Studie

| Auftraggeber | Beat Walter |
| --- | --- |
| Projektleiter | Agash Thamotharampillai |
| Autor | Agash Thamotharampillai |
| Klassifizierung | Extern |
| Status | In Arbeit |
|  |  |

Änderungsverzeichnis

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | Version | Änderung | Autor |
| 11.02.2014 | 0.1 | Initialversion | Agash Thamo. |
|  |  |  |  |
|  |  |  |  |

Inhaltsverzeichnis

1 Situationsanalyse 2

1.1 Ausgangslage 2

1.2 Stärken 2

1.3 Schwächen 2

2 Ziele 2

3 Liste der Stakeholder 3

4 Anforderungen *(Initial Product Backlog)* 3

5 Lösungsvarianten 4

5.1 Variantenübersicht 4

5.2 Beschreibung der Varianten 4

6 Bewertung der Varianten (Tabelle) 4

7 Lösungsbeschreibung 4

8 Schutzbedarfsanalyse 4

9 Empfehlung 5

# Situationsanalyse

## Ausgangslage

Beschreibung der IST-Situation:

Im Moment steht uns als Experimentiergrundlage noch kein solches Programm/solche Anwendung zur Verfügung.

Da es noch kein solches Projekt gibt und unser vorgesehenes Projekt der Ausbildung dient, gibt es an sich keine Stärken oder Schwächen zur Ausgangslage.

# Ziele

**Systemziele**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr | Kategorie | Beschreibung | Messgrösse | Priorität |
| 1 | Client | Der Keylogger soll die Tasteneingaben einlesen und speichern können. | - | M |
| 2 | Kommunikation | Der Keylogger soll die Daten die er erfasst/gespeichert hat an den Server senden. | - | 1 |
| 3 | Server | Der Server-Teil des Keyloggers soll die Daten vom client analysieren können. | Schreibgewohnheiten vom Benutzer sollen erkennbar sein. | 1 |
| 4 | Server (GUI) | Der Server soll die analysierten Daten grafisch anzeigen können. | - | 2 |
| 5 | Kompatibilität | Der Client soll auf modernen Windows-Computern lauffähig sein. | Unterstützt sollen Windows 7-8.1 | M |
| 6 | Stabilität | Der Client ist unabhängig vom Server und ist auch ohne Netzwerkverbindung lauffähig. | - | 2 |
| 7 | Stabilität | Die Daten vom Client und vom Server sind auch nach einem Absturz noch verfügbar. | - | 2 |

Legende: Priorität: M=Muss /1=hoch, 2=mittel, 3=tief

**Vorgehensziele**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr | Kategorie | Beschreibung | Messgrösse | Priorität |
| 1 | Dokumentation | Alle Dokumente werden laufend nachgeführt und bei Änderungen im Projekt nachdokumentiert. | Alle notwendigen Dokumente sind vorhanden und aktuell. | M |
| 2 | Entwicklungsvorgehen | Bearbeitungen und Ergänzungen am Code werden nur dann in das Master-Repository commited, wenn der Code funktionstüchtig qualitativ hochwertig ist. | Der Code im Master-Repository ist zu jeder Zeit ausführbar. | M |
| 3 | Qualitätsmanagement | Bei jeder Projektstufe gibt es auf entsprechender Ebene (User-Stories, To-Do’s, Planung, Realisierung, etc.) gibt es Testfälle. Die testfälle werden zur gleichen Zeit erstellt wie das zugehörige Element. | Der Code/das Projekt wird auf verschiedenen Stufen getestet (Komponententests, Integrationstests, Systemtests und Abnahmetests). | 1 |

Legende: Priorität: M=Muss /1=hoch, 2=mittel, 3=tief

**Rahmenbedingungen**

Nach dem Projektende werden keine weiteren Supportleistungen oder Softwareerweiterungen geliefert.

**Abgrenzung**

* Das Programm soll auf keinen Fall eine Schadsoftware darstellen.
* Der Benutzer des Clients kann seine eigenen Daten **nicht** einsehen.

# Liste der Stakeholder

* Der Auftraggeber hat das Interesse uns bei dem Projektablauf zu unterstützen und dass wir als Entwickler einen Lerneffekt haben.
* Wir als Entwickler haben Interesse dabei uns mit Netzwerkkommunikation auseinander zu setzen.

# Anforderungen *(Initial Product Backlog)*

**Initial Product Backlog**

Das Initial Product Backlog wird in der Phase Initialisierung erstellt und bildet den Startpunkt für die Entwicklung nach Scrum.

Es enthält in listenform Anforderungen und Tätigkeiten („Product Backlog Items“) geordnet nach Priorität für das gesamte Projekt. Dabei ist der erste Eintrag in der Liste das am höchsten priorisierte Element. Die Priorisierung erfolgt nach dem Business Value der einzelnen Einträge.

Das Product Backlog wird üblicherweise in Story Points geschätzt. Die Story Points stellen den relativen Aufwand der Product Backlog Items untereinander dar.   
Die Schätzung wird im weiteren Verlauf des Projekts in regelmässigen Estimation Meetings während der Sprints fortgesetzt.

|  |  |
| --- | --- |
| Initial Product Backlog | |
| **Anforderungen / Tätigkeiten** | **Story Points** |
| Keyboard Inputs lesen |  |
| Clientdaten Zwischenspeichern |  |
| Verbindung Client-Server herstellen |  |
| Gespeicherte Daten senden |  |
| Empfangene Daten verarbeiten |  |
| Logging implementieren |  |

# Lösungsvarianten

## Variantenübersicht

* Implementierung mit .NET und C#
* Implementierung mit C/C++
* Implementierung mit Python

## Beschreibung der Varianten

### .Net und C#

Der Keylogger wird in C# mit .NET für Windows geschrieben. Die Daten werden in XML abgespeichert und die Datenkommunikation findet mit Websockets statt.

# Bewertung der Varianten (Tabelle)

Variante a)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kriterium | Gewicht | Punkte | Total |  |
| Abdeckung der Anforderungen | Bsp.  5 | 2 | 10 | <Erläuterung> |
| Realisierbarkeit, Risiken |  |  |  |  |
| Wirtschaftlichkeit |  |  |  |  |
| *ggf. weitere Kriterien* |  |  |  |  |
| **Gesamtbeurteilung** |  |  |  |  |

Variante b)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kriterium | Gewicht | Punkte | Total |  |
| Abdeckung der Anforderungen |  |  |  | <Erläuterung> |
| Realisierbarkeit, Risiken |  |  |  |  |
| Wirtschaftlichkeit |  |  |  |  |
| *ggf. weitere Kriterien* |  |  |  |  |
| **Gesamtbeurteilung** |  |  |  |  |

Beurteilungen: 1-10, Gewichtung 1-5

# Lösungsbeschreibung

Detailliertere Beschreibung der ausgewählten Variante

# Schutzbedarfsanalyse

Behandlung des zu fordernden Schutzbedarfs und der Erfüllung durch die Lösungsvariante

# Empfehlung

Weiterverfolgung der ausgewählten Variante, besondere zu ergreifende Massnahmen

Abgeleitet: Projektplanung (Verweis auf Dokument Projektplanung).