

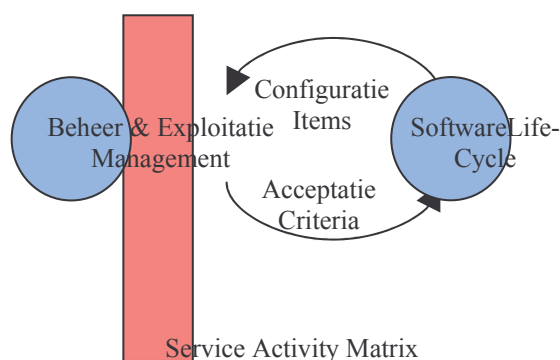
# ITIL en de SoftwareLifeCycle

Door ing. B. de Best

Het is momenteel al weer 10 jaar geleden dat het Britse Central Computer Telecommunications Agency (CCTA) het ITIL referentiemodel publiceerde. De meeste top-100 bedrijven in Nederland hebben momenteel wel enkele ITIL beheerprocessen ingevuld zoals de Helpdesk, Problem Management en Change Management. Verder doen veel bedrijven verwoede pogingen om meetbare dienstverlening te leveren door het Service Level Management proces in te vullen. De meerderheid van de bedrijven die dit proces invulling geven blijven echter steken bij het afsluiten van kwalitatieve Service Level Agreements (SLA's). De bedrijven die kwantitatieve SLA's afsluiten halen meestal niet de gestelde norm.

Een van de redenen dat service levels niet gehaald worden, is dat bedrijven beheerprocessen inrichten zonder te kijken naar de kwaliteit van de exploitatieprocessen en in nog mindere mate die van de SoftwareLifeCycleprocessen. Als je de beheerprocessen (ITIL Service Support Set en Service Delivery Set) goed wilt inrichten zul je eerst goed moeten kijken naar de inrichting van de exploitatieprocessen (ITIL Computer Operations Set, Network Service Management en dergelijke), alsmede de SoftwareLifeCycleprocessen (ITIL Software Support Set). Helaas blijft de scope van de meeste ITIL kenners beperkt tot de 10 ITIL boekjes van het operationeel en tactisch beheer, terwijl er 80 ITIL boekjes zijn.

De relatie tussen beheer en exploitatie enerzijds en de softwarelifecycle processen anderzijds is van groot belang om de afgesproken IT dienstverlening waar te kunnen maken. In dit artikel wordt deze relatie aan de hand van Figuur 1 uitgediept door, eerst een aantal begrippen op een rijtje te zetten, daarna het belang van de relatie aan te geven en tenslotte stil te staan bij de methodiek die ITIL aanbiedt om de relatie in kaart te brengen.



**Figuur 1, Beheer & Exploitatie versus de Softwarelifecycle**

In een vervolgartikel zal aansluitend worden ingegaan op een concretisering van de relatie, in termen van acceptatiecriteria zoals deze in het extended ISO-9126 model zijn gedefinieerd.

## Begrippen

Omdat de begrippen beheer, exploitatie en softwarelifecycle in dit artikel zo belangrijk zijn worden deze eerst kort toegelicht.

### Beheer:

Het uitoefenen van toezicht op en het besturen van de IT Infrastructuur. Hier zijn de volgende ITIL Sets bij betrokken: Managers Set (strategisch beheer), Service Delivery Set (tactisch beheer) en de Service Support Set (operationeel beheer)

### Exploitatie:

Het tegen gerechtvaardigde kosten ter beschikking stellen van de IT-Infrastructuur aan de gebruikers teneinde het met de opdrachtgever overeengekomen of verwachte dienstniveau te realiseren. Hierbij is met name de Computer Operations Set betrokken en beschrijft het Network Services Management de netwerkaspecten.

