Realisierungsbericht

|  |  |
| --- | --- |
| **Status** | Abgeschlossen |
| **Projektname** | Network Snapshot Appliance (NSA) |
| **Projektleiter** | Joel Meier |
| **Auftraggeber** | Stephan Zigerli |
| **Autoren** | Micha Simon, Laxushan Yogalingam, Joel Meier, Sven Trachsel |
| **Verteiler** | Anna Stettler |

**Änderungskontrolle, Prüfung, Genehmigung**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Datum | Beschreibung, Bemerkung | Name oder Rolle |
| 0.0.1 | 10.11.2018 | Erste Einträge in die Dokumentation | Micha Simon |
| 0.0.2 | 20.11.2018 | Systemdesign | Micha Simon, Laxushan Yogalingam, Sven Trachsel |
| 0.0.3 | 27.11.2018 | Systemdokumentation, Systemtest, Weiterführung der Projektplanung | Micha Simon, Laxushan Yogalingam, Joel Meier, Sven Trachsel |
|  |  |  |  |

**Definitionen und Abkürzungen**

|  |  |
| --- | --- |
| Begriff / Abkürzung | Bedeutung |
|  |  |

**Referenzen**

|  |  |
| --- | --- |
| Referenz | Titel, Quelle |
| [1] |  |
| [2] |  |
| [3] |  |

**Inhaltsverzeichnis**

[1 Zusammenfassung 3](#_Toc531077028)

[2 Technische Detailspezifikation 3](#_Toc531077029)

[2.1 Systemdesign 3](#_Toc531077030)

[2.1.1 Struktur 3](#_Toc531077031)

[2.1.2 Beschreibung der Elemente 3](#_Toc531077032)

[2.2 Schnittstellendefinitionen 4](#_Toc531077033)

[2.3 Sicherheit (ISDS) 4](#_Toc531077034)

[2.4 Anforderungszuordnung 5](#_Toc531077035)

[3 Systemdokumentation 5](#_Toc531077036)

[3.1 Konfigurations-Dokumentation 5](#_Toc531077037)

[3.2 Benutzerhandbuch 5](#_Toc531077038)

[3.2.1 Systemübersicht 5](#_Toc531077039)

[3.2.2 Anwenderfunktionalität 6](#_Toc531077040)

[3.3 Supporthandbuch 6](#_Toc531077041)

[3.3.1 Massnahmen bei Benutzerproblemen 6](#_Toc531077042)

[3.3.2 Massnahmen bei technischen Problemen 6](#_Toc531077043)

[3.3.3 Anhang zum Supporthandbuch 6](#_Toc531077044)

[4 Systemtest 6](#_Toc531077045)

[4.1 Testspezifikation 6](#_Toc531077046)

[4.1.1 Kritikalität der Funktionseinheit 6](#_Toc531077047)

[4.1.2 Testanforderungen 7](#_Toc531077048)

[4.1.3 Testverfahren 7](#_Toc531077049)

[4.1.4 Testkriterien 7](#_Toc531077050)

[4.1.5 Testfälle 7](#_Toc531077051)

[4.2 Testprozedur 7](#_Toc531077052)

[4.2.1 Vorbereitung 7](#_Toc531077053)

[4.2.2 Durchführung 8](#_Toc531077054)

[4.2.3 Nachbearbeitung 8](#_Toc531077055)

[4.3 Testprotokoll 8](#_Toc531077056)

[4.3.1 Testobjekt 8](#_Toc531077057)

[4.3.2 Testresultate 8](#_Toc531077058)

[4.3.3 Testauswertung 8](#_Toc531077059)

[5 Weiterführung der Projektplanung 8](#_Toc531077060)

[5.1 Abgleich von Planung und tatsächlichem Verlauf der Phase Konzept 8](#_Toc531077061)

[5.2 Aktualisierung der Risikosituation 9](#_Toc531077062)

[5.3 Planung der nächsten Phase 9](#_Toc531077063)

[6 Projektfreigabe 10](#_Toc531077064)

