



Professionele Bachelor Toegepaste Informatica



KPI visualisatie en beschrijving van het development proces

Joran Dirkzwager

Promotoren:

Peter Fastré
Niek Vandael

Level27
Hogeschool PXL Hasselt



Bachelorpaper Academiejaar 2018-2019



Professionele Bachelor Toegepaste Informatica



KPI visualisatie en beschrijving van het development proces

Joran Dirkzwager

Promotoren:

Peter Fastré
Niek Vandael

Level27
Hogeschool PXL Hasselt



Bachelorpaper Academiejaar 2018-2019

Dankwoord

Om te beginnen zou ik graag mijn dank betuigen aan alle medewerkers van Level27 voor hun geduld, motivatie en doorzetting. Zonder hun technische kennis maar ook algemene kennis zou dit eindwerk nooit tot een goed einde zijn gekomen.

Ik zou graag ook mijn bedrijfspromotor, Peter Fastré, nog eens apart willen bedanken voor de feedback en tips die me terug in de juiste richting stuurde als ik vast zat.

Tenslotte zou ik graag nog Niek Vandael, mijn hogeschoolpromotor willen bedanken. Niek las altijd alle ingediende opdrachten tijdig na en bezorgde me zo snel mogelijk de nodige feedback. Ook de mailtjes om ons te herinneren aan deadlines die eraan zaten te komen hebben me erg veel geholpen.

Joran Dirkwager

Abstract

Level27 is een bruisend bedrijf met als doel de dienstverlening van hostingbedrijven in een steeds veranderende IT-wereld te optimaliseren door een oplossing op maat aan te bieden. Na dertien jaar is Level27 dan ook geëvolueerd tot een bedrijfsverhaal dat geschreven wordt door een groep van toegewijde collega's die streven naar het leveren van betrouwbare, veilige en stabiele hostingdiensten op maat en daarbij de waarden van Level27 hoog in het vaandel dragen.

Om die diensten op een gebruiksvriendelijke manier aan te kunnen bieden is Level27 een controlepaneel aan het ontwikkelen waar wekelijks nieuwe functionaliteiten aan worden toegevoegd. Om ervoor te zorgen dat deze nieuwe ontwikkelingen zo vlot en efficiënt mogelijk bij de klant geraken, wordt er een bepaald stappenplan gevolgd dat in deel twee van deze bachelor proef wordt uitgeschreven. Dit stappenplan beschrijft het volledige ontwikkelingsproces van begin tot einde. Van *feature request* tot productie *release*. Van *bug* rapport tot *bug fix*.

Steeds meer bedrijven investeren in middelen voor het analyseren van resultaten om efficiënter en doelgerichter te werk te gaan. Ook Level27 is bezig met dit proces om zo snel mogelijk in te kunnen spelen op de snelgroeïende en veranderende markt.

Om een zicht te krijgen op de prestaties van de organisatie en zijn medewerkers wordt er gewerkt met een dataset van kritische succesfactoren, of KPI's, die worden bijgehouden in een Excel-bestand. Om die data visueel voor te stellen zijn er een aantal dashboards gemaakt met het softwarepakket Klipfolio. Dit softwarepakket heeft een aantal limitaties waaronder een *rate limiter* en het gebrek aan een mogelijkheid om de software zelf te kunnen hosten in plaats van de cloudapplicatie te gebruiken. Om die redenen werd er tijdens de stageperiode een vergelijkend onderzoek gedaan naar een *tool* die Klipfolio kan vervangen bij het visualiseren en analyseren van de KPI-data. In het onderzoek wordt er een vergelijking gemaakt tussen Klipfolio en drie alternatieven: Grafana, Power BI en QlikSense.

Tenslotte wordt de gekozen tool uitgewerkt en worden alle bestaande Klipfolio-dashboards overgezet naar de gekozen tool, Grafana.

Inhoudsopgave

Dankwoord	ii
Abstract	iii
Inhoudsopgave	iv
Lijst van gebruikte figuren	viii
Lijst van gebruikte tabellen	ix
Lijst van gebruikte code fragmenten	x
Lijst van gebruikte afkortingen	xi
Inleiding	1
I. Stageverslag	2
1 Bedrijfsvoorstelling Level27	2
1.1 Situering	2
1.2 Level27 in cijfers	2
1.3 Diensten	2
1.3.1 Classic hosting	2
1.3.2 Innovatieve projecten	2
1.3.3 Enterprise operations	2
1.4 Kwaliteit	3
2 Voorstelling stageopdracht 1: Migratie van Klipfolio naar Grafana	3
2.1 Probleemstelling	3
2.2 Doelstellingen	4
3 Uitwerking stageopdracht 1	4
3.1 Databank	4
3.1.1 Excel	5
3.1.1 MySQL	5
3.2 Tv-schermen	6
3.3 Refresh-rate	6
3.4 Grafana	7
3.4.1 Databronnen	7
3.4.2 Dashboards	7
3.4.3 Grafiektypes	7
3.4.4 Grafiek maken	8
3.4.5 Opdelingen maken	9
3.5 Dashboards	9
3.5.1 Wachtdienst	10

3.5.2	Logs	10
3.5.3	Delivery	11
3.5.4	Algemene KPI	12
3.5.5	Andere dashboards	12
3.6	Conclusie	12
4	Voorstelling stageopdracht 2: In kaart brengen van het ontwikkelingsproces	14
4.1	Probleemstelling.....	14
4.2	Doelstellingen.....	14
4.3	De software	15
4.3.1	Technologieën.....	15
4.3.2	Functionaliteiten	15
5	Uitwerking stageopdracht 2	18
5.1	Jira	18
5.1.1	Epic.....	19
5.1.2	Story	19
5.1.3	Task.....	19
5.1.4	Sub-Task.....	19
5.1.5	Bug	20
5.2	Sprintbord	20
5.3	Issue maken.....	20
5.3.1	Nieuwe functionaliteit	20
5.3.2	Bug rapportering	21
5.4	Planning	21
5.4.1	Schatting.....	21
5.4.2	Prioriteit.....	21
5.5	Discussie.....	22
5.6	Ontwikkeling.....	22
5.7	Documentatie	22
5.7.1	API-documentatie.....	22
5.7.2	Gebruikershandleiding	22
5.8	Testen.....	23
5.8.1	Manueel testen.....	24
5.8.2	Automatische API testen	24
5.9	Release.....	24
5.9.1	Git release.....	25
5.9.2	Jira release	25
5.10	Deployment.....	25

5.11	Communicatie	26
5.11.1	Interne communicatie	26
5.11.2	Communicatie naar klanten.....	26
5.12	Conclusie	26
5.12.1	Sprint planning	26
5.12.2	Automatische GUI testen	27
5.12.3	Interne kennisdeling	28
5.12.1	Sprint retrospective	28
5.12.1	Aantal releases	28
5.13	Reflectie	29
II.	Onderzoekstopic	31
1	Onderzoeksvraag	31
2	Onderzoeksmethode	31
3	Uitwerking onderzoek.....	32
3.1	Klipfolio	32
3.1.1	Uitwerking voorbeeld.....	34
3.1.2	Mobiele applicatie	34
3.1.3	Prijs.....	34
3.1.4	Delen van data	35
3.2	Power BI	35
3.2.1	Power BI installeren	35
3.2.2	Uitwerking voorbeeld.....	36
3.2.3	Mobiele applicatie	37
3.2.4	Prijs.....	38
3.2.5	Delen van data	38
3.3	Qlik.....	38
3.3.1	QlikView of QlikSense	38
3.3.2	QlikSense installeren.....	38
3.3.3	Uitwerking voorbeeld.....	39
3.3.4	Mobiele applicatie	40
3.3.5	Prijs.....	40
3.3.6	Delen van data	40
3.4	Grafana	40
3.4.1	Grafana installeren	40
3.4.2	Uitwerking voorbeeld.....	41
3.4.3	Mobiele applicatie	42
3.4.4	Prijs.....	42

3.4.5	Delen van data	42
3.5	Andere vergelijkingen	43
3.5.1	BI totale kostenplaatje	43
3.5.2	On-premise	43
3.5.3	Customer support.....	43
3.5.4	Vergelijkmatrix	44
	Conclusie	44
	Bibliografie	45
	Bijlage	48

Lijst van gebruikte figuren

Figuur 1: Voorbeeld Klipfolio dashboard	3
Figuur 2: tv's met dashboards op het kantoor van Level27	6
Figuur 3: Grafana refresh-rate	7
Figuur 4: Grafana Cirkeldiagram	7
Figuur 5: Voorbeeld lijngrafiek	8
Figuur 6: Voorbeeld lijngrafiek met balken	9
Figuur 7: Wachtdienst dashboard	10
Figuur 8: Log dashboard	11
Figuur 9: Delivery dashboard	11
Figuur 10: KPI dashboard	12
Figuur 11: CP4 - dashboard	15
Figuur 12: CP4 - Domeinregistratie	16
Figuur 13: CP4 - mailboxen	17
Figuur 14: CP4 - appcomponent aanmaken	17
Figuur 15: CP4 - systeem statistieken	18
Figuur 16: issue types	19
Figuur 17: Sprints	19
Figuur 18: Sprint bord	20
Figuur 19: reStructuredText	23
Figuur 20: Level27 guide	23
Figuur 21: Release proces	24
Figuur 22: Aantal releases van Januari tot en met Juni	28
Figuur 23: Dagen tussen release Januari tot en met Juni	29
Figuur 24: Logo Klipfolio	32
Figuur 25: Uitwerking voorbeeld in Klipfolio	34
Figuur 26: Klipfolio mobiele app	34
Figuur 27: Power BI logo [34]	35
Figuur 28: Uitwerking voorbeeld in Power BI	36
Figuur 29: HOUR functie in Power BI	36
Figuur 30: Analyse functie in Power BI	37
Figuur 31: Power BI mobiele app	37
Figuur 32: Logo Qlik	38
Figuur 33: Qlik suggesties	39
Figuur 34: Uitwerking voorbeeld in QlikSense	39
Figuur 35: Logo Grafana [37]	40
Figuur 36: Loginpagina Grafana	41
Figuur 37: Uitwerking voorbeeld in Grafana	41
Figuur 38: Grafana query-builder:	42
Figuur 39: Grafana browser op mobiel toestel	42
Figuur 40: Jaarlijkse TCO – verschil tussen platform en werk kosten	43

Lijst van gebruikte tabellen

Tabel 1: KPI Excel	5
Tabel 2: KPI database	5
Tabel 3: BI of Data analytics [22]	31
Tabel 4: Vergelijkmatrix PowerBI, Qlik en Grafana	44

Lijst van gebruikte code fragmenten

Code 1: Aantal inkomende en gemiste oproepen	8
Code 2: voorbeeld WHOIS	16
Code 3: voorbeeld Magallanes configuratie bestand	26
Code 4: Magallanes bouwproces starten.....	26
Code 5: query om voorbeeld uit te werken	32
Code 6: Grafana installeren.....	40
Code 7: Grafana-server starten	41

Lijst van gebruikte afkortingen

AWS	<i>Amazon Web Services</i>
DMS	<i>Database Management System</i>
DNS	<i>Domain Name System</i>
KPI	<i>Key Performance Indicator</i>
TTL	<i>Time To Live</i>
URL	<i>Uniform Resource Locator</i>
VPS	<i>Virtual Private Server</i>
BI	<i>Business Intelligence</i>
API	<i>Application Programming Interface</i>
JSON	<i>JavaScript Object Notation</i>
BDD	<i>Behaviour-Driven Development</i>
TCO	<i>Total Cost of Ownership</i>
FOSS	<i>Free and Open-Source Software</i>

Inleiding

Dit eindwerk is opgedeeld in twee delen: het eerste deel is een stageverslag over het in kaart brengen van het ontwikkelingsproces binnen Level27 en de migratie van een data visualisatie tool naar een, bij voorkeur, *open-source* alternatief. Het tweede deel is een onderzoek waaruit zal blijken welke software het meest geschikt is om de huidige data visualisatie tool te vervangen.

Er is bij Level27 geen documentatie geschreven dat het *development* proces in kaart brengt. Hierdoor is het moeilijk om nieuwe ontwikkelaars aan het team toe te voegen en verloopt het ontwikkelingsproces niet optimaal waardoor er wel eens een deadline wordt gemist. In het tweede deel van de stageopdracht wordt het *development* proces van begin tot einde volledig beschreven en schematisch voorgesteld.

Bij Level27 wordt er een Excel-sheet met *Key Performance Indicators* bijgehouden die op wekelijkse basis wordt aangevuld met de gegevens van de afgelopen week.

Deze data wordt visueel weergegeven met behulp van de cloudapplicatie, Klipfolio. Met deze software kunnen er een aantal databronnen toevoegen worden, zoals een Excel-sheet of een *API-call* naar een databron, die dan visueel voorgesteld kunnen worden in de vorm van grafieken. Klipfolio heeft echter een aantal limitaties, waaronder een restrictie op de *refresh-rate* van de databronnen. Level27 zou Klipfolio graag willen vervangen door een applicatie waar deze limitaties geen rol in spelen.

Vooraleer de stageopdracht volledig zal kunnen worden uitgevoerd zal er eerst een vergelijkend onderzoek gedaan moeten worden waarin de volgende vraag beantwoord wordt: welke *software* is het meest geschikt om Klipfolio te vervangen bij Level27 en waarom?

I. Stageverslag

Het eerste deel van dit eindwerk beschrijft de stageperiode bij Level27. Om de stageopdrachten wat beter te begrijpen wordt het bedrijf eerst voorgesteld. Daarna worden de twee opdrachten uitgewerkt. Bij iedere opdracht wordt er een conclusie geformuleerd.

1 Bedrijfsvoorstelling Level27

1.1 Situering

Level27 is een bedrijf opgericht door Peter Fastré die op zevenentwintigjarige leeftijd de uitdaging aanging om de dienstverlening van hostingbedrijven in een steeds veranderende IT-wereld te optimaliseren. Na dertien jaar is Level27 dan ook geëvolueerd tot een bedrijfsverhaal dat geschreven wordt door een groep van toegewijde collega's die streven naar het leveren van betrouwbare, veilige en stabiele hostingdiensten op maat en daarbij de waarden van Level27 hoog in het vaandel dragen.

1.2 Level27 in cijfers

Level27 is gesitueerd in hartje Bilzen, Limburg en telt 14 medewerkers die allen een rol spelen in het hosten van meer dan 5000 websites. Deze websites worden gehost op 1000 servers die Level27 heeft gebouwd op eigen infrastructuur in 4 datacentra.

1.3 Diensten

Level27 biedt drie soorten hosting aan namelijk: Classic hosting, Innovatieve projecten en Enterprise operations. [1]

1.3.1 Classic hosting

Het eerste soort hosting dat Level27 aanbiedt is het klassieke hosting verhaal. Dit omvat het registreren van domeinnamen, het hosten van PHP websites en MySQL databanken, het hosten van een mailinfrastructuur zodat klanten kunnen mailen vanuit een domeinnaam en het hosten van Virtual Private Servers met een publiek IP.

Sinds kort host ASP.NET websites die geserveerd worden van een Windows server en een connectie leggen met een MSSQL databank.

1.3.2 Innovatieve projecten

Het tweede soort zijn de innovatieve projecten waar Level27 zijn expertise aanbiedt. Level27 kan advies geven of helpen bij het opzetten van een op maat gemaakt deployment pijplijn of bij het *deployen* van een *custom-made* software project.

Ook AWS Autoscaling en Kubernetes opstellingen zijn toegevoegd aan het aanbod van Level27.

1.3.3 Enterprise operations

Tenslotte is er nog het laatste soort, de Enterprise operations. Deze diensten zijn gericht op grotere ondernemingen die het beheer van grote ERP-systemen zoals SAP willen uitgeven en behoefte hebben aan een 24/7 service desk.

Er kan ook gebruik gemaakt worden van de remote desktop diensten door het instellen van een VPN verbinding zodat de werknemers van een onderneming van thuis uit of vanuit het buitenland hun kantoor applicaties of netwerk kunnen gebruiken

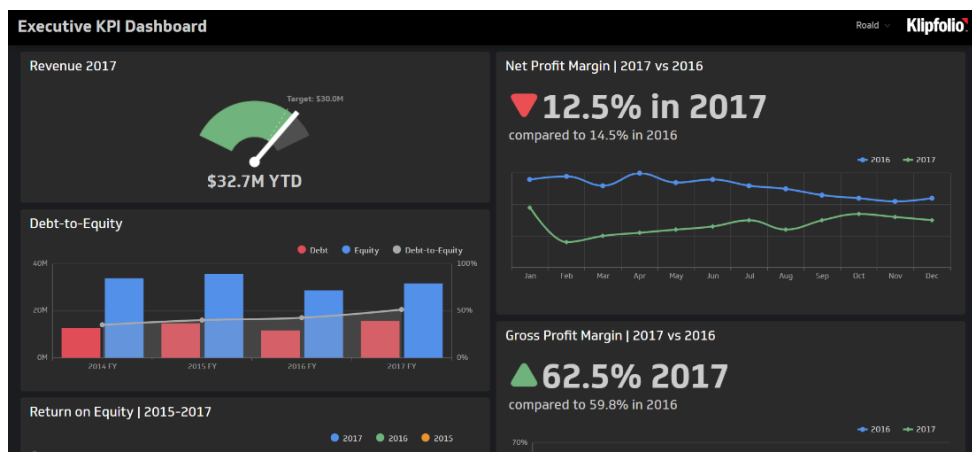
1.4 Kwaliteit

Eén van de waarden van Level27 is de primaire focus leggen op het leveren van kwaliteit. Deze kwaliteit wordt zowel voor, tijdens als na het opleveren van een dienst aangehouden. Om woorden in daden om te zetten werd er in 2017 concreet aan de slag gegaan met het behalen van twee ISO-certificaten. Als hostingbedrijf was ISO 27001 een logisch doel, want de klant wil graag een geruststelling krijgen dat zijn gegevens in veilige handen zijn. ISO 27001 beschrijft namelijk een aantal voorwaarden en specificaties op het vlak van *information security management*. [2] Na het behalen van ISO 27001 was het de bedoeling om nog een stapje verder te gaan, een stap richting consistentie. Het ISO 9001 certificaat betekent namelijk dat de kwaliteit van de eindproducten gegarandeerd wordt door de manier waarop Level27 werkt.

