



ADVIESRAPPORT

Internet of Things: GitHub

Shared Service Center
Scalda Vlissingen
Edisonweg 4a

Projectleider : Cihan Toksöz
Projectleden : Maikel Evegroen
Datum : 14 januari 2020
Klas : ICO43C

1 INLEIDING

Binnen in het Shared Service Center is er behoefte aan een gelijkwaardige document ter behoefte van vooruitgang van de informatievoorziening aan deelnemende leden en externe klanten. Deze behoefte is ontstaan uit de overdraagbaarheid van projecten aan projectleden. In dit adviesrapport is er beredeneerde advies voorgeschreven voor de opdrachtgever. Het advies sluit zich aan op een probleem en/of vraagstuk dat geadviseerd zou moeten worden bij de opdrachtgever. Op basis van onderzoeken is het document opgesteld. Als projectgroep hopen wij de opdrachtgever te kunnen overtuigen over de gegeven adviezen wat overgenomen kan worden bij mogelijke overdracht van GitHub.

MANAGEMENTSAMENVATTING

De vraag van de opdrachtgever was om een GitHub omgeving te ontwikkelen die wij in het Shared Service Center konden gaan gebruiken. Het moest beschikken over toegangscontrole en verschillende samenwerkingsfuncties. Wij als projectgroep moesten ervoor zorgen dat de wens van de klant gerealiseerd kon worden in een testomgeving. Het uiteindelijke doel was om een versiebeheersysteem te ontwikkelen 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. Deze behoefte is ontstaan uit de overdraagbaarheid van projecten aan projectleden.

INHOUDSOPGAVEN

| | | |
|---|-------------------------------------------------------------|---|
| 1 | INLEIDING | 2 |
| | MANAGEMENTSAMENVATTING | 2 |
| 3 | HET ONDERZOEK | 4 |
| | GITHUB-ANTWOORDEN | 4 |
| | WAT IS GIT? | 5 |
| | MEEST GEBRUIKTE WOORDEN ALS ER OVER GIT WORDT GEPRAAT | 6 |
| | GIT-SPECIFIEKE OPDRACHTEN | 7 |
| 4 | ALTERNATIEVEN | 8 |
| | GITKRAKEN | 8 |
| | GITLAB | 8 |
| | BITBUCKET | 8 |
| | SOURCEFORGE | 9 |
| 5 | HET ADVIES | 9 |

3 HET ONDERZOEK

Op basis van de centrale vraag is vastgesteld dat het gaat om een beschrijvend en implementerend onderzoek. Bij beschrijvend onderzoek gaat het om een beschrijving van een aantal kenmerken. In dit geval worden er kenmerken van de opdrachtgever uit de populatie beschreven. Het doel van dit onderzoek is inzicht verkrijgen in de wensen/behoefte van de opdrachtgever en de projectcoach. Het onderzoek vond plaats gedurende 2 sprints.

Tijdens het onderzoek was het belangrijk om te bepalen waar het onderzoek precies over ging. Om de vraag te kunnen beantwoorden, kozen wij als groep ervoor om een kwalitatief onderzoek te starten. Kwalitatief houdt in dat de dingen gedetailleerd omschreven worden. Toen het project net begon, kozen wij ervoor om langs te gaan bij IVS in Terneuzen. De jongens van IVS hadden een GitHub omgeving draaien, die blijkbaar samen met het SSC werkte. Tijdens de meeting met IVS kregen wij gelijk een beeld over GitHub. Wat je ermee kon doen en waar het voor bedoeld was. Hetzelfde was bij de opleiding applicatiebeheer. De applicatiebeheer gebruikte een tool Bitbucket. Bitbucket is een vergelijkbare versiebeheersysteem als GitHub. Tijdens een onderzoek is het altijd mooi meegenomen om een beetje fieldresearch te doen. Het meest gebruikte methode voor ons was toch een deskresearch.

Sinds 2013 kun je er niet meer omheen: je moet leren hoe je GitHub gebruikt. Waarom? Omdat het een sociaal netwerk is dat de manier waarop we werken volledig heeft veranderd. GitHub is begonnen als een samenwerkingsplatform voor ontwikkelaars en is nu de grootste online opslagruimte van samenwerkingswerken die er in de wereld bestaat. Voordat GitHub bestond, creëerden grote bedrijven hun kennis voornamelijk privé. Maar wanneer je de GitHub-pagina's opent van de grote bedrijven kun je alles wat ze aan het netwerk toevoegen downloaden, bestuderen en erop vertrouwen.

