones interne dienstverleners: Zij leveren enkel diensten aan hun respectievelijke afdelingen, de afdeling smartphones maakt geen gebruik van de afdeling televisies en omgekeerd.

Het tweede type is de **gedeelde service-eenheid**. Dit zijn de dienstverleners binnen het bedrijf die hun diensten leveren aan de verschillende afdelingen van het bedrijf. Een voorbeeld hiervan is de HR-afdeling. Ze staan los van de verschillende afdelingen en hebben een overkoepelende functie.

Een laatste type zijn de **externe dienstverleners**. Dit zijn de dienstverleners die geen onderdeel uitmaken van het bedrijf zelf maar wel een dienst leveren aan het bedrijf. In het schema gaat het bijvoorbeeld om een internetprovider. Je bedrijf is vaak maar een van de vele klanten van de externe dienstverlener.

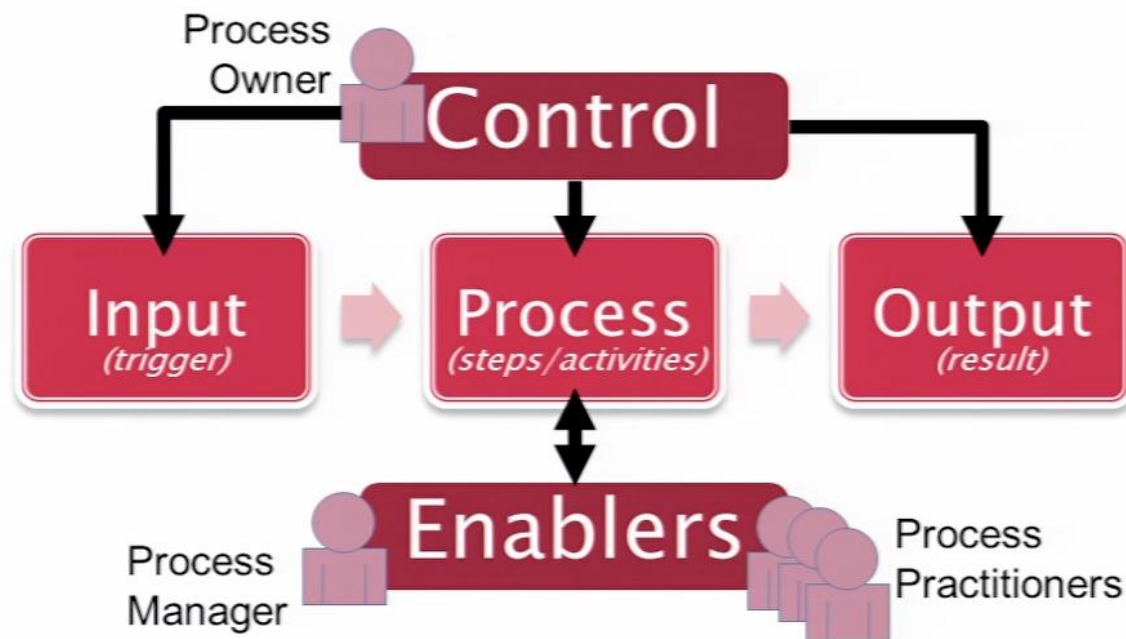


Afhankelijk van hoe een bedrijf is opgebouwd kan IT elk van deze drie plaatsen innemen. Gekoppeld aan een bepaalde afdeling, of gedeeld door het hele bedrijf, maar ook extern (wanneer er bijvoorbeeld met consultants wordt gewerkt.)

De **stakeholders** zijn de personen of bedrijven die interesse hebben in de dienst en hoe die geleverd wordt. Eigenlijk valt iedereen hieronder. Mensen binnen je eigen bedrijf, mensen binnen het klant-bedrijf of eindgebruikers.

2.5. Processen

ITIL is opgebouwd uit verschillende processen, 26 in totaal. Om er voor te zorgen dat ITIL universeel toepasbaar is in de IT zijn de processen “high level” gehouden, verwacht dus geen tot in het kleinste detail uitgeschreven stappenplannen, wel algemene aanbevelingen, zodat je ze zelf kan invullen naargelang de behoeften van je bedrijf.



In zijn meest eenvoudige vorm is een proces een reeks van stappen die reageren op een bepaalde trigger die er voor zorgt dat proces gestart wordt. Wanneer het proces doorlopen is, is er een bepaald resultaat.

Een voorbeeld:

Er wordt iemand nieuw aangenomen in het bedrijf (trigger)

1. Aankoopdienst bestelt een nieuwe laptop
2. Onderhoud maakt een bureau beschikbaar
3. Telecomdienst installeert een telefoonlijn en koppelt een telefoonnummer
4. Serverteam maakt een account en een e-mailadres aan
5. Netwerkteam activeert een netwerkpoort op het kantoor
6. Deskopteam configureert de laptop wanneer die arriveert
7. Deskopteam levert de laptop aan het nieuwe personeelslid op de eerste dag

Het nieuwe personeelslid kan op tijd beginnen (resultaat).

Elk proces moet de volgende eigenschappen hebben:

- Het proces moet meetbaar zijn. Het moet mogelijk zijn om te controleren of het proces goed werkt of niet.
- Een proces heeft een specifiek resultaat, het doel van het proces.
- Dit resultaat is van belang voor een stakeholder of klant.
- Een proces is een reactie op een bepaalde gebeurtenis
- Het proces moet gedocumenteerd zijn (hier wordt het vaakst tegen gezondigd.)

DE PROCESS OWNER

Om het proces in goede banen te leiden is er de **Process Owner**. De Process Owner is de persoon die het proces uitgeschreven heeft, na overleg met de stakeholders. Je kan de taken van de process owner als volgt beschrijven:

- De Process Owner **ontwerpt en documenteert** het proces.
- Hij **verbetert** het proces zodat het proces blijft aansluiten op de noden van het bedrijf
- Hij **meet** de resultaten van het proces en kijkt of deze in de lijn liggen van de verwachte resultaten.
- Hij geeft **training** aan het personeel betrokken bij het proces en zorgt ervoor dat de richtlijnen worden nagevolgd

Het uitvoeren en het resultaat van het proces is de **eindverantwoordelijkheid (accountability)** van de process owner.

Op een bepaald moment is er beslist om een proces te implementeren voor wat er moet gebeuren wanneer er een nieuwe werknemer in dienst komt. Iemand is aangesteld als process owner en overlegt met de verschillende stakeholders (de verschillende diensten, management) wat nodig is, wat mogelijk is, en wat een aanvaardbare termijn is voor het proces. Hij schrijft het proces uit in de verschillende stappen die ondernomen moeten worden. Wanneer de omstandigheden wijzigen omdat er bijvoorbeeld nieuwe software gebruikt wordt past de owner het proces aan zodat dit up-to-date blijft met de nieuwe omstandigheden. De owner meet de resultaten (hoeveel nieuwe werknemers zijn er op deze manier geïnstalleerd? Hoeveel keer was het bedrijf niet klaar op de eerste dag van de werknemer? Wat waren de oorzaken?) en verfijnt het proces indien nodig om toekomstige resultaten te verbeteren. Hij zorgt ervoor dat iedereen weet wat zijn rol is in het proces en zorgt ervoor dat iedereen zijn onderdeel volgens de regels uitvoert.

DE ENABLERS

Enablers zijn de personen die er voor zorgen dat het proces uitgevoerd kan worden. Aan de ene kant heb je de **Process Practitioners**: dit zijn de effectieve uitvoerders van een proces. Het personeel van de verschillende diensten die de verschillende stappen uitvoeren. De **Process Managers** zijn de personen die verantwoordelijk zijn voor de individuele Process Practitioners (bijvoorbeeld het hoofd van de Desktop-dienst).

In een groot bedrijf is het mogelijk dat al deze rollen ingevuld worden door een apart personeelslid. In een kleiner bedrijf zal er vaker overlap zijn tussen de verschillende rollen. Het is perfect mogelijk dat de Process Owner ook hoofd is van de Server-dienst, en het hoofd van de Telecom-dienst ook degene is die de telefoon installeert. Het is wel belangrijk dat in dat geval er nog altijd een onderscheid wordt gemaakt tussen de taken van een persoon als Process Owner, en zijn taken als Process Manager.

2.6. Functies vs. Rollen

Functies binnen ITIL zijn groepen binnen het bedrijf die gespecialiseerd zijn in een bepaald soort taken. In een groot bedrijf gaat het bijvoorbeeld over de afdeling die verantwoordelijk is voor het server beheer (Active Directory, e-mail, netwerkshares, ...). Inbegrepen in deze functies zijn ook de verschillende werktuigen die het team gebruikt om zijn functie uit te voeren. De Service Desk is de naam van de functie die instaat voor de ondersteuning van de gebruikers (de help desk), een programma dat een eindgebruiker gebruikt om zelf problemen op te lossen, of de status van een gemeld probleem te bekijken is ook een onderdeel van deze Service Desk functie. In grote lijnen komt de functie van een personeelslid overeen met wat er op zijn naamkaartje staat. In de meeste gevallen kan je functie vervangen door team.

Rollen zijn in ITIL erg specifiek: het is een verzameling van verantwoordelijkheden die in een proces gedefinieerd zijn, en aan een bepaalde persoon of een bepaald team toegewezen zijn. Rollen zijn dus afhankelijk van het specifieke proces dat op dat moment uitgevoerd wordt. Rollen zijn dus geen functiebeschrijving.

Een personeelslid is een computer techniekier binnen het desktop-team. Zijn functie is computer techniekier, en deze functie wijzigt normaal niet zolang hij niet wordt overgeplaatst naar een ander team. In de loop van een werkdag heeft hij echter verschillende taken: Wanneer hij een gebruiker helpt met een computerprobleem neemt hij een rol op binnen het Incident Management proces. Wanneer hij een half uur later een nieuwe printer installeert

heeft hij een rol binnen het Release & Deployment Management proces. In de loop van de dag zal hij zo verschillende rollen uitvoeren, maar zijn functie blijft ongewijzigd.

In ITIL worden vier functies expliciet gedefinieerd:

- **De Service Desk** lost problemen van de eindgebruikers op, maakt tickets aan, ... Dit team wordt ook vaak de helpdesk genoemd
- **IT Operations Control** werkt achter de schermen aan de IT-infrastructuur. Enkele belangrijke taken zijn het monitoren van het netwerk (**IT Operations Control**) en het beheren van de fysieke IT-infrastructuur (**Facilities Management**) zoals het beheren van serverlokalen, aanleggen en onderhouden van bekabeling, ...
- **Technical Management** is een ruime groep die onder andere verantwoordelijk is voor de configuratie van servers, instellen van routers, ... Waar IT Operations eerder het grote plaatje bekijkt is Technical Management meer gefocust op configuratie.
- **Application Management** zijn de mensen die bezig zijn met de applicaties van een bedrijf: Ze ontwerpen, implementeren, testen en installeren de gebruikte software. Programmeurs zijn typisch een onderdeel van Application Management.

Uiteraard is het mogelijk om deze groepen verder onder te verdelen in verschillende teams. Het zijn slechts grote, relatief vage omschrijvingen om er voor te zorgen dat ITIL universeel toepasbaar is. De eigenlijke structuur en benaming is erg bedrijfsafhankelijk. Een klein bedrijf zal typisch geen aparte afdeling hebben voor de monitoring van het netwerk bijvoorbeeld.

2.6. Het RACI-model

Veel processen lopen in het honderd omdat het niet volledig duidelijk is wie welke verantwoordelijkheid heeft. Om dit op te lossen is het aangeraden om voor elke proces ook een **RACI-model (of RACI-chart)** te maken. Dit is een verantwoordelijkheid van de proceseigenaar.

Een RACI-model is een erg eenvoudig hulpmiddel. Er wordt een matrix opgebouwd, met op de verticale as de verschillende deelstappen in een proces, en op de Y-as de personen of teams die een verantwoordelijkheid hebben binnen een proces. De matrix wordt ingevuld met vier letters om aan te geven wat iedereens rol is binnen elke stap:

Responsable (R) dient om aan te geven wie er uitvoerder van een bepaalde (deel)taak is. Meerdere afdelingen of personen kunnen uitvoerder zijn binnen dezelfde stap, er moet minstens één uitvoerder binnen een stap zijn.

Accountable (A) bepaald wie er de eindverantwoordelijkheid draagt voor deze stap. Dit hoeft niet altijd dezelfde persoon te zijn die de stap uitvoert. Per stap mag er maar één eindverantwoordelijke zijn.

Consult (C) zijn de personen waarmee overlegd moet worden om de taak uit te voeren.

Inform (I) zijn de personen die op de hoogte gehouden moeten worden.

Process Name / Description:	Installeren nieuwe werknemer	
Created On:	1-Jan-16	Revision: 12/03/2016
Created by:	Bram Verhulst	

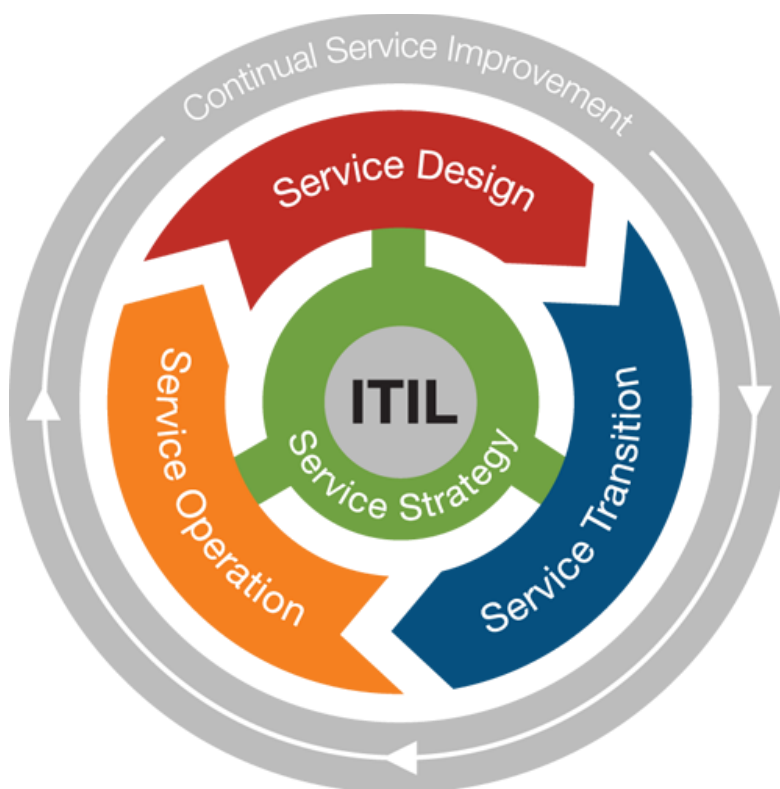
	Aankoop	Facilities	Network	Server	Desktop
Bestel een nieuwe laptop	A R	I	-	-	R C I
Maak een bureau klaar	-	A 'R	I C	-	I
Zorg voor een netwerkvervinding	-	I C	A R	I	-
Maak een account en emailadres	-	-	-	A 'R	I
Configureer de Laptop	-	I	-	I	A R
Lever de nieuwe laptop af	-	-	-	-	A R
R = Responsible, A = Accountable, C = Consulted, I = Informed					

Neem bijvoorbeeld de eerste deeltaak: Het desktopteam stelt de systeemeisen op waaraan de laptop moet voldoen. Het aankoopteam bekijkt de mogelijkheden en consulteert met het Desktopteam over de uiteindelijk te bestellen laptop, en plaatst de bestelling. Het aankoopteam is verantwoordelijk voor deze taak, dus zij moeten er voor zorgen dat ze op tijd de nodige info van het laptopteam krijgen. Aankoop informeert het Desktopteam en Facilities over de verwachte leverdatum van de laptop.

Omdat Aankoop de enige eindverantwoordelijke is voor deze stap kan er niet met vingers gewezen worden wanneer deze stap niet op tijd afgerond is. Hierom is het belangrijk dat er per onderdeel maar één eindverantwoordelijke is.

2.7. Het Service Lifecycle Model

Nieuw in versie 3 van ITIL was het Service Lifecycle Model. Via dit model wou men proberen om de verschillende processen van ITIL binnen een logische geheel onder te brengen. Als basis wordt de levensloop van IT-service genomen. grofweg de planning, het design, de implementatie en het in dienst zijn van een IT-dienst. Allesomvattend is er ook de continue verbetering van de dienstverlening, los van individuele projecten. Deze vijf onderdelen komen overeen met de vijf boeken van ITIL.



Boek 1: Service Strategy, gaat over de planning en strategie: Welke diensten verwachten de klanten (of kunnen binnen een bedrijf geïmplementeerd worden om de rest van het bedrijf te ondersteunen), wat zijn de financiële mogelijkheden, welke capaciteiten moeten ontwikkeld worden, ... Het is de bedoeling om het bedrijf op een strategische manier over de toekomst te laten denken, en moet helpen om onderbouwde beslissingen te nemen. Omdat het een grote invloed heeft op de rest van de onderdelen staat het in dit schema centraal. Het bepaalt **waarom** we de andere onderdelen uitvoeren.

Boek 2: Service Design gaat over het ontwerpen van een service die voldoet aan de vereisten die we in het Service Strategy gedeelte hebben vastgelegd.

Boek 3: Service Transition neemt het design uit de vorige stap en zorgt er voor dat er een werkende oplossing komt. Dit gaat over het bouwen van de service, maar ook testen en uitrollen.

Boek 4: Service Operation is voor de klanten het belangrijkste onderdeel. De oplossing is geïmplementeerd en uitgerold, en in gebruik. In dit onderdeel van ITIL gaat het vooral over hoe er bijvoorbeeld gereageerd moet worden op problemen, hoe er wijzigingen moeten worden aangebracht worden, wie er toegang moet krijgen tot de dienst.

Boek 5: Continual Service Improvement is vooral bezig met het verbeteren van de bestaande dienstverlening, door de resultaten te bekijken en te evalueren, te leren uit fouten en de dienstverlening efficiënter te maken.

Elk van deze boeken is verder onderverdeeld in processen (27 in totaal) die specifieke aspecten van het onderdeel beschrijven.

Hoewel elk proces maar in één van de boeken voorkomt is kunnen processen uit één fase een invloed hebben op andere processen uit andere fasen. Het gaat dan vooral over de processen uit boek 1 en 5 die samen de **Service Lifecycle Governance Processes** genoemd worden, de andere drie boeken vallen onder **Operational Service Lifecycle Processes**. Boek 1 en 5 zijn eerder strategisch van aard, terwijl 2, 3 en 4 praktischer zijn. Het is logisch dat bijvoorbeeld Financial Service Management (alles wat met geld te maken heeft) een invloed heeft gedurende de hele looptijd van een proces.

EEN VOORBEELD

Neem bijvoorbeeld een bedrijf dat een e-mailsysteem wil uitwerken:

Fase 1: Service Strategy

Hier worden verschillende vragen gesteld: Heeft het bedrijf nood aan een e-mailsysteem? Hoeveel budget is er beschikbaar? Hoeveel gebruikers moeten gebruik maken van het systeem?,...

Fase 2: Service Design

Het technisch personeel neemt de vereisten uit de eerste fase en bekijkt hoe ze dit technisch kunnen verwezenlijken: Gebruiken we Exchange of gaan we voor Lotus Notes, hoeveel servers hebben we nodig, willen we fysieke of virtuele servers, hoe verbinden we met het internet? Hoe moeten de e-mailadressen er uit zien?,...

Fase 3: Service Transition

Hier wordt het design van het systeem geïmplementeerd. De configuratie wordt uitgevoerd, de servers worden geïnstalleerd, Outlook wordt uitgerold naar de computers van de gebruikers, alles wordt getest, ...

Fase 4: Service Operation

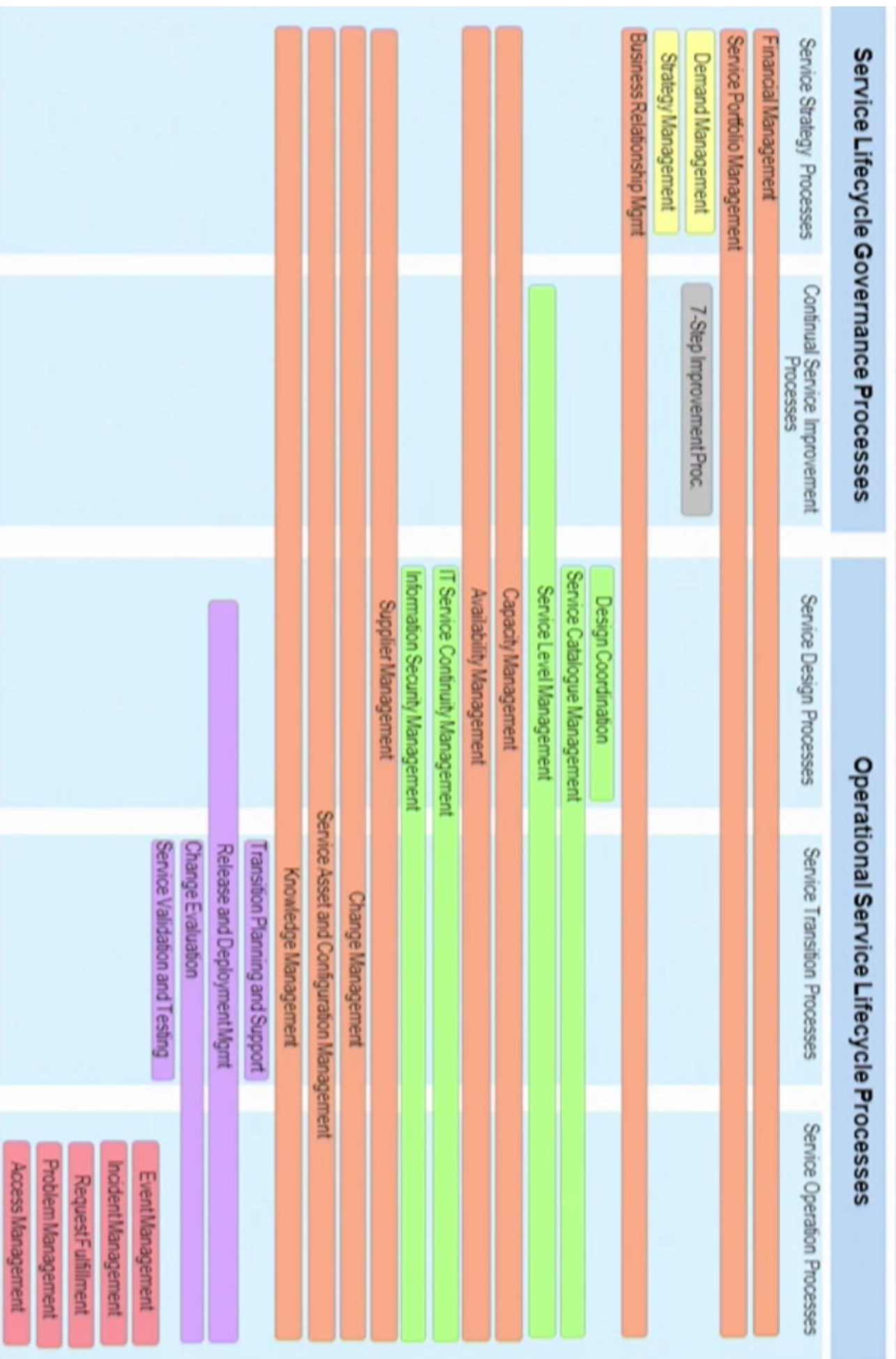
Het systeem werkt en is in gebruik. In deze fase worden problemen opgelost die tijdens het in gebruik zijn voorkomen, downtime wordt zoveel mogelijk vermeden, ...

Fase 5: Continual Service Improvement

In deze fase worden metingen uitgevoerd om te controleren of het systeem werkt zoals verwacht, en blijft presteren onder eventueel wijzigende omstandigheden.

Wanneer blijkt dat het systeem niet meer voldoet aan de eisen en er toevoegingen moeten gebeuren kom je in de fase terecht en kan de hele cyclus opnieuw plaatsvinden om de uitbreidingen te implementeren.

Service Lifecycle Processes



Service Strategy

- Strategy management for IT services
- Demand management
- Service portfolio management
- Business relationship management
- Financial management for IT services

Service Design

- Design coordination
- Service catalogue management
- Service level management
- Supplier management
- Availability management
- Capacity management
- IT service continuity management
- Information security management

Service Transition

- Transition planning and support
- Service asset and configuration management
- Change management
- Change evaluation
- Release and deployment management
- Service validation and testing
- Knowledge management

Service Operation

- Event management
- Incident management

- Problem management
- Request fulfillment
- Access management

Continual Service Improvement

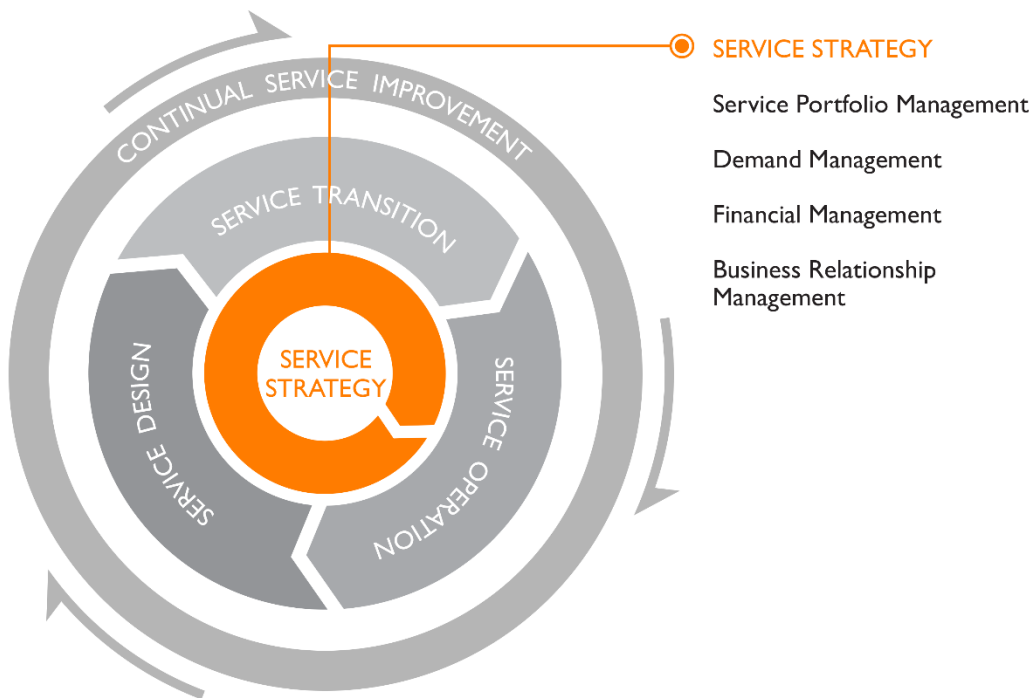
- The seven-step improvement process

HOOFDSTUK 3: SERVICE STRATEGY

Service Strategy is het eerste ITIL boek. Doel van Service Strategy is een antwoord te vinden op de volgende vragen:

- Welke diensten moeten we aanbieden, en aan wie?
- Hoe onderscheiden we ons van onze concurrenten?
- Hoe zorgen we voor echte meerwaarde voor onze klanten?
- Hoe moeten we strategische investeringen aanpakken?
- Hoe bepalen we de kwaliteit van een dienst?
- Hoe verdelen we middelen efficiënt over verschillende diensten?
- Hoe lossen we competitie op tussen verschillende diensten voor schaarse middelen?

Zoals je merkt houdt dit gedeelte van ITIL zich vooral bezig met de overkoepelende strategie. Daarom is dit ook het onderdeel waar je als beginnend ICT'er het minst mee in aanraking zal komen, zeker in grote bedrijven. Toch zijn er enkele belangrijke punten uit te halen die ook binnen een klein bedrijf, of als zelfstandige de moeite waard zijn om te bekijken. Ook als gewone werknemer binnen een IT-afdeling is het niet slecht om je deze vragen af en toe te stellen: Als interne IT'er kost je waarschijnlijk meer dan een externe consultant, hoe kan ervoor zorgen dat jouw functie dus een toegevoegde waarde is voor het bedrijf?



Omdat veel onderwerpen uit dit gedeelte een grote invloed hebben op de andere onderdelen, staat Service Strategy centraal in het hele ITIL-proces. Zo zullen de keuzes die gemaakt worden over het toewijzen van middelen een grote rol spelen bij het implementeren en draaiende houden van het project. Demand Management dat probeert de pieken en dal in gebruik te voorspellen kan belangrijk zijn om in de designfase de juiste keuzes te maken.

In Service Strategy draait het om **waarom** bepaalde dingen wil doen, terwijl de latere fasen vooral het **hoe** behandelen. In de ITIL-boeken worden voor Service Strategy de volgende doelen vermeld:

- Een begrip voorzien over wat de de strategie is.
- Zorgen voor een goede relatie tussen klanten en de IT-dienstverlener.
- Definiëren hoe waarde wordt gecreëerd.
- Bepalen welke diensten worden aangeboden en aan wie.

3.1. Het begrip waarde

Volgens ITIL's definitie is het hele punt van Service Management het leveren van waarde aan klanten onder de vorm van diensten. Voor we naar de processen van Service Strategy gaan kijken is het dus geen slecht idee om eens te bekijken wat waarde juist inhoudt. Het eerste dat bij mensen opkomt als er over de waarde van iets gesproken wordt is vaak de kostprijs van het product. Toch zijn er heel wat meer dingen die een rol spelen bij het bepalen van de

waarde. Niemand kijkt enkel naar kostprijs wanneer het gaat om de aankoop van een product, anders zouden we allemaal rond rijden met dezelfde wagen aangezien de goedkoopste ons normaal gezien even goed van punt A naar punt B kan brengen dan een Porsche.