2 Voorstelling stageopdracht 1: Migratie van Klipfolio naar Grafana

2.1 Probleemstelling

Key performance indicators of kritieke succesfactoren zijn gegevens waarmee de prestaties, efficiëntie en faalpunten van een organisatie mee gemeten en geanalyseerd kunnen worden. Een organisatie kan vele soorten prestatie gegevens bijhouden maar de KPI is de data die te maken heeft met het behalen van de primaire doelen van de organisatie. [3]



Figuur 1: Voorbeeld Klipfolio dashboard

Om deze KPI data visueel weer te geven maakt Level27 gebruik van Klipfolio, een browserapplicatie waarmee data visueel weergegeven kan worden aan de hand van grafieken. Om deze data weer te geven in Klipfolio maakt Level27 gebruik van een Excel-sheet waarin de KPI's bijgehouden worden. Deze Excel-sheet wordt iedere maandagochtend door de officemanager aangevuld met de gegevens van de afgelopen week.

Een Excel-sheet is relatief handig in gebruik maar om complexe en specifieke data snel te verkrijgen uit een reeks gegevens is het beter om gebruik te maken van een **databank** waar query's op uitgevoerd kunnen worden. Een eerste obstakel is dus het gebruik van Excel om gegevens bij te houden en op te vragen.

Een derde probleem is de data **refresh-rate** die wordt opgelegd door Klipfolio. Gegevensbronnen kunnen maar een maximumaantal keren per uur hernieuwd worden. Het laagste dat gegaan kan worden is één hernieuwing per minuut. Voor de KPI-data, die één keer per week wordt geüpdatet is dit geen enkel probleem, maar in de toekomst wilt Level27 nog andere data gaan visualiseren zoals server prestaties en het monitoren van servers. Als er gebruik gemaakt wordt van bijvoorbeeld Grafana, een *open-source* alternatief, kan er een *refresh-rate* van één keer om de vijf seconden ingesteld worden. [4]

Om de dashboards weer te geven op het kantoor van Level27 worden er **tv-schermen** gebruikt. Omdat er voor Klipfolio geen specifieke tv-applicatie bestaat wordt er via de browser van de tv naar de webapplicatie gesurft om de dashboards weer te geven. De tv-browsers worden niet ondersteund door Klipfolio waardoor er gewoon een zwart scherm wordt weergegeven.

2.2 Doelstellingen

De huidige KPI's zijn al een hele tijd geleden opgesteld, maar intussen is het bedrijf sterk gegroeid qua omzet, infrastructuur, klanten en werknemers dus er wordt gekeken of deze nog relevant zijn en of er extra indicatoren bij moeten komen.

Vervolgens wordt de data geëxporteerd van een Excel-*sheet* naar een *database* zoals MySQL. Om dit te doen wordt er eerst nagedacht over de structuur en opmaak van de *database* tabellen. Sommige data in de Excel-*sheet* heeft correctie nodig vooraleer deze geïmporteerd kan worden naar de database.

Hierna wordt er een omgeving opgezet om Grafana, het alternatief voor Klipfolio op te installeren. Deze zal geïnstalleerd worden en gebruik maken van de *database* als gegevensbron.

Dan zullen er query's geschreven worden om de correcte data uit de *database* te halen die gebruikt wordt om de data visueel voor te stellen.

Als de tool uiteindelijk toegang heeft tot de correcte data worden de Klipfolio Klips nagemaakt en worden de dashboards samengesteld welke dan moeten worden weergegeven op tv-schermen.

Verder zijn er nog andere uitbreidingen die uitgewerkt worden zoals het toevoegen van een Elasticsearch databron om real-time logs op te vragen en grafieken weer te geven om zo snel mogelijk in te grijpen waar nodig.

Een tweede uitbreiding werd uitgewerkt in het eindwerk "System data collector" door Berre Mertens, een tweede stagiair bij Level27 waar er een monitoringsysteem werd opgezet met behulp van Prometheus en Grafana. Er wordt hier binnen deze bachelor proef niet verder op ingegaan. [5]

3 Uitwerking stageopdracht 1

3.1 Databank

Zoals in de inleiding al aangehaald is, is het gebruik van een Excel-*sheet* niet ideaal om een reeks gegevens op een snelle en efficiënte manier op te halen. De huidige KPI Excel-*sheet* gaat geëxporteerd worden naar een nieuwe tabel in een bestaande *database*.

3.1.1 Excel

De Excel-sheet met de KPI-data gebruikt kolommen in plaats van rijen voor het opslaan van een nieuwe record. Om gebruik te maken van een database zal de data eerst op een andere manier bijgehouden moeten worden.

In Tabel 1: KPI Excel wordt er een voorbeeld gegeven van hoe de data nu in Excel wordt opgeslagen.

Category	Name	24/12/2018	17/12/2018	10/12/2018	03/12/2018
Accuracy	Office-hours	3	4	3	4
Help	Phone calls received	16	49	48	62
Help	Slack messages	1511	6048	6396	4749
Ops	Critical	894	666	736	452

Tabel 1: KPI Excel

3.1.1 MySQL

Een eerste oplossing voor het probleem is het roteren van de data zodat er met rijen in plaats van kolommen gewerkt wordt maar dit neemt nog steeds het probleem niet weg dat de tabel moeilijk uit te breiden is want één week blijft nog steeds maar één record in de database. Als er een nieuwe KPI wordt toegevoegd zal hier rekening mee gehouden moeten worden.

Een tweede manier om de data op te slaan op een manier die gebruiksvriendelijk is en gemakkelijk om uit te breiden is door gebruik te maken van één rij per waarde. Op deze manier zijn er uiteindelijk maar vier kolommen nodig en kan er gezocht worden op de naam van een bepaalde KPI waarde. Door dit database-schema te gebruiken kan er ook gemakkelijk uitgebreid worden, er kan gemakkelijk begonnen worden met het bijhouden een nieuwe KPI waarde. Het database-schema heeft de volgende kolommen:

- Id
- Date
- Name
- Value

De database ziet er na de migratie uit zoals weergegeven in Tabel 2: KPI database.

id	Date	name	value
1	2019-05-20 00:00:00	general-deliverydocs-created	0
22	2019-05-20 00:00:00	general-deliverydocs-errors	0
43	2019-05-20 00:00:00	general-deliverydocs-finished	1
64	2019-05-20 00:00:00	planning-deadline-missed	0

Tabel 2: KPI database

Iedere rij omvat nu de waarde voor een KPI dat gelinkt is aan een date-time.

Nadat de data geïmporteerd is kan er heel gemakkelijk data uit de database gehaald worden door te filteren op de naam van de KPI of door te filteren op de datum.

Het is echter wel belangrijk dat er gebruik gemaakt wordt van een Read-Only MySQL gebruiker omdat Grafana een connectie legt naar een bestaande MySQL instance op een andere server. Dit om veiligheid redenen zodat de Grafana gebruiker enkel kan lezen van de database en niets kan aanpassen.

3.2 Tv-schermen

Éen van de vereisten is dat Grafana weergegeven kan worden op een televisiescherm, een functionaliteit die niet werkt bij Klipfolio. Als er via de browser van een televisie gesurft wordt naar de Klipfolio webapplicaties verschijnt er een zwart scherm en wordt de volgende boodschap weergegeven: “Not connected”.

Er is contact opgenomen met de klantendienst van Klipfolio om het probleem op te lossen maar het is een gekend probleem en er wordt voorlopig niet gewerkt aan een oplossing en er wordt ook geen alternatief of *workaround* aangeboden.

Buiten het migreren is er nog een andere oplossing mogelijk, namelijk de aankoop van één Raspberry Pi per tv-scherm. Dit zijn minicomputers ter grootte van een creditkaart, die aangesloten kunnen worden op een monitor of tv. Dit zou echter een investering van ongeveer 300 tot 400 euro betekenen voor een functionaliteit die gratis is door over te stappen naar Grafana.

Doordat Grafana deze functionaliteit ondersteunt is het mogelijk om de verschillende dashboards weer te geven op de televisieschermen die ophangen in de kantoorruimte van Level27.

Na de migratie naar Grafana is het terug mogelijk om dashboards te tonen op tv-schermen, zoals gezien kan worden op Figuur 2: tv's met dashboards op het kantoor van Level27.



Figuur 2: tv's met dashboards op het kantoor van Level27

3.3 Refresh-rate

Klipfolio heeft rate-limiters die voorkomen dat gebruikers geen bronnen kunnen raadplegen die meer dan 10 MB aan data teruggeven en die het aantal hernieuwingen van een databron

limiteert tot één keer per minuut. Vooral het aantal hernieuwingen per minuut is een limitatie als er gebruik gemaakt gaat worden van het monitoren van systeem logs.

De *refresh-rate* wordt bij Klipfolio ingesteld op het databron-niveau, dit betekend dat ieder dashboard dat gebruik maakt van een databron ook wordt geüpdatet. Bij Grafana wordt de *refresh-rate* ingesteld op een dashboard niveau. Iedere bron waar gebruik van gemaakt wordt binnen een bepaald dashboard wordt geüpdatet op het interval dat ingesteld wordt op het openstaande dashboard.

Door gebruik te maken van Grafana kunnen de databronnen van een dashboard iedere 5 seconden geüpdatet worden wat het ideaal maakt om systemen te monitoren of om systeem logs weer te geven.



Figuur 3: Grafana refresh-rate

3.4 Grafana

3.4.1 Databronnen

Er zijn een aantal mogelijke databronnen beschikbaar in Grafana waaronder Elasticsearch, Prometheus, Azure en MySQL. Het voordeel van een groot assortiment aan databronnen is dat Grafana dan niet enkel gebruikt kan worden voor KPI-data maar ook voor het monitoren van servers of het bekijken van logs.

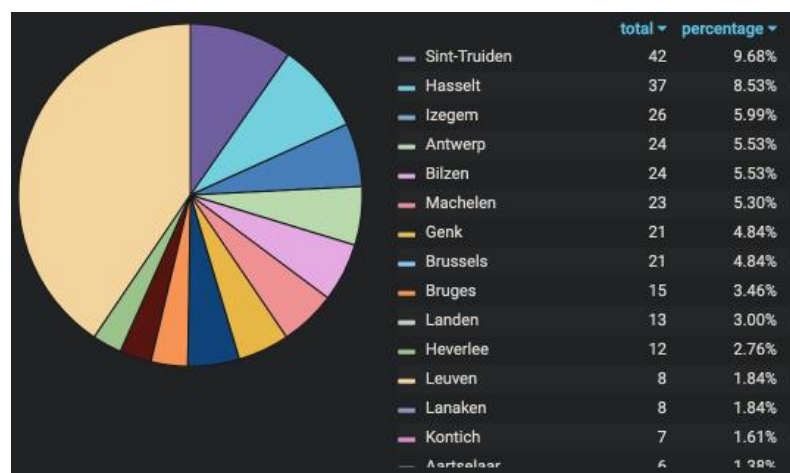
Voor de stageopdracht volstaat enkel een MySQL-connectie. Als uitbreiding op de stageopdracht is er ook een Elasticsearch connectie gemaakt om logs uit te lezen die worden verzameld door Logstash.

3.4.2 Dashboards

Wanneer er een databron is toegevoegd kan er een dashboard worden aangemaakt. Dashboard kunnen worden onderverdeeld in verschillende mappen om een bepaalde structuur aan te brengen. Zo kan er bijvoorbeeld een map gemaakt worden voor iedere afdeling binnen Level27 waarvoor er KPI-data wordt bijgehouden waarin dan verschillende dashboards aangemaakt kunnen worden.

3.4.3 Grafiektypes

Zoals er in Klipfolio verschillende Klips zijn, zijn er in Grafana ook een aantal visualisatie types die de naam panels krijgen. Bij de installatie zijn er 9 verschillende *panels* die gebruikt kunnen worden maar er zijn nog extra types die geïnstalleerd kunnen worden in de vorm van *plug-ins*. Een voorbeeld van een visualisatie dat niet standaard bij de installatie komt maar waarvoor wel een *plug-in* geïnstalleerd kan worden is de cirkeldiagram.



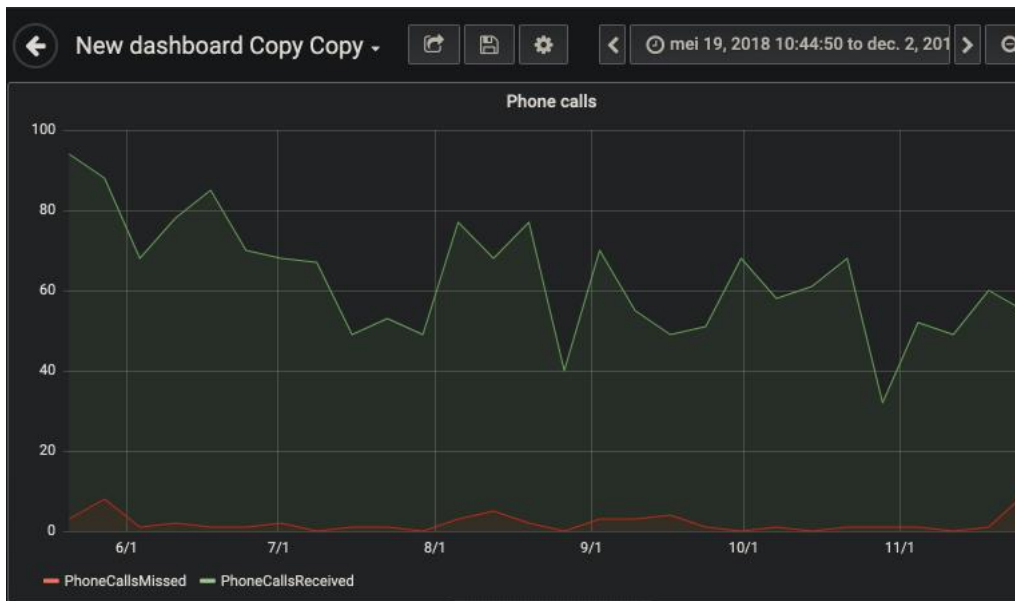
Figuur 4: Grafana Cirkeldiagram

Andere veelgebruikte vormen van diagrammen of visualisatie types zijn lijndiagrammen, staafdiagrammen en tabellen.

3.4.4 Grafiek maken

Een grafiek maken bestaat altijd uit vier stappen, waarbij de eerste stap altijd bestaat uit het kiezen van een databron en het opvragen van de gewenste gegevens met behulp van een query.

In Grafana wordt data altijd weergegeven over een bepaalde tijdspanne. Er wordt bijvoorbeeld altijd weergegeven hoe de dataset eruit zag vanaf een bepaalde datum tot een bepaalde datum. Daarom kan ten alle tijden in de rechterbovenhoek het tijdsbereik worden ingesteld.



Figuur 5: Voorbeeld lijngrafiek

Om ervoor te zorgen dat Grafana gebruik kan maken van de data over tijd functionaliteit moet er altijd een tijdsveld in de dataset aanwezig zijn.

Om gebruik te maken van de gegevens over tijd kan er gebruik gemaakt worden van één van de ingebouwde macro's in het SELECT gedeelte van de query en in het WHERE gedeelte van de query.

In

Figuur 5: Voorbeeld lijngrafiek

wordt er gebruik gemaakt van de volgende query:

```
1. SELECT
2.   $__timeGroupAlias(Date,$__interval),
3.   PhoneCallsMissed,
4.   PhoneCallsReceived
5. FROM kpiTable
6. WHERE
7.   $__timeFilter(Date)
8. ORDER BY 1
```

Code 1: Aantal inkomende en gemiste oproepen

In deze query wordt er, aan de hand van de \$__timeGroupAlias macro, aan Grafana verteld dat de Date het tijdsveld van de dataset is. Vervolgens worden de waarden van de kolom PhoneCallsMissed en PhoneCallsReceived opgehaald voor respectievelijk het aantal ontvangen en het aantal gemiste oproepen te verkrijgen. Verder wordt er aangegeven van welke tabel de data komt en wordt er in de WHERE clause aangegeven dat enkel de data van de aangegeven tijdsaanduiding moet worden weergegeven aan de hand van een tweede filter namelijk de \$__timeFilter() die de kolom met de tijd mee krijgt als argument.

In de tweede stap van het maken van een grafiek kan de opmaak worden gedefinieerd. Zo kan dezelfde lijngrafiek ook worden weergegeven in de vorm van balken in plaats van lijnen.



Figuur 6: Voorbeeld lijngrafiek met balken

3.4.5 Opdelingen maken

Veel bedrijven hebben eindeloos veel grafieken die cijfers voorstellen en die op zich wel allemaal nuttig zijn maar door zo veel grafieken te hebben kan het al snel gebeuren dat men het bos door de bomen niet meer ziet en dat het nut van de grafieken niet meer duidelijk is.

Dit kan verholpen worden door een selectie te maken en de dashboards op te splitsen in categorieën. Na het opsplitsen kan ieder dashboard gevuld worden met grafieken zodat er niet gescrold moet worden en de situatie in één oogopslag beoordeeld kan worden.

Bij Level27 is er gekozen om een opsplitsing te maken volgen afdeling. Het wil niet zeggen dat iedere afdeling maar één dashboard kan hebben maar het is wel dat ieder onderdeel maar één dashboard heeft.

3.5 Dashboards

De grafieken in Klipfolio hebben geen nuttige indeling en zijn allemaal aan één dashboard toegevoegd. Na de migratie zijn de grafieken opgedeeld in nuttige categorieën welke hieronder worden overlopen.

3.5.1 Wachtdienst

Iedere nacht is er een verantwoordelijke van wacht die paraat staat als er diensten kritieke meldingen geven. Aan de hand van het wachtdienst dashboard kan de wachtdienst de dag nadien in één oogopslag zien hoe vaak er problemen geweest zijn. Ook kan er gekeken worden op welke server het probleem zicht voorbeeld en van welk type de melding was.



Figuur 7: Wachtdienst dashboard

Door de tijdspanne te vergroten kan er ook gezien worden welke systemen de meeste problemen geven op week, maand of zelfs jaarbasis. De databron voor dit dashboard is een tabel in de MySQL database van de Nagios server, een systeem dat in staat is voor het monitoren van servers. Als er een melding binnenkomt wordt deze weggeschreven naar de database.

3.5.2 Logs

Een tweede dashboard is het log dashboard. Het dashboard geeft het aantal gegenereerde logs weer, opgesplitst per niveau. Hoe hoger het niveau, hoe sneller er aan de alarmbel getrokken moet worden. Het is ook mogelijk om een Slack integratie te maken voor Grafana. Zo kan er bijvoorbeeld een Slack-melding gestuurd worden als er een log van het emergency-niveau binnenkomt.

Een tweede zinvolle grafiek op dit dashboard is de Log history, op deze grafiek worden het aantal logs over tijd weergegeven. Zo is het gemakkelijk om grote concentraties van logs te zien of om grote pieken van logs te zien en te achterhalen waar ze vandaan komen.



Figuur 8: Log dashboard

Dit dashboard maakt gebruik van de ELK-stack om zijn data te verzamelen. Grafana kan verbinding maken met Logstash om zo doorheen de geïndexeerde logs te navigeren of om deze grafisch weer te geven. Het log dashboard is een voorbeeld van een dashboard dat voordeel haalt uit een snelle *refresh-rate*.

