|  |  |
| --- | --- |
|  | |
|  |  |
|  |  |
| Praxisarbeit NMAP-Projekt | |
| Leistungsbeurteilung LB1 - Dokumentation | |
|  |  |
|  |  |
| Modul | IET-122 - Praxisarbeit |
| Eingereicht von Gruppenleiter | Vranjes Vladan  INF-122-22C-Stammgruppe 3 |
| Eingereicht bei | Urs Dummermuth |
| Datum | 02.05.2023 |

Änderungsverzeichnis

| Datum | Version | Änderung | Autor |
| --- | --- | --- | --- |
| 17.04.2023 | 0.5 | Vorlage | Urs Dummermuth |
| 02.05.2023 | 1.0 | Projektbeschreibung | Vladan Vranjes |
| 09.05.2023 | 1.1 | Ziele und Anforderungen | Vladan Vranjes |
| 16.05.2023 | 1.12 | Absätze: Skript & Programm und Usecases & Fälle | Vladan Vranjes Yanis Riedo |
| 23.05.2023 | 1.2 | Abschliessende Veränderungen | Vladan Vranjes |

Inhaltsverzeichnis

[1 Ziele und Anforderungen 3](#_Toc132625387)

[1.1 Einleitung 3](#_Toc132625388)

[1.2 Zweck des Skriptes 3](#_Toc132625389)

[1.3 Ziele 3](#_Toc132625390)

[1.4 Anforderungen 3](#_Toc132625391)

[2 Ablaufdiagramm 4](#_Toc132625392)

[2.1 PAP / Flussdiagramm / Struktogramm 4](#_Toc132625393)

[2.2 Kommentar / Beschreibung 4](#_Toc132625394)

[3 Skript/Programm (Realisierung) 5](#_Toc132625395)

[3.1 Technologie Bash / Powershell / Python 5](#_Toc132625396)

[3.2 Ein- und Ausgabe 5](#_Toc132625397)

[3.3 Kontrollstrukturen 5](#_Toc132625398)

[4 Integration und Sicherheit 6](#_Toc132625399)

[4.1 Implementierung 6](#_Toc132625400)

[4.2 Sicherheit 6](#_Toc132625401)

[4.3 Kompatibilität 6](#_Toc132625402)

[4.4 Betrieb und Wartung 6](#_Toc132625403)

[5 Usecases und Testfälle 7](#_Toc132625404)

[5.1 Usecase 7](#_Toc132625405)

[5.2 Testfall 7](#_Toc132625406)

[6 Präsentation, Dokumentation 8](#_Toc132625407)

[6.1 Demo-Video 8](#_Toc132625408)

[6.2 Vortrag 8](#_Toc132625409)

[6.3 Journal 8](#_Toc132625410)

[7 Reflexion 9](#_Toc132625411)

[7.1 Journal 9](#_Toc132625412)

[7.2 Auswertung 10](#_Toc132625413)

[7.3 Fazit 10](#_Toc132625414)

[8 Anhang 11](#_Toc132625415)

[8.1 Skript/Programm-Listing 11](#_Toc132625416)

# Ziele und Anforderungen

## Einleitung

Wir simulieren ein Netzwerk, indem wir 5 Textdateien erstellen mit IP-Adresse, MAC-Adresse und Hostname

## Zweck des Skriptes

Das Skript simuliert einen Netzwerkscan, indem es alle geforderten Daten wie MAC-Adresse, IP-Adresse (privat, Subnetz, Öffentlich) und Windows Version aus einem Text File ausliest.

Von diesen Textfiles gibt es 5, jedes dieser files simuliert einen PC und enthaltet die Informationen, die ausgelesen werden müssen. Das Skript ist darauf ausgelegt genau 5 Dateien zu lesen.

Die gesammelten Informationen werden dann übersichtlich in einem GUI dargestellt.

Würde das Skript mit einem echten Netzwerkscan arbeiten, würde es die Arbeitseffizienz steigern, indem man sich die Zeit dazu spart, die einzelnen Werte auszulesen und diese stattdessen an einem Ort sieht.

## Ziele

**Ziel 1:**

Das Skript macht einen Netzwerkscan und erstellt anhand von dem eine Übersicht, die alle verfügbaren Daten des lokalen Rechners anzeigt.

