**\*\*Architekturdokumentation\*\***

**# 1 Einführung und Ziele**

*Beschreibt die wesentliche Anforderungen und treibenden Kräfte, die*

*Softwarearchitekten und Entwicklungsteams berücksichtigen müssen. Dazu*

*gehören die*

*- zugrunde liegenden Geschäftsziele, wesentliche Aufgabenstellung und*

*essenzielle fachliche Anforderungen an das System*

* Möglichkeit für Studenten online zu lernen (ortsunabhängig, jederzeit, plattformunabhängig)
* Die Webanwendung soll zur Prüfungsvorbereitung dienen und alle fach-/modulspezifischen Kontrollfragen enthalten.
* Weiterhin ist geplant, dass die Studenten ihren Lernfortschritt speichern können

**Qualitätsziele für die Architektur**

* Übersichtlichkeit (Benutzerfreundlich)
* Schnell und einfach zugänglich
* Stabilität

**relevante Stakeholder und deren Erwartungshaltung**

* Entwickler (einfache Bedienung, einfache Struktur, Stabilität)
* User (einfache Bedienung, ansprechende Benutzeroberfläche, Speicherung der Erfolge/Fortschritte)

**Aufgabenstellung**

*\*\*Inhalt.\*\**

*\*\*Kurzbeschreibung der fachlichen Aufgabenstellung, treibenden Kräfte,*

*Extrakt (oder Abstract) der Anforderungen. Verweis auf (hoffentlich*

*vorliegende) Anforderungsdokumente (mit Versionsbezeichnungen und*

*Ablageorten).\*\**

*\*\*Motivation.\*\**

*\*\*Aus Sicht der späteren Nutzer ist die Unterstützung einer fachlichen*

*\*\*Form.\*\**

*\*\*Kurze textuelle Beschreibung, eventuell in tabellarischer Use-Case Form.*

*Sofern vorhanden sollte die Aufgabenstellung Verweise auf die*

*entsprechenden Anforderungsdokumente enthalten.\*\**

*\*\*Halten Sie diese Auszüge so knapp wie möglich und wägen Sie Lesbarkeit*

*und Redundanzfreiheit gegeneinander ab.\*\**

*Aufgabe oder Verbesserung der Qualität der eigentliche Beweggrund, ein*

*neues System zu schaffen oder ein bestehendes zu modifizieren*.\*\*

* Im Rahmen der Veranstaltung BWI50202 .NET Vertiefung soll eine

Webanwendung erstellt werden. Die angestrebte technische Lösung für das

Projekt soll folgende Bestandteile enthalten:

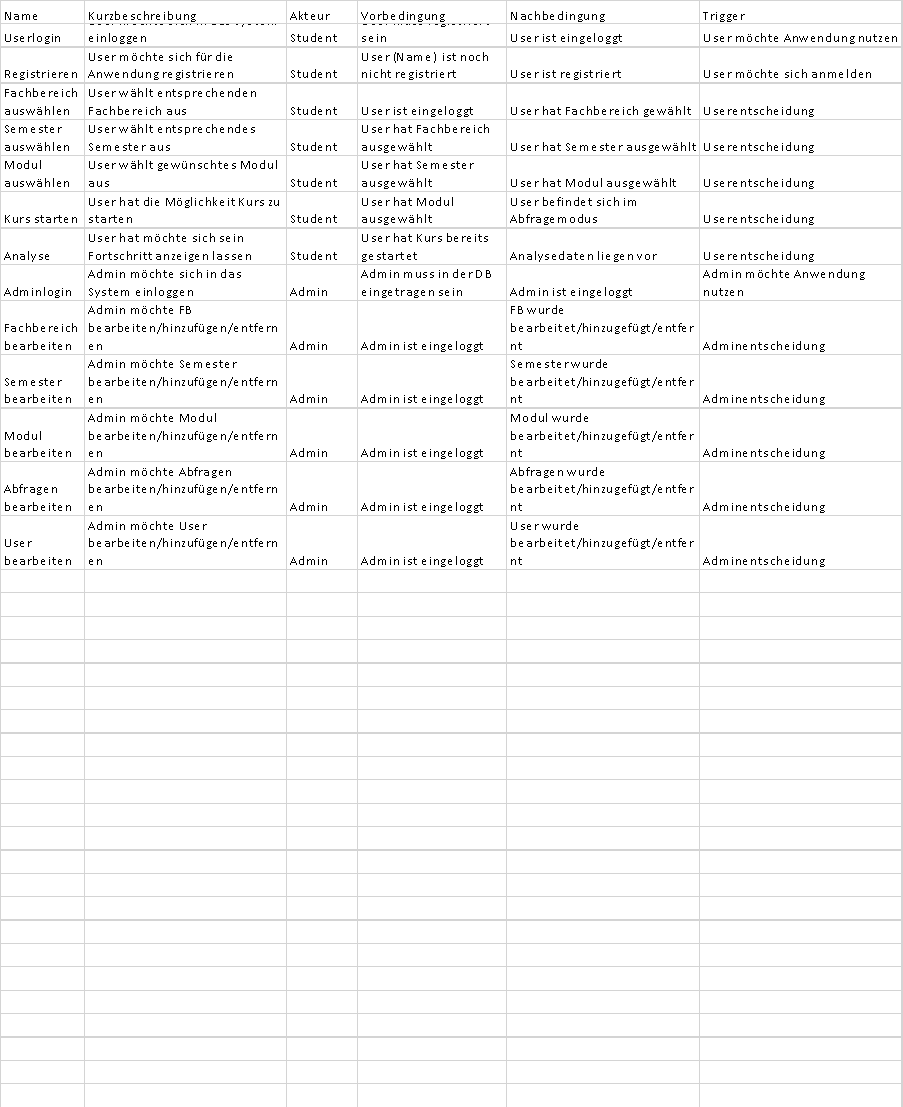
* Frontend (Präsentation + Geschäftslogik): Webanwendung (ASP. NET Core / Java EE)
* Backend (Geschäftslogik + Datenhaltung): Web Services (ASP .NET Core Web-API / Spring Boot & Java)
* Siehe Pflichtenheft (auf Trello)
* Die Aufgabe liegt darin mit nur wenigen Vorgaben eine Webanwendung zu

entwickeln, welche eigenständig konkretisiert und ausgearbeitet werden muss.