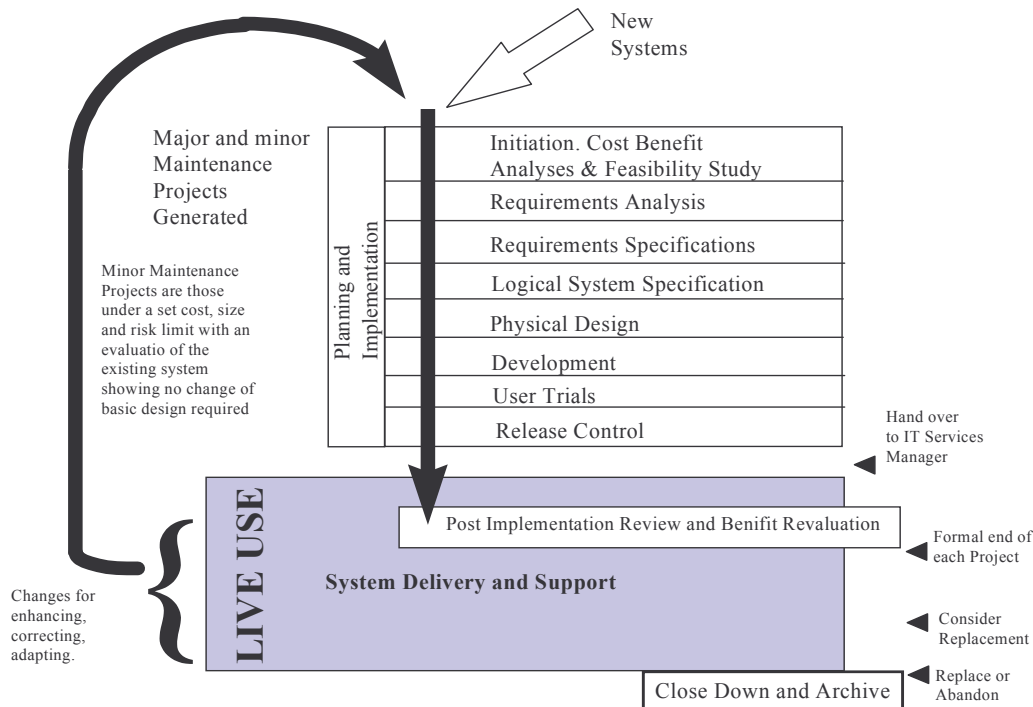
### Softwarelifecycle:

Het CCTA heeft naast het referentiemodel 'ITIL' ook 'ISEL' gepubliceerd. ISEL staat voor Information System Engineering Library. Het boekje 'Management of Software Maintenance' [ISEL 93] is een boekje uit deze library en is het lezen zeker waard. In [ISEL 93] wordt de softwarelifecycle opgedeeld in systeemontwikkeling en applicatiebeheer. Deze opdeling is heel gebruikelijk.

ISEL geeft de volgende definitie van applicatiebeheer:

*Software maintenance covers all changes to the software of an application system, including enhancements, after the original implementation and sign-off. Only the total redevelopment of the software is excluded. [ISEL 1993]*

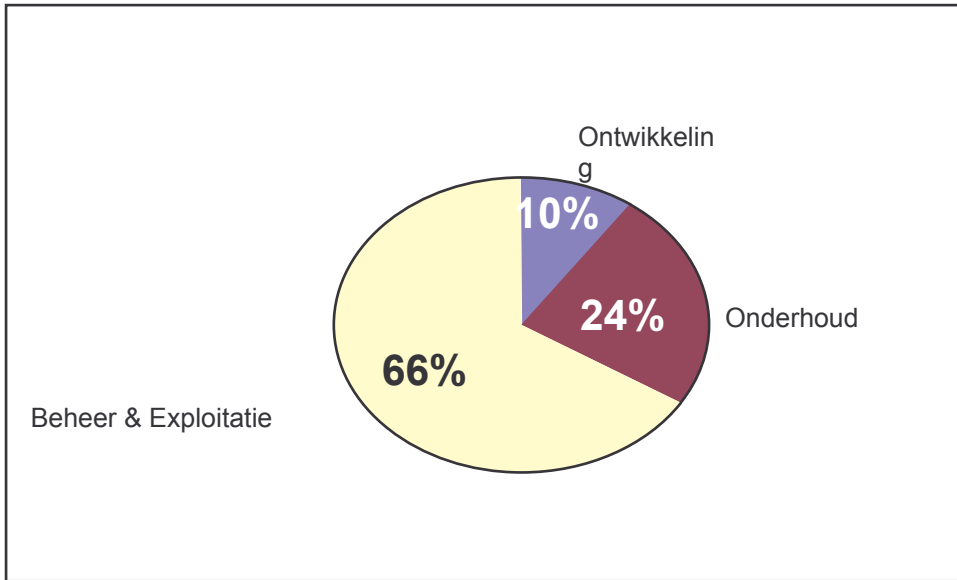
ISEL stelt dat systeemontwikkeling en applicatiebeheer dezelfde procesfasen doorlopen. In Figuur 2 is dit overzichtelijk weergegeven, in deze figuur is te zien dat systeemontwikkeling en applicatiebeheer bij dezelfde fase 'Initiation' starten. Het verschil is dus met name het tijdsaspect. Dit is een belangrijke stelling omdat hierdoor bij de beschrijving van de relatie tussen beheer & exploitatie en de softwarelifecycle geen onderscheid hoeft te worden gemaakt tussen systeemontwikkeling en applicatiebeheer.