Dies ist erfüllt, wenn mindestens die Daten aufkommen, welche man in den Adaptereinstellungen des Gerätes findet.

**Ziel 2:**

Die Daten, die das Skript gesammelt hat, werden im zweiten Teil des Programmes in einem GUI angezeigt.

Das Ziel ist erfüllt, wenn die Datensätze nicht in einem Shell Fenster, sondern in einem Separaten Fenster angezeigt werden.

Ziel 3:

Das GUI ist übersichtlich und strukturiert, dass es auch eine Person mit weniger wissen benutzen kann.

Dieses Ziel ist erfüllt, sobald das man strukturiert, angeordnet Daten mit Benennung und dahinter Datensatz ablesen kann.

**Ziel 4:**

Die Suche durch das Skript ist schneller, als wenn man dies manuell machen würde.

Das Ziel ist erfüllt, sobald die für das Programm benötigte Zeit tiefer ist als der Manuelle suchaufwand eines durchschnittlichen Informatikers.

## Anforderungen

### Allgemeine Anforderungen

Das Ziel ist erreicht, sobald man das Skript auf einer Windows basierten Maschine keine Fehlercodes entstehen und diese gut und klar ablesbar sind.

Um das Programm laufen zu lassen benötigt es ein Funktionierendes Netzwerk.

Beim Anfang des Programmes erfolgt die Auswahl, welches Netz gescannt werden soll.

### Programmtechnische Anforderungen

Das Programm benötigt ein funktionsfähiges Netzwerk und auch Internetzugang.

S

# Ablaufdiagramm

## PAP / Flussdiagramm / Struktogramm

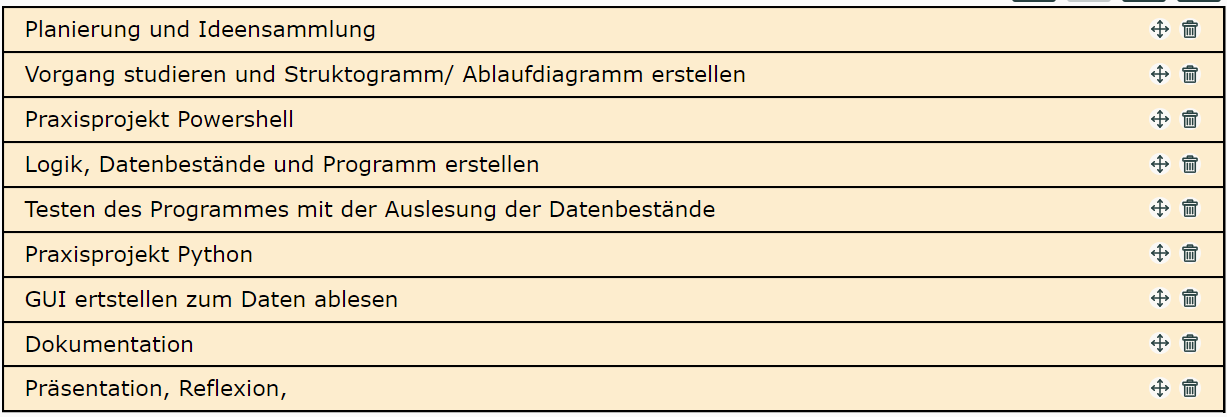


Abb. 1 Diagramm

## Kommentar / Beschreibung

Wir gehen in diesem Projekt nach der Wasserfall Methode, bei der man Schritt für Schritt alles durchgeht. Zuerst das Planen, danach Realisieren, dann Testen, Auswerten etc.

Als Verbesserungsmöglichkeit könnte man auch mit der SCRUM-Methode arbeiten, die zwar fortgeschrittener ist, jedoch auch effizienter bei meisten Arbeitsgruppen.

[SCRUM Methode kurz erklärt](https://www.agile-heroes.de/magazine/die-scrum-methode/)

# Skript/Programm (Realisierung)

## Technologie / Powershell / Python

Wir werden in diesem Modul mit Powershell ISE und Python arbeiten.

Dieses Skript wird wie die meisten neueren Programme nur mit Windows 8 und neueren Versionen und Windows Server 2019 und neuer funktionieren und kompatibel ohne Fehler laufen.

