



---

# Projektplan

## Studienarbeit FS-2020

25. Februar 2020

---

*Autoren:*

Mike SCHMID  
mike.schmid@hsr.ch

Janik SCHLATTER  
janik.schlatter@hsr.ch

*Supervisor:*

Prof. Stettler BEAT  
beat.stettler@hsr.ch

Dieses Werk einschließlich seiner Teile ist **urheberrechtlich geschützt**. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Autors unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen sowie die Einspeicherung und Verarbeitung in elektronischen Systemen.

---

## Zweck

Dieses Dokument beschreibt den Projektplan und liefert eine Übersicht über das Projekt Network Unit Testing System, dessen Planung und Organisation, sowie über weitere Bereiche des Projektaufbaus. Der Projektplan dient als Grundlage und Referenz für nachfolgende Projektdokumente

## Änderungsgeschichte

| Datum      | Version | Änderung      | Autor           |
|------------|---------|---------------|-----------------|
| 20.02.2018 | 1.0     | Initial Setup | Janik Schlatter |

## Inhaltsverzeichnis

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Einführung</b>                      | <b>1</b> |
| 1.1      | Sprache . . . . .                      | 1        |
| 1.2      | Referenzen . . . . .                   | 1        |
| 1.3      | Vorarbeit NUTS . . . . .               | 1        |
| <b>2</b> | <b>Projektübersicht</b>                | <b>2</b> |
| 2.1      | Projektübersicht . . . . .             | 2        |
| 2.2      | Zweck und Ziel . . . . .               | 2        |
| 2.3      | Lieferumfang . . . . .                 | 2        |
| 2.4      | Annahmen und Einschränkungen . . . . . | 2        |
| <b>3</b> | <b>Projektorganisation</b>             | <b>3</b> |
| 3.1      | Projektmitglieder . . . . .            | 3        |
| 3.2      | Externe Schnittstellen . . . . .       | 3        |
| <b>4</b> | <b>Management Abläufe</b>              | <b>4</b> |
| 4.1      | Kostenvoranschlag . . . . .            | 4        |
| 4.2      | Zeitliche Planung . . . . .            | 4        |
| 4.3      | Phasen/Iterationen . . . . .           | 4        |
| 4.4      | Meilensteine . . . . .                 | 5        |
| 4.5      | Besprechungen/Protokolle . . . . .     | 5        |
| <b>5</b> | <b>Risikomanagement</b>                | <b>7</b> |
| 5.1      | Risiken . . . . .                      | 7        |
| 5.2      | Umgang mit Risiken . . . . .           | 7        |
| <b>6</b> | <b>Infrastruktur</b>                   | <b>8</b> |
| 6.1      | Übersicht der Tools . . . . .          | 8        |
| <b>7</b> | <b>Qualitätsmassnahmen</b>             | <b>9</b> |
| 7.1      | Allgemein . . . . .                    | 9        |
| 7.2      | Testing . . . . .                      | 9        |
| 7.3      | Besprechungen . . . . .                | 9        |
| 7.4      | Versionskontrolle . . . . .            | 9        |
| 7.5      | Dokumente . . . . .                    | 9        |
| 7.6      | Code-Qualität . . . . .                | 9        |

## 1 Einführung

### 1.1 Sprache

Die allgemeine Projektsprache (Dokumentation, Use Cases, etc.) wird in Deutscher Schriftsprache verfasst. Der Code, das GitHub-Repository und die Versionskontrolle wird in Englischer Sprache geschrieben.

### 1.2 Referenzen

Alle Dokumente werden auf dem GitHub-Repository abgelegt und verwaltet.

Git Repository     <https://github.com/EkoGuandor229/Network-Unit-Testing.git>  
Vorarbeit NUTS    <https://github.com/HSRNetwork/Nuts.git>

### 1.3 Vorarbeit NUTS

Die Studienarbeit aus dem Jahr 2016 hat ein Programm erarbeitet, die mit SaltStack, einer Lösung für die Automatisierung von Projekten, das Testen von Netzwerkumgebungen mittels Python ermöglicht. Dabei wurden umfangreiche Tests in der Serialisierungssprache YAML spezifiziert und umgesetzt. Die Schwierigkeit lag vor allem darin, dass nicht jedes Netzwerkgerät dieselben Funktionen für spezifische Befehle bietet, da Hersteller Unterschiedliche Befehle für ihre Geräte verwenden. Darauf aufbauend wird in dieser Arbeit die Testdefinition weiter verwendet und nach Bedarf ergänzt oder ausgebaut.