In diesem Fall wurde eine eigene Projektidee vorgeschlagen und mit dem

Dozenten abgestimmt.

* Siehe Use-Case-Beschreibung (UseCasesBeschreibung.Excel-Dokument auf Trello)



**## Qualitätsziele**

*\*\*Inhalt.\*\**

*\*\*Die Top-3 bis Top-5 der Qualitätsziele für die Architektur, deren*

*Erfüllung oder Einhaltung den maßgeblichen Stakeholdern besonders*

*wichtig sind. Gemeint sind hier wirklich Qualitätsziele, die nicht*

*unbedingt mit den Zielen des Projekts übereinstimmen. Beachten Sie den*

*Unterschied.\*\**

*\*\*Motivation.\*\**

*\*\*Weil Qualitätsziele grundlegende Architekturentscheidungen oft*

*maßgeblich beeinflussen, sollten Sie die für Ihre Stakeholder relevanten*

*Qualitätsziele kennen, möglichst konkret und operationalisierbar.\*\**

\*\*Form.\*\*

\*\*Tabellarische Darstellung der Qualitätsziele mit möglichst konkreten

Szenarien, geordnet nach Prioritäten.\*\*

* Die Webanwendung muss für die verschiedenen Benutzer (insb. Studierende und Dozenten) übersichtlich sein
* Die Webanwendung muss performant sein (schnell sein)
* Die Webanwendung soll die Kontrollfragen zu verschieden Modulen und Fachbereichen enthalten können
* zu 1. jede Funktion muss schnell, d.h. mit max. 7(bis zu 10) Klicks erreichbar sein
* zu 2. die Webanwendung muss für mehrere Benutzer (für 50 Benutzer) gleichzeitig erreichbar sein, ohne dass es aus Sicht eines einzelnen Benutzers zu Laufzeiteinbußen kommt
* zu 3. Nach einem Jahr müssen >90% der Module in der Webanwendung sein und diese müssen Kontrollfragen erhalten.
* zu 1: Test der Webanwendung
* zu 1: Benutzertest mit Studierenden
* zu 2: Test der Webanwendung mit mehreren Benutzern (bis zu 50)
* zu 3: Statistische Auswertung: Zählen der Kurs in moodle und Vergleich mit dem Veranstaltungsangebot

**## Stakeholder**

*\*\*Inhalt.\*\**

*\*\*Expliziter Überblick über die Stakeholder des Systems, d.h. über alle*

*Personen, Rollen oder Organisationen...\*\**

*\*\*Motivation.\*\**

*\*\*Sie sollten die Projektbeteiligten und -betroffenen kennen, sonst*

*erleben Sie später im Entwicklungsprozess Überraschungen. Diese*

*Stakeholder bestimmen unter anderem Umfang und Detaillierungsgrad der*

*von Ihnen zu leistenden Arbeit und Ergebnisse.\*\**

*\*\*Form.\*\**

*\*\*Tabelle mit Rollen- oder Personennamen, sowie deren Erwartungshaltung*

*bezüglich der Architektur und deren Dokumentation.\*\**

* Entwickler (wir als Studenten)
* User (Benutzer die die Webanwendung möglicherweise später nutzen)
* Dozent (der Veranstaltung)

*| Rolle | Kontakt | Erwartungshaltung |*

| Prof. Dr. Daniel Retkowitz (Dozent) | daniel.retkowitz@hs-niederrhein.de | --- |

| Tobias Jansen (Student) | Tobias.Jansen@stud.hn.de | Erfolgreiche Umsetzung des Projektes |

| Henning Schilder (Student) | Henning.Schilder@stud.hn.de | Erfolgreiche Umsetzung des Projektes |

| Jari Elfers (Student) | jari.elfers@stud.hn.de | Erfolgreiche Umsetzung des Projektes |

| Aleks Adamovic (Student) | Aleks.Adamovic@stud.hn.de | Erfolgreiche Umsetzung des Projektes |

| User (Student) | diverse | Erfolgreiche Nutzung der Anwendung |

**# 2 Randbedingungen**

*\*\*Inhalt.\*\**

*\*\*Fesseln und Vorgaben, die ihre Freiheiten bezüglich Entwurf,*

*Implementierung oder Ihres Entwicklungsprozesses einschränken. Diese*

*Randbedingungen gelten manchmal organisations- oder firmenweit über die*

*Grenzen einzelner Systeme hinweg.\*\**

*\*\*Motivation.\*\**

*\*\*Als Architekt sollten Sie explizit wissen, wo Ihre Freiheitsgrade*

*bezüglich Entwurfsentscheidungen liegen und wo Sie Randbedingungen*

*beachten müssen. Sie können Randbedingungen vielleicht noch verhandeln,\*\**

*zunächst sind sie aber da.*

*\*\*Form.\*\**

*\*\*Einfache Tabellen der Randbedingungen mit Erläuterungen. Bei Bedarf*

*unterscheiden Sie technische, organisatorische und politische*

*Randbedingungen oder übergreifende Konventionen (beispielsweise*

*Programmier- oder Versionierungsrichtlinien, Dokumentation- oder*

*Namenskonvention)\*\**

**Technische Vorgaben**

|  |  |
| --- | --- |
| **Programmiersprache** | C# |
| **Framework** | .NET Core 2.0 |
| **IDE** | Visual Studio 2017 |
| **Zielumgebung** | Plattformunabhängig (Webanwendung) |
| **Grafische Darstellung** | Webbrowser |

**Organisatorische Vorgaben**

|  |  |
| --- | --- |
| **Zeitrahmen** | Mitte Oktober 2017 bis Mitte Januar 2018 (drei Monate) |
| **Team** | Gruppe aus vier Studenten |
| **Dokumentation** | Pflichtenheft und Architekturdokumentation |
| **Präsentation** | 30 Min. bestehend aus PP-Präs. und Poster |
| **Versionsverwaltungssystem** | GitHub |