## Ein- und Ausgabe

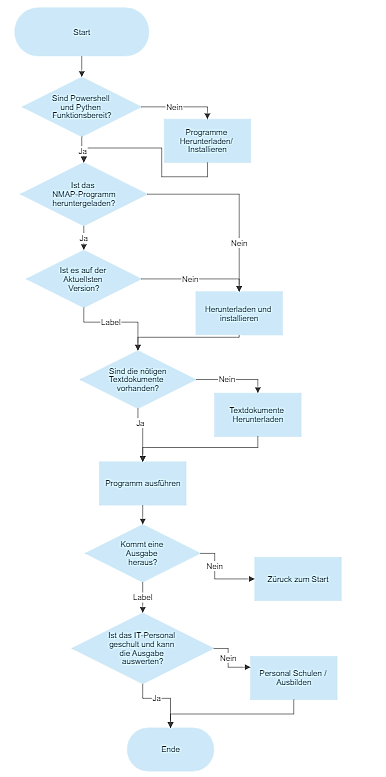
Als Eingabe sollte nur ein Knopfdruck genügen zum Starten des Programmes.

Die Ausgabe zeigt IP-Adressen, MAC und auch weitere Netzinformationen wie die öffentliche IP an.

In der ersten Version über PowerShell erfolgt die Ausgabe über die Shell, die die Daten anzeigt.

Die zweite Version mit GUI würde diese Daten in Python ablesen und grafisch darstellen.

## Kontrollstrukturen



# Integration und Sicherheit

## Implementierung

**Vor dem Gebrauch des Programmes müssen zuerst die Textdateien in den richtigen Ordner versschoben werden, damit sie vom Programm lesbar sind. Dazu mehr im beigelegten README File.**

Implementierung des Nmap-Programms:

* Installation der neuesten Version
* Gewährleistung der erforderlichen Berechtigungen
* Durchführung von Testläufen

## Sicherheit

**Dieser Abschnitt wird mit dem Gedanken geschrieben, dass das Programm mit einem echten Scan läuft.**

Sicheres Einbringen des Nmap-Programms:

* Firewall-Konfiguration anpassen damit das Gerät vom Scan erfasst wird.
* Schulung und Sensibilisierung des IT-Personals

## Kompatibilität

Das Programm ohne GUI funktioniert ausschliesslich unter Windows Betriebssystemen und auf keinen anderen Plattformen.

Das Programm mit GUI ist kompatibel mit MacOS und Windows Systemen.

## Betrieb und Wartung

Wartung des Nmap-Programms:

Aktualisierung auf die neueste Version

Schulung und Weiterbildung der Programmnutzer

# Usecases und Testfälle

**Bitte diesen Absatz genau durchlesen!**

Da man mit NMAP leicht einen Fehler machen kann, und somit gegen das Gesetz verbricht, benutzen wir keine effektive Umgebung als unsere Datenquelle des Programms, sondern selbst erstellte Textdokumente, welche IP-/MAC-Adressen, Subnetze etc. Enthalten.

Deshalb werden die Folgenden Usecases nur theoretisch gemacht, nicht wirklich ausgeführt.

Die Folgenden Usecases sind vollkommen erfunden und nicht ernst zu nehmen!

## Usecase mit Testfall 1 Netzwerküberwachung

**Umfeld:**

Ein Unternehmen mit einem internen Netzwerk möchte mithilfe eines Nmap-Programms eine Übersicht über alle Geräte und deren IP-Adressen in ihrem Netzwerk erhalten. Sie haben eine Liste von IP-Adressen und Subnetzen, die in ihrem Netzwerk verwendet werden.

**Schritte:**

* Das Nmap-Programm erhält eine Textdatei mit den IP-Adressen und Subnetzen des Netzwerks.
* Das Programm liest die Textdatei ein und analysiert die Netzwerktopologie.
* Das Programm führt einen Scan des Netzwerks durch, um Informationen über die erkannten Geräte und deren IP-Adressen zu sammeln.
* Das Programm erstellt eine Übersichtsliste, die alle erkannten Geräte mit ihren IP-Adressen auflistet.

