

# SharePointCustomer

## SharePoint Ontwikkeling



# 1 CONTENTS

---

2	Farm vs SandBox vs Apps .....	5
3	tools, strategie en mogelijkheden voor het monitoren.....	7
3.1	Overview van de monitoring tools.....	7
3.2	Het identificeren van verschillende monitoring scenario's .....	10
3.3	Bepaal de dagelijkse, wekelijkse en maandelijkse te monitoren items.....	10
3.4	Advies voor het monitoren binnen de SharePointCustomer Farm omgeving in Azure .....	10
3.5	Bepaal de mensen die gaan monitoren .....	12
3.6	Het creëren van een Respons Plan .....	12
3.7	Related Resources.....	13
4	Apps voor SharePoint en Office .....	14
4.1	Type Apps.....	14
4.2	Voorbereiden omgeving voor de Apps .....	14
4.2.1	Related Resources.....	15
4.3	Deployment.....	16
4.3.1	Publiceren naar een app catalogus.....	16
4.4	Management.....	16
4.4.1	Governance: bepaal de App for SharePoint policy voor de organisatie .....	16
4.4.2	Deployment richtlijnen vanuit SharePointCustomer .....	19
4.4.2.1	App mogelijkheden, eisen en wensen .....	19
4.4.2.2	Controle op ontwikkeling en stabiliteit – Code controle .....	19
4.4.2.3	Security .....	20
4.4.2.4	Documentatie .....	21
4.5	Monitoring .....	21
4.6	Ontwikkelproces en methodiek.....	23
4.7	Testen.....	24
4.7.1	Related Resources.....	24
5	Sandbox solutions en CSOM/REST.....	25
5.1	Deployment.....	25
5.1.1	Deployment richtlijnen vanuit SharePointCustomer .....	25
5.1.1.1	App mogelijkheden, eisen en wensen .....	25
5.1.1.2	Controle op ontwikkeling en stabiliteit – Code controle .....	26

5.2	Management.....	28
5.3	Monitoring .....	28
5.4	Ontwikkelp proces en methodiek.....	28
5.5	Testen.....	28
6	farm solutions .....	29
6.1	Deployment.....	29
6.1.1	Deployment considerations.....	29
6.1.2	Zaken om extra aandacht aan te besteden .....	31
6.1.3	Code Patterns.....	31
6.1.4	Deployment richtlijnen vanuit SharePointCustomer .....	31
6.1.4.1	Farm Solution mogelijkheden .....	31
6.1.4.2	Controle op ontwikkeling en stabiliteit – Code controle .....	32
6.1.4.3	Security .....	33
6.1.4.4	Documentatie .....	33
6.1.4.5	Monitoring .....	33
6.2	Management.....	34
6.3	Monitoring .....	34
6.4	Ontwikkelp proces en methodiek.....	34
6.5	Testen.....	34
7	Het testproces voor de verschillende solutions.....	35
7.1.1	Stap 1: Een productrisicoanalyse .....	35
7.1.2	Stap 2: Een testplan .....	36
7.1.3	Stap 3: De uitvoering van een systeemtest .....	36
7.1.4	Stap 4: Acceptatietest .....	36
8	Bibliography .....	38
9	Appendix A.2.....	40
10	Appendix A.4.4.1.....	42
11	Appendix A.3.1 .....	44
12	Appendix A.3.2.....	45
13	Appendix A.3.4.....	46
14	Appendix A.3.5.....	47
15	Appendix A.4.1 .....	48
16	Appendix A.4.2.....	49

17	Appendix A.4.4.2.3.....	50
18	Appendix A.4.3.1.....	51
19	Appendix A.4.6.....	52
20	Appendix A.5.1.1.....	54

## 2 FARM VS SANDBOX VS APPS

---

Voor SharePoint 2010 introduceerde het SharePoint team een uitgewerkte architectuur -model voor het hosten van sandbox- oplossingen, die werd verstrekt als een aantrekkelijk alternatief voor Farm oplossingen aan te bieden. Farm -oplossingen worden gedeployed naar de GAC of web applicatie map bin, en hebben het potentieel om de SharePoint- farm destabiliseren. Ze vereisen van IT-professionals om een praktische kennis te hebben van Code Access Security ( CAS ) als u de functionaliteit wilt beperken en / of inzicht wilt hebben in de mogelijkheden van een Farm oplossing. In het echte leven dit bleek dit een uitdaging.

Sandbox- oplossingen hebben dat allemaal veranderd. De Sandbox is een apart proces waarin SharePoint- oplossingen ( zogenaamde sandbox -oplossingen ) geïsoleerd draaien in de User Code Service, die onder een zeer strikte CAS beleid draait maar, dan mag je nog steeds service calls doen op de client- side of full trust proxy calls aan de server kant. Om het nog beheersbaarder te maken, was er een Sanbox resource beperking mechanisme, waarin IT-professionals resource throttling instellingen konden opgeven om bij Sandbox oplossingen te voorkomen dat er meer server resources gebruikt zouden gaan worden. Plotseling waren sandbox oplossingen zo belangrijk dat bijna elke gezaghebbende bron het advies gaf dat duidelijk dicteerde dat je altijd Sandboxed moet ontwikkelen tenzij anders gedwongen. Dit was waarschijnlijk de meest aanbevolen ontwikkeling van beste praktijken voor SharePoint 2010 , en het was een logisch en goed advies. Of was het dat wel?

Er is een nieuw model ontwikkelt; het App model. SharePoint Apps kunnen worden gehost in een geïsoleerde SharePoint-site, of los van de SharePoint- farm, hetzij op een speciale self- hosted applicatie server of in de cloud ( Azure ). SharePoint Apps moeten dan gebruik maken van de uitgebreide en verbeterde client object model om te connecteren met de SharePoint- farm als ze daar wat werk willen doen ( SharePoint server - side code is niet toegestaan / mogelijk voor Apps ). De belangrijkste voordelen van SharePoint apps zijn tweeledig:

Een gescheiden app op zich heeft geen invloed op de prestaties van de SharePoint-farm op enige wijze, en hoeft niet vanuit de SharePoint- farm te worden beheerd. Dat gezegd hebbende, dienen we in gedachten te houden dat Apps gebruik maken van het SharePoint- client object model wat natuurlijk de SharePoint farm prestaties beïnvloed op een indirecte manier.

Als ontwikkelaar / software bedrijf, kun je Apps distribueren via de MS App Store die het vinden van je Apps sterk vergemakkelijkt en het mogelijk maakt met de Apps geld te verdienen.

De nieuwe ontwikkeling van best practices is om SharePoint- Apps te bouwen in situaties waar je eerder zou hebben gekozen om een Sandbox oplossing te bouwen. Opmerkelijk is het dat sommige eerdere voorstanders van sandbox- oplossingen 180 graden gedraait zijn en nu beweren dat de sandbox- oplossingen uiteraard nutteloos waren vanaf het begin, omdat ze niets mochten doen. Dat is onterecht. Een meer correcte manier om het te formuleren is dat tegenwoordig ook Microsoft vindt dat wat je ook deed met Sandbox- oplossingen, dit op een betere manier gedaan kan worden via SharePoint Apps.

Toen we voor het eerst leerden over SharePoint Apps, leken ze als een logische uitbreiding op de bestaande opties. Het feit dat de Sandbox -oplossingen nu lijken te zijn gedeprecieerd, is een verassing.

Wat we eerder zullen tegenkomen, ook binnen SharePointCustomer is een beleid van “Apps tenzij”, waarbij tenzij betekent dat er Farm solutions ontwikkelt worden als de Business Case daarom vraagt en datzelfde geldt voor SandBoxed solutions.

Wat in dit document geprobeerd wordt, is duidelijk te maken wat de verschillende opties zijn en er vindt een sterkere focus plaats op de Apps ontwikkeling. Van deze Apps ontwikkeling zullen de verschillende varianten nader bekeken worden en er zal een voorkeur uitgesproken worden voor een bepaalde variant. Van alle mogelijke oplossingen, Farm, SandBoxed en Apps zal er beschreven worden wat de wijze is waarop binnen SharePointCustomer met deze varianten gewerkt zal worden. Bij voorkeur zullen leveranciers zich daaraan commiteren. Wanneer dat niet het geval is, dan zal per leverancier bekeken worden wat de Business Case is van deze opstelling. A.d.h.v. deze analyse zal een stellingname van SharePointCustomer bepaalt worden.

Het is interessant om een vergelijking te doen. Onderstaande tabel is onderhevig aan veranderingen:

Zie Appendix A.2. : [Appendix A.2.](#)

### 3 TOOLS, STRATEGIE EN MOGELIJKHEDEN VOOR HET MONITOREN

---

Om ervoor te zorgen dat SharePoint 2013 goed blijft draaien, moeten IT-professionals nauwlettend farms, servers, applicaties, diensten en prestaties in de gaten houden. Je moet preventief onderhoud regelmatig doen om problemen te voorkomen, en een maken plan dat de schade zal minimaliseren wanneer een probleem zich voordoet.

Microsoft biedt een groot aantal tools die ons helpen om de algemene gezondheidstoestand van de SharePoint 2013 omgeving te controleren. Als er iets mis gaat, kunnen we een overvloed aan middelen vinden en de controle-instrumenten om logs te analyseren gebruiken om de oorzaak te identificeren, en vervolgens passende acties nemen om de problemen op te lossen.

We moeten ook plannen hoeveel mensen we de SharePoint 2013 omgeving willen laten monitoren, en een reactieplan beschrijven voor acties die we moet nemen wanneer een probleem optreedt.

#### 3.1 OVERVIEW VAN DE MONITORING TOOLS

- SharePoint Health Analyzer
  - Op de Central Admin startpagina, klik op Monitoring en kies Health Analyzer. We kunnen deze ingebouwde functie gebruiken voor het analyseren en oplossen van problemen op de volgende gebieden: veiligheid, prestaties, configuratie en beschikbaarheid. Health Analyzer regels zijn voorgedefinieerd en draaien op geplande tijdstippen, zoals uur, dagelijks, wekelijks en maandelijks. Als er een fout wordt gedetecteerd, wordt de bijbehorende regel geactiveerd. Elke regel heeft een korte uitleg over de reden waarom de fout optreedt en biedt een link naar een uitgebreid artikel dat stap-voor-stap begeleiding bevat om het probleem op te lossen. Wanneer we acties neemt door het volgen van de begeleiding, kunnen we opnieuw de regel om de oplossing te controleren. Als de fout niet wordt weergegeven in de lijst, is het probleem opgelost.
- Timer Jobs
  - Op de Central Admin startpagina, klik op Monitoring en kies Timer Jobs. SharePoint 2013 maakt gebruik van configureerbare Timer Jobs om gezondheidsgegevens te verzamelen en schrijft vervolgens de gegevens naar de logging folder en de Logging database. De gegevens worden vervolgens gebruikt in de rapporten om de gezondheidsstatus van de Farm servers weer te geven.
  - We kunnen een Timer Job reschedulen, in-of uitschakelen, en deze ook activeren on demand. Dagelijkse, wekelijkse en maandelijkse schema's omvatten ook een venster van executie. De Timer Service zal binnen een willekeurig tijdstip binnen de interval beginnen te lopen om de Timer Job op elke applicatie server te draaien. Deze feature is geschikt voor high-load jobs die draaien op meerdere servers op in de Farm.

- Het draaien van Timer Jobs tegelijkertijd op alle servers in de Farm kan de systeemprestaties beïnvloeden. We moeten ze zorgvuldig plannen om overlappings te voorkomen met andere Timer Jobs.
- Reporting
  - Op de Central Admin startpagina, klik op Monitoring en kies Reporting. Met deze functie kunnen we diagnostische gegevens en het verzamelen van gegevens configureren en administratieve en gezondheid rapporten bekijken. Omdat sommige configuraties veel schijfruimte zullen gebruiken en een negatieve invloed op de systeemprestaties kunnen hebben, moeten we zorgvuldig plannen welke configuraties we dienen in te stellen.
- Windows PowerShell
  - Windows PowerShell is een krachtig hulpmiddel voor het toezicht op SharePoint 2013. We kunnen opdrachten uitvoeren om de exacte logs die we willen bekijken, te verkrijgen. Voor meer informatie. Meer informatie: <http://technet.microsoft.com/en-us/library/ff463595.aspx>
- Systems Center 2012 – Operations Manager met Systems Center Management Pack voor SharePoint Server 2013
  - System Center 2012 - Operations Manager is een krachtige monitoring platform waarmee we diensten, apparaten en bewerkingen kunnen controleren op een groot aantal computers in een enkele console. Door het gebruik van Operations Manager, kunnen we de status, gezondheid, prestatie-informatie en waarschuwingen gegenereerd voor beschikbaarheid, performance, de configuratie en beveiliging situaties bekijken. Voor meer informatie, zie Operations Manager :
    - <http://technet.microsoft.com/en-us/library/hh205987.aspx>
  - Om Operations Manager te gebruiken om SharePoint-producten controleren, moeten we System Center Management Pack voor SharePoint Server 2013 installeren. We kunnen deze tool gebruiken om voor het controleren van gebeurtenissen, het verzamelen van SharePoint component-specifieke performance counters op een centrale locatie, en waarschuwingen verzamelen voor ingrijpen als dat nodig is.
  - Systems Center Management Pack voor SharePoint Server 2013
    - <http://www.microsoft.com/en-us/download/details.aspx?id=35590>
  - System Center Management Pack voor SharePoint Foundation
    - <http://www.microsoft.com/en-us/download/details.aspx?id=35590>
- Event Viewer
  - Event Viewer is een Microsoft Management Console (MMC)-module. Het laat ons event logs bekijken en beheren. Het is een zeer handig hulpmiddel voor het oplossen van problemen. We kunnen een filter instellen voor bepaalde gebeurtenissen in meerdere logboeken, en opnieuw gebruiken van event filters als aangepaste weergaven.
    - <http://technet.microsoft.com/library/cc766042>
- SharePoint Developer Dashboard
  - Deze tool biedt diagnostische informatie die kan helpen een ontwikkelaar of systeembeheerder de prestaties van SharePoint-webpagina's te laten analyseren. Dit hulpprogramma kan helpen als een pagina langzaam wordt geladen, een webpart niet werkt, of als er een database query op de pagina niet presteert. De SharePoint



Developer Dashboard is standaard uitgeschakeld. We kunnen het inschakelen met Windows PowerShell.

- <http://msdn.microsoft.com/en-us/library/ff512745.aspx>

- Windows Management Instrumentation
  - WMI biedt vele classes voor ons om de SharePoint 2013 omgeving te monitoren. Voor elk beheersbaar bron, is er een overeenkomstige WMIklasse. Omdat de mogelijkheden zo uitgebreid zijn, is het noodzakelijk dat wanneer hier gebruik van wordt gemaakt, dat er een plan aan ten grondslag ligt voordat dit wordt gebruikt en geïmplementeerd.
    - <http://technet.microsoft.com/library/ee198925.aspx>
- SQL Server Reporting Services (SSRS)
  - SQL Server Reporting Services biedt een volledig range van kant-en-klare hulpmiddelen en diensten om te helpen bij het maken, implementeren en managen van rapporten voor de organisatie. Het heeft ook de programming features die ons de mogelijkheid geven om rapportage functionaliteit uit te breiden en aan te passen. Door het gebruik van SQL Server Reporting Services, kunnen we interactieve, tabellen, afbeeldingen, of free form rapporten van relationele, multidimensionale, of XML-gebaseerde gegevensbronnen maken. We kunnen rapporten publiceren, rapportverwerking schedulen, of de toegang tot rapporten on-demand inregelen. We kunnen gebruik maken van SQL Server Reporting Services om rapporten op basis van vooraf gedefinieerde modellen te maken, en om interactief data te verkennen binnen het model. We kunnen kiezen uit een verscheidenheid viewing formats, rapporten exporteren naar andere toepassingen, en ons abonneren op gepubliceerde rapporten. De rapporten die we maken kunnen worden bekeken via een web-based-verbinding of als onderdeel van een Windows-applicatie of een SharePoint-site.
    - [http://technet.microsoft.com/library/ms159106\(v=sql.105\).aspx](http://technet.microsoft.com/library/ms159106(v=sql.105).aspx)
- SharePoint Health Analyzer, Timer Jobs, Rapportage, en Windows PowerShell zijn ingebouwde functies, terwijl Operations Manager, System Center Management Pack voor SharePoint Server 2013 en SQL Server Reporting Services onafhankelijke gereedschappen zijn. SharePoint Developer Dashboard en WMI zijn ingebouwde gereedschappen en bestemd voor ontwikkelaars en systeembeheerders. Deze tools zijn complementair en toe te passen bij verschillende scenario's.
- De volgende tabel geeft een overzicht van deze tools. We moeten de balans van de voor- en nadelen van de monitoring tools overwegen wanneer we bepalen welke hulpmiddelen te gebruiken in bepaalde situaties.
  - Zie Appendix A.3.1. : [Appendix A.3.1](#)

Voor een SharePoint-farm met niet meer dan 10 servers, raden wij aan in ieder geval het installeren van de volgende onafhankelijke gereedschappen:

- System Center 2012 - Operations Manager met System Center Management Pack voor SharePoint Server 2013
  - Bewaakt de gezondheidstoestand van SharePoint-producten.
- SQL Server Reporting Services
  - Implementeren als u de Reporting Services gebruiken om Report Definition Language (RDL) bestanden, en het ontwerp verslagen van Visual Studio en PerformancePoint Dashboard Designer bekijken.