3.5.3 Delivery

Op het delivery dashboard kan er in een oogopslag gezien worden hoeveel nieuwe opleveringsdocumenten er gemaakt zijn binnen een periode. Een opleveringsdocument wordt opgesteld wanneer een klant een nieuwe VPS bij ons zal afnemen en bevat alle informatie qua specificaties van de hardware, details zoals versienummers en OS, en andere wensen van de klant.

Als zo een opleveringsdocument te weinig of foutieve informatie bevat moet dit gemeld worden aan het sales-team en zal dit worden opgenomen in de KPI. Dit dashboard gebruikt een connectie naar de KPI MySQL databank als databron.



Figuur 9: Delivery dashboard

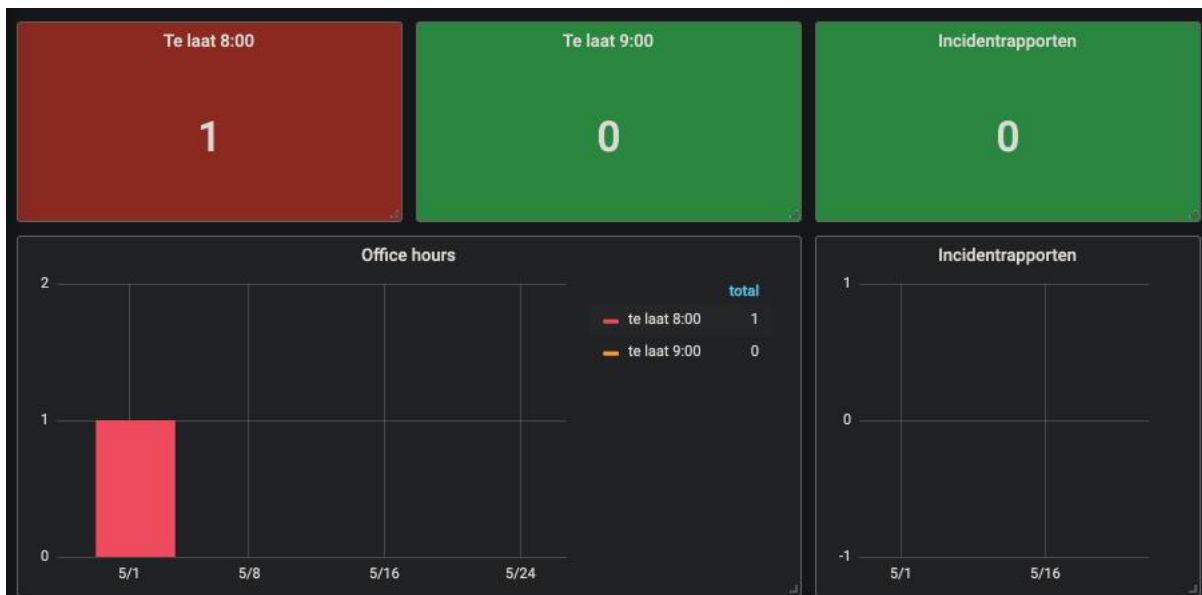
3.5.4 Algemene KPI

Een laatste voorbeeld zijn algemene KPI's die de dagelijkse werking van het bedrijf weergeven. Zo wordt het aantal mensen dat te laat komt of het aantal incidentenrapporten bijvoorbeeld weergegeven. Een incidenten rapport wordt opgemaakt als er iets misloopt en de klant er impact aan ondervindt. Het rapport beschrijft dan vier puntjes:

- Wat er is gebeurd
- Een tijdlijn van stappen die zijn ondernomen tot het probleem is opgelost
- De oorzaak van het incident
- Een plan om herhaling te voorkomen

Na afloop wordt ontvangt de klant een kopie van het rapport.

Ook dit dashboard maakt gebruik van de KPI MySQL database om gegevens op te vragen.



Figuur 10: KPI dashboard

3.5.5 Andere dashboards

Een aantal andere dashboards zijn:

- Het support dashboard dat support statistieken weergeeft zoals het aantal inkomende telefoonoproepen en het aantal binnenkomende support-tickets.
- Het development dashboard dat ontwikkeling cijfers weergeeft zoals het aantal releases en het aantal *commits*.
- Het Finance dashboard geeft vooral een overzicht van openstaande facturen en het aantal bijhorende herinneringen die uitgestuurd zijn geweest.

Al deze dashboards worden nog steeds wekelijks aangepast en bevinden zich dus ook nog lang niet in hun finale staat.

3.6 Conclusie

In de probleemstelling worden er drie thema's aangehaald, namelijk: *databank*, *refresh-rate* en tv-schermen.

In Klipfolio, de oude tool, staat er een limiet op het aantal databronnen en is er een refresh-rate limitatie die de databronnen er van weerhoudt om vaker dan één keer per minuut te hernieuwen. Na de migratie naar Grafana vormt dit geen probleem meer en kunnen er een

ongelimeerd aantal databronnen toegevoegd worden die allen iedere vijf seconden hernieuwd kunnen worden.

Vóór de migratie is ook de Excel-sheet geïmporteerd naar een MySQL databank. Nu kan de data snel opgevraagd worden door niet enkel Grafana maar ook door, indien nodig, andere applicaties.

Tenslotte is het terug mogelijk om de dashboards weer te geven op de televisieschermen die verspreid hangen over het kantoor. Op deze manier kunnen de juiste mensen, aan de hand van de dashboards, tijdig ingrijpen waar nodig.

4 Voorstelling stageopdracht 2: In kaart brengen van het ontwikkelingsproces

4.1 Probleemstelling

Level27 is geen bedrijf dat software ontwikkelt maar dat diensten aanbiedt. Toch heeft het bedrijf één voltijdse en één deeltijdse ontwikkelaar in dienst voor het maken van een controlepaneel. Met dit controlepaneel kunnen er allerlei instellingen van domeinnamen, applicaties, *mailboxes* en systemen aangepast worden.

Bij Level27 is er een Confluence wiki opgezet waar alle informatie, instructies en gegevens worden bijgehouden. Deze informatie gaat van een instructie over het opzetten van een cluster tot het weekschema voor de afwas.

Er is dus zeer veel informatie beschikbaar en om deze informatie overzichtelijk te houden is de informatie onderverdeeld in verschillende *spaces*. Van alle *spaces* is de “*Development*” *space* de minst gevulde. Er staat wel wat informatie in, maar over het ontwikkelingsproces zelf staat er bijna niets gedocumenteerd.

Omdat er onvoldoende documentatie voor handen is, is het zeer moeilijk om op een gestructureerde manier te werk te gaan. Er zijn geen duidelijke afspraken gedefinieerd zoals dat wel het geval is voor andere afdelingen binnen Level27. Dit is nodig om de hoogst mogelijke kwaliteit te garanderen aan zowel de klanten als de werknemers van Level27.

Een tweede probleem van geen documentatie te hebben is dat er geen vast ontwikkelingsschema gevolgd kan worden. Er worden Jira *issues* gemaakt voor de ontwikkelaars maar deze zijn vaak onvolledig, waardoor er na een release nog bugs gevonden worden, en er wordt geen tijdsschatting op geplakt, waardoor er deadlines gemist worden. Ook is het moeilijk om het ontwikkelingsteam uit te breiden. Omdat de kennis die gedeeld wordt nergens beschreven staat is het moeilijk om er zeker van te zijn dat de uitleg correct en volledig is.

Om op voorgaande probleemstelling een antwoord te bieden wordt de volgende vraag gesteld: hoe kan de development workflow geoptimaliseerd worden om de kwaliteit te verhogen en de doorlooptijd te verlagen?

4.2 Doelstellingen

Er is geen documentatie omtrent het ontwikkelingsproces van het controlepaneel. Dit maakt het moeilijk om vast te stellen wat de complexiteit van het proces is en waar de struikelblokken liggen.

De basisfunctionaliteit van het controlepaneel wordt eerst kort beschreven om kadering te geven wat er juist ontwikkelt wordt.

Vervolgens wordt er over het ontwikkelingsproces van Level27 gegaan. Dit proces is specifiek voor Level27 en beschrijft de situatie voor Level27. Dit proces begint bij het vaststellen van een bug en eindigt bij het releasen van nieuwe functionaliteit.

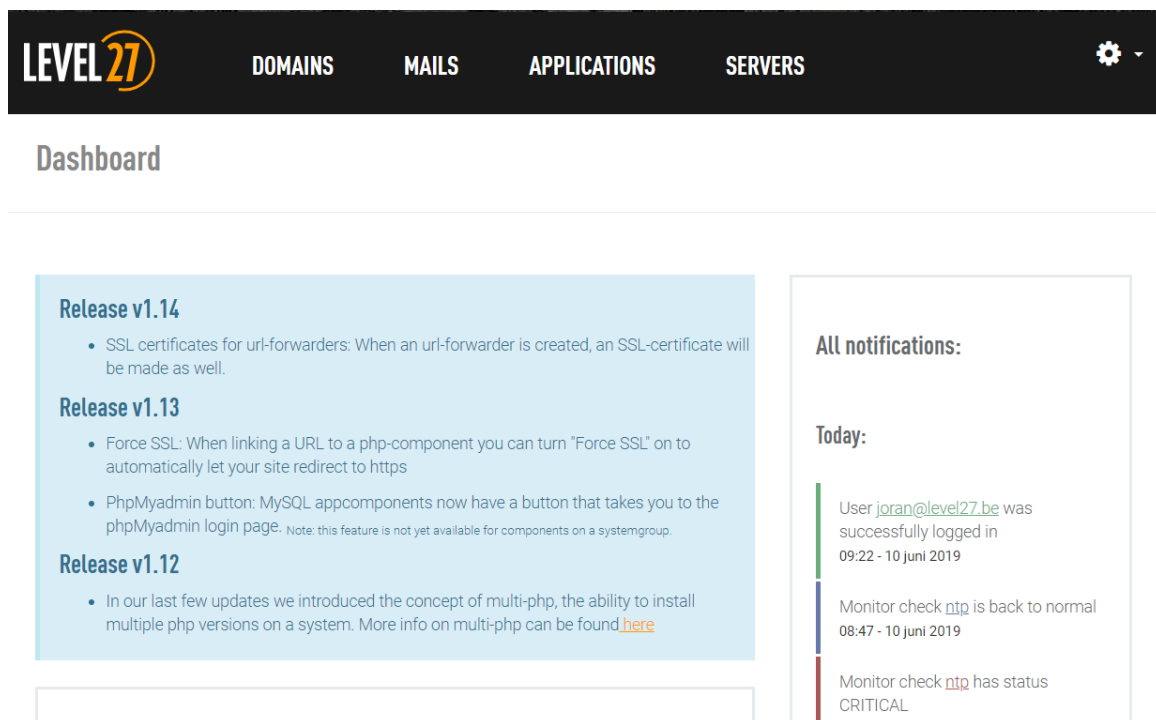
Tenslotte wordt er een conclusie gemaakt over waar Level27 staat en waar er verbeteringen mogelijk zijn.

4.3 De software

Binnen de hosting-wereld is het gebruikelijk dat er gebruik gemaakt wordt van een controlepaneel om domeinnamen, mailboxen en systemen te beheren. Ook Level27 heeft zijn eigen controlepaneel waar allerhanden entiteiten en aangemaakt kunnen worden en instellingen veranderd kunnen worden.

4.3.1 Technologieën

Het controlepaneel heeft binnen Level27 de naam CP4, dat staat voor Controle Paneel 4, gekregen. Deze naam wordt intern gebruikt om enige verwarring tussen het oude controlepaneel, CP3, en CP4 uit te sluiten. Het controlepaneel is geschreven in PHP en maakt gebruik van het Symfony Framework.



Figuur 11: CP4 - dashboard

4.3.2 Functionaliteiten

Het controlepaneel is opgedeeld in vier grote onderdelen: domeinen, *mail*, applicaties en systemen.

4.3.2.1 Domeinen

In het controlepaneel kan de naam van een domeinnaam ingeven worden waarna het systeem een “whois” zal doen van dat domein. WHOIS is een protocol dat gebruikt wordt om allerhande gegevens van een domeinnaam of IP-adres te weten te komen. Een “whois” zal vier gegevens proberen terug te geven als resultaat:

- De informatie van het domein;
- de contactgegevens van de registrant;
- de contactgegevens van de administratie;
- de contactgegevens van het technische contact. [6]

Het kan zijn dat bepaalde gegevens wegens privacy redenen afgeschermd zijn waardoor een “whois” de gegevens niet terug zal kunnen geven. [7]

Een “whois” van “pxl.be” zal veel gegevens teruggeven maar om te bepalen of een domein al bezet is of nog vrij is, is in dit geval het enige relevante gegeven de status van het domein:

```
1. $ whois pxl.be
```

Code 2: voorbeeld WHOIS

Domain: pxl.be

Status: NOT AVAILABLE

Registered: Thu Jan 8 2004

In het antwoord wordt de status meegegeven, welke in dit geval “NOT AVAILABLE” is. Het domein is bezet en kan dus niet geregistreerd worden, tenzij er een transfer code van de vorige eigenaar opgegeven wordt om het domein naar een nieuwe eigenaar te transfereren. Als het domein nog vrij is kan er verder gegaan worden met de registratie van het domein.

The image shows a domain registration interface. At the top, there is a progress bar with three steps: 'Choose your domain' (highlighted in green), 'Configure your domain parameters', and 'Confirm your configuration'. Below this, there is a text input field labeled 'Enter the domain you want to register:' containing the text 'pxlit.be'. To the right of the input field is an orange button with a magnifying glass icon and the text 'Check domain'. Below the input field, a green message box states 'pxlit.be is free and can be registered!'. At the bottom of the interface, there is an orange button labeled 'Next step'.

Figuur 12: CP4 - Domeinregistratie

Als een klant beslist de domeinnaam te registreren zal het controlepaneel de domeinnaam automatisch bij de juiste instantie registreren. De instantie bij wie een domeinnaam geregistreerd wordt hangt af van de extensie van de domeinnaam. Zo moet een domeinnaam met de “be” extensie geregistreerd worden bij dns.be.

Verder kunnen hier de DNS-records en de DNS Time-To-Live (DNS-TTL) van een domein worden ingesteld.

4.3.2.2 Mail

Als een gebruiker beschikt over een domeinnaam, kunnen er één of meerdere *mailboxes* aan dat domein gekoppeld worden. Voor bijvoorbeeld het domein ‘pxl.be’ kan er een mailadres aangemaakt worden zoals ‘info@pxl.be’. Ook *mailforwarders* kunnen ingesteld worden in de mailinstellingen van het controlepaneel.

Status	Username	Name ^	Emails		
OK	mb00053	info@pxlit.be	info@pxlit.be	Edit	Delete
OK	mb00054	koen@pxlit.be	koen@pxlit.be	Edit	Delete
OK	mb00055	pauline@pxlit.be	pauline@pxlit.be info@pxlit.be	Edit	Delete
OK	mb00056	support@pxlit.be	support@pxlit.be	Edit	Delete
OK	mb00057	willem@pxlit.be	willem@pxlit.be	Edit	Delete

Figuur 13: CP4 - mailboxen

4.3.2.3 Applicaties

Er zijn een aantal applicaties die geïnstalleerd kunnen worden op een VPS met een paar klikken. Omdat de configuratie van deze applicaties in een Chef cookbook geconfigureerd staan, hoeft er enkel nog op OK geklikt te worden bij het aanmaken van een applicatie. Voorlopig zijn de volgende applicaties beschikbaar:

- PHP
- MySQL
- SFTP
- MSSQL
- ASP.NET

In de toekomst worden er nog andere applicaties toegevoegd zoals NodeJS, Redis, Memcached en Solr applicaties.

The screenshot displays the CP4 application creation interface. At the top, there are three steps: 'Choose your block type' (highlighted in green), 'Configure your block parameters', and 'Confirm your configuration'. Below the steps, three application options are presented in boxes: 'Mysql server' with a MySQL logo, 'Php' with a PHP logo (highlighted with an orange border), and 'Sftp' with a gear icon. Below these options is a search bar labeled 'Nieuw appcomponent'. Underneath the search bar, the 'Selected location (shared server):' section shows 'web50 (php 7.2)' with a 'remove' button. At the bottom of the interface, there are two buttons: 'Cancel' and 'Next step'.

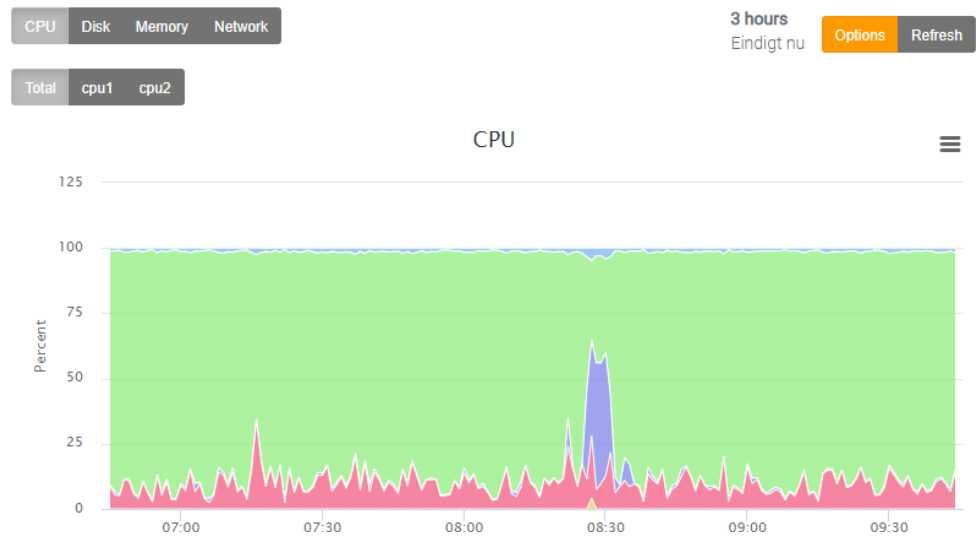
Figuur 14: CP4 - appcomponent aanmaken

4.3.2.4 Systemen

In het systemen onderdeel van het controlepaneel kunnen nieuwe servers opgezet worden of kunnen bestaande servers worden toegevoegd aan een systeemgroep. Bij het aanmaken van een systeem kunnen er een aantal specificaties gekozen worden zoals het aantal processoren, de beschikbare schijfruimte en de grootte van het werkgeheugen.

Afhankelijk van de overeenkomst met de klant worden er door Level27 monitoring checks toegevoegd aan het systeem. Hierdoor wordt er altijd een *duty-officer* op de hoogte gebracht als er zich een probleem voordoet op het systeem.