**Testszenario:**

Angenommen, das Unternehmen hat eine Textdatei namens "network.txt" mit folgendem Inhalt:

**IP-Adressen:**

192.168.0.1

192.168.0.2

192.168.0.3

**Subnetze:**

10.0.0.0/24

172.16.0.0/16

**Erwartete Ergebnisse:**

* Das Nmap-Programm analysiert die Netzwerktopologie basierend auf den angegebenen IP-Adressen und Subnetzen.
* Das Programm führt einen Scan durch und erfasst Informationen zu den Geräten und deren IP-Adressen im Netzwerk.
* Das Programm erstellt eine Übersichtsliste, die alle erkannten Geräte mit ihren IP-Adressen auflistet.

**Beweis der Durchführung:**

Ein Screenshot der erstellten Übersichtsliste, die die erfassten Geräte und ihre IP-Adressen im Netzwerk zeigt.

## Usecase mit Testfall 2 Netzwerkstatusüberwachung

**Umfeld:**

Ein kleines Unternehmen möchte mithilfe eines Nmap-Programms den Status seiner Netzwerkgeräte überwachen. Sie haben eine Liste von IP-Adressen, die sie überprüfen möchten.

**Schritte:**

* Das Nmap-Programm erhält eine Textdatei mit den IP-Adressen der Netzwerkgeräte.
* Das Programm liest die Textdatei ein und analysiert die angegebenen IP-Adressen.
* Das Programm führt einen einfachen Ping-Scan auf den IP-Adressen durch, um den Status der Netzwerkgeräte zu ermitteln.
* Das Programm erstellt eine Liste der erreichbaren Geräte und markiert die nicht erreichbaren Geräte als offline.

**Testszenario:**

Angenommen, das Unternehmen hat eine Textdatei namens "devices.txt" mit folgendem Inhalt:

IP-Adressen:

192.168.0.1

192.168.0.2

192.168.0.3

**Erwartete Ergebnisse:**

Das Nmap-Programm analysiert die IP-Adressen und überprüft den Status der Netzwerkgeräte.

Das Programm führt einen Ping-Scan auf den angegebenen IP-Adressen durch.

Das Programm erstellt eine Liste der erreichbaren Geräte und markiert die nicht erreichbaren Geräte als offline.

**Beweis der Durchführung:**

Ein Screenshot der erstellten Liste mit den erreichbaren Geräten und deren Status, der zeigt, welche Geräte online und offline sind.

# Präsentation, Dokumentation

## Demo-Video

Free-Trial at <https://www.techsmith.com/download/camtasia/>

VLC Media Player

## Vortrag

VORTRAG/ VIDEO ERSTELLEN

## Journal

**Luan:**

17.04.2023

Mit der Gruppe sammelten wir ersten Ideen zusammen, damit wir beim nächsten mal direkt mit der Planung weiterfahren können.

Es dauerte nicht lange bis wir uns auf ein gemeinsames Projekt einigten, welches einen Netzwerkscan automatisieren und simulieren soll.

02.05.2023

Wir zeigten unsere Idee unserem Lehrer, welcher dieses Projekt auch erlaubte.

Das Projekt aber mussten wir mit vorsicht angehen, denn ein kleiner Fehler könnte schon das Gesetz brechen. Deshalb einigten wir uns auf eine Simulation, welches dies nachspielen sollte.

09.05.2023

Ich fing mit Vladan an erste Prototypen mit PowerShell zu erstellen, damit wir überprüfen können ob unsere Idee so umsetzbar ist, wie wir sie geplant haben.

Wir schrieben mehrere male das Programm um, bis wir die richtigen Befehle für genau unsere Situation fanden.

Statt einem richtigen Netzwerkscan, welches sich gesetzlich im Graubereich befindet, benutzten wir stattdessen Textdokumente welche als PC dienen sollten, die im Netzwerk auffindbar wären. Diese Textdokumente sind mit wichtigen PC-Daten gefüllt wie IP-Adresse, Subnetz, MAC etc.

16.05.2023

Nachdem der Prototyp gelungen war, gab ich das Programm direkt Vladan und Yanis, damit sie es Testen können auf verbesserungsmöglichkeiten. Diese würden wir dann auch im richtigen Programm implementieren.

Ich bekam die Testergebnisse von ihnen zurück und fing direkt damit an, das Hauptprogramm zu schreiben, welches dann auch mit GUI bedienbar sein soll.

