# Teilnehmer/innen des Teams:

|  |  |
| --- | --- |
| Klasse:  AP17c | Team:  Joel Rauber & Luca Schweizer |

# Anforderungsdefinition (Meilenstein A)

|  |  |
| --- | --- |
| „USB-Device-Notifier (Geräte-Manager)“ | |
| **Fachlicher Inhalt:**  (Allgemeine Beschreibung) | Nutzen: Mit dem Programm sollen Geräte, die per USB eingesteckt werden, automatisch erkannt werden. Diese werden dann durch ein Pop-Up angezeigt. In Haupt-Tab sieht man die verschiedenen Eigenschaften des Gerätes. Unter anderem kann man der Namen anpassen und den Gerätetyp ändern. Im Treiber-Tab werden die aktuellen Treiber und deren Eigenschaften des Gerätes angezeigt. Man findet dort auch einen Button, um die Treiber zu aktualisieren. Im letzten Tab «Ereignisse» sieht man die History des Gerätes. Also wann es installiert wurde, aktiviert wurde usw.**Automation:**  Das Programm verwaltet die USB-Geräte, die in den Laptop/PC eingesteckt werden.  **Details:**   * Konfiguration: Es ist keine Konfiguration nötig. * Integration: Das Programm muss gestartet werden und läuft im Hintergrund, bis ein USB-Gerät erkannt wird * Administration: Es sind keine zusätzlichen Rechte nötig. * Sicherheitsaspekte: keine speziellen Sicherheitsaspekte   **Ein Bild, das Text, Whiteboard enthält.  Automatisch generierte Beschreibung**  Haupt-Tab  Ein Bild, das Text, Whiteboard enthält.  Automatisch generierte Beschreibung  Treiber-Tab  **Erkenntnisse aus der Machbarkeitsabklärung in Windows Powershell:**  Wir haben überprüft ob Treiber über Powershell aktualisiert werden können. |
| **MUSS**  **Kriterien:**  (Konkrete Features, die umzusetzen sind) | **Folgende Features sollen implementiert werden, um einen produktiven Ablauf sicherzustellen:**   * Geräteeigenschaften anschauen * Gerätenamen ändern * Gerätetyp ändern * Treiber anschauen * Treiber aktualisieren * Ereignisse anschauen |

|  |  |
| --- | --- |
| **KANN**  **Kriterien:**  (Konkrete Features, die optional sind) | **Folgende Features können zusätzlich implementiert werden: (Varianten, Kreativität)**   * Gerät deaktivieren * Gerät deinstallieren   **Ein Bild, das Text, Whiteboard enthält.  Automatisch generierte Beschreibung**  Ereignisse-Tab |

## Planung Meilensteine (LB1 / LB2)

|  |  |  |  |
| --- | --- | --- | --- |
| *MS* | *Tätigkeit / Abgabe* | *Soll-Datum* | *Ist-Datum* |
| A | ProjektstartTeam Bildung, Kollaborationsplattform, GitHub Repos \*, LehrerzugangWahl / Ausarbeitung der Anforderungsdefinition Abnahme Anforderungsdefinition durch Lehrperson | 06.03.2020 | 06.03.2020 |
| B | Teamaufgabe 1:Abgabe: Lösungsdesign  (Funktionsmodell / GUI / PAP / Storyboard) | 20.03.2020 | 20.03.2020 |
| B2 | Teamaufgabe 2: (Nur LB2)Abgabe: Testvorschrift und Testfälle | 20.03.2020 | 20.03.2020 |
| C | Einzelaufgabe 2 (LB1) / 3 (LB2):Abgabe Programmcode und DokumentationFachgespräch Projektabnahme | 07.05.2020 | 07.05.2020 |
| C2 | Einzelaufgabe 4: (Nur LB2)Abgabe: Ausgefüllter Systemtest | 07.05.2020 | 07.05.2020 |