### 3.2 HET IDENTIFICEREN VAN VERSCHILLENDE MONITORING SCENARIO'S

Identificeer de scenario's die we willen controleren, bijvoorbeeld gezondheid, workflows, zoeken, SQL Server, virtuele omgevingen en prestaties. De volgende tabel toont een aantal belangrijke scenario's en de monitoring tools die we kunnen gebruiken om deze scenario's te volgen.

Zie Appendix A.3.2. : [Appendix A.3.2](#)

### 3.3 BEPAAL DE DAGELIJKSE, WEKELIJKSE EN MAANDELIJKSE TE MONITOREN ITEMS

We kunnen zelf bepalen welke items dagelijks, wekelijks en maandelijks gecontroleerd moeten worden op basis van de paragraaf "Daily Tasks" (op pagina 43), de sectie "Weekly Tasks" (op pagina 51), de sectie "Maandelijks Tasks" (op pagina 52), en de sectie "Impromptu Tasks" (op pagina 53) in de Operations Framework en checklists (white paper: ).

<http://technet.microsoft.com/en-us/library/gg277248.aspx>

<http://go.microsoft.com/fwlink/p/?LinkId=203289>

### 3.4 ADVIES VOOR HET MONITOREN BINNEN DE SHAREPOINTCUSTOMER FARM OMGEVING IN AZURE

Monitoren vind op veel verschillende onderdelen plaats. Er zijn ook verschillende elementen die bekeken moeten worden, verschillende applicaties. Binnen SharePointCustomer zullen drie verschillende soorten applicaties gemonitored worden. Dat zijn de SharePoint Apps, de SandBoxed Solutions en de Farm solutions. Alle hebben hun eigen eisen betreffende de monitoring. Dit is in dit document beschreven. In eerste instantie zullen deze verschillende onderdelen op minimaal niveau gemonitored te worden. Vanuit deze minimale basis dient bekeken te worden welke extra onderdelen erbij gevoegd moeten worden om de stabiliteit van de SharePoint Farm stabiel en veilig te houden. Buiten deze drie applicatie onderdelen die op regelmatige basis bekeken moeten worden, dient ook de Farm als geheel gemonitored te worden. Ook de infrastructuur van Azure of on premise waarop de SharePoint farm draait is onderdeel van het monitoringsplan.

Kort samengevat zijn er de volgende onderdelen te onderscheiden om te monitoren binnen SharePointCustomer. De informatie die gegevens is in dit document en die berekening hebben opmonitoren, kunnen op deze onderdelen worden toegepast.

- Monitoren van Apps
- Monitoren van Sandbox Solutions
- Monitoren van Farm Solutions
- Monitoren van samenspel tussen soorten applicaties (bovenstaand)
- Monitoren gegevensbronnen die binnen SharePoint ontsloten zijn.
- Monitoren gebruikers binnen SP en AD en combinatie.
- Scm integratie
  - Dit is een apart hoofdstuk
  - Er is een Scm element voor SP en databases
  - Er zijn nog andere Scm elementen voor infra
  - Dit monitoren vergt in eerste instantie een goed plan daaromtrent.
  - Dit plan dient zelf gemonitored te worden.
- Monitoren van de Farms als geheel
  - On premise
  - In de Cloud
  - Combinatie on premise en in de cloud
- Monitoren van de infrastructuur
  - Database Servers (Windows Event Logs, database logs)
  - App Servers (Windows Event logs)
  - WFE Servers (Windows Event Logs)
  - Netwerk monitoring (Routers, kabels, throughput etc)

### 3.5 BEPAAL DE MENSEN DIE GAAN MONITOREN

We dienen in te schatten hoeveel mensen we nodig hebben om toezicht op de SharePoint 2013 omgeving te houden. Afhankelijk van de grootte van de SharePoint-omgeving, zou je toegewijde mensen moeten hebben om toe te zien op de gezondheid van SharePoint-omgeving. Voor zeer grote Enterprise Farms, hebben we meestal een of twee mensen nodig die toegewijd zijn aan elke farm.

Bij de planning voor de monitoring personeel kunnen we rekening houden met het volgende:

- Kleine farms (kleine gebruikersgroep, of branche (LOB) applicaties)
- Medium farms (Enterprise, service-applicatie, enz.)
- Grote enterprise farms (Large Enterprise, Vele service-applicatie, Self-Provisioning, enz.)
- Hoge beschikbaarheids eisen (service level agreements, SLA's)
- Hoge prestatie-eisen (pagina render keer)
- Gebruiker grootte basis (hoeveel unieke gebruikers)
- Gelijktijdige gebruikers (hoeveel gebruikers actief te raken van de portaalsite in een keer)
- Aanvragen per seconde (ook tijdens de piekuren)
- operationele looptijd
- management tools

De volgende tabel geeft bij benadering aan hoeveel mensen van verschillende niveaus nodig zijn om toezicht te houden op de SharePoint-omgeving, afhankelijk van het aantal servers. Merk op dat deze tabel slechts een zeer ruwe schatting geeft.

Zie Appendix A.3.4. : [Appendix A.3.4](#)

### 3.6 HET CREËREN VAN EEN RESPONS PLAN

Als een best practice, maak een response plan om voor te bereiden op mogelijke problemen en om aan te geven welke acties te nemen wanneer er een probleem optreedt.

Het response plan moet SharePoint Health Analyzer regels die al werkbare oplossingen bieden, uitsluiten. Voor een probleem dat nog geen onmiddellijke oplossingen heeft, moeten we de logs onderzoeken met behulp van monitoring tools zoals de Event Viewer om een oplossing te vinden.

De volgende tabel toont een aantal factoren die je moet overwegen wanneer we een reactie plan ontwikkelen.

Zie Appendix A.3.5. : [Appendix A.3.5](#)

### 3.7 RELATED RESOURCES

[Monitor health in SharePoint 2013](#)

[Overview of monitoring in SharePoint 2013](#)

[Monitor health in SharePoint 2013](#)

[Monitor apps for SharePoint for SharePoint Server 2013](#)

[View search diagnostics in SharePoint Server 2013](#)

[Monitoring and maintaining SharePoint Server 2013](#)

[Monitor cache performance in SharePoint 2013](#)

[Monitor SharePoint Server by using Visio Services and System Center Operations Manager](#)

[Monitor workflows \(SharePoint Server 2010\)](#)

[Storage and SQL Server capacity planning and configuration \(SharePoint Server 2013\)](#)

[Monitor SharePoint Server by using Visio Services and System Center Operations Manager](#)

[Monitoring: virtual environment](#)

[Monitoring and Diagnostic logging in Business Connectivity Services overview \(SharePoint Server 2010\)](#)

[Performance Monitoring Getting Started Guide](#)

[Network Monitor](#)

[Monitoring Server Planning and Architecture Guide for PerformancePoint Server 2007](#)

## 4 APPS VOOR SHAREPOINT EN OFFICE

---

In SharePoint 2013, heb je nu de mogelijkheid om Apps voor SharePoint maken. Dus wat is een app? Het is best te omschrijven als een oplossing die een lichte voetafdruk draagt en gebruik maakt van op standaarden gebaseerde technologieën zoals HTML5, JavaScript en OAuth. Apps hebben een lichte voetafdruk, omdat ze niet echt op de host server te installeren, en dat betekent dat ze niet te veel van een SharePoint-site vragen met overmatige API calls. In plaats daarvan, heb je een verscheidenheid van hosting opties voor een App die je de flexibiliteit geeft om deze diensten toe te voegen aan uw SharePoint-site beheren.

Dus Apps voor SharePoint bieden een nieuwe methode om bepaalde informatie of functionaliteit te leveren aan een SharePoint-site. Een App voor SharePoint is een klein, makkelijk te gebruiken, stand-alone SharePoint applicatie die een specifieke eindgebruikers of zakelijke behoefte oplost. Site-eigenaren kunnen Apps voor SharePoint vinden en downloaden van een openbare SharePoint Store of intern van hun organisatie App Catalog om deze te installeren op hun SharePoint-sites.

Deze Apps voor SharePoint integreren het beste van het web met SharePoint 2013. Zij vervangen niet SharePoint-features solution packages, die SharePoint-sites aanpassen of verbeteren. In tegenstelling tot de features en solutions, die SharePoint Administrator dienen te installeren, zijn Apps voor SharePoint stand-alone applicaties die eigenaren van sites kunnen toevoegen aan hun SharePoint-sites. De Apps voor SharePoint hebben een eenvoudige levenscyclus - ze kunnen worden geïnstalleerd, opgewaardeerd, en verwijderd door site-eigenaren.

### 4.1 TYPE APPS

Onafhankelijk van de gekozen hosting variant dienen een aantal zaken te gebeuren aan de kant van SharePointCustomer om het mogelijk te maken Apps te installeren en te gebruiken. Omdat we bij SharePointCustomer te maken hebben met een OnPremise SharePoint omgeving in Azure, zal alleen de eerste variant gebruikt worden. De tweede variant van Azure autohosted is alleen te gebruiken voor SharePoint Online Services.

De derde variant geeft te veel beheer uit handen en zal in eerste instantie niet gebruikt worden. Deze variant wordt wel verder onderzocht op implicaties betreffende infra, beheer, test, backup en security. Als deze onderdelen naar behoeven zijn besproken binnen SharePointCustomer, dan zal deze optie wellicht ook aangeboden worden. Op dit moment is dat nog niet het geval.

Zie Appendix A.4.1. : [Appendix A.4.1](#)

### 4.2 VOORBEREIDEN OMGEVING VOOR DE APPS

Om het mogelijk te maken Apps te draaien in een SharePoint omgeving dienen er een aantal zaken ingericht te worden. Dit is belangrijk voor zowel SharePointCustomer alsook voor de leveranciers die hun Apps willen testen. Allereerst is er een geldig domein nodig zoals bijvoorbeeld contosoApps.com. Vervolgens dienen er een aantal zaken ingericht te worden op de DNS server, zoals een forward lookup

en een CNAME alias van het App domein naar het SharePoint domein. Voor SSL dient er een wildcard SSL certificaat te zijn.

Wanneer de zaken geregeld zijn op de DNS server, zijn er wat acties die binnen SharePoint gedaan moeten worden. De Subscription Settings service applicatie moet geconfigureerd worden en datzelfde geldt voor de App Management service applicatie. Als deze beide zaken geregeld zijn binnen SharePoint, dan moeten de App URLs geconfigureerd worden in SharePoint.

Zie voor het plaatje Appendix A.4.2 : [Appendix A.4.2](#)

Het kan voorkomen dat de Subscription Settings Service Application wel aanwezig is bij de “Manage Services” in de Central Admin, maar dat deze niet te vinden is bij de “Manage Service Applications”. Om de Apps mogelijk te maken dienen we ook de Apps URLs te configureren. Als we echter de Service Application “Subscription Settings Service Application” niet hebben en de bijbehorende proxy niet is geconfigureerd, dan komt de volgende melding:

## Configure App URLs ⓘ

The Subscription Settings service and corresponding application and proxy needs to be running in order to make changes to these settings.

We komen hier terecht als we in CA kiezen voor Apps en vervolgens voor Configure App Urls. Om de “Subscription Settings Service Application” te installeren en te configureren dienen we een PowerShell script draaien die dat allemaal voor ons doet. Het is belangrijk om in dit script ook de DatabaseServer en DatabaseCredentials op te geven.

### 4.2.1 Related Resources

<http://technet.microsoft.com/en-us/library/fp161236.aspx>

<http://blogs.msdn.com/b/kvosters/archive/2013/04/11/the-subscription-settings-service-and-corresponding-application-and-proxy-needs-to-be-running-in-order-to-make-changes-to-these-settings.aspx>

## 4.3 DEPLOYMENT

De App zal als een App package worden aangeleverd door de leverancier. SharePointCustomer zal de App uploaden en deployen nadat de technische en functionele documentatie door alle betrokken partijen is goedgekeurd.

### 4.3.1 Publiceren naar een app catalogus

Omdat we in eerste instantie Apps maken voor gebruik binnen onze eigen onderneming of voor een specifieke zakelijk klanten, in plaats van het grote publiek, willen we de Apps publiceren naar een interne App catalogus gehost binnen onze SharePoint farm. Een eigen App catalogus is een speciale site collectie in een SharePoint 2013 webapplicatie (of een SharePoint Online huurovereenkomst) dat documentbibliotheken heeft die gastheer zijn voor Apps voor SharePoint en Apps voor Office. Aanbrengen van de catalogus in zijn eigen siteverzameling maakt het makkelijker voor de web applicatie beheerder om machtigingen voor de catalogus te beperken.

Uploaden van een App voor SharePoint naar een corporate App catalogus is zo eenvoudig als het uploaden van een bestand naar een SharePoint-documentbibliotheek. U vult een pop-up vorm waarin u de lokale URL van de App-pakket opgeeft en andere informatie, zoals de naam van de App. Als u de App aan een App catalogus toevoegd, zijn er vergelijkbare controles en Apps die niet voorbij deze controles komen, zijn gemarkeerd als ongeldig of worden uitgeschakeld in de catalogus.

Site-eigenaren kunnen Apps toevoegen voor SharePoint om hun sites. Als een App SharePoint componenten bevat, worden deze componenten opgeslagen in een subweb van de site die automatisch wordt aangemaakt wanneer je de App installeert. Apps hebben hun eigen, geïsoleerde URL's, die gescheiden is van de URL van de site die de App bevat. Als de App een Provider-hosted of Windows Azure autohosted App is, dan worden de App componenten opgeslagen op die locaties. Bijvoorbeeld, in het volgende diagram, App1 bevat aangepaste business logica en wordt opgeslagen op een externe server - het is een Windows Azure autohosted app en geen content in een subweb van de site. App2 is een SharePoint hosted App met alleen SharePoint componenten. Inhoud van App2 wordt opgeslagen in een subweb van de site waarop het is geïnstalleerd. Voor SharePointCustomer geldt alleen de App2 variant.

Voor plaatje zie Appendix A.4.3.1. : [Appendix A.4.3.1.](#)

## 4.4 MANAGEMENT

Voordat we toestaan dat site-eigenaren Apps installeren in een SharePoint-omgeving, moet we plannen hoe we ze willen steunen. We dienen het beleid van de organisatie rond Apps voor SharePoint te bepalen. Plan de configuratie-instellingen om mede te bepalen hoe de Apps te beheren en controleren.

### 4.4.1 Governance: bepaal de App for SharePoint policy voor de organisatie

De eerste beslissing over apps voor SharePoint is de mate waarin je ze wilt gebruiken in de organisatie en het beleid voor het gebruik ervan . De volgende vragen kunnen ons helpen dit beleid te bepalen:

- Willen we dat site-eigenaren in staat zijn om apps te installeren en te gebruiken voor SharePoint ?
  - Als dat zo is , blijvend lezen en vormgeven van ons beleid.
  - Zo niet , dan kunnen we de instellingen gebruiken in de SharePoint- Store om te controleren of site-eigenaren Apps kunnen kopen van de SharePoint- Store. Je kunt geen site-eigenaren blokkeren van het bekijken van de SharePoint- Store. Echter, we kunnen



voorkomen dat ze kopen of downloaden van Apps voor SharePoint . Als we ervoor kiezen om te voorkomen dat site-eigenaren Apps kunnen installeren voor SharePoint vanaf de SharePoint- Store, dan moeten we een beleidsverklaring maken en site-eigenaren vertellen dat we niet hebben gekozen om Apps te ondersteunen voor SharePoint. Anders zal de enige boodschap naar site-eigenaren een fout zijn wanneer zij proberen om een App voor SharePoint te installeren.

