



Bachelorarbeit
im Studiengang Medieninformatik

**Möglichkeiten und Grenzen von
Webtechnologien bei der Entwicklung von
Kiosksoftware anhand der
Implementierung einer Sharing-Station für
Museen und Ausstellungen**

vorgelegt von
Marcus Schreiter
an der Hochschule der Medien Stuttgart
am 06.04.2020

zur Erlangung des akademischen Grades
Bachelor of Science (B.Sc.)

Erstprüfer: Prof. Uwe Schulz
Zweitprüfer: Joakim Repomaa

Hiermit versichere ich, Marcus Schreiter, ehrenwörtlich, dass ich die vorliegende Bachelorarbeit mit dem Titel: "Möglichkeiten und Grenzen von Webtechnologien bei der Entwicklung von Kiosksoftware anhand der Implementierung einer Sharing-Station für Museen und Ausstellungen" selbstständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen wurden, sind in jedem Fall unter Angabe der Quelle kenntlich gemacht. Die Arbeit ist noch nicht veröffentlicht oder in anderer Form als Prüfungsleistung vorgelegt worden.

Ich habe die Bedeutung der ehrenwörtlichen Versicherung und die prüfungsrechtlichen Folgen (§26 Abs. 2 Bachelor-SPO (6 Semester), § 24 Abs. 2 Bachelor-SPO (7 Semester), § 23 Abs. 2 Master-SPO (3 Semester) bzw. § 19 Abs. 2 Master-SPO (4 Semester und berufsbegleitend) der HdM) einer unrichtigen oder unvollständigen ehrenwörtlichen Versicherung zur Kenntnis genommen.

Stuttgart, 06.04.2020

Marcus Schreiter

Zusammenfassung

Inhaltsverzeichnis

| | |
|---|------------|
| Zusammenfassung | iii |
| 1 Einleitung | 1 |
| 1.1 MESO | 1 |
| 1.2 Dialogmuseum | 2 |
| 1.3 Sharing-Station | 2 |
| 1.3.1 Sharing-Station als Produkt | 3 |
| 2 Grundlagen | 5 |
| 2.1 Kiosksysteme | 5 |
| 2.2 Webtechnologien | 7 |
| 3 Anforderungsanalyse | 8 |
| 3.1 Funktionale Anforderungen | 9 |
| 3.2 Nicht-funktionale Anforderungen | 11 |
| 4 Systemarchitektur | 13 |
| 5 Softwarestack | 14 |
| 6 Hardwareschnittstellen | 15 |
| 7 Fazit | 16 |
| Abbildungsverzeichnis | 18 |
| Tabellenverzeichnis | 19 |
| Literatur- und Quellenverzeichnis | 20 |

1 Einleitung

1.1 MESO

MESO Digital Interiors GmbH ist eine Digitalagentur mit Sitz in Frankfurt am Main. Gegründet wurde sie 1997 und begann damals als Bürogemeinschaft von Programmierern und Designern. Von Beginn an lag der Fokus auf Grafik, Musik, Elektronik, 3D-Visualisierungen und Softwareentwicklung. Kommerzielle Projekte und eigene Projekte wurden gemeinschaftlich umgesetzt.

Über die Jahre entstanden aus dieser Bürogemeinschaft mehrerer Firmen: Aspekt1, MESO Web Scapes, MESO Digital Interiors, MESO Image Spaces und MESO Digital Services. Bis zuletzt arbeiteten MESO Digital Interiors, MESO Digital Services und MESO Image Spaces als Firmenkollektiv zusammen, ehe sie im Jahr 2018 schließlich durch die drei Geschäftsführer zu der MESO Digital Interiors GmbH zusammengelegt wurden.

Heute hat MESO rund 30 feste und einige freie Mitarbeiter. Diese arbeiten zum Großteil vor Ort, teilweise aber auch remote. Viele Mitarbeiter sind Designer oder Informatiker. Einige haben einen handwerklichen Hintergrund, wie Schreiner oder Goldschmied. Geführt wird das Unternehmen von Sebastian Oschatz, Max Wolf und Mathias Wollin.