Figuur 2, Softwarelifecycle volgens [ISEL 1993].

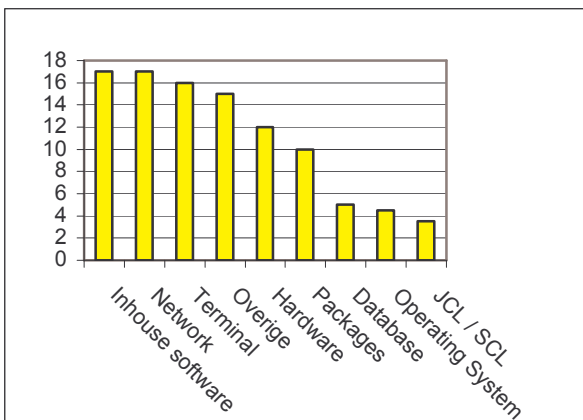
### Het belang van de relatie

Omdat applicaties een onderdeel zijn van de IT Infrastructuur, hebben de beheerders en exploitanten er erg veel belang bij dat de applicaties foutvrij zijn (stabiliteit en kosten) en onderhoudbaar zijn (flexibiliteit en kosten). Recente onderzoeken tonen echter aan dat de applicatiebeheerkosten veel hoger zijn dan de ontwikkelingskosten (zie Figuur 3) en de stabiliteit van applicaties te wensen overlaat (zie Figuur 4 en Figuur 5).

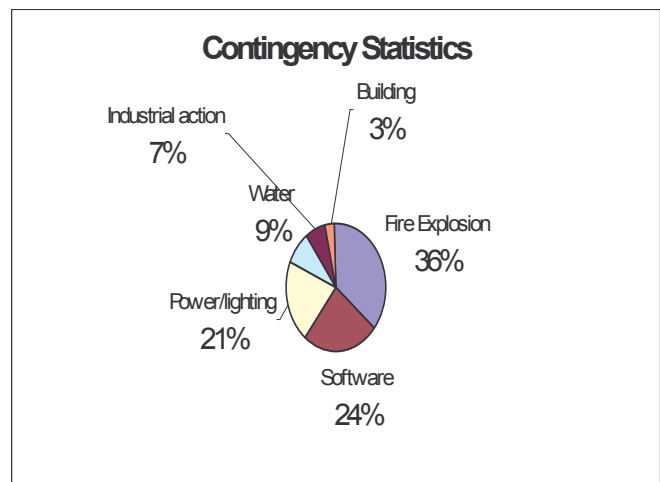


**Figuur 3, De relatie van kosten tussen software ontwikkeling, onderhoud en beheer en exploitatie, bron KPMG**

In Figuur 4 zijn in percentages de oorzaken van incidenten weergegeven (op basis van 56 helpdesks). Het blijkt dat zelf ontwikkelde software 17% van de incidenten veroorzaakt.



**Figuur 4, Bron XEPHON [ITIL 1995]**



**Figuur 5, Bron BIS Applied Systems [ITIL 1998]**

Naast incidenten komen ook nog situaties voor die leiden tot de noodzaak tot uitwijk. In figuur 5 is het resultaat weergegeven van een onderzoek naar de oorzaak van calamiteiten, in 24% van de 172 gevallen blijkt software de oorzaak van een calamiteit te zijn.

Wereldwijde onderzoeken wijzen uit dat de gemiddelde programmeur op de 10 programmaregels 1 fout maakt. Dit is heel ongerust makend, zeker als je bedenkt dat volgens de Engelsman Les Hatton de hoeveelheid software elke 18 maanden verdubbeld. Als je daarbij in overweging neemt dat het aantal functies dat in de software zit almaar toeneemt alsmede het aantal softwarelagen, dan wordt het serieus kijken naar exploitatiebaar en beheerbaar maken en houden van software steeds belangrijker.