- Aangezien het erop lijkt dat we bij SharePointCustomer over het algemeen Apps zullen ondersteunen, dienen alleen de site eigenaren geïnformeerd te worden waarvan men niet wil dat die specifieke eigenaren Apps kunnen installeren.
- Willen we de Apps beperken of controleren die site-eigenaren kunnen installeren en gebruiken ?
  - Als dat zo is , dan kunnen we een van de volgende of beide doen :
    - Het opzetten van een App Catalog om een set van Apps bieden voor SharePoint die site-eigenaren kunnen installeren en gebruiken .
    - Gebruik de App functie Verzoek aan de inkoop en licentiëring van Apps voor de controle.
    - In eerste instantie zullen we binnen SharePointCustomer ervoor kiezen om de toegestane Apps op een plek te zetten die via een procedure kenbaar wordt gemaakt aan de betreffende site-eigenaren en die zo beschikbaar worden gesteld om te gebruiken.
- In welke omgeving willen we dat Apps voor SharePoint beschikbaar te zijn? Bijvoorbeeld , intranet , extranet of internet omgevingen ?
  - We moeten beslissen welke omgevingen geschikt zijns voor het gebruik van Apps voor SharePoint en configureer de instellingen voor Apps voor SharePoint alleen in die farms of voor die webapplicaties .
  - In eerste instantie zullen de Apps intern gebruikt worden en alleen interne Apps worden gebruikt. Voor Apps vanuit de publieke store geldt een aparte procedure en alleen daarvoor aangewezen personen mogen deze publiek Apps downloaden en installeren. Hier zal een procedure voor opgesteld worden.
- Willen we bepalen wie Apps kunnen kopen voor SharePoint ?
  - Geen expliciete instelling kan voorkomen dat een site-eigenaar een App kan aankopen voor SharePoint . We kunnen echter een verzoek proces definiëren dat vereist dat site-eigenaren een verzoek indienen dat door de organisatie wordt gecontroleerd om ervoor te zorgen dat alleen de juiste personen aankopen.
- Willen we controleren wie Apps voor SharePoint kunnen installeren?
  - Als minimum , moet een gebruiker rechten voor het beheren van een website hebben en machtigingen voor het maken subsites, voordat deze gebruiker een App voor SharePoint kan installeren. Standaard worden deze machtigingen alleen beschikbaar gesteld voor gebruikers die het machtigingsniveau Volledig beheer hebben of die in de groep site-eigenaren zitten.
  - Om te kunnen controleren wie Apps installeren, zal een groep aangemaakt worden waarin personen worden toegevoegd die Apps mogen installeren. Dit geldt voor interne alsook voor publieke Apps. Naast deze groep zal er ook een groep zijn die alleen interne Apps mag installeren. Hierover moeten nog een aantal beslissingen genomen worden.
- Willen we specifieke Apps bieden voor SharePoint in onze omgeving ?

- Dan kunnen we Apps toevoegen voor SharePoint voor onze eigen organisatie in de App Catalog .
- In voorgaande delen is aangegeven wie welke Apps mogen installeren. Hoewel deze procedures nog verder uitgewerkt moeten worden, kunnen we al wel stellen dat dus niet iedereen zomaar Apps mag installeren, publieke alswel interne, ook al zijn het site-eigenaren. Om problemen te voorkomen gaan we ervan uit dat alle Apps die geïnstalleerd dienen te worden in een aanvraagprocedure komen.
- Willen we controleren hoeveel en welke apps voor SharePoint er geïnstalleerd worden?
  - We moeten expliciet de Apps kiezen voor SharePoint om te controleren voor een farm.
  - Hier dient een procedure voor opgezet te worden welke Apps we hoe, wanneer, door wie, waarmee en welke we willen monitoren. Uiteraard dient in de procedure opgenomen te zijn welke acties wanneer genomen moeten worden.

Zie voor flows Appendix A-4.4.1. : [Appendix A.4.4.1](#)

#### 4.4.2 Deployment richtlijnen vanuit SharePointCustomer

Als we een App oplossing creëren, zijn er grenzen aan het type bronnen die we kunnen implementeren. Een App mag in principe geen server side code draaien. Echter, onze organisatie wil toch een aantal zaken beperken of verbieden bij de inzet van App solutions vanwege beveiliging, mogelijkheden of prestatie. In veel gevallen kan de aanvraag van een leverancier ook een formele code review ondergaan voordat de oplossing binnen de server-omgeving kan worden geïmplementeerd.

##### 4.4.2.1 App mogelijkheden, eisen en wensen

Apps kunnen niet worden uitgevoerd zonder enige code toegangsbeveiliging of beperkingen en lopen niet via dezelfde proces identiteit als de code die de Farm Solutions oproepen. De code zal niet worden uitgevoerd in het proces IIS (W3wp.exe), de SharePoint Timer-proces (Owstimer.exe), of een service application process, onafhankelijk van de uitvoering van de logica. Als gevolg daarvan, zal de code worden uitgevoerd met beperkingen, met andere woorden, kan de code niet doen wat het SharePoint-platform zelf kan doen. Wanneer de veiligheid of de stabiliteit niet voor significante problemen kan zorgen, of wanneer de toepassing een hoog gehalte aan functionele en schaalbaarheids testen dient te ondergaan, dan kan de Farm solution een geschikte keuze zijn. In een ander geval dienen we te overwegen om alleen de componenten die specifiek een Farm implementatie-oplossing nodig hebben ook te deployen als een Farm Solution. Onderdelen die binnen de sandbox-omgeving kunnen draaien moeten worden ingezet in een Sandbox oplossing. Aangezien het App model binnen SharePointCustomer de voorkeur heeft, geldt de regel: "Een App tenzij".

Bovenstaande betekent dus dat alleen Farm solutions worden toegestaan wanneer de leverancier iets heeft gemaakt dat niet in een App of Sandbox solution gemaakt kan worden. Het kan ook zijn dat de leverancier redenen heeft om van deze policy af te wijken. Deze reden zal dan door SharePointCustomer bekeken worden en beoordeeld op validiteit. Per geval zal hier een beslissing genomen worden.

##### 4.4.2.2 Controle op ontwikkeling en stabiliteit – Code controle

Apps zijn o.a onderworpen aan controle en resource throttling mechanieken. Slecht geschreven code in een Farm oplossing kan de prestaties en stabiliteit van de server farm als geheel niet in gevaar brengen. In tegenstelling tot Farm Solutions hoeven dit soort problemen niet voorkomen te worden. Ondanks deze ingebouwde veiligheid dienen we zorgvuldig de App code te analyseren op problemen die geheugenlekken of proces timeouts kunnen veroorzaken. Bijvoorbeeld, ontwikkelaars worden vaak geconfronteerd met de volgende valkuilen die de prestaties negatief kunnen beïnvloeden:

- De ontwikkelaar heeft niet juist de dispose gebruikt bij de objecten SPSite en SPWeb voorwerpen na gebruik.
- De ontwikkelaar heeft basic iteratie gebruikt om door de items van grote lijsten te lopen i.p.v. het gebruik van queries op lijsten
- De ontwikkelaar heeft veel for each loops gebruikt om data te aggregeren i.p.v. het gebruik van SPSiteDataQuery of andere recommended data aggregatie methoden
- De ontwikkelaar kan recursieve methode hebben gebruikt om informatie te doorlopen in elke site binnen een site collectie.

- De ontwikkelaar heeft niet alle verbindingen gesloten bij externe systemen na gebruik.
- De ontwikkelaar kan heeft niet alle time-outs ondervangen bij het aansluiten op externe systemen.
- De ontwikkelaar kan overmatig gebruik of onjuist gebruik gemaakt hebben van de Session State.
- Er worden zaken weggeschreven in de Web.config of andere files en dat veroorzaakt problemen
- De foutafhandeling is niet afdoende ingericht. Het is de bedoeling dat er geen vage SharePoint standaard fouten meer verschijnen, maar dat deze worden ondervangen en via een custom foutformulier aan de gebruiker worden getoond. De Solution zou niet mogen crashen en alle crashpunten dienen ondervangen te zijn. Alle code waarin risico's kunnen ontstaan, dienen in een Try Catch te worden of aan een error systeem gekoppeld te worden.

Dit is geen uitputtende lijst, maar het laat zien dat er veel verschillende manieren zijn waarop we onnodig onze SharePoint omgeving kunnen vertragen. Om risico's voor de Farm stabiliteit te minimaliseren, moeten we de App code reviewen tegen alle best practice bij de desbetreffende functionele gebieden. Risico's moeten beschreven zijn.

In het beste geval zou SharePointCustomer ook de ontwikkelmethodiek moeten kunnen voorschrijven met coding guidelines, ontwikkelproces, testproces, OTAP voorschriften, code review e.d. Wanneer dit niet het geval is, dan moet tenminste enig zicht zijn op deze onderdelen bij de leverancier.

#### 4.4.2.3 Security

Hoewel het App model aanzienlijk veiliger is dan Farm Solutions zijn de risico's wat subtieler. SharePoint 2013 Apps gebruiken App permission request scopes en permission requests om het level aan te geven waarbinnen een App mag draaien. Tevens dient di tom het permissie level aan te geven data aan de App is toegewezen. Deze App permission request scope geeft de locatie aan binnen de SharePoint 2013 hierarchy waarop de permissions betrekking hebben. Wanneer een te grote permissie wordt toegewezen aan een App, terwijl een lagere de bedoeling was, dan is er sprake van een risico. **SharePointCustomer wil dat de leverancier aangeeft per App wat het permission level dient te zijn, waarom en voor wie.** Als hiervan wordt afgeweken, dan dient dit met de betreffende stakeholders van SharePointCustomer besproken te worden. We onderscheiden de volgende permission request scopes:

- SPSite – Deinieert de app permission request scope als een SharePoint 2013 Site Collectie
- SPWeb – Deinieert de app permission request scope als een SharePoint 2013 web site
- SPList - Deinieert de app permission request scope als een SharePoint 2013 List
- Tenancy - Deinieert de app permission request scope als een SharePoint 2013 tenancy

Als een app toestemming wordt verleend voor een bepaalde scope, dan geldt de toestemming ook voor de kinderen van die strekking. Bijvoorbeeld, als een app toestemming is verleend voor een website met behulp van de SPWeb scope, dan heeft de app ook toestemming voor elke lijst (SPList scope) dat is opgenomen in de SPWeb reikwijdte en alle lijstitems binnen elke lijst. Omdat toestemmings verzoeken worden gedaan zonder informatie over de topologie van de siteverzameling waar de app wordt geïnstalleerd, wordt de reikwijdte uitgedrukt als een type in plaats van als de URL van een specifiek locatie.

Deze scope types worden uitgedrukt URI. Inhoudsdatabase gerelateerde toestemmingen worden georganiseerd onder de URI: <http://sharepoint/content>. De volgende tabel geeft een URI voorbeeld voor de scope van elke App toestemmings verzoek.

Zie Appendix A.4.4.2.3 : [Appendix A.4.4.2.3](#).

#### 4.4.2.4 Documentatie

Voor App solutions is het noodzakelijk dat de gehele solution goed is gedocumenteerd. Dit betekent dat er een HLD, DLD, FO en TD aanwezig dient te zijn. Deze documenten dienen met de betreffende stakeholders binnen SharePointCustomer gecommuniceerd te worden en ze moeten door deze stakeholders worden goedgekeurd. Wanneer de stukken niet worden goedgekeurd, dan is er sprake van een issue waar een manager met voldoende bevoegdheid een uitspraak over dient te doen. Het kan voorkomen dat een afkeuring van één van de documenten of delen ervan overruled worden. De reden van dit overrulen moet dan worden beschreven en gedocumenteerd. Dit dient in de log van het betreffende project te worden opgeslagen. Risico's dienen in de risicolog van het betreffende project toegevoegd te worden.

## 4.5 MONITORING

Buiten het monitoren van de Apps zelf, blijft het belangrijk dat de algemene elementen van SharePoint monitoring gebruikt worden. Het primaire doel van de monitoring is om een gezonde SharePoint 2013 omgeving te waarborgen zodat we onze diensten en prestatiedoelstellingen zoals korte responstijd kunnen bereiken. We kunnen de controle functies van het Centraal beheer van de SharePoint website, System Center Management Pack voor SharePoint Server 2013 en Windows PowerShell scripts gebruikt voor het controleren van de SharePoint 2013 omgeving en diensten.

Logs en rapporten controleren de status van onze SharePoint 2013 omgeving en de status van verschillende diensten. We kunnen de logboeken van de logboekdatabase lezen. Het voordeel van het gebruik logboekdatabase is dat we onze view op de werkelijkheid kunnen configureren en exporteren van de logs naar Excel. De logs en rapporten van het Centraal Administratie helpen begrijpen hoe het SharePoint 2013-systeem actief is, te analyseren en te repareren bij problemen, en het bekijken van statistieken voor de sites. Bovendien, System Center Management Pack voor SharePoint Server 2013 biedt een end-to-end monitoring en rapportage systeem dat we kunnen gebruiken om SharePoint 2013 te controleren.

Het Monitoring van de SharePoint 2013 omgeving omvat de volgende taken:

Zie hoofdstuk 3 in dit document:

[tools, strategie en mogelijkheden voor het monitoren](#)

Naast deze onderdelen zijn er de elementen die nodig zijn om Apps te monitoren. We kunnen het Centraal beheer van SharePoint gebruiken om Apps toe te voegen en te verwijderen voor SharePoint en te checken op details en fouten. De Monitor Apps pagina toont de Apps voor SharePoint die een Farm Administrator kan monitoren. Elke App voor SharePoint dat wordt vermeld op deze pagina bevat informatie om administrator te helpen om de performance in de gaten te houden. Bijvoorbeeld, elke App voor SharePoint biedt de volgende eigenschappen: Naam, Status, Bron, licenties in gebruik, licenties,

geïnstalleerde locaties, en runtime fouten. Een Farm beheerder kiest om Apps toe te voegen, te verwijderen en te controleren binnen SharePoint.

De Monitor Apps pagina vereist de volgende timer jobs die actief moeten zijn: de Search Analytics en de Usage File Import timer job:

- ECM analytics timer job naam: Usage Analytics timer job for Search Service
- Usage DB timer job naam: Microsoft SharePoint Foundation Usage Data Import

Er zijn meerdere manieren waarop een beheerder de fouten en het gebruik van gegevens voor Apps voor SharePoint kan bekijken. Door een App te selecteren in de Monitor Apps pagina kan een beheerder het lint gebruiken om de fout of het gebruik van gegevens voor die App te bekijken. Een beheerder kan ook op een App klikken in de lijst op de Monitor Apps pagina om de App detailpagina openen en dezelfde fout of gebruiksdetails te bekijken.

Het gebruik van App foutdetails gegevens kunnen worden uitgesteld tot 29 uur. De App gegevens zijn afhankelijk van wanneer de ECM analytics timer job is gepland. Wanneer de timer job loopt, verzamelt het gebeurtenissen van de vorige dag. Bijvoorbeeld, als de timer taak is gepland om 5 uur, dan zijn de meest recente gebeurtenissen die worden verzameld degene vanaf 23:59 de vorige dag. Een gebeurtenis die zich voordoet op 00:01 zal niet verschijnen in de App informatie pagina's tot maximaal 29 uur later.

Merk op dat als je de App foutendetails pagina bekijkt voor een bepaalde instantie van een App, het aantal fouten voor de App is gesynchroniseerd met de foutmeldingen in de lijst. Dit gebeurt omdat het aantal fouten worden weergegeven in de App error pagina met details in plaats van de gebeurtenissen die worden verwerkt door de ECM analytics timer job.

Voor SharePointCustomer is het raadzaam om een procedure te hebben voor twee onderdelen:

- Het monitoren van de Farm als geheel
- Het monitoren van de verschillende Apps
- Monitoring van services buiten SharePoint (voor als SharePoint down is)

Later zal ook beschreven worden hoe het monitoren van de Sandbox/Rest en Farm solutions dient te gebeuren. Net zoals er specifieke monitorings-elementen zijn voor de Apps, zijn deze ook aanwezig voor Sanbox/REST en Farm solutions. Omdat Farm Solutions over het algemeen een veel grotere vrijheid hebben dan Apps, is vooral bij Farm solutions het monitoren van essentieel belang.

Zie voor verdere monitoring informatie hoofdstuk 3 van dit document:

[tools, strategie en mogelijkheden voor het monitoren](#)

## 4.6 ONTWIKKELPROCES EN METHODIEK

Het ontwikkelproces van de Apps zal vooral geschieden onder regie van het proces van de leverancier. Wat wel gebruikelijk is, is dat SharePointCustomer op de hoogte is van de iteraties (sprints) en oplevermomenten van deelfunctionaliteit. SharePointCustomer zelf hanteert de Scrum methodiek voor het ontwikkelen van Software en van haar leveranciers verwacht ze een soortgelijk proces. Op zijn minst dient er een zekere mate van inzichtelijkheid aanwezig te zijn betreffende het ontwikkelproces van de leveranciers van SharePoint functionaliteit voor SharePointCustomer. Dit heeft twee belangrijke redenen:

- Bewaking van de kwaliteit van de software
- Bewaking van de doorlooptijd en de kosten van de software

Hoewel het ontwikkelproces een element is dat onder de verantwoordelijkheid valt van de leverancier van de software, heeft SharePointCustomer wel een aantal voorkeuren m.b.t. de onderdelen die aanwezig dienen te zijn binnen een ontwikkelproces en de methodiek daarbinnen en omheen. Aangezien SharePointCustomer voor de meeste zaken best practices volgt van Microsoft, tenzij anders voorgeschreven, geldt dit ook voor het ontwikkelproces en de methodiek. Daar dient aan toegevoegd te worden dat SharePointCustomer de algemene Scrum methodiek volgt. Deze methodiek is echter ingebed in de ontwikkeltools van Microsoft als ontwikkelproces template. De werkwijze van de leverende partij dient bij voorkeur hier niet teveel van af te wijken. Hieronder geven we een indicatie van de elementen van de ALM die de voorkeur heeft van SharePointCustomer.