De waarde van een product wordt bepaald door de **verwachtingen van de individuele klant**. Als ik McDonalds binnenwandelen verwacht ik snel en relatief goedkoop iets te eten. Als ik met mijn schoonouders ga eten dan ben ik bereid om meer te betalen voor goed restaurant. In een fastfood restaurant verwacht ik iets minder kwalitatief, mocht een mij goed restaurant een Big Mac voorschotelen dan ben ik minder tevreden. Je verwacht niet dezelfde kwaliteit van een Medion van 400 euro als van een high-end gaming laptop, maar afhankelijk van wat ik wil of verwacht van de laptop kan de Medion de betere keuze zijn. Ook **de voorkeuren van de klant** bepalen de waarde van een product voor die klant: Een vegetariër zal niet dezelfde waarde hechten aan een all-you-can-eat Barbecue buffet als iemand die wel vlees eet. Tenslotte spelen ook **eerdere ervaringen** van de klant spelen een rol: Wanneer ik al eerder problemen heb gehad met een bepaald merk van laptop zal ik minder geneigd zijn dit zelfde merk opnieuw te kopen. Deze drie zaken zorgen ervoor dat iemand een **oordeel** kan vellen over de waarde van een bepaalde dienst.

Er zijn twee factoren die de waarde van een dienst of een product bepalen: de **“utility” (nut)** van een dienst, en de **“waranty” (waarborg)**. Het nut van een dienst in ITIL gaat over het feit dat de dienst de eigenschappen heeft die nodig zijn om een positief effect op de handelingen en taken die nodig zijn om het gevraagde oplossing te bereiken verbeterd, of eenvoudiger gezegd: **Is het geschikt voor het doel?** Wanneer ik een nieuwe koelkast nodig heb moet ik met andere woorden niet thuiskomen met een diepvriezer, ook niet als de diepvriezer goedkoper is. Waranty draait om het feit of ik de dienst kan gebruiken wanneer ik die nodig heb, of er voldoende capaciteit beschikbaar is, en of het gebruik veilig, en langdurig is. **Is het geschikt om te gebruiken?** Een koelkast die niet 24 uur op 24 uur bruikbaar is, of te klein is voor wat ik er wil insteken heeft minder waarde voor mij dan één die wel groot genoeg is en continu beschikbaar is.

Een bedrijf of dienst kan waarde creëren door zijn **mogelijkheden (capabilities)** en **hulpbronnen (resources)** te combineren. Hulpbronnen zijn tastbaar, het gaat over dingen als kapitaal, infrastructuur, software, ... die je als je genoeg kapitaal hebt eenvoudig kan opbouwen. Mogelijkheden zijn moeilijker te verkrijgen: het gaat over dingen als ervaring, goed beleid, prestige, ... Deze dingen zijn moeilijker te verkrijgen. Het zijn vaak de mogelijkheden die een goed bedrijf onderscheiden van een gewoon bedrijf. In theorie heeft iedereen toegang tot dezelfde resources, dus de mogelijkheden zijn belangrijk om het verschil te maken. Waarom betalen mensen 5 euro voor een koffie bij Starbucks, terwijl je in elk tankstation voor

50 cent vergelijkbare koffie kan krijgen? Starbucks schenkt ook maar gewoon gemalen koffiebonen en warm water. Het zijn de mogelijkheden die bij Starbucks het verschil maken: de kennis bij het selecteren van de juiste koffiebonen, de sfeer in de winkels, de reputatie van het merk. Hiermee levert Starbucks een dienst die voor veel mensen een hogere waarde heeft.

3.2. Strategy Management for IT-Services

Geen enkele IT-dienst of IT-bedrijf werkt in een vacuüm. Zelfs een interne IT-dienst moet bepalen hoe ze relevant kan blijven en zichzelf kan blijven onderscheiden van externe aanbieders. Het is het doel van het Strategy Management-proces is om te leiden tot een **strategisch plan**, en daarna te bekijken of dit strategisch plan gevolgd wordt in de beslissingen van het bedrijf of de afdeling. Hoewel ITIL in onnoemelijk veel bedrijven wordt toegepast is dit vaak een proces dat over het hoofd wordt gezien. Te veel bedrijven werken nog reactief: ze reageren enkel op vragen van de markt en bekijken hun succes vooral op korte termijn. Door na te denken over de evoluties in de markt, en daar als bedrijf een antwoord op te bieden is het echter mogelijk om **pro-actief** te werken en te anticiperen op uitdagingen. Denk maar aan de vele kleine computerwinkeltjes die de populariteit van tablets en smartphones en de bijhorende dalende vraag naar desktop computers niet of te laat gezien hebben.

Belangrijk voor een strategisch plan is dat je goed weet in welke markt je je bevindt. Enkele belangrijke vragen waarop een goed strategisch plan antwoorden geeft: Wie zijn onze klanten? Wat willen onze klanten? Leveren onze huidige diensten wat de klanten willen? Hoe zetten we ons in de markt zodat wij de logische aanbieder zijn voor onze klanten? Kunnen we met onze huidige diensten nieuwe markten aanboren? Zijn er nieuwe diensten die we kunnen aanbieden om een gat in de markt te vullen? Uiteraard is zo'n strategisch plan niet iets dat je eenmaal schrijft, en daarna onveranderlijk wordt uitgevoerd. Op regelmatige basis moet je het plan bijschaven om te blijven beantwoorden aan veranderende markt. Vaak is een strategisch plan ook iets wat moet goedgekeurd worden door andere onderdelen van een bedrijf en zijn er enkele kladversies voor je tot een definitief document komt.

De strategie die uit dit proces komt kan je gebruiken om de andere processen binnen Service Strategy te ondersteunen.

3.3. Demand Management

Een volgende proces uit Service Strategy is Demand Management. Bedoeling van Demand Management is het **meten, voorspellen en sturen van het gebruik van een IT-dienst** om op deze manier de nodige capaciteit beschikbaar te stellen. Dit proces hangt nauw samen met Capacity Management uit Service Design, al gaat het bij Demand Management eerder om de bedrijfsactiviteiten. Het zijn de gegevens die we in deze fase ontdekken die we zullen gebruiken om met Capacity Management een technische oplossing uit te werken. In ITIL wordt voor deze gegevens een de term **bedrijfsactiviteitenpatroon (Pattern of Business Activity)** gebruikt.

Om te kunnen **inspelen op pieken en dalen** in het gebruik is het uiteraard handig dat je weet wanneer deze pieken en dalen typisch voorkomen. Voor een dienst die al in productie is, is dit eenvoudig te meten. Op voorhand de vraag voorspellen kan ook als je een goed begrip hebt van de klant, of zijn activiteiten: Zo weet je dat software die lonen berekent eerder op het einde van de maand gebruikt zal worden. Databanken die op kantoor gebruikt worden zien meestal de meeste activiteit tussen 8 en 18u. Inschrijvingssoftware voor scholen zal vooral in augustus en september vaak gebruikt worden, ... Wanneer deze kennis er niet is, is het belangrijk met de klant te praten over de manier waarop zij werken om op die manier een zo goed mogelijke voorspelling te maken. Als je een goed contact heb met je klant is het meestal ook mogelijk om uitbreidingen bij de klant (extra personeel of extra klanten) op tijd op te vangen.

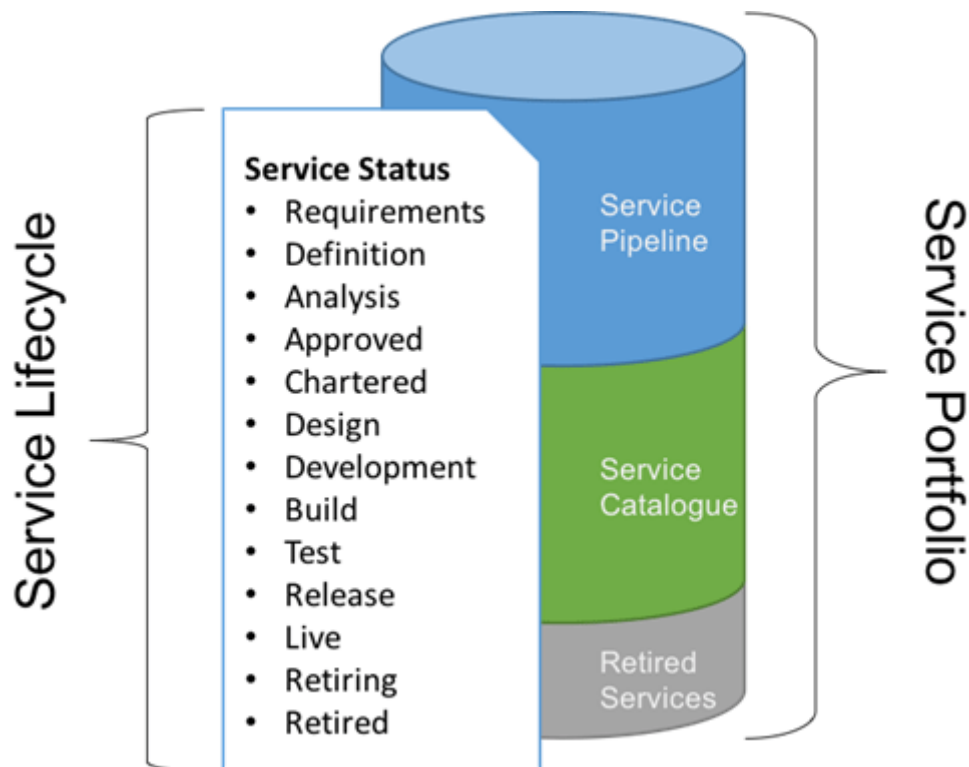
Het **sturen van de vraag** is moeilijker te doen, maar niet onmogelijk, afhankelijk van de dienst die je aanbiedt. Een voorbeeld hiervan dat iedereen kent is het niet (of half) laten meetellen van data tijdens daluren. Je provider doet dit niet alleen om je een plezier te doen, ook om er voor te zorgen dat belastend verkeer verschuift naar uren dat er netwerkcapaciteit op overschot is. Een ander voorbeeld hier is het gratis bellen in de daluren, of kortingen in webwinkels buiten de typisch drukke perioden.

3.4. Service Portofolio Management

Service Portofolio Management is gericht op er voor zorgen dat een dienstverlener **de juiste mix van diensten** aanbiedt om voor een evenwicht te zorgen tussen de investeringen in IT en de doelen van het bedrijf. Ook dit hangt samen met het Strategisch plan dat je in de eerste stap hebt opgesteld. Volgens dit proces neem je beslissingen over welke diensten je aanbiedt en aan wie, welke diensten op welke manier gefinancierd worden, de investeringen die in

een dienst worden gemaakt tijdens de levensduur van een project, en analyseer je welke diensten best buiten dienst gesteld kunnen worden.

Om deze stappen uit te voeren is het belangrijk dat je weet welke diensten je momenteel aanbiedt en welke diensten op elk moment in ontwikkeling zijn en welke diensten je niet meer aanbiedt. Je kan ook best voor elke dienst duidelijk vermelden wat de positieve en negatieve punten zijn, en waarom deze dienst nuttig is om op deze manier beslissingen makkelijker te kunnen nemen. De verzameling van deze informatie wordt het **Service Portfolio** genoemd.



De **Service Pipeline** is bedoeld voor de diensten die momenteel in ontwikkeling zijn, maar nog niet in productie zijn. Hierin zitten zowel projecten die nog in de planningsfase zitten, of projecten die al geïmplementeerd worden. De **Service Catalogue** is voor de producten die wel al beschikbaar zijn voor klanten en dus op dat moment lopend zijn. **Retired Services** zijn diensten die niet meer aangeboden worden. Het is belangrijk om ook deze projecten bij te houden omdat zelfs uit een mislukt project lessen te trekken zijn, delen eventueel hergebruikt kunnen worden, of op zijn minst de reden bewaard blijft waarom een bepaald project is stopgezet.

Het oplijsten van de diensten is slechts een deel van het werk. Op regelmatige basis moet worden bekeken of het huidige portofolio nog voldoet aan de strategische eisen van het bedrijf. Hiervoor is het belangrijk dat voor elk van de diensten een **Business Case** wordt bijgehouden, een rechtvaardiging van een bepaald project. Het is niet omdat een dienst op een bepaald moment populair was, dat het nog steeds de moeite is om de dienst actief te

houden. Zo heeft Proximus beslist om vanaf december 2017 de telegramdienst af te schaffen omdat door de opkomst van e-mail de dienst nog amper gebruikt werd. Je kan op deze manier ook beslissingen nemen over de waarde van een dienst voor je bedrijf, en beslissingen nemen over prioriteiten (op gebied van financiering of werkuren).

Een Business Case moet minimaal het doel van een dienst bevatten en de groep waaraan deze dienst geleverd wordt. De impact van een dienst op het bedrijf zowel positief als negatief (bijvoorbeeld deze dienst zorgt voor een toename van de productiviteit van x %, voor een kost van x euro), niet alle impact is meetbaar maar probeer je analyse wel zo veel mogelijk meetbaar te maken. De risico's die zijn verbonden met de dienst, en hoe met deze risico's wordt omgegaan, en een aanbeveling.

3.5. Financial Management voor IT-Services

Financial Management is een proces dat, uiteraard, volledig rond geld draait. Bedoeling van dit proces is om **financiële transparantie en aansprakelijkheid** te creëren (Welke dienst kost ons het meest en waarom?) en op die manier er voor te zorgen dat het aanwezige kapitaal op een verantwoorde te investeren. Dit proces heeft invloed op alle andere processen, omdat het de budgetten bepaalt die op elk moment in de cyclus beschikbaar zijn.

Financial Management is ook van belang omdat geld nu eenmaal iets is wat iedereen in het bedrijf begrijpt, ook buiten de IT-afdeling. Wanneer je technische capaciteiten kan uitdrukken in een monetaire waarde kan je niet-technische werknemers makkelijker overtuigen van het belang van IT. Een boekhoudafbeelding die het bedrijfsnetwerk gebruikt om films te downloaden zal je niet overtuigd krijgen om te stoppen door te zeggen dat ze op een maand 15TB aan data gebruikt hebben, maar wanneer je zegt dat dit het bedrijf maandelijks 3000 euro kost zullen ze hopelijk sneller luisteren. Het is de taak van Financial Management om op de verborgen IT-kosten een prijs te plakken. Dit is ook belangrijk als je je Management wil overtuigen om investeringen te doen in IT.

Financial Management valt uiteen in drie subprocessen. **Budgetering (Budgeting)**, **Boekhouding (Accounting)** en **Facturering (Billing)**, enkel het laatste van deze deelprocessen is optioneel. Budgetering bepaald welke budgetten beschikbaar zijn, en hoe deze worden toegewezen, bij dit gedeelte speelt de strategie en het service portofolio een belangrijke rol. Boekhouding gebruik je om te controleren waar de middelen effectief naar toe zijn gegaan. Facturering gaat over het opvolgen van de betalingen van klanten (indien van toepassing). ITIL

legt een nadruk op het zo correct mogelijk inschatten van de verbonden kosten (**Total Cost of Ownership vs aankoopprijs**).

3.6. Business Relationship Management

Business Relationship Management is een nieuw proces in de 2011-versie van ITIL. Het lost een cruciaal probleem op dat voor veel frustratie zorgde tussen de klant enerzijds en de leverancier van een IT dienst anderzijds. De Business Relationship Manager is verantwoordelijk voor de **communicatie tussen beide partijen**. Het zorgt ervoor dat de klant één aanspreekpunt heeft binnen de dienst of het bedrijf van de leverancier. Bij het opstarten van een project zorgt hij ervoor dat de noden van het bedrijf van de klant vertaald worden naar doelstellingen die door het IT personeel begrepen worden, en dat opmerkingen van de IT'ers naar de klant worden doorgegeven op een manier die voor niet technisch personeel verstaanbaar is. Hij is ook bij problemen het aanspreekpunt bij de leverancier. In ITIL zijn zijn taken als volgt beschreven: "Hij zorgt ervoor dat de verwachtingen van de klant niet hoger zijn dan wat de klant wil betalen, en dat de dienstverlener in staat is de verwachtingen van de klant kan beantwoorden, voor er akkoord gegaan wordt de dienst te leveren."

Dit hele proces is een reactie tegen verkopers die de neiging hebben om de klant onmogelijke dingen beloven om toch maar een contract binnen te halen, terwijl de IT-ers eigenlijk weten dat wat gevraagd is technisch niet haalbaar is, of niet binnen het budget van de klant past. Voor 10.000 euro is het niet mogelijk een volledig op maat gemaakt boekhoudsysteem te leveren, maar misschien is het wel mogelijk een bestaand systeem met kleine aanpassingen te leveren voor dat bedrag. Om zijn taak goed uit te voeren moet de Business Relationship Manager zowel op de hoogte zijn van de technische kant van de dienstverlener, als van de bedrijfsactiviteit van de klant. Door communicatie met beide zijden zorgt hij ervoor dat er geen dingen beloofd worden die later niet geleverd kunnen worden en vermijdt op die manier frustratie tussen de twee partijen.

3.7. Het resultaat van Service Strategy

Op het einde van deze fase, en dus voor er effectief begonnen wordt aan het ontwerpen van een dienst zouden de volgende dingen aanwezig moeten zijn:

- Een overzicht van de diensten die worden aangeboden.

- Een lijst van doelstellingen van het bedrijf, en een akkoord om diensten te financieren.
- Een lijst van voorwaarden vanuit de bedrijfskant of uitdagingen die opgelost moeten worden (Let op: dit zijn geen technische voorwaarden, in deze fase wordt er enkel gekeken naar de zakelijke kant.)
- Gedocumenteerde gebruikspatronen die de basis zullen vormen voor Capacity Management in de ontwerpfase.
- Elke regelgeving, strategie of beperking waarbinnen de design fase moet werken (bijvoorbeeld: wij willen ons profileren als een groen bedrijf.)

Deze documenten zijn de basis waarmee het technisch personeel in Service Design aan de slag kan.

HOOFDSTUK 4: SERVICE DESIGN

In de Service Strategy-fase is beslist wat we willen bouwen, welk budget er voor beschikbaar is, wanneer de oplossing beschikbaar moet zijn, en aan welke eisen ze moet voldoen,... Deze gegevens zijn verzameld door de Business Relationship Manager na contact met de opdrachtgever. Het is de bedoeling van Service Design om deze eisen om te zetten naar een ontwerp voor een technische oplossing. Het bouwen, testen en uitrollen van een oplossing gebeurt in een latere fase.

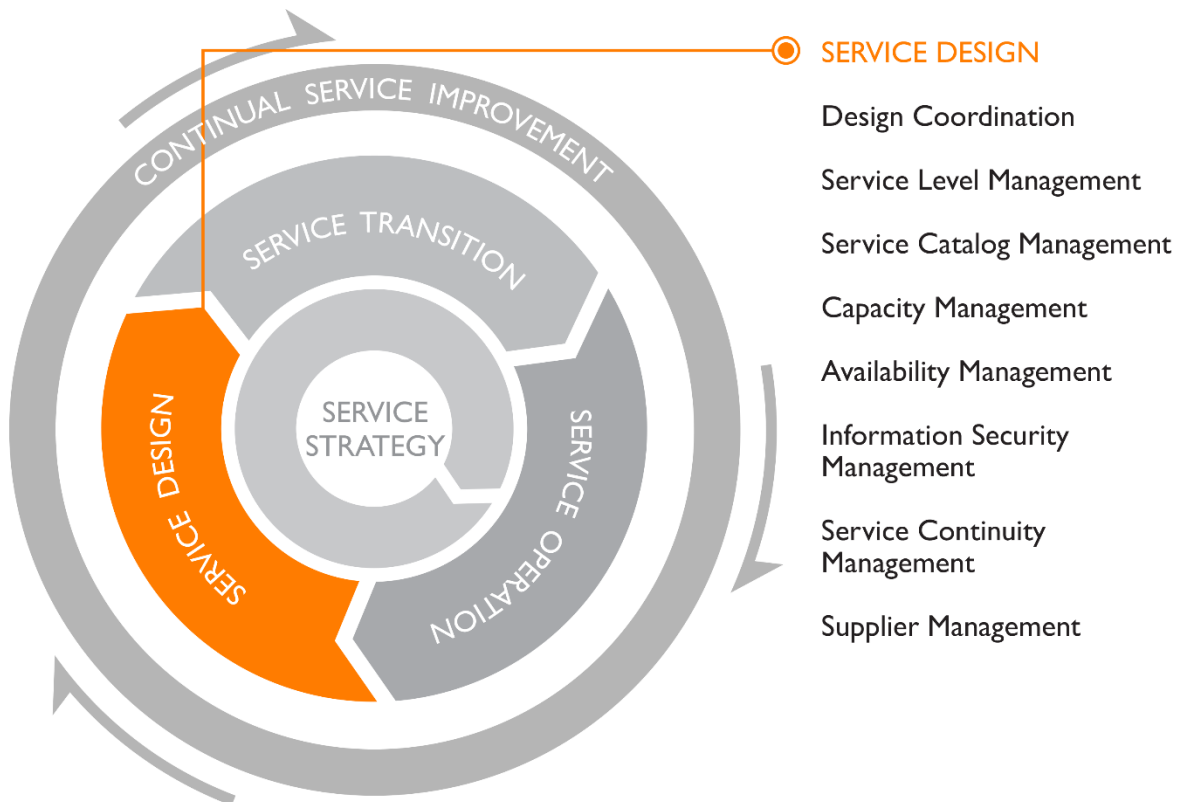
Doelen van Service Design:

- Het ontwerpen van nieuwe of gewijzigde diensten met als doel deze in een productieomgeving te gebruiken
- Alle processen, praktijken en beleid ontwerpen die nodig zijn om de strategie van de dienstverlener uit te voeren.
- Het samenstellen van een **Service Design Package** voor de volgende stappen in de levenscyclus van een dienst.

Het **Service Design Package** is de verzameling van documentatie die nodig zijn om een dienst volledig te beheren: Het bevat alle documentatie die Service Transition nodig heeft om een dienst te implementeren, testen en installeren. Het bevat alle gegevens die Service Operation nodig heeft om een in productie zijnde dienst te beheren (zo worden bijvoorbeeld alle tests beschreven die zullen gebruikt worden om te controleren of de dienst werkt zoals gewenst.). Het bevat de business-eisen die in de Service Strategy-fase zijn verzameld zodat kan worden gecontroleerd of de oplossing aan alle eisen voldoet, ... Met andere woorden: Een Service Design Package bevat documenten die alle aspecten van een IT-dienst gedurende de volledige levenstijd van die dienst beschrijven.

Voor elke nieuwe IT-dienst, elke grote wijziging binnen een dienst, en het uit dienst stellen van een IT-dienst moet een Service Design Package gemaakt worden. Bij het stopzetten van een dienst is het nuttig omdat er ongetwijfeld aanpassingen gemaakt moeten worden in de infrastructuur, processen moeten aangepast worden, ... Ook al kan het een stuk korter zijn als een Service Design Package bij het ontwerpen van een nieuwe dienst, is het dus belangrijk dit niet uit het oog te verliezen om niet voor verrassingen te komen staan.

Je kan je een Service Design Package best voorstellen als een **blauwdruk** voor een dienst die alles bevat voor het implementeren, uitrollen, meten, en verbeteren van een dienst. Na de Service Design-fase wordt dit pakket doorgegeven aan de Service Transition fase om het ontwerp effectief te implementeren.



Service Design is een toegevoegde waarde voor het bedrijf om de volgende redenen:

- Betere kwaliteit van dienstverlening:

Een goed ontwerp zorgt voor een dienst die beter zal zijn eens de dienst in productie is. Wanneer je je tijd hebt genomen voor een ontwerp zou het aantal verrassingen tijdens het uitrollen en gebruik van de dienst miniem moeten zijn. Een nadeel van de ontwerpfase is dat het pas tastbare resultaten oplevert nadat het ontwerp geïmplementeerd is, toch is het belangrijk in de fase genoeg tijd en middelen te investeren. Veel IT-projecten komen net in de problemen omdat het ontwerp snel is gebeurd waardoor er bij de implementatie moet bijgestuurd worden en er kostbare tijd en middelen verloren gaan.

- Het is mogelijk diensten beter te laten aansluiten bij de doelen van het bedrijf zelf:

Tijdens de Service Strategy-fase kunnen er bepaalde bedrijfsdoelen of strategieën naar boven zijn gekomen, dingen als wij willen ons profileren als een groen bedrijf, of we willen vooral cloudtechnologie gebruiken. De Design-fase zorgt ervoor dat deze doelen

geen dode letter blijven, omdat zij een totaal overzicht hebben over de hele levensduur van een project, en deze waarden in elke fase kunnen implementeren.

- De design-fase kan na verloop van tijd de Total Cost of Ownership naar beneden halen. Hier hoort wel een disclaimer bij: dit is enkel waar als alle onderdelen op een goede manier ontworpen zijn, en als alle aanbevelingen in de latere fasen gevolgd worden en correct geïmplementeerd zijn.

4.1. Good - Fast - Cheap

Bij het ontwerpen van een dienst zal je je vaak moeten verantwoorden tegenover externe stakeholders. Als het van opdrachtgevers afhangt, dan is alles wat je ontwerpt tegelijkertijd perfect volgens de verwachtingen, snel geïmplementeerd, en goedkoop. In de praktijk is dit zo goed als onmogelijk. Een gezegde dat al jaren meegaat is dat een product goedkoop, snel of goed kan zijn, maar niet alle drie tegelijkertijd.



HEINLEY

Eén van je taken als ontwerper van een dienst is er voor zorgen dat je de klant, of je eigen management kan overtuigen wanneer hun eisen onrealistisch zijn. Als je een goede band hebt met deze persoon kan je hem gerust bovenstaande afbeelding voorschotelen, maar meestal zal je iets diplomatischer moeten zijn.

Je kan de drie aspecten van een product als volgt vertalen: Goed draait om de functionaliteit van een product, bijvoorbeeld de features die een bepaalde software-oplossing moet hebben. Goedkoop zijn de beschikbare middelen die je hebt om het project te ontwikkelen. Dit kan

slaan op de aankoop van ondersteunende technologie, of beschikbare manuren. Snel is de planning die voor een project gevolgd moet worden, meestal de datum dat een project in productie gebracht moet worden.

Om het naar een eenvoudig voorbeeld terug te brengen: Als een klant je vraagt om met je team van 5 programmeurs de nieuwe Word te ontwikkelen binnen het jaar zal hij bepaalde toegevingen moeten doen: Met hetzelfde team en vereisten zal vijf jaar een stuk realistischer zijn, of het team moet uitgebreid worden naar 50 personen om de oorspronkelijke deadline te halen, of de klant moet met hetzelfde budget en deadline tevreden zijn met iets dat meer op kladblok zal lijken.

Meestal zijn de oorspronkelijke vereisten niet het probleem, die zijn hopelijk tijdens de oorspronkelijke onderhandelingen realistisch vastgelegd, de grootste moeilijkheden doen zich voor wanneer de voorwaarden wijzigen tijdens het ontwerp van een project: De klant moet de oplossing onverwacht twee maanden eerder hebben, een deel van het beschikbare personeel is overgeplaatst naar een ander project, of de klant komt met extra functionaliteit die niet in het oorspronkelijke ontwerp is opgenomen. Op zich is dit geen onoverkomelijk probleem, zolang dat de klant dan duidelijk wordt gemaakt dat hij op één van de andere punten toegevingen zal moeten doen: Komt hij met extra functionaliteit zal het product langer in ontwikkeling zijn, of moet er budget worden vrijgemaakt om extra mensen aan te nemen. Moet een project sneller afgerond worden zullen er extra investeringen nodig zijn of zal het niet mogelijk zijn alle gevraagde functionaliteit te implementeren. Worden er mensen van het project gehaald worden, dan zal de deadline verschoven moeten worden, of moeten de vereisten teruggeschoefd worden.

Bij dit hele proces speelt **de Business Relationship Manager** een belangrijke rol om de eisen van de klant in lijn te houden met wat technisch mogelijk is. Aan de andere kant is het als dienstverlener erg belangrijk om goed te communiceren en geen dingen te beloven die je later niet kan waarmaken.

4.2 Service Design: Concepten

Voor we in de verschillende processen van Service Design duiken is het belangrijk nog enkele overkoepelende aspecten te belichten. Dit is een raamwerk waarin je de verschillende processen kan inpassen.

DE VIJF ASPECTEN VAN SERVICE DESIGN

De input voor het hele Service Design-proces zijn de business requirements (bedrijfseisen) die uit de Service Strategy-fase komen. De vijf aspecten van Service Design zijn de uitkomst van het proces, de vijf dingen die binnen deze fase ontworpen moeten worden.

1. Het ontwerp voor een **oplossing voor een dienst**, met inbegrip van alle technische vereisten, nodige hulpbronnen en mogelijkheden die nodig en overeengekomen zijn.

Het resultaat van Service Strategy zijn de eisen waaraan een oplossing moet voldoen vanuit het standpunt van het standpunt van het bedrijf: Bijvoorbeeld een manier om de gegevens van klanten op te slaan die voor elke werknemer raadpleegbaar is. Binnen de design-fase moet dit omgezet worden naar een technische oplossing: bijvoorbeeld een webpagina waarop een werknemer kan inloggen om gegevens uit een relationele databank met klantgegevens te halen.