Dieses schrieb ich in Python, da es am besten ist für Automatisationen und grafische Darstellungen.

23.05.2023

An den Onlinekonferenzen teilten wir möglichkeiten zur Verbeserung unseres Programmes und machte die finalen Abstimmungen an diesem. Hauptsächlich beschäftigte ich mit damit, das GUI so benutzerfreundlich und übersichtlich zu gestalten wie nur möglich.

Endzeit bis Abgabe:

In der Freizeit habe ich meinen Teil des Scriptes in einem Video zusammengefasst und habe dieses mit Vladan Editiert, sodass wir am Ende ein vollständiges Video haben, welches das Script vollständig erklärt.

**Vladan:**

17.04.2023

Wir sammelten unseren ersten Ideen und sammelten sie zusammen, sodass wie zügig anfangen können mit der Teamarbeit.

02.05.2023

Wir Präsentierten unsere Idee unserem Lehrer Herrn Dummermuth und haben damit weitergemacht unsere Konzepte aufzubauen und unseren Vorgang zu planen.

Als Gruppe arbeiteten wir hauptsächlich daran, wie wir alles nach der Wasserfall-Methode planen können, um spätere Komplikationen zu vermeiden.

09.05.2023

Mit Janis habe ich an der Dokumentation weitergearbeitet, dabei haben wir unseren bestehenden Texte und Vorgänge and dem Bewertungsraster angepasst.

Auch habe ich mit Luan erste Prototypen zu unserem Script geschrieben und angeschaut.

16.05.2023

Nachdem der erst funktionierende Prototyp gelungen war, habe ich dieses direkt mit Yanis anfangen zu testen, damit wir in den späteren Testfällen unsere Forschritte sehen können und auch Verbesserungsmöglichleiten sehen. Dieser Prototyp lief vorerst auf PowerShell ohne jegliche GUI zu verwenden. Dies war für mich eher eine Herausforderung, doch konnte ich von diesem Teil etwas zum Hauptprogramm hinzufügen.

23.05.2023

Bei den Online Konferenzen haben wir jeweils unsere verbesserungsmöglichkeiten miteinander geteilt, dabei nahm ich die Verbesserungen in der Dokumentation in angriff und lud die entsprechend benötigten Dateien hoch.

Endzeit bis Abgabe:

In der Freizeit habe ich meinen Teil des Scriptes in einem Video zusammengefasst und habe dieses mit Luan Editiert, sodass wir am Ende ein vollständiges Video haben, welches das Script vollständig erklärt.

**Janis:**

17.04.2023

Wir haben zusammen Ideen gesammelt und dann ausgewählt. Auch haben wir überlegt, wie wir unser Projekt realisieren wollen und können.  
02.05.2023

Heute haben wir weiter an unserem Projekt gearbeitet und die Dokumentation angefangen. Wir haben mir Herr Dummermuth unser Projekt besprochen.

09.05.2023

Ich habe mit Vladan zusammen die Dokumentation weitergeschrieben und die Punkte, die wir bereits hatten, habe ich auf die Kriterien des Bewertungsrasters angepasst.

16.05.2023

Ich habe weiter an der Dokumentation gearbeitet und die Dinge die nicht richtig geschrieben, ausformuliert oder nicht vollständig waren korrigiert.

23.05.2023

Im Onlineunterricht haben wir noch letzte Verbesserungen an der Dokumentation besprochen und diese dann in die Dokumentation geschrieben.

**Yanis:**

17.04.2023  
Beim ersten Mal hatten wir noch nicht viel am Projekt gearbeitet, wir haben heute mehr versucht eine gute Projektidee im Team zu finden und zu überlegen, ob wir diese realisieren können und was wir anpassen müssten, um dies zu machen. Wir haben ende Lektion jedoch noch nicht voll sicher ein Projektthema auswählen können. Die Zeit war an dem Nachmittag auch etwas knapper.

02.05.2023  
Vorerst haben wir mit der Idee unseres Projektes weitergemacht und bei einem Gespräch mit Herr Dummermuth unsere Projektidee und wie weit wir sie realisieren können angepasst. So konnten wir dann aber erstmals mit der Realisierung anfangen, ich habe aber mich vor allem auf die Dokumentation mit Vladan konzentriert. Im Team konnten wir dann die Ziele und Anforderungen noch definieren und Vladan schrieb sie nieder. So konnte der Tag gut abgeschlossen werden und ich kam gut voran.