- Test Driven Development
- Code Coverage
- Model First Design
- UML Diagramming and Design
  - Vanuit UML naar code en visa versa
- Daily Build strategy
- TFS Source control en version control
- Branching strategy
- Code review en tooling
  - Resharper
  - Style Cop
  - Code Analysis van Microsoft
  - FxCop
- OOAD en OOAP
- Iteratief design en ontwikkeling van de software

Wanneer een ALM goed is ingevoerd, dan kan met een burn down chart dagelijks de voortgang van het ontwikkelproces bekeken worden. Om dit mogelijk te maken dienen de workitems voortdurend van de juiste remaining hours voorzien te worden en de stand up meetings van Scrum zorgen voor een voortdurende verdeling en voortgang van de Use Cases, die vertaald zijn in de workitems. Ondanks het feit dat de leverende partij vrij is om het ontwikkelproces te kiezen en de ALM in te richten, zullen er toch enige vraagtekens ontstaan als hiervan teveel wordt afgeweken.

Het ALM systeem is ingebed binnen een set aan applicaties. Bij Appendix A.4.6. is goed te zien welke elementen dit zijn. Wanneer een leverancier de Apps maakt met andere tools, dan kan dit voorbeeld dienen als referentie betreffende welk soort elementen in een ALM aanwezig dienen te zijn.

[Appendix A.4.6.](#)

## 4.7 TESTEN

Zoals eerder vermeld is SharePointCustomer een voorstander van Test Driven Development. Voor elk stuk functionaliteit wordt dan een testcase gemaakt die dagelijks bekeken kunnen worden. Binnen Visual Studio betekent dat, dat er een rol belegd is die dagelijks kijkt naar de uitkomsten van de testcases. Dat kan via kleuren. Wanneer alle testcases groen zijn, dan betekent dat de gerelateerde functionaliteit werkt. Wanneer rode kleuren ontstaan, dan betekent dat dat er functionaliteit niet meer werkt. Als dit gecombineerd wordt met Daily builds en controles daarop, dan zal dagelijks de kwaliteit van de code inzichtelijk zijn. Voeg hieraan toe code coverage en de dekking is behoorlijk compleet. Hier kan door de leverende partij van afgeweken worden, maar ook hier geldt dat als deze afwijking groot blijkt te zijn, er bij SharePointCustomer toch vraagtekens zullen ontstaan die reden kunnen zijn voor een discussie omtrent kwaliteit en inzichtelijkheid.

### 4.7.1 Related Resources

[http://msdn.microsoft.com/en-us/library/aa730844\(v=vs.80\).aspx](http://msdn.microsoft.com/en-us/library/aa730844(v=vs.80).aspx)

[http://en.wikipedia.org/wiki/Test-driven\\_development](http://en.wikipedia.org/wiki/Test-driven_development)



## 5 SANDBOX SOLUTIONS EN CSOM/REST

---

Een Microsoft SharePoint Server solution is een inzetbare, herbruikbare verpakking die functies, sitedefinities, en andere functionaliteit kan bevatten. Oplossingen kunnen worden ingeschakeld of individueel uitgeschakeld. U kunt een oplossing direct implementeren op uw SharePoint Server farm, of u kunt de oplossing implementeren in een Sandbox. Een Sandbox is een beperkte executie omgeving die programma's in staat stelt om toegang te krijgen tot alleen bepaalde bronnen, en dat houdt problemen beperkt die zich voordoen in de Sandbox en van invloed kunnen zijn op de rest van de serveromgeving. Oplossingen die u implementeert in een Sandbox, die bekend staan als Sandbox-oplossingen, kunnen geen gebruik maken van bepaalde computer-en netwerkbronnen, en kunnen niet buiten de siteverzameling toegang krijgen tot gegevens.

### 5.1 DEPLOYMENT

#### 5.1.1 Deployment richtlijnen vanuit SharePointCustomer

Als we een Sandbox oplossing creëren, zijn er grenzen aan het type bronnen die we kunnen implementeren. Een Sandbox Solution mag in principe geen server side code draaien. Echter, onze organisatie wil toch een aantal zaken beperken of verbieden bij de inzet van Sandbox solutions vanwege beveiliging, mogelijkheden of prestaties. In veel gevallen kan de aanvraag van een leverancier ook een formele code review ondergaan voordat de oplossing binnen de server-omgeving kan worden geïmplementeerd.

##### *5.1.1.1 App mogelijkheden, eisen en wensen*

Sandbox solutions kunnen niet worden uitgevoerd zonder enige code toegangsbeveiliging of beperkingen en lopen niet via dezelfde proces identiteit als de code die de Farm Solutions oproepen. De code zal niet worden uitgevoerd in het proces IIS (W3wp.exe), de SharePoint Timer-proces (Owstimer.exe), of een service application process, onafhankelijk van de uitvoering van de logica. Als gevolg daarvan, zal de code worden uitgevoerd met beperkingen, met andere woorden, kan de code niet doen wat het SharePoint-platform zelf kan doen. Wanneer de veiligheid of de stabiliteit niet voor significante problemen kan zorgen, of wanneer de toepassing een hoog gehalte aan functionele en schaalbaarheids testen dient te ondergaan, dan kan de Farm solution een geschikte keuze zijn. In een ander geval dienen we te overwegen om alleen de componenten die specifiek een Farm implementatie-oplossing nodig hebben ook te deployen als een Farm Solution. Onderdelen die binnen de sandbox-omgeving kunnen draaien moeten worden ingezet in een Sandbox oplossing. Aangezien het App model binnen SharePointCustomer de voorkeur heeft, geldt de regel: "Een App tenzij".

Bovenstaande betekent dus dat alleen Farm solutions worden toegestaan wanneer de leverancier iets heeft gemaakt dat niet in een App of Sandbox solution gemaakt kan worden. Het kan ook zijn dat de leverancier redenen heeft om van deze policy af te wijken. Deze reden zal dan door SharePointCustomer bekeken worden en beoordeeld op validiteit. Per geval zal hier een beslissing genomen worden.

#### 5.1.1.2 Controle op ontwikkeling en stabiliteit – Code controle

Sandbox solutions zijn o.a onderworpen aan controle en resource throttling mechanieken. Slecht geschreven code in een Sandbox oplossing kan de prestaties en stabiliteit van de server farm als geheel niet in gevaar brengen. In tegenstelling tot Farm Solutions hoeven dit soort problemen niet voorkomen te worden. Ondanks deze ingebouwde veiligheid dienen we zorgvuldig de Sandbox code te analyseren op problemen die geheugenlekken of proces timeouts kunnen veroorzaken. Bijvoorbeeld, ontwikkelaars worden vaak geconfronteerd met de volgende valkuilen die de prestaties negatief kunnen beïnvloeden:

- De ontwikkelaar heeft niet juist de dispose gebruikt bij de objecten SPSite en SPWeb voorwerpen na gebruik.
- De ontwikkelaar heeft basic iteratie gebruikt om door de items van grote lijsten te lopen i.p.v. het gebruik van queries op lijsten
- De ontwikkelaar heeft veel for each loops gebruikt om data te aggregeren i.p.v. het gebruik van SPSiteDataQuery of andere recommended data aggregatie methoden
- De ontwikkelaar kan recursieve methode hebben gebruikt om informatie te doorlopen in elke site binnen een site collectie.
- De ontwikkelaar heeft niet alle verbindingen gesloten bij externe systemen na gebruik.
- De ontwikkelaar kan heeft niet alle time-outs ondervangen bij het aansluiten op externe systemen.
- De ontwikkelaar kan overmatig gebruik of onjuist gebruik gemaakt hebben van de Session State.
- Er worden zaken weggeschreven in de Web.config of andere files en dat veroorzaakt problemen
- De foutafhandeling is niet afdoende ingericht. Het is de bedoeling dat er geen vage SharePoint standaard fouten meer verschijnen, maar dat deze worden ondervangen en via een custom foutformulier aan de gebruiker worden getoond. De Solution zou niet mogen crashen en alle crashpunten dienen ondervangen te zijn. Alle code waarin risico's kunnen ontstaan, dienen in een Try Catch te worden of aan een error systeem gekoppeld te worden.
- **Sandbox Specifiek**
  - Vermijd het creëren van Static members
    - Wanneer een Sandbox solution voor de eerste keer geopend wordt op een server waarop de SharePoint Foundation Sandboxed Code Service draait (User Code Host Service), dan wordt er een applicatie domein voor de Sandbox worker process gemaakt en alle statische variabelen in de solution worden geladen in dat domein. Deze applicatiedomein blijft actief en wordt hergebruikt als deze zelfde sandbox solution elders wordt gebruikt vanaf een andere site collectie en een andere gebruiker. Een neven effect is dat deze statische

variabelen niet opnieuw worden geïnitialiseerd. De content van deze statische variabelen hoeven dus niet relevant te zijn in een andere context en dat kan leiden tot onvoorziene resultaten, oftewel functionele instabiliteit.

- Vermijd het raisen van unhandled Exceptions

- De code zou geen fouten moeten raisen die het niet afhandeld. Een unhandled exception in een Sandbox solution stopt alle andere sandbox solutions in de worker process, niet alleen dege met de unhandled exeptie. Om dezelfde reden is het belangrijk om alle mogelijke fouten op te vangen en via een gebruikersvriendelijke interface aan de gebruiker te tonen.

- Gebruik zowel de AllowPartiallyTrustedCallers Attribute alsook de SharePointPermission Attribute

- Code in de sandbox worker process kan alleen assemblies aanroepen die voorzien zijn van het attribuut AllowPartiallyTrustedCallersAttribute. Dat betekent dat de meeste custom assemblies die gedeployed worden in een Sandbox solution voorzien moeten zijn van dat attribuut. Ook ASP.NET vereist dit attribuut voor elke assembly dat overerft van WebPart. Om deze reden wordt dit attribuut door Visual Studio automatisch toegevoegd aan de AssemblyInfo file elke keer als een Sandbox Solution Project gestart wordt.
- Assemblies echter die dit attribuut hebben, kunnen een security risico gaan vormen als ze ooit eens worden gebruikt in een Farm Solution. Het zou namelijk elke partially trusted caller de gelegenheid geven om de classes van deze assemblies aan te roepen in een full trust omgeving. Als een class het SharePoint object model zou aanspreken, dan zou elke partially trusted caller dus toegang hebben tot het SharePoint objecten model. Om dit te voorkomen dient elke class die gemarkeerd is met AllowPartiallyTrustedCallers dat het SharePoint Object model aanroept voorzien moeten zijn van het volgende attribuut:  
`[Microsoft.SharePoint.Security.SharePointPermission(System.Security.Permissions.SecurityAction.LinkDemand, ObjectModel=true)]`
- Dat attribuut zorgt ervoor dat alleen callers met toegang tot het SharePoint objecten model deze class kunnen aanroepen

Dit is geen uitputtende lijst, maar het laat zien dat er veel verschillende manieren zijn waarop we onnodig onze SharePoint omgeving kunnen vertragen. Om risico's voor de Farm stabiliteit te minimaliseren, moeten we de Sandbox code reviewen tegen alle best practice bij de desbetreffende functionele gebieden. Risico's moeten beschreven zijn. Farms stabiliteit zal hier vooral bestaan uit functionele Far stabiliteit. Sandbox solutions worden bekeken met punten die eraan toegewezen zijn. Dit zijn resource punten en een soort strafpunten. Wanneer een Sandbox solution zich 'slecht' gedraagt, dan zullen de punten eraan toegewezen snel op zijn. Wanneer een Sandbox solution wordt gestopt en

andere solutions zijn daarvan afhankelijk, dan kan er een functionele instabiliteit ontstaan. Deze afhankelijkheden dienen onderzocht te worden en daarom ook is een analyse vooraf en een codereview belangrijk.

In het beste geval zou SharePointCustomer ook de ontwikkelmethodiek moeten kunnen voorschrijven met coding guidelines, ontwikkelprocess, testprocess, OTAP voorschriften, code review e.d. Wanneer dit niet het geval is, dan moet tenminste enig zicht zijn op deze onderdelen bij de leverancier.

## 5.2 MANAGEMENT

Zie 4.4 : [Management](#)

## 5.3 MONITORING

Zie 4.5 : [Monitoring](#)

## 5.4 ONTWIKKELPROCES EN METHODIEK

Zie 4.6 : [Ontwikkelproces en methodiek](#)

## 5.5 TESTEN

Zie 4.6 : [Testen](#)

## 6 FARM SOLUTIONS

---

Terwijl iedereen het over Apps heeft, zijn er nog steeds aanzienlijke investeringen in Full Trust Solutions (aka Farm Solutions) en het is zeker dat velen OnPrem implementaties deze nog steeds zullen overnemen bij het upgraden naar SharePoint 2013. De nieuwe SharePoint 2013 upgrade model maakt het mogelijk dat Sites blijven draaien in 2010 modus na upgraden en elke Site Collection dient expliciet individueel te worden opgewaardeerd. Niet de manier waarop het werkte in 2010 met Visual Upgrade, maar deze keer is er eigenlijk zowel een map Root 14 en 15 aanwezig en alle features en layouts van SharePoint 2010 worden mede uitgerold als onderdeel van de installatie van 2013.

Er is een hoofdmap waar SharePoint het grootste deel van haar programma plaatst-en de standaard locatie voor dit is "C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions \ [SharePoint Internal Version]". De uitvoeringen voor de laatste versies zijn 60 (6.0), 12, 14 en 15 nu. De locatie is ook bekend als "The xx hive."

Dit is geweldig in een upgrade-scenario, waar we na een platform upgrade eerst of alleen de nieuwe features van de 2013 willen delen met een paar gebruikers met behoud van een ongewijzigd ervaring voor de rest van de organisatie. Dit geeft ons ook de mogelijkheid om verschillende functies en mogelijkheden te hebben voor sites die de 2010 en 2013 mode hebben. Dit vereist echter wel wat extra gedachten bij het ontwikkeling en implementatie proces.

Omdat we nu kunnen hebben sites die lopen in zowel 2010 en 2013 mode, introduceert SharePoint 2013 een nieuw concept betreffende het Compatibility niveau. Nu kan het alleen zijn 14 of 15, maar je kunt je voorstellen dat er ruimte is voor groei. Dit compatibiliteits niveau is beschikbaar op siteverzameling en site (web-) niveau en kan gebruikt worden in codeconstructies en PowerShell-opdrachten. We zullen beginnen met het uitleggen hoe je het gebruikt bij het opbouwen en inzetten van WSP-bestanden voor SharePoint 2013 en vervolgens een paar dingen om op te letten en wat code tips.

### 6.1 DEPLOYMENT

#### 6.1.1 Deployment considerations

Als we de WSP-bestanden nemen van SharePoint 2010 en deployen met Add-SPSolution -> Install-SPSolution zoals we deden in 2010, dan zal SharePoint aannemen dat het een 2010-oplossing of een "14"-modus oplossing is. Als het niveau niet is opgegeven in de PowerShell command, dan wordt dit bepaalt op basis van de waarde van de SharePointProductVersion attribuut in het Solution manifest bestand van de WSP-pakket. De waarde kan op dit moment zijn 15,0 of 14,0. Als dit kenmerk ontbreekt, wordt aangenomen dat het 14,0 (SharePoint 2010) is en aangezien dit attribuut niet bestond in 2010, hebben slechts zeer goed geïnformeerd mensen dit opgenomen in de bestaande packages.

Voor PowerShell cmdlets met betrekking tot het installeren van solutions en features, is er een nieuwe parameter bijgekomen genaamd CompatibilityLevel. Dit kan de instellingen van het pakket zelf overschrijven en kan de volgende waarden aannemen: 14, 15, New, Old, All en "14,15" (het laatste betekent ook alles). De parameter is beschikbaar voor Install-SPSolution, Uninstall-SPSolution, Install-SPFeature en Uninstall-SPFeature. Er is geen manier om "All" versies op te geven in het pakket zelf - alleen de beoogde doelgroep - en daarom moeten deze parameters worden opgegeven als we willen implementeren voor beide doelen.

Het is belangrijk op te merken dat de Compatibility Level alleen bestanden beïnvloed die zijn gedeployed naar de map Templates in de 14/15 Root. Dat zijn: Features, layout-bestanden, afbeeldingen, ControlTemplates, etc. Dit betekent dat de bestanden buiten deze map (bv. een WCF service ingezet om de map ISAPI) zal worden gedeployed naar de 15/ISAPI, ongeacht welk niveau is ingesteld in de manifest file of in PowerShell. Bestanden zoals Assemblies in de GAC / Bin en bepaalde bronbestanden zullen ook worden gedeployed naar dezelfde locatie, ongeacht het compatibiliteits niveau.

Het is mogelijk om dezelfde oplossing te installeren zowel 14 en 15 modus, maar alleen als dit gebeurt in hetzelfde commando - waarin Compatibility Level ofwel "All" of "14,15" is. Indien het eerst wordt ingezet met 14 en vervolgens met 15, zal het een exceptie veroorzaken. Het kan worden geïnstalleerd met de Force-parameter, maar dit wordt niet aanbevolen omdat het andere fouten kan verbergen en kan leiden tot een onbekende toestand van het systeem.

