

WEBDOCUMENTARIES

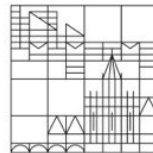
Leitlinien für Design und Produktion

OLIVER BARWIG

Bachelorarbeit

Vorgelegt an der Universität Konstanz

Universität
Konstanz



Arbeitsgruppe Mensch-Computer-Interaktion
Informatik und Informationswissenschaft
Universität Konstanz

29. Juni 2016

Oliver Barwig: *Webdocumentaries*, Leitlinien für Design und Produktion, © 29. Juni 2016

GUTACHTER:
Prof. Dr. Harald Reiterer
Jun.-Prof. Dr. Bela Gipp

ORT:
Konstanz

ZUSAMMENFASSUNG

Das Internet hat sich zur Plattform neuer und interaktiver Präsentationsformen entwickelt. Eine davon ist die Webdocumentary, welche Medien wie Video, Audio, Text und Bilder zu einem nichtlinearen Narrativ verschmilzt und es Nutzern ermöglicht, dieses auf interaktive Weise zu explorieren. Allerdings existieren zur Entwicklung dieses Formats bisher nur wenige Informationen. Nach dem UX-Lifecyclemodell von Hartson und Pyla wurden zwei Webdocumentaries mit *Klynt*, einem führenden Editor in diesem Bereich, umgesetzt. In dieser Arbeit werden technische Erweiterungen für Klynt vorgestellt, welche die Funktionalität einer Webdocumentary erweitern. Auf Basis des UX-Lifecyclemodells und den Erfahrungswerten aus der Umsetzung der beiden Webdocumentaries entwickelt diese Arbeit Leitlinien für die Implementation künftiger Anwendungen. Diese werden in das UX-Lifecyclemodell, welches sich für die Umsetzung benutzerfreundlicher Anwendungen etabliert hat, integriert und vereinfachen und standardisieren so den Entwicklungsprozess der Webdocumentary.

ABSTRACT

The internet has established itself as a platform for presenting modern and interactive media. One such format is the webdocumentary, which blends together video, audio, text, and images in a non-linear narrative that enables the user to experience that medium in an exploratory manner. Unfortunately, there is little information available regarding the development of this medium. Following Hartson and Pyla's UX Lifecycletemplate, two webdocumentaries were developed using *Klynt*, a leading editor in this field. This work presents technical extensions for *Klynt* in order to expand the functionality of webdocumentaries. Based on the UX Lifecycletemplate and the gained hands-on experience, this work derives guidelines for improving the implementation of webdocumentaries. These guidelines are embedded in the UX Lifecycletemplate, an established methodology in creating user-friendly applications, in order to streamline and standardize the process of webdocumentary development in the future.

VORANGEGANGENE ARBEITEN

Einige Inhalte und Bilder dieser Thesis wurden bereits in früheren Arbeiten vorgestellt:

- [1] Oliver Barwig. „Webdocumentary – Mediale Eigenschaften, Technische Hintergründe und Autorenwerkzeuge“. Seminararbeit. 2015.
- [2] Oliver Barwig. „Webdocumentaries mit Klynt – Eine Toolbox mit Anleitungen, Praxisbeispielen und Erweiterungen im Kontext einer multimedialen Ausstellung“. Projektbericht. 2016.

INHALTSVERZEICHNIS

1	EINFÜHRUNG	1
2	DIE WEBDOCUMENTARY	3
2.1	Anwendungsfälle	6
2.1.1	Ausstellungskontext	6
2.1.2	Präsentationskontext	8
3	KONZEPTIONELLE UMSETZUNG	11
3.1	Analysephase	12
3.1.1	System Concept Statement	12
3.1.2	Strukturieren der Inhalte	14
3.2	Designphase	16
3.2.1	Storyboarding	16
3.2.2	Navigation	19
3.2.3	Styleguide	31
3.3	Prototypingphase	37
3.3.1	Werkzeuge	38
3.3.2	Workflow	40
3.4	Evaluationsphase	42
3.5	Zusammenfassung der Leitlinien	44
4	TECHNISCHE ERWEITERUNGSMÖGLICHKEITEN	47
4.1	Die Webdocumentary als App	47
4.1.1	iPad-Anwendung	47
4.1.2	Desktop App	51
4.2	Erweiterungen für Klynt	53
4.2.1	Verwendung eigener Buttons	53
4.2.2	Verwendung eigener CSS-Klassen	57
4.2.3	Verwendung eigener Inhalte durch IFrames	59
4.2.4	Skalierung des Klynt-Players	63
5	REFLEXION UND AUSBLICK	67
	LITERATUR	73
A	APPENDIX – DIGITALE ANLAGEN	75
B	APPENDIX	77

ABBILDUNGSVERZEICHNIS

Abbildung 1	Vereinfachtes UX-Lifecyclemodell nach Hartson und Pyla	2
Abbildung 2	Die Webdocumentary als iPad-Anwendung der Ausstellung im Bildungsturm, gerade in Benutzung	7
Abbildung 3	Die iPad-Anwendung der Ausstellung im Bildungsturm mit Fokus auf den Bildschirm . . .	7
Abbildung 4	Die Webdocumentary zum 50-jährigen Jubiläum der Universität	9
Abbildung 5	Konzeptionelle Umsetzung anhand des UX Lifecyclemodells. Aktuelle Phase: Analyse	11
Abbildung 6	Auszug aus dem Inhaltsverzeichnis der iPad-Anwendung zur Ausstellung	14
Abbildung 7	Konzeptionelle Umsetzung anhand des UX Lifecyclemodells. Aktuelle Phase: Design	16
Abbildung 8	Storyboard der Webdocumentary zum 50-jährigen Jubiläum der Universität	18
Abbildung 9	Menüstruktur, Quelle: http://getbootstrap.com/ – zuletzt aufgerufen am 16.06.2016	20
Abbildung 10	Tabstruktur, Quelle: http://getbootstrap.com/ – zuletzt aufgerufen am 16.06.2016	21
Abbildung 11	Breadcrumbstruktur, Quelle: http://getbootstrap.com/ – zuletzt aufgerufen am 16.06.2016	21
Abbildung 12	Content Carousel, Quelle: http://getbootstrap.com/ – zuletzt aufgerufen am 16.06.2016	21
Abbildung 13	Navigationskonzept der iPad-Anwendung. Seitenleiste, Navigationspfeile und Informationsbutton sind sichtbar	24
Abbildung 14	Fenster mit Informationen zu den Inhalten der aktuellen Sequenz aus der iPad-Anwendung des Bildungsturms	24
Abbildung 15	Geographische Übersichtskarte der Webdocumentary zum 50-jährigen Jubiläum der Universität, das Element „Weg zum Garten“ ist aktiv	27
Abbildung 16	Timeline im Foyer der Webdocumentary zum 50-jährigen Jubiläum der Universität, Tour 5 ist aktiv	28
Abbildung 17	Timeline und Navigationspfeile in einer Untersequenz der Webdocumentary zum 50-jährigen Jubiläum der Universität	29

Abbildung 18	Auszug aus dem Styleguide der Webdocumentary zum 50-jährigen Jubiläum der Universität	34
Abbildung 19	Auszug aus dem Styleguide (Corporate Design) der Ausstellung im Bildungstrum	35
Abbildung 20	Auszug aus dem Styleguide der iPad-Anwendung aus dem Bildungsturm	36
Abbildung 21	Konzeptionelle Umsetzung anhand des UX Lifecyclemodells. Aktuelle Phase: Prototyping .	37
Abbildung 22	Konzeptionelle Umsetzung anhand des UX Lifecyclemodells. Aktuelle Phase: Evaluation . .	42
Abbildung 23	Die Webdocumentary „Kulturgüter in Gefahr!“ als Apple iPad-Anwendung in der Ausstellung des Bildungsturms	49
Abbildung 24	Die Webdocumentary „Kulturgüter in Gefahr!“ als MacOSX Desktopanwendung	51
Abbildung 25	Der Button-Editor in Klynt	54
Abbildung 26	Button (rot) und dessen Hoverstate (weiß) nebeneinander	55
Abbildung 27	Ausgangsbasis für die Erstellung eigener Buttons: Erstellen der Datei in Adobe Illustrator .	56
Abbildung 28	Pulsierender Button in der Webdocumentary zum 50-jährigen Jubiläum der Universität. Rechteck, unten mittig.	58
Abbildung 29	Abbildungen verschiedener Arbeitsschritte für die Entwicklung der Navigationskarte in der Webdocumentary zum 50-jährigen Jubiläum der Universität	62
Abbildung 32	Konzeptionelle Umsetzung anhand des UX Lifecyclemodells zur Extraktion der Leitlinien .	67
Abbildung 33	Xcode – Übersicht Hauptmenü	78
Abbildung 34	Xcode – Übersicht Main Storyboard	79

LISTINGS

Listing 1	CSS-Code für die Einbindung eines eigenen Buttondesigns	56
Listing 2	CSS-Code für einen pulsierenden Button . . .	58
Listing 3	Codeauszug aus der HTML Datei für die Ent- wicklung der Navigationskarte in der Webdo- cumentary zum 50-jährigen Jubiläum der Uni- versität	61
Listing 4	Codeauszug: Javascript für die Entwicklung der Navigationskarte in der Webdocumentary zum 50-jährigen Jubiläum der Universität	61
Listing 5	Javascript Code für die Skalierung des Klynt Plyers	65
Listing 6	Code mit Kommentaren aus der Viewcontrol- ler.swift Datei der iPad-Anwendung	80
Listing 7	Code mit Kommentaren für AppDelegate.swift Datei	81

EINFÜHRUNG

„Wir sind immer in Geschichten. Geschichten in Geschichten in Geschichten. Man weiß nie, wo eine endet und eine andere beginnt! In Wahrheit fließen alle ineinander. Nur in Büchern sind sie säuberlich getrennt.“

— Daniel Kehlmann [10]

Das Teilen von Erlebnissen, Entdeckungen, Vorstellungen und Träumen ist tief in der Geschichte der Menschheit verankert, meist verpackt in eine Erzählung. Während solche Geschichten früher mündlich, später mit den aufkommenden Druckmöglichkeiten auch schriftlich verbreitet wurden, existiert heute eine Vielzahl unterschiedlicher Medien und Präsentationsformen. Die Narration findet sich nunmehr in Computern, Fernsehgeräten, Spielekonsolen und Smartphones. Eine rasante Entwicklung für die Geschichte, die früher noch am Lagerfeuer erzählt wurde. Insbesondere das Internet hat sich, aufgrund der hypertextuellen und nicht-linearen Struktur als Plattform für das Erzählen von Geschichten unter Verwendung moderner Technologien etabliert. Eine davon ist die Webdocumentary.