**# 3 Kontextabgrenzung**

*\*\*Inhalt.\*\**

*\*\*Die Kontextabgrenzung grenzt das System von allen Kommunikationspartnern*

*(Nachbarsystemen und Benutzerrollen) ab. Sie legt damit die externen*

*Schnittstellen fest.\*\**

*\*\*Differenzieren Sie fachlichen Kontext (fachliche Ein- und Ausgaben) und*

*technischen Kontext (Kanäle, Protokolle, Hardware), falls nötig.\*\**

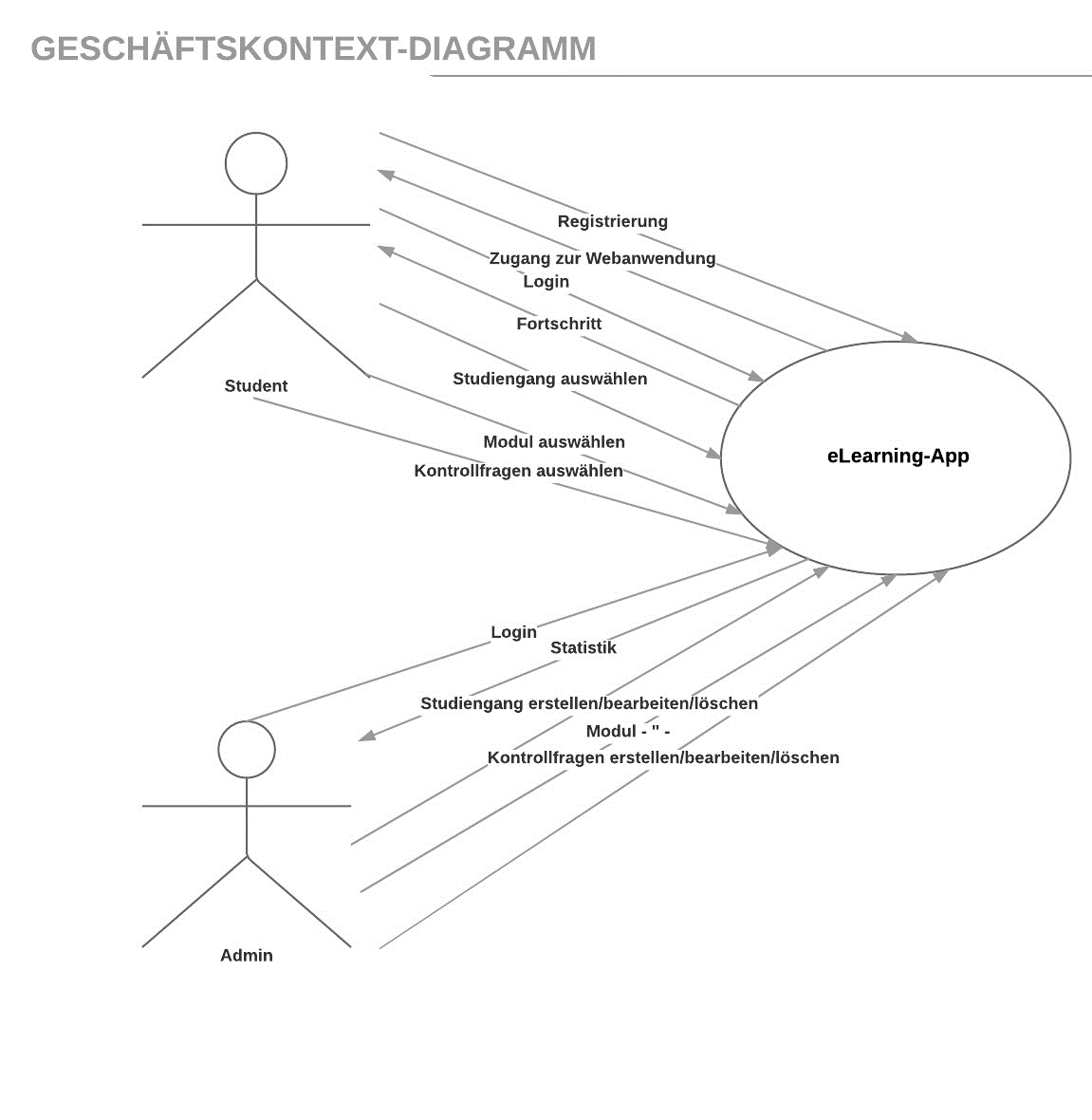
*\*\*Motivation.\*\**

*\*\*Die fachlichen und technischen Schnittstellen zu Kommunikationspartnern*

*gehören zu den kritischsten Aspekten eines Systems. Stellen Sie sicher,*

*dass Sie diese komplett verstanden haben.\*\**

*\*\*Form.\*\**



**## Fachlicher Kontext**

*\*\*Inhalt.\*\**

*\*\*Festlegung \*\*aller\*\* Kommunikationspartner (Nutzer, IT-Systeme, …) mit*

*Erklärung der fachlichen Ein- und Ausgabedaten oder Schnittstellen.*

*Zusätzlich bei Bedarf fachliche Datenformate oder Protokolle der*

*Kommunikation mit den Nachbarsystemen.\*\**

*\*\*Motivation.\*\**

*\*\*Alle Beteiligten müssen verstehen, welche fachlichen Informationen mit*

*der Umwelt ausgetauscht werden.\*\**

*\*\*Form.\*\**

*\*\*Alle Diagrammarten, die das System als Black Box darstellen und die*

*fachlichen Schnittstellen zu den Nachbarn beschreiben.\*\**

*\*\****Alternativ** *können Sie eine Tabelle verwenden. Der Titel*

*gibt den Namen Ihres Systems wieder; die drei Spalten sind:*

*Kommunikationspartner, Eingabe, Ausgabe.\*\**

*\*\*&lt;Diagramm und/oder Tabelle&gt;\*\**

*\*\*&lt;optional: Erläuterung der externen fachlichen Schnittstellen&gt;\*\**

* **Frontend: Webanwendung (Browserfenster)**

|  |  |  |
| --- | --- | --- |
| **Kommunikationspartner** | **Eingabe(-daten)** | **Ausgabe(-daten)** |
| Student | Registrierung(-daten) | Abgeschlossene Registrier. |
| Student | Login(-daten) | Erfolgreicher Login |
| Student | Auswahl Studiengang | Anzeige aller Studiengänge |
| Student | Auswahl Modul | Anzeige aller Module |
| Student | Auswahl Kontrollfragen | Anzeige aller Kontrollfragen |

* **Frontend: Webanwendung (Browserfenster)**

|  |  |  |
| --- | --- | --- |
| **Kommunikationspartner** | **Eingabe(-daten)** | **Ausgabe(-daten)** |
| Admin | Registrierung(-daten) | Abgeschlossene Registrier. |
| Admin | Bearbeiten/Hinzufügen/  Entfernen von Studiengängen | … Studiengang |
| Admin | Bearbeiten/Hinzufügen/  Entfernen von Modulen | … Modul |
| Admin | Bearbeiten/Hinzufügen/  Entfernen von Kontrollfragen | … Kontrollfrage |