2. Het ontwerp van **beheersdiensten en werktuigen**, zoals de Service Portofolio

Wanneer je een nieuwe dienst ontwerpt, zal je ook wijzigingen moeten uitvoeren aan de bestaande tools die je gebruikt om je IT-diensten te beheren, zo zal je bijvoorbeeld je Service Portofolio moeten wijzigen, of de software van je Service Desk moeten aanpassen zodat problemen met het nieuwe systeem ingegeven kunnen worden. Wil je met een afzonderlijke bug-tracker werken, of integreer je die in de bestaande tracker. Krijgt deze dienst een apart telefoonnummer om contact op te nemen met de helpdesk? Deze wijzigingen moeten ook in deze fase ontworpen moeten worden.

3. Het ontwerp van **de infrastructuur** die noodzakelijk is voor de dienst.

Naast de oplossing zelf moet je ook de ondersteunende infrastructuur ontworpen worden. Voor mijn klantendatabank zal ik bijvoorbeeld de login-pagina moeten kunnen uitvoeren. Gebruik hiervoor een webserver met Apache, of verkies ik Microsoft ISS. Kies ik MySQL of Oracle voor mijn databank, Gebruik ik fysieke Servers, of zal ik de servers via HyperV draaien,

4. Het ontwerp van de nodige **processen**

In de design-fase moeten ook alle processen uitgeschreven worden die nodig zijn om met de nieuwe oplossing te werken. Hoe geef ik een nieuwe werknemer toegang tot het systeem, of hoe zorg ik ervoor dat de toegang wordt afgesloten voor iemand die niet

langer in het bedrijf werkt. Hoe worden nieuwe gegevens aan de databank toegevoegd, Hoe worden gegevens gewijzigd,

5. Hoe worden metingen uitgevoerd, en hoe moeten de resultaten geïnterpreteerd worden. Een laatste onderdeel dat ontworpen moet worden in de design-fase zijn de metingen die je zal uitvoeren om te controleren of het systeem naar wens werkt. De testen die je hier ontwikkeld zullen bij de Service Operation-fase uitgevoerd worden. Om verder te gaan bij mijn voorbeeld bijvoorbeeld een test om de reactiesnelheid van de databank te controleren: Hoeveel milliseconden duurt het voor een werknemer een antwoord heeft op zijn opzoeking, en binnen welke waarde moet deze liggen om aanvaardbaar te zijn. Moet dit dagelijks, wekelijks of maandelijks gemeten worden?

Samenvattend: je moet aan het einde van deze fase het ontwerp van een dienst hebben die beantwoord aan de gedocumenteerde eisen van het bedrijf, en een Service Design Package dat gebruikt kan worden door de Service Transition-fase om het ontworpen systeem te bouwen en uit te rollen. Het uiteindelijke ontwerp kan je best laten bevestigen door de opdrachtgever. Het vormt je garantie, het akkoord dat de ontworpen oplossing voldoet aan de verwachtingen. Het kan je beschermen tegen de gevreesde feature creep.

DE VIER INVLOEDEN OP SERVICE DESIGN

Als de vijf aspecten beschrijven wat je moet ontwerpen, dan zijn de vier invloeden de dingen waarmee je rekening moet houden tijdens het ontwerpen. Deze vier invloeden zijn: Mensen (People), Processen, Producten en Partners. Ze worden ook de vier P's van Service Design genoemd.

Tijdens het ontwerp zal je je constant de vraag moeten stellen of je voor de oplossing die je aan het ontwerpen bent, **de nodige mensen** beschikbaar hebt om je ontwerp uit te voeren. Heb je genoeg kennis beschikbaar om je te helpen bij het ontwikkelen van je oplossing, heb je collega's die je uitgedachte systeem ook kunnen implementeren. (Wanneer je bedrijf bijvoorbeeld vooral ervaring heeft met het ontwikkelen van Web Applicaties, kan je bij je design misschien beter niet uitgaan van een Desktop-oplossing), hebben we voldoende kennis om het uitgerolde systeem te onderhouden en te ondersteunen? Afhankelijk van je antwoord zal je bepaalde keuzes moeten maken: Je kan je systeem aanpassen aan wat er beschikbaar is van personeel, of je doet een aanbeveling om mensen met een bepaald profiel aan te trekken.

Een tweede element dat een rol kan spelen bij je ontwerp zijn de **bestaande processen**. Zal je je oplossing zo ontwikkelen zodat het past binnen een bepaald proces, en moeten er dan

wijzigingen worden aangebracht in dat proces. Of is een andere oplossing beter, maar moet er dan een nieuw proces geschreven worden?

Producten slaat op alles wat gebruikt kan worden om het systeem te ontwerpen, te implementeren en operationeel te houden. Dit is een verzameling **van hulpmiddelen, technologie en diensten**. Gebruikt het bedrijf Linux-servers, dan zal een e-mailoplossing die werkt op Microsoft Exchange waarschijnlijk niet de beste oplossing blijken. Welke bandbreedte is mogelijk op de locatie van het bedrijf? Hebben we de correcte licentie om de technologie op een bepaalde manier te gebruiken?

Partners ten slotte zijn **leveranciers, producenten en verkopers**, iedereen die je nodig hebt om je oplossing te kunnen leveren. Kan een leverancier voldoende Servers leveren om je uitgedachte systeem op te zetten? Als we willen dat een ander bedrijf ondersteuning levert voor een dienst, waarmee moeten we dan rekening houden?

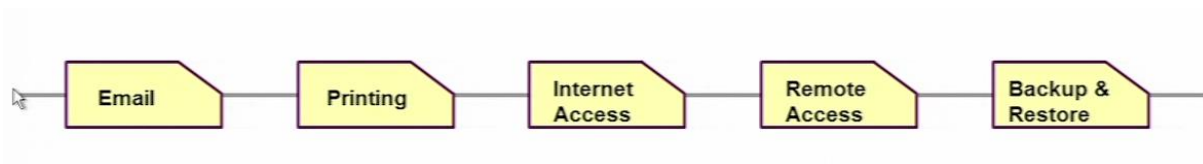
Tijdens je ontwerp zal je deze vier invloeden steeds in je achterhoofd moeten houden.

4.3. Service Catalogue Management

Een eerste proces uit Service Design is Service Catalogue Management. Je herinnert je uit het onderdeel Service Strategy Service Portofolio Management, waar bekeken werd welke diensten voor het bedrijf zinnig waren om aan te bieden. Het Service Portofolio was hierbij het belangrijkste hulpmiddel. Binnen dit portofolio was de Service Catalogue, het gedeelte met de diensten die momenteel aangeboden worden het belangrijkste stuk. Service Catalogue Management is het gedeelte binnen ITIL dat verantwoordelijk is om de gegevens binnen die Service Catalogue steeds zo up-to-date mogelijk te houden.

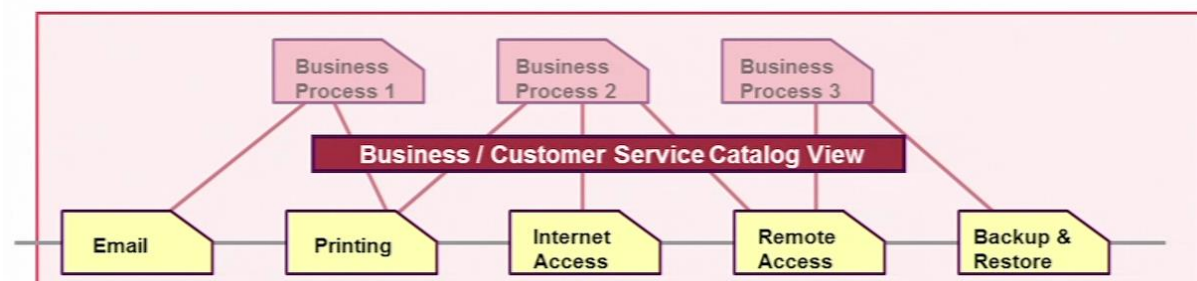
Volgens ITIL is het doel van Service Catalogue Management er voor te zorgen dat de Service Catalogue de **enige bron is voor informatie** voor alle operationele diensten. Daarnaast zorgt Service Catalogue Management er voor dat deze bron van informatie **toegankelijk is voor iedereen die toestemming heeft** de informatie te gebruiken. In de meeste gevallen is dit iedereen binnen het bedrijf, zowel de IT-kant als de bedrijfskant. De catalogue is een soort menu waar vermeld staat wat momenteel beschikbaar is, dit kan voor verschillende sectoren van het bedrijf praktische informatie zijn. Verkopers om te kijken wat onmiddellijk leverbaar is om op de vraag van een klant te antwoorden, IT om te weten wat ze bijvoorbeeld als ondersteunende dienst binnen een oplossing kunnen gebruiken. Tenslotte moet Service Catalogue Management ook alle **details, behoeften, status en interfaces** bijhouden van

de diensten die momenteel in productie zijn, of voorbereid worden om in productie te zijn. Voorbereid om in productie te zijn is niet alles wat nog in de design of transitie-fase zit, dit hoort thuis in de pipeline, wel de diensten die klaar zijn om uit te rollen. Alles wat een klant kan bestellen hoort thuis in de Catalogue, vanaf er een leverdatum is vastgesteld. Je kan de Catalogue bekijken als de folder waaruit een klant diensten kan kiezen, en voor de dienstverlener een lijst van wie welke diensten gebruikt.

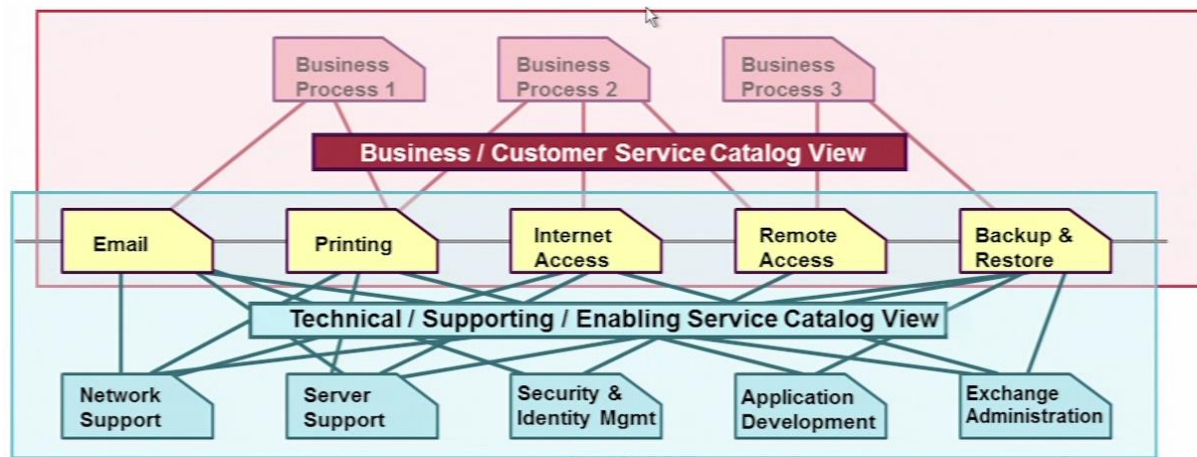


In zijn eenvoudigste vorm is de Service Catalogue niet meer dan een lijst van de diensten die een dienstverlener aanbiedt. Binnen de verschillende diensten vind je dan bijvoorbeeld informatie over wie de dienst mag gebruiken, de technische vereisten voor de dienst, etc. Voor veel bedrijven volstaat deze eenvoudige oplijsting, en ITIL maakt zich niet druk over welke vorm deze lijst aanneemt, zolang hij bestaat, dit kan een eenvoudig Excel-document zijn bijvoorbeeld.

Als je je Catalogue grafisch wil voorstellen is het wel mogelijk enkele handige toevoegingen te doen



Bovenstaande is een voorbeeld van een Business/Customer Service Catalogue View. Naast een lijst van de aangeboden diensten, wordt er ook duidelijk gemaakt welke bedrijfsdiensten de verschillende IT-diensten gebruiken. Dit kan handig zijn om in één oogopslag te zien welke bedrijfsdiensten (of klanten) nadeel hebben wanneer je een bepaalde dienst uitschakelt voor onderhoud bijvoorbeeld.



Je kan dit schema nog verder uitbreiden, door ook de ondersteunende diensten op te nemen. Dit zijn de diensten die de aangeboden diensten ondersteunen, de **enabling services**. Terwijl het voorgaande schema belangrijk is voor de klant, en dus de Business Relationship Manager, is dit laatste schema vooral nuttig voor het IT-personeel. De klant maakt het niet uit of een dienst uit is door een gecrashte service, of een beveiligingsprobleem, voor IT is het een goed hulpmiddel om de impact van fout in de ondersteunende diensten in te schatten.³

4.4. Service Level Management

Service Level Management is binnen ITIL een cruciaal proces. In het inleidende hoofdstuk heb ik op een bepaald moment een heel simpel stappenplan gegeven voor wat je als IT-bedrijf wil doen: Maak een afspraak over wat je wil doen, doe het, en zorg dat je kan bewijzen dat je het gedaan hebt. Service Level Management draait volledig rond stappen één en drie van deze cyclus.

³ Een voorbeeld van een klantgerichte service catalogue: <https://its.ucsc.edu/services/index.html>

Service Level Management heeft de volgende doelen:

- Het onderhandelen, eens worden over, en documenteren van bereikbare IT-doelen in **Service Level Agreements**
- Het monitoren en produceren van rapporten over de geleverde diensten (ism Service Operation)
- Er voor zorgen dat de klant een duidelijke en eenduidige verwachting heeft over het niveau van de dienstverlening (ism. alle fasen van ITIL)
- De relatie en communicatie met de klant verbeteren (ism. Bussiness Relationship Management)

SERVICE LEVEL REQUIREMENTS

Een belangrijk hulpmiddel in Service Level Management is de Service Level Agreements. Elke nieuw te ontwikkelen dienst begint met de vraag van een klant. De klant zal aan iemand binnen het bedrijf, hopelijk de Bussiness Relationship Manager, vragen een nieuwe dienst te ontwikkelen, en daarbij zijn eisen stellen over waar de dienst aan moet voldoen, de **Business Requirements**. De BRM hoeft geen technisch aangelegd persoon te zijn, zijn taak is enkel het overbrengen van de verwachtingen van de klant naar de Service Level Manager.

De Service Level Manager krijgt de eisen van de klant, en zal deze vertalen naar Service Level Requirements, dit zijn de bussiness requirements omgezet naar technische taal, zodat ze voor iemand die in IT werkt betekenisvol zijn. Vervolgens wordt er met de klant samengezeten om te kijken of deze SLRs een goede vertaling zijn van zijn oorspronkelijke eisen.

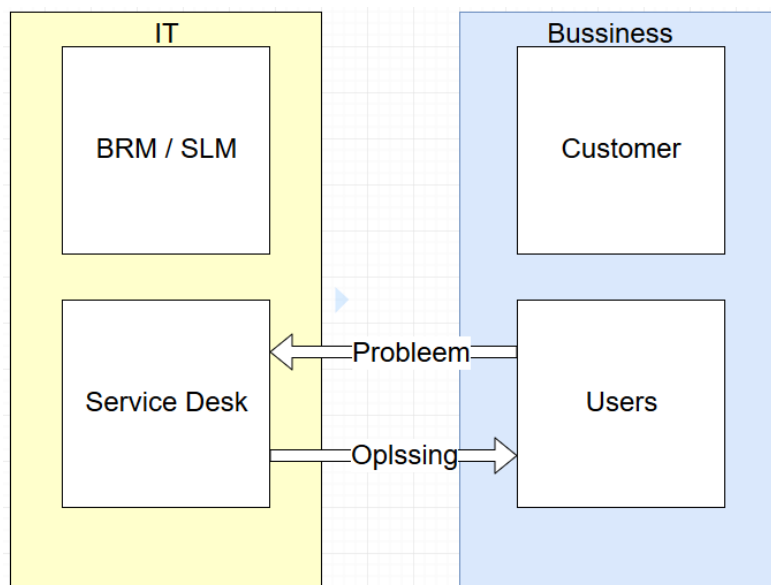


De SLR is dus een technische vertaling van de oorspronkelijke eisen van de klant. Dit document dient als basis voor verdere onderhandeling, en wordt waarschijnlijk nog enkele keren aangepast. Bedoeling is om tijdens de onderhandelingen de eisen van de klant te verzoenen met wat technisch haalbaar is. Wanneer er een akkoord wordt bereikt wordt deze SLR omgezet naar een Service Level Agreement, de **minimumvoorwaarden** waaraan de dienst moet voldoen waartegen latere beslissingen in de designfase worden afgetoetst.

COMMUNICATIE

Met de Service Level Manager is er een nieuwe persoon bijgekomen die contact heeft met de klant. Daarom is het niet slecht de verschillende communicatielijnen tussen de dienstverlener en het bedrijf te bekijken.

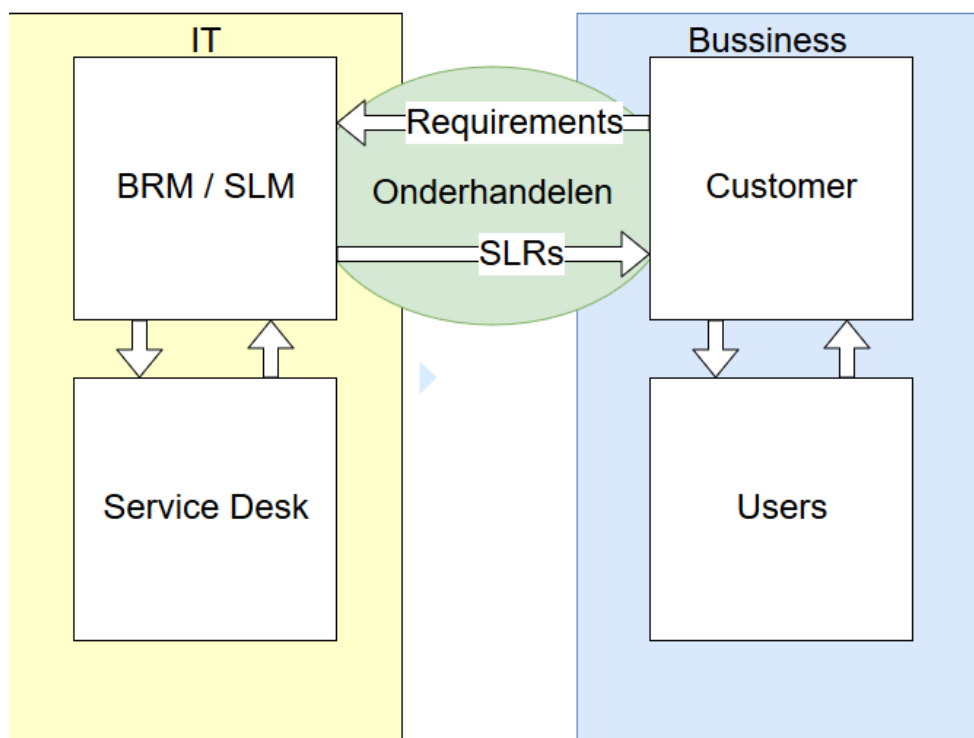
Aan de kant van het bedrijf moeten we een duidelijk onderscheid maken tussen het management en de gebruikers van de IT-dienst. De gebruikers hebben geen beslissingsrecht of budget om IT-diensten te kopen, management wel en dit is dus in ITIL de klant voor een IT-dienst. Aan de kant van de dienstverlener is er een zelfde soort onderscheid: aan de ene kant heb je mensen die beslissingen kunnen nemen (bijvoorbeeld de BRM en SLM), en aan de andere kant de uitvoerders (waarvan de Service Desk als enige contact heeft met het bedrijf van de klant.)



Normaal gezien is de enige communicatie tussen beiden tussen de eindgebruikers en de Service Desk: een gebruiker heeft een probleem en contacteert de Service Desk. De Service desk geeft de oplossing en communiceert die naar de klant. Intern kan de Service Desk met iedereen overleggen, maar het is belangrijk dat communicatie naar buiten door iemand van de Service Desk gebeurt. (Alleen al om te vermijden dat een eindgebruiker volgende keer rechtstreeks contact opneemt met iemand van bijvoorbeeld de Server-dienst).

Wanneer een eindgebruiker merkt dat een bepaalde oplossing niet meer voldoet aan de eisen zal hij geen contact opnemen met de Service Desk (die kunnen de dienst niet aanpassen) of met de BRM/SLM (hij heeft geen beslissingsrecht), maar met zijn eigen management dat als ze dit zelf ook een goed idee vindt de vraag aan de BRM zal stellen.

De BRM zal kijken wat de klant wil, en zal eventueel extra informatie vragen (zie Demand Management), kijken welke diensten momenteel aangeboden worden die eventueel gebruikt kunnen worden, overleggen met de rest van het bedrijf,... en uiteindelijk enkele opties aan de klant geven. Wanneer het begint te gaan over de technische details wordt de SLM betrokken. Binnen deze onderhandelingen wordt door de BRM/SLM input gevraagd van de verschillende diensten van de dienstverlener. Zij communiceren enkel binnen hun eigen bedrijf.



De onderhandelingen zelf kunnen er bijvoorbeeld als volgt uit zien: De klant wil een boekhoudoplossing. De BRM heeft twee mogelijkheden, een direct leverbaar algemeen systeem of een op maat gemaakt, en dus duurder, systeem. De klant wil het op maat gemaakte systeem, maar wil niet zo veel betalen. De SLM zegt dat dit eventueel mogelijk is, als er geen ondersteuning moet geboden worden voor het pakket in het weekend zodat er geen personeel moet betaald worden op die momenten. Een andere vraag die gesteld moet worden is wat de uptime van het systeem moet zijn. Uiteraard zegt de klant 100%, en de SLM reageert hierop dat dit weer extra kosten met zich meebrengt omdat dan extra servers als back-up moeten gebruikt worden, ... Op die manier wordt verder onderhandeld tot er een akkoord is over wat geleverd moet worden, en tegen welke prijs. Het resultaat wordt in een Service Level Agreement gedocumenteerd en door beide partijen ondertekend.

Een laatste belangrijke stap de communicatie van de klant naar de eindgebruikers: Stel je een situatie voor waarbij een dienst gewend is geraakt aan een bepaald niveau van dienstverlening van de Service Desk, maar door besparingen is de SLA aangepast en de helft van de Service

Desk ontslagen. De gebruikers die gewend zijn om binnen het uur antwoord op hun vraag te krijgen moeten plots een dag wachten. Wanneer dit door het management van de klant niet gecommuniceerd wordt naar de gebruikers zal dit tot een hoop frustratie leiden tussen de gebruikers en de service desk (terwijl die twee partijen niets aan de situatie kunnen doen). BRM / SLM moeten aan hun afdelingen de SLA communiceren zodat duidelijk is wat van hun verwacht wordt, maar ook de klant moet de verwachtingen van de gebruikers realistisch houden.

VERSCHILLENDE SOORTEN OVEREENKOMSTEN

Naast een Service Level Agreement zijn er nog enkele andere overeenkomsten die bepalen of een bepaald niveau van dienstverlening gehaald wordt. Om dit te illustreren een voorbeeld: Een gebruiker heeft internet nodig om een rapport te verzenden om 9 uur 's morgens, maar heeft momenteel geen netwerkverbinding. De eerste overeenkomst die hier van tel is de **Service Level Agreement (SLA)**. Hierin is bijvoorbeeld bepaald dat de gebruiker de service desk telefonisch kan contacteren en dat internetproblemen binnen het uur opgelost worden. De Service Level Agreement is een afspraak **tussen klant en de dienstverlener**. De SLA wordt opgemaakt door de Service Level Manager.

De gebruiker neemt contact op met de service desk, en er worden enkele stappen uitgevoerd om het probleem op te lossen. Na een kwartier blijkt dat het probleem niet door de Service Desk kan opgelost worden, en wordt het intern bij de dienstverlener doorgegeven aan de Desktopdienst. De desktopdienst ontvangt het bericht om 9u16. Omdat de dienst zijn wekelijkse teamvergadering heeft is er echter niemand aanwezig om het ticket op te vangen, en het duurt tot 9u48 voor de desktopdienst aan de slag kan met het probleem. Na wat tests blijkt het geen computerprobleem te zijn en de desktopdienst stuurt het terug naar de helpdesk met de vermelding dat het om een netwerkprobleem gaat. De helpdesk stuurt het door naar het netwerkteam om 9u55. Omdat de desktopdienst enkel heeft aangegeven dat het niet hun probleem is moet de netwerkdienst opnieuw beginnen met troubleshooten en wordt de deadline van één uur gemist, en moet de dienstverlener een financiële tegemoetkoming aan de klant betalen, vastgelegd in de SLA. Om dit soort problemen te vermijden zijn er tussen de verschillende afdelingen van een dienstverlener **Operational Level Agreements (OLA)**. Deze zijn intern aan de dienstverlener en bevatten de voorwaarden waaraan de diensten zich moeten houden tegenover de andere diensten van de dienstverlener. In een OLA kan bijvoorbeeld zijn opgenomen dat er steeds iemand van een dienst beschikbaar moet zijn om een probleem op te lossen, of dat berichten die tussen

diensten verstuurd worden voldoende informatie moeten bevatten. Deze OLA's worden door de Service Level Manager onderhandeld met de verschillende diensten, en zijn nodig om er voor te zorgen dat de doelen in de SLA met de klant behaald kunnen worden.

Stel dat het probleem zich niet voordoet op het netwerk van de dienstverlener, maar bij de provider van de dienstverlener. In dat geval is voor de oplossing van het probleem een derde partij nodig. Tussen de dienstverlener en zijn leveranciers bestaat ook een overeenkomst: de **Underpinning Contracts**, of een Derde Partijcontract. Deze overeenkomsten worden onderhandeld door Supplier Management. De voorwaarden van deze contracten kunnen meegerekend worden in de SLA met de klant (Als je provider vier uur garandeert om een probleem op te lossen kan je moeilijk een uur beloven in de SLA met je klant), of er worden verschillende termijnen geplakt op interne problemen en problemen bij de leverancier. Een Service Level Agreement wordt dus afgesloten met een klant, op basis van de Operational Level Agreements en de Underpinning Contracts van de dienstverlener.

WAARUIT BESTAAT EEN SERVICE LEVEL AGREEMENT

Service Level Agreement		
General Introduction <ul style="list-style-type: none"> Parties Signatures Service Description Period Glossary Reporting & Reviewing <ul style="list-style-type: none"> Content Frequency Incentives and Penalties	Support Service Hours Service Targets (<i>e.g., incident resolution targets</i>) Change Procedures Contact Points & Escalation	Delivery Availability Reliability Transaction Response Times Batch Turnaround Times Contingency & Security Charging

Er is niet zoiets als een standaard Service Level Agreement, elk bedrijf heeft zijn eigen idee over wat er in zo'n contract moet staan. Toch zal ik proberen enkele belangrijke onderdelen van een dergelijke overeenkomst te verduidelijken.

Het eerste gedeelte is **algemene informatie** over de SLA. Wie zijn de partijen waartussen de SLA wordt afgesloten: wie is, met andere woorden, de dienstverlener en de klant. Belangrijk is uiteraard ook dat beide partijen tekenen voor akkoord. Het is de bedoeling dat een SLA afdwingbaar is, zowel bij problemen bij de dienstverlening als onrealistische eisen van de klant. Er is niet zoiets als een eenzijdige Service Level Agreement. Er moet een beschrijving zijn van de te leveren diensten, en een periode waarin de SLA afdwingbaar is. Afhankelijk van de situatie kan dat bijvoorbeeld een jaar zijn. Ik raad wel af om bij langlopende contracten de SLA een te lange geldigheid mee te geven: Het is soms beter om ze jaarlijks te herbekijken samen met de klant en aanpassingen te maken, dan vast te zitten aan een SLA die door veranderende omstandigheden onhaalbaar is.

In het algemene gedeelte worden ook de details opgenomen van hoe gecontroleerd wordt of er aan de eisen voldaan wordt. Er wordt afgesproken wat er gemeten moet worden, hoe het rapport er uit moet zien, en niet te vergeten hoe vaak dit rapport wordt opgesteld (maandelijks, jaarlijks, ...).

In dit gedeelte kunnen ook de bonussen of straffen afgesproken worden wanneer doelen overschreden worden, of net niet gehaald worden. In dit overzicht staan ze in het zwart omdat ze voor een interne dienstverlener uiteraard optioneel zijn.

Onder het **Support gedeelte** geef je aan welke ondersteuning je biedt. Het is belangrijk hier specifiek te zijn. Als je niet vermeldt dat de Service Desk in het weekend niet beschikbaar is, zal de klant er van uitgaan dat deze 24 op 24 bemand is bijvoorbeeld. Ook de targets van de Service Desk lijst je in dit gedeelte op. Bijvoorbeeld een oplossingstijd van 24 uur, of een gegarandeerd antwoord (dit is niet hetzelfde als een oplossing) binnen het uur. Wanneer je werkt met prioriteiten, bijvoorbeeld een kritiek probleem wordt opgelost binnen het uur, dan moet je hier ook vermelden wat een kritiek probleem juist is. De Change Procedure is de procedure die gevolgd moet worden wanneer er een wijziging in de SLA, of de dienst moet gebeuren. Hoe moet de klant bijvoorbeeld een uitbreiding van de capaciteit aanvragen. Tenslotte is er ook een lijst van contacten die de klant bij specifieke problemen of vragen kan bereiken. Wie moet hij bellen als hij een vraag heeft over de facturering, of wie moet de dienstverlener contacteren als een dienst tijdelijk niet beschikbaar is.