Het controlepaneel beschikt ook over een aantal statistieken over het systeem die geraadpleegd kunnen worden. Statistieken



Figuur 15: CP4 - systeem statistieken

5 Uitwerking stageopdracht 2

Alles begint met het definiëren van wat er precies gedaan of opgelost moet worden, en eindigt met de oplossing of functionaliteit te introduceren aan de productieomgeving, en dus, de klanten.

5.1 Jira

Binnen Level27 wordt er voor het plannen van *sprints* en managen van *feature requests* en *bugs* gebruik gemaakt van Jira. [6]

Met Jira kunnen *user stories* en *issues* gemaakt worden die ingepland worden in sprints, een gegeven periode om te werken aan bepaalde opdrachten. De taken die behoren tot een bepaalde *story* of *issue* kunnen worden toegekend aan de leden van een ontwikkelingsteam.

Door het onderverdelen van de taken onder de verschillende teamleden kan de sprintmanager ook gemakkelijk de ontwikkelingstijd te monitoren en in te grijpen waar nodig.

Doorheen de beschrijving worden er een aantal structurelementen van Jira aangehaald. Om hier meer duidelijkheid over te scheppen wordt er eerst een overzicht gegeven van de Jira structuur.

Om het volledige potentieel uit Jira te halen raadt Atlassian, de ontwikkelaar van Jira, aan om gebruik te maken van hun epic-story-task structuur.

5.1.1 Epic

Een *Epic* kan vergeleken worden met een container die gevuld kan worden met *user stories* om gemakkelijk de taken die naar een bepaald doel toe werken op een hoog niveau te beheren. Als de *issues* zijn ondergebracht onder de correcte *epic* kan er gesorteerd worden of een overzicht verkregen worden van alle *issues* die te maken hebben met een bepaald doel.

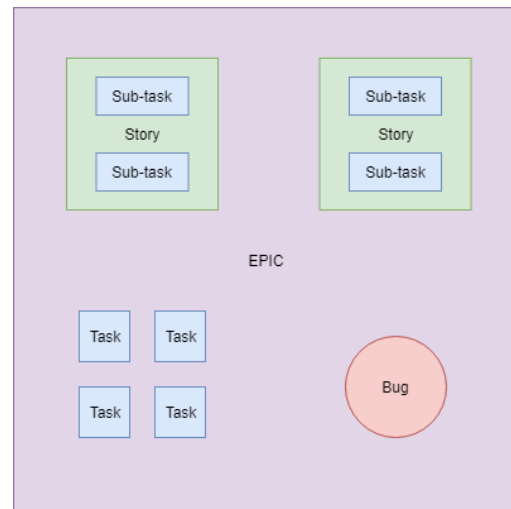
Waar een *story* gewoonlijk wordt afgerond in één sprint, kan het meerdere sprints duren vooraleer een *epic* is afgerond. [7] [8]

5.1.2 Story

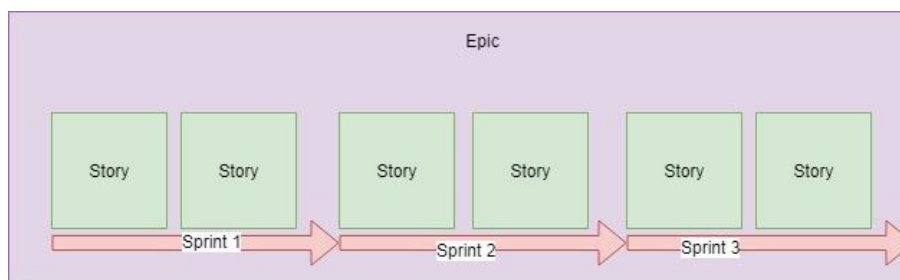
Een *story* is een niveau lager dan een *epic* en beschrijft een bepaalde vereiste in een niet te technische taal zodat duidelijk wordt wat het uiteindelijke doel is dat bereikt wil worden.

Om ervoor te zorgen dat een *story* duidelijk is voor alle *stakeholders* wordt er aangeraden om te werken met *use cases* die zich aan de volgende syntax houden: [8]

As a <type of user>, I want <some goal> so that <some reason>. [9]



Figuur 16: issue types



Figuur 17: Sprints

Door deze syntax aan te houden is het duidelijk wie wat wilt voor welke reden zonder dat er wordt ingegaan op hoe dit zal gebeuren.

Over het algemeen is het de bedoeling dat een *story* wordt opgelost in één sprint, als dit niet het geval is wordt er gekeken om het *issue* op te delen in verschillende *story's* die allemaal onder dezelfde *epic* horen. Het is ook belangrijk dat een *story* niet wordt toegekend aan één lid van het team.

5.1.3 Task

Een taak stelt een enkele opdracht voor die uitgevoerd wordt. Omdat een taak niet onder een bepaalde *story* valt is het dus de bedoeling dat zowel de technische als de niet technische informatie wordt gegeven in de beschrijving. Een *task* kan wel onder een bepaalde *epic* hangen omdat het een losstaande opdracht is maar nog steeds behoort tot een groter geheel. [8]

5.1.4 Sub-Task

Een *sub-task* is vergelijkbaar met een gewone *task* maar behoort tot een *story*. Het is dus een opdracht met als doel het uitwerken van een bepaalde *story*.

In een *story* wordt het doel beschreven maar niet hoe. *Sub-tasks* verdelen het werk van een *story* op in verschillende delen en krijgen een meer technische beschrijving waarin uitgelegd wordt op welke manier het doel bereikt wordt en door wie. [8] [10]

Individuele sub-taken kunnen toegekend worden aan één of meerdere ontwikkelaars.

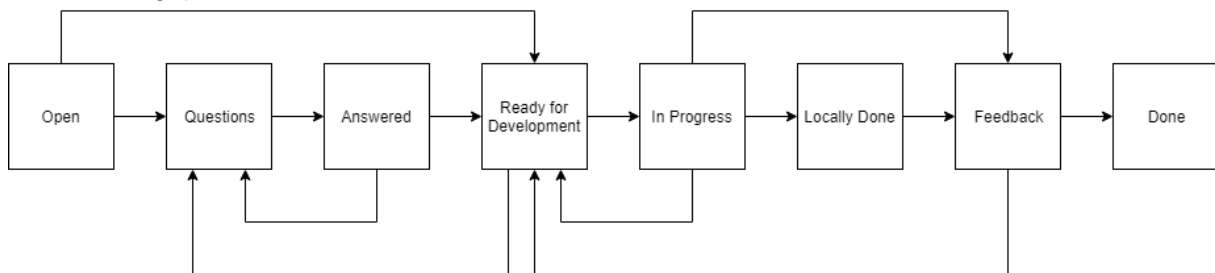
5.1.5 Bug

Een laatste *issue* type is de *bug*. Een *bug* is een foutje in of het falen van een computerprogramma waardoor het onverwachte of foutieve resultaten zal geven als gevolg. De meeste *bugs* ontstaan door een menselijke fout bij het programmeren van de software.

5.2 Sprintbord

Een tweede reeks terminologie die regelmatig terugkomt doorheen de beschrijvingen zijn de verschillende stappen op het sprintbord van Level27.

Het sprint-bord van Level27 bestaat uit acht kolommen om de verschillende fases in het ontwikkelingsproces aan te duiden.



Figuur 18: Sprint bord

Als een nieuwe sprint gestart wordt staan de nieuwe *issues* op Open, dit is de *sprintbacklog*. Als een ontwikkelaar het *issue* volledig heeft gelezen en weet hoe te beginnen wordt het *issue* naar de *Ready for Development* kolom gesleept. Als dit niet het geval is en er vragen zijn wordt de vraag onderaan het *issue* in de vorm van een reactie bijgevoegd en wordt het *issue* naar de kolom *Questions* gesleept. Dit is om aan te geven aan de *productowner* dat er vragen rond het *issue* zijn. Als de *productowner* of een ander teamlid de vragen heeft beantwoord zal het *issue* naar de *Answered* kolom gesleept worden. Dit proces wordt herhaald tot er geen vragen meer zijn en het *issue* naar *Ready for Development* wordt gesleept. Vervolgens neemt een developer het *issue* op, werkt het af en zet het op feedback. Als een *issue* op feedback staat wordt het getest en kan het op done gezet worden. Elk van deze stappen wordt verder toegelicht in de onderstaande hoofdstukken.

5.3 Issue maken

5.3.1 Nieuwe functionaliteit

Sommige processen nemen veel tijd in beslag, maar zouden efficiënter kunnen zijn door het implementeren van een nieuwe *feature*. Op deze manier kan het dagelijks werk van de gebruiker van de software vergemakkelijkt worden. De gebruiker kan in dit geval zowel de klant als een medewerker van Level27 zijn.

Een andere reden voor het aanvragen hiervan kan zijn om extra functionaliteiten aan de eindgebruikers aan te bieden die de concurrentie niet heeft en zo een marktvoordeel te verkrijgen.

Feature requests kunnen in de vorm van een volledig nieuw idee of in de vorm van een aanpassing of verbetering van een bestaande functionaliteit komen.

Iedereen kan nieuwe *features* aanvragen, maar uiteindelijk zal er eerst overlegd moeten worden of de *feature* geïmplementeerd wordt of niet. Als de *feature* te veel tijd of middelen

kost ten opzichte van de voordelen zal deze hoogstwaarschijnlijk niet geaccepteerd worden en dus niet uitgewerkt worden.

5.3.2 Bug rapportering

Als een *bug* gemeld wordt door een klant of medewerker wordt er een *bug* rapport gemaakt, ongeacht of de melding effectief een *bug* is of niet. Iedere meldingen wordt serieus genomen.

Vaak worden gevonden *bugs* ook gewoon genegeerd of in een persoonlijk notitieblok gezet om ze dan later gebundeld te melden. Dit is een slechte manier van werken want dit zorgt ervoor dat de planning van de ontwikkelaars overhoopgehaald wordt door ongepland werk. Als deze onmiddellijk gemeld worden, kunnen ze ook onmiddellijk ingepland worden in een sprint.

5.4 Planning

Om te beslissen welke *issues* in een sprint terecht komen wordt er rekening gehouden met een aantal criteria.

5.4.1 Schatting

Om een min of meer correcte planning op te stellen is het belangrijk dat de ontwikkelaars een tijdsschatting toekennen aan de verschillende *issues*. Zo kan er gekeken worden of het haalbaar is om het *issue* mee op te nemen tijdens de eerstvolgende sprint of dat het beter is om deze op te nemen tijdens een daaropvolgende sprint.

Er zijn twee mogelijkheden om een *issue* in te schatten, oftewel met tijd, oftewel met punten. Omdat Level27 maar twee ontwikkelaars heeft die tevens op afstand werken, verkiest Level27 om te werken met een schatting op basis van tijd.

Bij het inschatten met tijd gaat de ontwikkelaar proberen in te schatten hoeveel tijd er nodig is om het *issue* af te maken. Op deze manier kan de sprintplanner kijken hoeveel tijd er voorhanden is om dan de sprint aan te vullen met *issues* totdat de som van de geschatte uren overeenkomt met de tijd voorhanden. Het is belangrijk dat een *issue* niet op meer dan één dag geschat mag worden. Als dit toch het geval is, wil dit zeggen dat het *issue* moet worden opgedeeld in verschillende *issues* van kleinere groottes.

5.4.2 Prioriteit

Omdat er een bepaalde prioriteit gesteld wordt stelt de *product owner* een *product roadmap* op, een soort kalender waar de grote en belangrijke doelen in worden beschreven die gehaald moeten worden.

Om aan te geven welke *issues* in de *backlog* en sprint het belangrijkste zijn, wordt een prioriteit toegewezen aan alle *issues*. De prioriteiten die ingesteld kunnen worden zijn van de hoogste prioriteit naar de laagste: *Blocker*, *Critical*, *Major*, *Minor* en *Trivial*.

Voor het geven van prioriteiten is er voor de rest niet echt een regel die gevolgd kan worden. Er kan alleen rekening gehouden worden met de *issues* die *blocking* zijn. Als een *issue* ervoor zorgt dat ander werk niet gedaan kan worden of het de voortgang van andere ontwikkelaars blokkeert dan krijgt dat *issue* de *blocking* prioriteit toegewezen.

De overige *issues* krijgen van de *product owner* zelf een prioriteit toegewezen.

5.5 Discussie

Vóór het effectieve programmeren maar na de sprintplanning is er nog een kleine tussenfase waar de ontwikkelaars gaan nadenken over hoe ze een *issue* gaan oplossen en hoe ze tewerk gaan. Vaak duiken er hierdoor nieuwe vragen op die nog niet behandeld zijn tijdens de sprintplanning.

5.6 Ontwikkeling

Vanaf het moment dat er geen vragen meer zijn over een *issue* en alles duidelijk is krijgt het *issue* de *Ready for Development* status. Vanaf dan mag de toegewezen ontwikkelaar beginnen werken aan het *issue* door deze op het Sprintbord naar *In Progress* te slepen.

Het is de bedoeling dat een ontwikkelaar maar aan één *issue* tegelijk werkt, dus normaal heeft maar één *issue* per persoon de status *In Progress*. Omdat de ontwikkelaars altijd op een aparte git-branch werken kunnen ze altijd een *issue* terug de *Ready for Development* status geven, wisselen van branch en een ander *issue* op *In Progress* zetten.

Omdat ontwikkelaars niet altijd kunnen wachten tot alles getest en goedgekeurd is om naar productie te gaan wordt er gebruik gemaakt van een extra status: *Locally done*. Een *issue* krijgt deze status als deze af is op een lokale repository van de ontwikkelaar maar nog niet naar de development omgeving gepusht is. Op deze manier kan bijgehouden worden welke *issues* klaar zijn voor de ontwikkelaar maar nog niet op de testomgeving staan.

5.7 Documentatie

Naast het schrijven van code wordt er ook de nodige documentatie geschreven. Deze komt in twee vormen: API-documentatie en een gebruikershandleiding.

5.7.1 API-documentatie

Het controlepaneel maakt gebruik van een API om verschillende acties uit te voeren. Deze API is ook toegankelijk voor gebruikers om bijvoorbeeld hun eigen platform te maken bovenop het Level27 controlepaneel of om gemakkelijk automatisch servers of applicaties aan te maken.

Omdat zowel werknemers van Level27 als gebruikers op de hoogte moeten zijn van de beschikbare API-calls is het belangrijk dat er een actuele versie van de documentatie beschikbaar is. Deze documentatie beschikt over instructies over hoe de API gebruikt en geïmplementeerd kan worden. [11]

Deze informatie bestaat uit een opsomming van de verschillende *endpoints* met hun respectievelijke *request body* en *response*. Verder staan er ook de verschillende *response-codes* die teruggegeven kunnen worden met hun betekenis en een voorbeeld.

5.7.2 Gebruikershandleiding

Op de supportafdeling van Level27 komen er dagelijks e-mails binnen waarin klanten vragen stellen over het controlepaneel. Omdat dit relatief eenvoudige vragen zijn die regelmatig gesteld worden, verliest het supportteam hier zeer veel tijd mee die anders gespendeerd zou kunnen worden aan grotere of mogelijkere problemen. Om de werkdruk wat te verlichten is er de beslissing genomen om een handleiding te schrijven waarin een aantal processen in het controlepaneel staan uitgelegd.

Bij Level27 wordt er gebruik gemaakt van Sphinx, een Python-applicatie dat documentatie genereert aan de hand van tekst die is uitgeschreven in het reStructuredText formaat. [12]

reStructuredText is een makkelijk leesbaar *plain-text markup syntax* systeem. Hiermee kan *in-line* de opmaak van de tekst meegegeven worden. In Figuur 19: reStructuredText wordt er een voorbeeld van reStructuredText weergegeven. Aan de linkerkant van de figuur wordt weergegeven hoe de tekst wordt geschreven en aan de rechterkant wordt de *output* weergegeven. [13]



Figuur 19: reStructuredText

Als alle documentatie geschreven is in het reStructuredText formaat wordt er een *make* bestand uitgevoerd die Sphinx zijn werk zal laten doen en de juiste HTML-code zal genereren.

Na het genereren van de code kan het HTML-bestand geopend worden om het eindresultaat te bekijken.



Figuur 20: Level27 guide

5.8 Testen

Vóór iedere release worden er twee soorten tests uitgevoerd namelijk automatische API testen door gebruik te maken van het Behat raamwerk en manuele testen.

5.8.1 Manueel testen

De eerste manier van testen die wordt gebruikt bij Level27 is het manueel testen. Dit is de meest tijdrovende manier van testen maar zorgt er wel voor dat de basis functionaliteit getest wordt door echte mensen die weten hoe de software hoort te werken en hoe de software hoort uit te zien.

Belangrijk bij het testen is dat er getest wordt vanuit het administrator standpunt maar ook vanuit het gebruikersstandpunt want beide partijen hebben vaak andere toegangsrechten en een andere GUI.

5.8.2 Automatische API testen

Voor het testen van de API maakt Level27 gebruik van het Behat Framework. Behat is een *open source Behaviour-Driven Development (BDD)* raamwerk voor PHP.

Behat is gebouwd op basis van Cucumber en werkt aan de hand van voorbeelden; voorbeelden die de uitkomst van een scenario beschrijven als er bepaalde acties worden uitgevoerd. [14] [15]

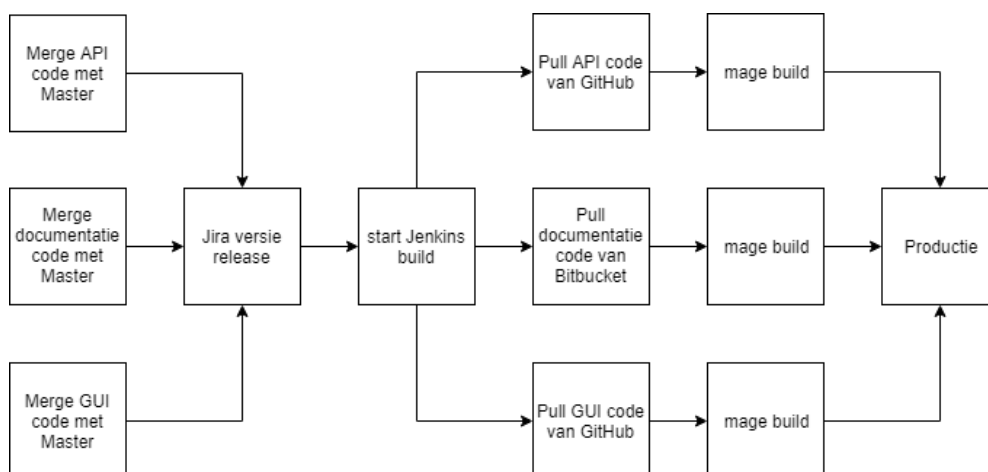
Omdat er wordt gewerkt met een RabbitMQ job systeem werden de automatische API testen opgesplitst in twee grote groepen: testen die gebruik maken van het job systeem en testen die geen gebruik maken van het job systeem.