GITHUB-ANTWOORDEN

De meeste artikelen over 'GitHub voor beginners' die wij lazen waren ver boven onze hoofden. Dat kwam waarschijnlijk, omdat er geen sterke programmeerachtergrond was, zoals de meeste GitHub-gebruikers hebben.

Waar er misschien in vergist wordt is dat er genoeg redenen zijn om GitHub te gebruiken als je geen programmeur bent. Volgens de educatieve video's van GitHub kan elke kenniswerker profiteren die gebruik maakt van een computer.

Een van de belangrijkste misvattingen over GitHub is dat het een ontwikkelingstool is, evenzeer een onderdeel van codering als computertalen en programma's. GitHub zelf is echter niet veel meer dan een sociaal netwerk zoals Facebook of Flickr. Je maakt een profiel aan, uploadt projecten om te delen en maakt contact met andere gebruikers door hun accounts te "volgen". En hoewel veel gebruikers programma's en codeprojecten opslaan, staat niets je in de weg om tekstdocumenten of andere bestandstypen in uw projectmappen te bewaren.

Wat meer is, je kunt GitHub eigenlijk gebruiken zonder elke code te kennen. Je hebt niet echt een tutorial nodig om je aan te melden en rond te klikken. Maar ik denk wel dat het verdienen is om dingen eerst op de harde manier te leren, waarmee ik bedoel, met gewoon oude codering in Git. GitHub is immers toevallig een van de meest moeiteloze grafische interfaces voor de Git-programmeertaal.

WAT IS GIT?

Git is software voor versiebeheer, wat betekent dat het wijzigingen in een project beheert zonder een deel van dat project te overschrijven. En het zal niet snel verdwijnen, vooral omdat Torvalds en zijn collega-kernelontwikkelaars Git gebruiken om de kernel voor Linux te helpen ontwikkelen.

Waarom zouden we Git gebruiken? Stel dat een medestudent beide pagina's op dezelfde website bijwerkt. Je brengt je wijzigingen aan, slaat ze op en uploadt ze terug naar de website. Tot nu toe, zo goed. Het probleem komt wanneer uw collega op dezelfde pagina werkt als u tegelijkertijd. Een van jullie staat op het punt je werk te overschrijven en te wissen.

Een versiebeheerapplicatie zoals Git voorkomt dat jij en je medestudent elkaars werk niet kunnen overschrijven. Als er samen wordt gewerkt in een project kan elk lid revisies uploaden naar dezelfde pagina, en Git zal twee exemplaren opslaan. Later kunt je jouw wijzigingen samenvoegen zonder onderweg iets te verliezen. Je kunt zelfs op elk gewenst moment terugkeren naar een eerdere versie, omdat Git een "momentopname" bijhoudt van elke wijziging die ooit is aangebracht.

GitHub maakt Git op twee manieren gemakkelijker te gebruiken. Als je eerst de GitHub-software naar uw computer downloadt, biedt deze een visuele interface om je te helpen jouw versie-gecontroleerde projecten lokaal te beheren. Ten tweede brengt het maken van een account op GitHub.com uw versiebeheerde projecten op het web en verbindt het sociale netwerkfuncties voor een goede maatregel.

Je kunt bladeren door projecten van andere GitHub-gebruikers en zelfs exemplaren downloaden die je zelf kunt aanpassen en ervan kunt leren. Andere gebruikers kunnen hetzelfde doen met uw openbare projecten, en zelfs fouten ontdekken en oplossingen voorstellen. Hoe dan ook, er gaan geen gegevens verloren omdat Git een "momentopname" van elke wijziging opslaat.

Hoewel het mogelijk is om GitHub te gebruiken zonder Git te leren, is er een groot verschil tussen gebruik en begrip. Voordat ik Git ontdekte, kon ik GitHub gebruiken, maar ik begreep niet echt waarom.

MEEST GEBRUIKTE WOORDEN ALS ER OVER GIT WORDT GEPRAAKT

Opdrachtregel: het computerprogramma dat we gebruiken om Git-opdrachten in te voeren. Op een Mac heet dit Terminal. Op een pc is het een niet-native programma dat je downloadt wanneer je Git voor de eerste keer downloadt. In beide gevallen typt u op tekst gebaseerde opdrachten, ook wel prompts genoemd, in het scherm in plaats van een muis te gebruiken.

Repository: een map of opslagruimte waar jouw projecten kunnen leven. Soms verkorten GitHub-gebruikers dit tot "repo". Het kan lokaal zijn naar een map op uw computer, of het kan een opslagruimte zijn op GitHub of een andere online host. Je kunt codebestanden, tekstbestanden, afbeeldingsbestanden, noem maar op, in een repository bewaren.