Voor een goed overzicht van waar de bestanden terecht komen afhankelijk van de parameters en attributen die zijn opgegeven, kan gevonden worden in Appendix A.5.1.1.

Bij het retracten van de oplossingen, is er ook een optie om het compatibiliteit niveau opgeven. Als we dit niet opgeven, zal het allemaal retracted worden - zowel 14 en 15 bestanden, indien deze zijn geïnstalleerd. Wanneer gedeployed niveaus, kan er één ingetrokken worden, maar het echt belangrijke ding om te begrijpen is dat het niet alleen de bestanden zal trekken uit de versie-map, maar ook alle versie neutrale dossiers - zoals de Assemblies, ISAPI ingezet bestanden, etc. - waardoor alleen de bestanden uit de map Root niet retracted worden.

Om één en ander goed te plannen is het volgende wellicht raadzaam tijdens development/deployment:

- Als we alleen sites in 2013 mode te draaien, dan deployen van de oplossingen met CompatibilityLevel 15 of SharePointProductVersion 15.0.
- Als we uitvoeren met zowel de 2010 en 2013-modus, en we willen features en lay-out bestanden delen, dan deployen met (All of "14,15").
- Als we willen de bestanden en features die worden gebruikt in 2010 en 2013 mode differentiëren, dan moeten de solutions worden gesplitst in twee of drie solutions:
  - Een oplossing ("Xxx - SP2010"), waarin de bestanden en functies worden ingezet te deployen in de map 14 voor 2010-modus, inclusief code-behind (voor zaken als feature activation en Application pagina's), maar exclusief gedeelde assemblies en bestanden.
  - Een oplossing ("Xxx - SP2013"), waarin de bestanden en functies worden ingezet om te deployen naar de 15 map voor 2013-modus, inclusief code-behind (voor zaken als feature activation en Application pagina's), maar exclusief gedeelde assemblies en de bestanden.
  - Een oplossing ("Xxx - Common"), welke gedeelde bestanden (bijvoorbeeld gemeenschappelijke assemblies of webservices) bevat. Deze solution zou ook alle webapplicatie scoped features zoals bin-deployed assemblies en assemblies met SafeControl, dienen te bevatten.
- Als we alleen twee oplossingen willen hebben om verschillende redenen, kan de Common solution worden samengevoegd met de SP2013 solution want dit is waarschijnlijk degene die je het langste zal houden.
- De assemblies die worden gebruikt als code-bestanden voor de artefacten in SP2010 en SP2013 moeten verschillende namen hebben of tenminste verschillende versies om ze van elkaar te

onderscheiden. Web Parts moeten in de Common package gestopt worden en over de versies worden gedeeld, maar de geïnstalleerde Web Part templates kunnen uniek zijn voor een bepaalde versie modus.

#### 6.1.2 Zaken om extra aandacht aan te besteden

Er zijn een paar zaken die de moeite waard zijn ons bewust van te zijn, die in toekomstige updates kunnen zijn opgelost, maar waar we nu toch nog voor moeten oppassen.

Er kan een probleem zich voordoen bij het installeren van dezelfde oplossing in beide niveaus. Als je het installeert met niveau Alle en vervolgens probeert te verwijderen met level 14 twee keer, zal de logica denken dat het de oplossing volledig verwijderd heeft, maar de bestanden in de map 15/Templates zullen er nog steeds zijn. Om dit te herstellen kunnen we het met-Force installeren in Orphan-Level om vervolgens te verwijderen. Het is beter om niet in deze situatie te komen.

Een ander scenario waarmee we problemen kunnen krijgen is als we een oplossing in een Compatibility Level (hetzij door middel van PowerShell Parameter of manifest bestandsattribuut) installeren en vervolgens verwijderen met het andere niveau. Het zal dan de gemeenschappelijke bestanden verwijderen, maar laat de specifieke 14 of 15 map-bestanden in tact terwijl het wel aangeeft de solution volledig verwijderd te hebben.

Helaas is er geen publieke API te bevragen die Compatibiliteit Levels van een package laat zien. Dus we moeten het goed doen de eerste keer of zo snel mogelijk verhuizen naar native 2013 modus en packages (dit is waar we allemaal naartoe willen hoe dan ook).

#### 6.1.3 Code Patterns

Een extra tip is om te zoeken naar hard gecodeerde paden in aangepaste code zoals `_layouts` en `_controltemplates`. De `SPUtility` class is bijgewerkt met statische methoden om te helpen de huidige locatie op basis van de upgrade status van de site te ontleden. Zo zal `SPUtility.ContextLayoutsFolder` ons het pad naar de juiste map-outs geven. <http://msdn.microsoft.com/en-us/library/ms464967.aspx>

#### 6.1.4 Deployment richtlijnen vanuit SharePointCustomer

Als we een full-trust Farm oplossing creëren, zijn er geen grenzen aan het typen bronnen die we kunnen implementeren. Noch zijn er beperkingen op de locaties binnen het file systeem van de server waaraan we onze oplossingen kunnen toevoegen. Echter, onze organisatie wil toch een aantal zaken beperken of verbieden bij de inzet van Farm solutions vanwege beveiliging of prestaties. In veel gevallen kan de aanvraag van een leverancier ook een formele code review ondergaan voordat de oplossing binnen de server-omgeving kan worden geïmplementeerd.

##### 6.1.4.1 Farm Solution mogelijkheden

Full-trust Farm Solutions kunnen worden uitgevoerd zonder enige code toegangsbeveiliging of beperkingen en lopen via dezelfde proces identiteit als de code die de Farm Solutions oproepen. Meestal zal de code worden uitgevoerd in het proces IIS (W3wp.exe), de SharePoint Timer-proces (Owstimer.exe), of een service application process, afhankelijk van de uitvoering van de logica. Als gevolg daarvan, zal de code worden uitgevoerd zonder beperkingen, met andere woorden, kan de code doen wat het SharePoint-platform zelf kan doen. Wanneer de veiligheid of de stabiliteit niet voor significante problemen kan zorgen, of wanneer de toepassing een hoog gehalte aan functionele en schaalbaarheids testen dient te ondergaan, dan kan de Farm solution een geschikte keuze zijn. In een

ander geval dienen we te overwegen om alleen de componenten die specifiek een Farm implementatie-oplossing nodig hebben te deployen binnen een Farm Solution. Onderdelen die binnen de Sandbox-omgeving kunnen draaien, moeten worden ingezet als een Sandbox oplossing. Aangezien het App model binnen SharePointCustomer de voorkeur heeft, geldt de regel: “Een App tenzij”.

Bovenstaande betekent dus dat alleen Farm solutions worden toegestaan wanneer de leverancier iets heeft gemaakt dat niet in een App of Sandbox solution gemaakt kan worden. Het kan ook zijn dat de leverancier redenen heeft om van deze policy af te wijken. Deze reden zal dan door SharePointCustomer bekeken worden en beoordeeld op validiteit. Per geval zal hier een beslissing genomen worden.

#### **6.1.4.2 Controle op ontwikkeling en stabiliteit – Code controle**

Farm Solution zijn niet onderworpen aan enige controle of resource throttling mechanieken. Slecht geschreven code in een farm oplossing kan de prestaties en stabiliteit van de server farm als geheel in gevaar brengen. Om deze problemen te voorkomen, dienen we zorgvuldig de Farm code te analyseren op problemen die geheugenlekken of proces timeouts kunnen veroorzaken. Bijvoorbeeld, ontwikkelaars worden vaak geconfronteerd met de volgende valkuilen die de prestaties negatief kunnen beïnvloeden:

- De ontwikkelaar heeft niet juist de dispose gebruikt bij de objecten SPSite en SPWeb voorwerpen na gebruik.
- De ontwikkelaar heeft basic iteratie gebruikt om door de items van grote lijsten te lopen i.p.v. het gebruik van queries op lijsten
- De ontwikkelaar heeft veel for each loops gebruikt om data te aggregeren i.p.v. het gebruik van SPSiteDataQuery of andere recommended data aggregatie methoden
- De ontwikkelaar kan recursieve methode hebben gebruikt om informatie te doorlopen in elke site binnen een site collectie.
- De ontwikkelaar heeft niet alle verbindingen gesloten bij externe systemen na gebruik.
- De ontwikkelaar kan heeft niet alle time-outs ondervangen bij het aansluiten op externe systemen.
- De ontwikkelaar kan overmatig gebruik of onjuist gebruik gemaakt hebben van de Session State.
- Er worden zaken weggeschreven in de Web.config of andere files en dat veroorzaakt problemen
- De foutafhandeling is niet afdoende ingericht. Het is de bedoeling dat er geen vage SharePoint standaard fouten meer verschijnen, maar dat deze worden ondervangen en via een custom foutformulier aan de gebruiker worden getoond. De Solution zou niet mogen crashen en alle crachpunten dienen ondervangen te zijn. Alle code waarin risico's kunnen ontstaan, dienen in een Try Catch te worden of aan een error systeem gekoppeld te worden.

Dit is geen uitputtende lijst, maar het laat zien dat er veel verschillende manieren zijn waarop we onnodig onze SharePoint omgeving kunnen vertragen. Om risico's voor de Farm stabiliteit te



minimaliseren, moeten we de Farm Solution code reviewen tegen alle best practice bij de desbetreffende functionele gebieden. Wanneer een solution dingen wegschrijft in bestanden, de web.config bijvoorbeeld, dan moet dat goed gedocumenteerd zijn en dienen er bij de toevoegingen commentaarregels aanwezig te zijn. Risico's moeten eveneens beschreven zijn.

In het beste geval zou SharePointCustomer ook de ontwikkelmethodiek moeten kunnen voorschrijven met coding guidelines, ontwikkelproces, testproces, OTAP voorschriften, code review e.d. Wanneer dit niet het geval is, dan moet tenminste enig zich zijn op deze onderdelen bij de leverancier.

#### **6.1.4.3 Security**

Farm code wordt uitgevoerd in dezelfde proces ruimte als SharePoint zelf. Deze processen draaien met bevoegde accounts. Beide factoren vergroten de mogelijkheden voor schade als de code niet juist in elkaar zit of als de code onjuist gebruikt wordt. Zelfs als we code implementeren met de BIN / CAS benadering en beperkende code toegang vanwege het beveiligingsbeleid dat van toepassing is, is toch het risico van een schadelijke security exploit aanzienlijk hoger dan wanneer we Sandbox of App solutions zouden gebruiken. We dienen vooraf de code te reviewen voor beveiligingsproblemen voordat de solutions worden geïmplementeerd.

Wanneer de leverancier zegt dat het rekening heeft gehouden met alle onderdelen die risico's zouden kunnen opleveren, dan wordt er toch door SharePointCustomer gereviewed en goedgekeurd door de betreffende stakeholders. Hiervan afwijken betekent afwijken van de procedure en dat moet per geval bekeken worden.

#### **6.1.4.4 Documentatie**

Voor Farm solutions is het noodzakelijk dat de gehele solution goed is gedocumenteerd. Dit betekent dat er een HLD, DLD, FO en TD aanwezig dient te zijn. Deze documenten dienen met de betreffende stakeholders binnen SharePointCustomer gecommuniceerd te worden en ze moeten door deze stakeholders worden goedgekeurd. Wanneer de stukken niet worden goedgekeurd, dan is er sprake van een issue waar een manager met voldoende bevoegdheid een uitspraak over dient te doen. Het kan voorkomen dat een afkeuring van één van de documenten of delen ervan overruled worden. De reden van dit overrulen moet dan worden beschreven en gedocumenteerd. Dit dient in de log van het betreffende project te worden opgeslagen. Het spreekt voor zich dat het overrulen bijna per definitie bij Farm Solutions risico's met zich meebrengt. Deze risico's dienen in de risicolog van het betreffende project toegevoegd te worden.

#### **6.1.4.5 Monitoring**

SharePointCustomer heeft het liefst een goede logging van Farm solutions. Dat kan op meerdere manieren. Twee manieren dienen echter minimaal aanwezig te zijn: SharePoint logging en Windows Event logging. Farm Solution moeten gecontroleerd kunnen worden via de SharePoint log files en via de Windows Event Viewer, waar de belangrijkste loggingsregels naartoe geschreven moeten worden. Daarnaast kan het handig zijn om een individueel logging mechanisme te hebben. Wanneer dat laatste het geval is, dan dient dit met de betreffende stakeholders van SharePointCustomer gecommuniceerd te zijn. Wanneer wordt afgeweken van deze wensen van SharePointCustomer, dus wanneer er geen regels worden weggeschreven naar het SharePoint logmechanisme of de Windows Event logs, dan moet dit beschreven zijn en dient SharePointCustomer daarvan op de hoogte gesteld te worden. Er zal dan per individueel geval bekeken worden of SharePointCustomer accoord gaat met deze afwijking(en).

Zie voor verdere informatie betreffende monitoring : [tools, strategie en mogelijkheden voor het monitoren](#)

## 6.2 MANAGEMENT

Zie 4.4 : [Management](#)

## 6.3 MONITORING

Zie 4.5 : [Monitoring](#)

## 6.4 ONTWIKKELPROCES EN METHODIEK

Zie 4.6 : [Ontwikkelproces en methodiek](#)

## 6.5 TESTEN

Zie 4.6 : [Testen](#)

## 7 HET TESTPROCES VOOR DE VERSCHILLENDE SOLUTIONS

---

Zie 4.6 : [Testen](#)

Paragraaf 4.6 geeft een overzicht van Test Driven development en wat er meer komt kijken bij kwaliteit aanbrengen binnen het testproces. Hier valt in het algemeen meer over te zeggen dat geldig is voor de verschillende solutions. We hebben het dan over:

- Apps
- Sandboxed Solutions
- Farm Solutions

Alereerst dient opgemerkt te worden dat het testen van de code, functionaliteit, veiligheid en dergelijke het zwaartepunt aan de onderkant van deze lijst ligt. Farm Solutions hebben een grote vrijheid van uitvoer en de code en functionaliteit van deze solutions dienen dan ook optimaal getest, geanalyseerd en gemonitored te worden. Dat is uiteraard ook het geval voor de Apps en de Sandboxed Solutions, maar deze laatste elementen zijn intrinsiek beter beveiligd en afgeschermd dan de Farm Solutions en kunnen minder schade aanrichten als ze niet goed werken of verkeerde dingen doen. Hieruit vloeit voort dat een strategische testaanpak wellicht de juiste wijze van benadering is.

Een strategische testaanpak is vooral gericht op het onderkennen van risico's. Het is onmogelijk om SharePoint in zijn geheel te testen. Hiervoor is gewoon weg geen tijd en bovendien is het niet noodzakelijk. Dit is immers al door Microsoft gedaan. Bij het uitvoeren van een functionele test is het van belang om dat gedeelte te testen waar het zwaartepunt ligt. Wat is de kans dat iets mis gaat, en wat is het gevolg hiervan?

### 7.1.1 Stap 1: Een productrisicoanalyse

Een productrisicoanalyse (PRA) analyseert het te testen product met als doel dat de tester en andere belanghebbenden tot een gezamenlijk beeld komen over wat de meer of minder risicovolle kenmerken het product zijn. De grondigheid van testen relateert hieraan. Bij het testen van SharePoint is vooral het schatten van risico's met betrekking tot maatwerk van groot belang. Daar waar wordt afgeweken van de standaard is het risico op fouten groter. De realisatie van een maatwerkcomponent kost meer tijd en de zwaarte en grondigheid van de test wordt hierop aangepast.

Uiteraard geldt hier dat de risico's verschillen per soort applicatie en daar dient de productrisicoanalyse op in te spelen. De risico's van maatwerk zijn bij Farm Solutions wezenlijk verschillend dan bij Apps. De risicoanalyse dient dus per soort applicatie te verschillen. Vooral de productrisicoanalyse van Farm Solutions is belangrijk en deze stap kan lichter gedaan worden bij Apps en Sandboxed solutions. SharePointCustomer mag ervan uitgaan dat de leveranciers een afdoende productanalyse toepassen op hun producten. Mocht VolkerWesseld door bepaalde verschijnselen hier niet van overtuigd zijn geraakt, dan kan het zijn dat dit ter discussie komt te staan en met de betreffende stakeholders besproken zal worden.

### 7.1.2 Stap 2: Een testplan

De PRA vormt de basis voor een testplan. In dit testplan wordt op basis van het PRA de teststrategie en testaanpak uitvoerig beschreven. Daarnaast wordt er in het testplan ook aandacht besteed aan organisatorische voorwaarden. Voor een SharePoint solution kun je hierbij denken aan de beschikbaarheid van omgevingen. Een OTAP-straat is hier een belangrijk onderdeel van. Een goede test uitvoeren, heeft alleen nut wanneer dit gebeurt op een representatieve omgeving met representatieve (test)users.

