# Teilnehmer/in des Teams:

|  |  |  |  |
| --- | --- | --- | --- |
| Name: Afkhami | | Vorname: Sorusch | |
|  | | | |
| Abgabedatum : 09.11.2021 | Klasse: BI 20 a | | Team: GrAf |

# 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

GitGui Repository und darin enthaltener Test Folder

## Testvorgaben

Git muss installiert sein.

GitHub Account muss vorhanden sein. Man muss auf GitHub von dem Benutzen CLI aus angemeldet sein.

Folgendes Git repository abzweigen:

Git Repository auf PC Clonen

Test Folder des Git Gui repository an einen Pfad kopieren, der kein git repository ist.

## Abbruchkriterien

(Rote Fehlermeldungen in der Konsole, Programm reagiert nicht mehr, endlos Schlaufe)

# Testprotokoll - Testvalidierung

|  |  |
| --- | --- |
| **Projektname** | *GitGui* |
| **Version** *(getestetes Programm)* | *1.0.0* |
| **Projekt-Code** *(Dateien)* | *GitGui.ps1, GitGui\_Branches.ps1,*  *GitGui\_Clone\_Repository.ps1* |
| **Fachlicher Ansprechpartner** *(Namen der Lehrperson)* | *Herr Kellenberger* |
| **Autor des Testprotokolls** | *Daniel Greil/Sorusch Afkhami* |
| **Testdatum** |  |
| **Name Tester** |  |

## Testgruppe <<A>>

| **Beschreibung Testgruppe**: Einbindung des Projekt Ordner | | | | |
| --- | --- | --- | --- | --- |
| **Setup**: **Drücken von «Open Project» Knopf** | | | | |
| **#** | **Nötige Eingabe/Aktion** | **Erwartete Ausgabe/Aktion** | **Tatsächliche Ausgabe/Aktion** | **👍** |
| **A1** | **Einfach wieder schliessen** | **Keine Fehlermeldung- Txtboxen leer** |  |  |
| **A2** | **Leeren Ordner auswählen** | **Korrekte Bezeichnung von** |  |  |
| **A3** | **Git ordner auswählen** | **Txtbox Path: Anzeige ganzer Pfad,**  **Txtbox Repro: Anzeige Repository Name**  **Txtbox Branch: Branch sollte angezeigt werden** |  |  |
| **A4** | **Ordner mit files und kein git repro ist** | **Txtbox Path: Anzeige ganzer Pfad,**  **Txtbox Repro: Anzeige Repository Name**  **Txtbox Branch: Anzeige repro kein git projekt** |  |  |
| **A5** | **Ordner mit Abstand im Namen** | **Korrekte Anzeige von ganzem Pfad** |  |  |

## Testgruppe <<B>>

| **Beschreibung Testgruppe**: Dateien in der Listbox angezeit. | | | | |
| --- | --- | --- | --- | --- |
| **Setup**: - | | | | |
| **#** | **Nötige Eingabe/Aktion** | **Erwartete Ausgabe/Aktion** | **Tatsächliche Ausgabe/Aktion** | **👍** |
| **B1** | **Ohne geöffneten Pfad auf «Show Files Drücken»** | **Eine Fehlermeldung, Keine Internen Fehler** |  |  |
| **B2** | **Ordner gewählt.** | **Korrekte Ausgabe der Dateien im Ordner** |  |  |
| **B3** | **Ordner wählen, ihn dann löschen, dann auf «Show Files» drücken** | **Fehlermeldung, die Sagt, dass Pfad nicht existiert** |  |  |

## Testgruppe <<C>>

| **Beschreibung Testgruppe**: Projekt in den Pfad klonen | | | | |
| --- | --- | --- | --- | --- |
| **Setup**: «Clone Project» drücken | | | | |
| **#** | **Nötige Eingabe/Aktion** | **Erwartete Ausgabe/Aktion** | **Tatsächliche Ausgabe/Aktion** | **👍** |
| **B1** | **Nichts eingeben und auf Clonen drücken** | **Eine Fehlermeldung, die sagt, dass Eingabe Leer ist.** |  |  |
| **B2** | **Fenster wieder schliessen** | **Keine Internen Fehler** |  |  |
| **B3** | **«Kein Link» als Link einfügen und Clone Path eingeben** | **Fehlermeldung mit inaktivem Link** |  |  |
| **B4** | **Korrekten Link eingeben und inkorrekten Pfad eingeben** | **Fehlermeldung mit nicht korrektem Pfad** |  |  |
| **B5** | **Korrekter Link und Korrekter Pfad eingeben** | **Korrekte kopieren auf gewähltes Repository** |  |  |