So Transdisziplinär wie die Mitarbeiter sind auch die Projekte. MESO konzipiert, gestaltet und entwickelt Ausstellungen, Messeauftritte, Showrooms, Webapplikationen und Apps. Die Projekte bewegen sich dabei meist an der Schnittstelle zwischen Raum, Kommunikation und Technik. Kunden sind dabei oft Firmen aus der Automobilindustrie (BMW, Mercedes-Benz, Yanfeng oder Moovel), aus der Technologiebranche (HERE Technologies, Siemens, Keyence) oder Institutionen öffentlicher Träger wie Museen, Hochschulen oder Städte (Hochschule Mainz, Senckenberg Museum, Stadthalle Karlsruhe).

Bei Medieninstallationen und Echtzeit 3D-Grafik arbeitet MESO weitestgehend mit dem eigenentwickelten Tool VVVV [19] und mit der Spiel-Engine Unreal [17]. Für Backend-Systeme, Apps, Interfaces und Applikationen werden meist Webtechnologien

eingesetzt.

1.2 Dialogmuseum

Das Dialogmuseum Frankfurt [4] gibt es seit 2005 und ist ein privates soziales Unternehmen, das sich zum Ziel gesetzt hat informativ, integrativ und wirtschaftlich mit den Themen Blindheit und Sehbehinderung umzugehen. Dabei ist es ein Ort für sehende, blinde und sehbehinderte Menschen gleichermaßen. Allein im Jahr 2017 zählte das Museum 53.000 Besucher*innen [5].

Das Ausstellungskonzept beruht auf dem 1988 von Prof. Dr. Andreas Heinecke entwickelten *Dialog im Dunkeln*. Dabei werden Besucher*innen von blinden Menschen durch eine völlig im dunkeln gehaltene Ausstellung geführt. Dabei erleben sie Alltagssituationen wie Szenen im Park, der Stadt oder einer Bar, müssen sich dabei jedoch einzig auf den Guide und ihre übrigen Sinne verlassen.

Das Ausstellungskonzept ist mittlerweile ein weltweites Franchisesystem, welches vom Dachverband Dialogue Social Enterprise GmbH (DSE) [6] mit Sitz in Hamburg verwaltet wird. Zu Dialog im Dunkeln sind weitere Konzepte wie *Dialog im Stillen* und Business Workshops hinzugekommen. Diese schaffen weltweit Begegnungen integrativer Art und geben dabei Menschen mit Behinderung einen Arbeitsplatz.

Dialog im Dunkeln zählt seit 1989 weltweit rund 6,5 Millionen Besucher und Besucherinnen [3].

Das Museum in Frankfurt ist seit Ende 2018 geschlossen und soll Ende März 2020 in neuen Räumen, in der B-Ebene der Bahnstation Frankfurter Hauptwache, wiedereröffnet werden. Während das Ausstellungskonzept weitestgehend gleich bleibt, soll der Foyerbereich, sowie der Raum zu dem die Besucher zum Abschluss der Führung gelangen, neugestaltet werden. Dabei bekam MESO den Projektauftrag für zwei digitale Exponate. Zum einen soll ein digitales barrierefreies Gästebuch entstehen, sowie die Sharing-Station, welche Gegenstand dieser Thesis ist. Das Gästebuch soll sich dabei später in dem genannten Reflexionsraum am Ende der Ausstellung und die Sharing-Station im Foyer des neuen Museums befinden.

1.3 Sharing-Station

Das im Rahmen dieser Thesis entwickelte Produkt trägt den Namen Sharing-Station. Der Name ist eine interne Bezeichnung und soll die zwei Haupteigenschaften des Pro-

dukts zusammenbringen. *Sharing*, da das Produkt in der Hauptaufgabe Informationen mit Benutzern teilen soll. Und *Station* da es sich um ein Kiosksystem handelt.

Das Produkt besteht dabei aus der Software und mehreren Hardwarekomponenten. In der Hauptsache wird sich diese Arbeit mit der Softwareentwicklung des Produkts beschäftigen. In ^{To do} (1) wird zwar auf die Hardwareanbindung eingegangen, jedoch liegt auch hier der Fokus auf der Softwarekommunikation.

