



SHARED SERVICE CENTER



# IMPLEMENTATIEPLAN

Internet of Things: GitHub

Shared Service Center  
Scalda Vlissingen  
Edisonweg 4a

Projectleider	: Cihan Toksöz
Projectleden	: Maikel Evegroen
Datum	: 20 november 2019
Klas	: ICO43C



## Inhoud

<b>Achtergrond</b>	<b>3</b>
<b>Projectdoel</b>	<b>3</b>
<b>Gebruikers en beheerders</b>	<b>3</b>
Gebruikers	3
Beheerders	3
<b>Technische installatie</b>	<b>4</b>
<b>Testen</b>	<b>5</b>
<b>Aanpak/werkwijze</b>	<b>6</b>
Implementatie	6
<b>Planning en mogelijke risico's</b>	<b>6</b>



## Achtergrond

De vraag van de opdrachtgever is geweest om een GitHub omgeving te ontwikkelen die wij in het Shared Service Center kunnen gaan gebruiken. Dit plan vormt een onderdeel van het onderzoeks- en implementatieplan. Het moest beschikken over toegangscontroles en verschillende samenwerkingsfuncties, zoals issue tracker, een forum voor het aanvragen van functies, takenlijsten en wiki's voor project.

## Projectdoel

Het doel van dit project is om de broncodes van de applicaties die wij gebruiken binnen in de projecten op te slaan op de server van GitHub. Het versiebeheersysteem is als implementatie ontwikkeld voor de kennisgroep IoT die als doel heeft om files als README, CONFIG files of andere bestanden in mappenstructuur te kunnen waarborgen onder de server van GitHub. Bij mogelijke uitbreiding moet het gebruikt kunnen worden in heel het Shared Service Center.

## Gebruikers en beheerders

### Gebruikers

De omgeving zal uiteindelijk overgedragen worden aan de kennisgroep IoT. De kennisgroep IoT is verdeeld onder 4 subgroepen: Home Assistant, Node-RED, databeheer en het praktijklab. GitHub valt onder de groep van de databeheer. De kennisgroep ontvangt samen met het product een installatie- en gebruikershandleiding, wat het zal leiden tot gebruik van de GitHub omgeving.

### Beheerders

Binnenin de kennisgroep zullen er leden zijn van databeheer, die de GitHub zullen gaan beheren. Het beheer van GitHub zal in het begin een lastig gebeuren zijn. Naarmate de tijd, na ermee geoefend en rondgesnuffeld te hebben zal het wat soepeler werken. De beheerders zullen de rechten krijgen en uiteindelijk hebben om de afgesproken benodigde README, CONFIG files of andere bestanden in mappenstructuur te kunnen plaatsen. Het oriënteren en beheren van de GitHub zullen voor mogelijke ontwikkelingen zorgen binnenin de groep en omgeving.



## Technische installatie

GitLab	
Besturingssysteem	Linux
Programeertaal	Ruby (MRI) 2.6
Opslagruimte	Benodigde ruimte op de harde schijf hangt grotendeels af van de grootte van de repo's die in GitLab opgeslagen moeten worden
CPU	1 core ondersteunt maximaal 100 gebruikers 2 cores minimum aantal cores en ondersteunt maximaal 100 gebruikers 4 cores ondersteunt maximaal 500 gebruikers 8 cores ondersteunen maximaal 1000 gebruikers 32 cores ondersteunen maximaal 5000 gebruikers
Geheugen	4 GB RAM + 4GB swap ondersteunt maximaal 100 gebruikers 8 GB RAM ondersteunt maximaal 100 gebruikers 16 GB RAM ondersteunt maximaal 500 gebruikers 32 GB RAM ondersteunt maximaal 1000 gebruikers 128 GB RAM ondersteunt tot 5000 gebruikers
Database	PostgreSQL
Mac (macOS 10.10 of hoger)	Windows (Microsoft Windows 7 of hoger)
Visit GitHub Desktop download page Choose Download for Mac In your computer's Downloads folder double-click the GitHub Desktop zip file After the file has been unzipped, double-click GitHub Dekstop	Visit GitHub Desktop download page Choose Download for Windows In your computer;s Downloads folder double-click GitHub Desktop In the pop-up window, click install  After the program has been installed, click Run
Waarschuwing: je moet een 64-bits besturingssysteem hebben om GitHub Desktop uit te voeren.	