* **Backend: Webserver/-service**

|  |  |  |
| --- | --- | --- |
| **Kommunikationspartner** | **Eingabe(-daten)** | **Ausgabe(-daten)** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**## Technischer Kontext**

*\*\*Inhalt.\*\**

*\*\*Technische Schnittstellen (Kanäle, Übertragungsmedien) zwischen dem*

*System und seiner Umwelt. Zusätzlich eine Erklärung (\*mapping\*), welche*

*fachlichen Ein- und Ausgaben über welche technischen Kanäle fließen.\*\**

*\*\*Motivation.\*\**

*\*\*Viele Stakeholder treffen Architekturentscheidungen auf Basis der*

*technischen Schnittstellen des Systems zu seinem Kontext.\*\**

*\*\*Insbesondere Infrastruktur- oder Hardwareentwickler entscheiden auch*

*über diese technischen Schnittstellen.\*\**

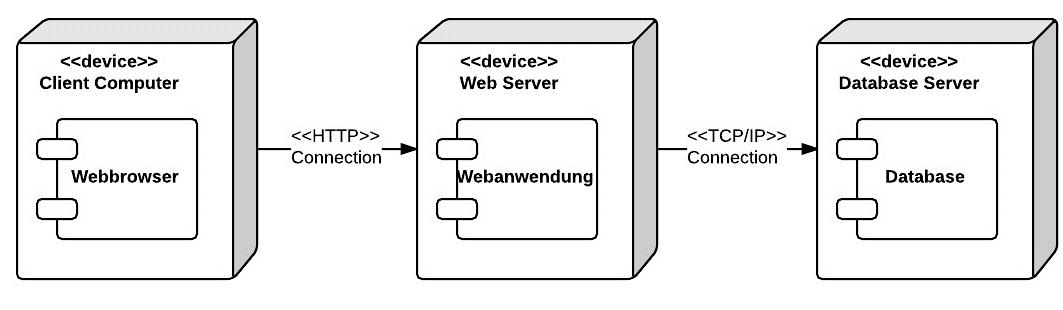
*\*\*Form.\*\**

*\*\*Beispielsweise UML Deployment-Diagramme mit den Kanälen zu*

*Nachbarsystemen, begleitet von einer Tabelle, die Kanäle auf*

*Ein-/Ausgaben abbildet.\*\**

**UML Deployment-Diagramm 09-LernWebApp**



**# 4 Lösungsstrategie**

*\*\*Inhalt.\*\**

*\*\*Kurzer Überblick über die grundlegenden Entscheidungen und*

*Lösungsansätze, die Entwurf und Implementierung des Systems prägen.*

*Hierzu gehören:\*\**

*\*\*- Technologieentscheidungen\*\**

*\*\*- Entscheidungen über die Top-Level-Zerlegung des Systems,*

*beispielsweise die Verwendung gesamthaft prägender Entwurfs- oder*

*Architekturmuster\*\**

*\*\*- Entscheidungen zur Erreichung der wichtigsten Qualitätsanforderungen\*\**

*\*\*- relevante organisatorische Entscheidungen, beispielsweise für*

*bestimmte Entwicklungsprozesse oder Delegation bestimmter Aufgaben*

*an andere Stakeholder.\*\**

*\*\*Motivation.\*\**

*\*\*Diese allerwichtigsten Entscheidungen bilden wesentliche „Eckpfeiler“*

*der Architektur. Von ihnen hängen meistens viele weitere Entscheidungen*

*oder Implementierungsregeln ab.\*\**

*\*\*Form.\*\**

*\*\*Fassen Sie die zentralen Entwurfsentscheidungen \*\*kurz\*\* zusammen.*

*Motivieren Sie ausgehend von Aufgabenstellung, Qualitätszielen und*

*Randbedingungen, was Sie entschieden haben und warum Sie so entschieden*

*haben. Verweisen Sie eher auf weitere Ausführungen in Folgeabschnitten.\*\**

* Wir haben uns entschieden die Webanwendung („09-eLearning-App“) in der Programmiersprache C# zu entwickeln. Weiterhin werden wir VisualStudio als Entwicklungsumgebung nutzen.

VisualStudio haben wir bereits in der Grundlagenveranstaltung kennen gelernt und intensiv genutzt, darüber hinaus ist die Dokumentation zum Thema ASP.NET Core MVC und Entity Framework Core sehr umfangreich. Die Zielumgebung soll plattformunabhängig sein, daher bietet sich das Framework ASP. NET an. Die grafische Darstellung soll über einen Webbrowser realisiert werden. Das Pflichtenheft dient als Grundlage für den angestrebten Funktionsumfang, die vorhandenen Use-Cases unterstützen uns dabei die Funktionen umzusetzen. Um die oben aufgeführten Qualitätsziele zu erreichen werden bei der Entwicklung stets umfangreiche Funktionstests durchgeführt, um mögliche Fehler schnell zu finden und zu eliminieren. Die Bearbeitung des Projekts findet unter ständigem Austausch der Projektmitglieder statt. Zur Unterstützung der Kommunikation greifen wir auf das Versionsverwaltungssystem GitHub zurück. Die Entwicklung des Programmcodes wird hauptsächlich durch die Mitglieder mit dem höheren Verständnis für die Programmierung übernommen. Die anderen Mitglieder kümmern sich verstärkt um die Organisation und Dokumentation des Projektes.