Diese Arbeit stellt Leitlinien für Design und Produktion einer Webdocumentary vor und soll hierdurch den Entwicklungsprozess zukünftiger Webdocumentaries erleichtern. Auf Basis des UX-Lifecyclemodells nach Hartson und Pyla [8], zu sehen in Abbildung 1, wurden in universitären Projekten zwei Webdocumentaries umgesetzt. Die Erfahrungswerte aus beiden Projekten bilden gemeinsam mit den auf die Webdocumentary angepassten Schritten im UX-Lifecyclemodell die Grundlage für die Leitlinien. Neben der Einführung in die Thematik der Webdocumentary und der abschließenden Reflexion über die vorgestellten Leitlinien und Werkzeuge teilt sich diese Arbeit in zwei große Kapitel: „Konzeptionelle Umsetzung“ und „Technische Erweiterungsmöglichkeiten“. Das Kapitel „Konzeptionelle Umsetzung“ greift die vier Phasen *Anyalyse*, *Design*, *Prototyping* und *Evaluation* des UX-Lifecyclemodells als Abschnitte auf und wendet die enthaltenen Schritte auf die Entwicklung einer Webdocumentary an. Da sich eine Webdocumentary in einigen Fällen jedoch von der Softwareentwicklung unterscheidet, werden diese Schritte auf Basis der Erfahrungswerte aus der Umsetzung von zwei Webdocumentaries – der iPad-Anwendung „Kulturgüter in Gefahr“, welche in einer Ausstellung im Bildungsturm Konstanz gezeigt wurde und der Webdocumentary zum 50-jährigen Jubiläum der Universität – angepasst und

erweitert. Zusätzlich werden sie durch aktuelle Techniken und Methoden verwandter Disziplinen wie der Webentwicklung ergänzt. Aus den vorgestellten Schritten wird anschließend jeweils eine Leitlinie für die Entwicklung einer Webdocumentary extrahiert, die sich in den UX-Lifecycle einbettet. Für eine gute Übersicht steht sie am Anfang eines Abschnitts und leitet so in die darauffolgende Erklärung ein.

Im Kapitel „Technische Erweiterungsmöglichkeiten“ werden Lösungsvorschläge für technikbezogene Probleme bei der Entwicklung und Veröffentlichung einer Webdocumentary präsentiert und Möglichkeiten aufgezeigt, die Webdocumentary auch als Tablet- oder Desktopanwendung verfügbar zu machen. Im folgenden wird der Begriff *Tablet*, durch *iPad-Anwendung* ersetzt, weil eine der Webdocumentaries spezifisch für dieses Gerät entwickelt wurde. Außerdem werden spezifische Erweiterungsmöglichkeiten für die Software „Klynt“ vorgestellt, welche für die Umsetzung der beiden erwähnten Webdocumentaries verwendet wurde. Zur besseren Übersicht werden diese Möglichkeiten als instruktive Schritt-für-Schritt Anleitungen aufgeführt, die auf konkreten Beispielen der Umsetzung beider vorgestellter Webdocumentaries beruhen.

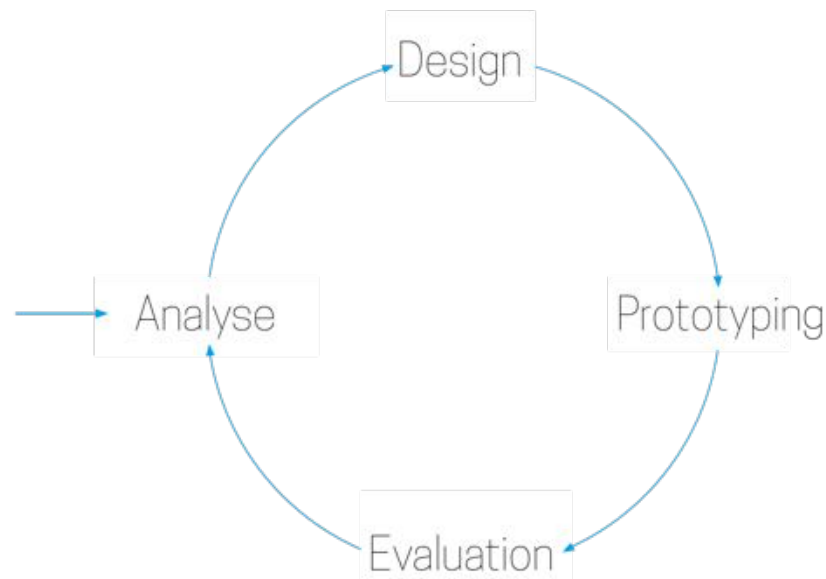


Abbildung 1: Vereinfachtes UX-Lifecyclemodell nach Hartson und Pyla

Teile dieses Kapitels basieren auf der Seminararbeit von Oliver Barwig [1].

Bereits 1989 produzierte das Apple Multimedia Lab eine bis dato unbekannte Präsentationsform. Eine Dokumentation mit dem Titel „Moss Landing“¹ zeigte Videos, die das tägliche Leben der Bewohner des gleichnamigen Hafens darstellte – gespeichert in einer Datenbank und abrufbar vom Benutzer durch ein Hyperlink-Interface. Dies war eine der ersten, interaktiven Dokumentationen. Obwohl dieses System nicht „online“ war, legte es den Grundstein für die zukünftige Entwicklung webbasierter, interaktiver Dokumentarformate.

Aston und Gaudenzi definieren in ihrem Artikel „Interactive Documentary: setting the field“ die *Interactive Documentary* als jedes Projekt, welches die Intention verfolgt, das Reale zu dokumentieren und dabei digitale und interaktive Technologien zur Realisierung nutzt [3]. Auch Dayna Galloway führt eine ähnliche Definition an. Nach ihr bezeichnet der Begriff eine Dokumentation, die im Kern Interaktivität integriert und zur Umsetzung nutzt [5]. Die Webdocumentary ist damit eine Weiterentwicklung der interaktiven Dokumentation und eine Portierung dieser ins Internet. Die angeführten Definitionen sind treffend, allerdings wird der mediale Aspekt der Webdocumentary nicht explizit erläutert. „Digitale und interaktive Technologien“ werden nicht genauer benannt, der Vergleich zu bestehenden Medien fällt dadurch schwer. Der folgende Abschnitt führt deshalb technische und mediale Eigenschaften einer Webdocumentary genauer aus.

Untersucht man verschiedene Webdocumentaries, stellt man fest, dass hierbei meist ein Narrativ mit journalistischem, beziehungsweise dokumentarischem Charakter im Vordergrund steht, welches oft filmisch realisiert wird. Dieses Narrativ besitzt in einer Webdocumentary meist eine Besonderheit: Es ist in der Regel nicht linear, sondern in verschiedene Handlungsstränge unterteilt, die der Nutzer individuell durchlaufen kann. Durch diese Nichtlinearität des Handlungsstrangs und die daraus folgenden Auswahlmöglichkeiten für den Nutzer wird die Webdocumentary interaktiv. Gut veranschaulichen lässt sich dies durch eine Metapher eines Straßensystems mit Autobahnen, Bundesstraßen und Feldwegen: Während die bisherige Dokumentation aus Film oder Buch einem exakt vorgegebenen Weg in-

¹ <http://docubase.mit.edu/project/moss-landing/> – zuletzt aufgerufen am 17.05.2016

nerhalb dieses Systems entspricht, können in der Webdocumentary verschiedene Wege individuell „befahren“ werden. Die Webdocumentary nutzt Medien wie Video, Audio & Text und verschmilzt diese zu einem modernen, digitalen Medium – dargestellt im Browser und implementiert durch Technologien wie HTML, CSS und Javascript. Hierdurch besitzt die Webdocumentary unter anderem folgende mediale und technische Eigenschaften:

- **MULTIMODALITÄT** beschreibt aus psychologischer Sicht die Informationsaufnahme durch verschiedene Sinneskanäle [9]. Informatisch betrachtet ist es die Interaktion von Computer und Mensch durch die Verwendung von Ein- und Ausgabemodalitäten wie Maus, Tastatur und Bildschirm. Diese Modalitäten können beispielsweise auf Spracheingabe oder haptisches Feedback ausgeweitet werden [15].
- **MULTIMEDIALITÄT** bezeichnet die parallele Existenz und Integration mehrerer, unterschiedlicher Medien. Issing und Klimsa sehen darin überdies hinaus ein Konzept, dass zwei Dimensionen — eine technische und eine anwendungsbezogene — integriert. Die technische Dimension umfasst sowohl zeitunabhängige Medien wie Text und statische Grafiken, als auch zeitabhängige Medien wie Video, Audio und Animationen. Die anwendungsbezogene Dimension bezieht Datenbanken, Kommunikationssysteme, Hypermedia, Umgebungen sowie Tools und virtuelle Realitäten mit ein [9].
- **INTERAKTIVITÄT** beschreibt eine wechselseitige Kommunikation von zwei Objekten, bei der Informationen ausgetauscht werden. In Bezug auf die Webdocumentary wird hier auf eine Definition von Rafaeli zurückgegriffen, welcher die Interaktivität als variable Charakteristik von Kommunikationsumgebungen sieht. In einem Informationsaustausch beziehen sich nachfolgende „Nachrichten“ zu einem bestimmten Grad auf vorhergehende, wodurch sich eine gewollte Rekursivität ergibt [16]. Die Webdocumentary betrachtend, bezeichnet Interaktivität hierbei die Entscheidungen des Nutzers, einen Handlungsstrang zu begeben. Je nach Anordnung der Handlungsstränge ist die Auswahl und der darauffolgende Handlungsstrang abhängig von den Vorhergehenden.
- **MULTILINEARITÄT** wird in der Webdocumentary durch die Verwendung unterschiedlicher Handlungsstränge herbeigeführt, die nicht linear angeordnet sind.
- **NAVIGATION & ORIENTIERUNG** belegen eine Schlüsselposition in der Webdocumentary, nicht zuletzt, weil sie die Wahrnehmung eines Benutzers in Bezug auf die Usability der Webanwendung

beeinflussen [4]. Webdocumentaries lassen sich der Klasse von Webanwendungen zuordnen und benötigen, gerade weil sie viele, nichtlineare Handlungsstränge besitzen, eine gute Navigation. Dies stellt sicher, dass ein Nutzer stets in der Lage ist, sich einen Überblick zu verschaffen und zwischen den verschiedenen Handlungssträngen wechseln kann.

Je nach Form der Webdocumentary werden nicht immer alle der genannten Eigenschaften verwendet. Kate Nash beispielsweise sieht diese Eigenschaften als Schnittmenge aus Multimedialität und Interaktivität unter dem Dach einer dokumentarischen Arbeit [13]. Unter Betrachtung einiger zeitgemäßer Webdocumentaries, lässt sich diese Definition um die Eigenschaft der Multilinearität erweitern. Dies impliziert auch die bereits von Nash genannte Interaktivität, weil der Nutzer sich durch eine Abfolge von Handlungssträngen bewegt. Trotz der vorgestellten Eigenschaften stellt sich die Frage der Distanz zu anderen Medien, die dem ersten Anschein nach bereits einen Großteil der vorgestellten Eigenschaften repräsentieren. Tatsächlich liegt der Hauptunterschied zu anderen Medien in der Interaktion mit diesen und was die Interaktion letztendlich herbeiführt. Tabelle 1 zeigt eine Gegenüberstellung der Eigenschaften unterschiedlicher Medien. Deutlich zu erkennen ist hierbei, dass sich die Webdocumentary insbesondere durch die Eigenschaften *Interaktion* und *Narration* von den anderen Medien abgrenzt – dies deckt sich auch mit der von Kate Nash definierten Schnittmenge.