# Lösungsdesign (Meilenstein B: Teamaufgabe 1)

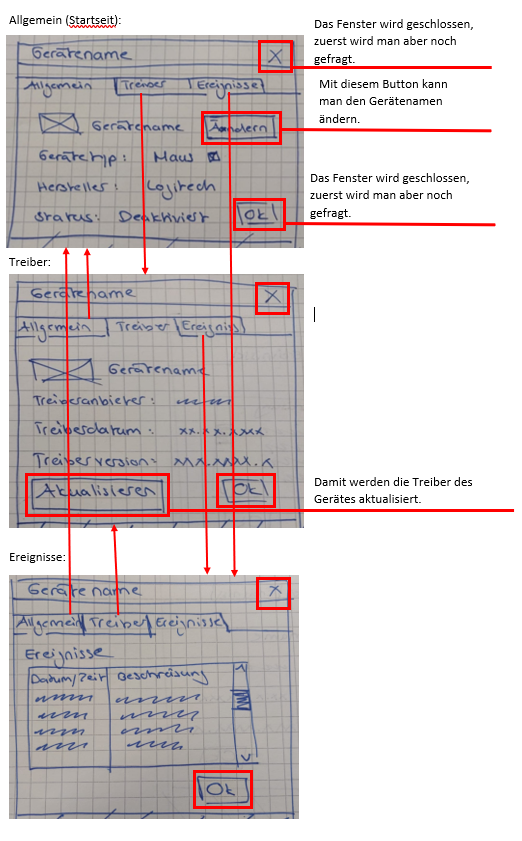
Anhand der Analyse wurde folgendes Lösungsdesign entworfen:

## Schematische Darstellung der Funktionalität, sog. Funktionsmodell

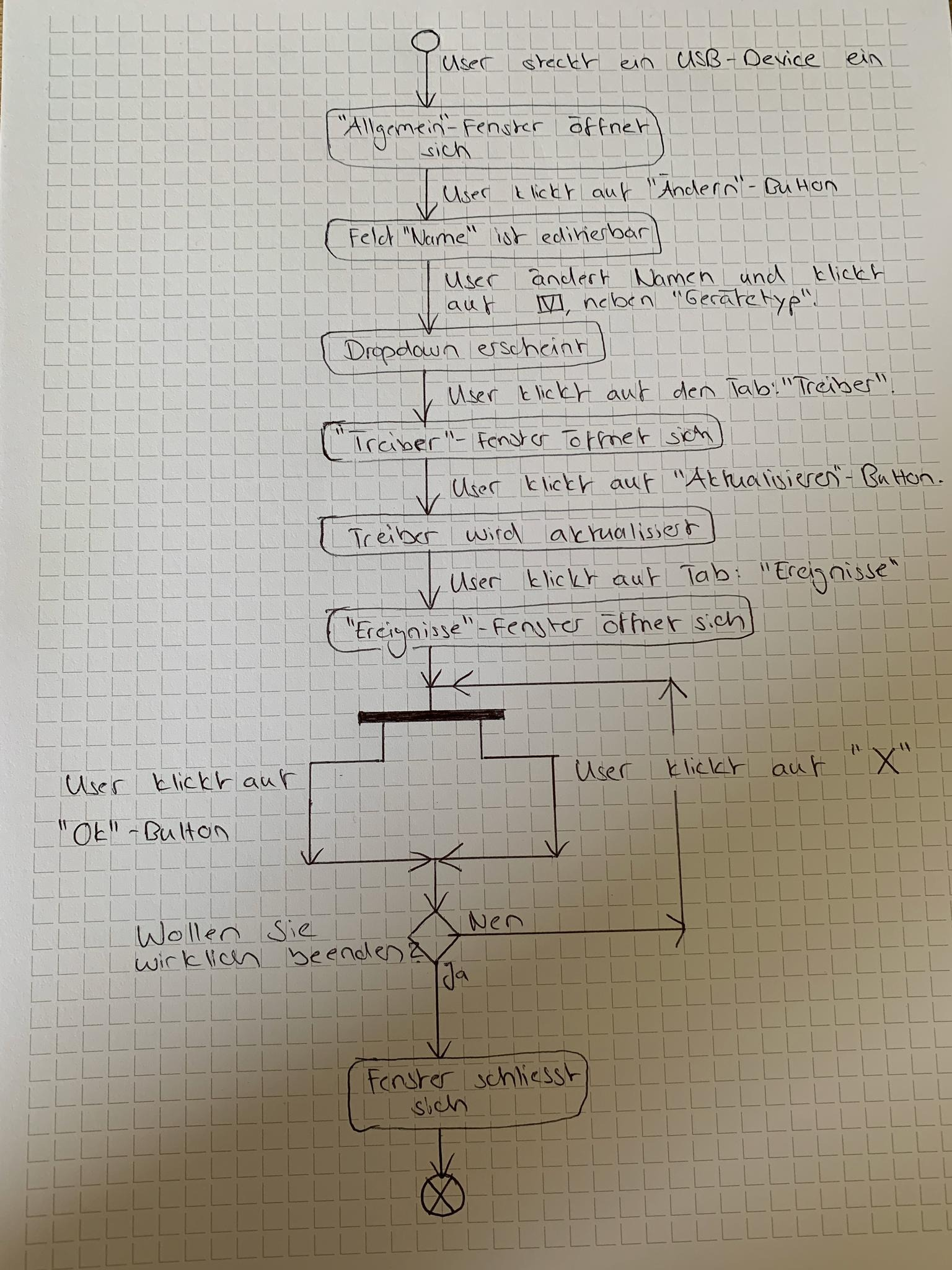
Im Folgenden ist die erwartete Funktionalität dargestellt und erklärt:

## 

## Graphische Benutzer Schnittstelle (GUI) zur Konfiguration des Ablaufs



## Ablauf der Automation



# Testvorschrift (LB2 Meilenstein B2: Teamaufgabe 2)

**Testbeschrieb**

Ziel des Tests

Das vorliegende Automations-Script soll auf folgende Kriterien mit adäquater Abdeckung getestet werden:

* Vorgaben des Automationsablaufes (Daten, Struktur, Konfiguration)
* Integration des Scripts ins Betriebssystem
* Korrekte Ausführung des Automationsablaufes ("Happy Path")
* Sicherstellen der Robustheit bei Falscheingaben und Systemanomalien
* Sicherheitsrelevante Aspekte des Automationsablaufes

**Art des Tests**

Blackbox-Test

**Verwendete Hilfsmittel**

USB-Stick, PC/Laptop

**Anforderung an das Testobjekt**

Funktionstüchtige USB-Devices müssen beim Einstecken automatisch erkannt werden.

**Testvorgaben**

PC/Laptop mit USB-Anschluss und funktionstüchtiger USB-Stick. Powershell-Programm muss ausgeführt werden.

**Abbruchkriterien**

* Programmabsturz
* Rote Fehlermeldungen in der Konsole

**Testprotokoll - Testvalidierung**

|  |  |
| --- | --- |
| **Projektname** | *USB-Device-Notifier (Geräte-Manager)* |
| **Version** *(getestetes Programm)* | *1.0* |
| **Projekt-Code** *(Dateien)* |  |
| **Fachlicher Ansprechpartner** *(Namen der Lehrperson)* |  |
| **Autor des Testprotokolls** | *Joel Rauber und Luca Schweizer* |
| **Testdatum** |  |
| **Name Tester** |  |

**Den Aspekt «Sicherheit» haben wir besprochen und berücksichtigt. Da bei unserem Programm kein Nutzen für irgendwelche spezielle Sicherheit benötigt ist, haben wir dies ausgelassen.**

## Testgruppe <<Tab Allgemein>>

| **Beschreibung Testgruppe**: Tab Allgemein wird getestet | | | | |
| --- | --- | --- | --- | --- |
| **Setup**: Powershell-Skript ausgeführt und USB-Stick vorhanden | | | | |
| **#** | **Nötige Eingabe/Aktion** | **Erwartete Ausgabe/Aktion** | **Tatsächliche Ausgabe/Aktion** | **👍** |
| **A1** | USB-Stick wird eingesteckt | Pop-up erscheint im «Allgemein»-Tab und Geräteeigenschaften werden korrekt angezeigt |  |  |
| **A2** | Gerätename ändern auf «Schule\_Stick» | Gerätename ändert sich auf «Schule\_Stick» |  |  |
| **A3** | Gerätename ändern auf «Schule!Stick» | Name ändert sich nicht und Fehlermeldung:  «Es dürfen nur folgende Sonderzeichen verwendet werden: ‘-‘ und ‘\_’ » |  |  |
| **A4** | Gerätetyp ändern auf «Maus» | Gerätetyp wird auf «Maus» geändert und Symbol ändert auf eine Maus |  |  |
| **A5** | Treiber-Tab anklicken | Treiber-Tab öffnet sich |  |  |
| **A6** | Ereignisse-Tab anklicken | Ereignisse-Tab öffnet sich |  |  |
| **A7** | X-Button betätigen | Bestätigung wird verlangt |  |  |
| **A8** | Ok-Button betätigen | Bestätigung wird verlangt |  |  |