**Abbildungsverzeichnis**

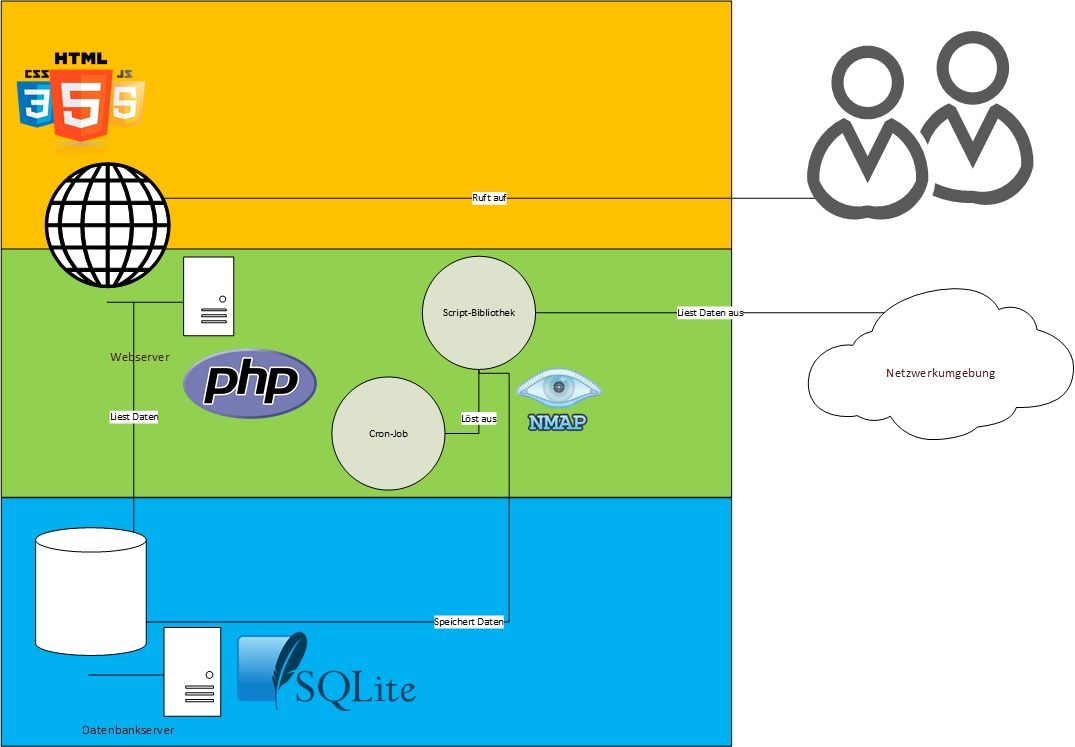
# Zusammenfassung

# Technische Detailspezifikation

## Systemdesign

Beim Umsetzen des Projektes wurde die Systemarchitektur gemäss Konzeptbericht berücksichtigt.

### Struktur



### Beschreibung der Elemente

|  |  |  |  |
| --- | --- | --- | --- |
| Nr. | Modul | Teile | Beschreibung |
| 1 | Frontend (Webseite) | Darstellung | Die Darstellung wird mittels HTML, CSS und JavaScript dargestellt. |
| Logik | Die Logik des Frontend, wird mit PHP gesteuert und von JavaScript Assentiert.  Der Logikteil bezieht die Daten aus der S2 und übergibt die dem Darstellungsteil, der sie dann darstellt. |
| 2 | Datenbeschaffung | Script-Bibliothek | Die Script-Bibliothek deren Scripts meistens NMAP benutze, wird von Server ausgeführt und fragt die verschiedenen Clients und Server im Netzwerk ab und gibt die der S1 zur Verfügung. |
| 3 | Datenspeicherung | Datenbank (Diagramm im Anhang) | Die Datenbank bekommt ihre Daten von der Datenbank Schnittstelle und stellt die Daten über die S2 zur Verfügung. |

Auf der Darstellungsebene werden HTML, CSS und PHP verwendet, als Webserver verwenden wir einen Apache Webserver. Zur Datenbeschaffung benutzen wir eine Scriptbibliothek basierend auf Python mit NMAP-Commands. Diese wird von einem Cron-Job gestartet. Als Datenbank wird ein SQLite3 Server benutzt.

