

## TITEL

# Overheidsorganisatie verkleint risico's binnen het Software Lifecycle Management-proces.

Door Marco Vos

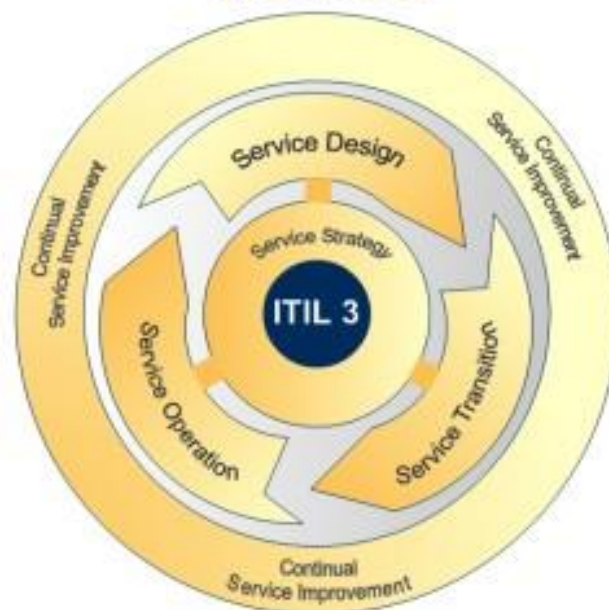
## INTRODUCTIE

Voor de serviceverlening van de overheid wordt al vele decennia software ontwikkeld. Die ontwikkeling verandert snel qua software-architectuur en qua software-ontwikkelmethoden. Wat niet verandert is de behoefte om de business- en beheerrisico's in het software lifecycle managementproces te beheersen. Dit artikel beschrijft de resultaten van een onderzoek bij de overheid naar de mate waarin software lifecycle management in control is. De doelgroep van het onderzoek is het management van de onderzochte overheidsorganisatie. Ook andere (overheids)instanties kunnen hun voordeel doen met de uitkomsten van het onderzoek. Het onderzoek betrof een Agile-softwareontwikkelafdeling.

## DOELSTELLING

Bij veel overheidsorganisaties wordt gekeken hoe de time-to-market kan worden versneld en de kosten van softwareontwikkeling kunnen worden gereduceerd. Hierbij moet steeds een balans worden gevonden tussen de behoefte aan snelheid en efficiëntie enerzijds en de risicobeheersing (het "in control zijn") anderzijds. Het onderzoek beschrijft de problemen die zijn onderkend en geeft aan hoe een overheidsorganisatie deze balans heeft gevonden en geborgd. Als kader is in dit onderzoek gebruik gemaakt van de fasen van de servicelevenscyclus zoals deze in ITIL 2011 zijn gedefinieerd (zie figuur 1). De service lifecycle fasering loopt gelijk aan die van de software. Vandaar dat dit artikel de geïnventariseerde SLCM-knelpunten bespreekt vanuit deze ITIL 2011 fasering. De motivatie voor het uitgevoerde onderzoek is het optimaliseren van softwareontwikkeling binnen de Nederlandse overheid.

## ITIL 2011



Figuur 1: ITIL-levenscyclusfasen

Bron: [http://wiki.en.it-processmaps.com/index.php/Main\\_Page](http://wiki.en.it-processmaps.com/index.php/Main_Page)

## HET ONDERZOEK

De betrokken organisatie ontwikkelt al vele jaren zelf software. Hierbij zijn vooral de fasen service-design, service-transition en service-operation benadrukt. Het gebrek aan control is veroorzaakt door de komst van nieuwe softwarepakketten, het bijwerken van bestaande softwarepakketten en het uitfaseren van verouderde software. De effecten van onvoldoende control zijn terug te vinden in het feit dat architectuurplaten niet actueel zijn, requirements niet bekend of getoetst zijn en dat risico's niet bekend zijn. Daarnaast is gebleken uit field- en deskresearch dat software niet altijd wordt ontwikkeld op basis van eisen en wensen van de eindgebruikers. De behoefte aan een betere control is ontstaan vanuit de werkvloer. De medewerkers op de werkvloer hebben het management verzocht om een onderzoek te starten naar het verbeteren en optimaliseren van het softwareontwikkelproces binnen de onderzochte organisatie. Door het optimaliseren van het softwareontwikkelproces wordt de softwareontwikkeling efficiënter qua tijd en geld. Dit is een voordeel waar ook ketenpartners profijt van kunnen hebben.