Het derde gedeelte zou je het **Operationele gedeelte** kunnen noemen. Bij de beschikbaarheid geef je de gegarandeerde uptime van de dienst weer, dit wordt meestal uitgedrukt in een percentage. Let er op dat dit realistisch moet zijn: 100% wordt in deze contracten nooit gebruikt, 99,999% soms wel maar hou er rekening mee dat dit vertaalt naar een downtime van vijf minuten over de loop van een jaar. Enkel voor uiterst kritieke, letterlijk levensbelangrijke diensten is dit echt noodzakelijk. Reliability hangt hier mee samen, maar

betekent eerder om de hoeveel tijd er een panne van het systeem mag zijn. Transaction Response Times worden gebruikt om de reactietijd van het systeem te meten, en doelen vast te stellen. Hoe traag mag het systeem worden bij zware belasting bijvoorbeeld. Elke gebruiker klaagt over een traag systeem, hier definieer je wat volgens de SLA acceptabel is. Contingency bepaald hoe voorbereid een dienst moet zijn op onvoorziene omstandigheden, bijvoorbeeld een stroomuitval, de veiligheid bepaalt dan weer hoe goed data in het systeem is afgeschermd en wie toegang moet krijgen. Af sluiten doe je met de kostprijs van de dienst. Ook wanneer je niets aanrekent is het geen slecht idee om dat expliciet te vermelden. Je merkt dat de SLA erg afhankelijk is van andere processen, de SLM moet dan ook de nodige input vragen van andere verantwoordelijken bij het opstellen van de SLA.

Over het algemeen moet je er naar streven om je SLA in een **duidelijke taal** te schrijven, **zonder dat verschillende interpretaties** mogelijk kunnen zijn. Probeer het document **niet te technisch** te maken (het moet ook voor leken leesbaar zijn), en **vermijd advocatentaal**. In tegenstelling tot een EULA wil je dat dit document zo veel mogelijk gelezen wordt, zodat alle partijen weten waar ze aan toe zijn. Zorg dat alles wat je opneemt in de SLA meetbaar is (zodat je geen filosofische discussies moet houden over wat snel en traag betekenen). Als het niet in de SLA staat mag de klant het niet verwachten.

Een SLA kan van toepassing zijn op een bepaalde dienst. Dit is vooral nuttig als alle klanten de dienst op een bepaalde manier gebruikt. Het e-mail systeem bijvoorbeeld werkt voor iedereen binnen het bedrijf hetzelfde. Deze worden **Service Based SLA's** genoemd. Een andere mogelijkheid zijn **Customer Based SLA's**. Bij dit type is de SLA afhankelijk van de groep van klanten die het systeem gebruikt. Voor een personeelsadministratieprogramma is het bijvoorbeeld mogelijk dat gewone werknemers die het systeem gebruiken om vakantiedagen aan te vragen voldoende hebben aan een beperktere ondersteuning, terwijl de HR-afdeling het volledige pakket moet hebben. De meeste bedrijven werken met **Multi-level SLA's**. Bij een multi-level SLA zullen bepaalde afspraken gemaakt worden voor het hele bedrijf, en specifiekere SLA's afhankelijk van welke diensten ze gebruiken of welke functie ze hebben.

DE SERVICE LEVEL MANAGER

Service Level Management is een hele boterham, daarom samenvattend nog eens de hoofdtaken van de Service Level Manager:

- Samen met de BRM achterhaalt de SLM de Business Requirements van de klant.

- Hij onderhandelt de Service Level Agreements en Operational Level Agreements, sluit een akkoord af en documenteert de afspraken
 - Hij herbekijkt en past deze akkoorden op regelmatige basis aan (bijvoorbeeld jaarlijks)
 - Hij werkt samen met de Supplier Manager om de SLA's en OLA's af te stemmen op de Underpinning Contracts
- Hij monitort de uiteindelijke dienstverlening
 - Hij voert de metingen uit
 - Stelt de rapporten op
- Hij zoekt naar mogelijkheden om de dienstverlening te verbeteren
- Hij behandelt klachten (en complimenten) omtrent de dienstverlening

In ITIL versies van voor 2011 kwamen hier ook nog al de bevoegdheden van de Business Relationship Manager bij. Deze nieuwe functie werd in het leven geroepen om de SLM te ontlasten. Dit is ook de reden dat de BRM en de SLM over het algemeen nauw samenwerken: De BRM neemt de meer strategische, lange termijn gerichte klantencontacten voor zijn rekening, terwijl de SLM de meer operationele contacten onderhoudt.

Een van de belangrijke taken is het uitvoeren van een **Service Review**. Bij een Service Review wordt gekeken of de dienst aan alle voorwaarden van de SLA voldoet. Hij rapporteert zijn bevindingen aan de service owner⁴ en de andere stakeholders. De resultaten van deze meting worden opgenomen in een SLA Monitoring-chart, of **SLAM-chart**. Bij een SLAM-chart is het de bedoeling dat in een oogopslag kan bekeken worden of aan alle voorwaarden voldaan wordt of niet. In een rapport kan dan verder ingegaan worden op de details.

⁴ De **service owner** is eindverantwoordelijkheid voor een bepaalde dienst. Dit is typisch iemand uit het management die beslissingen kan nemen over de dienst, en de dienst vertegenwoordigt in strategische vergaderingen.

Service Period: April 1 – April 30

<i>Metric</i>	<i>Target</i>	<i>Actual</i>
First Call Resolution Rate	60%	62%
Service Desk Abandon Rate	3.3%	3.27%
Email Service Uptime	96%	95.91%
Number of Priority 1 Incidents	< 4	2
Percentage of Priority 1 Incidents resolved on time	100%	100%
Percentage of Priority 2 Incidents resolved on time	98%	94.8%
Percentage of Priority 3 Incidents resolved on time	97%	98.04%

Wanneer de resultaten van de meting tegen vallen moet er een plan worden opgesteld om de dienstverlening te verbeteren. Ook dit is de verantwoordelijkheid van de SLM. Meer over deze plannen in het hoofdstuk Continual Service Improvement.

Business Relationship Management	Service Level Management
Managing the business relationship	Managing the service levels
Strategic/tactical	Tactical/operational
Focuses on understanding how services meet customer requirements	Provides information about the levels of services agreed and achieved
Nurtures the overall business relationship by keeping customers informed and happy	Only interacts with customers when BRM is there
Gathers high-level business requirements	Writes the SLAs and OLAs based on input from BRM
Warm fuzzies, kissing babies, playing golf, keeping the customer happy 😊	Documenting and negotiating SLAs and their targets, compiling reports
<i>BRM identifies customer needs and ensures that the service provider is able to meet the customers' needs.</i>	<i>SLM ensures that agreed achievable levels of service are provided to the customer and users.</i>

4.5. Capacity Management

De overige processen binnen Service Design zijn een stuk minder uitgebreid als Service Level Management, maar spelen wel een belangrijke rol binnen het opstellen van de SLA. Capacity Management werkt nauw samen met Demand Management binnen Service Strategy om er voor te zorgen dat de doelstellingen van de SLA behaald kunnen worden. Capacity Management is net als Service Level Management een proces dat de hele levenscyclus van een dienst omvat.

Capacity Management heeft de volgende taken:

- Het opstellen van een up-to-date en geschikt capaciteitsplan
- Advies geven aan de zakenkant en IT-kant over alles wat met capaciteit te maken heeft.
- Er voor zorgen dat de gemeten prestaties van het systeem de prestatiedoelen overstijgen, en de prestaties en capaciteit van de dienst en hulpbronnen zo aanpassen dat dit mogelijk wordt.
- Diagnoserende en oplossen van alle problemen die te maken hebben met capaciteit.
- De impact van veranderingen in het systeem op de capaciteit voorspellen
- Proactief maatregelen nemen om de prestaties van het systeem te verbeteren wanneer de kosten verantwoord zijn.

Binnen het Design gedeelte is het belangrijk om te achterhalen wat de toekomstige behoeften van de klant zullen zijn (zie ook Demand Management). Er moet geprobeerd worden de toekomstige behoeften van de klant om te zetten naar kwantitatieve gegevens. De klant kan bijvoorbeeld zeggen dat ze verwachten 100 nieuwe mensen in dienst te nemen. Wat betekent dat voor het bandbreedteverbruik van het netwerk? Welke diensten zullen deze mensen vooral gebruiken, en moet daar extra capaciteit voorzien worden (extra opslag voor de fileservers bijvoorbeeld). Moet er nieuwe hardware gekocht worden (laptops voor deze extra werknemers?) Capacity Management probeert van de algemenere resultaten van Demand Management over te gaan naar technische vereisten.

Om dit proces goed te laten verlopen is het belangrijk dat de juiste data wordt bijgehouden en raadpleegbaar is. Dit gebeurt in een **Capacity Management Information System**. Je kan dit zien als een soort databank waar de volgende zaken worden bijgehouden:

- Informatie over de noden van het bedrijf wat betreft capaciteit (aantal werknemers, aantal aankopen in een webwinkel, ...)
- De noden van de verschillende aangeboden diensten qua capaciteit
- Beschikbaarheid van hardware (aantal computers in gebruik, servers op overschot, ...)

- Financiële informatie (Welke budgetten zijn beschikbaar, wat kost extra capaciteit, ...)

Met behulp van deze gegevens wordt een **capaciteitsplan** opgesteld. Dit plan bevat wat er momenteel aan capaciteit geleverd wordt, en wat de verwachtingen zijn over een bepaalde periode. Meestal wordt dit plan jaarlijks opgesteld. Aan de hand van dit plan kan bijvoorbeeld gevraagd worden naar extra middelen mocht de huidige capaciteit niet volstaan voor toekomstig gebruik.

Het belangrijkste is dat Capacity Management zo vroeg mogelijk betrokken wordt bij de planning van diensten. Te vaak nog wordt er pas naar capaciteit gekeken op het moment dat een dienst in productie gaat en er klachten komen over de reactietijd. Goed uitgevoerd Capacity Management vermijdt heel wat frustratie tussen eindgebruikers en de IT-afdeling. Iedereen die al eens tickets heeft proberen bestellen voor een populair evenement weet hoe irritant het kan zijn wanneer capaciteit en gebruik niet op elkaar afgestemd zijn.

Het valt je misschien op dat er heel wat overlapping is tussen Demand Management en Capacity Management. Dit komt omdat pas sinds de laatste versie van ITIL Demand Management als een apart proces is opgenomen. In eerdere versies was Demand Management gewoon een onderdeel van het Capacity Management proces.

4.6. Availability Management

Availability Management vertoont heel wat gelijkenissen met Capacity Management. Waar het bij Capacity Management draait om de prestaties van een systeem (reactietijd, verwerkingssnelheid, ...) gaat het bij Availability om de beschikbaarheid van het systeem, simpelweg om up- en downtime.

- Het opstellen van een up-to-date en geschikt beschikbaarheidsplan
- Advies geven aan de zakenkant en IT-kant over alles wat met beschikbaarheid te maken heeft.
- Er voor zorgen dat de beschikbaarheid van het systeem de beschikbaarheidsdoelen haalt of overstijgt
- Diagnostiseren en oplossen van alle problemen die te maken hebben met beschikbaarheid.
- Proactief maatregelen nemen om de beschikbaarheid van het systeem te verbeteren wanneer de kosten verantwoord zijn.

Zoals je ziet zijn dit bijna dezelfde doelstellingen als bij Capacity Management, je hoeft enkel prestaties te vervangen door beschikbaarheid.

Ook Availability Management omvat de hele levenscyclus van een IT-dienst. Je kan het onderverdelen in een **proactief gedeelte** (tijdens Service Strategy, Service Design, Service Transition) en een **reactief gedeelte** (Service Operation).

In de designfase is Availability Management nauw betrokken bij het ontwerp van nieuwe of gewijzigde diensten. Het is de taak van Availability Management om op zoek te gaan naar kwetsbaarheden binnen het ontwerp die problemen kunnen veroorzaken voor de beschikbaarheid van een dienst. Tijdens de Transition-fase worden oplossingen voor deze kwetsbaarheden geïmplementeerd, en wordt de oplossing onderworpen aan tests om te kijken of de dienst “weerbaar” genoeg is: Als er zich een fout voordoet, kan het systeem zichzelf dan herstellen of zal er een crash veroorzaakt worden.

Wanneer een dienst operationeel is moet de beschikbaarheid van diensten gemeten worden, moeten de resultaten geanalyseerd en gerapporteerd worden. Bij elke niet beschikbaarheid van het systeem of onderdelen van het systeem moet de oorzaak gezocht worden, en kunnen er wijzigingen worden voorgesteld om dit probleem in de toekomst te vermijden.

Naast Availability, pure uptime, zijn ook enkele andere termen uit Availability Management belangrijk om te kennen. Uptime alleen is maar een deel van het verhaal, daarnaast is ook **betrouwbaarheid** belangrijk, betrouwbaarheid bepaalt hoe lang het systeem kan werken zonder onderbrekingen. Daarnaast zijn ook niet alle pannes gelijk: een panne in een **Vital Bussiness Function** is een stuk erger dan een panne bij een enhancing service. Een Vital Bussiness Function is een taak waarzonder het bedrijf zijn activiteit niet kan uitvoeren. Een bankautomaat die geen bonnetje kan drukken is vervelend, maar de meeste mensen zouden deze automaat niet als kapot beschouwen. Een bankautomaat die geen geld uitdeelt daarentegen... al deze variabelen moet je meenemen wanneer je het hebt over beschikbaarheid van een dienst (in de Intermediate cursussen ITIL wordt er dieper ingegaan op formules die je kan gebruiken om dit precies te berekenen).

Omdat beschikbaarheid op verschillende manier kan worden geïnterpreteerd is het belangrijk dat je in je SLA duidelijk maakt wat beschikbaar juist betekent. Om nog eens terug te keren naar een programma dat lonen uitkeert: Ben je iets met een uptime van 95% per maand als je programma telkens stopt met werken op het einde van de maand? Availability Management is een van de moeilijkste dingen om correct te krijgen binnen een bedrijf, net omdat er zo veel interpretatie mogelijk is over wanneer een dienst werkt en wanneer niet. Net hierom is het meestal ook een van de meest gedetailleerde onderdelen van een SLA.

4.7. IT Service Continuity Management

IT Service Continuity Management is het onderdeel van ITIL dat de meeste mensen gewoon Disaster Recovery noemen. Hoe kan een minimale dienstverlening verleend worden volgens een Service Level Agreement in het geval van onvoorspelbare omstandigheden. Dit proces dient als ondersteuning voor een algemener, bedrijfswijd plan. Een bedrijf dat goed voorbereid is op alles heeft meestal een plan over wat er minimaal moet gedaan worden om de deuren van het bedrijf open te houden. IT Service Continuity beslist onder andere over welke diensten er in dat geval aangeboden moeten worden om dat plan uit te voeren. Er kan besloten worden om bepaalde diensten niet te leveren, en enkel de meest noodzakelijke zo snel mogelijk terug op te starten.

Om deze taak uit te voeren moet Continuity Management regelmatig **Business Impact Analysis** uitvoeren Dit is eigenlijk een “wat als...” oefening. Wat gebeurt er bijvoorbeeld als een bepaalde dienst niet meer werkt. Er wordt een onderscheid gemaakt tussen een harde impact (financieel verlies, inbreuk op wetgeving, ...) en een zachte impact (impact op de goede naam van het bedrijf, verlies van competitiviteit, ...). Een voorbeeld van iets met een harde impact is het verlies van de boekhouding van het bedrijf, aangezien dit iets wat wettelijk gezien steeds aanwezig en opvraagbaar moet zijn, een voorbeeld van een zachte impact is het lekken van privé-gegevens van klanten.

Een tweede hulpmiddel zijn risicoanalyses. Hier stel je je de vraag wat de kans is dat er iets fout loopt. Je moet je de volgende vragen stellen: Wat is het risico juist? Wat is de impact van dat risico op het bedrijf, kan ik me ertegen beschermen, en is dat financieel de moeite? Ten slotte bekijk je de stappen die je moet uitvoeren om je systeem te beschermen.

Het resultaat van de Business impact Strategy en de Risico-analyse zijn plannen over hoe er met een bepaalde situatie moet omgegaan worden. Verantwoordelijkheden moeten worden vastgelegd (wat als het afdelingshoofd bijvoorbeeld niet aanwezig is), personeel moet training krijgen om zijn taken uit te voeren, en de plannen moeten worden getest en eventueel bijgestuurd worden, waar moeten de plannen opgeslagen worden (ze zijn vrij nutteloos als ze enkel op een server staan die tijdens de noodsituatie niet meer werkt.)

Veel kleine en middelgrote bedrijven werken met externe leveranciers om garanties te leveren in het geval van een noodsituatie. Veel bedrijven hebben bijvoorbeeld een backup van belangrijke gegevens in een extern datacenter, zodat in het geval van bijvoorbeeld een brand geen informatie verloren gaat. Continuity Management is verantwoordelijk voor de

contracten met dergelijke leveranciers. Ook wanneer de opslag binnen een andere vestiging van hetzelfde bedrijf gebeurt zijn zij er verantwoordelijk voor dat dit steeds goed verloopt.

Omdat rampen per definitie onverwacht zijn moeten Continuity Management steeds bekijken of veranderingen een invloed op hun plannen heeft, en indien nodig de plannen aanpassen.

4.8. Information Security Management

Information Security Management heeft als doel de data van het bedrijf te beschermen wat betreft vertrouwelijkheid, integriteit (kan data corrupt geraken) en beschikbaarheid (kan iedereen die aan de informatie moet geraken er aan geraken). Het niveau van beveiliging wordt bepaald door de Business kant van het bedrijf, IT implementeert de beveiliging. De mate van beveiliging kan bepaald worden door wetgeving, of gewoon door het beschikbare budget. Information Security Management hangt samen met Availability Management, en Access Management.

Doel is om een Information Security Policy op te stellen, een plan met alle onderdelen die belangrijk zijn om de veiligheid van gegevens te waarborgen. De volgende onderdelen kunnen in een dergelijk plan worden opgenomen worden:

- Regels rond gebruik en misbruik van IT-middelen (mag een partner van een werknemer een werklaptop gebruiken? ...)
- Regels rond wachtwoorden (Hoe complex, hoe lang blijft een wachtwoord geldig voor het gewijzigd moet worden, ...)
- Regels rond e-mail (Mag werk-email gebruikt worden voor persoonlijke correspondentie?)
- Regels rond internet (Welk type sites mag je op het werk niet gebruiken?)
- Regels rond antivirus (Moet dit op elk toestel geïnstalleerd worden, mag het tijdelijk uitgeschakeld worden?)
- Regels rond het classificeren van informatie (Wat is gevoelige informatie en wat niet?)
- Regels rond het classificeren van documenten
- Regels rond externe toegang (Mag er van buitenaf worden ingelogd op het bedrijfsnetwerk, en tot welke informatie heb je dan toegang?)
- Regels rond toegang van leveranciers (Tot welke onderdelen van het netwerk heeft een consultant toegang?)

- Regels rond uit dienst nemen van hardware (Worden overbodige computers vernietigd of verkocht, wat met de harde schijf?)

Dit plan moet voor iedereen binnen het bedrijf beschikbaar worden gesteld, en voor externen als ze de infrastructuur gebruiken.

Naast regels die voor het hele bedrijf gelden, is het ook mogelijk om voor specifieke groepen bepaalde extra regels op te stellen bijvoorbeeld omdat ze met gevoelige data werken (HR, boekhouding, ...). Bij het ontwerpen van nieuwe diensten moet er ook nagedacht worden over wie er tot welk onderdeel toegang moet hebben (de eigenlijke toepassing gebeurt in Access Management).

4.9. Supplier Management

Supplier Management houdt zich bezig met alles wat te maken heeft met leveranciers: de externe bedrijven die jij nodig hebt om je diensten aan te kunnen bieden. De Supplier Manager zorgt er voor dat de leveranciers hun geld waard zijn. Hij sluit contracten af en zorgt er voor dat de afgesloten contracten voldoen aan de vereisten van het bedrijf. Hij volgt de prestaties van de leveranciers op en probeert indien nodig bij te sturen. Hij zorgt er ook voor dat de Underpinning Contracten in lijn liggen met de Service Level Agreements van het bedrijf. Hij houdt alle gegevens bij in de **Supplier and Contract Management Information System**. Deze databank bevat gegevens over welke leverancier gebruikt wordt, dat die doet, eventuele informatie rond garanties, ...

4.10. Design Coordination

Service Design is een erg uitgebreide fase, met heel wat verschillende onderdelen. Om alles in goede banen te leiden is er een apart proces dat er voor zorgt dat al deze onderdelen samen kunnen werken.

Bedoeling is dat de Design Coördinator als aanspreekpunt geldt voor de gehele designfase. Hij blijft op de hoogte van het werk van alle processen en zorgt er voor dat alle teams up to date zijn met de laatste ontwikkelingen: hij geeft wijzigingen door in voorwaarden, leveranciers, ondersteuning. Daarnaast bewaakt hij de planning, hulpbronnen en lost hij conflicten op wanneer die zich voordoen. Hij stelt de uiteindelijke **Service Design Packages** op, de

documenten waar alle beslissingen uit de designfase verzameld worden, en beheerd en verbeterd het volledige designproces.

Ook de Design Coördinator is nieuw in de 2011 versie van ITIL. In eerdere versies waren er heel wat onduidelijkheden over wie er nu juist verantwoordelijk was voor bijvoorbeeld het verzamelen van de resultaten van de design-fase, of wie de toewijzing deed van de beperkte middelen aan de verschillende teams. De Design Coördinator moet dit gat vullen door het overzicht tussen de verschillende processen te bewaren.

4.11. De resultaten van de Designfase

Aan het einde van de designfase moeten er twee resultaten zijn:

- Een oplossing die beantwoordt aan alle voorwaarden die door de Service Strategy zijn gesteld. De voorwaarden van Service Strategy was de input voor Design Strategy. Deze voorwaarden worden gebruikt als basis voor de onderhandeling door de Service Level Manager, en het resultaat wordt door beide partijen goedgekeurd.
- Een volledig Service Design Package dat doorgegeven wordt aan Service Transition zodat zij de ontworpen oplossing kunnen implementeren. Dit Service Design Package bevat input van de Application Group (programmeurs), Netwerktechnici, de Process Owners van de verschillende Service Design-processen, ... Alle mogelijke informatie die aan het einde van de Designfase beschikbaar is over een project.

Omdat Service Design een erg grote invloed heeft op de andere fasen van de levenscyclus is de kwaliteit van wat uit de Designfase komt vaak doorslaggevend voor het succes van een dienst.

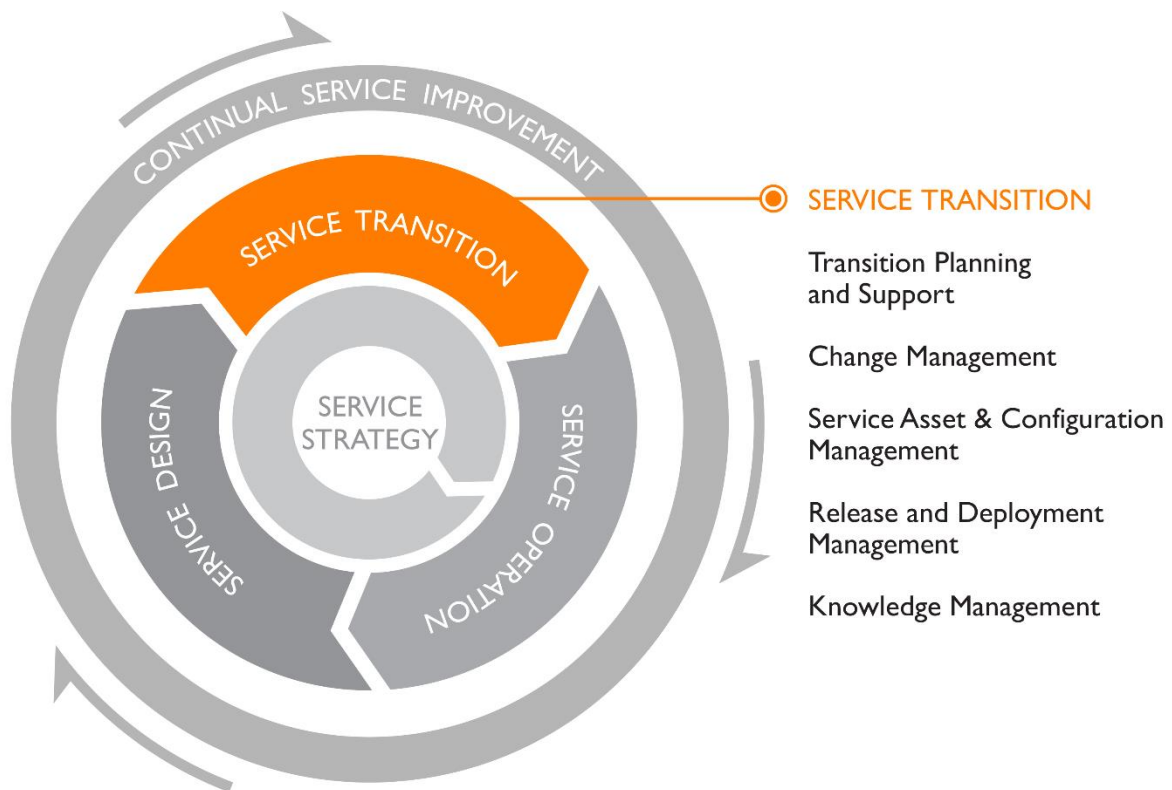
HOOFDSTUK 5: SERVICE TRANSITION

Service Transition is de fase binnen de ITIL levenscyclus waar je het ontwerp uit de design fase bouwt, test, installeert en uitrolt. Binnen Service Design zijn vijf dingen ontworpen: de dienst, ondersteunende tools, de architectuur, processen en metingen (de vijf aspecten van Service Design). In Service Transition neemt de blauwdruk en bouwt er een effectieve oplossing mee. Om het belang van Service Transition in de verf te zetten: Tachtig procent van alle incidenten binnen IT worden veroorzaakt door wijzigingen in bestaande systemen. Via Service Transition wordt gepoogd dit percentage naar beneden te brengen door deze wijzigingen op een gestructureerde en verantwoorde manier aan te brengen, en zo de belasting op IT-personeel te verlichten.

Verantwoordelijkheden van de Service Transition fase

- Het beheren van de complexiteit en de risico's verbonden met het aanbrengen van wijzigingen in bestaande diensten
- Het introduceren van nieuwe diensten, applicaties of onderdelen
- Het wijzigen van diensten, applicaties of onderdelen
- Het stopzetten van diensten, applicaties of onderdelen
- Het overzetten van diensten van en naar een andere dienstverlener

De benodigdheden die bepaald werden in Service Strategy, en uitvoerig uitgediept werden in Service Design worden in Service Transition succesvol gebouwd, getest en uitgerold, zodat ze gebruikt kunnen worden door het bedrijf in Service Operation. In deze fase mag het technische personeel zijn handen vuilmaken, na het zakelijke Service Strategy en Service Design dat een mix is van zakelijke en technische taken.



Volgens ITIL heeft Service Transition de volgende doelen: Eerst en vooral moet Service Transition er voor zorgen dat wijzigingen in diensten de verwachte waarde levert voor het bedrijf, door het ontwerp zo goed mogelijk te implementeren. Daarnaast moet het de risico's van wijzigingen zo veel mogelijk beperken, bijvoorbeeld door de wijzigingen uit te rollen op het moment dat een dienst niet of weinig in gebruik is, plannen maken zodat de wijziging zo efficiënt en effectief mogelijk gebeurt, er voor zorgen dat gebruikers de correcte verwachtingen hebben over de prestaties en het gebruik van het nieuwe of gewijzigde systeem en goede, up-to-date, informatie leveren over diensten en onderdelen van diensten.

Wanneer Service Transition op een correcte manier uitgevoerd wordt dan heeft het de volgende voordelen voor een bedrijf:

- Een hoger volume van succesvolle wijzigingen
- Het delen en hergebruiken van middelen
- Het terugbrengen van tijdsverlies door onverwachte conflicten en afhankelijkheden
- Juistere verwachtingen scheppen voor alle stakeholders
- Het vertrouwen dat een dienst volgens specificatie geleverd kan worden, zonder dat die een negatieve impact heeft op andere diensten en stakeholders.

5.1. Service Asset And Configuration Management

Service Asset And Configuration Management (of kortweg Configuration Management zoals het meestal genoemd wordt) is een proces dat als doel heeft er voor te zorgen dat alle middelen die een geleverde dienst nodig heeft om correct te werken **gecontroleerd** zijn, en dat **informatie over deze middelen beschikbaar is** waar en wanneer nodig.

Om het nut van dit proces te begrijpen moet ik even vooruit lopen naar Change Management, het proces verantwoordelijk voor het aanbrengen van wijzigingen: Stel je bent verantwoordelijk voor het uitrollen van een nieuwe applicatie, en om de applicatie te laten werken moet je een wijziging in de configuratie van de server maken. Je voert deze wijziging uit, maar hierdoor werkt een andere applicatie van een tweede team niet meer. Dit team gaat op zoek naar het probleem en voert wijzigingen uit waardoor jou applicatie plots stopt met werken, ...