## Schnittstellendefinitionen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Intern / Extern | Schnittstelle | Beschreibung | Daten |
| S1 | Intern | Schnittstelle von Modul Datenbeschaffung zu Datenspeicherung | Die Scripts parsen am Ende die erhaltenen Daten und schreiben sie dann über SQLite3 in die Datenbank speichert. | Resultate der abfragen der Clients im Netzwerk. Die zuvor von der Bibliothek zurechtgeschnitten wurden. |
| S2 | Intern | Schnittstelle von Modul Datenspeicherung zu Frontend | Die Schnittstelle besteht aus dem Programm PHP das Abfragen auf der Datenbank macht und die Daten dann der Webseite übergibt. | Die Daten die in der Datenbank gespeichert sind. |
| S3 | Extern | Schnittstelle von Modul Datenbeschaffung ins Netzwerk | Die Schnittstelle ist die Netzwerkkarte bzw. das Tool NMAP | Angaben über den Angefragten Client oder Server |
| S4 | Extern | Schnittstelle von Darstellung zum User | Die Schnittstelle ist eigentlich die Webseite, die der Benutzer sieht. | Die dargestellte Webseite und Kommandos. |

Python Scripts: Liest die verschiedenen Geräte im Netzwerk au und schreibt sie in die Datenbank

Datenbank: Wird verwendet um die vom Script ausgelesenen Daten zu speichern

Website: Das Frontend von dem aus das Script ausgeführt werden kann und die Daten der Datenbank angezeigt wird.

## Sicherheit (ISDS)

|  |  |  |  |
| --- | --- | --- | --- |
| ID | SAF-01 | Titel | Einloggen |
| User Story | **Rolle:**   * Benutzer | | |
| **Ziel:**   * Ein Passwort wird benötigt, wenn man auf die Webseite geht. | | |
| **Grund:**   * Um die Netzwerkdaten vor Unbefugten zu schützen. | | |
| Akzeptanz | * Beim ersten aufrufen wird ein Feld mit der Erklärung und einem Passwortfeld sowie einem Knopf angezeigt. * Falls der User das Passwort falsch eingibt, wird eine passende Meldung angezeigt. * Falls der User das Passwort richtig eingibt, wird für ihn eine Session erstellt und die Hauptseite der Appliance wird angezeigt. | | |
| Mock-Up | Ein Bild, das Screenshot enthält.  Mit sehr hoher Zuverlässigkeit generierte BeschreibungEin Bild, das Screenshot enthält.  Mit sehr hoher Zuverlässigkeit generierte Beschreibung | | |

## Anforderungszuordnung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AFo.-Nr.** | **Anforderung (Stichwort)** | **Webseite** | **Scripts** | **Datenbank** |
| AF-01 | Automatisierte Ausführung |  | **X** |  |
| AF-02 | Genauigkeit |  | **X** |  |
| AF-03 | freundliche Benutzeroberfläche | **X** |  |  |
| AF-04 | benutzerfreundliche Erweiterbarkeit | **X** | **X** | **X** |
| AF-05 | Freeware | **X** | **X** | **X** |
| SAF-01 | Einloggen | **X** |  |  |

# Systemdokumentation

## Konfigurations-Dokumentation

Netzwerk- oder Systemdokumentation mit allen Konfigurationsdaten, geordnet nach Elementen.

## Benutzerhandbuch

### Systemübersicht

 Ein Bild, das Karte enthält.

Mit hoher Zuverlässigkeit generierte Beschreibung

**Beschrieb:** Wenn der Benutzer die Webseite auf dem Webserver aufruft, liest der Webserver die benötigten Daten aus der Datenbank aus.

In bestimmten Abständen, die noch zu definieren sind, führt die Script Bibliothek sich selber aus und liest die Daten aus dem Netzwerk mit abfragen aus. Einige Angaben, zum Beispiel die der DNS anfragen, werden mit Anfragen aus dem Internet verglichen. Die Daten werden dann in die Datenbank eingetragen. Die Webseite liest diese Daten aus und stellt sie dar.

