



Anforderungsanalyse

Studienarbeit FS-2020

10. März 2020

Autoren:

Mike SCHMID
mike.schmid@hsr.ch

Janik SCHLATTER
janik.schlatter@hsr.ch

Supervisors:

Prof. Stettler BEAT
beat.stettler@hsr.ch

Baumann URS
urs.baumann@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.

Änderungsgeschichte

Datum	Version	Änderung	Autor
27.02.2020	1.0	Initial Setup	Janik Schlatter

Inhaltsverzeichnis

1	Allgemeine Beschreibung	1
2	Use Cases	2
2.1	Personas	2
2.2	Use Cases Brief	2
2.2.1	Tests CRUD	2
2.2.2	Device erfassen	2
2.2.3	Command erfassen	2
2.2.4	Ergebnis erfassen	2
2.2.5	Tests ausführen	3
2.2.6	Logs	3
2.3	Use Case Diagramm	3
3	Nichtfunktionale Anforderungen	4
3.1	Änderbarkeit	4
3.1.1	Analysierbarkeit	4
3.1.2	Modifizierbarkeit	4
3.1.3	Stabilität	4
3.1.4	Testbarkeit	4
3.1.5	Szenario: Neue Netzwerkschnittstelle	4
3.1.6	Szenario: Verständlichkeit von generiertem Code	5
3.2	Scenario: Schnelle Fehlerlokalisierung	5
3.3	Benutzbarkeit	6
3.3.1	Verständlichkeit	6
3.3.2	Erlernbarkeit	6
3.3.3	Bedienbarkeit	6
3.3.4	Szenario: Einfachheit der Testdefinitionen	6
3.4	Sicherheit	7
4	Weitere Anforderungen	8
4.1	Schnittstellen	8
4.2	Randbedingungen	8

1 Allgemeine Beschreibung

2 Use Cases

2.1 Personas

Person	Beschreibung	Technisches Wissen
Net-Admin	Der Netzwerk-Administrator ist der Chef über das ganze Netzwerk und trägt die ganze Verantwortung darüber.	Der Netzwerk-Administrator hat grundlegendes Wissen über Python.
Net-Engineer	Der Netzwerk-Engineer ist die Person, welche das Netzwerk aufgesetzt hat und es am laufen hält.	Der Netzwerk-Engineer hat Wissen über Python und sollte im Stande sein, das Programm in Betrieb zu nehmen.
Net-Techniker	Der Netzwerk-Techniker unterstützt den Netzwerk-Engineer bei der Wartung und erledigt den Support.	Der Netzwerk-Techniker hat kein Wissen über Python. Er ist nur im Stande das Programm zu benutzen, aber nicht aufzusetzen.

2.2 Use Cases Brief

2.2.1 Tests CRUD

Der User kann Tests mit der definierten Sprache erstellen.

2.2.2 Device erfassen

Für einen Test sind ein- oder mehrere Devices notwendig. Devices haben Eigenschaften wie beispielsweise Name, IP-Adresse, Device-Typ (Router, Switch,...) und Login Daten. Diese Devices müssen mittels einem Setup definiert werden.

2.2.3 Command erfassen

Sobald man ein Device erfasst hat, möchte man Kommandos auf diesem ausführen. Dies könnten beispielsweise show-Befehle oder andere Programme zum Senden von Daten sein. Diese Kommandos sind als Queries auf die Devices zu verstehen.

2.2.4 Ergebnis erfassen

Sobald Devices und Commands erfasst wurden, kann nun das erwartete Ergebnis formuliert werden.

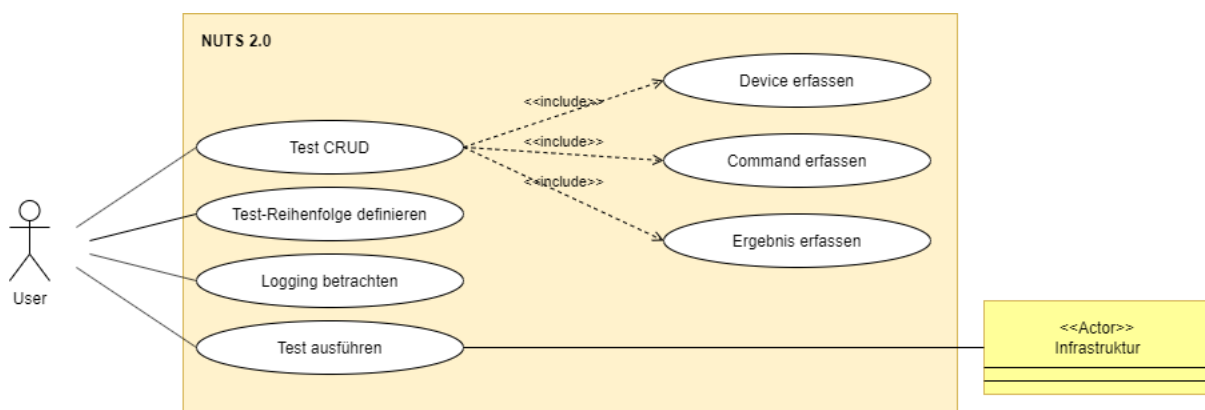
2.2.5 Tests ausführen

Ein fertig formulierter Test kann ausgeführt werden. Mit diesem Vorgang wird die Verbindung zum Device aufgebaut und das im Test definierte Kommando ausgeführt.

2.2.6 Logs

Die Auswertung der jeweiligen Durchführung der Tests wird in einem Logging gespeichert, welches der User jederzeit ansehen kann.