Dit werd gedaan om tijd te besparen, aangezien de testen die jobs gebruiken ongeveer 4 uur nodig hebben voor een test-run en de testen die geen jobs gebruiken ongeveer 20 minuten duren. Door na een *commit* een *jobless* test-run uit te voeren weten de ontwikkelaars al dat er één van de basisfunctionaliteiten niet meer correct werkt.

Bij het releasen worden zowel de regressietesten met als zonder RabbitMQ uitgevoerd om alle functionaliteit te testen.

5.9 Release

Als alle software door het testproces is geraakt en is goedgekeurd door de *product-owner* kan het release proces gestart worden.



Figuur 21: Release proces

5.9.1 Git release

De eerste stap van het release proces bestaat uit het *mergen* van de code met de *master-branch*. Dit gebeurt nog handmatig maar dit zou geautomatiseerd kunnen worden om het proces nog gestroomlijnder te laten verlopen en de kans op fouten nog verder te minimaliseren.

Github houdt de Jira-release nummering aan zodat als er later bugs gevonden worden er gekeken kan worden bij welke versie de fout werd geïntroduceerd.

5.9.2 Jira release

Als de code gemerget is met de *master-branch* worden ook de Jira issues gemarkeerd als gereleased. Alle issues die tot een bepaalde versie horen krijgen een versienummer toegekend.

Na het releasen kan Jira een aantal rapporten en *release-notes* genereren die dan respectievelijk geanalyseerd of gedeeld kunnen worden met gebruikers.

5.10 Deployment

Bij Level27 gebeurt de deployment van nieuwe code meestal met behulp van Magallanes, een *deployment* tool voor PHP. Magallanes kan code op deployen naar zo veel servers als nodig is en kan voor of na het deployen van nieuwe code nog andere taken uitvoeren zoals bijvoorbeeld het leegmaken van de *cache*.

Om gebruik te maken van Magallanes zijn er maar enkele stappen nodig. De eerste stap is het klonen van de Magallanes *git-repository* en een *composer install* te doen. Vervolgens moet het project geïnitieerd worden, dit moet één maal gebeuren voor ieder nieuw project. Het commando dat gebruikt wordt om het project te initialiseren zal een *.mage* directory aanmaken waarin later logs en configuratie bestanden van de verschillende omgevingen kunnen worden terug gevonden.

Tenslotte moet er een omgeving geïnitieerd worden. Het initialiseren van een omgeving kan zo vaak gedaan worden al nodig is. Zo kan er bijvoorbeeld een omgeving geïnitieerd worden voor een development, productie of *staging* omgeving.

De configuratie bestanden die worden aangemaakt bij het initialiseren van een omgeving kunnen later worden aangepast naargelang het project. In Code 3: voorbeeld Magallanes configuratie bestand wordt een voorbeeld gegeven van een Magallanes configuratie bestand.

```
1. deployment:
2.     user: username
3.     from: ./
4.     to: /var/web/username/deploy
5.     excludes:
6.         - '.mage/*'
7.         - '.git/*'
8.     strategy: rsync
9. releases:
10.    enabled: true
11.    max: 10
12.    symlink: current
13.    directory: releases
14. hosts:
15.     - hostname.domain.be
```

```
16.  tasks:
17.      pre-deploy: null
18.      on-deploy: null
19.      post-release: null
20.      post-deploy: null
```

Code 3: voorbeeld Magallanes configuratie bestand

In het voorbeeld wordt er gedefinieerd dat er vanaf de root gedeployd wordt maar dat de .mage- en de .git-directory overgeslagen moeten worden.

Het effectieve deploy-process kan in Jenkins worden gestart door het shell commando dat wordt weergegeven in Code 4: Magallanes bouwproces starten, toe te voegen als bouwstap. In de laatste parameter van het commando kan er meegegeven worden naar welke omgeving er gebuild moet worden.

```
1. php /var/lib/jenkins/level27/magallanes/vendor/andres-
montanez/magallanes/bin/mage deploy to:dev
```

Code 4: Magallanes bouwproces starten

5.11 Communicatie

5.11.1 Interne communicatie

Bij het releasen van nieuwe functionaliteiten of *bugfixes* is het niet alleen belangrijk dat de klanten, maar ook de medewerkers van Level27 op de hoogte worden gebracht.

Als er een bepaalde *workaround* wordt gebruikt om klanten tijdelijk verder te helpen terwijl de bug wordt opgelost is het belangrijk dat de *support-engineers* op de hoogte worden gebracht zodat ze de applicatie terug kunnen gebruiken zoals het hoort.

Niet allen bij *bugfixes* maar ook bij het introduceren van nieuwe functionaliteiten is het belangrijk dat de *support-engineers* hiervan op de hoogte zijn om eventuele vragen te kunnen beantwoorden. Ook het *sales-team* wordt op de hoogte gebracht van nieuwe functionaliteit zodat deze aan potentiële klanten kunnen worden aangeboden.

5.11.2 Communicatie naar klanten

Als er nieuwe features worden geïntroduceerd is het belangrijk dat klanten hiervan op de hoogte worden gebracht. Dit wordt gedaan aan de hand van *release notes* op de hoofdpagina van het controlepaneel.

5.12 Conclusie

Na het in kaart brengen van het ontwikkelingsproces kan er duidelijk worden vastgesteld wat de huidige situatie van het ontwikkelingsproces is, waar het goed gaat en waar er eventueel nog verbeteringen mogelijk zijn.

zijn er wel een aantal deelprocessen die verbeterd kunnen worden.

5.12.1 Sprint planning

De sprint planningsmeeting bestaat meestal uit twee delen. In deel één van de meeting wordt er over de *backlog* gegaan om een globaal beeld te scheppen van het project. Tijdens deze fase worden er door beide partijen veel vragen gesteld.

Tegen het einde van de meeting wordt er een sprintdoel gekozen: een beschrijving die in één zin zegt wat het doel is. Dit zal later te pas komen wanneer er diepgaandere vragen gesteld worden. Als er bijvoorbeeld de vraag gesteld wordt of een bepaalde functionaliteit gebouwd moet worden kan er gekeken worden of het aansluit op het sprintdoel. Als het werk niet direct aansluit op het sprintdoel, dan wordt deze in de sprint opgenomen.

Tijdens het tweede deel van de vergadering wordt er beslist op welke manier een bepaald issue uitgewerkt wordt. Het team gaat van start met het opsplitsen van de *sprintbacklog* in kleinere werktaken en voegt de taken vervolgens toe aan de sprint zonder over de tijdslijmiet van de sprint te gaan.

Wat belangrijk is bij het plannen van een sprint is definiëren wanneer een *issue* af is. Verschillende mensen hebben een verschillende '*definition of done*' maar door erover te praten en een afspraak te maken is er nadien geen discussie meer mogelijk. [16]

5.12.2 Automatische GUI testen

Hoewel er veel tijd kruipt in het schrijven van deze automatische testen en deze ook veel tijd kosten om te updaten nadat nieuwe software geschreven wordt zijn ze toch een aanwinst in het arsenaal van testen en wordt de kwaliteit van het product ook verhoogd.

Bij Level27 wordt er nog geen gebruik gemaakt van automatische GUI testen zoals dat wel het geval is voor de API. Er wordt door het uitvoeren van automatische API testen heel wat tijd bespaart en de kwaliteit is ook gestegen maar dit zou nog veel verder gedreven kunnen worden door het schrijven van automatische GUI testen.

Bij het schrijven van nieuwe software wordt deze wel getest door een aantal realistische en *edge-case* scenario's na te bootsen in de GUI. Het probleem hiermee is dat er geen regressie testen worden gedaan waardoor er toch regelmatig bugs door de mazen van het net geraken die niet worden opgepikt door de automatische API testen.

Door gebruik te maken van een *test framework* zal het manueel testen van nieuwe functionaliteiten nog moeten gebeuren maar het testen van bestaande functionaliteit zal vanaf dan automatisch gebeuren.

Om automatische GUI-testen te schrijven kan er gebruik gemaakt worden van Selenium, een browser automatisatie tool. [17] Om gebruik te maken van Selenium zal er een nieuw project aangemaakt moeten worden in een van de programmeertalen die door Selenium worden ondersteund. Selenium ondersteund de volgende programmeertalen:

- Java;
- C#;
- Python;
- Ruby;
- PHP;
- Perl;
- Javascript. [18]

In deze programmeertalen kan de Selenium *webdriver* aangesproken worden om instructies aan de browser te geven zoals het klikken op een knop en het invoeren van data in een *textbox*. Op deze manier kan bepaald worden of door het uitvoeren van de actie het gewenste resultaat behaald wordt. [17]

Omdat Level27 gebruik maakt van Behat, een test framework gebouwd op PHP, is PHP ook voor de automatische GUI testen een goede keuze. Een tweede goede keuze is Ruby, dit omdat alle automatisatie gebeurt via Chef, een tool die gebouwd is op Ruby.

5.12.3 Interne kennisdeling

Na iedere release worden alle werknemers op de hoogte gebracht van nieuwe features of bug fixes door middel van release notes die geplaatst worden in een Slack-kanaal. Hoewel deze *release notes* veel informatie bevatten blijkt dit toch niet altijd voldoende te zijn. Vaak worden er een aantal weken na een release toch nog vragen gesteld omdat de *release notes* niet worden gelezen, omdat ze onduidelijk zijn of omdat ze niet genoeg informatie bevatten. Dit zou verholpen kunnen worden door aan *knowledge sharing* te doen. Iedere vrijdag wordt de week afgelopen door een sessie waar iemand iets verteld over een nieuwe technologie of waar bepaalde ervaring wordt gedeeld. Dit moment zou ideaal zijn om een korte sessie te houden waar nieuwe features besproken en getoond worden aan de hand van een korte demo en waar er naderhand vragen gesteld kunnen worden.

5.12.1 Sprint retrospective

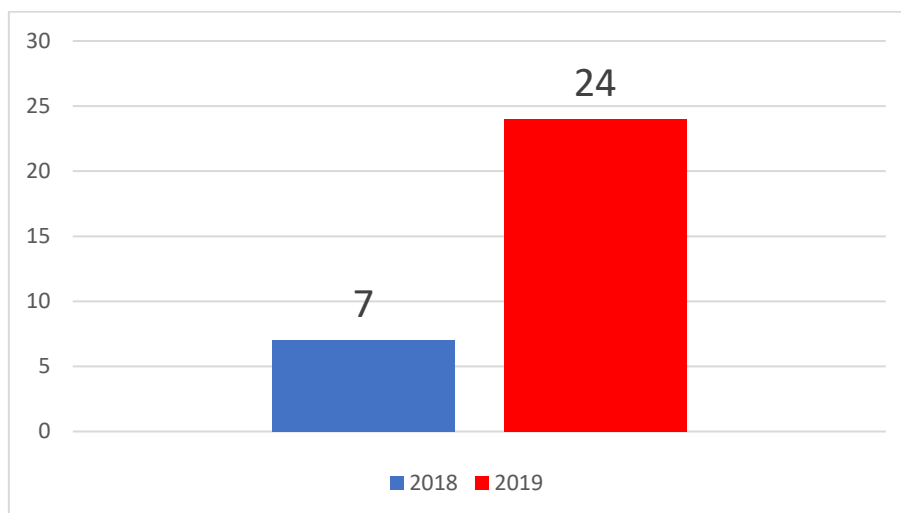
Tijdens een sprint *retrospective* wordt er terug geblikt op de afgelopen sprint en worden er positieve en negatieve punten aangehaald. Een *retrospective* gebeurt dus altijd op het einde van een sprint. Hoewel hier veel voordeel uit gehaald kan worden, gebeurt dit zelden of nooit.

Door de negatieve punten van een sprint te bespreken kan er voorkomen worden dat bepaalde struikelblokken zich niet vaker zullen voordoen en dat er geleerd kan worden uit fouten. Zo kan er gemakkelijk worden bijgestuurd waar nodig.

Niet alleen de negatieve punten zijn van belang tijdens een *retrospective* maar ook de positieve punten. Door positieve punten aan te halen wordt het moreel hoog gehouden en wordt er appreciatie getoond voor de leden van het team.

5.12.1 Aantal releases

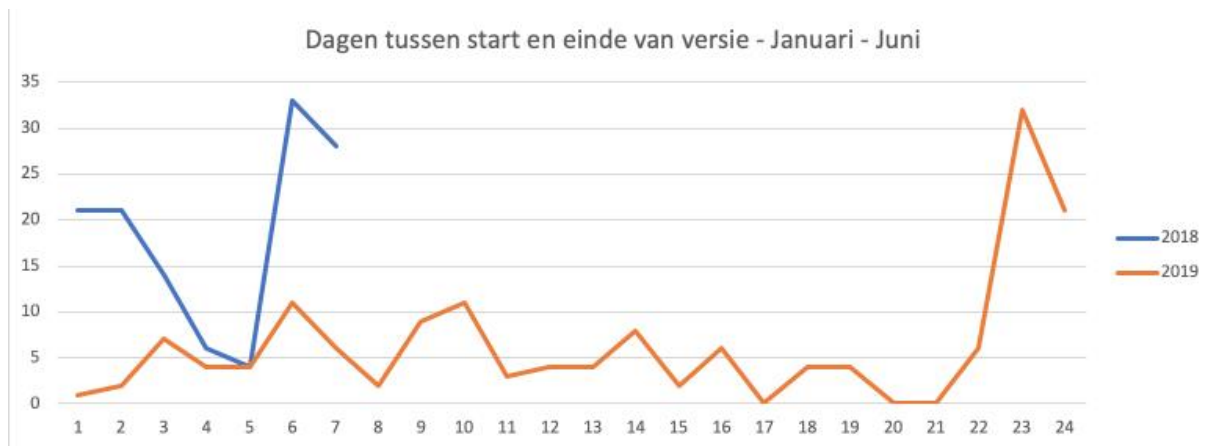
Het aantal software releases is in de loop van vorig jaar sterk toegenomen met van één keer per maand te releasen naar minstens één keer per week te releasen.



Figuur 22: Aantal releases van Januari tot en met Juni

Hoewel dit het doel is, gebeurt dit niet altijd even consistent. In het jaar 2018 zijn er van Januari tot en met Juni 7 releases gebeurd. Over dezelfde periode in 2019 zijn er 24 releases geweest. Dit is een stijging van 343% van het aantal releases en resulteert in een gemiddelde van 4 releases per maand. Hoewel dit een sterke verbetering is, is het aantal

dagen tussen de releases nog niet consistent. Soms zijn er meerdere releases per week en soms is er helemaal geen release. Dit wordt duidelijk weergegeven in Figuur 23: Dagen tussen release Januari tot en met Juni. Het aantal dagen tussen een release fluctueert tussen 0 en 10 dagen gemiddeld. Idealiter zit dit rond de 7 dagen.



Figuur 23: Dagen tussen release Januari tot en met Juni

5.13 Reflectie

Oorspronkelijk had ik maar één stageopdracht, maar mijn hogeschoolpromotor twijfelde of dit wel voor genoeg inhoud zou zorgen. Daarom heb ik in overleg met mijn bedrijfspromotor en hogeschoolpromotor besloten om een tweede opdracht toe te voegen aan mijn eindwerk.

Ik heb lang getwijfeld of dit wel een goed idee was omdat er veel meer werk kwam kijken dan ik initieel gepland had. Eén van mijn eerste struikelblokken was daarom dat ik te laat begonnen ben met het opstellen van een nieuwe planning en in de plaats gewoon ben begonnen. Een goede planning maken zorgt ervoor dat zowel ik als mijn bedrijf niet voor grote hoeveelheden ongepland werk komen te staan. Ook heb ik geleerd dat het soms goed is om een buffer in te calculeren bij het opstellen van een planning zodat ik niet achter kom te liggen als ik tegen struikelblokken aan loop.

Na het uitvoeren van mijn stageopdrachten en onderzoek heb ik zeker en vast een aantal dingen bijgeleerd. Ik heb veel bijgeleerd over ontwikkelingsprocessen en hoe ik deze met een kritisch oog moet bekijken. Veel bedrijven denken dat ze goed bezig zijn maar er is altijd wel iets dat verbeterd kan worden. Maar aan problemen aanduiden heeft niemand iets. Ik heb geleerd dat het in de professionele wereld belangrijk is dat ik niet met problemen kom maar met oplossingen.

Ik ben mijn stagebedrijf heel dankbaar dat ik de kans heb gekregen om over de loop van een volledig jaar het ontwikkelingsproces te onderzoeken, verbeteren en overnemen. Dit was geen evidente taak maar met de hulp van mijn bedrijfspromotor ben ik er toch in geslaagd.

Het is voor mij ook een voldoening als ik bij Level27 binnenwandelen en ik de tv-schermen met de dashboards zie hangen. Het feit dat de werknemers de schermen effectief gebruiken en ik een bijdrage heb geleverd geeft motivatie en goede moed.

Ik ben tijdens mijn stageperiode ook tegen enkele andere struikelblokken aangelopen maar uit de tegenslagen heb ik wel iets geleerd. Zo heb ik bijvoorbeeld het filesysteem van mijn Grafana server stuk gemaakt waardoor het systeem niet meer te redden viel en ik aan een collega heb moeten vragen om een back-up terug te zetten. Door probleem ben ik wat voortgang kwijt geraakt omdat ik nog geen back-up systeem had gemaakt.

Uit dit voorval heb ik een aantal dingen geleerd. Ten eerste is het zinvol om bij de planningsfase al na te denken aan een vorm van back-up. Als ik onmiddellijk had gezorgd voor een back-up was ik geen voortgang kwijtgeraakt.

Een tweede les die ik eruit heb getrokken is dat ik bij het experimenteren met nieuwe software best eerst het advies vraag van iemand die er meer ervaring mee heeft. Het kan nooit kwaad om me eerst te informeren over iets en vragen te stellen.

Tenslotte heb ik geleerd dat ik niet altijd alles in een *Docker-container* moet steken, zeker als ik er geen ervaring mee heb.

II. Onderzoekstopic

Het tweede deel van dit eindwerk beschrijft de onderzoeksopdracht waarop stageopdracht 1 is verder gebouwd. In het eerste deel wordt de onderzoeksvraag gesteld waarna de onderzoeksmethoden worden toegelicht. Tenslotte worden de resultaten van het onderzoek besproken en wordt er een conclusie getrokken.

1 Onderzoeksvraag

Is Power BI, Grafana of Qlik beter dan Klipfolio om KPI-data te visualiseren en te analyseren? Zo ja, welke van de drie is het meest geschikt hiervoor?