Hieraan ten grondslag dient ook een ontwikkel en testmethodiek aanwezig te zijn die afdoende door processen en tools ondersteund worden. Een testplan kan alleen afdoende werkzaam zijn als er strakke procedures gelden voor zowel de tools en de processen alsmede de infrastructuur en data waarop de testen dienen plaats te vinden. Zeker met betrekking tot de OTAP straat worden er vaak consessies gedaan. SharePointCustomer is van mening dat dit grote risico's met zich meebrengt op de duurzaamheid en kwaliteit van software producten gerelateerd aan SharePoint in dit geval. Het kan voorkomen dat SharePointCustomer inzicht verlangt op dit vlak en dat zal dan gecommuniceerd worden aan de belanghebbende stakeholders. Acties die daaruit voortvloeien, kunnen niet op voorhand volledig duidelijk gemaakt worden maar zullen zeker te maken hebben met het verbeteren van onderdelen betreffende dit onderdeel.

### 7.1.3 Stap 3: De uitvoering van een systeemtest

Op basis van scenario's worden de gerealiseerde software getest en getoetst. Het toetsen van de solution gebeurt aan de hand van de testbasis. Je kunt hierbij denken de functionele eisen of een grafisch ontwerp wat is opgesteld voorafgaand aan de realisatie van de SharePoint solution. Op deze manier controleer je of de oplossing voldoet aan de kwaliteitseisen die vooraf zijn gesteld.

Bovenstaande klinkt algemeen en eenvoudig. Dat is het volstrekt niet. Zeker niet in relatie tot SharePoint. De oplossingen van SharePoint kenmerken zich door hun veelzijdigheid en heterogene samenstelling. Het vaststellen van een testbasis is dan ook niet eenvoudig. Bedenk daarbij dat we te maken hebben met verschillende soorten oplossingen. Zelfs het bepalen van de kwaliteitseisen vooraf is een niet eenvoudig proces. Dit zou samen opgesteld kunnen worden met het testplan en daar een onderdeel van kunnen zijn. Meegenomen dient te worden hier dat de functionele eisen en het grafische ontwerp passend moeten worden toegepast op de gekozen architectuur en beide dienen in de systeemtest meegenomen te worden.

De heterogene samenstelling van SharePoint Solutions zorgen er verder voor dat de systeemtests vaak behoorlijk gedifferentieerd zijn. Het verdient aanbeveling om meerdere rollen daarbij te betrekken, te denken valt aan de functionel designer, solution architect, sharepoint architect, technisch architect, infrastructuur specialist, grafisch designer, business process analyst, testmanager en lead developer. Uiteraard hoeven deze rollen niet altijd gezamenlijk aanwezig te zijn. Het is belangrijk dat deze rollen bij het proces betrokken worden.

### 7.1.4 Stap 4: Acceptatietest

Na het uitvoeren van de systeemtest en het (eventueel) oplossen van bugs kan de gerealiseerde software worden opgeleverd aan de opdrachtgever. Wanneer de ontwikkeling van software wordt uitbesteedt, dan vindt doorgaans direct na oplevering een acceptatietest plaats. Hiermee worden de activiteiten van de opdrachtgever bedoeld die erop zijn gericht om in een overeengekomen testperiode

het geleverde product systematisch te testen en beoordelen. Het doel van de acceptatietest is vast te stellen dat de software voldoet aan de eisen en wensen en dat de software geschikt is voor bedrijfsmatige ingebruikname.

Het is van belang dat de acceptatietest niet als een waterval gedaan wordt. Beter is het om in increments te werken en de Scrum methodiek dwingt dit ook af. Wanneer de acceptatietest niet overeenkomt met de geleverde software of als de acceptatietests niet de juiste verwachtingen weet te scheppen, dan is er sprake van een discrepantie tussen wat SharePointCustomer wil en wat de leverancier biedt. Dat hoeft niet direct tot problemen te leiden, maar is altijd de aanleiding om het proces eens aandachtig te bekijken. Bijstelling zal dan waarschijnlijk het gevolg zijn. Als de acceptatietests echter blijvend onvoldoende de verwachtingen van SharePointCustomer weet te bekoren, dan zal dieper op het proces ingezoomd worden met de betreffende stakeholders. Bijsturing op meerdere onderdelen is dan onvermijdelijk. De algehele kwaliteit van het testproces heeft dan de aandacht.

## 8 BIBLIOGRAPHY

---

- Bruggeman, M. (2012, 9 3). *SharePoint 2013: What To Do? Farm Solutions vs Sandbox vs App*. Retrieved from Technet SharePoint 2013: <http://social.technet.microsoft.com/wiki/contents/articles/13373.sharepoint-2013-what-to-do-farm-solution-vs-sandbox-vs-app.aspx>
- Bruggeman, M. (2013, 8 27). *SharePoint 2013 Best Practices*. Retrieved from Microsoft Technet Social: <http://social.technet.microsoft.com/wiki/contents/articles/12438.sharepoint-2013-best-practices.aspx>
- Damgard, M. (2013, 2 4). *Planning Deployment of Farm Solutions for SharePoint 2013*. Retrieved from Blogs Technet: <http://blogs.technet.com/b/mspfe/archive/2013/02/04/planning-deployment-of-farm-solutions-for-sharepoint-2013.aspx>
- Microsoft. (2006, 5). *Guidelines for Test driven Developmenr*. Retrieved from Microsoft Developer Network: [http://msdn.microsoft.com/en-us/library/aa730844\(v=vs.80\).aspx](http://msdn.microsoft.com/en-us/library/aa730844(v=vs.80).aspx)
- Microsoft. (2010, 5 12). *Sandboxed solutions overview (SharePoint Server 2013)*. Retrieved from Technet SharePoint 2013: [http://technet.microsoft.com/en-us/library/ee721992\(v=office.14\).aspx](http://technet.microsoft.com/en-us/library/ee721992(v=office.14).aspx)
- Microsoft. (2011). *Best Practices for Developing Sandboxed Solutions in SharePoint 2010*. Retrieved from Microsoft Developers Network: [http://msdn.microsoft.com/en-us/library/gg615455\(v=office.14\).aspx](http://msdn.microsoft.com/en-us/library/gg615455(v=office.14).aspx)
- Microsoft. (2012, 7 16). *Build Apps for SharePoint*. Retrieved from Developer Network: <http://msdn.microsoft.com/en-us/library/jj163230.aspx>
- Microsoft. (2012, 7 16). *Farm Solutions*. Retrieved from Microsoft Developers Network: <http://msdn.microsoft.com/en-us/library/ff798425.aspx>
- Microsoft. (2012, 7 16). *Monitor apps for SharePoint for SharePoint Server 2013*. Retrieved from Technet SharePoint: <http://technet.microsoft.com/en-us/library/jj219567.aspx>
- Microsoft. (2012, 7 16). *Monitoring SharePoint 2013*. Retrieved from Technet Microsoft SharePoint: <http://technet.microsoft.com/en-us/library/jj219701.aspx>
- Microsoft. (2012, 7 16). *Overview of Apps for SharePoint*. Retrieved from Technet SharePoint 2013: <http://technet.microsoft.com/en-us/library/fp161230.aspx>
- Microsoft. (2012, 10 16). *Overview of Monitoring in SharePoint 2013*. Retrieved from Technet SharePoint 2013: <http://technet.microsoft.com/en-us/library/ee748636.aspx>
- Microsoft. (2012, 7 16). *Plan app permissions management in SharePoint 2013*. Retrieved from Technet SharePoint: <http://technet.microsoft.com/en-us/library/jj219576.aspx>
- Microsoft. (2012, 7 16). *Publish Apps for SharePoint*. Retrieved from Developer Network: <http://msdn.microsoft.com/en-us/library/jj164070.aspx>

Microsoft. (2012, 10 16). *View Diagnostic Logs in SharePoint 2013*. Retrieved from Technet SharePoint:  
<http://technet.microsoft.com/en-us/library/ff463595.aspx>

Microsoft. (2013, 4 30). *Configure an environment for apps for SharePoint*. Retrieved from Technet SharePoint: <http://technet.microsoft.com/en-us/library/fp161236.aspx>

Microsoft. (2013, 2 26). *Plan Apps for SharePoint 2013*. Retrieved from Technet SharePoint:  
<http://technet.microsoft.com/en-us/library/fp161237.aspx>

Microsoft. (2013). *SPUtility Properties*. Retrieved from Microsoft Developer Network:  
<http://msdn.microsoft.com/en-us/library/ms464967.aspx>

Vosters, K. (2013, 5 11). *The Subscription Settings service and corresponding application and proxy needs to be running in order to make changes to these settings*. Retrieved from MSDN Blogs:  
<http://blogs.msdn.com/b/kvosters/archive/2013/04/11/the-subscription-settings-service-and-corresponding-application-and-proxy-needs-to-be-running-in-order-to-make-changes-to-these-settings.aspx>

Wikipedia. (2013). *Test-Driven Development*. Retrieved from Wikipedia TDD:  
[http://en.wikipedia.org/wiki/Test-driven\\_development](http://en.wikipedia.org/wiki/Test-driven_development)


## 9 APPENDIX A.2.

Figure 1 : Farm vs Apps vs Sandbox

	<b>Sandbox</b>	<b>Apps</b>	<b>Farm</b>
When to use	Deprecated. Therefore, it's unadvisable to build new sandboxed solutions.	Best practice. Create apps whenever you can.	Create farm solutions when you can't do it in an app. See <a href="http://www.learningsharepoint.com/2012/07/20/sharepoint-2013-apps-vs-farm-solutions/">http://www.learningsharepoint.com/2012/07/20/sharepoint-2013-apps-vs-farm-solutions/</a> for more info.
Server-side code	Runs under a strict CAS policy and is limited in what it can do.	No SharePoint server-code. When apps are hosted in an isolated SharePoint site, no server-code whatsoever is allowed.	Can run full trust code. (Custom CAS policies are not supported in SharePoint 2013. All farm solution code runs in full trust even if it is not deployed to the GAC. Any custom CAS policies are ignored.)
Resource throttling	Run under an advanced resource management system that allows resource point allocation and automatic shutdown for troublesome solutions.	Apps run isolated from a SharePoint farm, but can have an indirect impact by leveraging the client object model.	Can impact SharePoint server-farm stability directly.

	<b>SandBox</b>	<b>Apps</b>	<b>Farm</b>
Runs cross-domain	No, and there's no need to since code runs within the SharePoint farm.	Yes, which provides a very interesting way to distribute server loads.	No, and there's no need to since code runs within the SharePoint farm.
Efficiency/Performance	Runs on the server farm, but in a dedicated isolated process. The sandbox architecture provides overhead.	Apps hosted on separate app servers (even cross-domain) or in the cloud may cause considerable overhead.	Very efficient.



	Sandbox	Apps	Farm
Safety	Very safe.	Apps rely on OAuth 2.0. The OAuth 2.0 standard is surrounded by some controversy (for example, check out what OAuth lead author Eran Hammer has to say about it here: <a href="http://hueniverse.com/2012/07/oauth-2-0-and-the-road-to-hell/">http://hueniverse.com/2012/07/oauth-2-0-and-the-road-to-hell/</a>  ). In fact, some SharePoint experts have gone on the record stating that security for Apps will become a big problem. We'll just have to wait and see how this turns out.	Can be very safe, but this requires additional testing, validation and potential monitoring.
Should IT pros worry over it?	Due to the limited CAS permissions and resource throttling system, IT pros don't have to worry.	Apps are able to do a lot via the client OM. There are some uncertainties concerning the safety of an App running on a page with other Apps. For now, this seems to be the most worry-able option, but we'll have to see how this plays out.	Definitely. This type of solutions run on the SharePoint farm itself and therefore can have a profound impact.
Manageability	Easy to manage within the SharePoint farm.	Can be managed on a dedicated environment without SharePoint. Dedicated app admins can take care of this.	Easy to manage within the SharePoint farm.
Cloud support	Yes	Yes, also support for App MarketPlace.	No, on-premises only.

## 10 APPENDIX A.4.4.1

Figure 1:Keuzes voor Governance betreffende Apps voor SharePoint

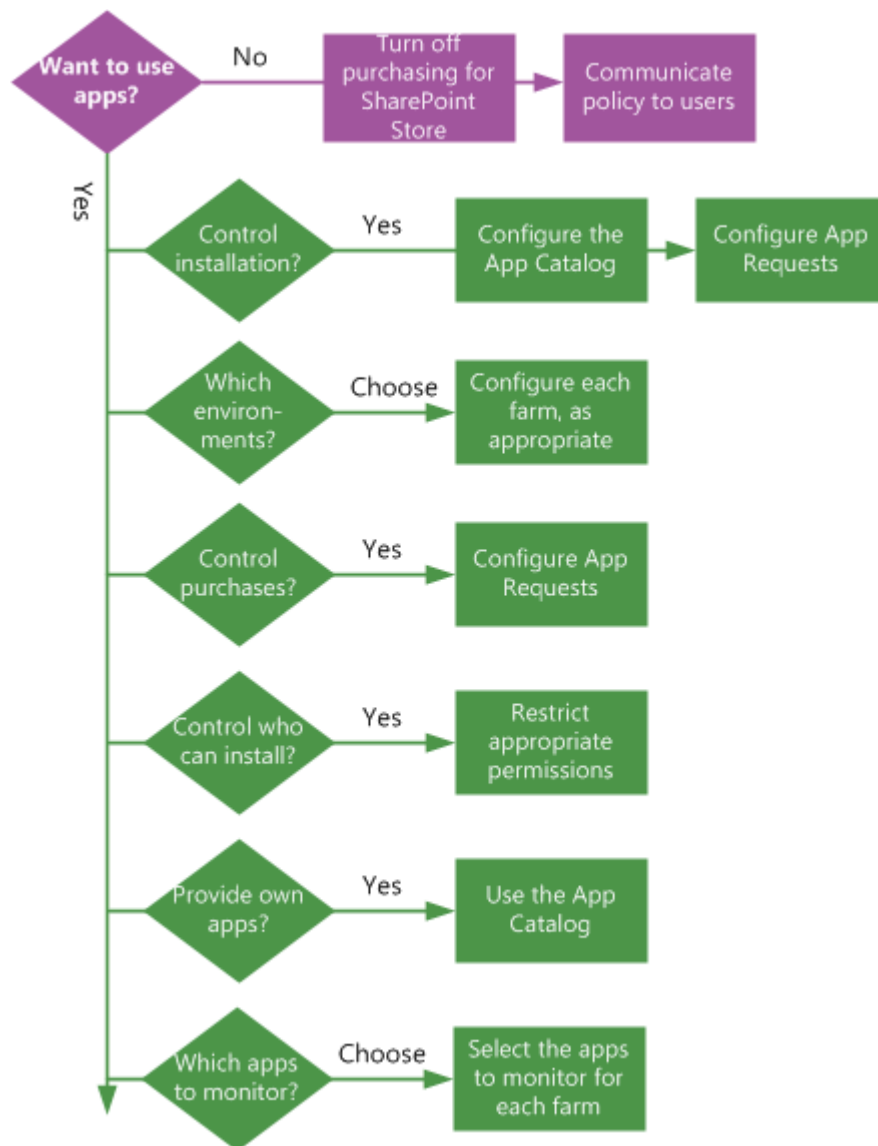
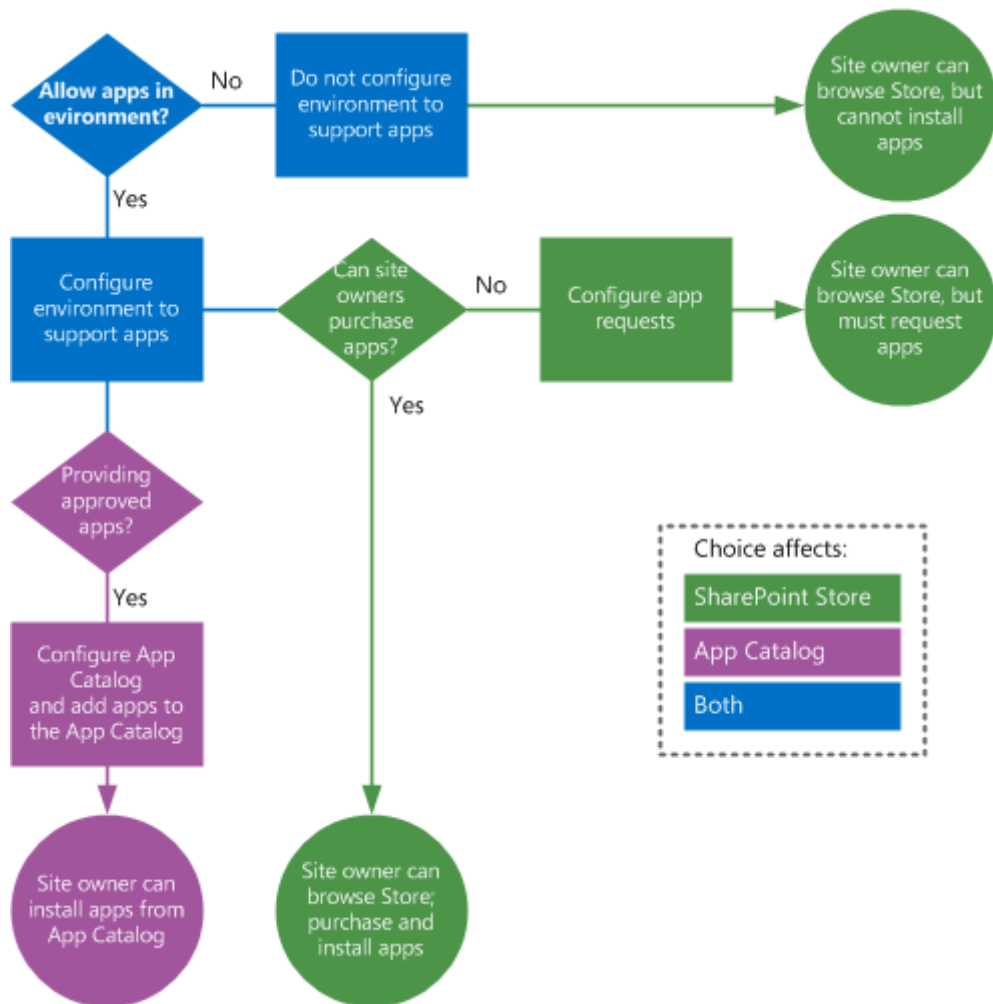