09.05.2023  
Heute habe ich wieder viel an der Doku gearbeitet, aber auch parallel mit unserem Produkt. Ich habe die ersten Testfälle definiert und aufgeschrieben und dann auch mit unserem Produkt diese getestet und dokumentiert. Dies verlief relativ gut und ich kam so gut voran  
und hatte mein Tagesziel erreicht.

16.05.2023  
In der ersten Hälfte des Nachmittags habe ich nochmal die Testfälle angepasst mit Vladan um sie am Produkt zutreffender zu gestalten. Dies haben wir sehr gut geschafft und waren nun relativ schnell zufrieden mit unseren Testfällen.

23.05.2023

Am letzten Nachmittag der Projektarbeit haben wir zusammen die Dokumentation angeschaut und allgemeine Verbesserungsvorschläge für das gesamte Projekt besprochen.

Endzeit bis Abgabe  
Viel mehr habe ich nicht gemacht, da ich operiert wurde und so sowieso in der Zeit nicht viel gemacht habe.

**Teamarbeit:**

Anfangs haben wir nach Ideen gesucht, die gut zur Automatisation sind und für uns umsetzbar sind. Die Idee war nach einer kleiner Absprache Runde gefunden und wir fingen an uns darüber Gedanken zu machen wie und in welcher Programmiersprache wir das Programm machen. Wir mussten aber auch schauen dass wir unser Projekt gut und effektiv umsetzen konnten, so haben wir dann verschiedene Aufgaben im Team aufgeteilt.

Wir suchten mehrere Methoden, die beste davon war, die Daten auszulesen mit PowerShell ISE und danach diese mit Python in einem GUI darzustellen.

# Reflexion

## Journal

| **Tag** | **Tätigkeit** | **Person** | **Bemerkungen** |
| --- | --- | --- | --- |
| 1 | Beschreibung | Kürzel | Erfolge, Probleme oder Schwierigkeiten. Hilfestellung. Reflexion über das Vorgehen, die Resultate. Fazit. Nächste Schritte |
| 2 | Struktogramm erstellen & erste Ideen gesammelt | Ganzes  Team | Struktogramm von Vladan |
| 3 | Abschnitte in Doku:   * Ziele & Anforderungen * Ablaufdiagramm * Vorbereitung Testfälle | Ganzes Team | Ziele & Anforderungen Janis  Ablaufdiagramm Vladan  Testvorbereitungen Yanis |
| 4 | Abschnitte erledigt:   * Usecases * Zweck des Skriptes * Anforderungen * Skripting | Ganzes Team | Usecases und Testfälle Vladan  Zweck des Skriptes Vladan  Skripting Luan |
| 5 | * Skripting * Kontrollstrukturen * Einleitung |  |  |
| 6 | Überarbeitung der Programmabhängigen Texte |  |  |
| 7 |  |  |  |
| 8 |  |  |  |
| 9 |  |  |  |

## Auswertung

Das Projekt zur Erstellung eines Python-Skripts mit GUI zur Darstellung von Informationen aus Textdokumenten wie IP-Adresse, MAC-Adresse und Subnetz war äußerst erfolgreich.

Hier sind einige der positiven Aspekte, die ich aus diesem Projekt ziehe:

* Effektive Datenvisualisierung: Das Python-Skript mit GUI hat es ermöglicht, die Informationen aus den Textdokumenten anschaulich darzustellen. Die Benutzeroberfläche bot eine benutzerfreundliche Möglichkeit, die Daten zu präsentieren.
* Flexibilität und Anpassungsfähigkeit: Das Skript konnte problemlos mit verschiedenen Textdokumenten arbeiten und die enthaltenen Informationen extrahieren. Es bot die Möglichkeit, verschiedene Datenquellen zu verwenden und die Darstellung entsprechend anzupassen.
* Zeiteinsparungen und Effizienzsteigerung: Durch die Automatisierung des Prozesses konnten wir Zeit sparen und die Effizienz steigern. Die manuelle Extraktion und Darstellung der Informationen aus den Textdokumenten wurde durch das Skript vereinfacht und beschleunigt.