De kwaliteit van de softwarelifecycleprocessen is dus mede maatgevend voor de kwaliteit van de te leveren IT services. Het bekijken van de relatie tussen de softwarelifecycleprocessen en de beheer- en exploitatieprocessen is dus de moeite waard. Het belang van deze relatie wordt in deze paragraaf vanuit twee gezichtspunten beschouwd, vanuit de softwarelifecycle enerzijds en vanuit de beheer en exploitatie anderzijds.

### **Softwarelifecycle gezichtspunt**

Vaak wordt gesteld dat de softwarelifecycleprocessen geen belang hebben bij beheer en exploitatie, deze processen moeten gewoon de producten binnen de grenzen van tijd en geld opleveren, daarna komt weer een nieuwe uitdaging. Dit is echter niet correct, zeker niet voor applicatiebeheerprocessen. Als er bijvoorbeeld Operational Level Agreements zijn afgesloten, zijn deze bepalend voor de uitvoering van de applicatiebeheerprocessen en is de relatie met beheer en exploitatie dus expliciet aanwezig. Het voordeel van Operational Level Agreements is dat de werkelijke kosten van de applicatiebeheer inzichtelijk worden. Ook de werkdruk wordt meetbaar waardoor het mogelijk wordt de noodzaak van extra functieplaatsen hard te maken.

Tevens is het hebben van een stabiele IT Infrastructuur van belang om ongestoord een applicatie te kunnen ontwikkelen en opleveren. Een middelgroot project heeft al veel schade bij het uitvallen van IT services. Steeds vaker worden beheer en exploitatie vanuit de lijnorganisatie betrokken bij het ontwikkelen van software. Bij megaprojecten komt het zelfs voor dat binnen de projectorganisatie geheel autonome beheer- en exploitatie omgevingen worden opgezet. Hierbij worden dan configuratie-items door ontwikkelteams opgeleverd en door acceptatieteams getest. Bij fouten of aanvullingen wordt gewoon het ITIL incident management toegepast inclusief problem management, change management en configuration management, zonder dat het systeem ook maar 1 seconde operationeel is geweest. Bij het in productie nemen van een op deze manier gebouwd systeem kan al vanaf het eerste moment geprofiteerd worden van een hele historie aan beheerinformatie (CI's, incidenten, problemen en known errors).

Tot slot kunnen de medewerkers die betrokken zijn bij de softwarelifecycleprocessen gebruik maken van de kennis en ervaring waarover de beheerders en exploitanten in de lijnorganisatie beschikken. Op basis van deze kennis en ervaring kan vaak al invulling worden gegeven aan acceptatiecriteria.

### **Beheer & Exploitatie gezichtspunt**

Vanuit het referentiemodel van ITIL bekeken leveren zowel systeemontwikkeling als applicatiebeheer, op verzoek van Change Management, nieuwe dan wel gewijzigde configuratie items (CI's) op. Deze configuratie items worden opgenomen in de IT Infrastructuur. Omdat de SLA's gebaseerd zijn op deze IT Infrastructuur is de kwaliteit van deze configuratie items dus mede bepalend of de afgesproken service levels worden gehaald. Het belang is dus de kwaliteit van de CI's.

Nu het belang van de relatie tussen beheer & exploitatie enerzijds en de softwarelifecycleprocessen anderzijds duidelijk is, wordt het tijd om te kijken naar de relatie zelf.

## De relatie in kaart gebracht

De relatie van de softwarelifecycle en beheer & exploitatie staat beschreven in het ITIL boek 'Software Lifecycle Support' [ITIL 1993]. Dit boek beschrijft diverse lifecyclemodellen zoals het watervalmodel, het spiraalmodel, en het evolutionaire model. Het boek beschrijft voorts de selectie en toepassing van lifecyclemodellen. Het meest interessante aan dit boek is dat ook de relatie wordt gelegd tussen de softwarelifecycle modellen en de ITIL beheer- en exploitatieprocessen. ITIL hanteert hiervoor een zogenaamde Service Activity Matrix (SAM), een voorbeeld is opgenomen in Tabel 1.