## DE ONDERKENDE PROBLEMEN

Het onderzoek heeft op de locatie van de organisatie plaats gevonden door middel van interviews met de volgende stakeholders: ontwikkelaars, systeembeheerders en het management. De belangrijkste knelpunten die zijn geïdentificeerd in het software lifecycle managementproces zijn in tabel 1 opgenomen.

Fase	Knelpunt	Uitleg knelpunt
<b>Service Strategy (SS)</b>	P1	De architectuurplaten zijn niet actueel (gebrek aan control op architect niveau).
<b>Service Design (SD)</b>	P2,	De non-functional requirements per service en per component zijn niet bekend (gebrek aan control op het niveau van service level management).
<b>Service Transition (ST)</b>	P3	De requirements zijn niet getoetst op een inhoudelijke risicoanalyse.
	P4	De impact en de risico's van de te releasen software zijn niet bekend en niet vertaald naar acceptatiecriteria.
	P5	Ontwikkelaars testen anders dan beheerders (gebrek aan control op het niveau van softwareontwikkeling).
	P6	Requirement Traceability: het is niet altijd aantoonbaar dat een eis van de klant ook daadwerkelijk is geïmplementeerd (gebrek aan control op het niveau van softwareontwikkeling).
<b>Service Operation (SO)</b>	N.V.T.	N.V.T.
<b>Continual Service Improvement (CSI)</b>	N.V.T.	N.V.T.

TABEL 1: KNELPUNTEN AFGEBEELD OP SOFTWARE LIFECYCLE FASEN

Door de knelpunten af te beelden op de software lifecycle fasen ontstaat een duidelijk beeld van de samenhang van de knelpunten. De knelpunten versterken elkaar fase voor fase in de software lifecycle. Om dit tij te keren is gekeken naar een aanpak die op een slimme manier tollgates definieert voor elke fase. De tollgates moeten de toetsingen bevatten om te komen tot een succesvolle volgende stap. Verder moet in de aanpak een risicobeheersing concreet zijn gemaakt.

In het onderzoek is hiertoe gebruik gemaakt van het GSA-stappenplan zoals in figuur 2 weergegeven. GSA staat voor Generieke en Specifieke Acceptatiecriteria. De generieke acceptatiecriteria zijn van toepassing op alle informatiesystemen en de specifieke acceptatiecriteria zijn alleen voor één informatiesysteem van toepassing. In het kort komt deze aanpak neer op het risico-gebaseerd accepteren van nieuwe of aangepaste informatiesystemen. Zie voor meer details intermezzo I.

Intermezzo I

Figuur 2: GSA-stappenplan.  
Bron: Acceptatiecriteria, de Best 2014

- Stap 1 (Service Strategy) omvat de afbakening van het informatiesysteem op basis van een landschapsplaat van businessprocessen, applicatie en infrastructuur.
- Stap 2 (Service Design) omvat de scopebepaling op basis van de architectuurbouwstenen.
- Stap 3 (Service Design) omvat het inventariseren van concrete risico's op basis van de applicatiebouwstenenplaten.
- Stap 4 (Service Transition) omvat het bepalen van toetsingen van de tegenmaatregelen van de risico's.
- Stappen 5, 6 en 7 (Service Transition) omvatten het testen van de acceptatiecriteria.

In dit onderzoek is deze aanpak gebruikt door per softwarelifecycle-fase de problemen te inventariseren en op basis daarvan de risico's en tegenmaatregel(en) te definiëren. De toetsingen van deze tegenmaatregelen worden gedefinieerd in de vorm van specifieke en generieke acceptatiecriteria.

	Stap 1	Stap 2	Stap 3	Stap 4	Stap 5	Stap 6	Stap 7
	Beeld	Scope	Risico	GSA	Focus	Plan	Test
<b>SS</b>	X						
<b>SD</b>		X					
<b>ST</b>			X	X	X	X	X
<b>SO</b>							
<b>CSI</b>							

Tabel 2: Stappen in de softwarelevenscyclus-fasen.

## ONDERZOEKSRISULTATEN

In tabel 3 zijn de problemen afgebeeld op de risicomatrix zoals in tabel 2 is afgebeeld.

	Stap 1	Stap 2	Stap 3	Stap 4	Stap 5	Stap 6	Stap 7
	Beeld	Scope	Risico	GSA	Focus	Plan	Test
SS	P1						
SD		P2					
ST			P3	P4	P5	P5	P5, P6
SO							
CSI							

Tabel 3: Problemen in de softwarelevenscyclus fasen.

