|  |
| --- |
| debug.png |
| Anforderungsspezifikation  Bitcoin Script Debugger  **Projektteam x**  **Rolf Zurbrügg**  **Samuel Egger**  **X0.2 10.10.17** |
| **Berner Fachhochschule**  Departement für Technik und Informatik |

**Inhaltsverzeichnis**

[1 Zweck des Dokuments 3](#_Toc495439568)

[2 Projektvision 3](#_Toc495439569)

[3 Projektziele 3](#_Toc495439570)

[4 Systemabgrenzung 3](#_Toc495439571)

[4.1 Prozessumfeld 3](#_Toc495439572)

[4.2 Systemumfeld 3](#_Toc495439573)

[4.3 Randbedingungen 3](#_Toc495439574)

[5 Anforderungen 4](#_Toc495439575)

[5.1 Quellen und Vorgehen 4](#_Toc495439576)

[5.2 Funktionale Anforderungen 4](#_Toc495439577)

[5.3 Technische Anforderungen 4](#_Toc495439578)

[5.4 Qualitätsanforderungen 4](#_Toc495439579)

[6 Glossar 5](#_Toc495439580)

[7 Abbildungsverzeichnis 5](#_Toc495439581)

[8 Tabellenverzeichnis 5](#_Toc495439582)

[9 Literaturverzeichnis 5](#_Toc495439583)

[10 Anhang 5](#_Toc495439584)

[10.1 Abstimmung der Anforderungen 5](#_Toc495439585)

[10.2 Definition of Ready - Checklist 5](#_Toc495439586)

[11 Versionskontrolle 5](#_Toc495439587)

# Zweck des Dokuments

Dieses Dokument beschreibt die Ziele und Anforderungen für den Bitcoin Script Debugger.

# Projektvision

Bitcoins können gekauft und verkauft werden. Der Wechsel von Besitzern wird Mittels Locking und Unlocking Scripts geregelt. Die Script Sprache ist eine Bitcoin eigene Script Sprache welche nicht Turing mächtig ist und Stack basiert ist. Die Script Sprache besteht aus so genannten OP Codes. Das Locking und Unlocking Script werden concatiniert und dann evaluiert. Wenn am Schluss der evaluation True auf dem Stack steht, dann ist eine Transaktion gültig.

Ziel dieses Projektes ist es eine Webseite zu entwickeln basierend auf HTML, CSS und JavaScript, welche dem Nutzer erlaubt Bitcoin Locking und Bitcoin Unlocking Scripts zu Testen / Debugen. Zur Unterstützung des Debuggings soll der Stack visualisiert werden. Der visualisierte Stack soll den Nutzer unterstützen beim Verstehen wie der Stack aussieht nach jeder ausführen eines OP Codes.

Durch dieses Projekt soll es Benutzern erst mal Ermöglicht werden Bitcoin Script in einem Sicheren Umfeld Offline zu Testen. Dies ist wichtig, weil der Besitz und Handel von Bitcoins basiert auf Asymmetrischer Kryptographie, dass bedeutet, dass man zum Handeln von Bitcoins Public und Private Keys benötigt. Diese werden mittels Script Sprache dann geprüft. Deswegen ist der Sicherheits Aspekt essenziell bei diesem Projekt.

Dieses Projekt soll eine Open Source Lösung sein, welche hauptsächlich zu Schulischen und Forschungs Zwecken eingesetzt wird. Deswegen wird kein neues Geschäftsmodell geschaffen.

# Projektziele

Ziel ist es eine Website mit JavaScript zu realisieren, welche dem Nutzer ermöglicht Bitcoin Skripte zu Debuggen. Dies soll mit Hilfe einer Visualisierung des Bitcoin Skript Stacks passieren.

Die Webseite soll offline voll funktionsfähig sein, damit verhindert wird, dass die eigenen privaten Keys nicht abgefangen werden können.