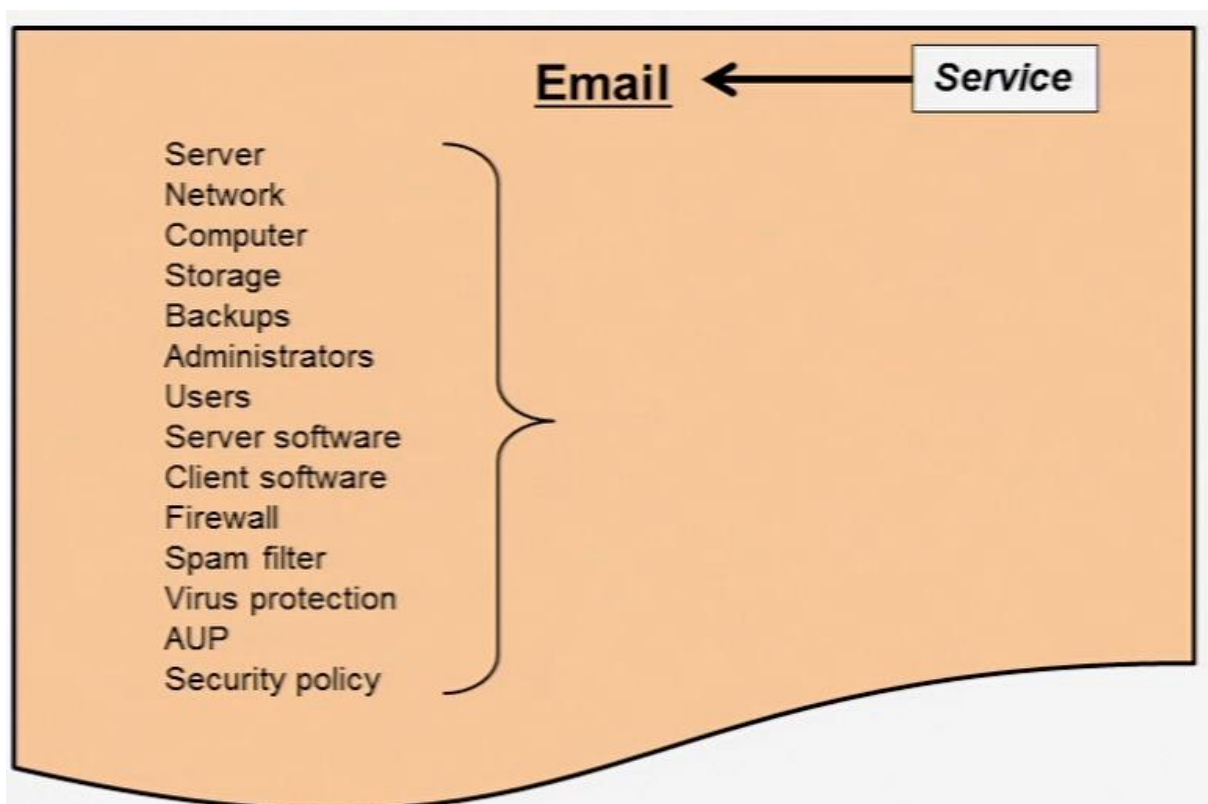
Om dergelijke problemen te vermijden levert Configuration Management een overzicht van je hele infrastructuur: de onderdelen, de instellingen, welke dienst afhankelijk is van welke middelen, ... Op deze manier kan je wanneer je een wijziging wil uitvoeren in een oogopslag zien wat de mogelijke impact van je wijziging is, en hoe je die impact kan beperken. Gecontroleerd uit de eerste alinea betekent in dit geval dat de middelen in deze databank zijn opgenomen.

De doelen van Configuration Management volgens ITIL:

- Identificeren, beheren, bijhouden en controleren van diensten en andere componenten, inclusief hun versies, standaardconfiguratie, samenstellende onderdelen, eigenschappen en relaties.
- Het beheren en beschermen van Configuration Items doorheen de levenscyclus door samen te werken met Change Management, zodat enkel goedgekeurde onderdelen gebruikt worden, en enkel goedgekeurde wijzigingen uitgevoerd worden.
- Het bijhouden van correcte informatie over de historische, huidige en geplande status van diensten en CI's
- Correcte informatie leveren aan andere processen zodat mensen op het juiste moment beslissingen kunnen nemen.

CONFIGURATION ITEMS

Het doel van Configuration Management is het bepalen en controleren van onderdelen van diensten en infrastructuur, en precieze data bijhouden over de historische, huidige en geplande status van diensten en infrastructuur. Om deze data te verzamelen moet je eerst een goed idee hebben van welke componenten een deel uitmaken van een bepaalde dienst. Als je als voorbeeld een email-dienst neemt dan zijn dat mogelijk de volgende onderdelen: Een server, een netwerk, computers, opslag, backups, gebruikers, beheerders, client-software, server-software, firewall, spamfilter, Acceptable Use Policy, Security-beleid, ...



Al deze onderdelen worden samen **Service Components** genoemd. Binnen deze groep zijn er speciale onderdelen die **Configuration Items (CI)** genoemd worden. Configuration Items zijn die onderdelen waarvan we beloven dat we ze vanaf het punt van aankoop, tot het punt dat we ze verwijderen zullen controleren. Met andere woorden de onderdelen die belangrijk genoeg zijn om er alle mogelijke informatie over bij te houden. De Configuration Items zijn de onderdelen die we zullen opnemen in de **Configuration Management Database**.

In deze databank worden alle Configuration Items bijgehouden en, erg belangrijk, worden de relaties tussen de verschillende Configuration Items bijgehouden. Computers en Server zijn een onderdeel van het netwerk, Client Software is geïnstalleerd op de computers, gebruikers moeten gehoorzamen aan Acceptable Use en Security Policy, De server is verbonden met de

Storage Array, ... Het zijn de relaties die de databank nuttig maken. Een lijst van losstaande onderdelen is niet erg interessant, maar als relaties tussen de onderdelen worden opgenomen dan wordt het plots mogelijk om de impact van wijzigingen in te schatten: Wijzig ik de poort voor uitgaande e-mail op de server, dan heeft dit invloed op de configuratie van de client software op de computers van de gebruikers. Het is mogelijk om te werken met meerdere CMDBs, bijvoorbeeld wanneer je bedrijf verschillende locaties heeft, het totaal van databanken wordt dan een **Configuration Management System** genoemd.

Er is geen gouden regel om te bepalen wat een CI juist is. Voor het ene bedrijf is het voldoende om een desktopcomputer te zien als één CI, terwijl een ander bedrijf scherm, toetsenbord, muis en de eigenlijke computer als aparte CI's ziet. Nog een ander bedrijf deelt de computer verder op in Harde Schijven, Software, besturingssysteem, ... Het belangrijkste is dat de keuze voor jouw bedrijf zinnig is. Weinig CI's betekenen een kleinere databank, die eenvoudig up-to-date te houden is, het grote nadeel is dat de informatie minder precies is. Een mogelijke vuistregel kan zijn: Alles wat individueel vervangen wordt, is een eigen CI. Al is ook deze regel niet in alle omstandigheden de juiste.

Stel een bedrijf heeft verschillende laptops in gebruik, en na aankoop blijkt dat er een probleem is met een bepaald model van harde schijven. Het bedrijf dat de harde schijven als aparte Configuration Items bijhoudt, zal eenvoudig de laptops kunnen opsporen die dat type harde schijf gebruiken en deze preventief vervangen. Het bedrijf dat dit niet doet zal alle laptops moeten nakijken, en indien nodig vervangen. Het is wel belangrijk alleen de nuttige informatie op te nemen, aangezien het onderhouden van de databank ook tijd kost en een databank die niet onderhouden is, even nuttig is als geen databank. Meer zelfs een verouderde of slechte CMDB zorgt er voor dat je beslissingen neemt op basis van foutieve informatie, terwijl er bij gebrek aan een databank hopelijk wel de tijd wordt genomen om de juiste informatie te achterhalen. Als je bedrijf geen techniekers heeft die een laptopscherm kunnen vervangen, en er dus steeds een nieuwe laptop gekocht wordt als er een probleem is met een scherm, is het tijdverspilling om de laptopschermen op te nemen als aparte CI's. De beslissing over wat moet opgenomen als CI is afhankelijk van hoe veel resources je bedrijf aan dit proces wil toewijden.

Een andere afweging die je moet maken is dat voor elke wijziging in een CI het Change Management proces moet doorlopen worden. Je kan je de vraag stellen of dit de moeite is bijvoorbeeld om een muis te vervangen, terwijl het misschien eenvoudiger is gewoon even naar de winkel te lopen. Voor een besturingssysteem is het misschien wel nuttiger aangezien het vervangen van een besturingssysteem invloed heeft op het licentiebeheer binnen het bedrijf.

Samenvattend:

Een Configuration Item is elk middel dat onder het beheer van Configuration Management staat en heeft de volgende eigenschappen:

- Kan een dienst of een onderdeel zijn
- verschillende soorten (hardware, software, mensen, documentatie)
- Opgenomen in een CMDB, beheerd door een Configuration Management System
- Elke CI-record heeft informatie (eigenschappen) en geschiedenis (status)
- CI's hebben een relatie tot andere CI's
 - Is een onderdeel van
 - Is samengesteld door
 - Is geïnstalleerd op
 - Is documentatie voor
 - Wordt beheerd door
 - ...

De Configuration Management Database is de databank waar deze CI's worden bijgehouden

Het Configuration Management System is de verzameling van databanken, tools, procedures om informatie over CI's en hun relaties te verzamelen, op te slaan, te beheren, bij te werken en weer te geven.

Een term die nog niet aan bod is gekomen is een **Configuration Baseline**. Dit is een momentopname van de configuratie van een systeem dat gebruikt kan worden als vergelijkingspunt, of om veranderingen terug te rollen naar een eerder werkende staat.

DEFINITIVE MEDIA LIBRARY (DML)

Een tweede "databank" die door Configuration Management wordt bijgehouden is de **Definitive Media Library**. Deze databank bevat alle officiële, goedgekeurde, versies van alle elektronische media die in het bedrijf gebruikt wordt. In eerdere versies van ITIL werd dit de DSL genoemd, waarbij de S staat voor software. In de huidige ITIL standaard is de definitie open getrokken om alle elektronische data te omvatten.

Databank kan in dit geval relatief zijn, het is gewoon de verzameling van software die in een bedrijf kan geïnstalleerd worden. In een klein bedrijf kan dit even goed de schuif van de systeembeheerder zijn waar de installatie-CD's verzameld worden. In deze bibliotheek is alle

zelf ontworpen software opgenomen, alle aangekochte software, alle licentie en activatiesleutels, en alle documentatie die onderworpen is aan Configuration Management.

Via deze databank heeft Configuration Management de totale controle over wat geïnstalleerd wordt, en welke versie geïnstalleerd wordt. Release en Deployment Management mag enkel software uit deze verzameling installeren. Het idee hierachter is dat alle software en updates eerst getest kunnen worden alvorens uitgerold te worden. Op die manier kan software die problemen oplevert met andere delen van de infrastructuur geweerd worden, of kan een bepaalde versie van een programma gekozen worden die werkt met de interne tools van het bedrijf. In veel bedrijven worden dingen als Windows Updates eerst getest binnen een test-omgeving, en pas nadat gecontroleerd is of er geen problemen zijn worden ze goedgekeurd (en dus opgenomen in de DML) en uitgerold op de computers van het bedrijf. (bijvoorbeeld via WSUS, de update server in een Windows domein omgeving).

Ook licenties worden in deze bibliotheek beheerd. Heb je een bepaalde licentie nodig dan vraag je die aan via Configuration Management. Op deze manier kan een bedrijf het overzicht houden over welke licenties zijn toegekend, en aan wie. En misschien nog belangrijker, er kan gegarandeerd worden dat elk product met de juiste licentie gebruikt wordt.

Het documentatie gedeelte zorgt er voor dat er steeds een officiële versie is van documenten. Het is uiteraard niet de bedoeling dat elk stukje papier hier wordt opgenomen, maar voor belangrijke documenten (bijvoorbeeld het IT continuity plan, de security policy) waarvan het cruciaal is dat er steeds één geldende versie is, is het nuttig om deze op te nemen. Omdat de impact van deze documenten groot kan zijn, moeten wijzigingen ook via Change Management lopen, het resultaat wordt hier opgenomen.

Drie processen zijn belangrijk voor de DML: Release and Deployment Management gebruikt deze bibliotheek voor alles wat ze installeren, Configuration Management onderhoudt en is eigenaar van de bibliotheek, en Change Management beslist wat er wordt opgenomen in de bibliotheek.

DE STAPPEN IN CONFIGURATION MANAGEMENT

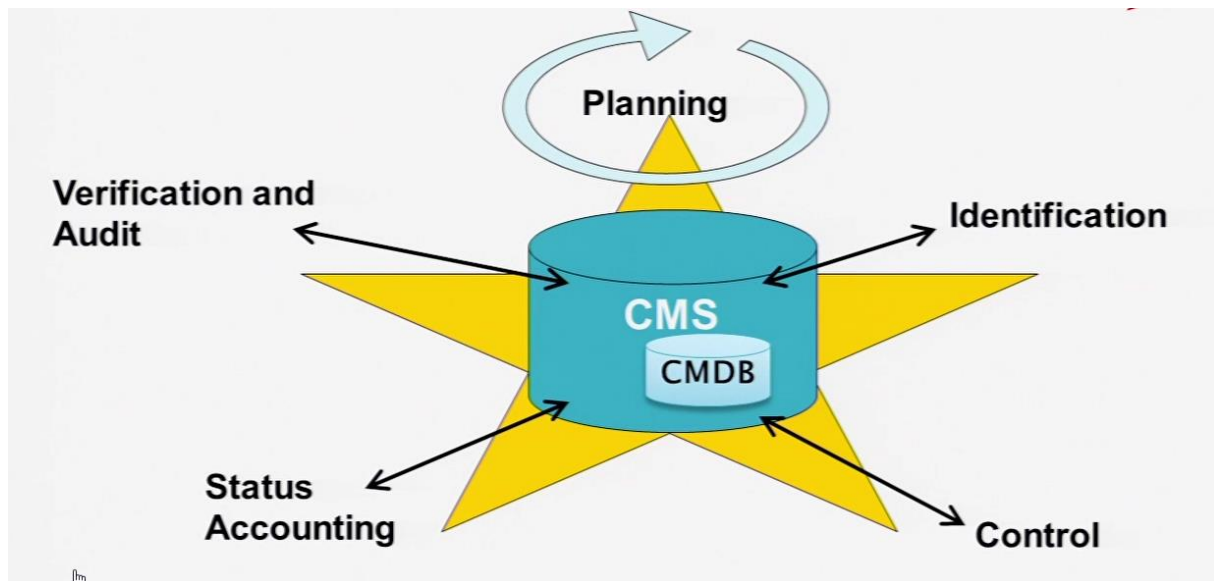
Configuration Management valt uiteen in vijf stappen. De eerste fase is de planning fase. Een eerste belangrijke vraag is of je als bedrijf Configuration Management wil toepassen. Hou er rekening mee dat het een erg arbeidsintensief proces is, en dat wanneer het slecht uitgevoerd wordt het schadelijker kan zijn dan het proces niet toepassen. In deze fase wordt ook beslist wie de proces owner is, wie meehelpt in het proces, ...

Een volgende stap in het toepassen van Configuration Management is het **identificeren** van de CI's die door Configuration Management moeten opgenomen worden. Zoals eerder gezien is niet alles even nuttig om aan Configuration Management te onderwerpen, aangezien het proces heel wat overhead kan teweegbrengen. Niet enkel wat moet opgenomen worden is een belangrijke vraag om te stellen, ook welke gegevens moeten worden bijgehouden per CI kan het verschil maken tussen een goede database en nutteloos extra werk. De vervaldatum van een garantie voor een computer is belangrijk om bij te houden, de kleur van de computer mogelijk niet.

Nadat je je databank hebt opgezet is het belangrijk dat je er ook voor zorgt voor **wijzigingscontrole** zodat de integriteit van de data behouden blijft. Je moet hier nadenken over hoe en wanneer wijzigingen aangebracht worden. Het is erg belangrijk in deze stap om samen te werken met Change Management zodat bij goedgekeurde wijzigingen ook de data in de CMDB bijgewerkt wordt. Ik heb het al een paar keer herhaald, maar omdat het zo belangrijk is doe ik het nog een keer: De databank is pas nuttig zolang hij up to date is!

Naast wijzigingscontrole is ook **statuscontrole** belangrijk. Je moet er voor zorgen dat voor elke CI een status is opgenomen, en dat deze status ook overeen komt met de werkelijkheid. Voorbeelden van een status kunnen zijn: in productie, besteld, in reserve, in onderhoud, buiten dienst, ...

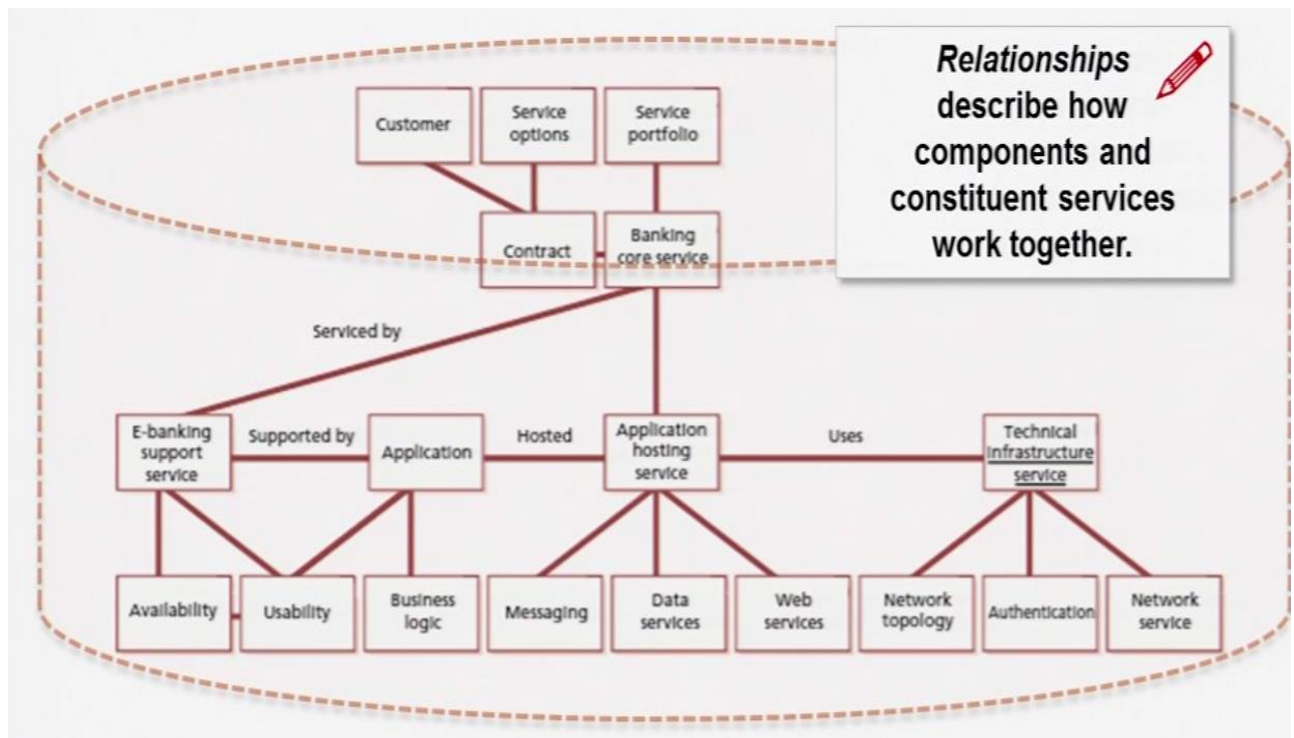
Perfectie bestaat niet, en zelfs wanneer je je plan voor 100% volgt kan het gebeuren dat bepaalde data in je databank niet meer klopt. Daarom is het cruciaal om op regelmatige tijdstippen je databank te **verifiëren**, zodat wat beschreven wordt nog steeds klopt met de situatie. Een mogelijke manier om dit te doen is willekeurige items uit je databank te halen en op de werkvloer te controleren of alle informatie overeenstemt. Wanneer blijkt dat dit niet klopt, dan zijn er twee mogelijkheden: De persoon die verantwoordelijk is voor de databank kan de wijzigingen niet bijhouden (als het proces niet geautomatiseerd worden), of waarschijnlijker: iemand heeft een wijziging doorgevoerd zonder Change Management te doorlopen (De laatste stap van Change Management is het inlichten van Configuration Management).



HET CONFIGURATION MANAGEMENT SYSTEM

Zoals gezien worden Configuration Items opgeslagen in een CMDB, de verschillende CMDB worden overkoepeld door een CMS. De CMDB zorgt er voor dat je eenvoudige vragen kan beantwoorden (Wat is het serienummer van Printer X bijvoorbeeld). Om op meer complexe vragen antwoord te geven bevat de CMS alle CMDBs van een bedrijf, maar bijvoorbeeld ook de Definitive Media Library, de databanken uit de andere processen (bijvoorbeeld de Supplier Management Database) en databanken van het bedrijf zelf (werknemers, klanten, ...). Door deze databanken te combineren kan het CMS antwoord geven op vragen als Welke Software versie is geïnstalleerd op de computers van klant X? Wie bij de leverancier moet ik aanspreken over de garantie voor router Y?

Om deze vragen op te lossen moeten de relaties tussen de verschillende onderdelen correct beschreven worden. Via deze relaties is het mogelijk een **Configuration Model** te maken van een bepaalde dienst. Dit model dient als basis voor Change Management om de impact van een wijziging zo goed mogelijk in te schatten. Wanneer Configuration Management correct wordt uitgevoerd kan het op deze manier van onschatbare waarde zijn voor een bedrijf, ook al is het een erg arbeidsintensief proces.



Dit maakt het mogelijk om de impact van veranderingen correct in te schatten.

5.2. Change Management

Een tweede proces uit Service Transition is Change Management. Zoals je misschien gemerkt hebt aan het aantal keer dat Change Management vernoemd is in het voorgaande stuk, werkt dit proces nauw samen met Configuration Management.

Om een vergelijking te maken met iets buiten de IT: Je kan Change Management vergelijken met een luchtverkeersleider. Wanneer een piloot wil veranderen van vlieghoogte dan vraagt hij toestemming aan de luchtverkeerleider. Die kijkt op zijn radar om te controleren of de verandering mogelijk is. In dit scenario is de piloot iedereen uit IT die een verandering in het IT-netwerk wil maken. Change Management is de luchtverkeerleiding: zij nemen de beslissing om de verandering toe te staan of niet. Om deze beslissing te nemen heeft hij de informatie nodig uit het Configuration Management System. Net als een piloot niet op eigen houtje mag beslissen om van vlieghoogte te wisselen, mogen er geen wijzigingen worden aangebracht zonder de toelating van Change Management. Wanneer dit wel gebeurt is het niet meer te voorspellen wat er zal gebeuren, en de wijziging wordt niet opgenomen in de CMS waardoor toekomstige problemen waarschijnlijker worden.

De doelen van Change Management zijn:

- Antwoorden op aanvragen voor veranderingen vanuit het bedrijf of IT, zodat de diensten beantwoorden aan de bedrijfsnoden
- Er voor zorgen dat veranderingen geëvalueerd en gedocumenteerd worden, en dat de toegelaten veranderingen een prioriteit krijgen, er een planning wordt opgesteld, getest, gedocumenteerd en gecontroleerd worden, en uiteindelijk geïmplementeerd worden op een gecontroleerde manier.
- Er voor zorgen dat alle wijzigingen van Configuration Items in het Configuration Management System door de Configuration Librarian.

Bij elk bedrijf zullen er regelmatig aanpassingen moeten gebeuren aan IT. Change Management moet er voor zorgen dat deze wijzigingen op een gecontroleerde manier worden uitgevoerd, in lijn met de doelen van het bedrijf. Let wel op: Change Management beheert het proces en speelt een coördinerende rol maar zal zelf de wijzigingen niet uitvoeren, dat is het werk van Release en Deployment Management.

Change Management speelt een rol bij alle wijzigingen in infrastructuur, processen, tools, metingen en documentatie, en alle IT-diensten en andere Configuration Items. Met andere woorden alles wat een invloed kan hebben op de IT-dienstverlening. Veel mensen denken vooral aan wijzigingen in programma's of netwerkinfrastructuur als het gaat om Change Management, maar ook bijvoorbeeld een wijziging in een SLA kan via het Change Management proces lopen. Alles wat door Configuration Management gecontroleerd wordt moet bij wijzigingen het Change Management proces doorlopen. Een wijziging is elke toevoeging, aanpassing of verwijdering die een effect heeft op IT-dienstverlening. Typisch worden deze wijzigingen aangevraagd via een Request For Change of RFC-document.⁵

SOORTEN WIJZIGINGEN

Binnen ITIL wordt er een onderscheid gemaakt tussen drie types van wijzigingen: normale wijzigingen (**normal change**) dringende wijzigingen (**emergency change**) en standaard wijzigingen (**standard changes**).

Veel IT'ers hebben schrik dat wanneer ITIL uitgerold wordt, ze geen werk meer kunnen uitvoeren zonder dat er eerst enkele uren in vergaderingen moeten doorbrengen om

⁵ Veel bedrijven gebruiken hiervoor een eigen term, Request For Change is de officiële ITIL-naam voor het document.

goedkeuring te krijgen. Door een onderscheid te maken tussen verschillende wijzigingen probeert ITIL dit probleem op te lossen.

Voor het eerste soort wijzigingen geldt de normale procedure: De aanvraag wordt op papier of met een online formulier gedaan. Hierna nemen de personen verantwoordelijk voor Change Management de verandering in beraad. Wanneer deze goedgekeurd wordt, wordt de wijziging opgenomen in de planning. Afhankelijk van de omvang van het bedrijf kan deze procedure enkele dagen of zelfs enkele weken in beslag nemen. Gelukkig zijn er manieren om hier onderuit te raken.

Een eerste uitzondering op de normale procedure zijn de **standaard wijzigingen**: over het algemeen zijn dit handelingen die een IT'er verschillende keren per dag moet uitvoeren: het installeren van een printer, updaten van software, vervangen van een harde schijf, ... Als voor dit soort wijzigingen elke keer de hele procedure doorlopen moet worden zou er amper nog iets uitgevoerd worden. De standaard wijzigingen zijn veranderingen die al op voorhand goedgekeurd zijn: het gaat meestal over routine taken, die een erg beperkte impact, en een laag risico met zich meebrengen. Vaak kosten ze ook niet veel. Het Change Management team kan beslissen om elke handeling op deze lijst bij te voegen. Voor deze acties is geen aanvraag nodig. Het is wel een goed idee om voor deze standaard wijzigingen ook een standaard procedure te voorzien, bijvoorbeeld om de CMDB up to date te houden (Dit wordt normaal door Change Management gedaan, maar aangezien ze niet betrokken zijn bij deze vooraf goedgekeurde wijziging moet er een alternatief voorzien worden).

De tweede uitzondering zijn de wijzigingen die dringend zijn, **de emergency changes**, en dus niet kunnen wachten tot de normale procedure doorlopen is. Bijvoorbeeld een server die is uitgevallen, bescherming tegen een nieuw virus dat aan het rondgaan is, het oplossen van een beveiligingslek, ... Wijzigingen die met andere woorden zo snel mogelijk moeten uitgevoerd worden. Toch is het belangrijk dat ook deze wijzigingen door Change Management lopen. Net als gewone wijzigingen moeten deze veranderingen aangevraagd worden en goedgekeurd worden voor ze uit te voeren. Voor deze wijzigingen hebben de meeste bedrijven een aparte procedure, en personen die in noodgevallen snel beslissingen kunnen nemen. Ook al is het drie uur 's nachts je moet deze personen bellen om de wijziging goed te laten keuren. Een mailtje met wat je gedaan hebt de volgende dag is niet voldoende. Op die manier probeert ITIL te garanderen dat er altijd minstens twee paar ogen naar een probleem hebben gekeken voor er een wijziging wordt uitgevoerd.

DE BESLISSINGSNEMERS

Zoals gezegd hebben bepaalde personen binnen een bedrijf beslissingsrecht over het aanbrengen van wijzigingen. De eindverantwoordelijkheid ligt bij de Change Manager, bijgestaan door het **Change Advisory Board**. Het CAB moet de Change Manager helpen bij het beslissingen rond grotere wijzigingen. Deze groep is samengesteld uit personen die de nodige autoriteit en ervaring hebben om beslissingen voor het bedrijf te nemen. Dit kunnen mensen uit IT-management of andere takken van het bedrijf zijn, proces owners, technisch personeel, ... De samenstelling kan wijzigen afhankelijk van de aard van de wijziging. Deze groep brengt enkel advies uit, de Change Manager neemt nog altijd de uiteindelijke beslissing (al wordt er in sommige bedrijven wel gewerkt met bindende stemmen).

Specifiek voor Emergency Changes is er een **Emergency Change Advisory Board**. Dit is meestal een subgroup van het CAB dat op korte termijn kan samengeroepen worden, en afhankelijk van de aard van het probleem kunnen ook andere mensen betrokken worden. Afhankelijk van de dringendheid kan dit erg informeel zijn. Het is wel belangrijk de Emergency procedure enkel voor de echte noodgevallen te gebruiken. Een gebrek aan planning waardoor je in tijdsgebrek komt valt niet onder een noodgeval.

WAAR KOMEN WIJZIGINGEN VANDAAN

Aanvragen van wijzigingen kunnen van verschillende bronnen komen. Het meest typische geval gebeurt via een RFC-document. Bijvoorbeeld een netwerkbeheerder die de configuratie van een router moet aanpassen. Een andere mogelijkheid zijn wijzigingen aangevraagd door eindgebruikers. Deze wijzigingen worden meestal via de Service Desk doorgegeven. Een designer die met een andere monitor wil werken valt bijvoorbeeld hieronder. Een derde bron zijn de wijzigingen die moeten worden aangebracht bij het opstarten van een nieuw project, deze wijzigingen worden aangevraagd via een Project Initiation Document. Deze worden meestal doorgegeven door de Project Manager van het betrokken project. De echt grote wijzigingen worden doorgegeven via een Change Proposal. Het gaat om erg ingrijpende wijzigingen: bijvoorbeeld het uitbesteden van de Service Desk.