### Anwenderfunktionalität

**Installation**

**Ein Bild, das Person, Gerät enthält.

Automatisch generierte Beschreibung**

Um die Dateien der Appliance zu kommen muss man zuerst das Repository Clonen, wie das gemacht wird ist im Bild oben ersichtlich.



Um die Appliance zu Installieren oder Änderungen anzuwenden, muss man das Installation Script «install.sh» mit Root rechten ausführen.

Ein Bild, das Gerät enthält.

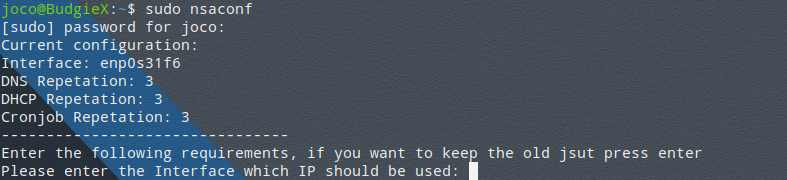
Automatisch generierte Beschreibung

Um die Appliance zu konfigurieren muss man das Kommando «nsaconf» mit Rootrechten ausführen. Dies ist nach jeder Neu-Installation erforderlich.

Ein Bild, das Text enthält.

Automatisch generierte BeschreibungMan muss angeben auf welchen Network Interface die Abfragen ausgeführt werden sollen. Dazu muss man auch angeben wie häufig die Discovery Scripts ausgeführt werden sollten. Dies ist erforderlich, wenn man mehrere DNS oder DHCP Server im Netzwerk hat. Zuletzt muss man angeben wie häufig die Appliance anfragen machen soll (in Minuten).

**Neukonfiguration**

****

Um die Appliance zu Konfigurieren muss man das Kommando «nsaconf» mit Root rechten ausführen. Falls eine Konfiguration schon vorliegt, wird sie ausgegeben. Man kann nun die Appliance neukonfigureren, dies Funktioniert genau gleich wie bei der Erstkonfiguration, jedoch kann man dieses Mal die Entertaste betätigen falls man die Alte Konfiguration beibehalten will.

**Benutzung**

Ein Bild, das Screenshot enthält.

Automatisch generierte Beschreibung

Die Webseite der Appliance ist unter <http://localhost> zu finden. Um sich anzumelden geben Sie folgende Informationen ein:

* Benutzername: NSA
* Passwort: sml12345

Ein Bild, das Screenshot enthält.

Automatisch generierte Beschreibung

Dann kommt man auf die bisher leere Webseite.

Ein Bild, das Screenshot enthält.

Automatisch generierte Beschreibung

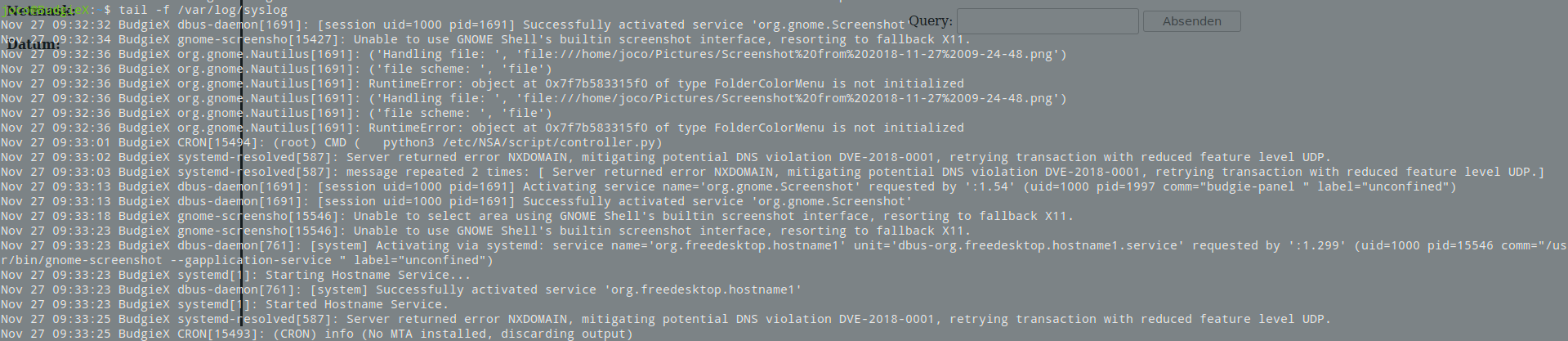
Man kann nach der Ersten erfolgreichen Ausführung im Feld oben rechts die Nummer der Ausführung eingeben, um die Resultate anzuschauen.