Tabelle 1: Gegenüberstellung unterschiedlicher Medien bezüglich ihrer technischen und medialen Eigenschaften

EIGENSCHAFTEN	WEBSITE	BUCH	FILM	WEBDOKU
Multimodalität	■	×	■	■
Multimedialität	■	×	□	■
Multilinearität	■	×	×	■
Interaktivität	■	×	×	■
Narrativ	×	■	■	■
Navigation	■	□	×	■

Legende: ■ = erfüllt □ = teilweise erfüllt × = unerfüllt

Zusammenfassend lässt sich die Webdocumentary als modernes Medium im Web mit dokumentarischem Charakter bezeichnen. Unter einem nichtlinearen Narrativ aus mehreren Handlungssträngen verschmilzt sie digitale Medien wie Text, Video, Bilder und Audio und gibt dem Nutzer die Möglichkeit, diese interaktiv zu explorieren.

2.1 ANWENDUNGSFÄLLE

Die Webdocumentary lässt sich in unterschiedlichen Anwendungskontexten einsetzen. So dient sie der Aufbereitung und Darstellung journalistischer und dokumentarischer Inhalte und wird hierfür bereits intensiv von einigen Filmproduktionsfirmen wie ARTE² genutzt. Das Einsatzgebiet der Webdocumentary lässt sich jedoch auch auf andere Bereiche übertragen und erweitern. Im Folgenden werden exemplarisch zwei Anwendungsfälle besprochen, die durch zwei universitäre Projekte entstanden sind.

2.1.1 Ausstellungskontext

Zwischen der Universität Konstanz und der HTWG Konstanz entstand innerhalb von vier Semestern die Ausstellung *Tell Genderes - 20 Meter Menschheitsgeschichte*, welche vom 19.12.2015 bis zum 31.01.2016 im Bildungsturm Konstanz gezeigt wurde. Im interdisziplinär angelegten Projekt konzipierten Studenten aus verschiedenen Fachrichtungen (Architektur, Geschichte, Informatik, Kommunikationsdesign) eine Ausstellung, die verschiedene Perspektiven auf einen über 5000 Jahre alten Siedlungshügel in Syrien zeigt, den frühere Bewohner des Ortes Schicht für Schicht geformt haben. Die Ausstellung bot die Möglichkeit, diesen Hügel durch Exponate und interaktive Installationen im Kontext eines eigens entwickelten Raumdesigns zu erforschen. Für die Ausstellung wurde die Webdocumentary *Kulturerbe in Gefahr! Raubgrabungen, Zerstörung und illegaler Handel von Antiken in Syrien* entwickelt. Sie bot dem Besucher einen zusätzlichen Informationskanal innerhalb der Ausstellung und war Teil der Installation im dritten Obergeschoss des Bildungsturms. Sie konnte auf einem Apple iPad abgespielt werden und zeigte beispielsweise die fiktive Reise einer antiken Vase, die außer Landes geschmuggelt wird. Zudem waren interaktive Interviews mit Experten aus verschiedenen Themengebieten und Kontextinformationen zu Ländern, Handel und Objekten zu sehen. Um die simultane Benutzung aller vier iPads zu gewährleisten, wurden Kopfhörer verwendet. Abbildung 2 und Abbildung 3 zeigen Bilder der Webdocumentary als iPad-Anwendung. Eine Onlineversion der Anwendung ist verfügbar unter:
<http://tell-genderes.de/webdoc/>.

² <http://www.arte.tv/sites/de/webdocs/?lang=de> – zuletzt aufgerufen am 24.05.2016



Abbildung 2: Die Webdocumentary als iPad-Anwendung der Ausstellung im Bildungsturm, gerade in Benutzung

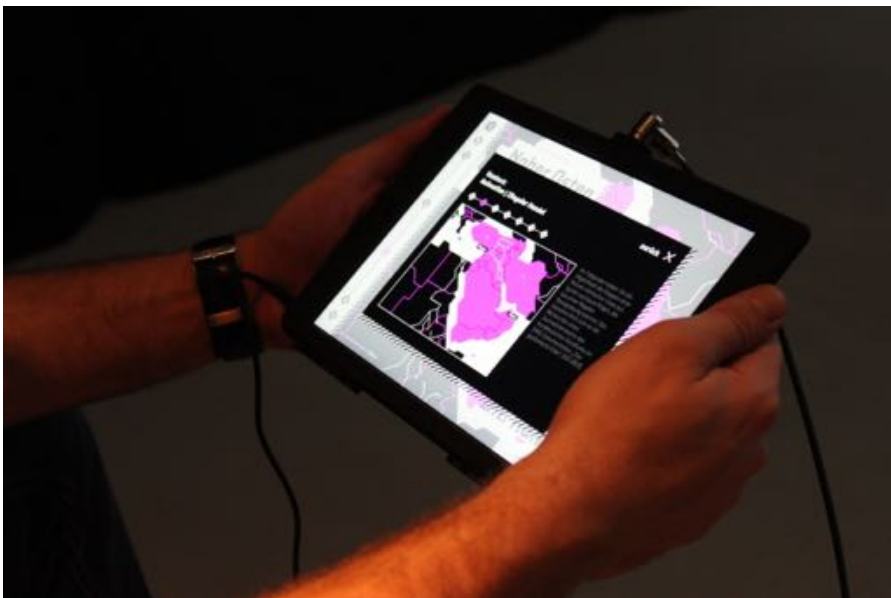


Abbildung 3: Die iPad-Anwendung der Ausstellung im Bildungsturm mit Fokus auf den Bildschirm

2.1.2 *Präsentationskontext*

Zum 50-jährigen Jubiläum der Universität Konstanz entstand eine Webdocumentary in enger Zusammenarbeit der Fachbereiche Literatur- mit Kunst- und Medienwissenschaften und Informatik & Informationswissenschaften, Arbeitsgruppe Mensch-Computer-Interaktion.

Mit der Webdocumentary soll das Image der Universität durch neue Medien nach außen präsentiert und über soziale Medien gestreut werden. Zudem soll sie Studierenden, Außenstehenden und Ehemaligen als Informationsquelle dienen, Emotionen wecken und die Entwicklung der Universität aufzeigen. Kern der Webdocumentary zum 50-jährigen Jubiläum der Universität sind Videos, Geschichten und Hintergrundinformationen zu bisher wenig bekannten Orten der Universität, die der Nutzer interaktiv begehen und erleben kann. So sind unter anderem die Katakomben, die Bienenstöcke auf dem Dach oder eine Reise auf einem Tablett durch die Essensausgabe der Mensa Bestandteil der Webdocumentary. Die einzelnen Sequenzen sind zu Touren zusammengefasst, die auf einer thematisch zusammenhängenden Reise erlebt werden können. Die Webdocumentary integriert ein geographisches und zeitliches Navigationskonzept. Zum einen kann der Benutzer vom Ausgangspunkt Foyer abhängig von der (simulierten) Tageszeit in unterschiedliche Touren starten, zum anderen bietet eine von jeder Sequenz aus abrufbare Karte der Universität Übersicht über die unterschiedlichen Touren und Sequenzen. Diese kann auch für die direkte Navigation zu einer für den Benutzer interessanten Sequenz genutzt werden. Abbildung 4 zeigt die Webdocumentary zum 50-jährigen Jubiläum der Universität am Jubiläumstag, dem 24.06.2016.

Das Projekt ist seit dem 24.06.2016 im Web unter uni.kn/50jahre/webdoku verfügbar und kann von jedem Gerät mit Internetbrowser, das Full-HD darstellen kann, aufgerufen werden.



(a)



(b)



(c)

Abbildung 4: Die Webdocumentary zum 50-jährigen Jubiläum der Universität

KONZEPTIONELLE UMSETZUNG

Das folgende Kapitel arbeitet entlang des UX-Lifecyclemodells nach Hartson und Pyla [8] und stellt in den einzelnen Phasen konzeptionelle Leitlinien für die Umsetzung einer Webdocumentary vor, die auf den Methoden des UX-Lifecyclemodells basieren und auf die Entwicklung einer Webdocumentary angepasst wurden. Grundlage für die Anpassungen bilden die Erfahrungswerte aus der Entwicklung der beiden vorgestellten Webdocumentaries aus Kapitel 1 und aktuelle Methoden und Techniken aus verwandten Disziplinen wie der Webentwicklung. Für einen besseren Überblick leitet Abbildung 5, welche die einzelnen Abschnitte und Unterabschnitte dieses Kapitels repräsentiert, in jeden Abschnitt ein. Die Phase, welche im entsprechenden Abschnitt behandelt wird, ist dunkelblau eingefärbt, abgehandelte Schritte werden durch eine hellblaue Farbe repräsentiert. An dieser Stelle leitet Abbildung 5 in den folgenden Abschnitt, die *Analysephase* ein.

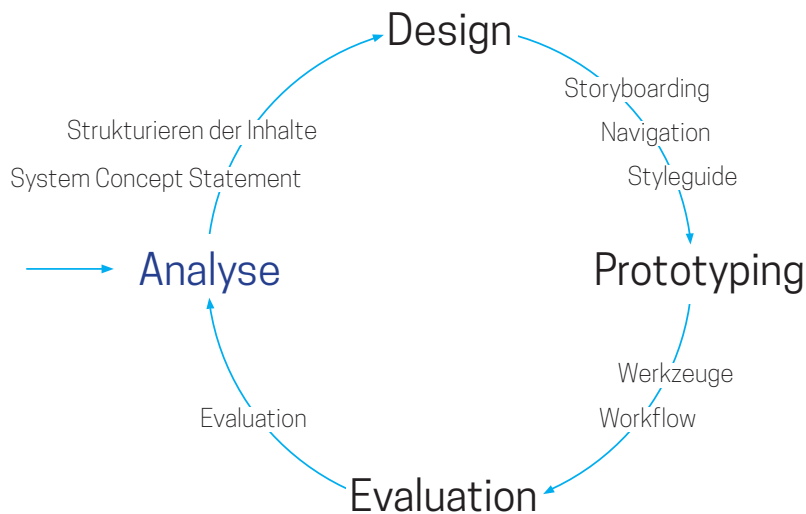


Abbildung 5: Konzeptionelle Umsetzung anhand des UX Lifecyclemodells.
Aktuelle Phase: Analyse

3.1 ANALYSEPHASE

Im UX-Lifecyclemodell nach Hartson und Pyla dient die Analysephase dazu, die Arbeitsweise des Nutzers und seine Anforderungen an eine Anwendung zu verstehen. Mit Techniken wie Interviews, Beobachtungen am Arbeitsplatz und Fragebögen werden diese erhoben. Anschließend werden daraus Requirements extrahiert, die in der darauffolgenden Designphase verwendet werden können [8]. Beim Entstehungsprozess der Webdocumentary können diese Punkte jedoch nicht ohne weiteres übernommen werden, weil Inhalte und Anforderungen oftmals bereits durch den Anwendungskontext vorgegeben sind – beispielsweise in einer Ausstellung die eine klare thematische Vorgabe hat. Die hier vorgestellten Leitlinien sind daher aus der Analysephase des UX-Lifecyclemodells übernommene und an die Webdocumentary angepasste Prozesse.

3.1.1 *System Concept Statement*

LEITLINIE 1:

Schreibe ein System Concept Statement.