| Lifecycle Stage >               | SF | SR | PD | DD | CO | UT | IN | PV | IM | ST | OM |
|---------------------------------|----|----|----|----|----|----|----|----|----|----|----|
| Service Function                |    |    |    |    |    |    |    |    |    |    |    |
| <b>Service Management Set</b>   |    |    |    |    |    |    |    |    |    |    |    |
| Planning & Control              | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Quality Audit                   | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Managing FM                     | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Customer Liaison                | X  | X  | -  | -  | -  | -  | -  | X  | X  | X  | X  |
| <b>Service Delivery Set</b>     |    |    |    |    |    |    |    |    |    |    |    |
| Service Level Management        | X  | X  | -  | -  | -  | -  | -  | X  | X  | X  | X  |
| Capacity Management             | X  | X  | X  | -  | -  | -  | -  | X  | X  | X  | X  |
| Contingency Planning            | X  | -  | -  | -  | -  | -  | -  | X  | X  | X  | X  |
| Availability Management         | X  | -  | -  | -  | -  | -  | -  | -  | X  | X  | X  |
| Cost Management                 | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| <b>Service Support Set</b>      |    |    |    |    |    |    |    |    |    |    |    |
| Configuration Management        | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Problem Management              | X  | -  | -  | -  | -  | -  | -  | -  | X  | X  | X  |
| Change Management               | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Help Desk                       | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Software Control & Distribution | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  |
|                                 |    |    |    |    |    |    |    |    |    |    |    |
| <b>Computer Operation Set</b>   | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  |

Tabel 1, Voorbeeld van een Service Activity Matrix (SAM), [ITIL 1993]

|    |                       |    |                              |
|----|-----------------------|----|------------------------------|
| SF | = System Feasibility  | IN | = Integration                |
| SR | = System Requirements | PV | = Product Verification       |
| PD | = Product Design      | IM | = Implementation             |
| DD | = Detailed Design     | ST | = System Testing             |
| CO | = Code                | OM | = Operations and Maintenance |
| UT | = Unit Test           |    |                              |

### Het opstellen van een SAM

De SAM is van toepassing bij bedrijven die zelf software ontwikkelen en tevens de opgeleverde configuratie items willen exploiteren en beheren. In [ITIL 1993] wordt gesteld dat bij de keuze van een softwarelifecycle (uiteraard voorafgaand aan een project) tevens moet worden gekeken in welke fase van de gekozen lifecycle de beheer- en exploitatieprocessen betrokken moeten worden. Hiervoor is overleg nodig tussen systeemontwikkeling, beheer en exploitatie.

De gekozen lifecycle fasen vormen dan de horizontale as van de SAM en de ingerichte beheer- en processen de verticale as van de SAM. De kruisjes worden dan alleen gezet in die hokjes waarbij sprake is van interactie tussen de softwarelifecycle en het betrokken beheer- of exploitatieproces. Voor alle duidelijkheid, het is dus mogelijk dat binnen één organisatie meer SAM's van toepassing zijn, omdat er bijvoorbeeld enerzijds met RAD is ontwikkeld en anderzijds een SDM (waterval) lifecycle is toegepast.

### Het gebruik van een SAM

Een kruisje in de SAM geeft dus de betrokkenheid aan van een beheer- of exploitatieproces bij een bepaalde fase in de lifecycle. In de bijlagen F1 t/m F18 van [ITIL 1993] wordt per ITIL proces in het kort de relatie aangegeven. In

bijlage F5 bijvoorbeeld wordt gesteld dat Service Level Management al bij de fase 'feasibility studies' (vooronderzoek) betrokken kan worden voor initieel advies met betrekking tot het gewenste service niveau van het voorgestelde systeem.

De SAM wordt met name ingevuld door te kijken naar de betrokkenheid van ITIL processen bij de lifecycle fasen. Aan het einde van het boekje [ITIL 1993], in bijlage G, wordt heel summier ingegaan op software metrics als gereedschap om de kwaliteit van de software meetbaar te maken en te verbeteren. Helaas is de relatie tussen de softwarelifecycle en de beheer- en exploitatieprocessen binnen ITIL dus beperkt tot de betrokkenheid van de ITIL processen per levenscyclusfase.

### **Acceptatiecriteria**

De in de vorige alinea opgemerkte beperking is jammerlijk, omdat naast betrokkenheid van een ITIL proces bij een fase van de software lifecycle tevens acceptatiecriteria kunnen worden gesteld die de uitvoering van het betrokken ITIL proces vereenvoudigen. Immers vanuit beheer- en exploitatie kunnen heel goed acceptatiecriteria worden gesteld aan de opgeleverde software configuratie items. Sterker nog, de SAM is een heel goed gereedschap om naast de betrokkenheid van de beheer- en exploitatieprocessen per levenscyclusfase ook de raakvlakken aan te geven waar sprake is van acceptatiecriteria.