Figure 2: Beslissingen voor de aankoop van Apps voor SharePoint



## 11 APPENDIX A.3.1

*Figure 1 : Summary of Monitoring tools*

Summary of monitoring tools

Tool	Optional or built-in	Skill level required	Pros	Cons	Resources
SharePoint Health Analyzer	Built-in	Basic	<ul style="list-style-type: none"> <li>Gives step-by-step instructions for resolving a problem.</li> <li>Customizable. You can disable some rules if you don't need them.</li> </ul>	<ul style="list-style-type: none"> <li>Does not cover all possible problems.</li> <li>A rule is triggered only after a problem has already happened.</li> </ul>	Configure SharePoint Health Analyzer rules in SharePoint 2013 View and resolve SharePoint Health Analyzer alerts in SharePoint 2013 How to: Create a Health Rule ( <a href="http://go.microsoft.com/fwlink/p/?LinkId=247474">http://go.microsoft.com/fwlink/p/?LinkId=247474</a> ). SharePoint Health Analyzer rules reference (SharePoint 2013)
Timer jobs	Built-in	Advanced: You have to consider the implications of enabling or disabling a timer job and changing schedules.	<ul style="list-style-type: none"> <li>Wide range of monitoring items.</li> <li>Easy to reschedule.</li> <li>Customizable. You can create new timer jobs to meet your specific requirements.</li> </ul>	Can affect system performance and conflict with one another.	Configure SharePoint Health Analyzer timer jobs in SharePoint 2013 Timer job reference (SharePoint 2013) Create a custom timer job ( <a href="http://go.microsoft.com/fwlink/p/?LinkId=253616">http://go.microsoft.com/fwlink/p/?LinkId=253616</a> )
Reporting	Built-in	Viewing skill level: Basic	Flexible. Lets you configure the severity of events to log, enable event log flood protection, and configure trace logs.	<ul style="list-style-type: none"> <li>Only shows administrative and health statistics, without any suggested solutions.</li> <li>You have to interpret the logging data.</li> <li>Can affect performance and disk usage.</li> </ul>	View reports and logs in SharePoint 2013
Windows PowerShell	Built-in	Advanced: you have to know what to look for, and you have to run Windows PowerShell commands.	Filters data, displays it in various ways, and outputs data to a grid with which you can filter, sort, group, and export the data to Excel 2013.	You have to know which Windows PowerShell commands to run.	View diagnostic logs in SharePoint 2013 Windows PowerShell ( <a href="http://go.microsoft.com/fwlink/p/?LinkId=206775">http://go.microsoft.com/fwlink/p/?LinkId=206775</a> )
System Center 2012 - Operations Manager with System Center Management Pack for SharePoint Server 2013	Optional but recommended	Advanced: you need to know what to look for and how to interpret data.	The tool can detect, diagnose, and alert you about software and hardware incidents, and refer you to TechNet knowledge articles. It helps you to do more monitoring with fewer people by monitoring many key scenarios.	Requires additional servers to deploy.	Operations Manager ( <a href="http://go.microsoft.com/fwlink/p/?LinkId=226376">http://go.microsoft.com/fwlink/p/?LinkId=226376</a> ) Microsoft SharePoint 2010 Products Management Pack for System Center Operations Manager 2007 ( <a href="http://go.microsoft.com/fwlink/p/?LinkId=203252">http://go.microsoft.com/fwlink/p/?LinkId=203252</a> )
Event Viewer	Built-in	Basic	You can view events from multiple event logs, save useful event filters as custom views that can be reused, schedule a task to run in response to an event, and create and manage event subscriptions.	Does not suggest resolutions.	Event Viewer ( <a href="http://go.microsoft.com/fwlink/p/?LinkId=253618">http://go.microsoft.com/fwlink/p/?LinkId=253618</a> )
SharePoint Developer Dashboard	Built-in	Advanced: you have to know what to look for and how to interpret data.	Easy to analyze performance of SharePoint pages.	Limited to monitoring performance of SharePoint pages.	SharePoint Developer Dashboard ( <a href="http://go.microsoft.com/fwlink/p/?LinkId=199580">http://go.microsoft.com/fwlink/p/?LinkId=199580</a> )
Windows Management Instrumentation (WMI)	Built-in	Advanced: you have to know what to look for and how to write WMI scripts for the managed objects.	Can monitor, track, and control system events that are related to software applications, hardware components, and networks.	<ul style="list-style-type: none"> <li>You have to identify which managed objects to monitor.</li> <li>You have to write WMI scripts.</li> </ul>	SharePoint 2010 SDK ( <a href="http://go.microsoft.com/fwlink/p/?LinkId=166675">http://go.microsoft.com/fwlink/p/?LinkId=166675</a> ) WMI Overview ( <a href="http://go.microsoft.com/fwlink/p/?LinkId=188652">http://go.microsoft.com/fwlink/p/?LinkId=188652</a> ) FAST Search: WMI classes ( <a href="http://go.microsoft.com/fwlink/p/?LinkId=203240">http://go.microsoft.com/fwlink/p/?LinkId=203240</a> )
SQL Server Reporting Services	Optional but recommended.	Advanced: you have to design and manage reports.	Comprehensive platform to create and manage reports.	Advanced skills required to create and develop solutions. You have to know PerformancePoint Dashboard Designer and Visual Studio.	SQL Server Reporting Services ( <a href="http://go.microsoft.com/fwlink/p/?LinkId=253387">http://go.microsoft.com/fwlink/p/?LinkId=253387</a> )

## 12 APPENDIX A.3.2

Figure 1 : Scenarios and monitoring tools

Scenarios and monitoring tools

Tool\Scenario	Health	Search	Databases	Performance	Workflows	Virtual environments	Business Data Connectivity	Business Intelligence	Access Services	Farms	Servers	Service applications	Web applications
SharePoint Health Analyzer	√	√	√	√	X	X	√	√	√	√	√	√	X
Timer jobs	√	√	√	√	√	X	X	√	X	X	X	√	√
Reporting	√	√	√	√	√	X	√	√	√	√	√	√	X
Windows PowerShell	√	√	√	√	√	X	√	√	√	√	√	√	√
System Center 2012 - Operations Manager with System Center Management Pack for SharePoint Server 2013	√	√	√	√	X	√	√	√	√	√	√	√	X
Event Viewer	√	√	√	√	√	X	√	√	√	√	√	√	√
SharePoint Developer Dashboard	X	X	X	√*	X	X	X	X	X	X	X	X	X
Windows Management Instrumentation (WMI)	√	√	√	√	X	√**	X	√	X	X	√	X	X
SQL Server Reporting Services	√	√	√	√	√	X	√	√	√	√	√	√	√

**Note:**

\*: SharePoint Developer Dashboard only monitors performance of web pages.

\*\* : The WMI interfaces can manage the Hyper-V services.

## 13 APPENDIX A.3.4

---

Figure 1 : Monitoring personeel nodig

### Monitoring personnel required

Number of servers	Personnel required	Skill level required
1~4	1	Basic
4~10	1~2	Advanced
10~40	3~4	Basic and Advanced
40~100	5+	Basic and Advanced
100+	10+	Basic and Advanced

## 14 APPENDIX A.3.5

---

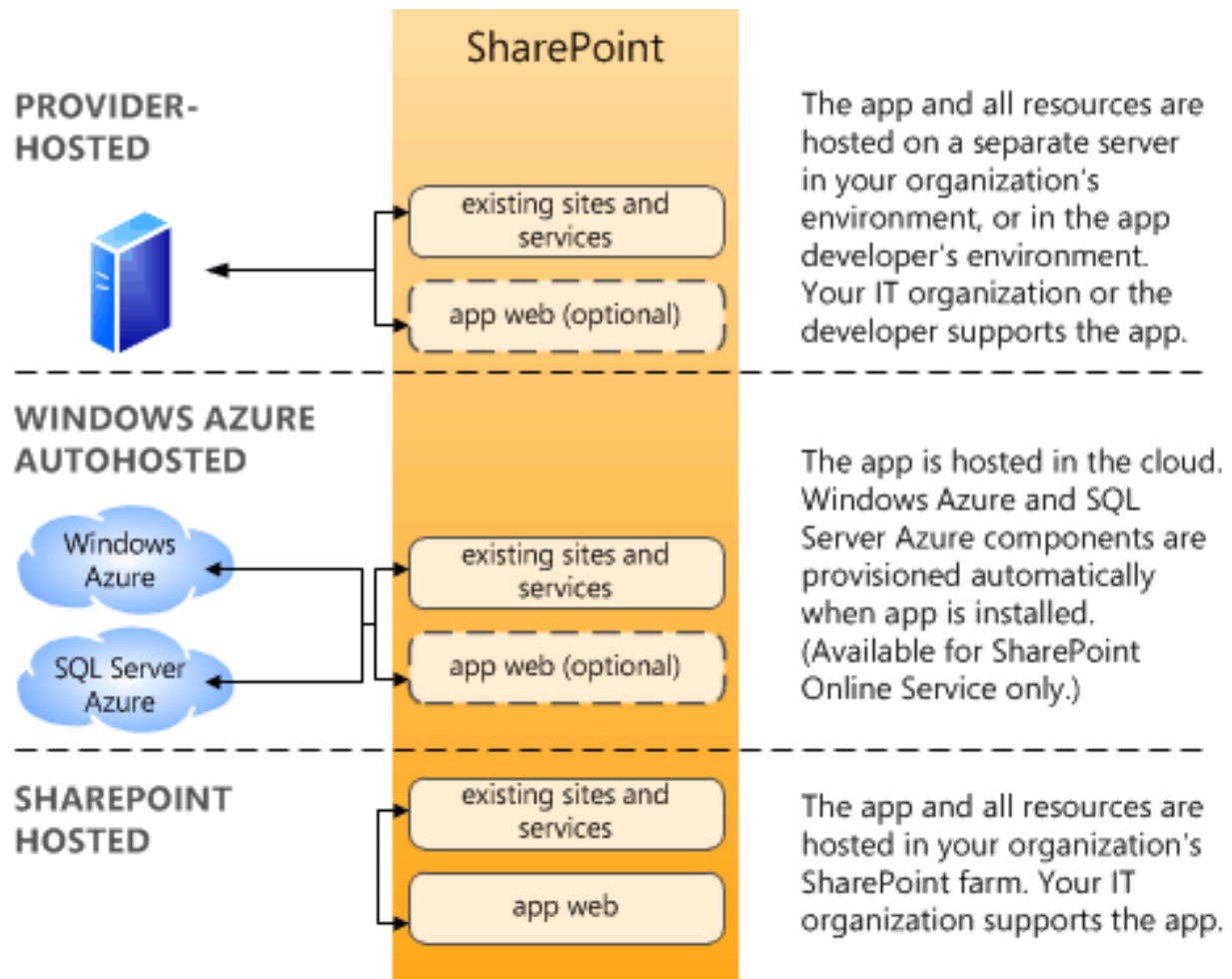
*Figure 1 : Voorgestelde items in een response plan*

### Suggested items in a response plan

Item	Description
Alert/Event/Problem	The verbatim message, the verbatim event, or description of the problem.
Affected services/applications	Services or applications that will be affected by the problem.
Symptom	Symptom of the problem.
Severity	Severity of the problem. Problems with high severity must have high priority.
Problem must be resolved in (minutes or hours)	Acceptable lapse time of service.
Possible causes	Possible causes of the problem.
Resolutions	Resolutions of the problem.
Contacts	People who should be contacted when this problem occurs.
Escalation	People or teams that should be contacted if the resolutions did not successfully resolve the problem.
Related resources	Any resources that may help resolve the problem, such as articles on the TechNet.
Note	Any issues that you want to highlight.

## 15 APPENDIX A.4.1

Figure 1 : Type of Apps





## 16 APPENDIX A.4.2

Figuur 1: Overzicht van het configureren van een omgeving voor Apps voor SharePoint

### Before you begin

Buy a domain for apps from a domain provider **ContosoApps.com**

---

### On your DNS Server



- ① Create a forward lookup zone for apps **ContosoApps.com**
- ② Create a CNAME alias from the app domain to the SharePoint domain **\*.ContosoApps.com -> SharePoint.Contoso.com**

### For SSL (https:// URLs)



- ③ Create a wildcard SSL certificate for the new app domain **https://\*.ContosoApps.com**

### On your SharePoint Servers



- ④ Configure the Subscription Settings service application by using Windows PowerShell.



- ⑤ Configure the App Management service application (Central Administration or Windows PowerShell).
- ⑥ Configure the App URLs in Central Administration **Domain: ContosoApps.com**  
**Prefix: Apps**

## 17 APPENDIX A.4.4.2.3.

---

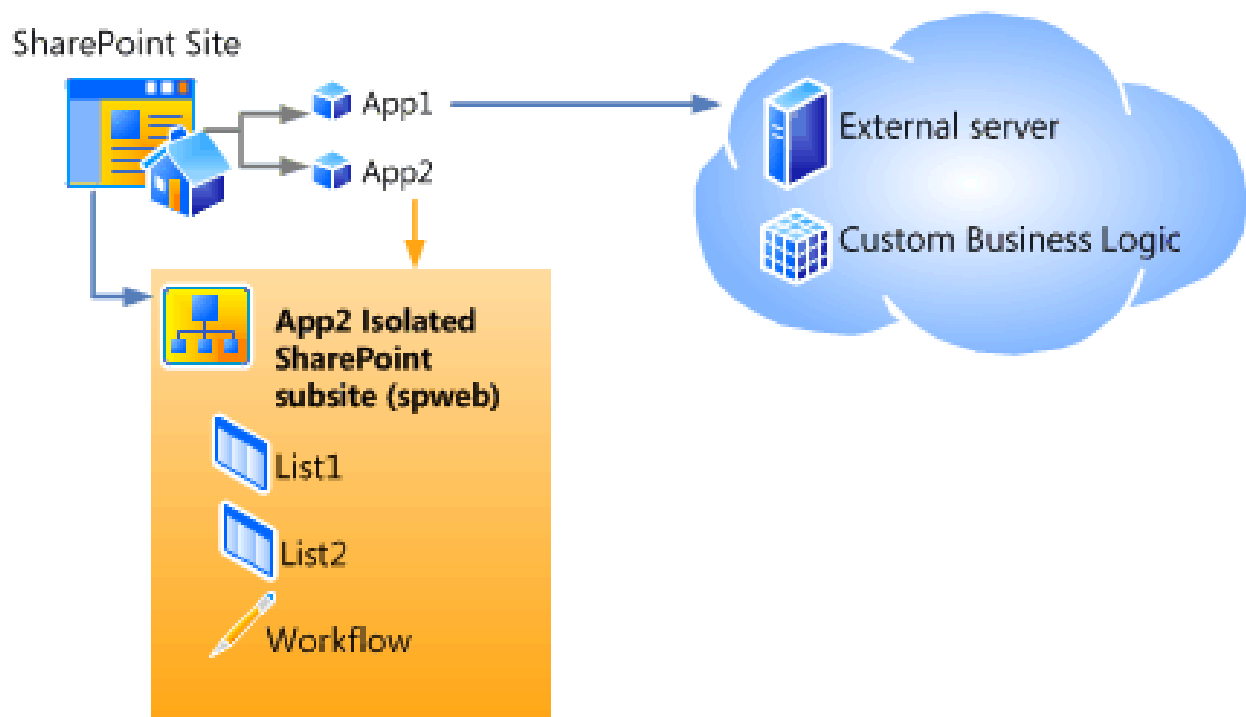
*Figure 2 : App permissions request scopes*

Scope	URI
SPSite	http://sharepoint/content/sitecollection/
SPWeb	http://sharepoint/content/sitecollection/web
SPList	http://sharepoint/content/sitecollection/web/list
Tenancy	http://<sharepointserver>/<content>/<tenant>/