## Testen

In de afgelopen fase aan het einde van de eerste kerntaak, was ervoor de testomgeving een testplan opgesteld, uitgevoerd en goedgekeurd. Tijdens het beoefenen doormiddel van de cursus (codecademy: Learn GIT) kwamen wij als projectgroep op het idee om een verbetering te kunnen maken voor de koppeling van de bestandsoverdracht. De bestaande bestanden die nog geupload worden op de Sharepoint en de kennisbank, zullen gekoppeld worden aan een GitHub repository. De koppeling zal ervoor zorgen dat er bij een Git command gelijk de laatste wijzigingen van bestanden doorgevoerd zullen worden naar de GitHub omgeving, zodat het up to date blijft. Daarnaast komen er een aantal testen bij die specifiek zijn voor de implementatie van het eindproduct en willen wij als groep alle functies van de GitHub omgeving willen testen voor het zekere, dat alles functioneert naar behoren van de opdrachtgever, coaches en natuurlijk alle andere eindgebruikers.

De testen zullen uitgevoerd worden in de periode van het implementeren. Het implementeren zal de tijd hebben tot aan het einde van review 4. Deze worden hoogstwaarschijnlijk uitgevoerd door zowel de projectleider Cihan Toksöz als de projectlid Maikel Evegroen in het Shared Service Center.

Testnummer	Test	Uitvoering	Resultaat
1	Alle documenten en projecten zijn terug te vinden die aangemaakt zijn	10 december 2019	✓
2	Issues kunnen gemeld worden	10 december 2019	✓
3	Er kunne contributors (leden) toegevoegd worden	10 december 2019	✓
4	Er kan een repository gemaakt worden	10 december 2019	✓
5	De projecten zijn local bereikbaar	10 december 2019	✓
6	Kan er gemakkelijk een pull request gemaakt worden?	10 december 2019	✓
7	Het verwerken van codes lukt soepel	10 december 2019	✓
8	Het gebruiken van een workflow is beschikbaar	10 december 2019	✓
9	De projecten kunnen privé gezet worden	10 december 2019	✓
Genoodzaakte testen			
Sharepoint synchroniseren met lokale schijf (C:)			
Systeemmap Scalda (SSC KG IOT – Documenten) koppelen aan GitHub			
Alle 1220 bestanden, 371 mappen van Systeemmap Scalda (SSC KG IOT – Documenten) uploaden naar GitHub door te pushen			
Bij wijzigingen in de bestanden van Sharepoint, vervolgens verwerken in GitHub door 1 command te gebruiken			



## Aanpak/werkwijze

### Implementatie

De implementatie van het project bestaat uit een aantal fasen. Door bestaande onderdelen nogmaals te testen, komen wij erachter of er misschien nog andere verbeterpunten aanwezig zullen zijn. Naast het testen van de omgeving zelf, zullen we ook kijken of het mogelijk zal zijn om de bestaande databeheersystemen te kunnen vervangen. Bij de uitrol van het product, zal het product opgeleverd worden aan de hand van de testen die geslaagd dienen te zijn voor een volledig compleet omgeving. De overdracht van het project zal aan het einde van dit project plaatsvinden. Het project wordt dan overgedragen aan databeheer van de kennisgroep IoT en zal dan ook beheerd worden door de kennisgroep.

Wanneer er bepaalde onderdelen van de testfase vergeten, niet getest of gefaald zijn, kunnen deze door de kennisgroep opnieuw getest worden. Het opnieuw testen zal ook oriënterend zijn voor het gebruik van GitHub. Zover het eruit ziet is zal het opnieuw testen niet nodig zijn.

### Planning en mogelijke risico's

Wat	Wie	Wanneer	Benodigde materialen	Afhankelijk van
GitHub opbouwen voor SSC	Projectgroep	20-12-19	Eigen laptop	
Systeemmap Scalda uploaden	Projectgroep	20-12-19	Eigen laptop	SSC KG IOT
Gewijzigde bestanden doorvoeren/vervangen GitHub	Projectgroep	∞	Eigen laptop	SSC KG IOT
Synchronisatie Sharepoint	Projectgroep	18-12-19	Eigen laptop	6

GitHub opbouwen voor SSC	Gemiddelde optredingskans	
Systeemmap Scalda uploaden	Lage optredingskans	
Gewijzigde bestanden doorvoeren/vervangen GitHub	Gemiddelde optredingskans	
Synchronisatie Sharepoint	Lage optredingskans	

Laag risico		Lage optredingskans	1
Gemiddeld risico		Gemiddelde optredingskans	2
Hoog risico		Hoge optredingskans	3