Zu erwähnen sei auch, dass der Autor sich in der Entwicklung des Produkts hauptsächlich um die Softwareentwicklung gekümmert hat. Produkt-, Screen- und Hardware-redesign sowie das Projektmanagement wurden von anderen Mitarbeitern der Firma MESO übernommen.

Ausgangspunkt für die Idee der Sharing-Station war der Umzug und die damit geplante Neueröffnung des Dialogmuseums. Es entstand der Wunsch im Foyer des Museums den Besucher*innen die Möglichkeit zu geben sich zum einen über das Museum und den Dachverband, sowie über andere ähnliche Projekte zu informieren. Auch sollte es die Möglichkeit geben sich aktiv zu engagieren und zu beteiligen, beispielsweise durch Spenden oder Hinterlassen seiner Kontaktdaten. Und weiter gab es den Wunsch, den Besucher*innen die Möglichkeit zu geben ein öffentliches, soziales Commitment zu hinterlassen.

Schon früh war die Idee da diese Anforderungen nicht beispielsweise durch Drucken und Auslegen von Broschüren und Flyern sowie Aufstellen einer Spendenbox zu erfüllen, sondern das Ganze in einer digitalen Form anzubieten. So entstand die Idee der Sharing-Station: Eine Art Terminal mit großem Touchscreen, an dem die Besucher*innen all die zuvor beschriebenen Dinge tun können und welches durch seine digitale Form in seinem Funktionsumfang beliebig erweiterbar ist.

1.3.1 Sharing-Station als Produkt

Von Anfang an wurde die Software der Sharing-Station unter den Gesichtspunkten der Adaptier- und Wiederverwendbarkeit entwickelt. Schon früh wurde von MESO erkannt, dass die Anforderungen, welche das Dialogmuseum an die Sharing-Station hat, Anforderungen sind, die in der Praxis häufig von Museen und Ausstellungsmachern gestellt werden. Die Forderung nach Alternativen zu klassischen Ausstellungsheften, Flyern und Newsletter-Listen, wird gerade im Zuge von Digitalisierung im Museums- und Ausstellungsbetrieb sehr häufig gestellt. So entstand die Idee, die Sharing-Station für das Dialogmuseum so zu entwickeln, dass sie später als Produkt auch anderen Kunden angeboten werden kann.

Schon während der Entwicklungszeit gab es erstes Interesse innerhalb des Dialogue

1 Einleitung

Social Enterprise GmbH Verbands, die Sharing-Station in anderen Museen einzusetzen.

2 Grundlagen

Zunächst sollen die zwei zentralen Begriffe dieser Thesis, *Kiosksystem* und *Webtechnologien*, erklärt und eingeordnet werden. Da der Begriff Kiosksystem, auch unter Softwareentwicklern, nicht unbedingt geläufig ist oder zumindest nicht ausreichend bekannt, wird er an dieser Stelle in seiner Wortherkunft und -bedeutung ausführlich erläutert. Der Begriff der Webtechnologien wiederum, sollte für die meisten Leser bekannt sein. Trotzdem soll er an dieser Stelle kurz erklärt, sowie die Bedeutung in der Entwicklung über die Zeit und im Zusammenhang mit dieser Arbeit analysiert werden.

2.1 Kiosksysteme

Wenn man in der Informationstechnologie von Kiosksystemen spricht, sind damit meist zugängliche Computersysteme gemeint die im öffentlichen oder halböffentlichen Raum platziert sind. Zudem besitzen sie eine Benutzerschnittstelle, sehr oft in Form eines Touchscreens. Sie bieten dabei in der Regel Zugang zu Informationen oder elektronischen Transaktionen [13].

Der Begriff *Kiosk* hat seinen Ursprung im Persischen und steht dort für ein zeltartiges Gartenhaus oder eine Art Erker an orientalischen Palästen [11]. Allgemeiner steht er in der islamischen Baukunst für einen pavillonähnlichen Bau [20]. In der heutigen, allgemeinsprachlichen Definition versteht man unter einem Kiosk eine Verkaufsstelle für Zeitungen und Zeitschriften [20, 11].