Business intelligence is een proces om data te analyseren en te presenteren. Dit het beslissingsproces te vergemakkelijken en om sneller in te spelen op een veranderende omgeving en markt. [19] [20]

Data analytics daarentegen is het analyseren van rauwe data om conclusies te trekken en de toekomst proberen te voorspellen. Het gaat hier dan ook over *real-time* data die eerder gebruikt wordt door experts en wetenschappers in plaats van door analisten. [21]

De volgende tabel kan gebruikt worden om het verschil nog duidelijker te maken:

	<ul style="list-style-type: none">• Business intelligence	<ul style="list-style-type: none">• Data analytics
Beantwoordt de volgende vragen:	<ul style="list-style-type: none">• Hoe?• Wie?• Wat?• Waar?	<ul style="list-style-type: none">• Waarom?• Gaat het vaker gebeuren?• Wat zou er gebeuren als we X aanpassen?
Bevat:	<ul style="list-style-type: none">• Rapporteringen• Dashboards• Onmiddellijk beschikbare query's	<ul style="list-style-type: none">• Statistische analyses• Datamining• Voorspellende modellen• <i>Big Data</i> analyses

Tabel 3: BI of Data analytics [22]

Omdat de data die bij Level27 verzameld wordt, gebruikt gaat worden door medewerkers en het management is het belangrijk dat de focus wordt gelegd op business intelligence en niet op *data analytics*.

Wat ook belangrijk is, is dat Level27 bestaat uit een team van 13 medewerkers en er dus geen volwaardig BI team aanwezig is of zal zijn.

2 Onderzoeksmethode

Het onderzoek bestaat uit vier delen. In ieder deel wordt er een *software* besproken met als eerste Klipfolio. Er wordt besproken wat de voor- en nadelen van iedere *software* zijn en er wordt tenslotte een praktisch voorbeeld uitgewerkt.

Aan de hand van de voor- en nadelen van ieder softwarepakket kan bepaald worden welke in theorie het meest geschikt is als vervanger. Door middel van een praktisch voorbeeld uit te werken kan ook beslist worden welke *software* in de praktijk het meest gebruiksvriendelijk is en de meeste voordelen biedt in het geval van Level27.

Het voorbeeld dat wordt uitgewerkt is het volgende: "Stel een grafiek op om in kaart te brengen wanneer de klantendienst van Level27 de meeste belasting ervaart."

In de *database* zit een tabel die bijhoudt van waar en om hoe laat een klant een chatbericht naar de klantendienst gestuurd heeft. De data wordt uit de *database* gehaald en in een grafiek weergegeven. De data die nodig is voor het voorbeeld kan in MySQL verkregen worden met de volgende *query*:

```
1. SELECT CONCAT(HOUR(startedAt), "h") AS Hour, COUNT(HOUR(startedAt)) AS  
   Amount  
2. FROM chatlio  
3. GROUP BY HOUR(startedAt)  
4. ORDER BY HOUR(startedAt);
```

Code 5: query om voorbeeld uit te werken

Aan de hand van deze data kan er geëvalueerd worden of het voorbeeld correct is uitgewerkt.

Als de vier delen zijn afgerond wordt er een conclusie gemaakt en wordt de gekozen *software* de vervanger van Klipfolio wat in deel 1 verder is uitgewerkt.

Anno 2019 is het altijd mooi meegenomen als er een mobiele versie van een softwarepakket bestaat. Daarom worden de mogelijkheden tot het bekijken van data-analyses *on-the-go* vergeleken.

3 Uitwerking onderzoek

3.1 Klipfolio



Figuur 24: Logo Klipfolio

Er zijn twee dashboard types: de dashboards die zich richten op het analyseren van data en de dashboards die zich richten op het operationele aspect van een onderneming. Klipfolio is een speler die zich bevindt aan de operationele kant van het dashboard speelveld. [23]

Klipfolio focust zich op KPI-data die op regelmatige basis verandert en dus meer opvolging nodig heeft om er direct op in te kunnen spelen. Daarom is het probleem dat Klipfolio probeert op te lossen het feit dat er bij zeer veel applicaties een dashboard functie zit ingebouwd, maar door al die verschillende *dashboards* kan men al vlug het bos door de bomen niet meer zien. Hoe meer dashboards er zijn, hoe minder aandacht eraan geschonken wordt. [23]

Het bedrijf wilt aan de hand van Klipfolio de *dashboards* van de verschillende applicaties bundelen in één applicatie en deze op een hele simplistische manier weergeven en daarmee ook de hoeveelheid data die wordt weergegeven doen verminderen door ervoor te zorgen dat enkel de echt belangrijke data wordt weergegeven. Allan Wille, CEO van Klipfolio inc., zegt het volgende in een *podcast* gehost door b-eye-network:

“Klipfolio is a KPI-dashboard and the ‘K’ in KPI is really important. Only the truly important metrics should appear, eliminating visual noise which lets the user focus on what is truly important”. [23]

Klipfolio werkt op basis van drie begrippen: *datasources*, dashboards en Klips. Voor gegevens visueel voorgesteld kunnen worden moet Klipfolio eerst toegang krijgen tot deze

gegevens. Hiervoor moet er een *datasource*, of gegevensbron, ingesteld worden. Een gegevensbron kan gemaakt worden aan de hand van verschillende diensten zoals een Excel-*sheet*, sociale media applicaties, Google Analytics of een API-call die Klipfolio om de zoveel minuten zal uitvoeren.

Vervolgens moet er een Klip gemaakt worden. Een klip is een *widget* die gegevens zal halen uit een *datasource* om deze visueel voor te stellen. Er zijn verschillende soorten Klips zoals cirkeldiagrammen, staafdiagrammen, lijndiagrammen, enzovoort. [24]

Tenslotte kunnen er verschillende Klips toegevoegd worden aan een dashboard. Dit dashboard kan dan weergegeven worden op een televisiescherm, monitor of op de mobiele applicatie van Klipfolio.

3.1.1 Uitwerking voorbeeld



Figuur 25: Uitwerking voorbeeld in Klipfolio

Omdat Klipfolio een cloudapplicatie is en dus niet zelf gehost wordt kan er geen gebruik gemaakt worden van een *database* connectie zonder een SSH-tunnel te maken. Wat wel kan is een JSON-object maken van de database data door gebruikt te maken van de `JSON_OBJECT` functie van MySQL. [25]

Het JSON-object kan opgeslagen worden en geüpload worden naar Klipfolio om toegang te krijgen tot dezelfde data.

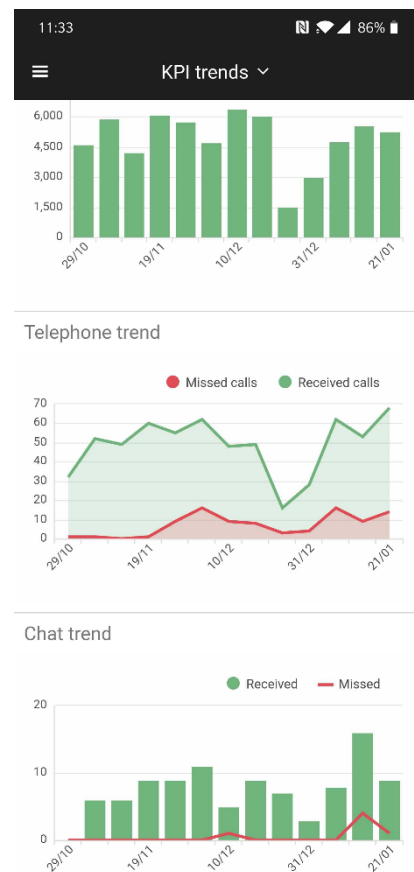
Een probleem wat hier onmiddellijk naar boven komt is het gebrek aan functies. Er zijn een aantal basisfuncties maar er is bijvoorbeeld geen functie om het uur uit een "Date/Time" object te halen. Een omweg hiervoor is een "substring" maken van het date/time object zodat alleen het uur overgehouden wordt. De "groupby" functie werkt wel heel vlot.

3.1.2 Mobiele applicatie

Ondanks de slechte ondersteuning met tv-schermen heeft Klipfolio wel een mobiele applicatie waarop dashboards op een gebruiksvriendelijke manier bekeken kunnen worden, overal ter wereld. Er is zowel een *light* als een *dark-theme* beschikbaar. Alle dashboards die in de browserapplicatie gemaakt zijn kunnen bekeken worden maar kunnen niet aangepast worden op de mobiele applicatie.

3.1.3 Prijs

Klipfolio kost ten tijde van de opstelling van dit document \$29,00 per maand voor twee dashboards, \$49,00 per maand voor 5 dashboards of \$99,00 per maand voor 20 dashboards met een aantal bijkomende voordelen. Level27 heeft een maandelijks abonnement voor 5 dashboards. Er staat geen *hard-limit* op het aantal Klips dat kan worden toegevoegd aan een dashboard maar omdat het vanaf 15 Klips nodig is om te scrollen verdwijnen de voordelen



Figuur 26: Klipfolio mobiele app

van een dashboard. Het doel van een dashboard is namelijk om in één oogopslag een inschatting van de situatie te kunnen maken. Ook is het niet mogelijk om te scrollen als het dashboard wordt weergegeven op een televisiescherm. [26] [27]

3.1.4 Delen van data

Met Klipfolio kan het delen van data op een aantal manieren bereikt worden.

De eerste manier is het downloaden van een Klip als PDF of als JPG, of door enkel de data te downloaden als JSON en deze vervolgens door te sturen.

De tweede manier om data te delen is door de data rechtstreeks te delen met een andere Klipfolio gebruiker of door HTML-code op te vragen om een Klip in te sluiten in een website.

3.2 Power BI



Figuur 27: Power BI logo [34]

Power BI is een *business* analyse tool, ontwikkeld door Microsoft, waar snel inzicht in een organisatie mee verkregen kan worden. Er kunnen verschillende databronnen gebruikt worden om een *dashboard* te maken.

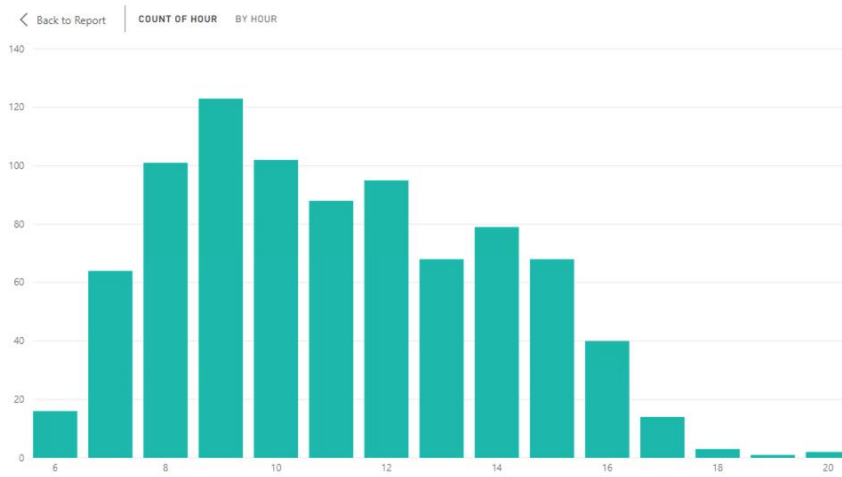
Waar Klipfolio eerder een tool is om data zo simplistisch en minimalistisch mogelijk weer te geven is Power BI meer een analyse tool. Dit komt omdat Power BI aan de achterkant een krachtige data *reader* heeft die de data analyseert, transformeert en combineert en deze vervolgens doorgeeft aan de voorzijde. Ook Power BI, zoals Klipfolio, kan gebruik maken van verschillende diensten om aan data te geraken. [28]

3.2.1 Power BI installeren

Bij het installeren van Power BI worden er twee opties aangeboden: “Power BI” en “Power BI Desktop for Windows”. Er is niet zo heel veel verschil tussen de twee applicaties, enkel dat Power BI Desktop een echte Windowsapplicatie is en Power BI een *cloud service applicatie* in een webbrowser is. [29]

Power BI Desktop kan op de Power BI website gedownload worden en vervolgens geïnstalleerd worden door de installatiewizard te volgen.

3.2.2 Uitwerking voorbeeld



Figuur 28: Uitwerking voorbeeld in Power BI

In Power BI kan er gesorteerd worden op dag, maand, jaar of kwartaal maar niet op uur. Gelukkig kan dit probleem gemakkelijk opgelost worden door gebruik te maken van DAX.

DAX is een collectie van functies, operatoren en constanten die gebruikt kunnen worden om berekeningen uit te voeren. [30]

Door een nieuwe kolom te maken en de functie "HOUR" te gebruiken op de "startedAt" kolom wordt het uur teruggegeven. [31]

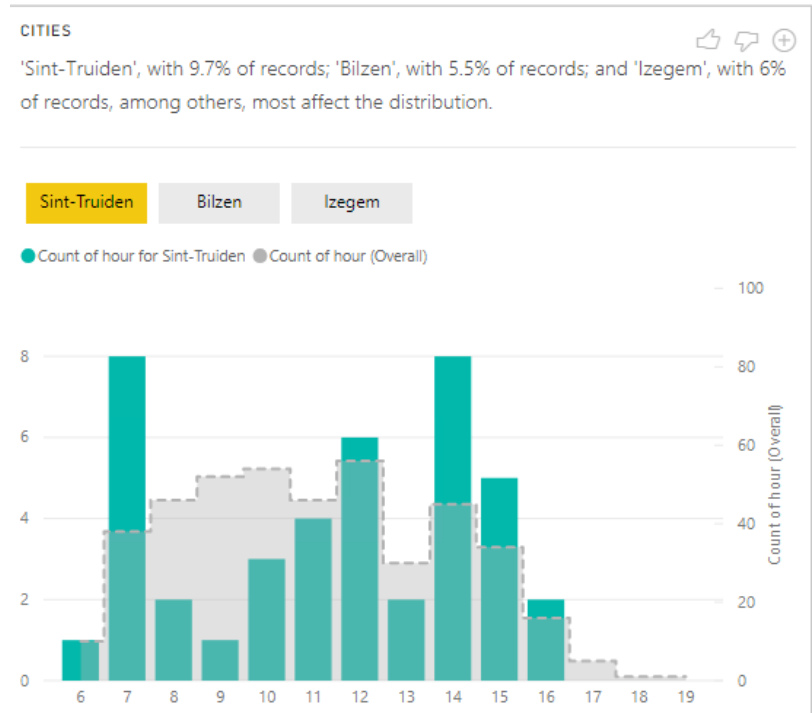
```
1 hour = HOUR('kpi chatlio'[startedAt])
```

Country	startedAt	endedAt	hour
Belgium	3/5/2018 2:00:09 PM	3/5/2018 2:50:26 PM	14
Belgium	6/13/2018 12:09:43 PM	6/13/2018 12:54:40 PM	12
Belgium	3/6/2018 3:36:40 PM	3/7/2018 4:03:08 AM	15
Belgium	3/6/2018 3:29:20 PM	3/6/2018 3:35:47 PM	15
Belgium	4/21/2018 4:29:48 PM	4/22/2018 8:48:57 AM	16
Netherlands	3/15/2018 12:11:35 PM	3/15/2018 12:21:10 PM	12

Figuur 29: HOUR functie in Power BI

Eén van de functionaliteiten die Power BI, in tegenstelling tot Klipfolio, wel heeft is de “Analyse” functie. Door met de rechtermuisknop op een grafiek te klikken wordt de analyse functie beschikbaar. Als er in het voorbeeld op geklikt wordt komt er een venster tevoorschijn wat de data zal gaan analyseren en een conclusie zal proberen te trekken.

Zonder dat er in het voorbeeld wordt gesproken over steden of landen, zal Power BI daar toch rekening mee gaan houden. Zo zal Power BI bijvoorbeeld de conclusie trekken dat de inwoners van Sint-Truiden proportioneel gezien veel vaker om 07:00 en van 14:00 tot 15:00 een chatbericht sturen.



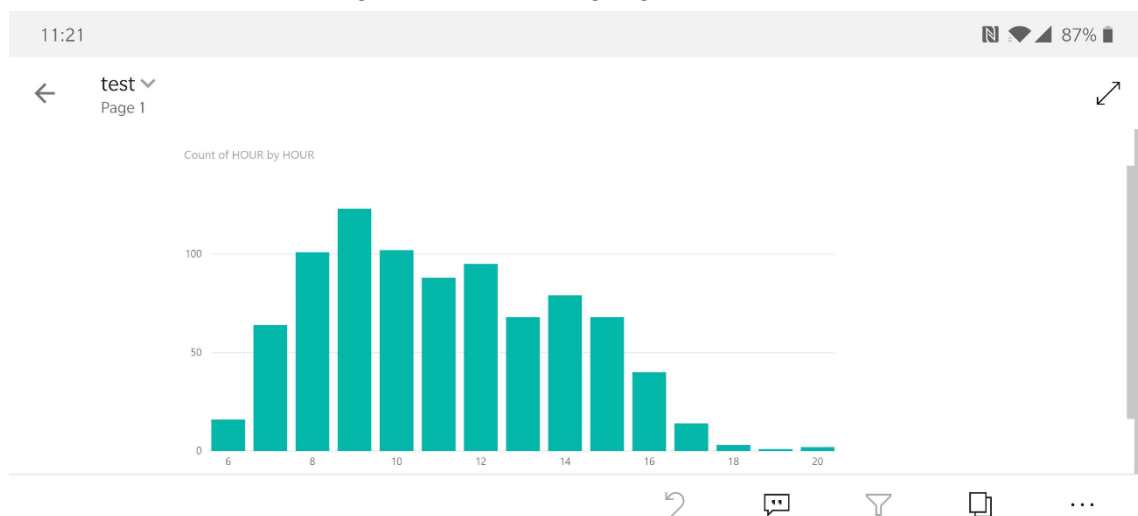
Figuur 30: Analyse functie in Power BI

Op zich is de analyse functie geen vereiste maar is wel goed om te vermelden dat deze functionaliteit er is.

3.2.3 Mobiele applicatie

De mobiele applicatie van Power BI is beschikbaar voor zowel iOS als voor Android toestellen. Met de applicatie kunnen de rapporten die geüpload zijn naar de cloud-app bekeken worden.

Bij ieder rapport kunnen aantekeningen gemaakt worden. Deze aantekeningen kunnen bekeken worden door alle gebruikers die toegang hebben tot het dashboard.



Figuur 31: Power BI mobiele app

3.2.4 Prijs

Power BI Desktop is gratis voor tot één gebruiker. Met deze gratis licentie kunnen er visualisaties gemaakt worden en gepubliceerd worden op Power BI Service, het webplatform van Power BI.

3.2.5 Delen van data

Power BI is heel sterk in het delen van datasets en dashboards. Alle dashboards kunnen worden toegevoegd aan Power BI service, het webplatform van Power BI. Op deze manier kunnen de dashboards gedeeld worden met iedereen die een Power BI account heeft. Op deze manier kunnen de dashboards ook getoond worden op mobiele toestellen waarop de Power BI app geïnstalleerd staat.