Webdocumentaries entstehen aufgrund ihrer vielzähligen medialen Komponenten und des technischen Rückgrats oft in interdisziplinären Teams. Die beiden umgesetzten Webdocumentaries aus Abschnitt 2.1 zeigen, wie vielfältig solche Teams zusammengestellt sein können. Für die Webdocumentary im Bildungsturm setzte sich das Team aus einem Informatik- und einem Geschichtsstudenten der Universität sowie zwei Kommunikationsdesignstudenten der HTWG zusammen. Bei der Webdocumentary zum 50-jährigen Jubiläum der Universität arbeiteten LKM-Studenten mit unterschiedlichen Schwerpunkten wie Film, Audio und Storytelling gemeinsam mit einem Informatikstudenten an der Umsetzung. Eine sehr gute Methode, um alle Beteiligten auf den gleichen Wissenstand bezüglich der Projektidee zu bringen und einen Fokus für das Projekt zu setzen ist das *System Concept Statement*. Hartson und Pyla definieren dieses folgendermaßen:

„A system concept statement is a concise descriptive summary of the envisioned system or product stating an initial system vision or mandate; in short, it is a mission statement for the project.“ [8]

Ein System Concept Statement sollte zwischen 100 und 150 Wörtern lang und so geschrieben sein, dass es auch Außenstehenden das Konzept und die Idee des Projekts vermitteln kann. Es soll keine Informationen zur technischen Umsetzung der Webdocumentary enthalten – Design, Navigationsstrukturen und Inhalte werden in einer eigenen,

konzeptionellen Phase erstellt. Laut Hartson und Pyla sollte das System Concept Statement anderen Fragen zu Systemnutzern, dem Systemnamen und der Systemfunktion beantworten [8]. Allerdings ist hierbei anzumerken, dass diese Anforderungen an ein System Concept Statement stark an die Softwareentwicklung angelehnt sind – daher müssen diese in Bezug auf eine Webdocumentary leicht abgeändert, beziehungsweise ergänzt werden. Die Entwicklung der beiden Webdocumentaries aus den Anwendungsfällen hat gezeigt, dass ein System Concept Statement unumgänglich ist. Für die Anwendung im Bildungsturm wurde kein System Concept Statement formuliert, was später zu Missverständnissen führte. Konkret wurden hierbei einige Elemente der Navigation beim Screendesign „vergessen“, weil ein spezieller Anwendungsfall nicht festgehalten worden war. Für die Webdocumentary zum 50-jährigen Jubiläum der Universität wurde ein System Concept Statement verfasst – es traten keine Kommunikationsprobleme bezüglich der Projektidee auf. Die Fehler aus dem einen und die Erfolge aus dem anderen Projekt bilden die Grundlage für Fragen, die ein System Concept Statement für eine Webdocumentary beantworten sollte:

- Welchen Zweck soll die Webdocumentary erfüllen?
- Wer wird die Webdocumentary benutzen?
- Welches Konzept steckt hinter der Webdocumentary?
- Welche Medien werden verwendet?

Der folgende Auszug aus der Projektskizze der Webdocumentary zum 50-jährigen Jubiläum der Universität von Prof. Dr. Harald Reiterer, Daniel Klinkhammer und Oliver Barwig zeigt exemplarisch das System Concept Statement des Projekts.

„Anlässlich des fünfzigjährigen Bestehens beabsichtigt die Universität Konstanz mittels einer interaktiven Webdokumentation das Image der Universität mit Hilfe neuer Medien nach außen zu präsentieren und über soziale Medien zu streuen. Neben Studenten soll dieses Medium auch Mitarbeitern und Ehemaligen als Informationsquelle dienen, Emotionen wecken, Geschichten erzählen und die Entwicklung der Universität aufzeigen. Das narrative, interaktive und transmediale Konzept einer Webdokumentation bietet das Potenzial, den Besucher aktiv einzubinden und ihm Inhalte abseits der klassischen Informationskanäle zur Verfügung zu stellen. Durch Verwendung neuartiger Medienkombinationen und ästhetischer Gestaltung wird die Präsentation einer emotionalen Perspektive auf die Konstanzer Universität ermöglicht und deren geschichtlichen Hintergrund auf interaktive Art und Weise beleuchtet.“

3.1.2 Strukturieren der Inhalte

LEITLINIE 2:

Strukturiere die Inhalte der Webdocumentary.

Nachdem durch das System Concept Statement der Fokus und die Ziele für die Webdocumentary gesetzt wurden, müssen nun die Inhalte analysiert, strukturiert und später gegebenenfalls auch produziert werden. Je nach Webdocumentary und Kontext sind jedoch die Inhalte bereits vorgegeben oder es gibt spezielle Vorgaben zur Produktion. Sowohl im Ausstellungskontext, als auch im Präsentationskontext geht es meist um ein bestimmtes Thema, lediglich das Format der Präsentation muss festgelegt werden. Inhalte können Videos, Texte, Bilder und Musikdateien sein. Es ist jedoch wichtig, dass sich diese thematisch zusammenfassen und zu einer Story zusammenfügen lassen. Aus diesem Grund sollte zu Beginn dieser Phase analysiert werden, welche Inhalte vorliegen, welche Inhalte produziert werden müssen und wie diese sich thematisch zusammenfassen lassen.

Die beiden aufgeführten Anwendungsfälle haben gezeigt, dass sich dies sehr gut durch ein Inhaltsverzeichnis realisieren lässt, welches die grobe Struktur der Webdocumentary aufnimmt und die Inhalte der jeweiligen Sequenzen aufführt. Bei der Webdocumentary im Bildungsturm wurden die Inhalte durch einen Geschichtsstudenten zusammengetragen. Dieser übernahm auch in Kooperation mit einem Studenten der HTWG die Durchführung der Interviews und deren filmische Produktion. Interviewfragen wurden bereits im Voraus festgehalten, die gesamte Webdocumentary in Abschnitte unterteilt und beschrieben, welche Inhalte in welchem Abschnitt verfügbar sein sollen. Umgesetzt wurde dies durch eine Powerpoint-Präsentation, die leicht weitergeleitet und bearbeitet werden konnte. Abbildung 6 zeigt Ausschnitte aus diesem Dokument.



Abbildung 6: Auszug aus dem Inhaltsverzeichnis der iPad-Anwendung zur Ausstellung

Die Vorgehensweise zur Entwicklung der Webdocumentary zum 50-jährigen Jubiläum der Universität war ähnlich. Auch hier wurde bereits im Voraus besprochen, welche Inhalte vorliegen, wie sich diese thematisch zusammenführen lassen und welche Inhalte noch produziert werden müssen. Hierzu wurde eine Übersicht aus fünf Touren angelegt, die ein Nutzer in der Webdocumentary durchlaufen kann. Diese besagten fünf Touren enthielt dann Informationen über die zugehörigen Inhalte. Allerdings wurde dieses Dokument nicht intensiv gepflegt und ausgearbeitet, wodurch im späteren Verlauf immer wieder Fragen auftraten, wo welche Inhalte in der Webdocumentary verortet sind. Der direkte Vergleich der beiden Anwendungsfälle zeigt deutlich, wie wichtig es ist, bereits im Voraus die Inhalte zu analysieren und zuzuordnen. Je genauer diese Arbeit ausgeführt wird, desto leichter fällt später das Storyboarding und die Implementation.

Die Inhaltsphase hilft zu verstehen, welche Inhalte präsentiert werden sollen und in welcher Form sie produziert werden müssen. In interdisziplinären Teams hilft es zusätzlich, Aufgaben zur Erstellung der Inhalte gezielt bestimmten Personen zuzuweisen. Für die Entwicklung der Webdocumentary zum 50-jährigen Jubiläum der Universität wurden so vier Produktionsteams für Film, Schnitt Audio und Text ins Leben gerufen.

3.2 DESIGNPHASE

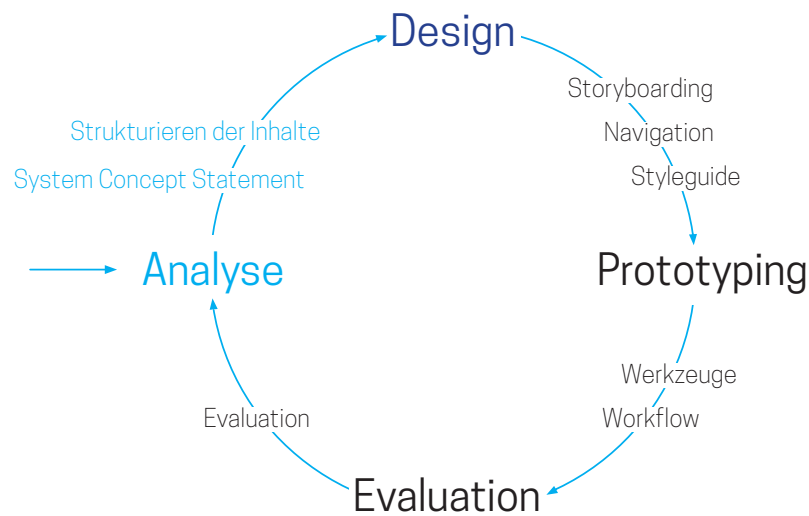


Abbildung 7: Konzeptionelle Umsetzung anhand des UX Lifecyclemodells.
Aktuelle Phase: Design

Die Designphase folgt auf die Analysephase im UX-Lifecyclemodell. Sie dient dazu, ein konzeptuelles Design für eine Applikation zu entwickeln und mögliche Nutzerinteraktionen zu beschreiben [8]. In Bezug auf die Webdocumentary umfasst diese Phase jedoch vielmehr das visuelle Design, die Darstellung der Inhalte und das Entwickeln eines intuitiven Navigationskonzepts. Einige Techniken wurden allerdings aus dem klassischen UX-Lifecyclemodell übernommen und angepasst. Abbildung 7 repräsentiert die aktuelle Position im UX-Lifecyclemodell und zeigt, welche Schritte in dieser Phase behandelt werden.

3.2.1 Storyboarding

LEITLINIE 3:
Entwickle ein Storyboard.

Nachdem klar ist, welche Inhalte vorliegen und welche noch produziert werden müssen, ist es wichtig, diese kontextuell einzuordnen und zu verknüpfen. Eine oft verwendete Technik, die Übersicht schafft, den Ablauf und die Nutzerinteraktionen beschreibt, ist das *Storyboarding*. Das Verfahren stammt ursprünglich aus der Filmbranche und wird dort verwendet, um bereits vor der Produktion einer Szene skizzenhaft darzustellen, was in dieser passieren wird. Auch im Usability Engineering wird es häufig benutzt, um Nutzerinterak-

tionen für bestimmte Anwendungsfälle einer Applikation darzustellen – meist eingebettet in einen zeitlichen Kontext [7]. Das Storyboard in der Webdocumentary soll darüber hinaus jedoch Verknüpfungen zwischen den einzelnen Sequenzen darstellen. Es soll Zusammenhänge zwischen ihnen visualisieren und damit Wege aufzeigen, die ein Nutzer in der Webdocumentary begehen kann. Je spezifischer diese Verknüpfungen sind, desto leichter fällt später die Implementation. Eine solche Übersicht ermöglicht im späteren Verlauf auch, Anforderungen an die Navigation zu extrahieren.

Ein solches an die Webdocumentary angepasstes Storyboard kann auf verschiedene Arten entwickelt werden, beispielsweise mit Papier und Stift, in einem Worddokument oder mit unterschiedlichen Werkzeugen, die eine Art Mindmap-Struktur erzeugen können.