2.3 Use Case Diagramm



3 Nichtfunktionale Anforderungen

In diesem Kapitel behandeln wir die nichtfunktionalen Anforderungen an das Projekt. Es werden Aspekte und Anforderungen aus den Bereichen Änderbarkeit, Benutzbarkeit, Effizienz, Zuverlässigkeit, Betriebbarkeit und Sonstige Qualitätsanforderungen. Die jeweiligen Aspekte werden in ihren Unterkapiteln genauer beschrieben. Es werden mögliche Szenarien beschrieben, die in der Erstellung oder dem Betrieb der Software auftreten können.

3.1 Änderbarkeit

Aufwand, der zur Durchführung von vorgegebenen Änderungsarbeiten benötigt wird. Unter Änderungen gehen Korrekturen, Anpassungen oder Veränderungen der Umgebung, Anforderungen oder funktionalen Spezifikation. Gem. ISO 9126 gehören zur Änderbarkeit folgende Teilmerkmale:

3.1.1 Analysierbarkeit

Aufwand, der benötigt wird, um das System zu verstehen, z.B. um Ursachen von Versagen oder Mängel zu diagnostizieren oder Änderungen zu spezifizieren.

3.1.2 Modifizierbarkeit

Wie leicht lässt sich das System anpassen, um Verbesserungen oder Fehlerbeseitigungen durchzuführen.

3.1.3 Stabilität

Wahrscheinlichkeit, dass mit Änderungen unerwartete Nebenwirkungen auftreten.

3.1.4 Testbarkeit

Wie gross wird der Aufwand, bei Änderungen die Software zu prüfen.

3.1.5 Szenario: Neue Netzwerkschnittstelle

Wenn zum bestehenden System eine neue Netzwerkschnittstelle definiert werden soll, so muss die dafür notwendige Software innerhalb von einer Arbeitswoche entwickelt, integriert und in Betrieb genommen werden können.

Geschäftsziele

Flexibilität und Reaktionsfähigkeit bei Änderungswünschen

Auslöser

Entwickler möchte für eine nicht vorhandene Netzwerkschnittstelle eine Erweiterung erstellen, z.B. Einbindung von OpenConfig.

Reaktion

Betreiber der Software können gem. ihren eigenen Anforderungen an die Netzwerkschnittstellen auf Basis vordefinierter Erweiterungspunkte das System erweitern.

Zielwert

Erweiterungen der Netzwerkschnittstellen sind innerhalb von 40 Personenstunden umsetzbar

3.1.6 Szenario: Verständlichkeit von generiertem Code

Generierter Code für die Testsuites und Testfälle ist leicht verständlich und manuell anpassbar.

Qualitätsziele

Verständlichkeit, Testbarkeit, Modifizierbarkeit

Geschäftsziel(e)

Tester können den generierten Code für die Testsuites und die Testfälle leicht verstehen und ihren eigenen Bedürfnissen anpassen.

Reaktion

Testcode (Python pytest-Code) für bestimmte Teile des Systems wird durch den Test-Generator in möglichst einfacher Form erzeugt.

Zielwert

Ein Tester kann den generierten Quellcode für einen Testfall in weniger als 30 Minuten verstehen und einfache Änderungen daran vornehmen.

3.2 Szenario: Schnelle Fehlerlokalisierung

Die Ursache von fehlgeschlagenen Tests (Software-Unittests) lässt sich in kurzer Zeit lokalisieren.

Geschäftsziele

Kurze Fehlerbehebungszeiten, Änderbarkeit, geringes Risiko bei Änderungen

Reaktion

Wenn ein Test fehlschlägt, kann ein Entwickler aufgrund von Fehler- und/oder Log-Nachrichten die Ursache in kurzer Zeit lokalisieren.

Zielwert

Fehlerlokalisierung findet durchschnittlich in unter 10 Minuten statt.

3.3 Benutzbarkeit

Zeitlicher Aufwand, der für die Erlernung der Benutzung des Programms benötigt wird. Die User werden hierfür in spezifische Nutzergruppen mit festgelegten Fähigkeiten unterteilt. Teilmerkmale:

3.3.1 Verständlichkeit

Aufwand für den Nutzer, die Konzepte und Menüführung der Anwendung zu verstehen.

3.3.2 Erlernbarkeit

Aufwand für den User, sich ohne Vorwissen in das System einzuarbeiten.

3.3.3 Bedienbarkeit

Aufwand für den Benutzer, die Anwendung zu bedienen.

3.3.4 Szenario: Einfachheit der Testdefinitionen

Die Definitionen von Tests in YAML sind so aufgebaut, dass ein User in kurzer Zeit die Struktur und den Aufbau versteht und eigene Tests implementieren kann.

Geschäftsziele

TODO FINISH ZHIS

3.4 Sicherheit

Um auf Devices verbinden zu können, muss man sich auf diesen authentifizieren. Administrator Zugänge müssen deshalb best möglichst geschützt sein. Die Übertragung muss verschlüsselt sein (SSH) und die Passwörter dürfen, wenn überhaupt, nur mittels sicherem Hashverfahren abgelegt werden.

TODO Update

4 Weitere Anforderungen

4.1 Schnittstellen

NUTS hat verschiedene interne Schnittstellen, welche hier aufgezeigt werden:

Schnittstelle	Beschreibung
Benutzerschnittstelle	TODO
Netzwerkschnittstelle	TODO

4.2 Randbedingungen