Een extra functionaliteit waar Microsoft op in speelt is het rechten systeem van Power BI. Er kan heel uitgebreid ingesteld worden welke gebruikers toegang krijgen tot welke dashboards en of ze lees of schrijfrechten daartoe hebben. Deze rechten kunnen op die manier ook ten alle tijden ontnomen worden. [32]

3.3 Qlik



Figuur 32: Logo Qlik

Qlik biedt een BI-platform aan voor zowel de producenten als de gebruikers van data in een organisatie. Het bedrijf heeft twee hoofdproducten, *QlikView* en *QlikSense*, die beiden gebruik maken van dezelfde *engine* maar die elk een ander doel hebben. De verschillen worden verder besproken in hoofdstuk 383.3.1.

De *Qlik* programma's data, die van verschillende bronnen kan komen, kan geanalyseerd en gevisualiseerd worden door zowel de software zelf als de gebruiker. [33]

Het werkgeheugen is de voornaamste opslagplaats van data voor *QlikSense*. Als een gebruiker een applicatie opvraagt dan zal *QlikSense* de nodige data in het geheugen opslaan. Omdat de data op dat moment opgeslagen zit in het geheugen zullen verdere berekeningen en selecties sneller verlopen. Als de dataset te groot is om in het geheugen op te slaan zal Qlik deze gewoon opvragen bij de bron wanneer nodig. [33] [34] [35]

3.3.1 QlikView of QlikSense

Het verschil tussen *QlikView* en *QlikSense* kan uitgelegd worden aan de hand van de volgende beschrijving van Qlik zelf: "*QlikView is for guided analytics; Qlik Sense is for self-service visualizations.*" [36]

Hiermee wordt bedoeld dat de *QlikView* dashboards worden gemaakt door ontwikkelaars, aan de hand van scripts, die uiteindelijk worden gebruikt door de eindgebruiker. Dit in tegenstelling tot *QlikSense* waar het de bedoeling is dat de eindgebruiker zelf dashboards probeert samen te stellen.

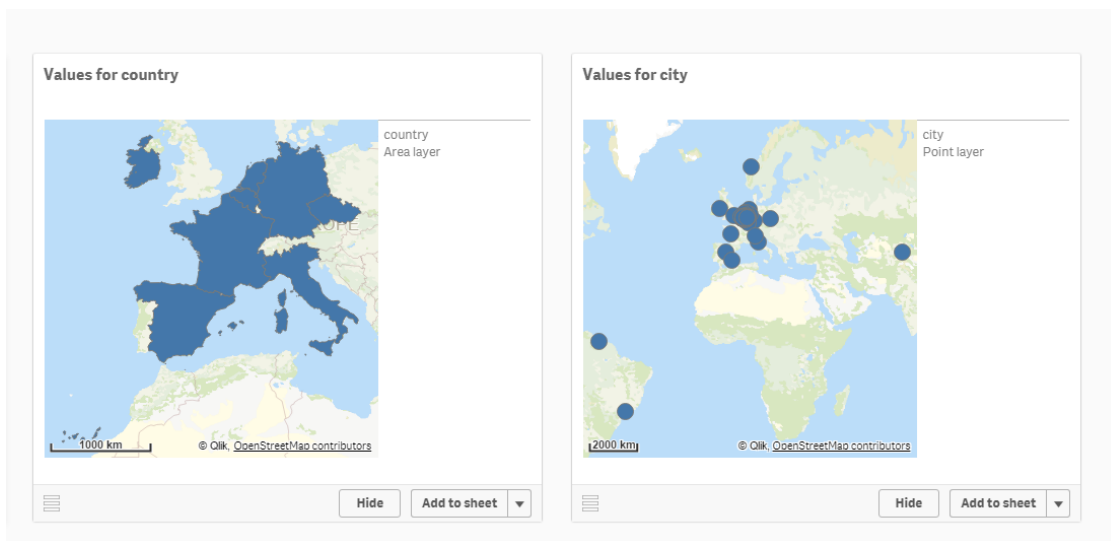
3.3.2 QlikSense installeren

Als *Qlik* gekozen wordt als vervanger van Klipfolio zal het de bedoeling zijn dat iedere werknemer van Level27 deze data kan opvragen en zelf dashboards kan maken naar

behoefte. Om deze reden is het dus aangeraden om *QlikSense* te gebruiken in plaats van *QlikView*. De *QlikSense Windows* desktopapplicatie kan gedownload worden op de website van *Qlik* maar *QlikSense* kan enkel gebruikt worden op desktopversies van *Windows* en dus niet op *Windows servers*.

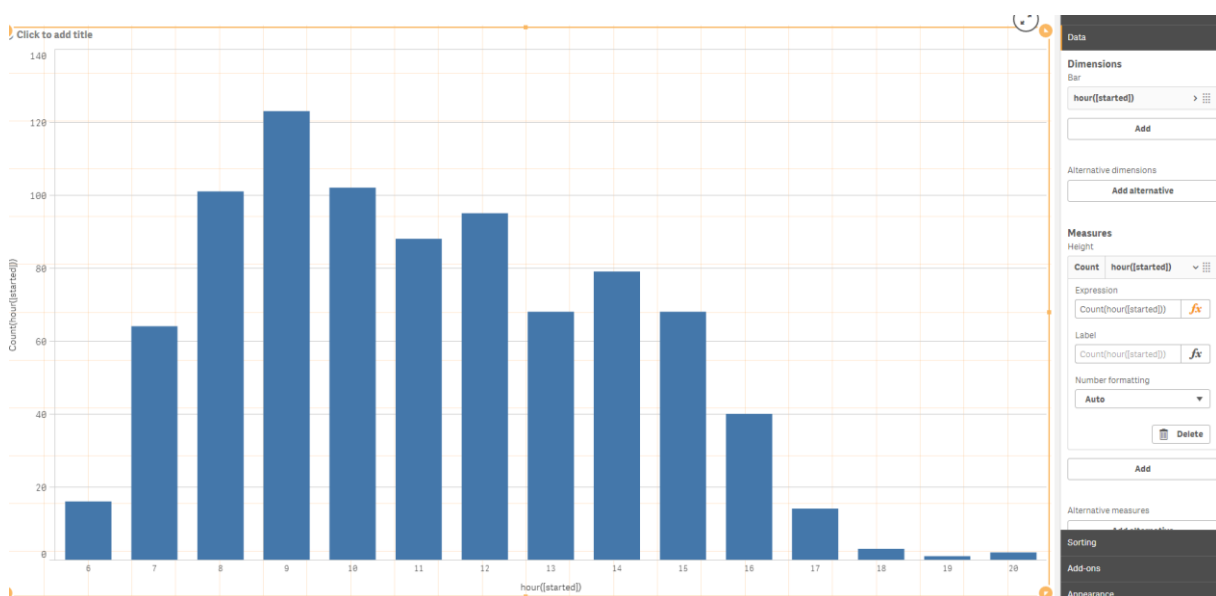
3.3.3 Uitwerking voorbeeld

Als er een databron wordt toegevoegd zal *Qlik* de data analyseren en automatisch een aantal grafieken voorstellen. Zo heeft *QlikSense* bijvoorbeeld zelf vastgesteld dat er geografische data wordt opgeslagen in de database. Binnen het *Insights* menu wordt er dan een suggestie gedaan voor een grafiek waarop de locatie van de afzender van een chatbericht wordt weergegeven op een Belgische- of wereldkaart.



Figuur 33: *Qlik* suggesties

In *Power BI* kan een *date-time* object niet gesorteerd kan worden op uur of seconde, enkel op dagen, maanden, kwartalen of jaren. Dit gaat wel bij *Qlik* omdat er een *HOUR()* functie voorzien is waarmee het uur uit een *date-time* gehaald kan worden.



Figuur 34: *Uitwerking voorbeeld in QlikSense*

3.3.4 Mobiele applicatie

Ook Qlik sense heeft een mobiele applicatie maar die is enkel beschikbaar voor iOS gebruikers waardoor het overgrote deel van de medewerkers van Level27 de applicatie niet kunnen gebruiken.

3.3.5 Prijs

Qlik sense komt in de vorm van een standaard gratis versie en een betalende versie. Deze betalende applicatie, QlikView, heeft meer functionaliteiten maar om een financiële afweging te kunnen maken moet er contact opgenomen worden met een medewerker van de verkoopafdeling.

3.3.6 Delen van data

Met Qlik Sense Desktop kan data op een aantal verschillende manieren gedeeld worden.

De eerste manier is het exporteren van een grafiek naar Excel. Op deze manier wordt er een Excel bestand met de grafiek data gegenereerd welke vervolgens doorgestuurd kan worden.

Een tweede van delen is het exporteren van een grafiek naar PDF of JPG formaat die dan doorgestuurd kan worden.

Een derde manier van delen is het doorsturen van het .qvf bestand waardoor de volledige applicatie gedeeld wordt. Om een dashboard te delen moet de ontvanger van het bestand Qlik Sense Desktop geïnstalleerd hebben en toegang hebben tot dezelfde gegevensbronnen.

Er bestaat in Qlik Sense Desktop geen manier om een link te genereren welke gedeeld kan worden met anderen.

3.4 Grafana



Figuur 35: Logo Grafana [37]

Grafana is een *open-source* tool om maatstaven te analyseren en te visualiseren. In Grafana kunnen dashboards met panelen gemaakt worden. Elk paneel stelt dan een specifieke data reeks voor die wordt weergegeven op een tijd-as.

Grafana biedt ondersteuning voor grafieken, tabellen en *heatmaps* maar er zijn ook veel plug-ins die geïnstalleerd kunnen worden die door de Grafana gemeenschap gemaakt worden. [38]

3.4.1 Grafana installeren

Om Grafana te installeren moet enkel de Grafana *repository* worden toegevoegd en de apt repositories worden geüpdatet. Hierna kan Grafana geïnstalleerd worden door gebruik te maken van het volgende commando:

```
1. $ sudo apt-get install grafana
```

Code 6:Grafana installeren

Als Grafana is geïnstalleerd kan de Grafana-server gestart worden door het commando te gebruiken dat wordt weergegeven in Code 7: Grafana-server starten.

```
1. sudo service grafana-server start
```

Code 7: Grafana-server starten

De Grafana-server zal opgestart worden en zal vervolgens bereikbaar zijn op de standaardpoort van Grafana: 3000. Door te surfen naar het IP-adres van de server gevolgd het poortnummer kan er op de loginpagina van Grafana terecht gekomen worden.

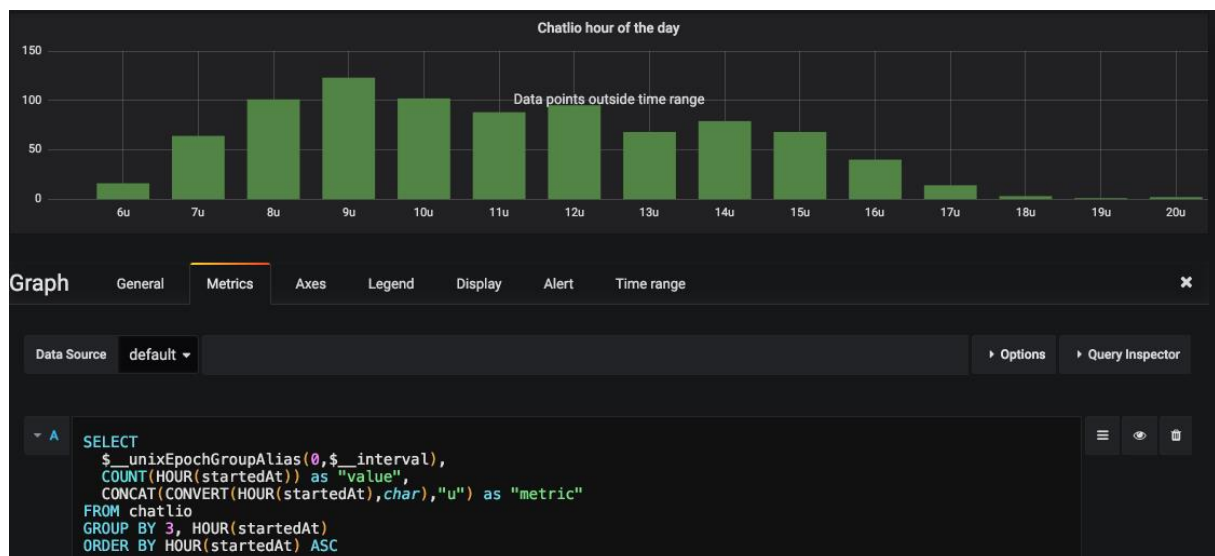
Het standaard wachtwoord en gebruikersnaam van Grafana zijn respectievelijk "admin" en "admin". Grafana zal na het inloggen vragen het wachtwoord te veranderen. [39]



Figuur 36: Loginpagina Grafana

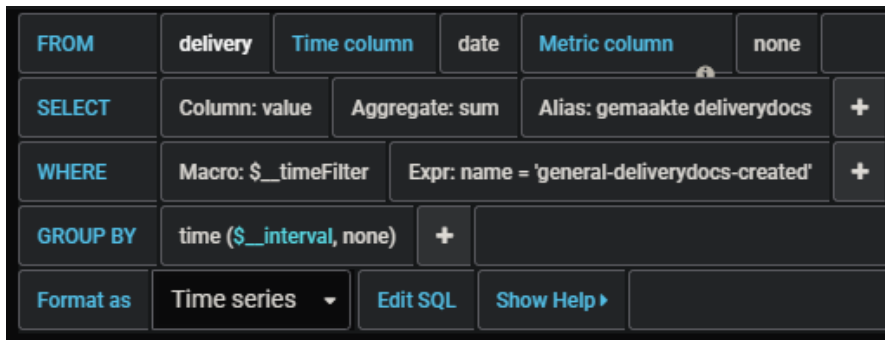
3.4.2 Uitwerking voorbeeld

Zoals beschreven in hoofdstuk 3.4 Grafana bestaat het toevoegen van een nieuwe grafiek uit vier stappen. In de eerste stap wordt er eerst een databron geselecteerd en wordt er een query opgegeven om de gewenste data op te vragen. In de tweede stap kan het type van de grafiek en de opmaak van de grafiek ingesteld worden. In de derde stap wordt er een naam en beschrijving gegeven aan de grafiek. In de laatste stap worden er meldingen ingesteld voor de grafiek.



Figuur 37: Uitwerking voorbeeld in Grafana

Bij Power BI en Qlik Sense bestaat het opmaken van een standaard grafiek meer uit het slepen van blokken. Ook Grafana heeft een query-builder waarmee query's kunnen worden opgebouwd in plaats van een query schrijven.



Figuur 38: Grafana query-builder:

3.4.3 Mobiele applicatie

Hoewel Grafana geen mobiele applicatie heeft, is de browser versie wel *responsive*. Alle elementen worden geschaald en zijn duidelijk leesbaar.

3.4.4 Prijs

Grafana is een *free and open-source software* (FOSS) waardoor alle functies en bijkomende functies die de gemeenschap maakt vrij te gebruiken zijn. Grafana biedt ook een *cloud-solution* aan die \$49,00 per maand kost.

3.4.5 Delen van data

Grafana biedt het delen van data aan op twee verschillende manieren, in de vorm van een deelbare link en in de vorm van een *snapshot*.

Als een deelbare link wordt doorgestuurd naar iemand anders kan die persoon, op voorwaarde dat de persoon een gebruikers account heeft, het dashboard bekijken en aanpassen.

Een snapshot is als het ware een onmiddellijke manier om de huidige status van een dashboard te delen met iemand. Iedereen die de link heeft kan de data zien maar niemand ziet enige achterliggende data of query's of kan aanpassingen maken. Het is dus aangeraden om tweemaal na te denken vooraleer de link gedeeld wordt met iemand.

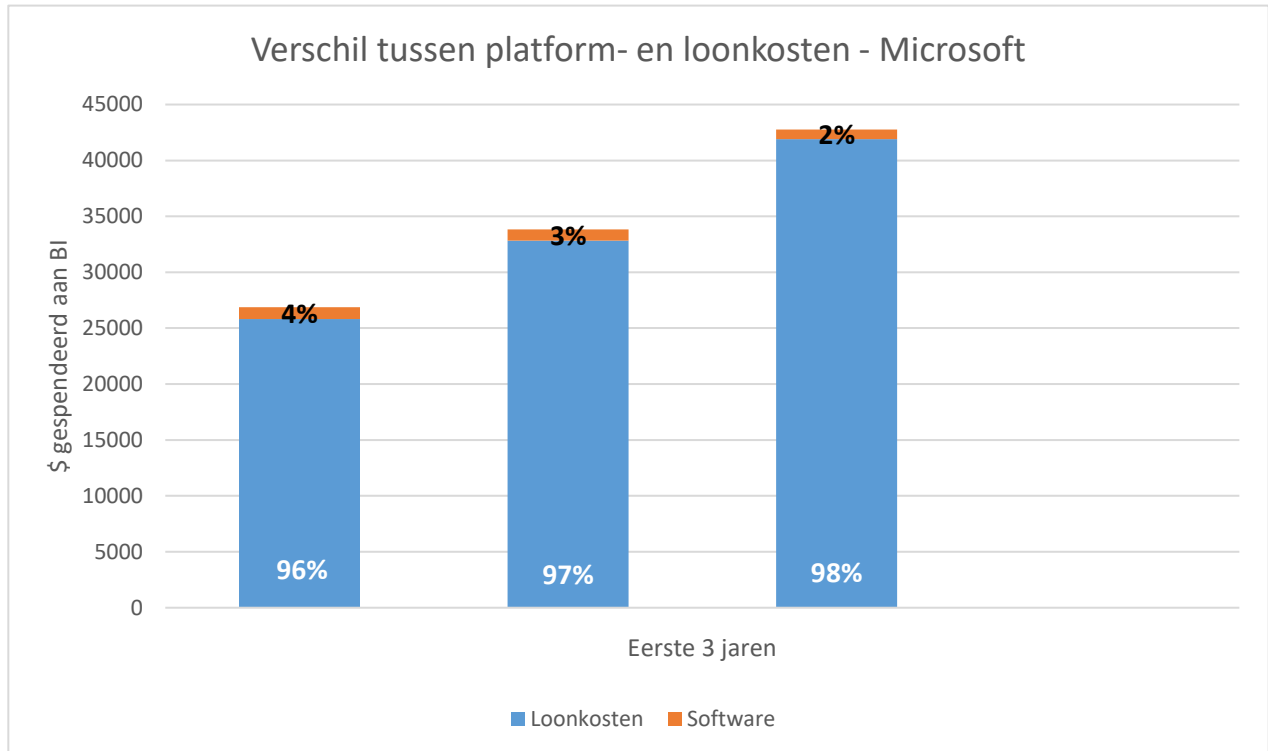


Figuur 39: Grafana browser op mobiel toestel

3.5 Andere vergelijkingen

3.5.1 BI totale kostenplaatje

Tableau, een grote speler op de BI markt, heeft in 2017 een rapport gemaakt over de *total cost of ownership* (TCO) van BI software. Hieruit blijkt dat de werkuren die gestoken worden in BI, over de loop van 3 jaar gemiddeld 22 keer hoger ligt dan de software licentie zelf. [40]



Figuur 40: Jaarlijkse TCO – verschil tussen platform en werk kosten

Door de complexiteit van de software en de analyse die uitgevoerd wordt lopen de kosten van BI vlug op waardoor het zich vaak niet loont voor kleine tot middelgrote ondernemingen om enorme budgetten vrij te maken voor BI investeringen.