In der Bedeutung hat unsere heutige Definition dabei die äußere Form, das Pavillonähnliche, des ursprünglichen Kiosks übernommen. Oft sind Kioske alleinstehende Häuschen, die in ihrer Form an einen Pavillon erinnern. Hinzugekommen zur Bedeutung ist das Öffentlichzugängliche^{To do (2)}. Während der Kiosk in seiner ursprünglichen Wortbedeutung für etwas steht was meist an einen Palast angegliedert ist – und somit vermutlich nur beschränkt zugänglich ist – verstehen wir heute unter dem Begriff einen Ort, dessen Zugang für jeden Menschen gedacht ist. Darüber hinaus ist er in der Regel an belebten und gut zugänglichen Orten in Städten, wie Straßen, Plätze und an Sehenswürdigkeiten, platziert.

Dieser Teil der Wortbedeutung hat sich auf das informationstechnische Kiosksystem übertragen [9]. Mit dem pavillonähnlichen Bau hat es nichts mehr zu tun, dafür aber

mit der Tatsache das es etwas Zugängliches im öffentlichen Raum darstellt. Holfelder beschreibt weitere Parallelen: die Art der Kunden und die Verweildauer. Zum einen gibt es die Laufkundschaft, die durch optische oder akustische Reize zum Herantreten animiert werden, sowie die Kunden die gezielt und mit einer bestimmten Absicht an den Kiosk oder das Kiosksystem herantreten. Die Verweildauer des Kunden ist in beiden Szenarien kurz, vergleicht man den Besuch in einem Kiosk mit dem in einem Kaufhaus, oder das Benutzen eines Kiosksystems mit dem Benutzen der eigenen elektronischen Geräte [9].

Kiosksysteme sind bekannte Systeme. Fasst man den Begriff weit, so ist beispielsweise auch der Geldautomat ein Kiosksystem. Aber auch Ticketautomaten oder der Self-Ordering Kiosk, wie ihn McDonalds 2011 in Europa eingeführt hat [16], sind bekannte Systeme und zum Teil nicht mehr wegzudenken. Oft trifft man in Bibliotheken oder öffentlichen Gebäuden auf Kiosksysteme, die spezifische Informationen und Transaktionen bereitstellen. Denkbar wäre ein Ausleih-Kiosk in einer Bibliothek oder ein Kiosk in einem Kaufhaus, welcher einen Lageplan und Informationen über die Geschäfte bereithält.

Es zeigt sich: Kiosksysteme können sehr unterschiedliche Zwecke haben. Borchers, Deussen und Knörzer [2] klassifizieren daher Kiosksysteme in vier Kategorien:

1. Informations-Kioske
2. Werbe-Kioske
3. Service-Kioske
4. Entertainment-Kioske

Mischformen sind denkbar und üblich. Tatsächlich kommen Kiosksysteme die nur in eine der genannten Kategorien fallen eher selten vor.

Der Informations-Kiosk hat die Aufgabe kontextbezogene Informationen bereitzustellen. Benutzer sind motiviert und gehen eher zielgerichtet vor. Der Werbe-Kiosk hat die Aufgabe eine Firma oder ein Produkt in der Öffentlichkeit zu bewerben. Benutzer müssen animiert werden das System zu nutzen und werden beispielsweise durch ein ansprechendes Design motiviert. Der Service-Kiosk ist ähnlich dem Informationskiosk. Zusätzlich kann ein Benutzer Transaktionen über eine Input-Schnittstelle tätigen. Beispielsweise ein Ticket kaufen. Der Entertainment-Kiosk hat außer der Unterhaltung des Benutzers keine weitere Aufgabe. Denkbar wäre ein solches System in einem Wartebereich.

2.2 Webtechnologien

Mit der Erfindung des World Wide Web durch Tim Berners-Lee und Robert Cailliau und damit auch der Erfindung von HTML [1] entstand auch der Begriff der Webtechnologien.

Der Begriff umfasst also all die Technologien, die beteiligt sind um eine Webseite von einem Server über ein Netz an einen Client zu übertragen und anzuzeigen.