Versiebeheer: kortom, het doel dat Git is ontworpen om te dienen. Wanneer je een Microsoft Word-bestand hebt, overschrijf je elk opgeslagen bestand met een nieuwe opslag of sla je meerdere versies op. Met Git hoeft je dat niet te doen. Het houdt "momentopnamen" bij van elk tijdstip in de geschiedenis van het project, zodat je het nooit kunt verliezen of overschrijven.

Commit: dit is het commando dat Git zijn kracht geeft. Wanneer je het vastlegt, maak je op dat moment een "momentopname" van uw repository, waarmee je een controlepunt krijgt waarmee je jouw project opnieuw kunt evalueren of in een eerdere staat kunt herstellen.

Branch: hoe werken meerdere mensen tegelijkertijd aan een project zonder dat Git ze in de war brengt? Meestal 'vertakken' ze zich van het hoofdproject met hun eigen versies vol met veranderingen die ze zelf hebben aangebracht. Nadat ze klaar zijn, is het tijd om die tak terug te "mergen" met de "master", de hoofddirectory van het project.

GIT-SPECIFIEKE OPDRACHTEN

Omdat Git is ontworpen met een groot project als Linux in gedachten, zijn er veel Git-opdrachten. Om de basis van Git te gebruiken, hoeft je echter maar een paar termen te kennen. Ze beginnen allemaal op dezelfde manier, met het woord 'git'.

git init: Initialiseert. Totdat je deze opdracht uitvoert in een repository of map, is het gewoon een gewone map. Pas nadat je dit hebt ingevoerd, accepteert het verdere Git-opdrachten.

git config: Afkorting voor 'configure', dit is het handigst wanneer je Git voor het eerst instelt.

git help: Een opdracht vergeten? Typ dit in de opdrachtregel om de 21 meest voorkomende git-opdrachten weer te geven. Je kunt ook specifieker zijn en 'git help init' of een andere term typen om erachter te komen hoe je een specifieke git-opdracht kunt gebruiken en configureren.

git status: Controleer de status van jouw repository. Bekijk welke bestanden erin zitten, welke wijzigingen nog moeten worden doorgevoerd en aan welke tak van de repository je momenteel werkt.

git add: Dit voegt geen nieuwe bestanden toe aan jouw repository. In plaats daarvan brengt het nieuwe bestanden onder de aandacht van Git. Nadat je bestanden hebt toegevoegd, worden ze opgenomen in Git's "snapshots" van de repository.

git commit: Git's belangrijkste commando. Nadat je een wijziging hebt aangebracht, voer je deze in om een "momentopname" van de gegevensopslagruimte te maken. Meestal gaat het git commit -m "Message here." De -m geeft aan dat het volgende gedeelte van de opdracht als een bericht moet worden gelezen.

git branch: Werk je met meerdere medewerkers en wil je zelf wijzigingen aanbrengen? Met deze opdracht kun je een nieuwe branch of tijdlijn van commits bouwen van wijzigingen en bestandsuitbreidingen die helemaal van jou zijn. Je titel gaat na het commando. Als je een nieuwe tak met de naam 'katten' wilde, typ je git branch cats.

git checkout: Hiermee kun je letterlijk een repository "uitchecken" die zich momenteel niet binnen bevindt. Dit is een navigatieopdracht waarmee je naar de repository kunt gaan die je wilt controleren. Je kunt deze opdracht gebruiken git checkout master om naar de hoofdtak git checkout cat te kijken, of om naar een andere tak te kijken.

git merge: Wanneer je klaar bent met het werken aan een branch, kun je jouw wijzigingen terugvoegen naar de master branch, die zichtbaar is voor alle bijdragers.

git push: Als je op je lokale computer werkt en je wilt dat je commits ook online zichtbaar zijn op GitHub, "push" je de wijzigingen naar GitHub met dit commando.

git pull: Als je op jouw lokale computer werkt en de meest up-to-date versie van uw repository wilt gebruiken, "haal" je de wijzigingen van GitHub naar beneden met deze opdracht.

4 ALTERNATIEVEN

Een belangrijk voordeel van GitHub is dat de dienst onbeperkt gratis, openbare code-repositories voor alle gebruikers beschikbaar stelt. Het gebruik van particuliere repositories is echter verbonden aan een maandelijks abonnement. In GitHub kun je ook 'organisaties' aanmaken: gedeelde accounts, waarmee meerdere mensen tegelijkertijd samen aan meerdere projecten kunnen werken.