Insgesamt war das Projekt ein großer Erfolg und hat dazu beigetragen, die Verarbeitung und Visualisierung von Netzwerkinformationen aus Textdokumenten zu verbessern.

## Fazit

Zwar haben wir nicht mit richtigen Netzwerkscans gearbeitet, doch aber lernten wir andere wertvolle Lektionen. Zum Beispiel wie man mit PowerShell und Python skriptet und auch Daten grafisch darstellen lassen.

Auch lernten wir dazu, wie man die Teamarbeit fördert und jeden dazu bringt etwas Neues für sich zu entdecken.

An manchen Stellen war es zwar etwas nervenbelastend, da Fehler immer wieder auftauchten, doch lernten wir auch unsere Geduld weiterzubilden.

# Anhang

## Skript/Programm-Listing

#Dieses Script Wurde im Modul 122 Erstellt und sollte Anhand von .txt Dateien ein Netzwerk Simulieren. Diese Dateien werden dann in Python Ausgelesen und in einem GUI dargestellt.  
  
#Modul Zeit Importieren  
import time  
  
#Scannen des Inhalts der simulierten PCs (Textdateien)  
datei = open('PC01.txt', 'r') #datei.read() zum auslesen  
datei1 = open('PC02.txt', 'r')  
datei2 = open('PC03.txt', 'r')  
datei3 = open('PC04.txt', 'r')  
datei4 = open('PC05.txt', 'r')  
  
#Endlosschlaufe mit Usereingabe  
i = 1  
while i == 1:  
#User Entscheidet sich ob er den Scann durchführen will  
Jaodernein = input("Willst du das Netzwerk Scannen (J/N)")  
if Jaodernein == "J": #Wenn er den scann durchführen will dann wird Gescannt  
# Simulierung für ein erlebniss eines Scanns  
print("Netzwerk-Scann wird Ausgeführt...")  
time.sleep(3)  
print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")  
print("5 Geräte Gefunden...")  
time.sleep(3)  
print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")  
print("Informationen Auslesen...")  
time.sleep(3)  
print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")  
  
# GUI erstellen  
from tkinter import \*  
  
from tkinter import ttk  
  
window = Tk()  
  
# Fenstergrösse setzen  
window.geometry('750x500')  
  
# Titel setzen  
window.title("Welcome to LikeGeeks app")  
  
# Verschiedene Seiten Erstellen  
tab\_control = ttk.Notebook(window)  
  
tab1 = ttk.Frame(tab\_control)  
  
tab2 = ttk.Frame(tab\_control)  
  
tab3 = ttk.Frame(tab\_control)  
  
tab4 = ttk.Frame(tab\_control)  
  
tab5 = ttk.Frame(tab\_control)  
  
# Titel der Seiten definieren  
tab\_control.add(tab1, text='PC01')  
  
tab\_control.add(tab2, text='PC02')  
  
tab\_control.add(tab3, text='PC03')  
  
tab\_control.add(tab4, text='PC04')  
  
tab\_control.add(tab5, text='PC05')  
  
# Inhalt der Seiten von den Simulierten PCs auslesen  
lbl1 = Label(tab1, text=datei.read())  
  
lbl1.grid(column=0, row=0)  
  
lbl2 = Label(tab2, text=datei1.read())  
  
lbl2.grid(column=0, row=0)  
  
lbl3 = Label(tab3, text=datei2.read())  
  
lbl3.grid(column=0, row=0)  
  
lbl4 = Label(tab4, text=datei3.read())  
  
lbl4.grid(column=0, row=0)  
  
lbl5 = Label(tab5, text=datei4.read())  
  
lbl5.grid(column=0, row=0)  
  
tab\_control.pack(expand=1, fill='both')  
  
#Fenster Permanent Ersichtlich Machen bis es geschlossen wird  
window.mainloop()  
break  
elif Jaodernein == "N": #Wenn der User Den Scann nicht Durchführen will ist das Programm Beendet  
print("Kein Netzwerk-Scann wird durchgeführt")  
break  
else: #Wenn der User etwas Anderes als J oder N eingibt dann fragt es ihn nochmal ob er Scannen will (Fehlerbehebung)  
print("Ungültige eingabe")