Im Rahmen dieser Arbeit soll der Begriff sich weiter nur auf die Softwaretechnologien beschränken. Dabei lassen sich die Softwaretechnologien in clientseitige und serverseitige Technologien einteilen.

Während sich serverseitig allerhand Programmiersprachen und Technologien finden, sind seit jeher die Softwaretechnologien auf der Clientseite im Kern auf HTML, CSS und JavaScript beschränkt. Dabei markiert JavaScript als neueste dieser drei Technologien [10] den Beginn der Rich-Client Applikationen und bietet die Möglichkeit im Browser vollständige, interaktive Applikationen statt nur statische Seiten anzuzeigen. Spätestens mit der Entwicklung und Quellcode-Veröffentlichung der JavaScript-Engine V8 [18] beschränkt sich die Nutzung von JavaScript auch nicht mehr nur auf den Browser. V8 ist ein in C++ geschriebener JavaScript-Interpreter,^{To do (3)} welcher Standalone oder eingebunden in einem C++ Programm genutzt werden kann.

Auf der V8-Engine beruht beispielsweise die asynchrone JavaScript Laufzeitumgebung Node.js, mit der in JavaScript geschriebene Programm direkt ausgeführt werden können [12]. Genutzt wird Node.js so meist für Serverapplikationen oder serverseitige Microservices. Auf Node.js und Chromium wiederum basiert das Framework Electron [8]. Mit diesem können Desktop-Anwendungen cross-plattform entwickelt werden.

So zeigt sich, dass der Begriff *Webtechnologien* nicht mehr nur in der ursprünglichen Definition gesehen werden kann, sondern viel weiter fasst. Vielmehr sind Webtechnologien heute ein Toolkit, welches in fast jeder Situation eingesetzt werden kann.

3 Anforderungsanalyse

Nun sind wir am Ausgangspunkt der Arbeit angelangt: Es soll die Software für ein Kiosksystem entwickelt werden. Im Rahmen dieser Arbeit und am Beispiel der Sharing-Station geschieht dies alleinig mit Webtechnologien. Um die Möglichkeiten und Grenzen dieser Technologien strukturiert und genau prüfen zu können, werden im ersten Schritt die Anforderungen an das System und an die Software des Systems analysiert. Dies ist auch nötig, da die Webtechnologien nun in einem neuen Kontext stehen. Anforderungen an diese Technologien in einem klassischen Webentwicklungskontext sind klarer und bekannter. Im Kontext des Kiosksystems bedarf es einer neuen und veränderten Betrachtung. Diese neuen und veränderten Anforderungen sollen im folgenden Kapitel untersucht und strukturiert werden.

Nach einer üblichen Methode der Systemanalyse werden die Anforderungen in funktionale und nicht-funktionale Anforderungen unterteilt. Dabei definieren funktionale Anforderungen die reine Funktionalität und geben Antworten auf die Frage "Was soll das System machen?". Nicht-funktionale Anforderungen bilden die restlichen Anforderungen ab und definieren dabei Anforderungen an die Eigenschaften und Qualitäten der Funktionen [15, S.20 f.]. Sie ergänzen dabei die funktionalen Anforderungen [7]. Nicht-funktionale Anforderungen werden oft noch weiter unterschieden, beispielsweise in Qualitätsanforderungen und Randbedingungen [15, 7]. Da sich der Anforderungsumfang in dieser Arbeit in Maßen hält, wird diese Unterscheidung im Folgenden nicht vorgenommen.

Funktionale Anforderungen sind immer sehr spezifisch, da sie die konkreten Features und Funktionalitäten abbilden. So beschreiben auch hier die funktionalen Anforderungen die konkreten Eigenschaften der Sharing-Station und geben dabei einen guten Überblick über den geplanten Funktionsumfang. Nicht-funktionale Anforderungen hingegen sind generischer und lassen sich leichter auf andere Projekte übertragen. So sind die im Folgenden beschriebenen nicht-funktionalen Anforderungen, Anforderungen, die auch typisch für Kiosksoftware im Allgemeinen sind. Sie sind für die Zielsetzung dieser Arbeit also von größerer Bedeutung.