Je kunt bijvoorbeeld stellen dat het helpdeskproces als acceptatiecriterium aan de technisch ontwerpfase stelt, dat naast het interactief tonen van een foutboodschap er ook een foutrecord in een foutenlog moet worden weggeschreven (bijvoorbeeld de applicationlog van de event viewer van Windows NT). Het extended ISO 9126 model geeft een indeling van acceptatiecriteria. Zoals eerder gesteld wordt in een vervolg artikel hier nader op ingegaan.

## Theorie en praktijk

Als een organisatie op de in dit artikel voorgestelde werkwijze te werk gaat, is dit nog steeds geen garantie voor het kunnen afsluiten van kwantitatieve SLA's. Waar kan het dan nog meer fout gaan? Het blijkt dat als alle acceptatiecriteria zijn opgesteld en gehanteerd worden als meetlat, dat er onderkend wordt dat applicaties te slecht zijn om in productie te worden genomen. Door de druk van de business echter wordt er gesteld dat het commercieel of politiek belang de voorrang krijgt en de applicatie in productie moet worden genomen. In plaats van een stoplichtfunctie krijgt de afdeling acceptatie steeds vaker een risicoanalyse functie: 'Wat kost het als iets fout gaat?', want in productie gaan we toch!

Het is cynisch genoeg dus vaak juist de business die deze situatie in de kaart speelt, door het aspect 'time to market' te prefereren boven de kwaliteit van de IT Infrastructuur. Liever snel een slecht softwarepakket om een nieuwe business te kunnen doen, dan een goed softwarepakket op het moment dat de markt al verdeeld is onder de concurrentie.

Naar mate organisaties meer marktgericht gaan opereren en de levenscyclus van maatwerksoftware nog korter wordt zal ook het in balans houden van de stabiliteit en flexibiliteit van de IT Infrastructuur een hachelijke zaak worden. Change managers, ik wens jullie veel sterkte toe.

## DO's en DONT's

Met een aantal DO's en DONT's wordt aangegeven waar quick wins te behalen zijn.

### Do's

- ◆ Gebruik de voorgestelde SAM technieken om bestaande acceptatiecriteria te completeren.
- ◆ Betrek zoveel mogelijk partijen bij het opzetten van een SAM
  - ◆ Bepaal welke beheer- en exploitatieprocessen ingericht zijn
  - ◆ Bepaal welke SoftwareLifeCycle methoden gehanteerd worden
  - ◆ Betrek beheer, exploitatie, software ontwikkeling en onderhoud bij het opstellen van een SAM, dit voor alle partijen erg leerzaam. Wellicht moet een awareness campagne hier aan vooraf gaan.
- ◆ Gebruik een controlemiddel om na te gaan of elke onderkende relatie in de SAM invulling heeft gekregen (b.v. laat door betrokken partijen de SAM paragraferen).
- ◆ De Service Level Manager moet niet beginnen met een SLA template te maken, maar zich in eerste instantie druk maken om de external specsheets op te stellen (business requirements). Door het opstellen van de external specsheets te combineren met het ontwerpen van functionele specificaties ontstaat een completer beeld van wat de klant wil.
- ◆ De Service Level Manager moet in een zo vroeg stadium van de Softwarelifecycle van een project worden betrokken.
- ◆ De external specsheets moeten vertaald worden naar interne specsheets (IT requirements / service metrics), betrek hierbij niet alleen de exploitatieprocessen maar ook de SoftwareLifeCycle processen.
- ◆ Bij de bouw van softwareproducten kunnen beheer- en exploitatiefuncties worden ingebouwd zoals het registreren van het gebruik van een module (opsporen van dood hout).
- ◆ Probeer de kosten van de beheer- en exploitatieomgeving te drukken door te investeren in het bouwen van robuuste en onderhoudbare applicaties.

### Dont's

- ◆ Gebruik de SAM niet als wetboek

## Literatuurlijst

- [ISEL 1993] West, R.; O'Neill, D. "MANAGEMENT OF SOFTWARE MAINTENANCE", HMSO, United Kingdom. 1993, ISBN 0-11-330584-2
- [ITIL 1993] CCTA, "SOFTWARE LIFECYCLE SUPPORT", 2e dr., Norwich, HMSO Publicity, 1989, ISBN 0 11 330559 1
- [ITIL 1995] CCTA, "HELPDESK", 5e dr., Norwich, HMSO Publicity, 1995, ISBN 0 11 330522 2
- [ITIL 1998] CCTA, "CONTIGENCY PLANNING", 9e dr. dr., Norwich , HMSO Publicity, 1998, ISBN 0 11 330524 9