Ein Bild, das Screenshot enthält.

Automatisch generierte Beschreibung

**Überwachung**

Um zu überwachen ob die Appliance ausgeführt wird kann man die Systemlogs anzeigen lassen



Rot: Die Abfrage wird ausgeführt.

Blau: Die Abfrage wurde Beendet.

## Supporthandbuch

### Massnahmen bei Benutzerproblemen

Bei jedem Problem sollte man zuerst folgende Checkliste durchprüfen:

* Stimmt die Linux Version: **Ubuntu 18.10**
* Wurde die Appliance korrekt konfiguriert: **nsaconf**
* Ist die Neuste Version der NSA Installiert**: git pull im Repository und neu Installieren**.
* Hat die Scriptlib ein Problem: **python3 /etc/NSA/script/controller.py und Output analysieren**
* Meldung an das Entwicklerteam über Github: **Issue erstellen mit Fehlerbeschrieb sowie output**

### Massnahmen bei technischen Problemen

**Appliance stürzt nach Ausführung ab:**

Wenn die Appliance nicht korrekt konfiguriert wurde wird sie nicht gestartet bzw. stürzt nach dem Start ab.

Andernfalls kann es sein das die Dependecies nicht installiert sind.

Massnahme: Neu konfigurieren oder install.sh ausführen.

**Nach Korrekter Ausführung werden keine Resultate angezeigt.**

Wenn das Network Interface falsch bei der Appliance konfiguriert wurde wird nichts abgefragt

Massnahme: nsaconf und richtig konfigureren.

# Systemtest

## Testspezifikation

### Kritikalität der Funktionseinheit

|  |  |  |  |
| --- | --- | --- | --- |
| **Funktionseinheit** | **Beschreibung** | **Kritikalität** | **Begründung** |
| Frontend / Webseite | Benutzerinterface, darüber greif der Benutzer auf unsere Applikation zu | Hoch | Für den Benutzer ist die Applikation nicht mehr erreichbar, wenn diese Funktionseinheit ausfällt |
| Scriptbibliothek | Datenbeschaffung, neue Daten werden hiermit beschafft | Mittel | Es können keine neuen Daten mehr beschafft werden, Alte werden aber immer noch angezeigt |
| Datenbank | Datenspeicher, hier liegen alle Daten | Hoch | Beschaffte Daten können nicht mehr gespeichert werden, das Frontend kann nichts mehr auslesen. Die Applikation würde nicht mehr funktionieren. |

### Testanforderungen

Die Tests sollen unter Normalbedingungen mit normalen Werten durchgeführt werden. Das heisst, dass die Appliance in einem funktionierenden Netzwerk mit einem Router, einem DNS- und einem DHCP-Server durchgeführt werden sollen.

### Testverfahren

|  |  |  |  |
| --- | --- | --- | --- |
| **Test** | **Vorbereitung** | **Durchführung** | **Auswertung** |
| T-001: Cron-Job | 1. Auf Server einloggen 2. Corn-Job starten und einige Zeit warten, er sollte mehrmalig durchlaufen | 1. sudo nsaconf ausgeführt 2. cat /etc/crontab \*/5 \* \* \* \* root python3 /etc/NSA/script/controller.py 3. tail –f /var/log/syslog nach Cron-Ausführungen geschaut | Der Cron-Job sollte regelmässig gelaufen sein. Das wird mit den vorher aufgeführten Befehlen sichtbar. |
| T-002: Website-Login | Webseite aufrufen | 1. Falsches Passwort eingeben 2. Richtiges Passwort eingeben | Beim falschen Passwort sollte man nicht weitergeleitet werden, wenn das richtige eingegeben wird schon. |
| T-003: Script | Webseite aufrufen | 1. Scripts ausführen 2. Bestimmen in welchem Intervall der Zweite Durchlauf gestartet wird 3. Vor dem zweiten Durchlauf eine Veränderung am Netzwerk durchführen | Eine Fehlermeldung dies bestätig das die zwei Scripts zusammen verglichen werden. |

### Testkriterien

Abdeckungsgrad:  
Da die Appliance schlussendlich Automatisch und selbständig laufen muss, wurde so breit wie möglich getestet um Fehler zu vermeiden.

