

**Telemetric Gateway Monitoring - IBM CIC**

Maciej Dzialoszynski, Sebastian Grünewald, David Jovanovic, Jordi Rieder

Sprint Abschlussbericht

Sprintdokumentation

06.03.2019 – 08.05.2019

Inhaltsverzeichnis

[Änderungsverzeichnis 2](#_Toc4775924)

[1 Sprintbericht 3](#_Toc4775925)

[1.1 4390 IBM Cloud 3](#_Toc4775926)

[1.2 4391 Frontend – Wireframes 3](#_Toc4775927)

[1.3 4392 Backend – Setup 3](#_Toc4775928)

[1.4 4393 Backend – GW – Cloud 3](#_Toc4775929)

[1.5 4413 Frontend - Setup 3](#_Toc4775930)

[1.6 Sprint Burndown Chart 7](#_Toc4775931)

[1.7 Product Burndown Chart 7](#_Toc4775932)

[1.8 Produktivität 7](#_Toc4775933)

# 

# Änderungsverzeichnis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Autor | QS | Datum | Status |
| 1.0 | David Jovanovic | Sebastian Grünewald | 29.03.2019 | erledigt |
| 2.0 | Maciej Dzialoszynski |  | 07.05.2019 |  |

# **1 Sprintbericht**

Wir haben unser Projekt in Userstories eingeteilt, welche aus mehreren Sub-Items bestehen. Die Userstories wurden den einzelnen Sprints zugewiesen.

## 4390 IBM Cloud

Hier werden grundlegende Konfigurationen auf der IBM-Cloud durchgeführt, welche notwendig sind, um das Projekt zu realisieren. Mit dem Auftraggeber wurde bereits eine Softwarearchitektur erarbeitet, welche es umzusetzen gilt. Weiters werden die Verknüpfungen zwischen den Microservices hergestellt.

## 4391 Frontend – Wireframes

Es werden Wireframes erstellt, welche die GUI repräsentieren. Die intuitive Bedienung wird angestrebt.

## 4392 Backend – Setup

Hier werden grundlegende Strukturen des Backends aufgebaut. Es müssen alle notwendigen Module installiert, verknüpft und lauffähig gemacht werden. Weiters soll es durch diese Userstory, den Entwicklern ermöglicht werden, direkt mit der Implementierung zu beginnen.

## 4393 Backend – GW – Cloud

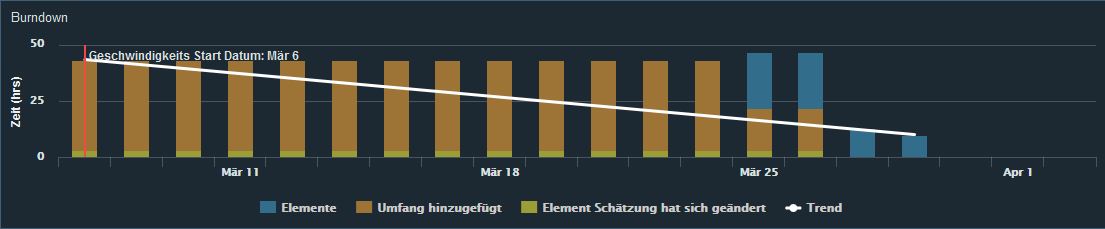
Es wird möglich sein Daten vom Gateway auf die Cloud zu übertragen, um damit in weiteren Schritten arbeiten zu können.