**# 5 Bausteinsicht**

*\*\*Inhalt.\*\**

*\*\*Diese Sicht zeigt die statische Zerlegung des Systems in Bausteine*

*(Module, Komponenten, Subsysteme, Klassen, Interfaces, Pakete,*

*Bibliotheken, Frameworks, Schichten, Partitionen, Tiers, Funktionen,*

*Makros, Operationen, Datenstrukturen…) sowie deren Beziehungen.\*\**

*\*\*Diese Sicht sollte in jeder Architekturdokumentation vorhanden sein. In*

*der Analogie zum Hausbau bildet die Bausteinsicht den \*Grundrissplan\*.\*\**

*\*\*Motivation.\*\**

*\*\*Behalten Sie den Überblick über den Quellcode, indem Sie die statische*

*Struktur des Systems durch Abstraktion verständlich machen.\*\**

*\*\*Damit ermöglichen Sie Kommunikation auf abstrakterer Ebene, ohne zu*

*viele Implementierungsdetails offenlegen zu müssen.\*\**

*\*\*Form.\*\**

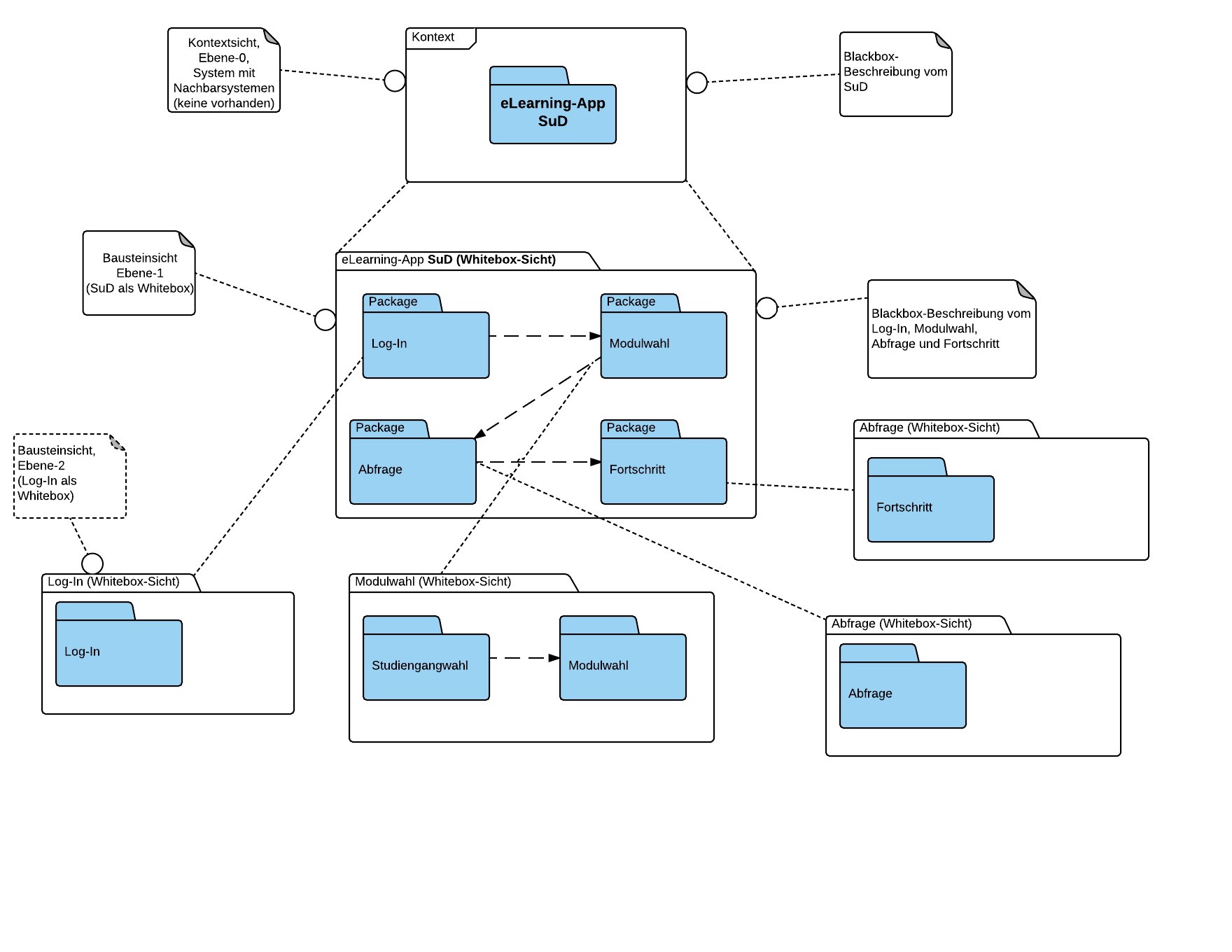
*\*\*Die Bausteinsicht ist eine hierarchische Sammlung von Blackboxen und*

*Whiteboxen (siehe Abbildung unten) und deren Beschreibungen.\*\**

**## Whitebox Gesamtsystem**

*An dieser Stelle beschreiben Sie die Zerlegung des Gesamtsystems anhand*

*des nachfolgenden Whitebox-Templates. Dieses enthält:*

**

**Ebene-1**

**eLearning-App Whitebox-Sicht**

• Zweck / Verantwortlichkeit: Gesamte eLearning Webanwendungen

• Schnittstelle(n): keine Nachbarsysteme (integrierte Datenbank)

• Erfüllte Anforderungen:

• Ablageort / Datei:

• Sonstige Verwaltungsinformation: Autor, Version, Datum, Änderungshistorie

**Ebene-2**

**Log-In Whitebox-Sicht**

• Zweck / Verantwortlichkeit: Ermöglicht den Zugang zur Anwendung

• Schnittstelle(n): Modulwahl

• Erfüllte Anforderungen:

• Ablageort / Datei:

• Sonstige Verwaltungsinformation: Autor, Version, Datum, Änderungshistorie

**Ebene-2**

**Modulwahl Whitebox-Sicht**

• Zweck / Verantwortlichkeit: Dient der Auswahl der Studiengänge/Module

• Schnittstelle(n): Log-In / Abfragen (Kontrollfragen)

• Erfüllte Anforderungen:

• Ablageort / Datei:

• Sonstige Verwaltungsinformation: Autor, Version, Datum, Änderungshistorie

**Ebene-2**

**Abfrage Whitebox-Sicht**

• Zweck / Verantwortlichkeit: Dient der Durchführung der Abfragen (Kontrollfragen)

• Schnittstelle(n): Modulwahl / Fortschritt

• Erfüllte Anforderungen:

• Ablageort / Datei:

• Sonstige Verwaltungsinformation: Autor, Version, Datum, Änderungshistorie

**Ebene-2**

**Fortschritt Whitebox-Sicht**

• Zweck / Verantwortlichkeit: Ermöglicht den Benutzern die Überwachung Ihres Fortschritts

• Schnittstelle(n): Abfragen (Kontrollfragen)

• Erfüllte Anforderungen:

• Ablageort / Datei:

• Sonstige Verwaltungsinformation: Autor, Version, Datum, Änderungshistorie

*- Blackbox-Beschreibungen der hier enthaltenen Bausteine. Dafür haben*

*Sie verschiedene Optionen*:

*- in \*einer\* Tabelle, gibt einen kurzen und pragmatischen*

*Überblick über die enthaltenen Bausteine sowie deren*

*Schnittstellen.*

Falls Sie die **tabellarische Beschreibung** wählen, so werden Blackboxen

darin nur mit Name und Verantwortung nach folgendem Muster beschrieben:

| Name | Verantwortung |

| --- | --- |

**Hier folgen jetzt Erläuterungen zu Blackboxen der Ebene 1.**

| Log-In | Ermöglicht den Login in die Anwendung mit Benutzername und Passwort |

| Modulwahl | Ermöglicht dem Benutzer die Auswahl von Studiengang/Modul, zuvor muss der Benutzer aber den Log-In durchführen |

| Abfrage | Ermöglicht dem Benutzer die Durchführung der Abfragen (Kontrollfragen), zuvor muss der Benutzer ein Studiengang/Modul ausgewählt haben |

| Fortschritt | Ermöglicht dem Benutzer eine Übersicht über seine bisherigen Fortschritte/Erfolge, Fortschritte/Erfolge werden nur erzeugt wenn der Benutzer Abfragen (Kontrollfragen) durchgeführt hat |

**Hier folgen jetzt Erläuterungen zu Blackboxen der Ebene 2.**

| Studiengangwahl | Ermöglicht dem Benutzer nachdem Log-In den geeigneten Studiengang auszuwählen, darauf folgt die Modulauswahl |

| Modulauswahl | Nach erfolgreicher Studiengangwahl muss der Benutzer ein passendes Modul auswählen, darauf folgt die Möglichkeit der Abfragen (Kontrollfragen) |

*Welche Bausteine Ihres Systems Sie hier beschreiben, müssen Sie selbst*

*entscheiden. Bitte stellen Sie dabei Relevanz vor Vollständigkeit.*

*Skizzieren Sie wichtige, überraschende, riskante, komplexe oder*

*besonders volatile Bausteine. Normale, einfache oder standardisierte*

*Teile sollten Sie weglassen.*

**# 6 Laufzeitsicht**

*\*\*Inhalt.\*\**

*Diese Sicht erklärt konkrete Abläufe und Beziehungen zwischen Bausteinen*

*in Form von Szenarien aus folgenden Bereichen:*

*- Wichtige Abläufe oder \*Features\*: Wie führen die Bausteine der*

*Architektur die wichtigsten Abläufe durch?*

*- Interaktionen an kritischen externen Schnittstellen: Wie arbeiten*

*Bausteine mit Nutzern und Nachbarsystemen zusammen?*

*- Betrieb und Administration: Inbetriebnahme, Start, Stop.*

*- Fehler- und Ausnahmeszenarien*

*Anmerkung: Kriterium für die Auswahl der möglichen Szenarien (d.h.*

*Abläufe) des Systems ist deren Architekturrelevanz. Es geht nicht darum,*

*möglichst viele Abläufe darzustellen, sondern eine angemessene Auswahl*

*zu dokumentieren.*

*\*\*Motivation.\*\**

*Sie sollten verstehen wie (Instanzen von) Bausteine(n) Ihres Systems*

*ihre jeweiligen Aufgaben erfüllen und zur Laufzeit miteinander*

*kommunizieren.*

*Nutzen Sie solche Szenarien in der Dokumentation hauptsächlich zur*

*besseren Kommunikation mit Stakeholdern, die statische Modelle (z.B.*

*Bausteinsicht, Verteilungssicht) weniger verständlich finden.*

*\*\*Form.\*\**

*Für die Beschreibung von Szenarien gibt es zahlreiche*

*Ausdrucksmöglichkeiten. Nutzen Sie beispielsweise:*

*- Nummerierte Schrittfolgen oder Aufzählungen in Umgangssprache*

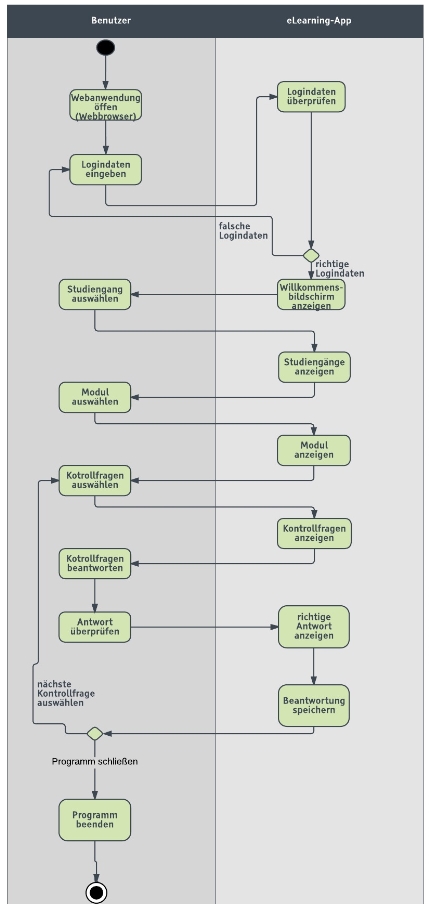
*- Aktivitäts- oder Flussdiagramme*

*- Sequenzdiagramme*

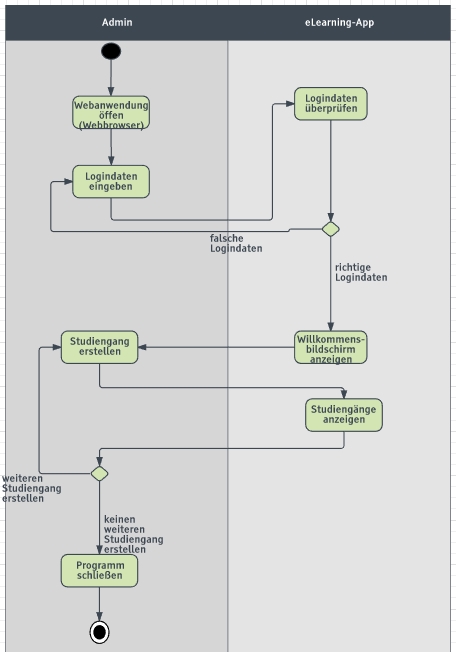
*- BPMN oder EPKs (Ereignis-Prozessketten)*