Toch zitten er ook beperkingen aan de bruikbaarheid en efficiëntie van GitHub. Soms ontstaan er problemen met de klant en werkgever, als bijvoorbeeld een particuliere server als host dient voor de geschreven code. Nog een reden om voor een GitHub-alternatief te kiezen, is het gebruik van een ander versiebeheersysteem dat niet wordt ondersteund door GitHub. Er zijn goede alternatieven voor GitHub beschikbaar, die we hieronder kort aan je willen voorstellen.

GITKRAKEN

GitKraken is het volgende GitHub-alternatief. GitKraken hecht waarde aan tijdbesparing en dat komt de gebruiker bij het testen van de code ten goede. Het systeem staat bekend om de bijzonder aantrekkelijke interface, de focus op snelheid en de eenvoudige Git-bediening. Met een handige 'ongedaan maken'-functie kunnen eventuele fouten direct worden gecorrigeerd. Dat maakt het werkproces veel eenvoudiger. De gratis versie is bedoeld voor bedrijven met minder dan twintig werknemers of non-profitorganisaties. De pro-versie biedt meer nuttige functies, zoals profielondersteuning om verschillende projecten eenvoudig van elkaar te scheiden.

GITLAB

Als alternatief voor GitHub biedt GitLab een groot aantal handige functies in zijn DVCS, zoals een geïntegreerde projectwiki waar je o.a. je styleguide in kwijt kan, en een projectwebsite. De voortdurende integratiecapaciteiten van GitLab automatiseren het testen en leveren van code – wat weer tijd scheelt in de testfase. Met GitLab krijg je door middel van een code-viewer, pull-verzoeken en praktische conflictoplossing gemakkelijk toegang tot alle belangrijke onderdelen van je project. De toepassing is voornamelijk geschreven in Ruby.

BITBUCKET


BitBucket is een versiebeheer-repository-hostingservice van Atlassian. Het is erg aangepast met andere Atlassian-projectbeheertools zoals Jira, HipChat en Confluence. Dit maakt het een voorkeur voor grote ondernemingen. Maar je hoeft geen grote onderneming te zijn om BitBucket te gebruiken. Het heeft iets voor alles. Als je naar de gehoste accountprijs kijkt, kunt u zien dat deze gratis is voor projecten met maximaal vijf teamleden. Open source-projecten met meer dan vijf leden kunnen BitBucket nog steeds gratis gebruiken. Alles wat je moet doen is solliciteren voor gemeenschapslicentie en zich te houden aan de open source-richtlijnen van Atlassian.

SOURCEFORGE

SourceForge is een andere grote naam op deze lijst met GitHub-alternatieven. SourceForge is populair geweest bij open-sourceprojecten. Veel Linux-distributies en -projecten leveren hun downloads via SourceForge. Het stelt ontwikkelaars in staat om open source-projecten te creëren door alle noodzakelijke tools te leveren. De populariteit van Source Forge werd geraakt door de toename van een meer intuïtieve GitHub. Echter, onder de nieuwe leiding van Logan Abbott, SourceForge heeft zijn interface opnieuw ontworpen en richt zich op het terugwinnen van zijn verloren plek in open source code hosting. Voor GitHub-migranten biedt SourceForge tools om volledige GitHub-repositories te importeren of geselecteerde repository's in bestaande projecten. Deze video laat zien hoe deze tool te gebruiken:

5 HET ADVIES

Tijdens het onderzoeken en het implementeren van GitHub, zijn wij als groep tegen vele dingen aangelopen. Achterstand in het beheersen van Linux kan ervoor zorgen dat je niet weet waar je mee bezig bent. Tuurlijk is het altijd belangrijk om te weten waar je precies mee bezig bent. Tijdens de cursus van Linux leer je de basis van Linux. De commands die gebruikt worden bij Linux zijn niet de commands die je kunt gebruiken in Git Bash. Naar mijn mening zou het een mooi advies zijn om meer instructie en les te krijgen op het gebied van GitHub zolang het gebruikt zal worden. Dit zal ervoor zorgen dat de kennisgroep die databeheer/GitHub beheert, verstand van heeft. Het platform GitHub kan zijn taak behouden om misschien in de toekomst sharepoint over te nemen. Binnen in GitHub kun je de broncodes die gebruikt worden in projecten, plaatsen en delen via GitHub. Dit is eigenlijk de bedoeling van GitHub

| | |
|-----------------------------------------------------|------------------------------------------------------------------------------------|
| Handtekeningen | |
| Student: <i>Naam : Cihan Toksöz</i> | |
| Projectcoach: <i>Naam : Jeroen van Os</i> |  |
| Opdrachtgever: <i>Naam : Kees Krijger</i> | |
| Datum: 16 januari 2020 | |