## 18 APPENDIX A.4.3.1.

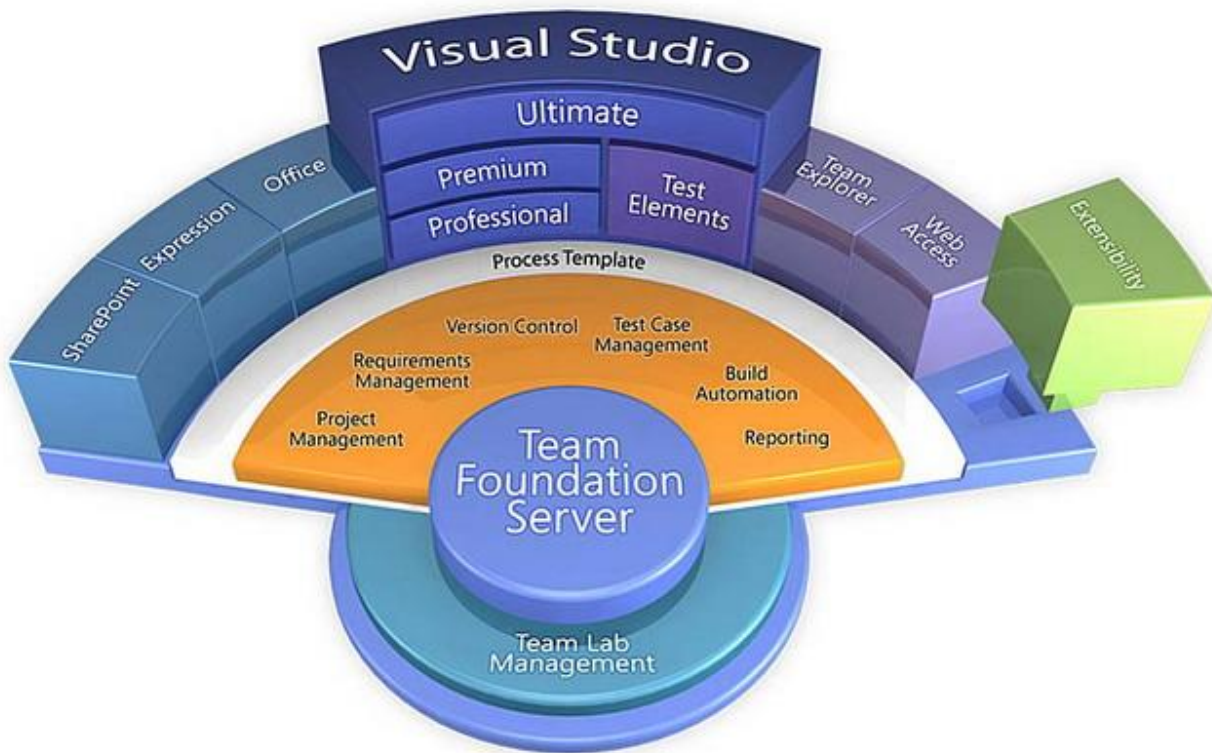
---

*Figure 1 : Publisieren naar een App Catalogus*

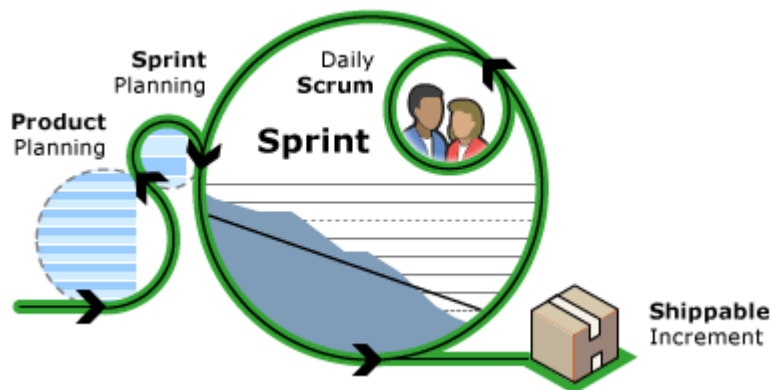


## 19 APPENDIX A.4.6.

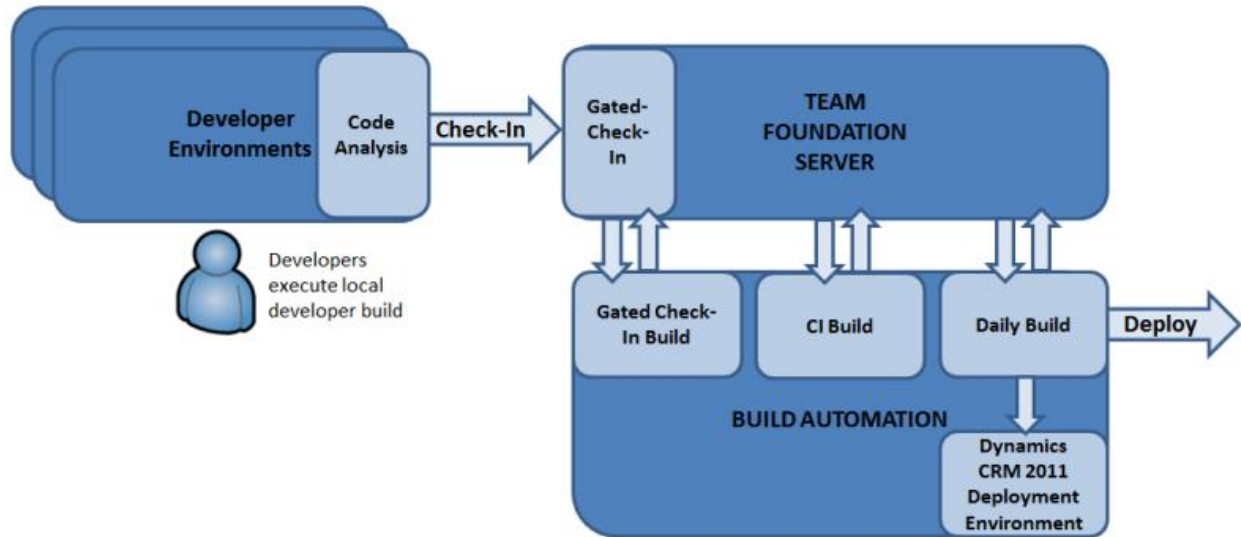
*Figuur 1: ALM software elementen*



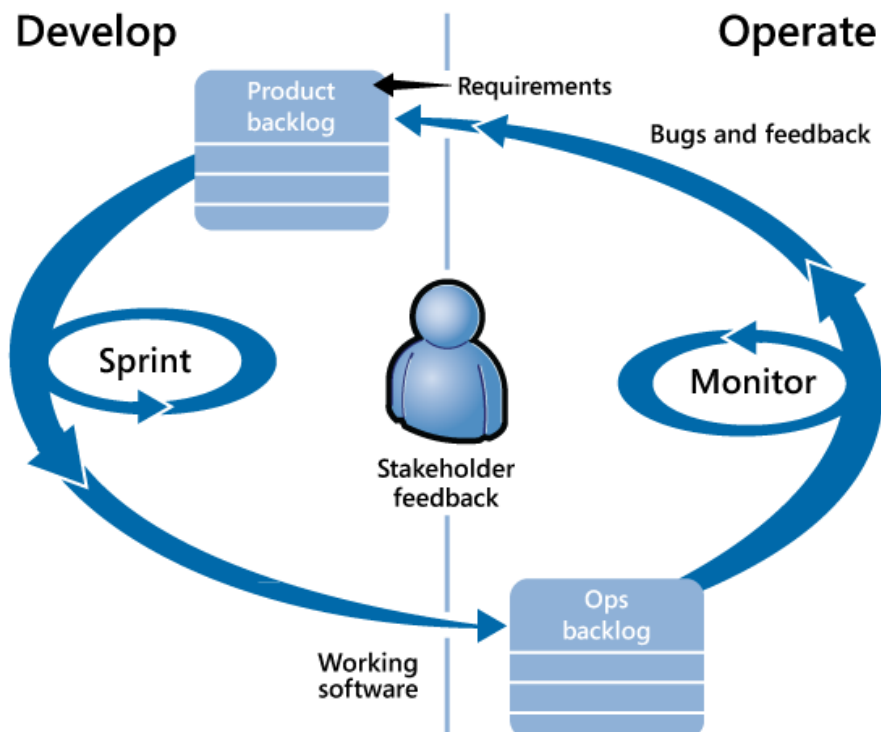
*Figuur 2: Scrum process voor software development*



Figuur 3 : Ontwikkelaar check in regime

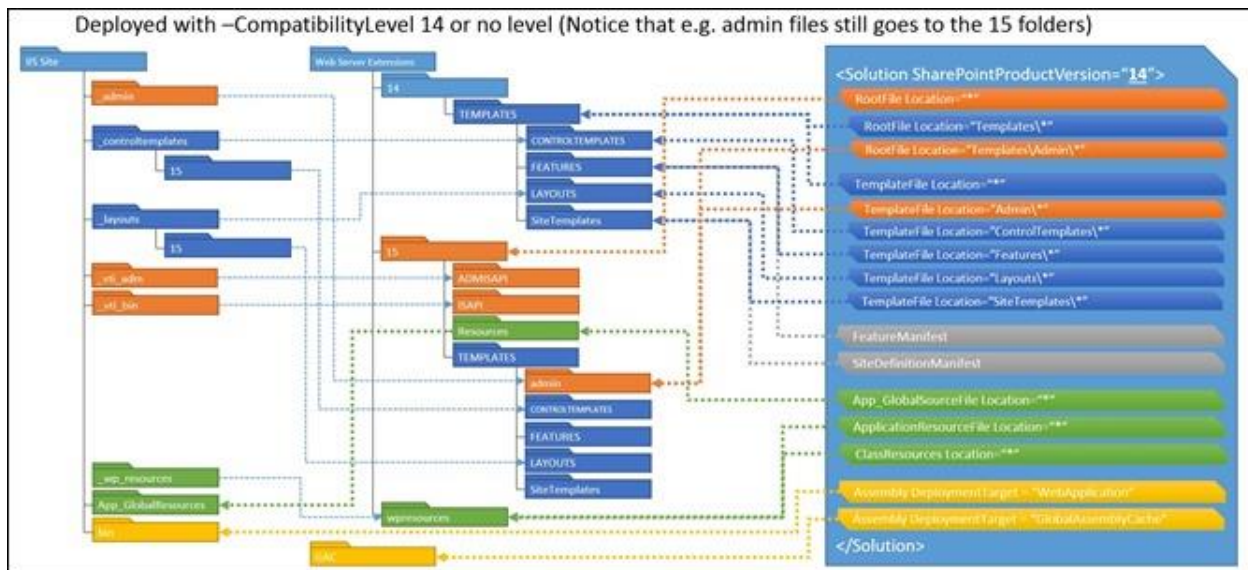


Figuur 4 : Overview ontwikkelproces hoog niveau



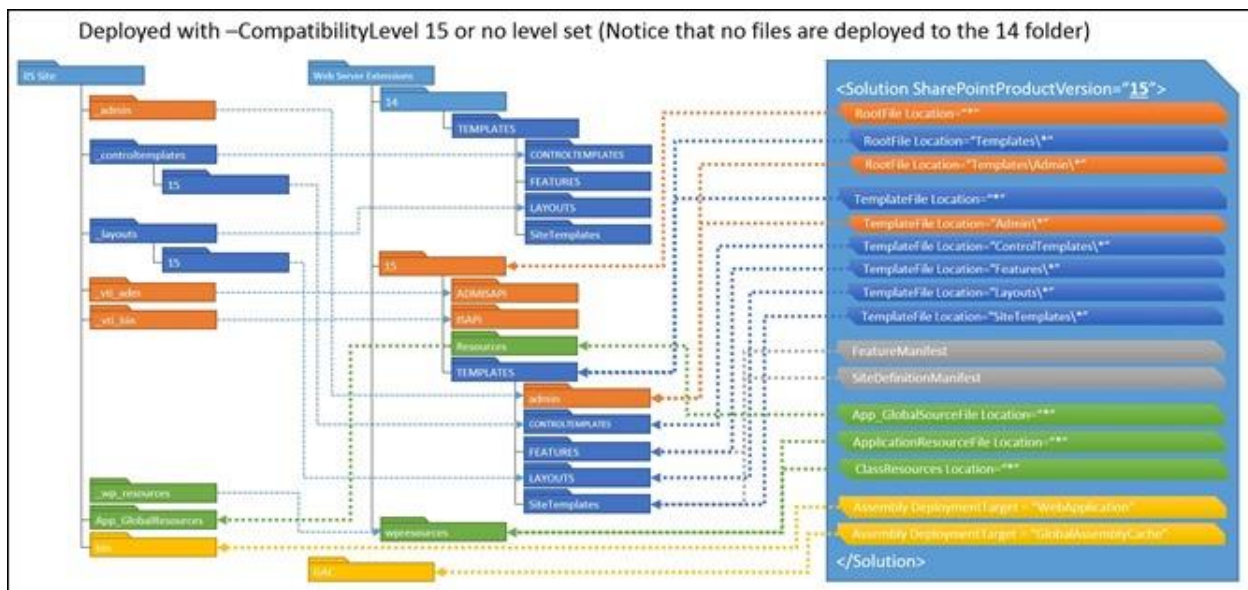
## 20 APPENDIX A.5.1.1.

Figure 1: deployed with compatibility level 14 or no level



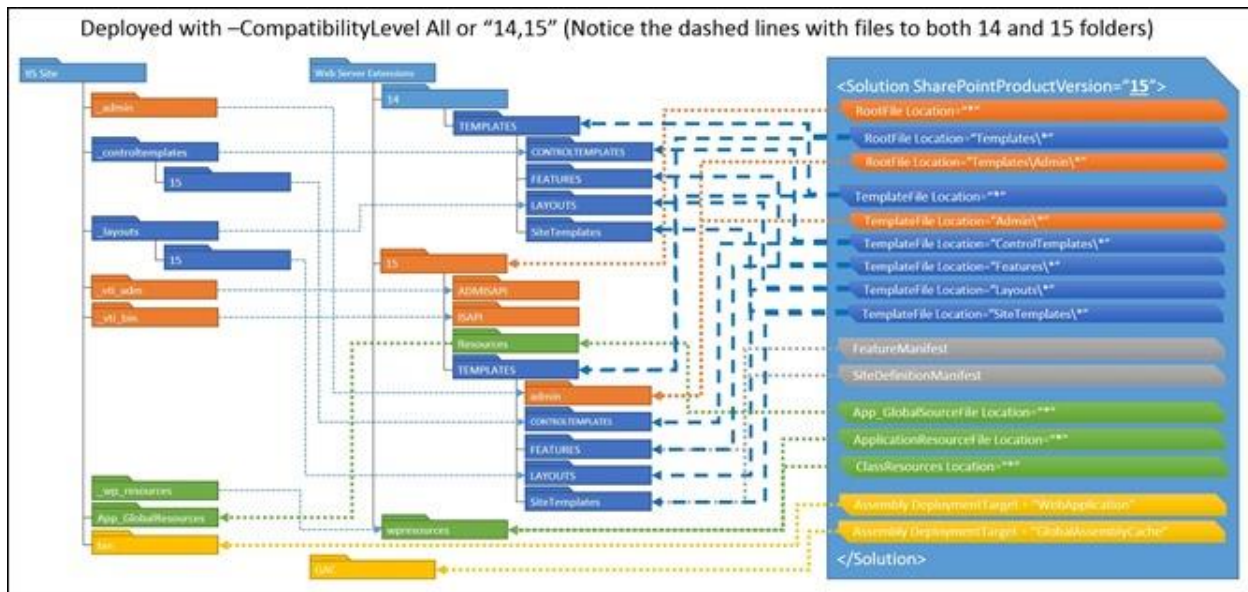
[http://blogs.technet.com/cfs-filessystemfile.ashx/\\_key/communityserver-blogs-components-weblogfiles/00-00-00-92-37-metablogapi/0246.CompatibilityLevelOld\\_5F00\\_1AB462C2.jpg](http://blogs.technet.com/cfs-filessystemfile.ashx/_key/communityserver-blogs-components-weblogfiles/00-00-00-92-37-metablogapi/0246.CompatibilityLevelOld_5F00_1AB462C2.jpg)

Figure 2: Deployed with compatibility level 15 or no level set



[http://blogs.technet.com/cfs-filessystemfile.ashx/\\_key/communityserver-blogs-components-weblogfiles/00-00-00-92-37-metablogapi/5633.CompatibilityLevelNew\\_5F00\\_5A80D9F8.jpg](http://blogs.technet.com/cfs-filessystemfile.ashx/_key/communityserver-blogs-components-weblogfiles/00-00-00-92-37-metablogapi/5633.CompatibilityLevelNew_5F00_5A80D9F8.jpg)

Figure3 : Deployed with compatibility level All or "14,15"



[http://blogs.technet.com/cfs-filessystemfile.ashx/\\_key/communityserver-blogs-components-weblogfiles/00-00-00-92-37-metablogapi/2068.CompatibilityLevelAll\\_5F00\\_499C3608.jpg](http://blogs.technet.com/cfs-filessystemfile.ashx/_key/communityserver-blogs-components-weblogfiles/00-00-00-92-37-metablogapi/2068.CompatibilityLevelAll_5F00_499C3608.jpg)