CHANGE MODEL

Voor wijzigingen die erg vaak uitgevoerd moeten worden is het mogelijk dat er met een change model gewerkt wordt: dit is een reeks van stappen die in een specifieke soort wijziging uit te voeren op een vooraf overeengekomen manier. Je hoeft niet voor elke uitgevoerde wijziging toestemming vragen, je vraagt gewoon aan de standaardprocedure uit te voeren.

In een Change model moeten de volgende elementen aanwezig zijn:

- De stappen die uitgevoerd moeten worden om de wijziging uit te voeren
- De chronologische volgorde waarin deze stappen uitgevoerd moeten worden
- Verantwoordelijkheden (Wie doet wat, Wie keurt de verandering goed)
- Een tijdsschaal en drempels voor het uitvoeren van de acties
- Een procedure voor escalatie: wie moet gecontacteerd worden en wanneer?

REMEDIATION PLAN

Bij elke aanvraag voor een wijziging moet een **remediation plan** opgenomen worden: een plan met wat er moet gebeuren wanneer een wijziging onsuccesvol is. Stel je voert een update van een programma uit, en na de update zijn er problemen tussen het programma en de databank, je moet de wijziging dus ongedaan maken: In je remediation plan beschrijf je exact de stappen die je moet nemen om de wijziging ongedaan te maken. Houd er ook rekening mee dat niet alle wijzigingen kunnen teruggedraaid worden, zorg in dat geval voor een plan om de impact te beperken. Dit plan moet eerst geëvalueerd worden voor de wijziging kan goedgekeurd worden.

DE LEVENSLOOP VAN EEN NORMALE WIJZIGING

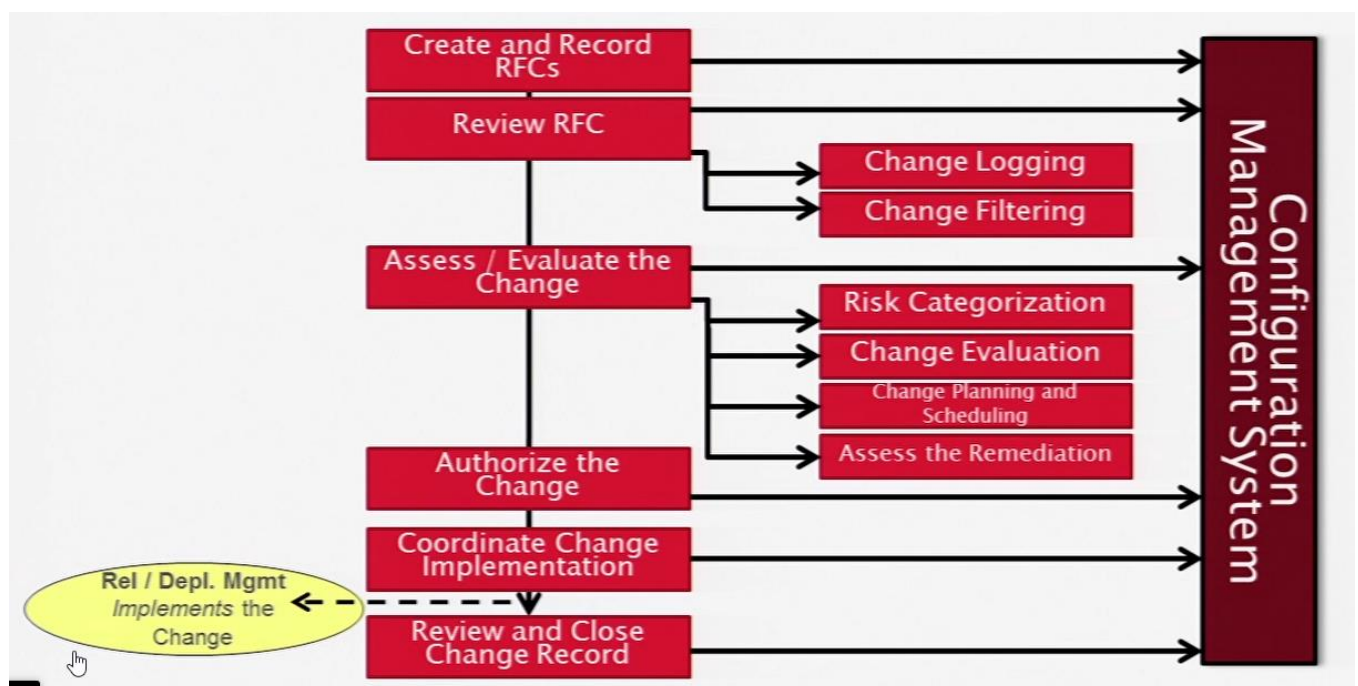
Hierboven heb ik al erg kort beschreven wat er met de normale changes moet gebeuren, in dit deel ga ik dieper in op deze procedure:

Elke change begint met een Request For Change. Wanneer de Change Manager de Request ontvangt doet hij een eerste controle en logt hij de request in het CMS. Bedoeling van deze stap is om er voor te zorgen dat er geen tijd verspilt wordt aan requests die niet uitgevoerd kunnen worden. De Change Manager kan beslissen om een wijziging niet uit te voeren omdat onder andere de wijziging praktisch onhaalbaar is, of niet belangrijk genoeg om tijd aan te besteden, of de juiste personen geen toestemming hebben gegeven voor de wijziging. (Een wijziging die invloed heeft op de capaciteit zal bijvoorbeeld eerst aan de Capacity Manager

moeten voorgelegd worden). Na deze stap blijven enkel de aanvragen over die effectief een kans maken om uitgevoerd te worden.

Hierna wordt de wijziging uitvoerig bekeken: enkele stappen die hier moeten gebeuren zijn het inschatten van het risico van een wijziging, de waarde van de wijziging zelf, er wordt een mogelijke planning opgesteld, en het remediation plan wordt geanalyseerd.

Wanneer de Change Manager hierna er nog steeds van overtuigd is dat de wijziging een goed idee is, wordt de wijziging voorgelegd aan het Change Advisory Board. De Change Manager houdt rekening met hun advies, en geeft uiteindelijk zijn goedkeuring aan de change. Aangezien de eigenlijke wijziging door Release and Deployment management wordt gedaan heeft hij verder een coördinerende rol. Wanneer de wijziging is uitgevoerd doet de Change Manager nog een controle (zijn er problemen opgetreden, moeten er verdere aanpassingen gedaan worden?) en sluit hij de change request af.



5.3. Release and Deployment Management

De vorige twee processen deden een heel gedeelte van het voorbereidende werk, maar het echte werk, het effectieve uitrollen van een oplossing, is de opdracht van Release and Deployment Management. Kort gezegd zorgt dit proces voor het bouwen, testen en uitrollen van wijzigingen of nieuwe systemen.

Inbegrepen in Release and Deployment Management zijn alle processen, systemen en functies die nodig zijn om een dienst die beschreven is in een Service Design Package te bouwen, testen en uit te rollen en de dienst over te dragen naar Service Operation. Release Management is dan ook verantwoordelijk voor alle Configuration Items van een oplossing, zoals de fysieke middelen (servers, computers, ...), de virtuele middelen (virtuele servers, cloud storage, ...), alle applicaties en software, training voor gebruikers en personeel, diensten met inbegrip van alle gerelateerde contracten en overeenkomsten. Vooral deze laatste twee worden al te vaak vergeten.

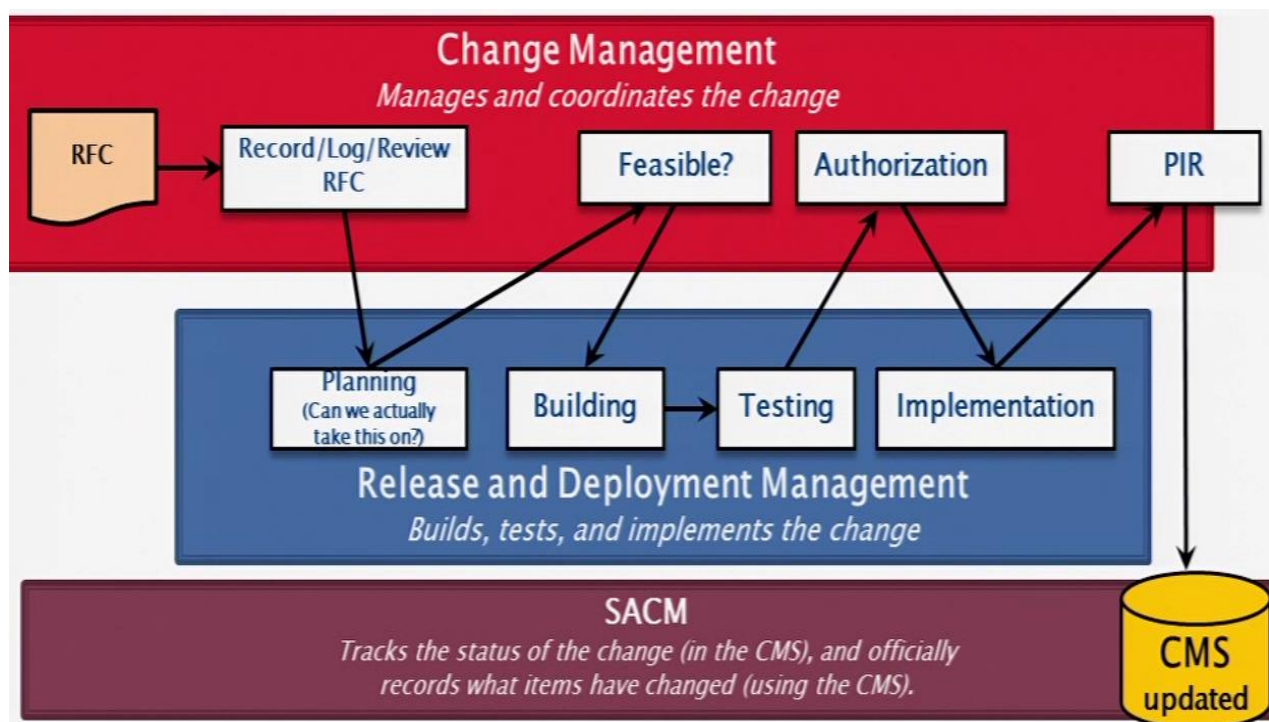
De officiële doelen volgens ITIL zien er als volgt uit:

- Het maken, testen en uitrollen van releases in de DML in een productieomgeving volgens een vooraf overeengekomen plan en planning.
- Het opstellen van en een akkoord bereiken over deployment management plannen met klanten en stakeholders
- Er voor zorgen dat alle release packages traceerbaar zijn, geïnstalleerd, getest, geverifieerd of ongeïnstalleerd kunnen worden, en dat er indien nodig een rollback kan gebeuren
- Er voor zorgen dat nieuwe of gewijzigde diensten en alle ondersteunende onderdelen het overeengekomen nut en waarborg kunnen leveren
- Kennisoverdracht:
 - Naar klanten en gebruikers zodat ze de dienst kunnen gebruiken
 - Naar Service Operation zodat zij de geleverde oplossing kunnen beheren en ondersteunen.

Het laatste puntje lijkt misschien minder belangrijk dan de rest, maar het is de kennisoverdracht die er voor zorgt dat de volgende fase, Service Operation, zijn werk kan doen. Je kan wel een nieuwe applicatie uitrollen, maar als support niet weet hoe de applicatie werkt kunnen ze ook gebruikers niet correct helpen met eventuele problemen. Jammer genoeg gebeurt het nog te vaak dat de overdracht niet vlot loopt. Dit is de schuld van Release

Management dat de documentatie of training moet voorzien, maar ook Change Management dat de wijziging nooit had mogen goedkeuren zonder dat het nodige werk voor de overdracht is uitgevoerd. Change Management heeft een controlerende taak voor Release Management.

Service Transition is dus eigenlijk een samenwerking tussen de voorgaande drie processen: Change Management heeft een overkoepelende functie en heeft de eindverantwoordelijkheid voor elke uitgerolde wijziging. Zij bepalen wat er moet uitgevoerd worden. Release and Deployment Management bouwt de wijziging, test deze en rolt die uit in productie. Configuration Management houdt de status van de wijzigingen bij en documenteert wijzigingen aan Configuration Items.



Figuur 2 PIR is Post Implementation Report

5.4. Knowledge Management

Een vierde process uit Service Transition is Knowledge Management. Het doel van Knowledge Management is er voor zorgen dat informatie, kennis en ervaring zo vlot mogelijk doorstromen tussen verschillende delen van het bedrijf, en de nodige informatie beschikbaar is voor de juiste mensen en op het juiste moment, zodat de juiste beslissingen kunnen genomen worden.

Officieel zijn de doelen:

- De kwaliteit van management beslissingen verbeteren door er voor te zorgen dat kennis, informatie en data beschikbaar is gedurende de hele levenscyclus.
- het bijhouden van een Service Knowledge Management System (SKMS) dat informatie beschikbaar stelt die geschikt is voor elk publiek doorheen de hele organisatie van de dienstverlener.

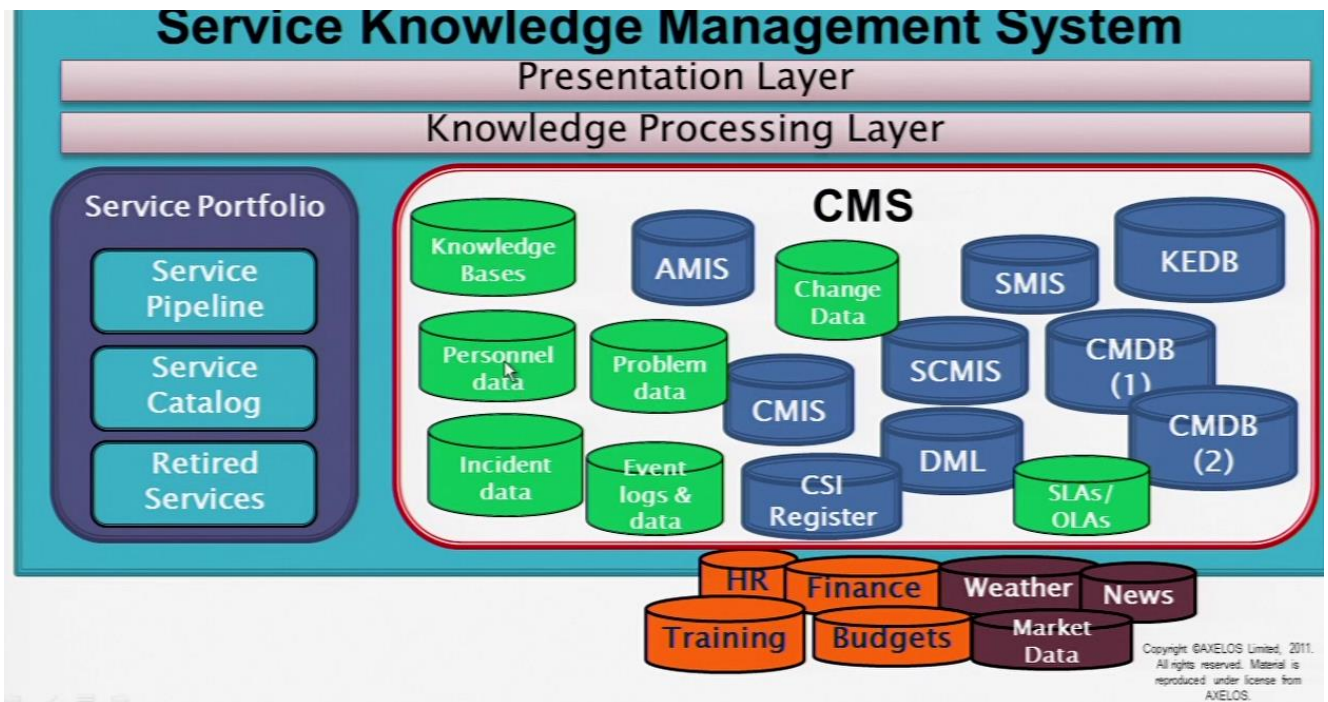
Er is een verschil tussen **data, informatie en kennis**:

Data is meestal het resultaat van een meting: het aantal gebruikers op een moment, de verbruikte bandbreedte, het aantal telefoontjes naar een Service Desk. Op zichzelf heeft data weinig betekenis. Je kan het niet gebruiken om beslissingen te nemen.

Pas wanneer je verschillende datapunten kan je een relatie tussen de punten zien en kom je tot informatie. Bijvoorbeeld: De meeste telefoontjes naar de Helpdesk gebeuren tussen 9u en 12u. Vanuit de relatie tussen de datapunten kan je een conclusie trekken. Al is het nog niet genoeg om beslissingen te nemen.

Kennis krijg je door informatie te combineren met ervaring en inzicht. Je weet dat de Service Desk in de namiddag minder oproepen krijgt, en uit ervaring weet je dat de Service Desk medewerkers liever andere dingen doen aan de telefoon zitten. Je kan dan de beslissing nemen om een deel van de medewerkers in de namiddag een andere taak uit te laten voeren. Wanneer je zoveel kennis hebt dat je een expert wordt over een bepaald onderwerp dat de kennis instinctief wordt, dan spreek je over wijsheid.

Het Service Knowledge Management System is de databank der databanken in ITIL: Het is een overkoepelende databank waarin onder andere de Service Portofolio en het Configuration Management System zijn opgenomen. Daarnaast zijn er ook databanken rond financiën, personeel, ... alle externe databronnen die je kan gebruiken om nuttige verbanden te leggen.



Data haal je uit de individuele databanken: één van de CMDBs, De Capacity Management Database, De CMS neemt deze individuele data en brengt verbanden aan zodat je tot informatie komt. De SKMS combineert informatie uit al deze bronnen om tot kennis te komen. Deze kennis kan gebruikt worden om beslissingen te ondersteunen.

5.5. Transition Planning and Support

Net als in de Design fase Design Cooperation een ondersteunende functie had voor de design-fase, is Transition Planning and Support verantwoordelijk voor het begeleiden van de Transition fase.

Ze is verantwoordelijk voor:

- Het bepalen van beleid, standaarden en modellen voor service transition activiteiten en processen
- Elke grote wijziging of nieuwe dienst door het hele Transition Proces begeleiden
- Het coördineren van alle hulpbronnen zodat verschillende projecten tegelijkertijd door Service Transition geleid kunnen worden.
- Prioriteiten toekennen aan conflicterende behoeften aan Service Transition hulpbronnen.

Ze zijn niet bezig met de gedetailleerde planning van individuele wijzigingen.

5.6. Het resultaat van Service Transition

Het resultaat van Service Transition is erg eenvoudig: Een werkende dienst die in de Service Operation-fase kan gebruikt worden.

Het is mogelijk dat de uiteindelijk afgeleverde oplossing nog gekende bugs bevat. Dit is op zich geen probleem, zolang IT de bugs documenteert en een datum voorziet waarop de bugs opgelost worden. De klant moet hier mee akkoord gaan voor het uitrollen van de dienst

Naast de applicatie zelf is het ook belangrijk dat de overdracht naar de volgende fase goed verloopt. Hiervoor zijn de volgende gegevens nodig:

- Alle documentatie die beschrijft hoe de dienst er uit ziet nu die operationeel is.
- Training voor de gebruikers
- Training voor het ondersteunende personeel
- Een draaiboek voor het operationeel houden van de dienst

HOOFDSTUK 6: SERVICE OPERATION

Het vierde boek van ITIL beschrijft alle functies die te maken hebben met het operationeel houden van een dienst. De dienst is ontworpen in Service Design, gebouwd en uitgerold in Service Transition. Wanneer de dienst in productie is valt ze onder de verantwoordelijkheid van Service Operation.

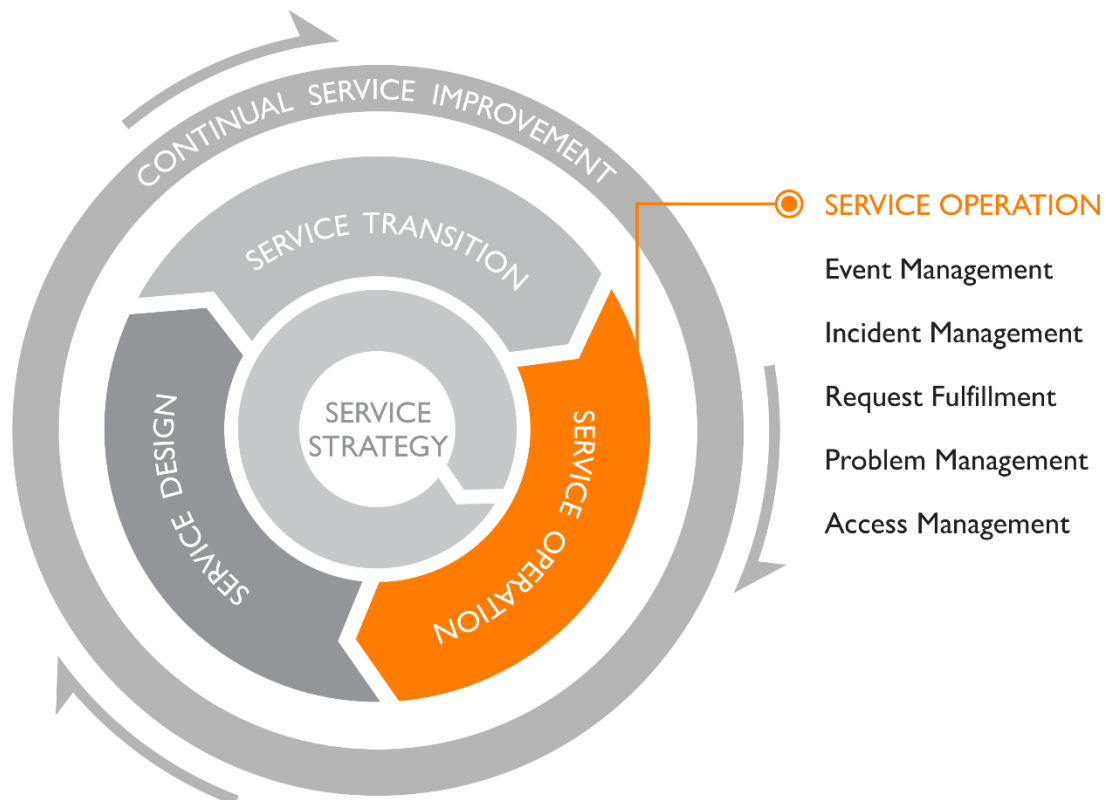
De doelen van Service Operation:

- Het coördineren en uitvoeren van activiteiten en processen die nodig zijn om een dienst te leveren en beheren, volgens de overeengekomen eisen van de klant en eindgebruikers.
- Het beheren van de technologie om de dienst te leveren en ondersteunen.
- De impact van storingen beperken.
- Er voor zorgen dat toegang tot de dienst enkel gegeven wordt aan personen die geautoriseerd zijn om de dienst te gebruiken.
- Zorgen voor gegevens omtrent het functioneren van de dienst.

Service Operation is ook de fase in de levenscyclus waar een dienst iets oplevert voor een klant. De dienst die hij aangevraagd heeft is eindelijk in gebruik, en voor de klant zal het goede verloop van deze fase een grote impact hebben op wat de waarde van de dienst is. De waarde van dit proces is dan ook evident. Enkele voordelen voor het bedrijf:

- Kosten terugbrengen door het efficiënt behandelen van het uitliggen van een dienst en de oorzaken aan te pakken.
- De duurtijd en frequentie van pannes worden beperkt.
- Er wordt data verzameld over de dienst, zodat beslissingen over investeringen voor verbeteringen onderbouwd kunnen worden.
- Er wordt gezorgd voor snelle en effectieve toegang tot IT-diensten
- Het automatiseren van sommige taken, zodat personeel kan focussen op belangrijker werk.

In de Service Operation-fase is communicatie erg belangrijk, meer nog dan in de andere fasen in de levenscyclus. Enkele voorbeelden van mogelijke communicatie zijn informatie over wijzigingen in de dienst, project updates, communicatie rond storingen of onderhoud,

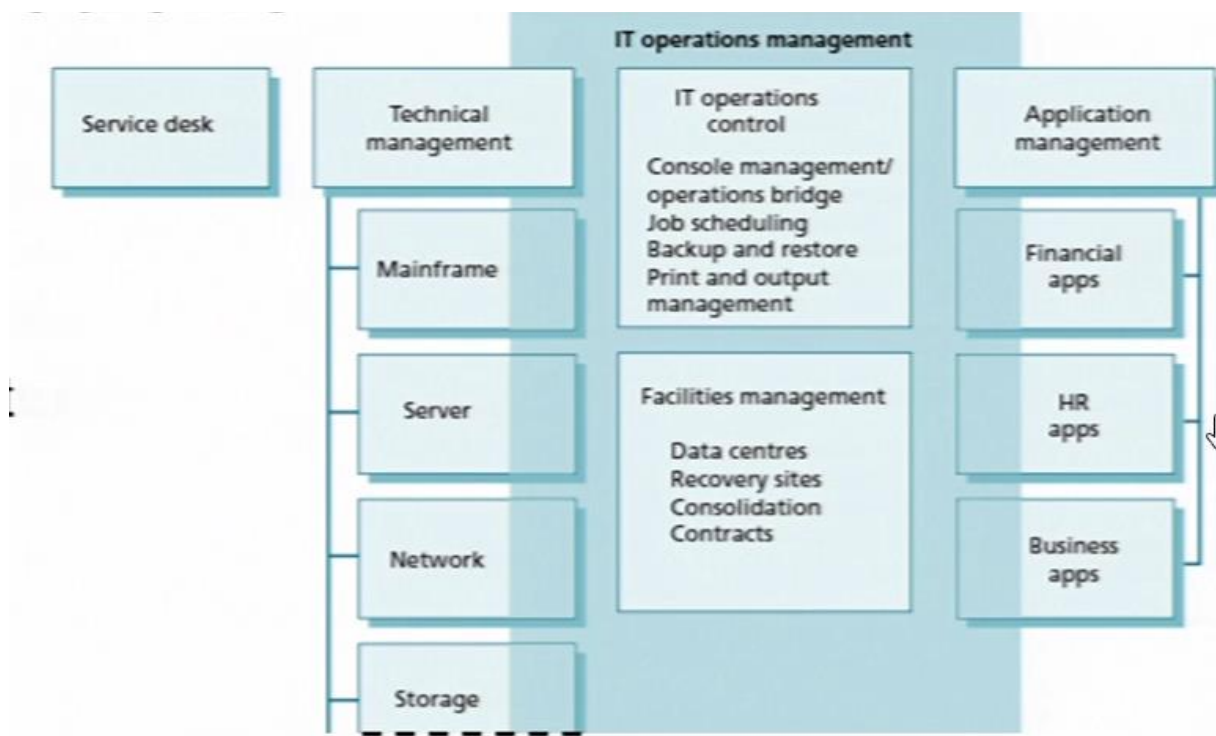


Training, rapporten over de prestaties van de dienst, ... Omdat communicatie in deze fase zo belangrijk is, is het goed om enkele richtlijnen rond communicatie te bekijken:

Ten eerste moet alle communicatie **een doel** hebben, of een bepaalde reactie veroorzaken. Iedereen kent wel mensen die om het minste een e-mail sturen. Het probleem hiermee is dat het soms moeilijk is om de belangrijke informatie te onderscheiden van “achtergrondruis”. Wanneer je taken communiceert, zorg dan ook dat het onmiddellijk duidelijk is wie verantwoordelijk is voor het uitvoeren van de taak. Daarnaast is het ook belangrijk dat je een duidelijk publiek voor ogen hebt in je communicatie, en je informatie ook enkel aan de personen bezorgd die nut hebben aan de communicatie. Technische informatie moet je niet doorsturen naar gewone eindgebruikers, het uitschakelen van een dienst voor onderhoud moet je niet naar het hele bedrijf sturen, enkel naar de gebruikers die die bepaalde dienst gebruiken. Om alles te stroomlijnen is het nuttig om regels omtrent communicatie op te nemen in **Standard Operating Procedures**. Bijvoorbeeld: alles omtrent wijzigingen aan een dienst moet via e-mail gecommuniceerd worden, bij het overdragen van een shift moet volgende informatie doorgegeven worden,

6.1. De Functies binnen ITIL

In het tweede hoofdstuk heb ik al kort de verschillende functies opgesomd die door ITIL beschreven worden: Technical Management, Application Management, IT Operations Management (opgesplitst in IT Operations Control en Facilities Management) en de Service Desk. Omdat het hoofddeel van het werk van deze diensten binnen Service Operation gebeurt, is het nuttig om deze functies in iets meer detail te bekijken. In ITIL betekent een functie een groep van mensen en hun hulpmiddelen om bepaalde processen uit te voeren.



In het bovenstaande schema zie je de vier functies schematisch weergegeven: Service Desk is de helpdesk, zij zijn verantwoordelijk voor het ondersteunen van gebruikers van een dienst. Technical en Application Management zijn de mensen die verantwoordelijk zijn voor de hardware enerzijds en de applicaties anderzijds. IT Operations Management zijn de mensen achter de schermen die het werk van Technical en Application Management mogelijk maken, zij werken vooral in opdracht van de Technical en Application Management.

Hoe groter je bedrijf, hoe duidelijker het onderscheid tussen de verschillende functies is. Binnen kleinere bedrijven is het perfect mogelijk dat een personeelslid taken heeft binnen verschillende functies.