3.5.2 On-premise

Omdat Level27 een bedrijf is met hosting als *core business* en beschikt over vier datacentres is het veel goedkoper om de data en de applicaties zelf te hosten en te beheren. Ook is er geen afhankelijkheid meer van externe partijen.

3.5.3 Customer support

Zodra er gebruik gemaakt wordt van de betalende versie van alle drie de software pakketten kan er gebruik gemaakt worden van de klantendienst. Maar Niet enkel de customer support van een product levert een bijdrage tot het gebruikers gemak en de mogelijkheid tot probleem solving maar ook een community of een forum draagt hierbij toe. Uit een enquête uitgevoerd door Synthetix blijkt dat 90% van de consumenten eerst het internet zullen raadplegen vooraleer ze overschakelen naar een telefoontje naar de klantendienst. [41]

Omdat zo een groot aantal gebruikers naar fora gaat om problemen op te lossen zijn er dus veel actieve gebruikers die elkaar kunnen helpen.

Het forum van Power BI heeft een 1000-tal actieve gebruikers, Qlik een 950-tal en Grafana ook een 1000-tal. Hieruit blijkt dat de fora van de drie alternatieven actief gebruikt worden en een geldig alternatief kunnen zijn voor een betalende klantendienst.

3.5.4 Vergelijkmatrix

Uit het voorgaande onderzoek kan Tabel 4: Vergelijkmatrix PowerBI, Qlik en Grafana opgesteld worden. Deze matrix is gebaseerd op de behoeften van Level27 en kan verschillen voor de *requirements* van andere organisaties.

	PowerBI	Qlik sense	Grafana
Complexiteit	✗	✗	✓
Prijs	✓	✓	✓
Mobiele applicatie	✓	✗	✗
Platform onafhankelijkheid	✗	✗	✓
Delen van data	✓	✗	✓
Support	✓	✓	✓
Ongelimiteerd aantal gebruikers	✗	✗	✓

Tabel 4: Vergelijkmatrix PowerBI, Qlik en Grafana

Conclusie

Om de limitaties van Klipfolio uit de weg te gaan is het noodzakelijk dat Level27 overschakelt naar een alternatief. Enkele mogelijke alternatieven zijn Power Bi Qlik Sense en Grafana.

Bij het vergelijken is er rekening gehouden met een aantal functionaliteiten van de verschillende software pakketten. Hoewel deze functionaliteiten geen hard-requirements zijn, is het toch goed om onderzoek te doen naar deze functionaliteiten om een gewogen beslissing te maken bij het migreren.

De reden waarom Grafana gekozen is omdat Grafana aan de onderstaande requirements voldoet:

- Het is licht, gemakkelijk op te zetten en uit te breiden;
- het is intuïtief voor zowel technisch als niet-technisch personeel
- Alle data kan gevisualiseerd worden in één centrale locatie.

Het installeren van Grafana op een server vergt maar één commando wat het zeer gemakkelijk maakt om de software op dezelfde manier op te zetten voor bijvoorbeeld de development-omgeving. Grafana kan geïnstalleerd worden op een standaard Ubuntu server wat het heel toegankelijk maakt voor een webhosting bedrijf.

Het ongelimiteerd aantal dashboards, grafieken, databronnen, hernieuwingen van databronnen en gebruikersaccounts is een enorm voordeel tegenover de alternatieven. Er zijn limitaties opgelegd door zowel Power BI als Qlik, die kunnen worden opgeheven door de aankoop van een duurdere licentie.

Door gebruik te maken van Grafana is het mogelijk om zowel om KPI data, Log data geleverd door de bestaande ELK-stack en monitoring data door middel van Prometheus te visualiseren, allemaal op één locatie. Door alle data op één centrale plaats te verzamelen is het overbodig om verschillende software licenties aan te schaffen en moeten de werknemers van Level27 niet van software wisselen om verschillende dashboards te raadplegen.

Bibliografie

- [1] Level27, „Diensten | Level27,“ Level27, 2019. [Online]. Available: <https://level27.be/diensten>. [Geopend 02 maart 2019].
- [2] ISO, „ISO/IEC 27001 Information security management,“ [Online]. Available: <https://www.iso.org/isoiec-27001-information-security.html>. [Geopend 03 mei 2019].
- [3] „Wat is een key performance indicator?,“ 2019. [Online]. Available: <https://www.marketingtermen.nl/begrip/key-performance-indicator>. [Geopend 02 maart 2019].
- [4] Grafana Labs, „Time Range,“ 2019. [Online]. Available: <http://docs.grafana.org/reference/timerange/>. [Geopend 02 maart 2019].
- [5] B. Mertens, System data collector, Bilzen, Limburg, 2019.
- [6] Atlassian, „Development and Collaboration Software Company | Atlassian,“ [Online]. Available: <https://www.atlassian.com/company>. [Geopend 06 april 2019].
- [7] D. Collins, „Jira: Using Epics vs Components vs Labels - Modus Create,“ Modus, 09 mei 2018. [Online]. Available: <https://moduscreate.com/blog/jira-using-epics-vs-components-vs-labels/>. [Geopend 06 april 2019].
- [8] L. Gildas, „Epic, Story, Task and Bugs - Developer Wiki - Confluence,“ ONAP, 12 juni 2017. [Online]. Available: <https://wiki.onap.org/display/DW/Epic%2C+Story%2C+Task+and+Bugs>. [Geopend 5 april 2019].
- [9] Mountain Goat Software, „User Stories and User Story Examples by Mike Cohn,“ [Online]. Available: <https://www.mountaingoatsoftware.com/agile/user-stories>. [Geopend 05 april 2019].
- [10] Atlassian, „Creating issues and sub-tasks - Atlassian Documentation,“ 05 april 2019. [Online]. Available: <https://confluence.atlassian.com/jirasoftwarecloud/creating-issues-and-sub-tasks-764478439.html>. [Geopend 05 april 2019].
- [11] K. Vasudevan, „What is API Documentation, and Why It Matters?,“ 06 juni 2017. [Online]. Available: <https://swagger.io/blog/api-documentation/what-is-api-documentation-and-why-it-matters/>. [Geopend 17 april 2019].
- [12] D. Procida, „Documentation using Sphinx and ReadTheDocs.org — Don't be afraid to commit 0.3 documentation,“ 2013. [Online]. Available: <https://dont-be-afraid-to-commit.readthedocs.io/en/latest/documentation.html>. [Geopend 25 april 2019].
- [13] D. Goodger en Tibs, „Quick reStructuredText,“ 14 juli 2009. [Online]. Available: <http://docutils.sourceforge.net/docs/user/rst/quickref.html>. [Geopend 25 april 2019].
- [14] K. Kudryashov, „Behat documentation,“ 2016. [Online]. Available: <http://behat.org/en/latest/guides.html>. [Geopend 09 mei 2019].
- [15] Cucumber Limited., „BDD Tutorial : Cucumber,“ [Online]. Available: <https://cucumber.io/docs/guides/bdd-tutorial/>. [Geopend 09 mei 2019].
- [16] D. Huether, „Simple Cheat Sheet to Sprint Planning Meeting - LeadingAgile,“ 09 september 2012. [Online]. Available: <https://www.leadingagile.com/2012/08/simple-cheat-sheet-to-sprint-planning-meeting/>. [Geopend 12 april 2019].
- [17] Selenium, „Selenium - Web Browser Automation,“ [Online]. Available: <https://www.seleniumhq.org/>. [Geopend 04 mei 2019].
- [18] „Selenium WebDriver — Selenium Documentation,“ 19 april 2019. [Online]. Available: https://www.seleniumhq.org/docs/03_webdriver.jsp. [Geopend 04 mei 2019].
- [19] M. Rouse, „What is business intelligence (BI)? - Definition from WhatIs.com,“ searchbusinessanalytics, februari 2019. [Online]. Available: <https://searchbusinessanalytics.techtarget.com/definition/business-intelligence-BI>. [Geopend 04 april 2019].
- [20] Y. Al-Hassani, „Why is Business Intelligence (BI) important to your company?,“ LinkedIn Corporation, 24 juni 2015. [Online]. Available: <https://www.linkedin.com/pulse/why-business-intelligence-bi-important-your-company-yass/>. [Geopend 07 april 2019].

- [21 J. Frankenfield, „Data Analytics,” 25 maart 2019. [Online]. Available:] <https://www.investopedia.com/terms/d/data-analytics.asp>. [Geopend 04 april 2019].
- [22 E. „What are the Business Intelligence Tools and How to Use Them?,” 21 januari 2019. [Online]. Available:] <https://gbksoft.com/blog/what-are-the-business-intelligence-tools-and-how-to-use-them/>. [Geopend 04 april 2019].
- [23 A. Wille, Interviewee, <http://www.b-eye-network.com/listen/11725>. [Interview]. 03 November 2009.]
- [24 Klipfolio inc., „Overview: Working with Klips,” [Online]. Available: <https://support.klipfolio.com/hc/en-us/articles/215545928-Overview-Working-with-Klips>. [Geopend 02 maart 2019].
- [25 Oracle Corporation, „MySQL :: MySQL 5.7 Reference Manual :: 12.17.2 Functions That Create JSON Values,” 2019. [Online]. Available: https://dev.mysql.com/doc/refman/5.7/en/json-creation-functions.html#function_json-object. [Geopend 23 maart 2019].
- [2 Klipfolio Inc., „Dashboard design considerations,” [Online]. Available: <https://support.klipfolio.com/hc/en-us/articles/216183507-Dashboard-design-considerations>. [Geopend 03 mei 2019].
- [27 Klipfolio Inc., „Pricing,” Klipfolio Inc., 2019. [Online]. Available: <https://www.klipfolio.com/pricing-plans>.] [Geopend 02 maart 2019].
- [28 datavibes, „Wat is Microsoft Power BI? - DataVibes,” januari 4. [Online]. Available:] <https://datavibes.nl/wat-microsoft-power-bi/>. [Geopend 4 maart 2019].
- [2 A. Ferrari en M. Russo, *Introducing Microsoft Power BI*, Redmond, Washington: Microsoft Press., 2016.]
- [30 Microsoft, „DAX basics in Power BI Desktop - Power BI,” [Online]. Available:] <https://docs.microsoft.com/en-us/power-bi/desktop-quickstart-learn-dax-basics>. [Geopend 2019 maart 23].
- [31 Microsoft, „HOUR function (DAX) - DAX,” 2019. [Online]. Available: <https://docs.microsoft.com/en-us/dax/hour-function-dax>. [Geopend 24 maart 2019].
- [32 Microsoft inc., „Share Power BI dashboards and reports with coworkers and others - Power BI,” 12 oktober 2018. [Online]. Available: <https://docs.microsoft.com/en-us/power-bi/service-share-dashboards>. [Geopend 26 april 2019].
- [33 QlikTech International AB, „What is QlikView? – QlikView,” november 2018. [Online]. Available:] https://help.qlik.com/en-US/qlikview/November2018/Content/QV_HelpSites/what-is.htm. [Geopend 29 maart 2019].
- [34 QlikTech International AB, „Differences between Direct Discovery and in-memory data – Qlik Sense,” [Online]. Available: https://help.qlik.com/en-US/sense/February2019/Subsystems/Hub/Content/Sense_Hub/DirectDiscovery/differences-between-direct-discovery-in-memory-data.htm. [Geopend 31 maart 2019].
- [35 „QLIKVIEW SERVER MEMORY MANAGEMENT AND CPU UTILIZATION,” september 2012. [Online].] Available: http://www.jansolo.at/QlikView%20Server%20Memory%20Management%20and%20CPU%20Utilization_Technical%20Brief.pdf. [Geopend 30 maart 2019].
- [36 J. Lovell, „Blog - QlikView vs. Qlik Sense,” 02 mei 2018. [Online]. Available:] <https://primaconsulting.co.uk/news/2018/3/26/qlikview-vs-qliksense>. [Geopend 30 maart 2019].
- [37 „Logo Grafana,” [Online]. Available: <https://io.wp.com/www.epmmarshall.com/wp-content/uploads/2018/07/grafana-logo.png?ssl=1>. [Geopend 05 maart 2019].
- [38 D. Tkachenko, „What is Grafana,” 03 oktober 2018. [Online]. Available: <https://www.quora.com/What-is-Grafana>. [Geopend 05 maart 2019].
- [39 Grafana Labs, „Installing on Debian / Ubuntu,” [Online]. Available:] <https://grafana.com/docs/installation/debian/>. [Geopend 26 mei 2019].
- [40 B. Kirwin en P. Brooks, „tco_report_1,” International Institute of IT Economics, Augustus 2017. [Online].] Available: https://www.tableau.com/sites/default/files/pages/tco_report_1.pdf. [Geopend 17 mei 2019].
- [41 A. Ashcroft, „7 out of 10 people think online customer service will dominate over traditional call-centres within a decade,” 13 december 2012. [Online]. Available: <https://synthetix.pr.co/35674-7-out-of-10-people->

think-online-customer-service-will-dominate-over-traditional-call-centres-within-a-decade. [Geopend 23 mei 2019].

- [42 „Docker logo,” 2019. [Online]. Available: [https://cdn.vox-cdn.com/thumbor/RQjqKBzrbO-n-W7rsBtjGVmdHvE=/oxo:792x613/1820x1213/filters:focal\(300x237:426x363\):format\(webp\)/cdn.vox-cdn.com/uploads/chorus_image/image/59850273/Docker_logo_011.o.png](https://cdn.vox-cdn.com/thumbor/RQjqKBzrbO-n-W7rsBtjGVmdHvE=/oxo:792x613/1820x1213/filters:focal(300x237:426x363):format(webp)/cdn.vox-cdn.com/uploads/chorus_image/image/59850273/Docker_logo_011.o.png). [Geopend 04 maart 2019].
- [43 AllesLinux.com, „Docker container hosting · AllesLinux.com,” [Online]. Available: <https://www.alleslinux.com/docker/>. [Geopend 04 maart 2019].
- [44 hexafraction, „What does "sudo apt-get update" do?,” 27 november 2012. [Online]. Available: <https://askubuntu.com/questions/222348/what-does-sudo-apt-get-update-do>. [Geopend 04 maart 2019].
- [45 M. Anderson, „What is a GPG key, and how do I create it? - Quora,” 28 februari 2018. [Online]. Available: <https://www.quora.com/What-is-a-GPG-key-and-how-do-I-create-it>. [Geopend 04 maart 2019].
- [4 „Stack Overflow Developer Survey 2017,” 2017. [Online]. Available: <https://insights.stackoverflow.com/survey/2017#technology>. [Geopend 07 03 2019].
- [47 J. Heeschen, „Install MySQL Server on Ubuntu,” Rackspace US, Inc., 27 februari 2019. [Online]. Available: <https://support.rackspace.com/how-to/installing-mysql-server-on-ubuntu/>. [Geopend 07 maart 2019].
- [48 J. Collins, „Adding a read only MySQL user,” 06 maart 2013. [Online]. Available: <http://www.alphadevx.com/a/388-Adding-a-read-only-MYSQL-user>. [Geopend 07 maart 2019].
- [4 J. Gardeniers, „How can I show user's privileges in MySQL?,” 27 februari 2010. [Online]. Available: <https://serverfault.com/questions/117525/how-can-i-show-users-privileges-in-mysql>. [Geopend 07 maart 2019].
- [50 Oracle Corporation, „MySQL :: MySQL 8.0 Reference Manual :: 1.3.1 What is MySQL?,” [Online]. Available: <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>. [Geopend 07 maart 2019].
- [51 InterServer.net, „WHOIS Lookup Explained - Interserver Tips,” [Online]. Available: <https://www.interserver.net/tips/kb/whois-lookup-explained/>. [Geopend 10 maart 2019].
- [52 „ISO 8601 Date and time format,” [Online]. Available: <https://www.iso.org/iso-8601-date-and-time-format.html>. [Geopend 17 maart 2019].
- [53 Klipfolio Inc., „Klipfolio functions,” [Online]. Available: <https://support.klipfolio.com/hc/en-us/articles/360011416893-Klipfolio-functions>. [Geopend 23 maart 2019].
- [54 senturus, „Senturus.com,” 2018 oktober 05. [Online]. Available: <https://senturus.com/wp-content/uploads/2016/05/Power-BI.jpg>. [Geopend 2019 maart 22].
- [55 D. Tkachenko, „Grafana as Yet Another Tool for Technical Monitoring of Software Products We Build,” logicify, 21 september 2018. [Online]. Available: <https://www.logicify.com/en/blog/grafana-as-a-yet-another-tool-for-technical-monitoring-of-software-products-we-build/>. [Geopend 24 maart 2019].
- [56 Chef Software Inc., „Chef: Deploy new code faster and more frequently. Automate infrastructure and applications | Chef,” [Online]. Available: <https://www.chef.io>. [Geopend 31 maart 2019].
- [57 Gartner, Inc., „Magic Quadrant Research Methodology,” [Online]. Available: <https://www.gartner.com/en/research/methodologies/magic-quadrants-research>. [Geopend 03 april 2019].
- [58 M. Cohn, „Agile Teams Estimate Product Backlogs & Sprint Backlogs Differently,” 11 september 2018. [Online]. Available: <https://www.mountangoatsoftware.com/blog/why-agile-teams-should-estimate-at-two-different-levels>. [Geopend 12 april 2019].
- [59 J. Fix, „Sprint Estimation Pointing Scales | Planning Poker,” [Online]. Available: <https://www.planningpoker.com/blog/sprint-estimation-pointing-scales/>. [Geopend 13 april 2019].
- [6 V. Paget, „Features vs Functionality: How to accurately compare software systems (Free Template) - o Boardingware Blog,” 19 juni 2017. [Online]. Available: <https://blog.boardingware.com/features-vs-functionality/>. [Geopend 11 april 2019].
- [61 M. d'Itri, „whois (1) - client for the whois directory service,” 20 december 2009. [Online]. Available: <https://www.unix.com/man-page/linux/1/whois/>. [Geopend 03 mei 2019].

Bijlage

Bijlage A: High level

A. High level