De gevonden knelpunten zijn geëvalueerd met de betrokken specialisten. Hieruit is gebleken dat de knelpunten niet uniek zijn voor het onderzochte informatiesysteem. In wezen zijn de knelpunten generiek en vereisen zij een generieke borging. Daarom zijn deze knelpunten vertaald naar de volgende risico's, tegenmaatregelen en acceptatiecriteria:

	Knelpunten	Risico's	Tegenmaatregelen	Acceptatiecriteria
P1	<b>De architectuurplaten zijn niet actueel.</b>	Impact van changes is niet helder.	Het bijwerken van architectuurplaten bij changes.	Bijwerken architectuurplaten voordat de risico-analyse wordt verricht.
P2	<b>De non-functional requirements per service en per component zijn niet bekend.</b>	Effect van changes is niet bekend. SLA-normen worden niet gehaald	Lijst met non-requirements opstellen op basis van een risicoanalyse.	Bijgewerkte lijst met non-functionele requirements per service/component. Bijgewerkte SLA.
P3	<b>De requirements zijn niet getoetst op een inhoudelijke risicoanalyse.</b>	Onduidelijk effect van aanpassing requirements.	Risicoanalyse verrichten en waar nodig nieuwe requirements toevoegen.	Getekende lijst van maatregelen en risico's die niet beheerst worden.
P4	<b>De impact en de risico's van de te releasen software zijn niet bekend en niet vertaald naar acceptatiecriteria.</b>	Versiebeheer problemen en integratie-problemen	Opstellen lijst versie / integratie risico's met impact op basis van de architectuurplaten en bouwstenen. Integratie testen van nieuwe releases in OTA-omgeving. Tijdig bijwerken CMDB.	Door architect afgetekende lijst van versie-eisen en integratie-eisen.
P5	<b>Ontwikkelaars testen anders dan beheerders.</b>	De dekking- graad van de testen is niet compleet. De software is niet beheerbaar.	Gemeenschappelijk master testplan voor testers en beheerders. Per bouwsteen is aangegeven welke risico's zijn onderkend en welke testsoorten door welke partij worden verricht.	Mastertestplan is leidend voor ontwikkelaars. 80% van de testcases moet gebaseerd zijn op risicovolle bouwstenen.
P6	<b>Requirement Traceability: het is niet aantoonbaar dat een eis van de klant ook daadwerkelijk is geïmplementeerd.</b>	Bij oplevering bestaat de kans dat wat is uitgerold niet is wat de klant wenst.	Afstemming met de klant zoeken, lijst met wensen/eisen opstellen, alvorens te starten met ontwikkelen.	Door de klant ingevulde en voor akkoord ondertekende checklist met requirements.

Tabel 4: Knelpunten, risico's, tegenmaatregelen en acceptatiecriteria.

## BORGING

Voor alle informatiesystemen zal voortaan acceptatie plaatsvinden op basis van de onderkende risico's in de software lifecycle. Hiertoe zal het GSA-stappenplan worden gebruikt.

## RESULTAAT

Binnen de organisatie waar het onderzoek is uitgevoerd, heeft het management positief gereageerd op de uitkomsten van het onderzoek. Vanuit het management zal verder worden onderzocht of de aanbevelingen deels of geheel kunnen worden geïmplementeerd. Hiervoor zal een tijdspad worden gedefinieerd, mogelijk in de vorm van een implementatieproject.

## CONCLUSIE

De conclusie vanuit het uitgevoerde onderzoek is dat organisaties, zowel overheidsinstanties als commerciële ondernemingen, voordeel kunnen halen uit het optimaliseren van hun softwareontwikkelprocessen. Het effectiever en efficiënter analyseren van de risico's en het definiëren van de tegenmaatregelen op basis van een architectuurdecompositie is een verrijking voor SLCM-proces. Het resulteert in tijdswinst, kostenbesparing en een verkorte "Time To Market".

## DANKBETUIGING

Hierbij dank ik mijn collega's voor hun medewerking bij de uitvoering van het onderzoek en het tot stand komen van dit artikel. Daarnaast dank ik Bart de Best voor de lesmomenten die hij heeft verzorgd en voor zijn begeleiding bij het schrijven van dit artikel.

## REFERENTIES

Referenties zijn op aanvraag beschikbaar via [marcovos@hotmail.com](mailto:marcovos@hotmail.com). Feedback op dit artikel mag naar dit e-mailadres worden gestuurd.

## LITERATUUR

Tijdens het onderzoek is gebruik gemaakt van het boek "Acceptatiecriteria" (ISBN-nummer 9789012581332) van Bart de Best. Bart de Best is trainer bij IT Management Group voor onder andere de trainingen "Masterclass IT Risicomanagement" en "Masterclass Agile Service Management met Scrum".