Beim Definieren der Anforderungen beschränkt sich diese Arbeit auf die Anforderungen an die Software. Dabei stellen manche Anforderungen auch Ansprüche an andere Teile des Systems. Diese Überschneidung lässt sich nicht vermeiden. Gerade die



Abbildung 3.1: Schablone für funktionale Anforderungen nach Rupp und SOPHIS-Ten [14]

funktionalen Anforderungen stellen meist Anforderungen an das Gesamtsystem.

3.1 Funktionale Anforderungen

Alle Anforderungen sind in natürlicher Sprache verfasst und folgen dabei einer bestimmten Syntax. Diese Syntax folgt den Schablonenregeln von Rupp und SOPHIS-Ten [14]. Abbildung 3.1 zeigt die Schablone für die Formulierung einer funktionalen Anforderung. *<System>* entspricht dem Subjekt^{To do (4)} um welches es geht, z.B. die Sharing-Station oder die Clientsoftware. Anschließend wird die Wichtigkeit festgelegt. Das Schlüsselwort *muss* bedeutet, eine Anforderung ist im rechtlichen Sinne verpflichtend. *Sollte* beschreibt eine Anforderung, welche nicht verpflichtend ist aber die Zufriedenheit erhöht. Und *wird* wird verwendet, um Anforderungen zu definieren, welche in der Zukunft erst integriert werden.

Die Art der Funktionalität beschreibt die unterschiedlichen Systemaktivitäten. Wird hier kein Schlüsselwort eingesetzt, so beschreibt die Anforderung eine selbständige Aktivität. Der Satzteil *die Möglichkeit bieten* setzt immer eine Benutzerinteraktion voraus. *Fähig sein* definiert eine Schnittstellenanforderung.

Schließlich folgt das Objekt, für welches die Funktionalität gefordert wird und das so genannte Prozesswort – das Verb des Satzes, welches die Funktionalität identifiziert. Gilt eine Anforderung nur unter einer bestimmten Bedingung, wird diese dem Wichtigkeit-Schlüsselwort vorausgestellt und das System-Schlüsselwort wird vor der Funktionalität platziert. Ein Beispiel hierfür ist FA8.

FA1 Die Sharing-Station muss den Besucher*innen die Möglichkeit bieten, Informationen über das Museum zu erhalten.

Diese Informationen sollen dabei in Artikelform und zu unterschiedlichen Themen abrufbar sein. Ein Menü soll Überblick über die verschiedenen Artikel geben.

FA2 Die Sharing-Station muss den Besucher*innen die Möglichkeit bieten, Informationen über andere Projekte zu erhalten.

Analog zu den Informationen über das Museum, sollen Informationen über andere, ähnliche Projekte abrufbar sein. Diese sollen dabei in einem eigenen Menü organisiert sein.

FA3 Die Sharing-Station muss den Besucher*innen die Möglichkeit bieten, eine Spende zu hinterlassen.

Das Spenden soll dabei kontaktlos per Kreditkarte, EC-Karte und Mobile Payment oder mit Bargeld, in Form eines Münzeinwurfs, möglich sein.

FA4 Die Sharing-Station muss den Besucher*innen die Möglichkeit bieten, ein Foto-Commitment zu hinterlassen.

Diese Funktion soll ähnlich wie eine Fotobox funktionieren. Eine an das System angeschlossene Webcam soll die Möglichkeit bieten ein Foto zu machen und dieses in einer Galerie zu speichern oder sich per Mail selbst zu schicken. Vor Ort sollen außerdem kleine Tafeln ausgelegt werden auf denen Besucher*innen ein soziales Commitment schreiben und dann auf dem Foto präsentieren können.

FA5 Die Sharing-Station sollte den Besucher*innen die Möglichkeit bieten, sich für einen Newsletter einzutragen.

Das Eintragen soll über ein Formular und eine Bildschirmtastatur möglich sein. Der Request soll dabei an das Wordpress-basierte Newslettersystem des Museums gehen.