Auch Bilder, Farben und kurze Texte sind hilfreich und können verwendet werden. Gemeinsam mit dem Inhaltsverzeichnis aus Unterabschnitt 3.1.2 entsteht so ein ganzheitliches Konzept der Webdocumentary mit Sequenzen, Inhalten, deren Verknüpfungen und Struktur. Während das visuelle Storyboard der iPad-Anwendung des Bildungstums nur anfangs ausgearbeitet und später hauptsächlich durch intensive Beschreibungen in Textform weitergeführt wurde, nutzte man bei der Webdocumentary zu 50-jährigen Jubiläum der Universität das Werkzeug „Open Mind“¹ zur Visualisierung aller verknüpften Handlungsstränge mit ihren jeweiligen Untersequenzen. Abbildung 8 zeigt das gut ausgearbeitete Storyboard der Webdokumentation zum 50-jährigen Jubiläum der Universität. Zu erkennen ist das Foyer als Dreh- und Angelpunkt der Webdocumentary. Von hier gehen alle Touren ab, von diesen wiederum die untergeordneten Handlungsstränge. Das Storyboard zeigt die Struktur der Webdocumentary auf und hilft später bei der Ausarbeitung eines Navigationskonzepts. Bezüglich des Formats und der Struktur ist dieses Storyboard aus Abbildung 8, gemeinsam mit den explizit ausgeführten Inhalten aus Abbildung 6 der iPad-Anwendung ein guter Ausgangspunkt für die Entwicklung einer Webdocumentary.

Die Software „Klynt“ bietet dem Anwender auch eine Storyboard-Funktionalität, welche alle Verknüpfungen und Verlinkungen der einzelnen Sequenzen darstellt. Diese Funktion kann für die Ausarbeitung eines Storyboards genutzt werden, sollte dann allerdings nicht direkt für die Entwicklung der Webdocumentary weiterverwendet werden. Klynt legt automatisch Buttons für Verlinkungen und Thumbnails für Sequenzen an. Diese später zu entfernen, beziehungsweise zu korrigieren, ist aufwändig und führt erfahrungsgemäß häufig zu Fehlern. Ein „frischer Start“ eines Projekts auf Basis eines mit Klynt erstellten Storyboards ist allerdings empfehlenswert.

¹ <http://openmindsoftware.org/> – zuletzt aufgerufen am 26.06.2016

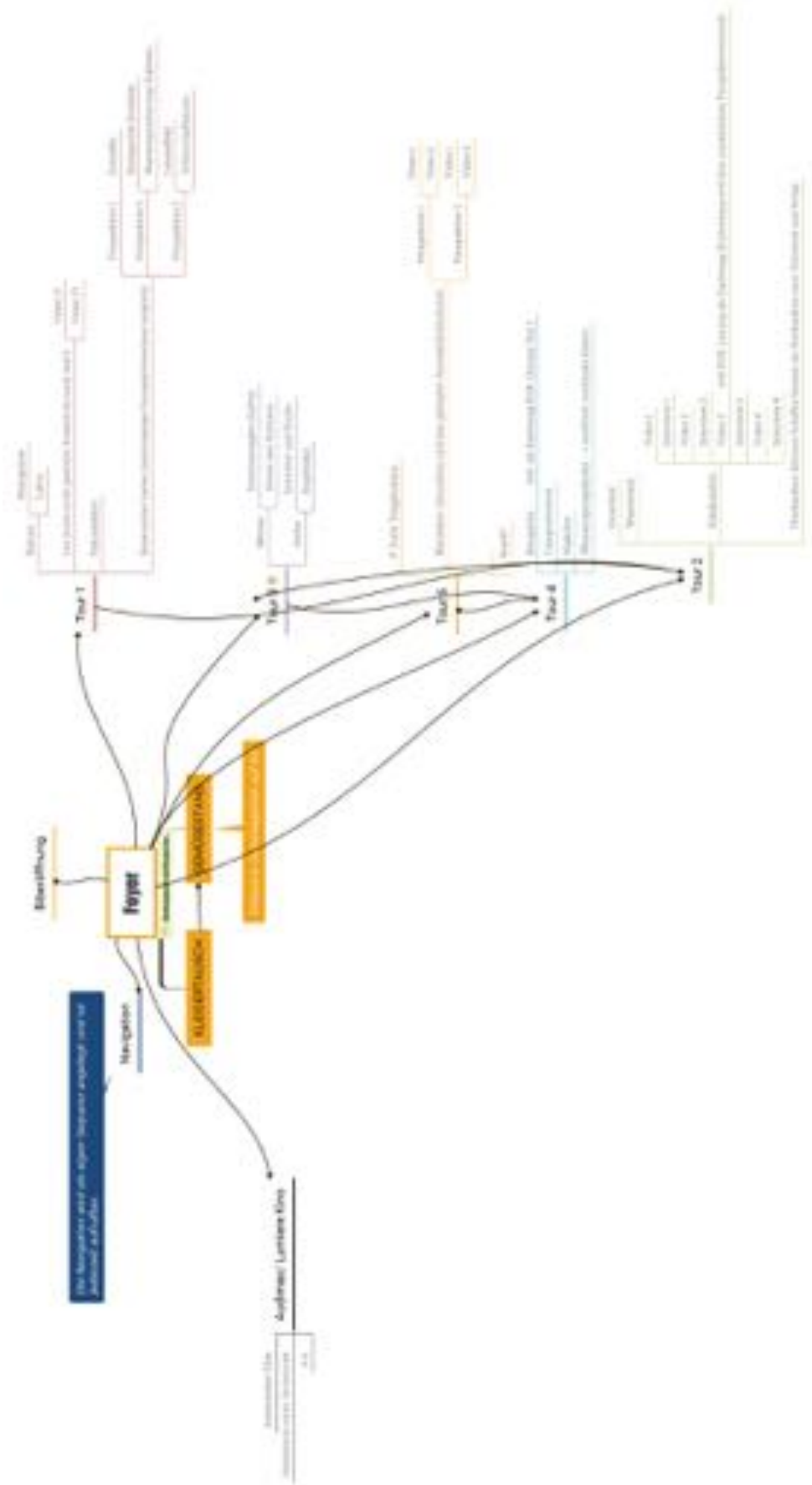


Abbildung 8: Storyboard der Webdocumentary zum 50-jährigen Jubiläum der Universität

3.2.2 Navigation

LEITLINIE 4:

Entwickle ein intuitiv benutzbares Navigationskonzept.

Wie bereits in Kapitel 2 erwähnt, spielt die Navigation eine Schlüsselrolle in der Webdocumentary. Durch die nicht lineare Struktur der Handlungsstränge muss die Navigation zunächst zwei Eigenschaften erfüllen: Sie sollte dem Nutzer Überblick verschaffen und ihm die Möglichkeit geben, durch die unterschiedlichen Handlungsstränge der Webdocumentary zu navigieren. Darüber hinaus sollten weitere Punkte, welche das Erscheinungsbild und die Benutzerfreundlichkeit betreffen, beachtet werden. Im Folgenden werden diese besprochen. Weil allerdings keine spezifischen Informationen zur Umsetzung der Navigation in einer Webdocumentary vorliegen, bietet es sich an, im Bereich der Webentwicklung nach Lösungsansätzen zu suchen. Denn Webseiten sind in der Regel nicht linear, sondern basieren auf Hyperlinkstrukturen – auch hier stehen Inhaltsübersicht und die Möglichkeit, diese Inhalte aufzurufen, im Vordergrund.

Die Nielsen Norman Group hat am 29. November 2015 einen Artikel mit dem Titel „Menu Design: Checklist of 15 UX Guidelines to Help Users“² veröffentlicht, der wichtige UX-Aspekte für die Implementation einer Websitenavigation aufzeigt. Die Anforderungen sind eine Weiterentwicklung der 1990 von Moch und Nielsen vorgestellten Checkliste für die Verbesserung von Mensch-Computer Dialogen [12]. Zusammengefasst lässt sich der von Kathryn Whitenton verfasste Artikel in vier Hauptanforderungen an eine Navigation einteilen:

- **SICHTBARKEIT:** Die Navigation sollte an bekannten Stellen positioniert werden (oben links, oben rechts, seitlich) und gut sichtbar sein. Die Sichtbarkeit bezieht sich hierbei auf einen guten Kontrast zum Hintergrund und eine dem Bildschirm angemessene Größe. Dem Nutzer sollte klar sein, dass es sich hierbei um eine Navigation handelt, mit der er interagieren kann.
- **AKTUELLE POSITION:** Dem Nutzer sollte klar sein, wo er sich gerade befindet, beziehungsweise welchem Punkt in der Navigation der aktuell sichtbare Bildschirminhalt entspricht.
- **ÜBERSICHT:** Die Navigationspunkte sollten Aufschluss über ihre Inhalte geben. Hierfür kann neben gut lesbarem Text auch eine zusätzliche Bildsprache verwendet werden. Lokale Untermenüs können verwendet werden, um eng verwandte Inhalte anzuzeigen.

² <https://www.nngroup.com/articles/menu-design/>
– zuletzt aufgerufen am 10.06.2016

- **MANIPULIERBARKEIT:** Dem Nutzer muss klar sein, dass die Elemente der Navigation klickbar sind. Dropdown-Menüs, die per Hover erscheinen, sollten nur sparsam eingesetzt werden.

Diese Anforderungen decken die zwei zu Anfang des Kapitels eingeführten Eigenschaften des Überblickbehaltens und der Navigation zu den Inhalten der Webdocumentary zwar im Bezug auf Interaktion und visuellen Ansprüchen zwar ab, allerdings wird kein konkretes Umsetzungskonzept vorgeschlagen. Diesbezüglich muss die Struktur der jeweiligen Webdocumentary genauer untersucht werden. Das Storyboard ist hierfür ein geeignetes Werkzeug, weil es Struktur, Wege und Inhalte der Webdocumentary vorgibt und dadurch die Anforderungen an die Navigation stellt. Es muss lediglich entschieden werden, auf welche Art und Weise der Nutzer zwischen den einzelnen Sequenzen und Handlungssträngen navigieren kann. In der Webentwicklung haben sich bereits seit längerer Zeit einige Konzepte etabliert, darunter:

- **MENÜBASIERTE NAVIGATION:** Diese wird meist durch Listenelemente umgesetzt, welche zum entsprechenden Inhalt verlinken. Dieses Konzept wird oftmals durch Dropdownmenüs erweitert. Die Listenelemente werden in der Regel durch einen kurzen Text repräsentiert und können horizontal oder vertikal ausgerichtet sein. Abbildung 9 zeigt ein Beispiel dieses Navigationskonzepts.