## 2 Projektübersicht

### 2.1 Projektübersicht

### 2.2 Zweck und Ziel

Das Testen von Netzwerkkonfigurationen findet auch heute noch hauptsächlich mit handgeschriebenen CLI-Befehlen oder kleinen Skripten statt. Wenn der Netzwerktechniker einen Test vergisst, oder die Formulierung nicht stimmt, kann es vorkommen, dass im Netzwerk Fehler auftreten, deren Ursprung schwierig zu ermitteln ist und eine komplette Repetition der (handgeschriebenen) Tests erfordert. Ein Programm, welches wie in der Softwareentwicklung vordefinierte und automatisch durchgeführte Tests, sogenannte Unit-Tests, ermöglicht, könnte diese Probleme stark verringern. Dabei können zwei grobe Arbeitsvorgänge beschrieben werden. Im ersten schreibt ein Netzwerktechniker Tests, die ein bestehendes Netzwerk möglichst genau abbilden/beschreiben sollen. Die Tests lassen sich jederzeit durchführen und testen den Zustand und die Konfiguration des Netzwerks. Falls nun ein Fehler auftritt, können die Tests automatisiert durchgeführt werden und dann, vorausgesetzt sie sind vollständig, sollte der Report aufzeigen, was genau schiefgegangen ist und wo der Fehler liegt. Der zweite Arbeitsvorgang entspricht dem in der Softwareentwicklung gängigen Test-Driven-Development (TDD). Beim TDD werden Tests geschrieben, bevor das System verändert wird oder bevor man neuen Code schreibt. Auf ein Netzwerk abstrahiert könnte beispielsweise ein Administrator, der eine Änderung am Netzwerk vornehmen will, zuerst die Tests schreiben, welche die Änderung testen sollen. Danach werden die Konfigurationen verändert und die Tests durchgeführt. Falls die Tests nun fehlschlagen, kann man die Konfiguration anpassen oder sogar auf einen früheren Zustand zurücksetzen. Beide Arbeitsvorgänge erleichtern die Fehlersuche und erhöhen die Stabilität des Netzwerks.

### 2.3 Lieferumfang

Im Rahmen der Studienarbeit wird folgendes erstellt:

- Ein Framework mit blablabla

### 2.4 Annahmen und Einschränkungen

## 3 Projektorganisation

### 3.1 Projektmitglieder

| Name            | Email  |
|-----------------|--|
| Janik Schlatter | <a href="mailto:jschlatt@hsr.ch">jschlatt@hsr.ch</a> |
| Mike Schmid     | <a href="mailto:mschmid@hsr.ch">mschmid@hsr.ch</a>   |

### 3.2 Externe Schnittstellen

| Name          | Email  | Zuständigkeit |
|---------------|--|---------------|
| Beat Stettler | <a href="mailto:beat.stettler@hsr.ch">beat.stettler@hsr.ch</a>     | Betreuer      |
| Urs Baumann   | <a href="mailto:urs.baumgartner@hsr.ch">urs.baumgartner@hsr.ch</a> | Betreuer      |

## 4 Management Abläufe

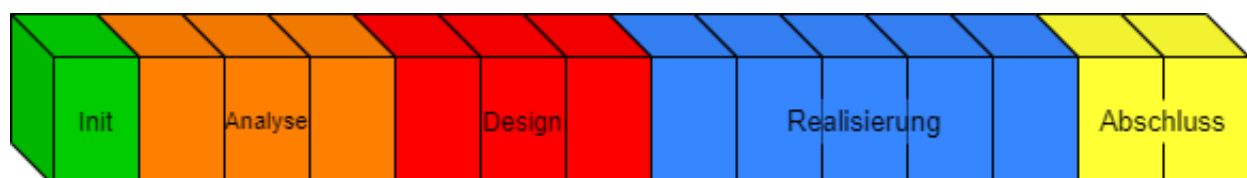
### 4.1 Kostenvoranschlag

Das Projekt wurde am 20.02.2020 gestartet und wird voraussichtlich am 28.05.2020 enden. Das heisst, es stehen 14 Wochen Zeit zur Verfügung während dem Semester. Jedes Projektmitglied arbeitet insgesamt 240 Stunden an dem Projekt, sprich 16 Stunden pro Woche pro Projektmitglied, inklusive sieben Dokumentationstermine jede zweite Woche mit ca 2-4 Stunden.

|                                     |            |
|-------------------------------------|------------|
| Projektdauer                        | 14 Wochen  |
| Anzahl Projektmitglieder            | 2          |
| Arbeitsstunden pro Woche und Person | 16         |
| Arbeitsstunden insgesamt            | 480        |
| Projektstart                        | 20.02.2020 |
| Projektende                         | 28.05.2020 |

### 4.2 Zeitliche Planung

Die 14 Wochen des Projekts werden in fünf Phasen unterteilt: Initialisierung, Analyse, Design, Realisierung und Abschluss.



