

Studie

Auftraggeber	Beat Walter
Projektleiter	Agash Thamothersampillai
Autor	Agash Thamothersampillai
Klassifizierung	Extern
Status	Zur Abnahme freigegeben

Änderungsverzeichnis

Datum	Version	Änderung	Autor
11.02.2014	0.1	Initialversion	Agash Thamo.
18.02.2014	1.0	Finalversion der Studie	Agash Thamo., Marc Trittibach
25.02.2014	1.1	Kapitel 9 Empfehlung wurde angepasst	Agash Thamo., Marc Trittibach, Dmitry Prudnikov

Inhaltsverzeichnis

1	Situationsanalyse	2
1.1	Ausgangslage	2
2	Ziele	2
2.1	Systemziele	2
2.2	Vorgehensziele	3
2.3	Rahmenbedingungen	3
2.4	Abgrenzung	3
3	Liste der Stakeholder	4
4	Anforderungen (<i>Initial Product Backlog</i>)	4
5	Lösungsvarianten	4
5.1	Variantenübersicht	4
5.2	Beschreibung der Varianten	4
5.2.1	.Net und C#	4
5.2.2	C/C++	5
5.2.3	Python	5
6	Bewertung der Varianten (Tabelle)	5
6.1	Variante 1: .NET und C#	5
6.2	Variante 2: C/C++	6
6.3	Variante 3: Python	6
7	Lösungsbeschreibung	7
8	Schutzbedarfsanalyse	7
9	Empfehlung	7

1 Situationsanalyse

1.1 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.

2 Ziele

2.1 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

2.2 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 committed, 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

2.3 Rahmenbedingungen

Nach dem Projektende werden keine weiteren Supportleistungen oder Softwareerweiterungen geliefert.

2.4 Abgrenzung

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

3 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.

4 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	30
Clientdaten Zwischenspeichern	5
Verbindung Client-Server herstellen	30
Gespeicherte Daten senden	10
Empfangene Daten verarbeiten	10
Logging implementieren	10

5 Lösungsvarianten

5.1 Variantenübersicht

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

5.2 Beschreibung der Varianten

5.2.1 .Net und C#

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

C# ist eine managed Sprache.

5.2.2 C/C++

Der Keylogger wird in C/C++ realisiert. Die Daten werden ebenfalls in XML gespeichert und die Datenkommunikation findet auch über Websockets statt.

Hier wird auf Windows-Spezifische Funktionen verzichtet.

Es wird unmanaged C/C++ verwendet. Das Speichermanagement muss selbst realisiert werden.

5.2.3 Python

Der Keylogger wird in der dynamischen Sprache Python realisiert. Die Daten werden in XML gespeichert und die Datenkommunikation findet ebenfalls über Sockets statt.

In Python gibt es keine effektive Möglichkeit Platz im Speicher freizugeben, der Garbage-Collector regelt dies automatisch.

6 Bewertung der Varianten (Tabelle)

6.1 Variante 1: .NET und C#

Kriterium	Gewicht	Punkte	Total	
Abdeckung der Anforderungen	5	10	50	Mit .Net und C# können unsere Anforderungen abgedeckt werden.
Realisierbarkeit, Risiken	5	9	45	Es bestehen wenige Risiken, da das Team bereits Erfahrungen mit der Technologie gesammelt hat und ausserdem ist C# relativ einfach zu implementieren.
Wirtschaftlichkeit	2	8	16	Der Zeitaufwand der Realisierungsphase ist relativ gering.
Plattformunabhängigkeit	1	3	3	.Net ist nur auf Windows Umgebungen verfügbar, was jedoch den Anforderungen genügt.
Gesamtbeurteilung			114	

Beurteilungen: 1-10, Gewichtung 1-5

6.2 Variante 2: C/C++

Kriterium	Gewicht	Punkte	Total	
Abdeckung der Anforderungen	5	10	50	Mit C/C++ können unsere Anforderungen abgedeckt werden.
Realisierbarkeit, Risiken	5	6	30	Mit C/C++ kann die Realisierung von gewissen Kernteilen kompliziert werden, was ein hohes Risiko darstellt. Die Möglichkeiten mit C/C++ sind fast uneingeschränkt.
Wirtschaftlichkeit	2	5	10	Der Zeitaufwand wird in dieser Variante höher ausfallen.
Plattformunabhängigkeit	1	8	8	C/C++ ist auf sämtlichen gängigen Plattformen verfügbar, jedoch muss bei der Realisierung dabei geachtet werden.
Gesamtbeurteilung			98	

Beurteilungen: 1-10, Gewichtung 1-5

6.3 Variante 3: Python

Kriterium	Gewicht	Punkte	Total	
Abdeckung der Anforderungen	5	10	50	Mit Python können unsere Anforderungen abgedeckt werden.
Realisierbarkeit, Risiken	5	4	20	Im Team besitzt nur Agash Kenntnisse über diese Technologie. Die Umsetzung der Netzchnittstelle könnte Probleme darstellen. Die Entwicklung kann nicht im Visual Studio stattfinden.
Wirtschaftlichkeit	2	5	10	Auch wenn vergleichsweise wenig Aufwand in die Programmierung nötig ist, muss sich das Team zuerst einarbeiten.
Plattformunabhängigkeit	1	10	10	Python ist Plattformunabhängig.
Gesamtbeurteilung			90	

Beurteilungen: 1-10, Gewichtung 1-5

7 Lösungsbeschreibung

Wir haben uns für die Variante mit .NET und C# entschieden. Die Realisierung ist vergleichsweise einfach und das Team hat bereits Erfahrungen mit der Entwicklung mit C#. Die Netzchnittstelle ist einfach zu implementieren.

Die anderen Varianten wären im Prinzip auch völlig möglich gewesen, jedoch fällt ein höherer Entwicklungsaufwand an.

8 Schutzbedarfsanalyse

Die an den Server gesendeten Daten sollen nicht an Drittpersonen weitergeleitet werden. Die Daten werden verschlüsselt auf dem Server und Client abgelegt.

9 Empfehlung

Wir empfehlen das Projekt weiterzuführen. Durch den Variantenvergleich (siehe Kapitel 5 Lösungsvarianten) und der Machbarkeitsstudie, die von Jonathan Camenzind durchgeführt wurde, folgt, dass das Projekt erfolgreich realisiert werden kann.