TECHNICAL MANAGEMENT

Binnen Technical Management vindt je de mensen met de technische capaciteiten en de middelen om de IT-infrastructuur te ondersteunen. Ze zijn de bron van alle technische kennis en expertise gerelateerd met het beheer van de IT-infrastructuur. Ze zijn betrokken bij alle fasen van de levenscyclus van een dienst. Ze ontwerpen in technische details van een dienst in Service Design, ze maken de infrastructuur klaar voor het uitrollen van een dienst tijdens Service Transition, en ze houden de dienst draaiend door ze technisch te ondersteunen in Service Operation. Meer algemeen plannen, implementeren en onderhouden ze de infrastructuur die een bedrijf nodig heeft om zijn doelen te bereiken.

APPLICATION MANAGEMENT

Wat Technical Management doet voor de infrastructuur-kant, doet Application Management voor de Applicatie-kant. Ze zijn verantwoordelijk voor het beheren van applicaties doorheen de levenscyclus, en het beheren en ondersteunen van operationele applicaties. Zij zijn de bron van kennis en expertise als het gaat over applicaties. Ze ontwerpen de applicatie in de design fase, bouwen de applicatie, beheren de applicaties als ze operationeel zijn. Meer algemeen zorgen ze voor de functionaliteit die een bedrijf nodig heeft om zijn doel te bereiken.

IT OPERATIONS MANAGEMENT

IT Operations Management is verantwoordelijk voor het dagdagelijkse, achter de schermen beheer van de IT infrastructuur.

IT Operations Control heeft als hoofdtaak het monitoren van de infrastructuur, en het uitvoeren van onderhoudstaken aan de infrastructuur. Voor grote netwerken valt hier bijvoorbeeld het **Network Operation Center** onder, ook dingen als back-ups behoren tot hun verantwoordelijkheden. **Facilities Management** is de subgroep van IT Operations Management die verantwoordelijk is voor het fysieke onderhoud van de infrastructuur. Data Centers, Server Rooms bijvoorbeeld vallen onder hun bevoegdheid: het beheer van bijvoorbeeld stroomvoorziening, en koeling.

SERVICE DESK

Een erg belangrijke rol binnen Service Operation is weggelegd voor de Service Desk. De Service Desk is het enige punt van contact tussen de eindgebruiker en de IT-dienst verlener. De Service Desk zal proberen al de vragen van de gebruikers te beantwoorden. Als ze de vraag zelf niet kunnen beantwoorden, dan escaleren ze het probleem naar een van de andere functies. (De communicatie blijft wel door de service desk lopen).

6.2. De Service Desk

De Service Desk is nauw verwant met enkele processen: uit de Service Operation-fase spelen ze een belangrijke rol binnen Incident Management, Een groot deel van hun werk is het oplossen van problemen voor gebruikers. Een ander proces waar ze bij zijn betrokken is Request Fulfillment Management. Ze moeten op de hoogte zijn van de SLA, dus zijn ook betrokken bij Service Level Management, en moeten een up-to-date kalender hebben met ingeplande wijzigingen (Change Management).

De Service Desk heeft de volgende taken:

- Het loggen van alle relevante incidenten of aanvragen
- Eerstelijns onderzoek en diagnose van problemen
- Problemen oplossen en aanvragen beantwoorden waar mogelijk, en escaleren naar andere diensten wanneer het niet mogelijk is.
- Gebruikers op de hoogte houden van de status van hun probleem of aanvraag
- Het afsluiten van alle opgeloste problemen en aanvragen
- De tevredenheid van gebruikers meten.

Een incident in ITIL is iets dat niet werkt zoals verwacht, een (dienst)aanvraag is wanneer een gebruiker iets nodig heeft van IT (Bijvoorbeeld toegang tot een document, ...)

Er zijn vier soorten Service Desks gedefinieerd in ITIL:

Een **lokale service desk** is een service desk die zich fysiek op de locatie van de gebruiker bevindt. Ze lossen enkel problemen op binnen deze locatie. In veel bedrijven is dit bijvoorbeeld de IT-dienst die je kan bellen om een probleem met bijvoorbeeld een printer op te lossen. Het grote voordeel is de bereikbaarheid, en de kennis die de service desk heeft van het bedrijf waar ze gevestigd zijn, nadeel is de kostprijs.

Een **gecentraliseerde service desk** is een service desk die verantwoordelijk is voor meerdere locaties. Door de service desk te centraliseren is het eenvoudiger om de service desk te controleren, en informatie te verspreiden. Nadeel is dat je de nabijheid van een lokale service desk op moet geven.

Een **virtuele service desk** is een service desk waar het niet uitmaakt waar het personeel zit, ze kunnen via toegang tot hetzelfde systeem en met gelijke procedures aan de slag. Meestal is er één telefoonnummer dat wordt doorgeschakeld naar verschillende medewerkers. Ook van thuis uit werken wordt op deze manier mogelijk. Een variant hierop is een 'Follow The Sun'-model waar medewerkers wereldwijd verspreid zijn, om op die manier 24/7 open te blijven zonder nachtwerk.

Een Service Desk is niet eenvoudig om juist te krijgen, enkele veel voorkomende problemen zijn:

- Een hoog verloop van personeel, hoge mate van stress en burn-out, slechte motivatie van het personeel. Een helpdeskmedewerker zit non-stop in de vuurlijn, ze vangen de problemen en irritaties van klanten op. Een goede manager moet hier rekening mee houden en proberen toch een aangename sfeer te creëren.
- Gebrek aan respect en vertrouwen voor de service desk van de gebruikers, of de andere diensten. De Service Desk blijft verantwoordelijk voor een incident, ook als het is doorgegeven aan een andere dienst. Wanneer de andere diensten de service desk niet serieus nemen, kan dat problemen opleveren voor het oplossen van tickets. De Service Desk moet er op kunnen rekenen dat de andere diensten voldoende informatie leveren zodat ze hun werk kunnen doen.
- Gebruikers en ondersteunende diensten werken buiten de service desk om.

6.3. Event Management

Een eerste proces uit Service Operation is Event Management. Het doel van Event Management is eenvoudig genoeg het detecteren, opvolgen en reageren op gebeurtenissen. Er moet een systeem aanwezig zijn om gebeurtenissen te detecteren, bijvoorbeeld metingen. De gebeurtenis moet geanalyseerd worden, en de juiste actie moet uitgevoerd worden.

Event Management kan toegepast worden op alle vlakken van service-beheer die gecontroleerd moeten worden, en geautomatiseerd kunnen worden:

- Configuration Items: bijvoorbeeld het automatisch pingen van een server om te controleren of deze reageert.
- Omgevingsvoorwaarden: Temperatuursensor in een serverlokaal, rookmelder, ...
- Softwarelicenties: het toewijzen van licenties, controleren of de voorwaarden niet overtreden worden.
- Netwerkbeveiliging: het detecteren van een indringer op het netwerk
- SLA-doelen: een ticket dat niet is opgelost, reactietijd van een dienst, ...

Een **event** (gebeurtenis) is volgens ITIL een verandering in de status van een Configuration Item of IT-dienst die belangrijk is voor het beheer ervan. ITIL onderscheidt drie soorten events: **informatieve events** zijn gebeurtenissen die tijdens de normale werking van een dienst voorkomen. een **waarschuwing** geeft aan dat er iets ongewoons is gebeurd bij een dienst maar de dienst zelf werkt nog, en een **uitzondering** geeft aan dat er iets abnormaals is gebeurt met een impact op de dienst. Een uitzondering is vaak de aanleiding voor een incident of een change. Sommige events leiden tot een **alarm**: een bepaalde drempel is overschreden. Een alarm moet altijd gevolgd worden door een actie.

6.4. Incident Management

Het doel van Incident Management is het zo snel mogelijk terug te keren naar normale dienstverlening, en de impact van een incident op de dienstverlening van een bedrijf te beperken, om op die manier er voor te zorgen dat de overeengekomen voorwaarden van de SLA behaald worden. Normale dienstverlening is dienstverlening zoals beloofd binnen de SLA.

Het is dus belangrijk een goede SLA te schrijven, dit bepaalt of iets een incident is of niet. Als je bijvoorbeeld in je SLA hebt staan dat gebruikers moeten kunnen printen tussen 7 uur 's morgens en 7 uur 's avonds. Belt iemand met een printerprobleem om 8 uur 's avonds dan is dit geen incident. Als een gebruiker overdag niet kan printen dan is het wel een incident. Stel dat je het onderzoekt en je merkt dat de printer A niet werkt, maar printer B wel dan kan je de gebruikers laten printen op printer B en ben je weer in orde met je SLA en kan je het Incident afsluiten. Het is de bedoeling van Incident Management om zo snel mogelijk terug te keren naar een situatie die aanvaardbaar is volgens de SLA. Incident Management zal niet proberen de kapotte printer te herstellen, als een andere optie sneller is. Het effectief herstellen van de kapotte printer is voor Problem Management. Problem Management is traag en gaat op zoek naar de bron van een probleem, Incident Management draait rond het snel oplossen van problemen eventueel met een tijdelijke workaround. In dit voorbeeld wordt

het Incident afgesloten, en een probleem-ticket opgemaakt zodat iemand de printer kan herstellen.

Incident-tickets worden door de Service Desk afgehandeld. Een probleem is de oorzaak die aan de grond ligt van incidenten. Wanneer de Service Desk een incident niet zelf kan oplossen (bijvoorbeeld omdat er maar een printer is) dan wordt een probleem gemeld, en doorgegeven aan de verantwoordelijke dienst. Niet elk incident leidt automatisch tot een probleem. Als een gebruiker belt omdat hij een Blue Screen of Death heeft gekregen op zijn laptop, en de Service Desk heeft gevraagd om te rebooten en het probleem is opgelost dan is het waarschijnlijk niet nuttig om een Problem Ticket aan te maken. Wanneer ditzelfde probleem meerdere keren per week voorkomt is het daarentegen wel noodzakelijk dat een PC-techniker de laptop nakijkt, en moet er dus een Problem Ticket aangemaakt worden.

In ITIL wordt een **incident** gedefinieerd als een ongeplande onderbreking van een IT-dienst, of een vermindering van de kwaliteit van een dienst. Een fout met een Configuration item dat nog geen impact heeft op de dienst is ook een incident. Wanneer een DNS-server uitvalt, en de backup DNS-server overneemt, dan zal de gebruiker dit niet merken. Toch moet hiervoor een incident aangemaakt worden.

Een **incident model** is een vooraf opgesteld stappenplan voor veel voorkomende problemen. Het bevat:

- Een chronologische volgorde van de stappen die uitgevoerd moeten worden
- Verantwoordelijkheden: Wie doet wat? (bijvoorbeeld met een RACI-chart)
- Voorzorgsmaatregelen die genomen moeten worden voor het incident wordt opgelost (bijvoorbeeld backup van gegevens)
- Een planning en deadlines voor de verschillende stappen
- Escalatieprocedure: wie moet gecontacteerd worden, en wanneer?
- Stappen voor eventueel bewijs (screenshots, logs)

Een **major incident** (groot incident) is een incident dat een groot aantal gebruikers treft, of een impact heeft op een kritieke dienst. Major Incidents krijgen een hogere prioriteit, hebben een kortere behandeltijd en aparte procedures.

Samenvattend zijn incidenten dus alle gebeurtenissen die de dienstverlening onderbreken, of dreigen te onderbreken. Ze worden aan de Service Desk gecommuniceerd door gebruikers, IT-werknemers, of geautomatiseerde monitoring tools. Incident Management is voor klanten en gebruikers het meest zichtbare gedeelte van Service Operation, en hierdoor is het eenvoudig om de waarde er van aan te tonen. Incident Management is dan ook een van de eerste ITIL processen die veel bedrijven installeren.

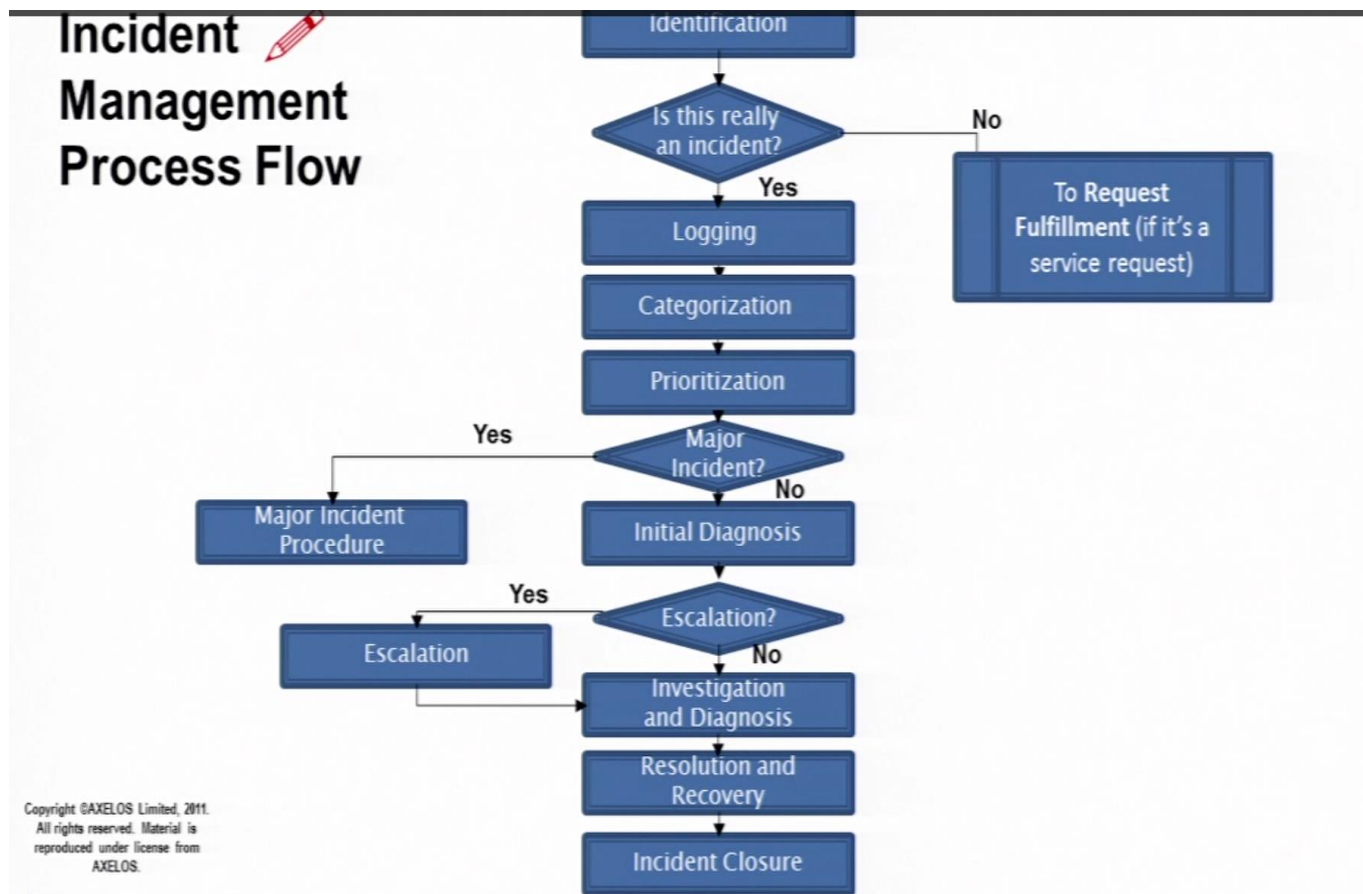
PRIORITEITEN

Een van de moeilijkste taken voor een Service Desk is het correct inschatten van een prioriteit voor een incident. Veel bedrijven hebben de neiging om van elk incident een “prioriteit 1 incident” te maken, waardoor de echte belangrijke incidenten verdrongen worden door kleinere problemen. De prioriteit van een incident bepaalt de volgorde waarin incidenten worden opgelost.

Om een correct inschatting te kunnen maken van een incident moet je naar twee dingen kijken: de **impact** is het effect van een incident op de bedrijfsvoering, de mate van afwijking van de SLA. Met andere woorden: Hoeveel schade brengt het het bedrijf toe. De **dringendheid** wordt bepaald door hoe lang het duurt voor een incident een voelbare impact heeft op het bedrijf (hoe snel moet dit hersteld worden?). Deze twee factoren vormen samen de basis van de prioriteit.

Het bepalen van de Prioriteit is één van de hoofdtaken van de Service Desk. Dit is een taak die niet in een vacuüm uitgevoerd kan worden: door te luisteren naar de klant, en kennis van de bedrijfssituatie kan de service desk medewerker een correcte inschatting maken. Veel service desk-software kent automatisch prioriteiten toe, toch is het een goed idee om dit steeds na te kijken, en eventueel de prioriteit te wijzigen.

INCIDENT MANAGEMENT PROCESS FLOW



Incident Management verloopt volgens een vast plan, de stappen zijn logisch en ook buiten ITIL werken de meeste bedrijven op een vergelijkbare manier. De eerste stap is het identificeren van een Incident: Een gebruiker of lid van het IT-team contacteert de service desk met een probleem: Een gebruiker belt omdat iets niet werkt, een van de netwerkbeheerders merkt dat een router slecht werkt, een automatisch systeem genereert een error... De Service Desk moet dan een beslissing nemen: Is de gebeurtenis een incident of niet? Als een gebruiker de Service Desk contacteert om bijvoorbeeld te vragen toegang te krijgen tot een extra dienst, of een klant belt zijn provider om te vragen extra kanalen toe te voegen aan zijn pakket dan gaat het om een diensaanvraag, dat in een apart proces behandeld wordt (Request Fulfillment).

Wanneer duidelijk is dat het effectief om een incident gaat, dan moet het eerst en vooral gelogd worden in het systeem. **Alle incidenten moeten volledig gelogd worden!** In deze log zit een beschrijving van het probleem, de gebruiker die het gemeld heeft, het tijdstip, ... Alles wat nodig kan zijn om een oplossing te vinden. Het incident wordt gecategoriseerd (is het een hardware-probleem? software? beveiliging?) en er wordt een prioriteit aan toegekend (combinatie van impact en dringendheid). Pas wanneer er een prioriteit op geplakt is kan de

Service Desk beslissen of het over een Major Incident gaat. Als dat zo is wordt de Major Incident procedure opgestart.

In een volgende stap wordt een eerste diagnose van het probleem gemaakt: is dit iets wat de Service Desk zelf kan oplossen, of moet het probleem ge-escaléerd worden (omdat er specifieke kennis nodig is, of omdat de service desk niet de juiste bevoegdheden heeft om het probleem op te lossen.), het incident wordt verder onderzocht en wordt een diagnose gesteld. De oplossing wordt uitgevoerd, en de dienstverlening wordt hersteld. Pas wanneer het incident opgelost is, dan wordt het incident afgesloten door de Service Desk. Hiervoor neemt de Service Desk contact op met de oorspronkelijke melder van het probleem om te controleren of hij tevreden is. Voor het sluiten is het ook een goed idee om de categorie na te kijken, en er voor te zorgen dat alles goed gedocumenteerd is. Eens een incident is afgesloten door de Service Desk kan het niet meer opnieuw geopend worden, dus het is belangrijk dat je zeker bent dat alles is opgelost.

MOEILIKHEDEN BINNEN INCIDENT MANAGEMENT

- Werknemers overtuigen dat alle incidenten gelogd moeten worden

Werknemers kunnen om verschillende redenen geen fan zijn van het aanmaken van tickets voor incidenten. Het probleem met het niet aanmaken van tickets is dat het werk dat je presteert niet gemeten kan worden. Soms is het sneller het probleem gewoon op te lossen dan het hele ticket in te vullen. Er komen pas problemen wanneer management op basis van gesloten tickets beslist hoeveel personeel een dienst nodig heeft. Bovendien wordt het zonder tickets moeilijk om te zien wat veel voorkomende problemen zijn die misschien opgelost moeten worden.

- Voldoende kennis om incidenten snel op te lossen

Te veel bedrijven zien de service desk nog te vaak als de locatie om minder getalenteerd IT-personeel te dumpen. Uiteraard heeft dit een invloed op de prestaties van de service desk. Een ander probleem is de doorstroming van informatie van de technische kant naar de service desk. Als de service desk de nodige informatie niet heeft kan ze ook geen incidenten oplossen.

- Integratie in het Configuration Management System

Om zijn werk goed te kunnen doen moet de Service Desk toegang krijgen tot de CMS, zodat ze bij het diagnosticeren van incidenten toegang heeft tot informatie van de Cis en de relatie tussen de Cis

6.5. Request Fulfillment

Het doel van Request Fulfillment is er voor zorgen dat alle aanvragen van gebruikers behandeld worden. Net als incidenten komen deze aanvragen binnen op de Service Desk. De eenvoudigste manier om een aanvraag te omschrijven is alle vragen die IT krijgt als het systeem werkt, bijvoorbeeld een nieuwe computer plaatsen, een vraag om informatie, een wachtwoord-reset,...). Aanvragen zijn vaak voorkomend, hebben een laag risico en lage kosten.

Request Fulfillment is een nieuw proces in ITILv3. Daarvoor kwamen deze talrijke kleine vragen terecht in ofwel Change Management ofwel Incident Management. Door een nieuw proces toe te voegen worden deze twee processen ontlast.

De doelen van Request Fulfillment zijn de volgende:

- Gebruikers en klanten tevreden maken door alle aanvragen efficiënt en professioneel op te lossen
- Een kanaal creëren voor standaardaanvragen van gebruikers die vooraf goedgekeurd zijn volgens een standaardprocedure
- Klanten en gebruikers op de hoogte houden van de beschikbaarheid van diensten, en de procedure om deze diensten te verkrijgen
- Onderdelen van standaardaanvragen beheren (licenties, software, ...)
- Informatie geven, klachten behandelen en opmerkingen van gebruikers bijhouden.

Elk bedrijf kan zelf beslissen welke wijzigingen in dit proces terechtkomen, en wat door Change Management gedaan wordt.

6.6. Problem Management

Problem Management heeft als doel alle stadia in de levensloop van een probleem op te volgen, van de identificatie, het onderzoeken en documenteren tot de uiteindelijke oplossing. In ITIL is een probleem de onbekende oorzaak van incidenten. Wanneer eenzelfde incident steeds terugkomt kan je steeds de work-around gebruiken (bijvoorbeeld elke keer rebooten als de netwerkkaart crasht (incident)), maar om de irritatie en het tijdverlies te beperken kan je ook op zoek gaan naar de oorzaak, het probleem. Je gaat dus incidenten vermijden door een permanente oplossing te vinden voor de oorzaak van de incidenten.

De doelen van Problem Management in ITIL zijn de volgende:

- De oorzaak van een probleem zoeken
- Een oplossing vinden voor een probleem
- Een RFC indienen zodat het probleem kan opgelost worden via Change Management en Release and Deployment Management
- Problemen en veroorzaakte incidenten vermijden (Proactief Problem Management)
- Terugkerende incidenten vermijden
- De impact van incidenten die niet vermeden kunnen worden beperken.

Problem Management is onzichtbaar voor de gebruikers, het is geen onderdeel van de Service Desk. De Service Desk kan wel Problem Management inschakelen. Problem Management is traag en gaat op zoek om een probleem volledig uit te zoeken en op te lossen, terwijl Incident Management zo snel mogelijk de dienstverlening wil herstellen, eventueel met een work-around. Elk bedrijf heeft eigen regels over wanneer Problem Management ingeschakeld wordt, maar over het algemeen gaat het ofwel om incidenten met een grote impact, of incidenten die regelmatig terugkeren.

Een incident is dus een gebeurtenis die de gebruiker meldt aan de Service Desk. De Service Desk zal proberen dit zo snel mogelijk op te lossen, eventueel met een workaround, een tijdelijke oplossing om de dienstverlening te herstellen. Een probleem is de onbekende oorzaak die aan de basis ligt van incidenten. Een andere term die in Problem Management terugkomt is de **Known Error**, dit is een gekende oorzaak van incidenten. Een probleem is dus een ongekende oorzaak, een known error is een bekende oorzaak.

Een gebruiker belt naar de Service Desk omdat hij niet kan printen op printer A. De service Desk herstelt de dienstverlening door alles wat afgedrukt moet worden op de computer van de gebruiker door te sturen naar printer B, en het incident wordt afgesloten. Wanneer 40 andere gebruikers met hetzelfde probleem bellen, kan de service desk beslissen om een

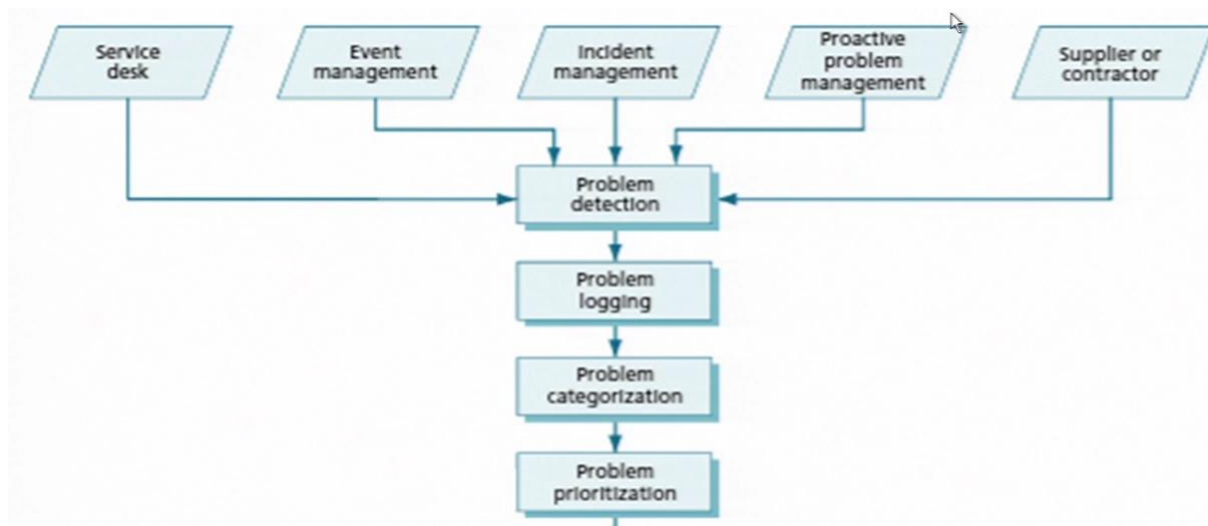
problem ticket aan te maken. (terwijl ze voorlopig de oplossing toepast zodat gebruikers kunnen printen op printer B). Door het problem ticket wordt een techniker ingeschakeld om naar de printer te kijken. Wanneer hij de oorzaak heeft gevonden (een defecte voeding bijvoorbeeld) verandert het probleem in een Known Error.

Zolang een known error niet opgelost is komt hij terecht in de **Known Error Database**. Deze databank bevat alle gekende fouten, en is onderdeel van de CMS. Known Errors kunnen op elk moment in het Problem Management proces aangemaakt worden. Als het een fout is die door de Service Desk gekend moet zijn om incidenten op te lossen, dan moet ze in de databank staan. In de database worden ook de gekende workarounds opgenomen, zodat de Service Desk efficiënt kan werken. Enkel de Problem Manager heeft de bevoegdheid om errors op te nemen om de databank proper te houden.

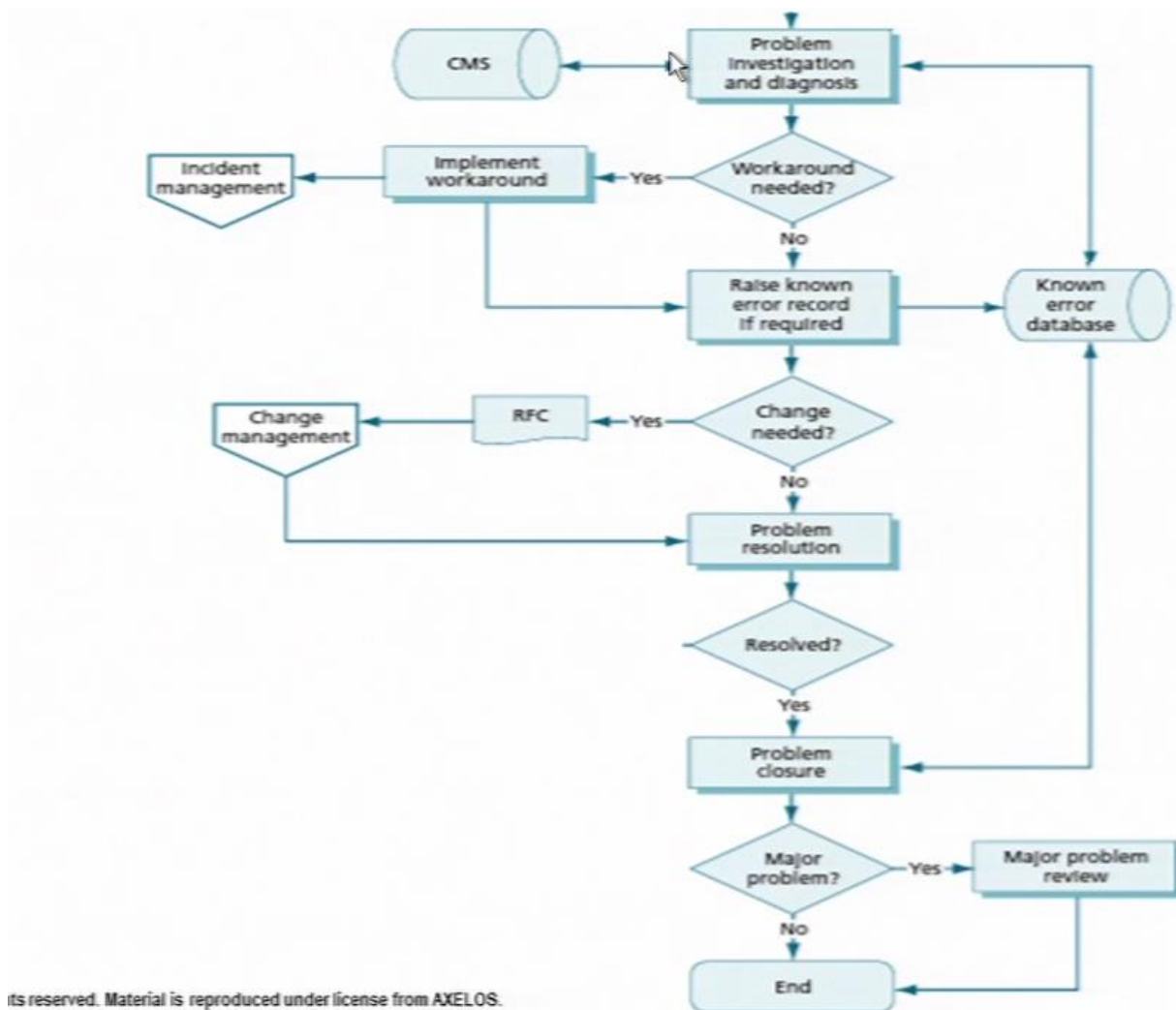
Problem Management kan zowel **reactief** zijn, wanneer je een probleem oplost dat gebruikers op dit moment beïnvloed, dus als reactie op een incident report, of **proactief** wanneer je een probleem oplost omdat je verwacht dat dit gebruikers zal beïnvloeden. Op dat moment is er geen incident, maar je kan wel een problem ticket openen. Bijvoorbeeld wanneer je merkt dat je bedrijf meer en meer bandbreedte verbruikt: op het moment is er nog geen incident, maar als de huidige trend zich verder zet zal er na verloop van tijd een punt komen waarop er door gebrek aan bandbreedte incidenten voorkomen. Je kan dan proactief een problem ticket aanmaken, de oorzaak zoeken en via een change request maatregelen nemen om deze situatie te vermijden. Er hoeft dus niet steeds een incident te zijn om van een probleem te spreken.