*- Zustandsautomaten*

*- …*



Aktivitätsdiagramm aus Benutzersicht (Normalablauf/-szenario)



Aktivitätsdiagramm aus Adminsicht, Erstellen neuer Studiengänge

**# 7 Verteilungssicht**

*\*\*Inhalt.\*\**

*Die Verteilungssicht beschreibt:*

*1. die technische Infrastruktur, auf der Ihr System ausgeführt wird,*

*mit Infrastrukturelementen wie Standorte, Umgebungen, Rechnern,*

*Prozessoren, Kanälen und Netztoplogien sowie sonstigen Bestandteilen*

*und*

*2. die Abbildung von (Software-)Bausteinen auf diese Infrastruktur.*

*Häufig laufen Systeme in unterschiedlichen Umgebungen ab, beispielsweise*

*Entwicklung-/Test- oder Produktionsumgebungen. In solchen Fällen sollten*

*Sie alle relevanten Umgebungen aufzeigen.*

*Nutzen Sie die Verteilungssicht insbesondere, wenn Ihre Software auf*

*mehr als einem Rechner, Prozessor, Server oder Container abläuft oder*

*Sie Ihre Hardware sogar selbst konstruieren.*

*Aus Softwaresicht genügt es auf die Aspekte zu achten, die für die*

*Softwareverteilung relevant sind. Hardwarearchitekten können bei Bedarf*

*die Infrastruktur mit beliebigen Details beschreiben.*

*\*\*Motivation.\*\**

*Software läuft nicht ohne Infrastruktur. Diese zugrundeliegende*

*Infrastruktur beeinflusst Ihr System und/oder querschnittliche*

*Lösungskonzepte, daher müssen Sie diese Infrastruktur kennen.*

*Das oberste Verteilungsdiagramm könnte bereits in Ihrem technischen*

*Kontext enthalten sein, mit Ihrer Infrastruktur als EINE Black-Box.*

*Jetzt zoomen Sie in diese Infrastruktur mit weiteren*

*Verteilungsdiagrammen hinein:*

*- Die UML stellt mit Verteilungsdiagrammen (Deployment diagrams) eine*

*Diagrammart zur Verfügung, um diese Sicht auszudrücken. Nutzen Sie*

*diese, evtl. auch geschachtelt, wenn Ihre Verteilungsstruktur es*

*verlangt.*

*- Falls Ihre Infrastruktur-Stakeholder andere Diagrammarten*

*bevorzugen, die Prozessoren und Kanäle zeigen, sind die hier*

*ebenfalls einsetzbar.*

**## Infrastruktur Ebene 1**

An dieser Stelle beschreiben Sie (als Kombination von Diagrammen mit

Tabellen oder Texten):

- Die Verteilung des Gesamtsystems auf mehrere Standorte, Umgebungen,

Rechner, Prozessoren oä. owie die physischen Verbindungskanäle

zwischen diesen,

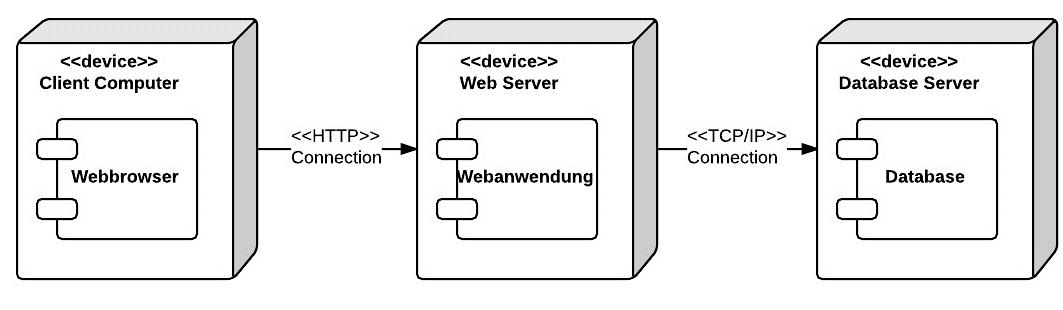
- Wichtige Begründungen für dieser Verteilungsstruktur,

- Qualitäts- und/oder Leistungsmerkmale dieser Infrastruktur,

- Zuordnung von Softwareartefakten zu Bestandteilen der Infrastruktur

Für mehrere Umgebungen oder alternatives Deployment kopieren Sie diesen

Teil von arc42 für alle wichtigen Umgebungen.



**Begründung**

Die Geschäftslogik der Webanwendung befindet sich auf dem Webserver. Die Präsentation der Webanwendung läuft über den Webbrowser des Client/Benutzer-Computer. Die Verbindung läuft über eine HTTP-Verbindung. Die Daten der Webanwendung befinden sich auf dem Datenbankserver. Der Webserver und Datenbankserver sind über eine TCP/IP verbindung miteinander verbunden.

**Qualitäts- und/oder Leistungsmerkmale**

Der Client benötigt eine (schnelle) Internetverbindung um den Webserver erreichen zu können. Weiterhin benötigt der Client einen aktuellen Webbrowser, das Betriebssystem ist “zweitrangig”, da die Webanwendung plattformunabhängig funktioniert.

**# 8 Entwurfsentscheidungen**

*\*\*Inhalt.\*\**

*Wichtige, teure, große oder riskante Architektur- oder*

*Entwurfsentscheidungen inklusive der jeweiligen Begründungen. Mit*

*"Entscheidungen" meinen wir hier die Auswahl einer von mehreren*

*Alternativen unter vorgegebenen Kriterien.*

*Wägen Sie ab, inwiefern Sie Entscheidungen hier zentral beschreiben,*

*oder wo eine lokale Beschreibung (z.B. in der Whitebox-Sicht von*

*Bausteinen) sinnvoller ist. Vermeiden Sie Redundanz. Verweisen Sie evtl.*

*auf Abschnitt 4, wo schon grundlegende strategische Entscheidungen*

*beschrieben wurden.*

*\*\*Motivation.\*\**

*Stakeholder des Systems sollten wichtige Entscheidungen verstehen und*

*nachvollziehen können.*

*\*\*Form.\*\**

*Verschiedene Möglichkeiten:*

**Da dieses Projekt doch eher einen relativ geringen Umfang besitzt, gibt es in dem Sinne keine teuren, großen oder riskanten *Architektur- bzw. Entwurfsentscheidungen*. Wichtige Entscheidungen wurden mit den uns zur Verfügung stehenden Mitteln bzw. den im Verlauf unseres Studiums angeeigneten Techniken abgebildet. Im Abschnitt 4 befindet sich eine deutliche Beschreibung zu den grundlegenden strategischen Entscheidungen.**