FA6 Die Sharing-Station sollte den Besucher*innen die Möglichkeit bieten, Informationen mitzunehmen.

Am Ende von Artikeln sollen weiterführende Informationen und Verweise auf Webseiten per QR-Code verlinkt sein. Besucher*innen können diese Informationen so auf dem eigenen Geräten lesen und speichern.

FA7 Die Sharing-Station sollte den Mitarbeiter*innen des Museums die Möglichkeit bieten, die Inhalte über ein CMS zu verwalten.

FA8 Nach einer gewissen Zeit ohne Benutzerinteraktion, sollte die Oberfläche der Clientsoftware in einen Idle-Modus wechseln.

Dieser Modus soll wie ein Bildschirmschoner wirken. Eine Animation soll dabei die Oberfläche überdecken und mit Schlagwörter und bewegten Formen die Besucher*innen zur Interaktion animieren.



Abbildung 3.2: Umgebungs- und Eigenschaftsschablone für nicht-funktionale Anforderungen nach Rupp und SOPHISTen [14]

3.2 Nicht-funktionale Anforderungen

Die Formulierung nicht-funktionaler Anforderungen anhand einer Schablone fällt schwer. Nicht-funktionale Anforderungen sind in der Literatur nicht einheitlich definiert und in ihrer Art unschärfer als funktionale Anforderungen. Trotzdem schlagen Rupp und SOPHISTen [14] drei verschiedene Schablonen für die Formulierung vor: Die Umgebungss-, Eigenschafts- und Prozessschablone. Abbildung 3.2 zeigt die ersten beiden dieser Schablonen. Sie ähneln dabei der Schablone in Abbildung 3.1. Wieder spielt das Wichtigkeit-Schlüsselwort eine große Rolle. Anders als vorher ist das Subjekt nun nicht mehr *<System>* sondern *<Betrachtungsgegenstand>*, welcher im Falle der Umgebungsschablone im Hauptsatz auf einen Teilbestandteil beschränkt werden kann.

Die Umgebungsschablone bildet Anforderungen ab, die von der Umgebung des Betrachtungsgegenstands abhängig sind. **NFA1** und **NFA2** sind nach dieser Schablone gebildet. **NFA2** beispielsweise fordert, dass die Software auch dann funktioniert wenn das Netz ausfällt in dem sich diese befindet. Das Netz ist dabei Teil der Umgebung und nicht Teil des Betrachtungsgegenstands.

Alle weiteren Anforderungen folgen grob der Struktur der Eigenschaftsschablone. Eigentlich fordert die Schablone immer einen definierten Wert für die beschriebene Eigenschaft, allerdings fordern nicht-funktionale Anforderungen auch oft nur die reine Existenz einer Eigenschaft, wie z.B. in **NFA3**. In solchen Fällen schiebt sich die Eigenschaft hinter das Wichtigkeit-Schlüsselwort und der Vergleichsoperator und der Wert werden weggelassen.

Die Prozessschablone stellt Anforderungen an Akteure, z.B. an den Auftragnehmer, und nicht an das System. Da in diesem Falle nur Anforderungen an die Softwaretechnologien betrachtet werden, findet diese Schablone keine Anwendung und ist deshalb auch nicht abgebildet.

- NFA1** Die Clientsoftware muss so gestaltet sein, dass sie zusammen mit Hardwarekomponenten betrieben werden kann.
- NFA2** Die Clientsoftware sollte so gestaltet sein, dass sie auch offline betrieben werden kann.
- NFA3** Die Oberfläche der Clientsoftware muss multilingual sein.
- NFA4** Die Oberfläche der Clientsoftware sollte in sich geschlossen sein.
Verlassen nicht möglich, Kein Laden zwischen den Seiten, Keine Internet-Links
- NFA5** Die Oberfläche der Clientsoftware sollte einfach zu bedienen und Touch-friendly sein.
- NFA6** Die Clientsoftware sollte plattformunabhängig sein.
- NFA7** Das Deployment der Clientsoftware sollte möglichst einfach und von außerhalb möglich sein.
- NFA8** Die Software sollte modular und erweiterbar gehalten sein.
- NFA9** Das CMS sollte über das Internet aufrufbar sein.
Online-Vorschau

4 Systemarchitektur

5 Softwarestack

6 Hardwareschnittstellen

7 Fazit

To do...