Checklisten:  
Die Checklisten sind im Kapitel 4.3 zu finden.

Ende-Kriterien:  
Ebenfalls unter Kapitel 4.3 zu finden.

### Testfälle

* Zu testen sind: Ob das Script das Netzwerk scannt, Ob das Script die Daten in die Datenbank schreibt, Ob die Website die Daten auslesen kann.
* Um die Website zu testen muss eine DB mit Testdaten erstellt werden. Das Script zu testen sollte es fertig sein (man kann auch nur die einzelnen kleinen Funktionen des Scrips testen, wenn diese bereits geschrieben wurden) und es sollten Geräte im Netzwerk sein.
* Auf der Website kann gibt es einen Button der das Script ausführt, wenn nach dem betätigen dieses Buttons Clients aufgeführt werden ist der Test erfolgreich
* Clients (ID, IP, Hostname, Devicetype, DNS, DHCP, Query), DHCP (Name, IP) und DNS(Name, IP, ON/OFF)

## Testprozedur

### Vorbereitung und Voraussetzungen

Die Grundvoraussetzung ist ein Linux Betriebssystem

**Script**

* Die einzelnen Scriptteile müssen fertig sein.
* Es müssen Geräte mit dem Netzwerk verbunden sein.
* Eine DB sollte erstellt sein um zu testen ob das Script hineinschreibt.

**Website**

* Eine DB mit Testdaten sollte erstellt sein um zu testen ob das PHP die Daten ausliest.

**Appliance**

* Und die Appliance zu testen sollte das Script und die Website fertig sein.

Konfiguration:

### Durchführung

Beim Script müssen die Verschiedenen Scriptteile ausgeführt werden

Bei der Website muss Die Website geöffnet werden und der Scanbutton gedrückt erden und schon sollten die Geräte angezeigt werden

### Nachbearbeitung

Die Testdaten wurden automatisch beim ausführen in der Datenbank gespeichert.

## Testprotokoll

### Testobjekt

|  |  |  |
| --- | --- | --- |
| ID / Bezeichnung | *T-001* | Cron-Job |
| Beschreibung | Überprüfung des Cron-Jobs in den Cron-Einstellungen | |
| Testvoraussetzung | Der Cron-Job sollte verfügbar sein. | |
| Testschritte | 1. Sudo nsaconf ausgeführt 2. cat /etc/crontab \*/5 \* \* \* \* root python3 /etc/NSA/script/controller.py 3. tail –f /var/log/syslog nach Cron-Ausführungen geschaut | |
| Erwartetes Ergebnis | Die Scripts werden regelmässig ausgeführt | |
| Tester | Joel Meier | |
| Datum Testdurchführung | 27.11.2018 | |

|  |  |  |
| --- | --- | --- |
| ID / Bezeichnung | *T-002* | Webseite Login |
| Beschreibung | Kontrolle des Logins von der Webseite | |
| Testvoraussetzung | Webseite sollte erstellt sein mit einem Login Formular | |
| Testschritte | 1. Webseite aufrufen 2. Falsches Passwort eingeben      1. Kontrollieren ob es funktioniert hat 2. Richtiges Passwort eingeben | |
| Erwartetes Ergebnis | Login nur mit dem korrektem Passwort | |
| Tester | Laxushan Yogalingam | |
| Datum Testdurchführung | 20.11.2018 | |

|  |  |  |
| --- | --- | --- |
| ID / Bezeichnung | *T-003* | Script |
| Beschreibung | Script sollte die zwei Resultate vergleichen und einen Fehler melden. | |
| Testvoraussetzung | Netzwerk | |
| Testschritte | 1. Scripts ausführen 2. Bestimmen in welchem Intervall der Zweite Durchlauf gestartet wird 3. Vor dem zweiten Durchlauf eine Veränderung am Netzwerk durchführen | |
| Erwartetes Ergebnis | Eine Fehlermeldung dies bestätig das die zwei Scripts zusammen verglichen werden. | |
| Tester | Joel Meier | |
| Datum Testdurchführung | 20.11.2018 | |