## 4413 Frontend – Setup

Hier werden grundlegende Strukturen des Frontends aufgebaut. Es müssen alle notwendigen Module installiert, verknüpft und lauffähig gemacht werden. Weiters soll es durch diese Userstory, den Entwicklern ermöglicht werden, direkt mit der Implementierung zu beginnen.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Name** | **Zugewiesen** | **Schätzung** | **Status** | **Akzeptanzkriterium** |
| **4390** | **IBM Cloud** |  | **3.5 hrs** | **In Progress** |  |
| 4394 | Anlegen einer IoT-Plattform | Maciej Dzialoszynski | 0.5 hrs | **Completed** | Die IoT-Plattform ist über die IBM-Cloud-Accounts der Teammitglieder zugänglich. |
| 4395 | Einrichten einer Frontend-Cloud-Foundary-App | Maciej Dzialoszynski | 0.5 hrs | **Completed** | Die Sample-App ist lauffähig. |
| 4396 | Einrichten einer Backend-Cloud-Foundary-App | David Jovanovic | 0.5 hrs | **Completed** | Die Sample-App ist lauffähig. |
| 4397 | Anlegen von Git-Repository für Frontend | Maciej Dzialoszynski | 0.5 hrs | **Completed** | Git-Repository ist klon- und veränderbar. |
| 4398 | Anlegen von Git-Repository für Backend | Jordi Rieder | 0.5 hrs | **Completed** | Git-Repository ist klon- und veränderbar. |
| 4399 | Kommunikation zwischen Backend und Datenbank (Cloudant) | Jordi Rieder | 0.5 hrs | **Open** | Verbindung ist hergestellt |
| 4400 | Kommunikation zwischen Backend und Frontend | Jordi Rieder | 0.5 hrs | **Open** | Verbindung ist hergestellt |
| **4391** | **Frontend - Wireframes** |  | **10 hrs** | **Completed** |  |
| 4405 | Login/Registrierung | Maciej Dzialoszynski | 1 hrs | **Completed** | Das Wireframe wurde von Papier auf das Mockup-Tool Balsamiq übertragen. |
| 4406 | Gerät-Registrierung | Sebastian Grünewald | 3 hrs | **Completed** | Das Wireframe wurde von Papier auf das Mockup-Tool Balsamiq übertragen. |
| 4407 | Gerät-Liste | Sebastian Grünewald | 3 hrs | **Completed** | Das Wireframe wurde von Papier auf das Mockup-Tool Balsamiq übertragen. |
| 4408 | Gerät-Details | Sebastian Grünewald | 3 hrs | **Completed** | Das Wireframe wurde von Papier auf das Mockup-Tool Balsamiq übertragen |
| **4392** | **Backend - Setup** |  | **3 hrs** | **In Progress** |  |
| 4409 | Ordnerstruktur | David Jovanovic | 1 hrs | **Completed** | Jedes Modul und jede zusätzliche Technologie hat einen eigenen Ordner. |
| 4410 | Node-Arbeitsumgebung aufsetzen | David Jovanovic | 2 hrs | **Completed** | Die Mitglieder haben Zugriff auf die Node.js App auf der Cloud und können ihre Änderungen pushen. Die neuste Version sollte dann immer verfügbar sein über den Link auf der Cloud |
| **4393** | **Backend - GW-Cloud** |  | **10 hrs** | **In Progress** |  |
| 4411 | GW-Datenempfang | David Jovanovic | 5 hrs | **Open** | Die korrekten Daten werden empfangen. |
| 4412 | Datenfilterung | Jordi Rieder | 5 hrs | **In Progress** | Empfang der gewünschten Daten |
| **4413** | **Frontend - Setup** |  | **7 hrs** | **In Progress** |  |
| 4414 | Ordnerstruktur | Sebastian Grünewald | 1 hrs | **Completed** | Jede Komponente und jede zusätzliche Technologie hat einen eigenen Ordner. |
| 4415 | React-App mit der Node-App verbinden | Sebastian Grünewald | 1 hrs | **Completed** | Datenaustausch zwischen jenen Apps möglich. |
| 4416 | Routen festlegen | Maciej Dzialoszynski | 4 hrs | **In Progress** | Navigation zwischen Login/Reg-Seite und Hauptseite funktioniert und die Navigation zwischen einzelnen Komponenten funktioniert. |
| 4417 | Sass aufsetzen | Maciej Dzialoszynski | 1 hrs | **Completed** | Sass in Projekt eingebunden und funktionstüchtig. |

## Sprint-Burndown-Chart



# **2 Sprintbericht 03.04 – 20.04**

2.1 Frontend – Setup – PSP: 3,2

Arbeitspaketname

Beschreibung

Lorem ipsum

Akzeptanz

Lorem ipsum

Durchführung

Lorem ipsum

Dokumentation

Lorem ipsum

# **3 Sprintbericht 20.04 – 08.05**

## 3.1 Frontend – Device Liste – PSP: 3,4

## 3.2 Frontend – Device Details – PSP: 3,6

## 3.3 Frontend – Api Calls – PSP: 3,7

## 3.4 Frontend – Navbar – PSP: 3,8

## Product-Burndown-Chart

## Produktivität