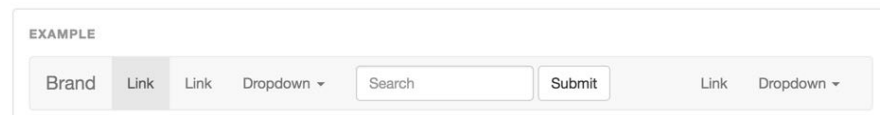


Abbildung 9: Menüstruktur, Quelle: <http://getbootstrap.com/> – zuletzt aufgerufen am 16.06.2016

- **TABBASIERTE NAVIGATION:** Dieses Navigationskonzept implementiert Tabs (Abbildung 10), die auf Klick den darunterliegenden Inhalt preisgeben und wird oftmals verwendet, um auf kompaktem Raum bereits gut unterteilte Informationen anzuzeigen. In der Regel wird der aktive Tab hervorgehoben, was dem Benutzer eine schnelle Zuordnung zum Inhalt ermöglicht.
- **HIERARCHISCHE NAVIGATION:** Ein passendes Beispiel hierfür sind Breadcrumbs, die dem Nutzer anzeigen, wo er sich in der hierarchischen Struktur der Website befindet. Dieses Konzept wird oft verwendet, wenn Navigationsstrukturen sehr tief reichen, beispielsweise in Blogs mit mehreren, teilweise untergeordneten Kategorien. Abbildung 11 zeigt ein entsprechendes Beispiel dieser Struktur.



Abbildung 10: Tabstruktur, Quelle: <http://getbootstrap.com/> – zuletzt aufgerufen am 16.06.2016



Abbildung 11: Breadcrumbstruktur, Quelle: <http://getbootstrap.com/> – zuletzt aufgerufen am 16.06.2016

- **CONTENTVISUALISIERENDE NAVIGATION:** Bekannte Beispiele für dieses Konzept sind Carousels (Abbildung 12) oder kartenbasierte Grids. Es eignet sich gut, um Inhalte visuell darzustellen oder – im Falle des Grids – direkt eine Gesamtübersicht über alle Inhalte zu geben. Dieses Konzept kann durch eine kurze Beschreibung des visuellen Inhaltes und des dahinterliegenden Contents erweitert werden. Die Social Media Plattform Pinterest³ nutzt dieses Konzept ausschließlich.

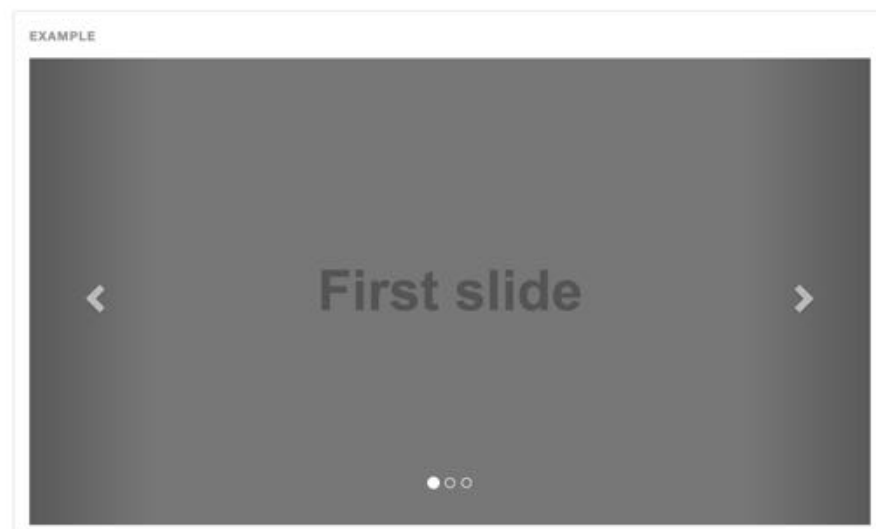


Abbildung 12: Content Carousel, Quelle: <http://getbootstrap.com/> – zuletzt aufgerufen am 16.06.2016

³ <http://pinterest.com> – zuletzt aufgerufen am 16.06.2016

Die angeführten Konzepte können entweder direkt auf Webdocumentaries angewendet, oder mit anderen Konzepten, beispielsweise Realweltmetaphern, kombiniert werden. Abhängig vom Inhalt der Webdocumentary können so interessante, intuitiv zu bedienende Navigationen entstehen. So kann eine geographische Karte als Grundlage für die Navigation einer Webdocumentary dienen, deren Sequenzen an unterschiedlichen Orten spielen. Ein solches Konzept wurde als Hauptnavigation für die Webdocumentary zum 50-jährigen Jubiläum der Universität implementiert. Auch andere Webdocumentaries zeigen die Verwendung dieses Konzepts^{4,5}. Ebenfalls möglich ist die Umsetzung eines explorativen Navigationskonzepts, bei dem sich ein Nutzer mit Hilfe von richtunggebenden Elementen wie Pfeilen horizontal oder vertikal durch die einzelnen Sequenzen bewegt ohne direkt zu sehen, welcher Inhalt ihn erwartet. Dies kann auch mit einem Übersichtskonzept kombiniert werden. Die Webdocumentary der Ausstellung im Bildungsturm integriert Richtungspfeile für die horizontale Bewegung durch die Webdocumentary, besitzt aber dennoch eine Seitenliste mit Markierungspunkten, welche dem Nutzer die aktuelle Position im Kontext aller vorhandenen Sequenzen anzeigt.

Weitere Konzepte können je nach Form und Inhalt der Webdocumentary ebenso Anwendung finden. Denkbar sind hier beispielsweise „räumliche“ Navigationen, bei denen Tafeln oder Schilder Navigationselemente und Übersichten darstellen, oder eine gridbasierte Navigation, die alle Sequenzen als Matrix darstellt. Hierbei gilt es abzuwägen, wie spezifisch dem Nutzer die Inhalte tatsächlich dargestellt werden sollen – in einer explorativ ausgelegten Webdocumentary soll beispielsweise nicht zu viel durch eine komplette Übersicht vorweggenommen werden. Darüber hinaus kann auch das Konzept einer zeitlich orientierten Navigation verwendet werden. Das Konzept einer zeitlich basierten Navigation wurde auch bei der Webdocumentary zum 50-jährigen Jubiläum der Universität verwendet und implementiert eine Sekundärnavigation in Form einer Zeitleiste mit Ankerpunkten, welche dem Nutzer zum einen aufzeigt, wo er sich in einer Tour befindet und zum anderen als Übersicht über vorhandene Sequenzen der jeweiligen Tour dient. Im Folgenden werden die Navigationskonzepte der Webdocumentary im Bildungsturm und der Webdocumentary zum 50-jährigen Jubiläum der Universität näher beschrieben.

4 <http://interactive.nfb.ca/#/outmywindow/> – zuletzt aufgerufen am 12.06.2016

5 <http://www.honkytonk.fr/index.php/webdoc/> – zuletzt aufgerufen am 12.06.2016

Webdocumentary als iPad-Anwendung im Bildungsturm

Die Webdocumentary im Bildungsturm basiert auf einer durch Listenelemente umgesetzten Navigation mit Gestaltung und Layout im Designkonzept der Ausstellung. Die Webdocumentary besteht aus sechs Episoden: *Prolog*, *Einführung*, *Aktuelle Lage Syriens*, *Funktionsweise des Handels im Nahen Osten*, *Schutz und Prävention* und *Epilog*. Jede dieser Sequenzen beinhaltet unter anderem Interviews mit Themenexperten und Kontextinformationen. Für die Implementation wurden folgende Navigationskonzepte und Elemente genutzt:

- **LISTENBASIERTES NAVIGATIONS-KONZEPT:** Eine globale, vertikale Navigation in Form einer Seitenleiste mit Markierungspunkten auf der linken Bildschirmseite repräsentiert die Episoden der Webdocumentary (Abbildung 13) durch rautenförmige Boxen mit Zahlen von null (Prolog) bis fünf (Epilog). Die Elemente führen auf Klick zur entsprechenden Episode. Eine magentafarbene Raute indiziert die aktuelle Sequenz, in welcher sich der Nutzer befindet.
- **EXPLORATIVES NAVIGATIONS-KONZEPT:** Pfeile am rechten und linken Seitenrand ermöglichen dem Nutzer die Navigation zur folgenden oder vorausgehenden Sequenz (Abbildung 13). Sie können geklickt, beziehungsweise getappt werden. Allerdings wurde auch eine Wischgeste implementiert, sodass der Nutzer diese von einem Tablet gewohnte Geste für die Navigation übernehmen kann.
- **ÜBERSICHT FÜR LOKALE INHALTE:** Ein über der globalen Navigation platzierter Informationsbutton (Abbildung 13) liefert dem Nutzer, abhängig vom Handlungsstrang, stets spezifische Inhaltsinformationen zur Sequenz. Ein Klick auf den Informationsbutton öffnet ein Fenster, welches diese Informationen enthält (Abbildung 14).

Beobachtungen während der Ausstellung im Bildungsturm zeigten, dass dieses Navigationskonzept gut funktioniert. Insbesondere die Pfeile wurden häufig zur Navigation benutzt. Manche Nutzer missinterpretierten jedoch das große *Schließen-Icon* in der Informationsübersicht (Abbildung 14) und klickten es häufig, ohne dass sich das Fenster schloss. Ein möglicher Lösungsansatz könnte die tatsächliche Belegung des Icons mit der Schließen-Funktion sein, um dem mentalen Modell des Nutzers zu entsprechen. Bei Funktionstests stellte sich heraus, dass die horizontale Bewegung mit Hilfe der Pfeile der vertikal implementierten Seitenleiste widerspricht. Eine mögliche Verbesserung hierfür wäre eine horizontale Hauptnavigation am oberen oder unteren Bildrand, um eine konsistente Navigation zu gewährleisten.

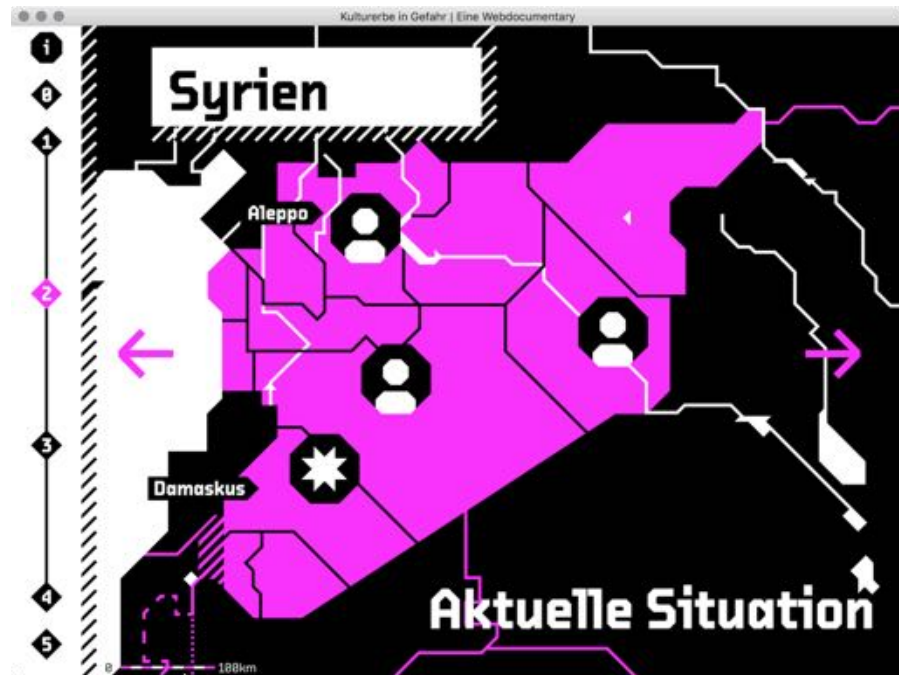


Abbildung 13: Navigationskonzept der iPad-Anwendung. Seitenleiste, Navigationspfeile und Informationsbutton sind sichtbar