Test Ergebnis zusammengefasst:

|  |  |  |
| --- | --- | --- |
| ID | Erfolgreich | Gescheitert |
| T-001 |  |  |
| T-002 |  |  |
| T-003 |  |  |

### Testresultate

Die Testresultate sind bei uns Positiv ausgegangen. Wir haben schon in der Entwicklung unser System schon angefangen mit dem Testing. Mit der Vorgehensweise konnten wir am Schluss die Tests erfolgreich Testen.

### Testauswertung

Die Teste waren bei uns alle erfolgreich. Die meisten Tests waren für die Scripts ob diese wirklich das machen was von uns erwartet wird. Dies haben aus dem Grund gemacht, wenn die Scripts nicht richtig würde funktionieren oder falsche Werte ausgibt dann wäre unser Komplette Appliance nichts wert, weil es falsche Informationen über das Netzwerk ausgibt

# Weiterführung der Projektplanung

## Abgleich von Planung und tatsächlichem Verlauf der Phase Konzept

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Aufgabe Nr.*** | ***Aufgabenname*** | ***Dauer*** | ***Anfangsdatum*** | ***Abschlussdatum*** |
| 1 | Vorarbeiten | 6 | 14.08.2018 08:00 | 21.08.2018 16:00 |
| 2 | Initialisierung | 16 | 21.08.2018 08:00 | 11.09.2018 16:00 |
| 3 | Konzept inkl. Ferien | 31 | 11.09.2018 08:00 | 23.10.2018 16:00 |
| 4 | Datenbank Realisiert | 0 | 30.10.2018 12:00 | 30.10.2018 16:00 |
| 5 | Scrips Realisiert | 0 | 13.11.2018 12:00 | 23.11.2018 16:00 |
| 6 | Realisierung | 21 | 23.10.2018 08:00 | 27.11.2018 16:00 |
| 7 | Einführung | 11 | 20.11.2018 08:00 | 04.12.2018 16:00 |
| 8 | Projektabschluss | 11 | 04.12.2018 08:00 | 18.12.2018 16:00 |
| 9 | Präsentationen | 6 | 08.01.2019 08:00 | 15.01.2019 16:00 |

Eine Grafische Darstellung ist im Anhang zu finden.

## Aktualisierung der Risikosituation

Grün = Niedriges Risiko

Gelb = Mittleres Risiko

Rot = Hohes Risiko

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Eintrittswahrscheinlichkeit | W5 |  | 1 | 8 |  |
| W4 |  |  |  |  |
| W3 |  | 5 | 4 & 3 |  |
| W2 |  | 9 | 6 & 7 | 2 |
| W1 |  |  |  |  |
|  | S1 | S2 | S3 | S4 |
| Penis | Schadensausmass | | | | |

Abbildung 2 Risikograph

|  |  |
| --- | --- |
| Risiken Nr. |  |
| 1 | Krankheit, Unfall |
| 2 | Datenverlust |
| 3 | Abgabetermin verpassen |
| 4 | Allgemeine Netzwerkprobleme |
| 5 | Fehler in der Konzeptphase |
| 6 | Hardware Ausfall (gibbix) |
| 7 | Script Fehler (Patch) |
| 8 | Ausfall von Auftraggeber |
| 9 | Ausfall von Gibb Netzwerk |

## Planung der nächsten Phase

In der nächsten Phase, der Einführung, werden wir den Schlussbericht schreiben. Dabei gehen wir das Dokument durch und werden uns die Zeit laufend einplanen.

# Projektfreigabe

Hiermit bestätigt der Auftraggeber die Freigabe des Projekts:

­­­­­­­­­­­­­­­­­­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Der Auftraggeber Der Projektleiter

(Ort, Datum, Unterschrift) (Ort, Datum, Unterschrift)

