# Anforderungsanalyse

Um den Aufwand für Planung und Implementierung einschätzen zu können, ist eine genaue Analyse der verschiedenen Anforderungen an das System notwendig. Die Analyse muss umfassend und detailliert genug sein, sodass es keine Missverständnisse oder Interpretationsmöglichkeiten während der Entwicklung gibt.

Eine gute Analyse ist der Grundstein für ein realistisches Zeitmanagement und in Projekten für eine realistische Kosteneinschätzung. [7]

Die folgenden Punkte listen einige Vorteile warum wir überhaupt diese Analyse grundsätzlich durchgeführt haben:

* Kosten einschätzen / senken
* Risiken verkleinern
* Erleichterung beim Erstellen von Testfällen und Testszenarien
* Gute Produkte erleichtern den Verkauf

Aus diesem Grund werden in diesem Abschnitt alle Anforderungen an den „TopologyEditor“ und das gesamte System detailliert beschrieben. Hierbei wird zwischen funktionalen und nicht-funktionalen Anforderungen unterschieden. Des Weiteren werden die Aufgaben beispielhaft an einigen Use Cases in einem Diagramm erläutert.

## Funktionale Anforderungen

Die funktionale Anforderungen (FR[[1]](#footnote-1)) spezifizieren die Funktionalität oder das Verhalten des Softwareproduktes unter festgelegten Bedingungen. In einem Wort kann man sich fragen was soll das Softwareprodukt tun?

Hierbei gibt es die Muss-Kriterien, das heißt die Kriterien die auf jeden Fall umgesetzt werden müssen. Sie beschreiben vor allem den Einsatz essentielle Funktionen des „TopologyEditor“. Weiterhin formuliert man die Kann-Kriterien, die umgesetzt werden, wenn es die Ressourcen und die Planung des Projektes zulassen und die nicht Einsatzfähigkeiten des Systems beeinträchtigen.

### Muss*-*Kriterien

*FR010*: Die Topologie ist in einer Ansicht darzustellen.

*FR020*: Die Darstellung der Topologie kann separat geschlossen werden.

FR030: Die Topologiedarstellung eines Projekts kann durch den Anwender separat geschlossen werden.

FR040: Das Editieren von Informationen einer Topologie durch den Nutzer muss möglich sein.

FR050: Dem Nutzer muss die Möglichkeiten geboten werden die Topologie wiederzuverwenden.

FR060: Nach dem Öffnen sind die wichtigsten Informationen in einer Ansicht bereitzustellen.

FR070: Die topologischen Informationen müssen grafische dargestellt werden.

FR080: Den Nutzer beim Editieren einer Verbindung optisch unterstützen.

FR090 Die Basisfunktionen wie Laden, Speichern, Importieren und Exportieren müssen bereitgestellt werden.

FR100 Darstellung verschiedenen Topologien.

FR110 Die Topologie-Ansicht zoomen.

### Kann-Kriterien

FR200: Bereitstellen einer „Über Uns“-Ansicht.

FR210: Funktion zum Umstellen der Sprache.

FR220: Funktion zur Umschaltung der Topologie-Art (siehe Abschnitt 2.2)

FR230 Funktion zum Suchen und Ersetzen von Informationen.

FR240 Funktion zur Filterung von Informationen und der Darstellung.

FR250 Bereitstellen der Topologie unabhängig vom Protokoll

FR260 Darstellen von Fehlern in einer eigenen Ausgabe (A8)

FR270 Bereitstellen einer Auto-Sortierungs-Funktion für die Darstellung.

## Nichtfunktionale Anforderungen

Nichtfunktionale Anforderungen (NFRs[[2]](#footnote-2)), auch technische Anforderungen genannt, beschreiben Aspekte, die typischerweise mehrere oder alle funktionalen Anforderungen betreffen bzw. überschneiden (cross-cut). Sie haben in der Regel einen Einfluss auf die gesamte Softwarearchitektur. Außerdem beeinflussen sich nichtfunktionale Anforderungen gegenseitig. Auch hier werden die Anforderungen in Muss-und in Kann-Kriterien aufgeteilt.

### Muss-Kriterien

NF010: Es muss möglich sein die Software zu erweitern oder Fehlerkorrekturen durchzuführen. (Wartbarkeit).

NF020: Die Software muss in der Lage sein, Mängel oder Ursachen von Fehlverhalten zu diagnostizieren oder änderungsbedürftige Teile zu identifizieren. (Analysierbarkeit).

NF030: Die Software muss die Änderung der Implementierungen einer Spezifikation zulassen. (Änderbarkeit)

NF040 Die Software soll Fehlertolerant sein. (Stabilität)

NF050: Die Software soll validierbar sein. (Testbarkeit)

NF060: Die Software ist ein Bestandteil des „Communication Studio“.

NF070: Der Datenaustausch erfolgt im XML-Format.

NF080: Die Software muss unter folgenden Webbrowsern Google Chrome, Mozilla Firefox und Internet Explorer benutzbar sein.

### Kann-Kriterien

NF110: Die Software soll zukünftig in einer Cloud zur Verfügung gestellt werden.

NF120: Die Software soll mit Angular 5 umgesetzt werden.

1. FR - Functional requirement [↑](#footnote-ref-1)
2. *NFR* - Nonfunctional requirement [↑](#footnote-ref-2)