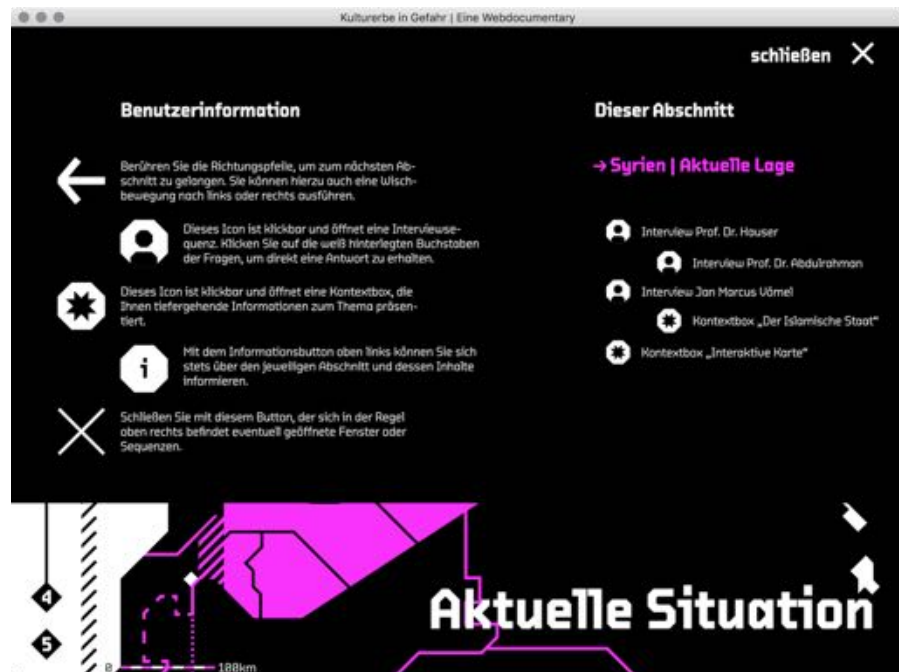


Abbildung 14: Fenster mit Informationen zu den Inhalten der aktuellen Sequenz aus der iPad-Anwendung des Bildungsturms

Webdocumentary zum 50-jährigen Jubiläum der Universität

Die Webdocumentary zum 50-jährigen Jubiläum der Universität integriert mehrere Navigationskonzepte. Ausgangspunkt der Webdocumentary ist das Foyer der Universität, das zu fünf unterschiedlichen Tageszeiten (Morgens, Mittags, Abends, Nachts) und aus unterschiedlichen Perspektiven gefilmt wurde. Jede dieser fünf einzelnen Sequenzen ist der Startpunkt für eine Tour, welche aus thematisch zusammenhängenden und an die Tageszeit angelehnten Handlungssträngen besteht. So kann der Nutzer beispielsweise mittags die Mensa besuchen oder nachts gemeinsam mit dem Team von AIGUK (Astronomische Interessengemeinschaft an der Universität Konstanz) in die Sterne blicken. Die folgenden Elemente wurden für die Umsetzung der Navigation genutzt:

- **GEOGRAPHISCHES NAVIGATIONS-KONZEPT:** Als Hauptnavigation und Übersicht wird ein geographisches Konzept verwendet, das den Grundriss der Universität als stilisierte Illustration von oben zeigt und Handlungsstränge durch Konturen kleiner, blauer Dreiecke symbolisiert – zu sehen in Abbildung 15. Am rechten Seitenrand werden fünf Haupttouren mit den jeweils untergeordneten Handlungssträngen noch einmal in textueller Form präsentiert. Auf diese Weise kann der Benutzer eine direkte Verbindung zwischen dem Handlungsstrang, dessen Namen und Verortung auf der Karte und der übergeordneten Tour herstellen. Werden die Dreiecke oder der Text geklickt, schließt die Navigationsübersicht und der entsprechende Handlungsstrang wird aufgerufen. Die Übersichtskarte kann von jeder Foyersequenz aus über ein oben links in der Ecke positioniertes Icon aufgerufen werden, welches an das Design einer Kompassnadel angelehnt ist. Die Karte besitzt am unteren, rechten Rand eine kleine Legende, die Aufschluss über die verwendeten Icons in der Webdocumentary gibt.
- **ZEITLICHES NAVIGATIONS-KONZEPT:** Touren und Handlungsstränge besitzen eine am unteren Rand positionierte Timeline, visualisiert durch eine Linie mit Markierungspunkten. Diese indizieren zum einen die einzelnen Handlungsstränge einer Sequenz, beziehungsweise Tour, zum anderen zeigen sie dem Nutzer, in welcher dieser Handlungsstränge er sich aktuell befindet. Hierzu wird der Markierungspunkt entsprechend der aktuellen Sequenz eingefärbt. Damit der Nutzer besser unterscheiden kann, ob er sich in einer der fünf Foyersequenzen oder einer Tour mit mehreren Handlungssträngen befindet, wurde für diese beiden Anwendungsfälle ein leicht abgeändertes Design der Timeline gewählt. Im Foyer, dem Ausgangspunkt für eine Tour, markieren konturierte Rechtecke die einzelnen Sequenzen, in

den Handlungssträngen einer Tour werden diese durch zwei dicke Balken ober- und unterhalb der Linie repräsentiert. Abbildung 16 zeigt die Timeline aus dem Foyer, Abbildung 17 die einer begonnenen Tour. Beim Hovern über die Markierungspunkte zeigen diese einen Hinweis in Form eines Tooltips, der textuell Aufschluss über die Tour und deren zugehörige Sequenz gibt (Abbildung 17 unten, mittig). Gleichzeitig dient die Timeline als Navigation – der Nutzer kann auf die jeweiligen Markierungspunkte klicken und gelangt so direkt zur entsprechenden Sequenz. Während der Entwicklung wurde dieses Konzept um kleine Indikatoren für einzelne Handlungsstränge einer Sequenz erweitert. Diese sind über dem Markierungspunkt positioniert und zeigen dem Nutzer an, wenn eine Sequenz mehrere Unterhandlungsstränge besitzt.

- **EXPLORATIVES NAVIGATIONSKONZEPT:** Einige Handlungsstränge der Webdocumentary besitzen Unterhandlungsstränge. Zwar visualisiert auch dort die Timeline mit entsprechend eingefärbtem Markierungspunkt die aktuelle Position des Nutzers, allerdings kann dieser nun mit zusätzlichen erscheinenden Richtungselementen am linken und rechten Seitenrand durch die Untersequenzen steuern und frei von Hinweisen oder Tooltips auf Entdeckungsreise gehen (Abbildung 17). Die Klickfunktion der Timeline-Markierungspunkte bleibt erhalten – der Nutzer kann jederzeit wieder zum Ausgangspunkt zurückkehren oder in einen anderen Handlungsstrang der Sequenz wechseln.

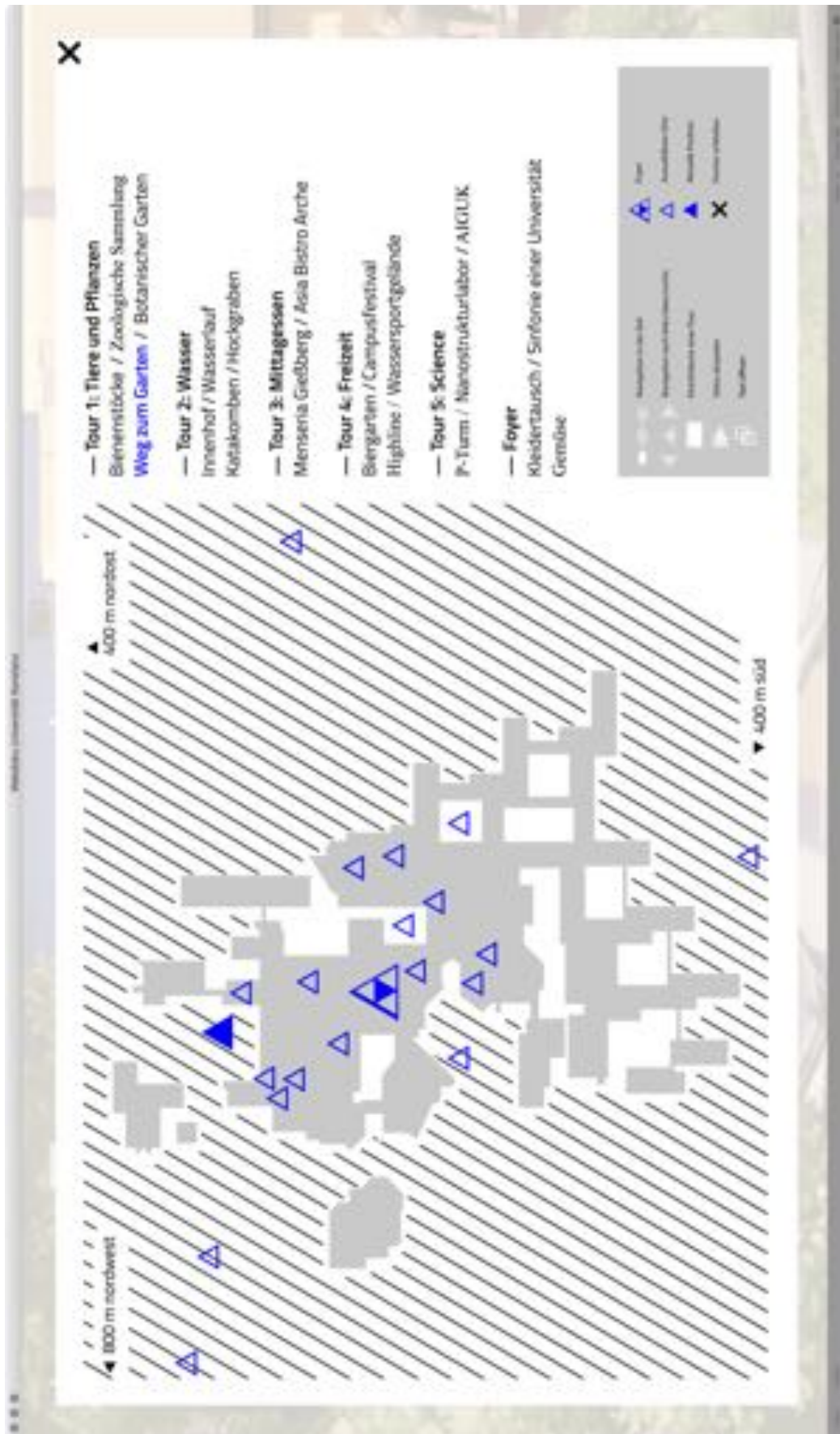


Abbildung 15: Geographische Übersichtskarte der Webdocumentary zum 50-jährigen Jubiläum der Universität, das Element „Weg zum Garten“ ist aktiv



Abbildung 16: Timeline im Foyer der Webdocumentary zum 50-jährigen Jubiläum der Universität, Tour 5 ist aktiv



Abbildung 17: Timeline und Navigationspfeile in einer Untersequenz der Webdocumentary zum 50-jährigen Jubiläum der Universität

Mit den vorgestellten Navigationskonzepten soll kein Anspruch auf Vollständigkeit erhoben werden. Sie sollen jedoch zeigen, wie vielfältig Navigationen umgesetzt werden können. Besonders wichtig ist im Kontext der Webdocumentary, auf die Struktur der Handlungsstränge zu achten – die Navigation soll diese sinnvoll unterstützen und nicht aushebeln. Dies unterscheidet die Webdocumentary mitunter auch von vielen Webseiten. Das Beispiel der kartenbasierten Navigation aus der Webdocumentary zum 50-jährigen Jubiläum der Universität zeigt deutlich, wie gut eine Navigation an die Struktur einer Webdocumentary angepasst sein kann: Alle Handlungsstränge werden repräsentiert, textuelle Informationen zeigen deren Zuordnung zu einer Tour – verraten aber nicht mehr, um nicht zu viel vorwegzunehmen. Gleichzeitig bietet die Karte dem Nutzer die Möglichkeit, zu den für ihn interessanten Sequenzen oder Touren zu navigieren. Ein Zitat von Jakob Nielsen aus dem Artikel „User Interface Design for the WWW“ unterstreicht die Notwendigkeit einer guten und an die Struktur der Anwendung angelehnten Navigation.