## Testgruppe <<D>>

| **Beschreibung Testgruppe**: Git Repository Initialisieren | | | | |
| --- | --- | --- | --- | --- |
| **Setup**: personal Access Token erstellen auf Github  Settings > Developer Settings > Personal Access Tokens > Generate New Token > Select Scopes: repo > Access Token notieren | | | | |
| **#** | **Nötige Eingabe/Aktion** | **Erwartete Ausgabe/Aktion** | **Tatsächliche Ausgabe/Aktion** | **👍** |
| **B1** | **Username & Personal Access token Feld leer lassen. Auf «Initialise Git» drücken** | **Eine Fehlermeldung, die sagt, dass login credentials fehlen** |  |  |
| **B2** | **Ordner gewählt. Ordner ist kein Repository. Username & Personal Access token ausfüllen. Auf «Initialise Git» drücken** | **Korrekte initialisierung von einem Git Repository. «.git» muss im Ordner vorhanden sein. Repository muss auf GitHub gelistet sein. Daten im GitHub und im Lokalem Git müssen Identisch sein.** |  |  |
| **B3** | **Pfad wählen, der schon ein git Repository ist. Auf «Initialise Git» drücken** | **Fehlermeldung, die Sagt, dass Ordner schon ein git Repository ist.** |  |  |

## Testgruppe <<E>>

| **Beschreibung Testgruppe**: Branches Verwalten | | | | |
| --- | --- | --- | --- | --- |
| **Setup**: - | | | | |
| **#** | **Nötige Eingabe/Aktion** | **Erwartete Ausgabe/Aktion** | **Tatsächliche Ausgabe/Aktion** | **👍** |
| **B1** | **Invaliden Pfad angeben und dann auf «Select Branch» drücken** | **Eine Fehlermeldung, die sagt, dass es den Ordner nicht gibt** |  |  |
| **B2** | **Pfad angeben, der gar kein git Repository ist. Dann Auf Branches drücken** | **Fehlermeldung** |  |  |
| **B3** | **Richtigen Pfad angeben, auf «Select Branch» drücken, neuen Branch mit Abständen angeben.** | **Fehler / Name wird ohne abstände erstellt** |  |  |
| **B4** | **Richtigen Pfad angeben, auf «Select Branch» drücken, Branches Wechseln** | **Einwandfreies wechseln von Branches, keine Fehler** |  |  |
| **B5** | **Richtigen Pfad angeben, auf «Select Branch» drücken, Branches löschen** | **Einwandfreies Löschen von Branches** |  |  |
| **B6** | **Richtigen Pfad angeben, auf «Select Branch» drücken, aktiven Pfad löschen** | **Automatisches Wechseln von aktivem Branch** |  |  |

## Testgruppe <<F>>

| **Beschreibung Testgruppe**: Git History anzeigen | | | | |
| --- | --- | --- | --- | --- |
| **Setup**: - | | | | |
| **#** | **Nötige Eingabe/Aktion** | **Erwartete Ausgabe/Aktion** | **Tatsächliche Ausgabe/Aktion** | **👍** |
| **B1** | **Ordner wählen, der kein Git ordner ist. Auf «History» drücken** | **Fehlermeldung, kein git ordner** |  |  |
| **B2** | **Ordner wählen, Es nicht gibt. Auf «History» drücken** | **Fehlermeldung, kein Ordner gefunden** |  |  |
| **B3** | **Ordner wählen, den es gibt. Auf «History» drücken** | **Git History wird auf List box angezeigt** |  |  |

## Testgruppe <<F>>

| **Beschreibung Testgruppe**: Git Status anzeigen. | | | | |
| --- | --- | --- | --- | --- |
| **Setup**: - | | | | |
| **#** | **Nötige Eingabe/Aktion** | **Erwartete Ausgabe/Aktion** | **Tatsächliche Ausgabe/Aktion** | **👍** |
| **B1** | **Ordner wählen, der kein Git ordner ist. Auf «Status» drücken** | **Fehlermeldung, kein git ordner** |  |  |
| **B2** | **Ordner wählen, Es nicht gibt. Auf «Status» drücken** | **Fehlermeldung, kein Ordner gefunden** |  |  |
| **B3** | **Ordner wählen, den es gibt. Auf «Status» drücken** | **Git Status wird auf List box angezeigt** |  |  |

| **Review des Testbeschriebs durch den Tester**:  *(Tester beurteilt Testbeschrieb nach erfolgter Ausführung.  Fehler in der Beschreibung?  Fehler im Protokoll?)* |
| --- |

# Sign-Off

|  |
| --- |
| **Mängelliste:**  *(Alle nicht mit OK markierten Testfälle hier auflisten und etwaige Beobachtungen und/oder Bemerkungen notieren, damit der Entwickler Anhaltspunkte zur Verbesserung erhält.)*   * A1: |
| **Der Test**  () wird **erfolgreich** abgenommen.  () wird eingeschränkt abgenommen (Mängel siehe oben).   Der Test wird **trotzdem als erfolgreich** abgenommen erklärt.  () wird **nicht** abgenommen (aufgetretene Mängel siehe oben)  Bis zum angegebenen Zeitpunkt werden alle oben beschriebenen Mängel beseitigt.  () Datum:  **Test ist beendet und wurde korrekt durchgeführt**  **Ja () Nein ( ) Unterschrift (Datum, Name** *Tester)***)**  **Ja () Nein ( ) Unterschrift (Datum, Name** *Autor)*  **Validierung**  **Ja () Nein ( ) Unterschrift (Datum, Name** *Experte***)** |