## Testgruppe <<Tab Treiber>>

| **Beschreibung Testgruppe**: Tab Treiber wird getestet | | | | |
| --- | --- | --- | --- | --- |
| **Setup**: Powershell-Skript ausgeführt und USB-Stick vorhanden | | | | |
| **#** | **Nötige Eingabe/Aktion** | **Erwartete Ausgabe/Aktion** | **Tatsächliche Ausgabe/Aktion** | **👍** |
| **B1** | Treiber-Tab wird geöffnet | Treiber-Details werden korrekt angezeigt |  |  |
| **B2** | Button «Aktualisieren» betätigen (falls Treiber aktuell) | Meldung: «Treiber sind auf dem neusten Stand» |  |  |
| **B3** | Button «Aktualisieren» betätigen (falls Treiber noch nicht aktuell) | Meldung: «Treiber wurde aktualisiert» |  |  |
| **B4** | Allgemein-Tab anklicken | Allgemein-Tab öffnet sich |  |  |
| **B5** | Ereignisse-Tab anklicken | Ereignisse-Tab öffnet sich |  |  |
| **B6** | X-Button betätigen | Bestätigung wird verlangt |  |  |
| **B7** | Ok-Button betätigen | Bestätigung wird verlangt |  |  |

## Testgruppe <<Tab Ereignisse>>

| **Beschreibung Testgruppe**: Tab Ereignisse wird getestet | | | | |
| --- | --- | --- | --- | --- |
| **Setup**: Powershell-Skript ausgeführt und USB-Stick vorhanden | | | | |
| **#** | **Nötige Eingabe/Aktion** | **Erwartete Ausgabe/Aktion** | **Tatsächliche Ausgabe/Aktion** | **👍** |
| **C1** | Ereignisse-Tab wird geöffnet | Ereignisse werden korrekt angezeigt |  |  |
| **C2** | Gerätename ändern (im Tab Allgemein) | Ereignis wird angezeigt |  |  |
| **C3** | Allgemein-Tab anklicken | Allgemein-Tab öffnet sich |  |  |
| **C4** | Treiber-Tab anklicken | Treiber-Tab öffnet sich |  |  |
| **C5** | X-Button betätigen | Bestätigung wird verlangt |  |  |
| **C6** | Ok-Button betätigen | Bestätigung wird verlangt |  |  |

## Testgruppe <<Happy Path>>

| **Beschreibung Testgruppe**: Der Happy Path wird durchgeführt | | | | |
| --- | --- | --- | --- | --- |
| **Setup**: Powershell-Skript ausgeführt und USB-Stick vorhanden | | | | |
| **#** | **Nötige Eingabe/Aktion** | **Erwartete Ausgabe/Aktion** | **Tatsächliche Ausgabe/Aktion** | **👍** |
| **D1** | USB-Stick wird eingesteckt, Gerätename ändern, Treiber Tab anklicken, Treiber aktualisieren, Ereignisse Tab anklicken | Gerätename geändert, Treiber aktualisiert, Ereignisse werden korrekt angezeigt |  |  |

# Testprotokoll (LB2 Meilenstein C2: individuelle Aufgabe 4)

Ausgefülltes Testprotokoll siehe Dokument   
***M122\_LB2\_Testvorschrift\_MS-C2\_Namen.docx***

# Systemdokumentation (Meilenstein C: individuelle Aufgabe 3)

Die erstellten Projekt-WPS-Scripts sind hier abgelegt und für Entwickler dokumentiert:

Öffentliche GitHub-URLs im Ablageordner auf dem BSCW ablegen! (pro Team)

Namenskonvention URL: **M122\_Klasse\_Thema\_Name\_Name**

*🡪 Ein* ***Branch*** *und separater Doku-Ordner pro Teammitglied erstellen*

## Umfang / Abgrenzung / Änderungen gegenüber Design

Aufgrund unten beschriebener Umstände sind Anpassungen des ursprünglichen Lösungsdesigns gemacht worden:

...

Umstände / Anpassungen / Veränderungen

## Funktionalität der Implementation.

Zusätzlich zu der Inline-Dokumentation sind hier folgende Funktionen / Eventhandler detailliert beschrieben:

...

Ausführliche Beschreibung der internen Funktionen (Eventhandler wie z.B: $btn\_ok\_Click), der Parameter und der Rückgabewerte

**Struktogramm** (BET / SYS) oder **UML Aktivitätsdiagramm** (API)

# Betriebsdokumentation (Meilenstein C: individuelle Aufgabe 3)

Für Administrator und Benutzer wird folgende Anleitung ausgeliefert ...

## Installationsanleitung für Administratoren

Das Programm ist folgendermassen zu installieren und konfiguriert ...

## Bedienungsanleitung für Benutzer

Das Programm ist folgendermassen zu bedienen ...