- ☐ 1 (p. 3): Kapitel einfügen
- ☐ 2 (p. 5): Schreibweise prüfen
- ☐ 3 (p. 7): check
- ☐ 4 (p. 9): Wirklich subjekt?

Abbildungsverzeichnis

| | | |
|-----|---|----|
| 3.1 | Schablone für funktionale Anforderungen nach Rupp und SOPHISTen [14] | 9 |
| 3.2 | Umgebungs- und Eigenschaftsschablone für nicht-funktionale Anforderungen nach Rupp und SOPHISTen [14] | 11 |

Tabellenverzeichnis

Literatur- und Quellenverzeichnis

- [1] T. Berners-Lee und R. Cailliau. *WorldWideWeb: Proposal for a HyperText Project*. <https://www.w3.org/Proposal.html>. [Accesed 23.01.2020]. Nov. 1990.
- [2] J. Borchers, O. Deussen und C. Knörzer. „Getting It Across: Layout Issues for Kiosk Systems“. In: *ACM SIGCHI Bulletin* ; 27, 4. - S. 68-74 (1995).
- [3] *Dialog im Dunkeln weltweit*. <https://dialogmuseum.de/mehr-dialog/dialog-im-dunkeln-weltweit/>. [Accesed 31.01.2020].
- [4] *Dialogmuseum Frankfurt*. <https://dialogmuseum.de>. [Accesed 31.01.2020].
- [5] *Dialogmuseum Frankfurt – Zahlen und Fakten 2017*. <https://dialogmuseum.de/wp-content/uploads/2018/04/Zahlen-und-Fakten-2017.pdf>. [Accesed 31.01.2020].
- [6] *Dialogue Social Enterprise*. <https://www.dialogue-se.com/>. [Accesed 23.01.2020].
- [7] C. Ebert. *Systematisches Requirements Engineering*. 6. Aufl. dpunkt.verlag, 2019.
- [8] *Electron – Build cross-platform desktop apps with JavaScript, HTML, and CSS*. <https://www.electronjs.org/>. [Accesed 31.01.2020].
- [9] W. Holfelder. *Multimediale Kiosksysteme*. 1. Aufl. Braunschweig Vieweg, 1995.
- [10] *JavaScript Press Release*. <https://web.archive.org/web/20070916144913/http://wp.netscape.com/newsref/pr/newsrelease67.html>. [Accesed 31.01.2020]. Dez. 1995.
- [11] *Meyers Großes Konversationslexikon*. <http://www.woerterbuchnetz.de/Meyers?lemma=kiosk>. [Accesed 31.01.2020].
- [12] *Node.js*. <https://nodejs.org>. [Accesed 31.01.2020].
- [13] J. Rowley und F. Slack. „Kiosks in retailing: the quiet revolution“. In: *International journal of retail and distribution management*, 31 (6), 329-339. (2003).
- [14] C. Rupp und die SOPHISTen. *Requirements-Engineering und -Management*. 6. Aufl. München Hanser, 2014.
- [15] C. Rupp und die SOPHISTen. *Systemanalyse kompakt*. 3. Aufl. Springer Vieweg, 2013.

- [16] A. Sawall. *McDonald's-Kunden sollen per Touchscreen bezahlen*. <https://www.golem.de/1105/83508.html>. [Accesed 31.01.2020]. Mai 2011.
- [17] *Unreal Game Engine*. <https://www.unrealengine.com>. [Accesed 31.01.2020].
- [18] *V8 – A high-performance JavaScript engine*. <https://opensource.google/projects/v8>. [Accesed 31.01.2020].
- [19] *vvvv - a multipurpose toolkit*. <https://vvvv.org/>. [Accesed 31.01.2020].
- [20] A. Zwahr. *Meyers großes Taschenlexikon*. Hrsg. von M. Lexikonredaktion. Bd. 11. Meyers Lexikonredaktion, 2006.