Stakeholder:

* Dozenten und Studierende mit entsprechender Fachrichtung
* Nutzer und Interessierte von Bitcoins

# Systemabgrenzung

## ~~Prozessumfeld~~

~~Geschäftsprozesse, die bei dem Einsatz der Lösung relevant sind~~

~~Welche davon werden durch das System unterstützt?~~

## Systemumfeld

Das System ist als Schulungssoftware in sich abgeschlossen.

## Randbedingungen

Technische Rahmenbedingungen

* Offline lauffähige Webapplikation
* Umsetzung mit JavaScript

Organisatorische Vorgaben

* Abgabe des Source Code für Review am 23.12.2017
* Code reviews erfolgen bis am 15.01.2017
* Projektpräsentation am 18.01.2017
* Projektabschluss am 20.01.2017

# Anforderungen

## Quellen und Vorgehen

Der Projektumfang und die Ziele sind bereits durch den Auftraggeber Kai Brünnler via Projekt-ausschreibung (Bitcoin-Skript-Debugger) vorgegeben. In der Beschreibung befindet sich ein zusätzlicher Verweis auf ein ähnliches Projekt (https://webbtc.com/script), dessen Funktionsumfang als Referenz für die Erhebung der Anforderungen hinzugezogen werden kann.

## Funktionale Anforderungen

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Status** | **Prio** | **Beschreibung** |
| **F1.1** | Entwurf | **M** | Die Applikation erlaubt die Eingabe eines Bitcoin Scriptes und dessen Eingangsparameter. |
| **F1.2** | Entwurf | **P1** | Die Syntax des Bitcoin Scriptes wird farblich hervorgehoben (Syntaxhighlighting). |
| **F1.3** | Entwurf | **P1** | Syntaxfehler werden in echtzeit (d.h. während der Eingabe) erkannt und angezeigt. |
| **F2.1** | Entwurf | **M** | Das Bitcoin Script kann zusammen mit den Eingangsparametern ausgeführt werden. Die Ausgabe wird angezeigt. |
| **F2.2** | Entwurf | **M** | Zeilenweises Ausführen des Bitcoin Scriptes. Bei jedem Ausführungsschritt wird die aktuelle Operation und der Zustand des Stacks visuell hervorgehoben. |

Tabelle 1: Funktionale Anforderungen

Attribute:

ID: eindeutige Identifikation

Status: Entwurf / geprüft / freigegeben

Priorität: Muss / Optional P1, P2, P3 / Wunsch (Nice to have)

## Technische Anforderungen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Status** | **Prio** | | **Beschreibung** | |
| **T1.1** | Entwurf | M | Das Projekt muss als offline Webapplikation ausgeführt werden können. | |
| **T2.1** | Entwurf | M | Der Source Code des Projekts ist Open Source. | |

Tabelle 2: Technische Anforderungen

## Qualitätsanforderungen

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Status** | **Prio** | **Beschreibung** |
| **Q1.1** | Entwurf | M | Es ist durch manuelle Tests sicherzustellen, dass das Projekt den erwarteten Anforderungen entspricht. |

Tabelle 3: Qualitätsanforderungen

# Glossar

# Abbildungsverzeichnis

**No table of figures entries found.**

# Tabellenverzeichnis

Tabelle 1: Funktionale Anforderungen 4

Tabelle 2: Technische Anforderungen 4

Tabelle 3: Qualitätsanforderungen 4

# Literaturverzeichnis

*Script.* 02 10 2017. https://en.bitcoin.it/wiki/Script.

# Anhang

## Abstimmung der Anforderungen

Eventuell aufgetretene Konflikte und die gewählte Lösung dafür werden kurz dokumentiert

## Definition of Ready - Checklist

# Versionskontrolle

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Datum** | **Beschreibung** | **Autor** |
| X0.1 | 02.10.2017 | Dokument erstellt | Rolf Zurbrügg |
| X0.2 | 10.10.2017 | Erfassen der Anforderungen | Samuel Egger |