Wanneer een problem ticket wordt aangemaakt is grotendeels afhankelijk van de werkwijze van het bedrijf, ITIL stelt hier geen vaste regels voor. Het is wel belangrijk om te onthouden dat een incident geen probleem wordt: het problem ticket wordt veroorzaakt door het incident. Het is dus perfect mogelijk om tegelijkertijd een incident en problem ticket open te hebben. Het incident wordt gemaakt door de Service Desk, het problem ticket door de achterliggende diensten. Incident Management zoekt naar een manier om dienstverlening zo snel mogelijk te herstellen, Problem Management zoekt de bronoorzaak van een incident.

PROBLEM MANAGEMENT PROCESS FLOW



Problem Tickets hebben verschillende bronnen: Ze kunnen van de Service Desk komen, uit Event Management, Proactive Problem Management via een geautomatiseerd meting, ... Elk bedrijf heeft regels over wat een incident (of meerdere incidenten) de moeite maakt om te onderzoeken in Problem Management. Voor sommige bedrijven is het de moeite om achter elk incident op zoek te gaan naar de oorzaak, andere bedrijven beperken zich eerder tot incidenten met een grote impact, of een bepaalde categorie (bijvoorbeeld incidenten met betrekking tot beveiliging). Wanneer een probleem gedetecteerd is, moeten dezelfde stappen doorlopen worden die Incident Management gebruikt: Het probleem wordt gelogd, via een aparte module in de software die door Incident Management gebruikt wordt, of in een Excel bestand, het maakt niet uit zolang het ergens gelogd is. Het probleem krijgt een categorie toegekend (Hier heeft het voordelen als Problem Management dezelfde categoriën gebruikt als Incident Management, omdat het dan eenvoudiger wordt om Incidenten die veroorzaakt worden door een probleem terug te vinden), en het probleem krijgt een prioriteit zodat er keuzes gemaakt kunnen worden over de toekenning van middelen.



Hierna wordt het hele proces iets ingewikkelder. Een eerste stap is het onderzoeken van het probleem en de diagnose. Om een correcte diagnose te stellen kan de CMS een belangrijk hulpmiddel zijn: je kan op zoek gaan naar andere incidenten die te maken kunnen hebben met dit probleem, of je kan nakijken of er recente wijzigingen zijn die het probleem veroorzaken.

In een volgende stap ga je kijken of een workaround mogelijk of wenselijk is. Als dat zo is test je de workaround en geef je dit door aan Incident Management zodat de Service Desk zijn werk kan doen. In beide gevallen geef je het probleem door in de Known Error Database. Denk er ook aan dat een Known Error op elk moment mag aangemaakt worden, ook voor de oorzaak nog niet gekend is. Een goede vuistregel is: maak een known error van af het moment dat het de Service Desk kan helpen. Dit kan dus onmiddellijk bij de detectie van een probleem gebeuren.

Op dit moment weet je ook of er een wijziging moet gebeuren om het probleem op te lossen. Als dat zo is neemt Change Management en meer algemeen Service Transition het over om

de wijziging door te voeren, anders voer je zelf de oplossing uit. Je kijkt of je oplossing het probleem heeft opgelost, en kan het problem ticket afsluiten. Vergeet op dit moment ook niet dat de Problem Manager de Known Error uit de databank moet halen: Deze databank is geen archief, enkel de huidig actieve problemen moeten in deze databank komen zodat ze eenvoudig doorzoekbaar is voor al wie daar belang bij heeft.

Is het ticket afgesloten, dan zijn er in het geval van een Major Problem nog enkele extra stappen. Wat juist een Major Problem inhoudt is net als een Major Incident afhankelijk van bedrijf tot bedrijf, afhankelijk van de impact of frequentie van het probleem. Bij een Major Problem wordt na het oplossen nog een extra overleg ingepland: Enkele vragen die meestal aan bod komen tijdens zo'n review:

- Wat liep er goed/fout bij het oplossen van het probleem?
- Wat kunnen we doen om dit probleem in de toekomst te vermijden?
- Wat moeten we in de toekomst beter doen bij de afhandeling?
- Was een eventuele derde partij verantwoordelijk? Moet dit opgevolgd worden?

Deze vergadering staat onder leiding van de Problem Manager.

Zoals je kan zien werkt Problem Management nauw samen met andere processen om zijn werk te doen: Een typisch probleem komt als incident binnen bij de Service Desk (Incident Management). Afhankelijk van het incident veroorzaakt het incident een probleem. Problem Management gaat op zoek naar de oorzaak, en de oplossing moet aangevraagd worden aan Change Management (als het niet om een standaardwijziging gaat). Als Change Management de wijziging goedkeurt, dan is het aan Release Management om de oplossing te implementeren. Nadat de wijziging is uitgevoerd zal Change Management controleren of het probleem effectief is opgelost door de Service Desk te contacteren (zijn er nog nieuwe incidenten rond dit probleem), en de Configuration Management inlichten over wijzigingen aan Configuration Items.

Dit is het ideale geval waarin elk probleem een oplossing heeft, jammer genoeg is het in de echte wereld niet altijd mogelijk om een probleem op te lossen, bijvoorbeeld omdat er geen budget voor is, of simpelweg omdat er nooit een oplossing gevonden wordt. In dat geval kan de Known Error open blijven staan, met alle bijkomende informatie die toch gevonden wordt, of een reden waarom het probleem niet is opgelost.

6.7. Access Management

Een laatste proces uit Service Operation is Access Management. Eenvoudig gezegd zorgt Access Management er voor dat gebruikers de toegang krijgen tot hulpbronnen die nodig zijn om hun werk uit te voeren. Volgens ITIL:

- Op een efficiënte wijze antwoorden op aanvragen tot toegang, toegangsrechten wijzigen of toegang ontzeggen zodat de rechten die geleverd worden op de juiste manier worden toegepast.
- De praktische toepassing van het beleid en acties bepaald in Information Security Management.
- Het controleren van toegang tot diensten en er voor zorgen dat de rechten niet misbruikt worden.

Het is een erg eenvoudig proces: wanneer een gebruiker toegang vraagt tot gegevens of een dienst wordt gekeken of deze gebruiker daar vanuit zijn functie recht op heeft. Als dat zo is wordt via een RFC (of voor kleinere wijzigingen via Request Fulfillment) een aanvraag gestuurd naar de beheerder van een applicatie om deze toegang te verlenen.

Enkele belangrijke begrippen uit Access Management:

Access (toegang) is het niveau en de mate van toegang van de functionaliteit van een dienst of data waarop een gebruiker recht heeft.

Identity (identiteit) de informatie over een gebruiker die hem op individuele basis onderscheidt, en die hun status binnen de organisatie bepaalt. (Wat wordt er gebruikt om de gebruiker te identificeren, bijvoorbeeld een gebruikersnaam).

Rights (rechten) de eigenlijke instellingen waardoor een gebruiker toegang verkrijgt

Service Groups de meeste gebruikers gebruiken verschillende diensten, en gebruikers met een gelijkaardige functie zullen gelijkaardige diensten gebruiken.

Directory Service een specifieke tool die gebruikt wordt om gebruikersrechten en gebruikerstoegang te beheren (bv Active Directory)

6.8. De resultaten van Service Operation

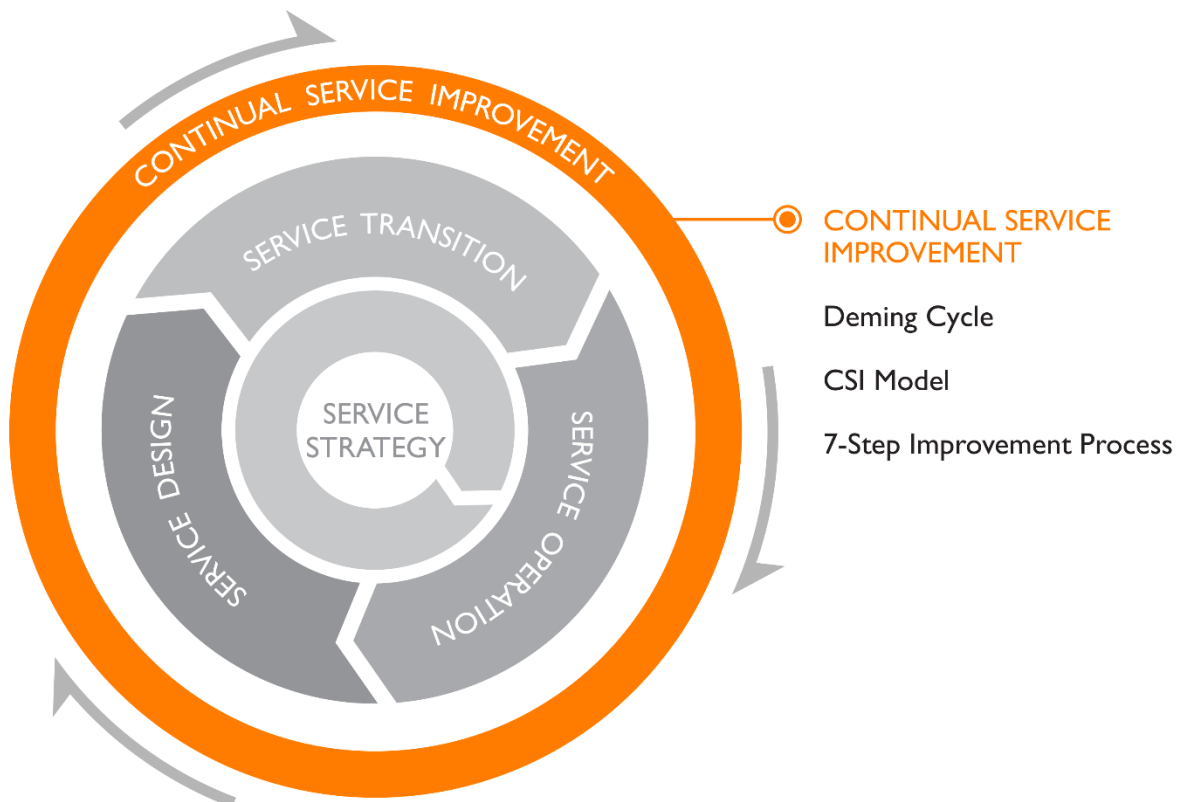
Het belangrijkste resultaat van Service Operation zijn de diensten die aan de gebruikers geleverd worden, op een manier die overeengekomen is in de Service Level Agreement. Service Level Management zal in deze fase zijn rapporten opmaken om te controleren of de voorwaarden van de SLA gehaald worden.

Belangrijk voor de volgende fase, Continual Service Improvement, zijn alle metingen die in deze fase gebeuren van zowel de dienst zelf als alle componenten. Deze resultaten kunnen door andere processen gebruikt worden om hun werk te controleren. CSI zal de resultaten analyseren en kijken of er eventueel verbeteringen aangebracht kunnen worden, waardoor de hele ITIL-cyclus herbegint.

HOOFDSTUK 7: CONTINUAL SERVICE IMPROVEMENT

Continual Service Improvement is het laatste boek van ITIL. Bedoeling van deze stap is controleren of alles verloopt zoals verwacht, en eventuele verbeteringen voorstellen. De verbeteringen doorlopen dan weer de hele ITIL levenscyclus tot ze uiteindelijk in productie zijn. Bedoeling van deze fase is dus steeds kritisch te kijken naar wat je levert, en alle mogelijkheden voor verbetering aan te grijpen. Om deze taak uit te voeren kijkt CSI naar de rapporten die in de Service Operation-fase aangemaakt worden.

Het hoofddoel van CSI is dus er voor zorgen dat de geleverde IT-diensten steeds aangepast zijn aan de wijzigende bedrijfssituatie. Het gaat steeds op zoek naar manieren om de geleverde dienst efficiënter te maken, de bijhorende processen efficiënter te maken, of kosten te besparen. Alles wat te maken heeft met IT-dienstverlening is het onderwerp van Continual Service Improvement. Hierbij is het belangrijk om in het achterhoofd te houden dat je enkel kan verbeteren wat je effectief meet, en dus dat alles wat je niet meet voor je bedrijf onbelangrijk is.



Volgens ITIL zijn dit de taken van CSI:

- Aanbevelingen maken voor verbeteringskansen voor alle fasen in de ITIL-levenscyclus
- Nakijken en analyseren van het bereiken van Service Level-doelen
- Identificeren en implementeren van activiteiten om de IT-dienstverlening te verbeteren
- Kostenefficiëntie verbeteren zonder dat dit ten koste gaat van klantentevredenheid
- Het kiezen en gebruiken van methoden voor kwaliteitsbeheer.
- Begrijpen wat er gemeten moet worden, waarom dat gemeten wordt, en wat succesvolle prestaties zijn.

In het ITIL-boek zijn drie methoden voor kwaliteitsbeheer opgenomen, maar je kan ook een andere methode gebruiken.

De voordelen van CSI voor een bedrijf zijn dan ook evident:

- CSI zorgt ervoor dat de IT-diensten afgestemd blijven op de noden van het bedrijf
- CSI zorgt voor een geleidelijke verbetering in de kostenefficiëntie, zodat ofwel de kosten worden teruggebracht, of het mogelijk wordt om voor de zelfde kost meer werk te verrichten

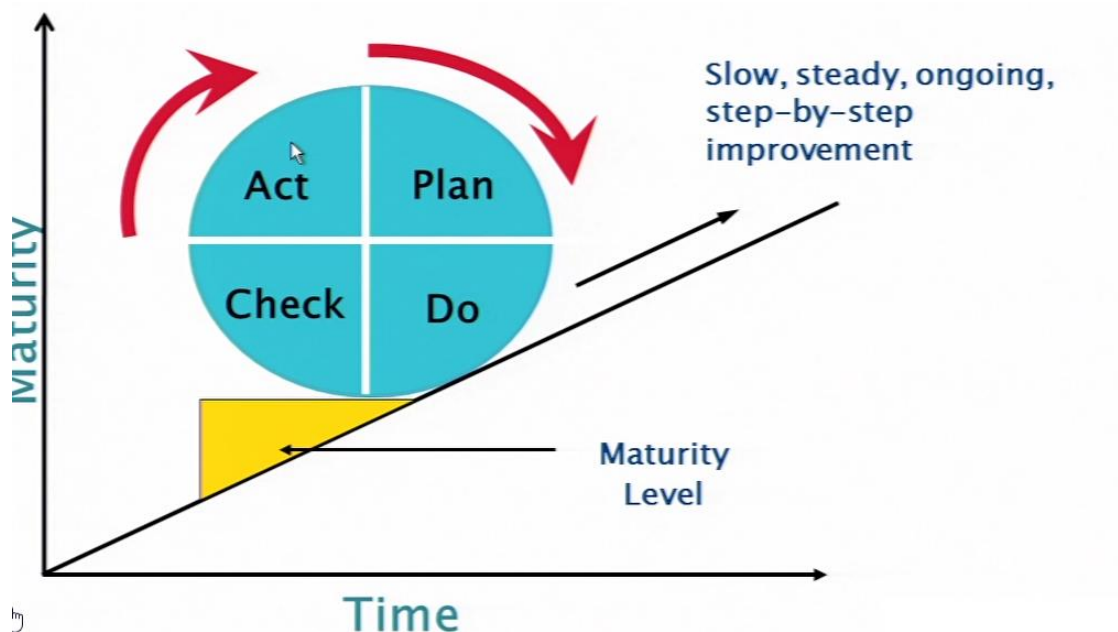
- CSI vindt verbeteringsmogelijkheden in de organisatiestructuur, partners, technologie, capaciteiten van werknemers en training en communicatie.

Alle informatie omtrent initiatieven die CSI onderneemt (of niet onderneemt) vindt je terug in het CSI Register, een onderdeel van het SKMS. Het is de verantwoordelijkheid van de CSI Manager om deze databank bij te houden. Het is een goed idee om alle initiatieven te werken met een indeling: bijvoorbeeld groot, gemiddeld, klein afhankelijk van de moeite die het kost om te implementeren, en een prioriteit.

7.1. De Deming-cyclus

Een eerste kwaliteitsmethode die door ITIL wordt voorgesteld, is ook buiten de IT-wereld erg bekend bij iedereen die ooit betrokken is geweest bij projectmanagement, of zelf al voorstellen heeft gedaan om de kwaliteit van zijn werk te verbeteren.

De cyclus valt uiteen in 4 fasen: **Plan - Do - Check - Act**. Deze methode is vooral erg geschikt voor trage, continue verbeteringen, minder voor grote sprongen.

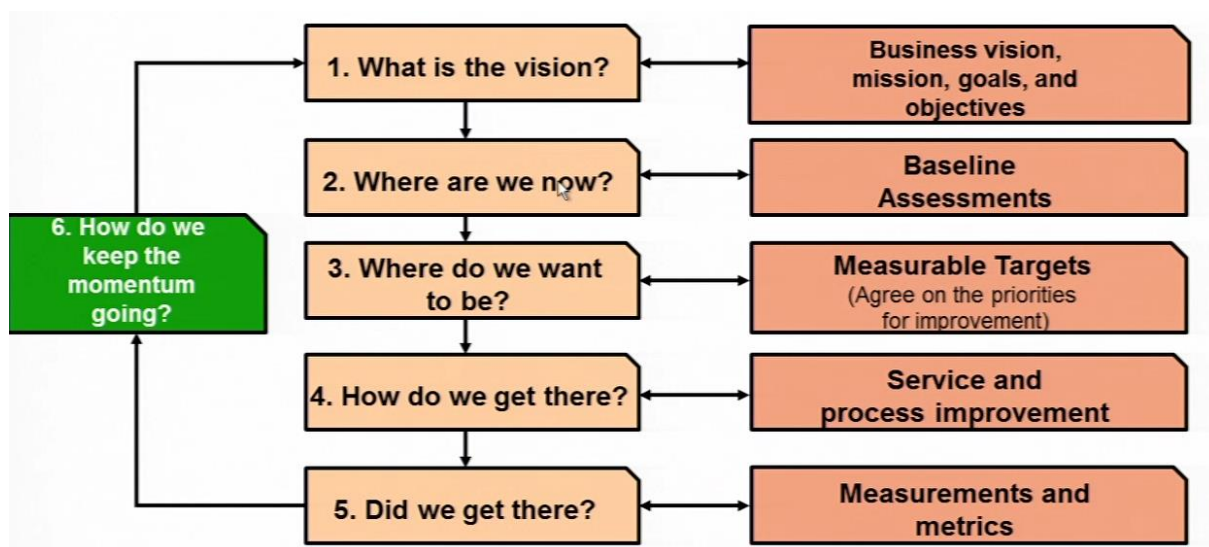


Het idee is vrij eenvoudig: door het proces correct uit te voeren rolt de bal omhoog langs de lijn, en stijgt de maturiteit van de dienst, zodat de bal niet kan terugrollen.

In de eerste stap plan je de verbetering, hierna voer je de verbetering uit, je controleert het resultaat, en afhankelijk van het resultaat onderneem je een actie (bijvoorbeeld het proces opnieuw starten als het resultaat niet is wat je verwacht had.)

7.2. De CSI-methode

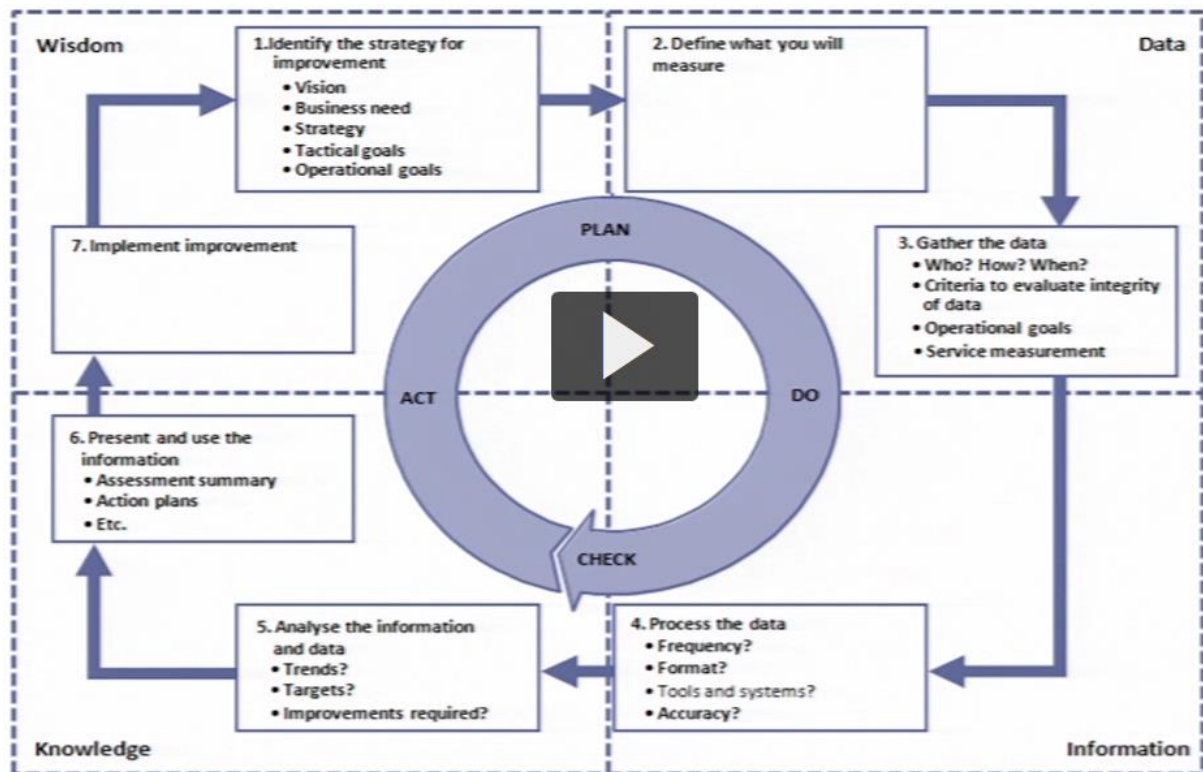
Een tweede methode die ITIL aanreikt is specifiek gericht op IT: De CSI-methode. De methode werkt door op regelmatige tijdstippen algemene vragen te stellen.



De eerste vraag beantwoordt waar je naar toe wil, wat is de visie die je wil bereiken? Om dat doel te kunnen bereiken moet je je afvragen waar je momenteel bent, dit kan bijvoorbeeld door metingen te nemen. Een volgende belangrijke vraag is waar je naartoe wil: hoe weet je wanneer je geslaagd bent in je opzet. Je stelt dus specifieke doelen op die je wil bereiken (denk aan SMART-doelen, deze doelen moeten meetbaar zijn). Stap 4 zoek je een methode om deze doelen te bereiken (dit kan door de dienst te verbeteren, processen te verbeteren, ...) Bij de vijfde vraag ga je na of je de doelen bereikt hebt, door opnieuw dezelfde metingen uit te voeren. De zesde vraag zoekt een antwoord op de vraag hoe je er voor kan zorgen dat je jezelf steeds blijft verbeteren, en start de cyclus opnieuw.

7.3. Het 7-stappen-verbeterings proces

De laatste methode is eigenlijk een combinatie van de twee voorgaande, en het past binnen het data-informatie-kennis-wisdom-concept dat je nog kent uit Knowledge Management.



Ook hier begin je met een strategische visie op wat je wil verbeteren. In stap 2 kies je welke metingen je zal uitvoeren, welke informatie verzameld moet worden. Besteed voldoende aandacht aan deze stap, welke meting je uitvoert kan een grote impact hebben op het uiteindelijke resultaat. In Stap 3 ga je hier dieper op in, door te bepalen hoe de data moet gemeten worden (wie, wanneer, wat is geldige data, ...) In de vierde stap zet je de data om naar informatie (op welke manier wordt de data samengebracht, wat is de nauwkeurigheid die we nodig hebben). Je zorgt ervoor dat je naar trends op zoek kan gaan. In de vijfde stap ga je naar de informatie kijken om er kennis uit te halen: zijn er bepaalde tendensen, bereiken we de doelen, is er verbetering nodig?. In een zesde stap ga je aan de slag met de opgebouwde kennis: je vat samen, stelt eventuele actieplannen op. De laatste stap is de uiteindelijke implementatie van de verbeteringen. Hierna begin je weer opnieuw (kan het nog beter? zijn er andere zaken die we kunnen aanpakken.)

7.4. De praktijk van CSI

Zoals je uit het voorgaande kan opmaken is er binnen CSI een belangrijke rol weggelegd voor metingen. Binnen CSI zijn er 2 verschillende soorten metingen: **Critical Succes Factors (CSF's)** en **Key Performance Indicators (KPI's)**. Een Critical Succes Factor is iets wat moet gebeuren

om van een dienst of proces, plan, ... een succes te maken, bijvoorbeeld IT diensten moeten beschermd worden tijdens wijzigingen. Key Performance Indicators worden gebruikt om te meten of CSF's behaald worden. KPI's zijn dan ook meetbare dingen die het mogelijk moeten maken om een trend weer te geven of een beweging te zien. Ze kunnen zowel kwalitatief als kwantitatief zijn. Mogelijke KPIs voor de bovenstaande CSF kunnen bijvoorbeeld zijn het aantal mislukte changes, het percentage changes dat een incident veroorzaakt, ... KPI's zijn altijd verbonden met een CSF, ze kunnen niet op zichzelf staan. Twee of drie KPI's zijn meestal voldoende per CSF, meer wordt snel verwarrend.

Binnen CSI wordt nog een onderscheid gemaakt tussen de soorten metingen:

- Technische metingen zijn metingen die te maken hebben met de prestaties van componenten of applicaties. Ze meten dingen als beschikbaarheid, capaciteit, ...
- Proces metingen zijn de metingen die iets zeggen over hoe succesvol een proces is: Het bovenstaande voorbeeld is een proces meting.
- Dienst metingen (End to End) meten de volledige dienst zoals de eindgebruiker die ervaart. Technische en Procesmetingen worden gecombineerd om iets zinnigs te zeggen over de prestaties van een volledige dienst. Om bijvoorbeeld de uptime van de e-maildienst te meten, meet je de uptime van alle componenten, plus de processen die er mee verbonden zijn. Hoeveel incidenten zijn er geweest in verband met e-mail, waren er problemen met de capaciteit.

CSI is nauw verbonden met alle onderdelen van de ITIL levenscyclus, en het zorgt ervoor dat communicatie en feedback tussen de verschillende onderdelen zo makkelijk mogelijk verloopt. De mensen die zich bezighouden zijn meestal ook de meest ervaren werknemers, die op de hoogte zijn van alle verschillende processen.

De CSI Manager heeft de volgende taken:

- Verantwoordelijk voor het succes van alle verbeteringspogingen
- Communiceert de visie van CSI doorheen het hele bedrijf
- Heeft de verantwoordelijkheid voor de productie en het bijhouden van het CSI register
- Bepaalt en rapporteert de CSI CSF's en KPI's, en voert de metingen uit.
- Coördineert CSI doorheen de hele levenscyclus
- Bouwt effectieve relaties met zowel het gewone management als het IT management
- Zorgt ervoor dat monitoring uitgevoerd wordt om data te verzamelen

- Werkt samen met Proces Owners en Service Owners omverbeteringen te identificeren en kwaliteit te verbeteren

7.5. De resultaten van CSI

CSI levert een lijst op van mogelijke verbeteringen, die worden doorgegeven aan Service Strategy, die dan de beslissing nemen over welke verbeteringen effectief uitgevoerd zullen worden.