**# 9 Qualitätsanforderungen**

*\*\*Inhalt.\*\**

*Dieser Abschnitt enthält möglichst alle Qualitätsanforderungen als*

*Qualitätsbaum mit Szenarien. Die wichtigsten davon haben Sie bereits in*

*Abschnitt 1.2 (Qualitätsziele) hervorgehoben.*

*Nehmen Sie hier auch Qualitätsanforderungen geringerer Priorität auf,*

*deren Nichteinhaltung oder -erreichung geringe Risiken birgt.*

*\*\*Motivation.\*\**

*Weil Qualitätsanforderungen die Architekturentscheidungen oft maßgeblich*

*beeinflussen, sollten Sie die für Ihre Stakeholder relevanten*

*Qualitätsanforderungen kennen, möglichst konkret und operationalisiert.*

**## Qualitätsbaum**

*\*\*Inhalt.\*\**

*Der Qualitätsbaum ( a la ATAM) mit Qualitätsszenarien an den Blättern.*

*\*\*Motivation.\*\**

*Die mit Prioritäten versehene Baumstruktur gibt Überblick über die*

*oftmals zahlreichen Qualitätsanforderungen.*

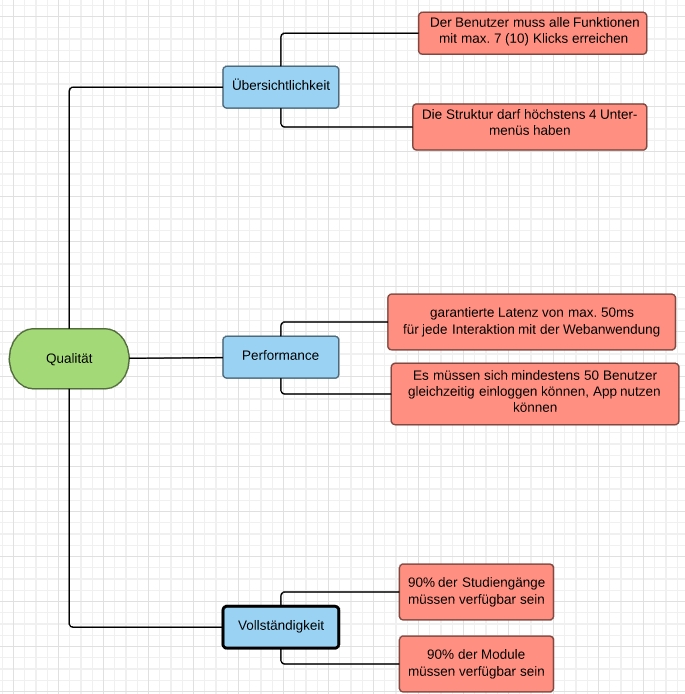
- Baumartige Verfeinerung des Begriffes „Qualität“, mit "Qualität"

oder Nützlichkeit als Wurzel.

- Mindmap mit Q-Oberbegriffen als Hauptzweige

In jedem Fall sollten Sie hier Verweise auf die Szenarien des folgenden

Abschnittes aufnehmen.



**## Qualitätsszenarien**

*\*\*Inhalt.\*\**

*Konkretisierung der (in der Praxis oftmals vagen oder impliziten)*

*Qualitätsanforderungen durch (Qualitäts-)Szenarien.*

*Diese Szenarien beschreiben, was beim Eintreffen eines Stimulus auf ein*

*System in bestimmten Situationen geschieht.*

*Wesentlich für die meisten Softwarearchitekten sind zwei Arten von*

*Szenarien:*

*- Nutzungsszenarien (auch genannt Anwendungs- oder*

*Anwendungsfallszenarien) beschreiben, wie das System zur Laufzeit*

*auf einen bestimmten Auslöser reagieren soll. Hierunter fallen auch*

*Szenarien zur Beschreibung von Effizienz oder Performance. Beispiel:*

*Das System beantwortet eine Benutzeranfrage innerhalb einer Sekunde.*

*- Änderungsszenarien beschreiben eine Modifikation des Systems oder*

*seiner unmittelbarer Umgebung. Beispiel: Eine zusätzliche*

*Funktionalität wird implementiert oder die Anforderung an ein*

*Qualitätsmerkmal ändert sich.*

*\*\*Motivation.\*\**

*Szenarien operationalisieren Qualitätsanforderungen und machen deren*

*Erfüllung mess- oder entscheidbar.*

*Insbesondere wenn Sie die Qualität Ihrer Architektur mit Methoden wie*

*ATAM überprüfen wollen, bedürfen die in Abschnitt 1.2 genannten*

*Qualitätsziele einer weiteren Präzisierung bis auf die Ebene von*

*diskutierbaren und nachprüfbaren Szenarien*.

*\*\*Form.\*\**

*Entweder tabellarisch oder als Freitext.*

**# 10 Glossar**

*\*\*Inhalt.\*\**

*Die wesentlichen fachlichen und technischen Begriffe, die Stakeholder im*

*Zusammenhang mit dem System verwenden.*

*Nutzen Sie das Glossar ebenfalls als Übersetzungsreferenz, falls Sie in*

*mehrsprachigen Teams arbeiten.*

*\*\*Motivation.\*\**

*Sie sollten relevante Begriffe klar definieren, so dass alle Beteiligten*

*1. diese Begriffe identisch verstehen, und*

*2. vermeiden, mehrere Begriffe für die gleiche Sache zu haben.*

*- Zweispaltige Tabelle mit &lt;Begriff&gt; und &lt;Definition&gt;*

*- Eventuell weitere Spalten mit Übersetzungen, falls notwendig.*

*| Begriff | Definition |*

*| --- | --- |*

*| \*&lt;Begriff-1&gt;\* | \*&lt;Definition-1&gt;\* |*

*| \*&lt;Begriff-2&gt;\* | \*&lt;Definition-2&gt;\* |*