### 4.3 Phasen/Iterationen

#### Phasen

Wir halten uns an die folgenden 5 Phasen:

| Farbe* | Bezeichnung     | Zeitraumen |
|--------|-----------------|------------|
| Grün   | Initialisierung | 1 Woche    |
| Orange | Analyse         | 3 Wochen   |
| Rot    | Design          | 3 Wochen   |
| Blau   | Realisierung    | 5 Wochen   |
| Gelb   | Abschluss       | 2 Wochen   |

## Iterationen

Die Iterationen werden wöchentlich gemacht. Da wir auch ein Mal wöchentlich das Meeting haben passt das gut aufeinander.

## 4.4 Meilensteine

| Nr | Bezeichnung             | Termin        | Beschreibung   |
|----|-------------------------|---------------|--|
| M1 | Projektplan             | So 01.03.2020 | Grundentwurf der Requirements, Risikoanalyse & -management, Projektorganisation, Managementabläufe, Infrastrukturentwurf, Qualitätsmassnahmen Grundentwurf.  |
| M2 | Requirements            | So 15.03.2020 | Ausgearbeitete Requirements, Nicht-funktionale Anforderung, Zu verwendende Tools und Schnittstellen beschrieben.   |
| M3 | Prototyp                | So 05.04.2020 | Architektur festgelegt, Schnittstellen angelegt, Architekturdokumentation, Testprozeduren eingerichtet und Unit-Tests erstellt, Erster lauffähiger Prototyp. |
| M4 | Feature Freeze          | So 03.05.2020 | Hauptfunktionalität der Software implementiert, Bugs sind bekannt und Dokumentiert, Codedokumentation zu 60% fertiggestellt.                                 |
| M5 | Codefreeze & Codeabgabe | So 17.05.2020 | Bugfixes erstellt, Tests sind alle erfolgreich, Codedokumentation zu 60% fertiggestellt.   |
| M6 | Projektabgabe           | So 24.05.2020 | Dokumentation fertiggestellt und abgegeben.  |

## 4.5 Besprechungen/Protokolle

Es wurden zwei Termine vereinbart, an welchen sich die Projektmitglieder treffen. Bei beiden Terminen stehen jeweils mindestens 6 Lektionen zur Verfügung.



*4 Management Abläufe*

---

| Nr | Wann                     | Beschreibung                            |
|----|--------------------------|---|
| 1  | Dienstag 10:00 - 17:00   | Gemeinsame Arbeit der Projektmitglieder |
| 2  | Donnerstag 08:00 - 17:00 | Gemeinsame Arbeit der Projektmitglieder |
| 3  | Donnerstag 14:00 - 15:00 | Besprechung mit Projektbetreuern        |

---

## 5 Risikomanagement

### 5.1 Risiken

Eine Risikoanalyse mit gewichtetem Schaden und Informationen zur Vorbeugung ist auf der Ablage zu finden (siehe Dokument TechnischeRisiken.xlsx)

### 5.2 Umgang mit Risiken

Um Probleme gerade während der Init/Analyse Phase möglichst früh zu erkennen, arbeiten wir wöchentlich zwei Tage nebeneinander, um uns über mögliche Probleme auszutauschen. Desweiteren suchen wir auch den Kontakt zum Betreuer sobald Unklarheiten im Team herrschen.

## 6 Infrastruktur

Alle Arbeiten zum Projekt werden von den Projektmitgliedern auf Ihrem jeweiligen Laptop verrichtet.  
(Evt erhalten wir noch ein physikalische Netzwerk um zu testen?!?!?)

### 6.1 Übersicht der Tools

Für die Umsetzung des Projektes werden folgende Tools verwendet:

| Bezeichnung | Beschreibung         |
|-------------|----------------------|
| Git         | Versionsverwaltung   |
| PyCharm     | Entwicklungsumgebung |

## **7 Qualitätsmassnahmen**

### **7.1 Allgemein**

### **7.2 Testing**

### **7.3 Besprechungen**

### **7.4 Versionskontrolle**

Sämtliche Dokumente werden in einem Git Repository abgelegt. Damit wird, dank der Versionskontrolle, jede Änderung nachvollziehbar sein und es können auf sämtlichen alten Versionen zugegriffen werden.

### **7.5 Dokumente**

### **7.6 Code-Qualität**