„Don't assume that users know as much about your site as you do. They always have difficulty finding information, so they need support in the form of a strong sense of structure and place. Start your design with a good understanding of the structure of the information space and communicate this structure explicitly to the user. Provide a site map and let users know where they are and where they can go. Also, you will need a good search feature since even the best navigation support will never be enough.”[14]

Auch wenn eine Suchfunktion in einer Webdocumentary selten Verwendung findet, weil Nutzer in der Regel nichts von den Inhalten wissen, nach welchen sie suchen könnten, ist Niensons Aussage ein guter Anhaltspunkt, den man bei der Entwicklung einer Navigation ständig im Hinterkopf behalten sollte. Bei der Webdocumentary zum 50-jährigen Jubiläum der Universität stellte sich bei Funktionstests heraus, dass die Navigation einige Probleme aufzeigt, die von den Entwicklern zunächst nicht realisiert wurden, weil sie die Inhalte, Verlinkungen und Strukturen kannten. Wichtig ist jedoch, dass diese auch für den Nutzer klar und verständlich präsentiert werden. In einer Iterationsphase, in welcher das Navigationskonzept um Indikatoren zur Visualisierung von Unterhandlungssträngen und eine andere Positionierung der Richtungspfeile erweitert wurde, konnte die Probleme jedoch gelöst werden.

3.2.3 Styleguide

LEITLINIE 5:

Entwurf einen Styleguide.

Das System Concept Statement, das Inhaltsverzeichnis, das Storyboard und das Navigationskonzept sind wichtige Elemente in der konzeptionellen Entwicklungsphase einer Webdocumentary. Sie bilden die Grundlage für das Design der Anwendung. Die Inhalte sollen den Nutzer auf ästhetischer Ebene ansprechen und die Interaktionselemente als solche wahrgenommen werden. Sie sollten konsistent dargestellt werden, um den Nutzer nicht zu verwirren und ein bestmögliches Nutzererlebnis zu bieten. Damit Designvorgaben festgehalten und später für die Implementierung der Anwendung verwendet werden können, bietet es sich an, einen Styleguide für Farben, Formen, Typographie und Interaktionselemente anzulegen. Ein solcher Styleguide kann in diesem Zusammenhang auch als *User Interface Styleguide* beschrieben werden, weil er in erster Linie Vorgaben an die Benutzeroberfläche definiert. Root und Uyeda beschreiben in ihrem Artikel „Headsup on GUI Styleguides: Report on the CHI’92 Styleguide SIG“ einen User Interface Styleguide folgendermaßen:

„A user interface styleguide is a compendium of rules, recommendations and guidelines for designing a user interface to a given specification. Styleguides are one of the more popular tools around for defining an interface’s look and feel, for ensuring conformance with company and industry standards, and for achieving consistency in interface design within a platform or a product line.“ [17]

Die Nielsen Norman Group definiert im von Page Laubheimer verfassten Artikel „Front-End Style-Guides: Definition, Requirements, Component Checklist“ den Styleguide im Bezug auf Webanwendungen noch etwas genauer.

„Front-end style guides are a modular collection of all the elements in your product’s user interface, together with code snippets for developers to copy and paste as needed to implement those elements. They include common UI components like buttons, form-input elements, navigation menus, modal overlays, and icons.“⁶

Beide Artikel bilden gemeinsam mit den Erfahrungswerten aus der Entwicklung der beiden Webdocumentaries aus Abschnitt 2.1 eine

⁶ <https://www.nngroup.com/articles/front-end-style-guides/> – zuletzt aufgerufen am 18.06.2016

gute Grundlage für den Entwurf eines auf die Webdocumentary angepassten Styleguides. Es muss lediglich beachtet werden, dass eine Webdocumentary im Vergleich zu einer Webanwendung weniger technisch aufgebaut ist und in der Regel keine Date-Time-Picker oder Texteingabefelder enthält. Dafür steht hier ein visuelles, das Thema und Konzept der Webdocumentary unterstützendes Modell im Vordergrund. Deshalb sollten folgende Elemente im Styleguide für eine Webdocumentary enthalten sein:

- **TYPOGRAPHIE:** In der Webdocumentary vorkommende Schriftarten sollten aufgeführt werden. Hieraus muss hervorgehen, wann und wo eine bestimmte Schriftart, beziehungsweise ein Schriftschnitt benutzt wird und welche Eigenschaften dieser bezüglich Größe, Linienhöhe, Laufweite usw. besitzt.
- **FARBEN:** Der Styleguide sollte eine Farbpalette aller im Projekt verwendeten Farben besitzen und in verschiedenen Formaten (RGB, CMYK, HEX) deklarieren.
- **ELEMENTE:** Alle in der Webdocumentary vorhandenen Elemente sollten aufgeführt werden und mit Funktionen, Maßen, Verwendungszweck und Positionsangabe versehen werden. Dies umfasst zum einen Interaktionselemente wie Buttons oder Icons, zum anderen Rahmen, Textfelder oder Videos.
- **CONSTRAINTS:** Hier sollten Vorgaben für bestimmte Elemente definiert werden, beispielsweise, dass sie nicht gemeinsam mit anderen Elementen verwendet werden dürfen.

Der Styleguide selbst sollte gut strukturiert sein, alle erforderlichen Elemente enthalten und in einem Inhaltsverzeichnis auflisten. Jeff Gothelf und Josh Seiden beschreiben in ihrem Buch „Lean UX Applying Lean Principles to Improve User Experience“ drei Charakteristika eines erfolgreichen Styleguides [6]:

1. Der Styleguide soll gut zugänglich sein. Das bedeutet, dass er zum einen von allen Projektbeteiligten jederzeit erreicht werden kann, zum anderen, dass er eine gute Struktur aufweist und leicht durchsuchbar ist. Außerdem stehen gute Verständlichkeit und einfache Benutzung im Vordergrund.
2. Durch das Hinzufügen neuer Elemente kann der Styleguide kontinuierlich verbessert und erweitert werden.
3. Der Styleguide soll ermöglichen, die definierten Elemente einfach zu erstellen, beziehungsweise zu reproduzieren.

In Bezug auf die Webdocumentary ist besonders der letzte Punkt interessant. In beiden Anwendungsfällen dienen die Styleguides zu-

gleich als Designvorlage und Produktionsgrundlage. Weil die Styleguides als Adobe InDesign Dokumente vorlagen, konnten die meisten Elemente direkt als Bilddateien ausgespielt werden.

Abbildung 18, Abbildung 19 und Abbildung 20 zeigen Auszüge der Styleguides für die iPad-Anwendung aus dem Bildungsturm und für die Webdocumentary zum 50-jährigen Jubiläum der Universität. Während für die Webdocumentary zum 50-jährigen Jubiläum der Universität ein dedizierter Styleguide vorlag, war der Styleguide für die iPad-Anwendung nicht explizit auf diese zugeschnitten. Er bestand aus zwei Komponenten: Der Dokumentation für Design und Layout der Ausstellung, worin Farben, Elemente und Typographie erläutert wurden und dem Screendesign für die iPad-Anwendung selbst. Die Arbeit mit beiden Varianten hat gezeigt, dass die Entwicklung einer Webdocumentary mit einem für sie bestimmten Styleguide deutlich leichter fällt, weil Elemente nicht erst aufwändig gesucht werden müssen, sondern direkt verwendet werden können. Vergleicht man das Screendesign (Abbildung 20) und das finale Design (Abbildung 23) der iPad-Anwendung für die Ausstellung im Bildungsturm, stellt man fest, dass diese etwas voneinander abweichen. Im Laufe der Implementation erfolgten einige, kleine Änderungen – beispielsweise das Setzen der Videos im Quer- statt Hochformat oder das Hinzufügen von Farbindikatoren in der Seitenleiste. Aufgrund des vorliegenden Styleguides als InDesign Datei, konnten die Elemente jedoch problemlos geändert und neu produziert werden. Hier zeigt sich, dass es von großem Vorteil ist, wenn ein Styleguide zusätzlich als Produktionsgrundlage dient.

Ein Styleguide setzt voraus, dass es Designelemente gibt, die darin aufgenommen und definiert werden können. In beiden Anwendungsfällen – der Webdocumentary im Bildungsturm und der Webdocumentary zum 50-jährigen Jubiläum der Universität – wurde die Gestaltung des Layouts und aller erforderlichen Elemente von Kommunikationsdesignern übernommen und ausgeführt. Sie kennen wichtige Gestaltungsprinzipien und sind in der Lage, diese mit entsprechenden Werkzeugen umzusetzen. Für die Entwicklung des Designs und auch des Styleguides einer Webdocumentary ist es deshalb unumgänglich, mit einem oder mehreren erfahrenen Designern zusammen zu arbeiten. Gemeinsam mit einem erfahrenen Entwickler, der die wichtigen Aspekte bezüglich guter Benutzerfreundlichkeit und Benutzererfahrung kennt, entsteht so ein starkes Team für die Erstellung interaktiver, ästhetischer und intuitiv zu bedienender Webdocumentaries.

07 | Übersichtskarte

Funktion:

- Die Übersichtskarte zeigt alle Stationen, die in der Webdocumentary besucht werden können auf dem Unigelände verortet an. Die Stationen, die weiter entfernt liegen, werden mit einer Richtungsangabe und Distanz angezeigt.
- Fährt der User mit der Maus über eine Station (blaues Dreieck), fährt sich der Name der Station auf der Liste rechts blau ein.
- Fährt der User mit der Maus über einen Namen in der Stationen-Liste, vergrößert sich das Icon der jeweiligen Station auf der Karte und zeigt so deren Position an.
- Über ein Untermenü auf der Übersichtskarte gelangt man an weitere Informationen, die hinterlegt werden können, z. B. zum Unijubiläum.

Auftreten:

- Aus dem Foyer heraus gelangt man über den Übersichtskarten-Button zur Karte.
- Die Karte legt sich vollflächig über das Foyer-Video, das innehält.

Aussehen:

- Deckkraft: weiße Fläche 80%, alles andere 100%
- Position: Randabstand zu allen Seiten 36 px

Maße:

- Maße: 1208 px x 728 px

Styleguide Webdocumentary Universität Konstanz © AB_Studio für Gestaltung

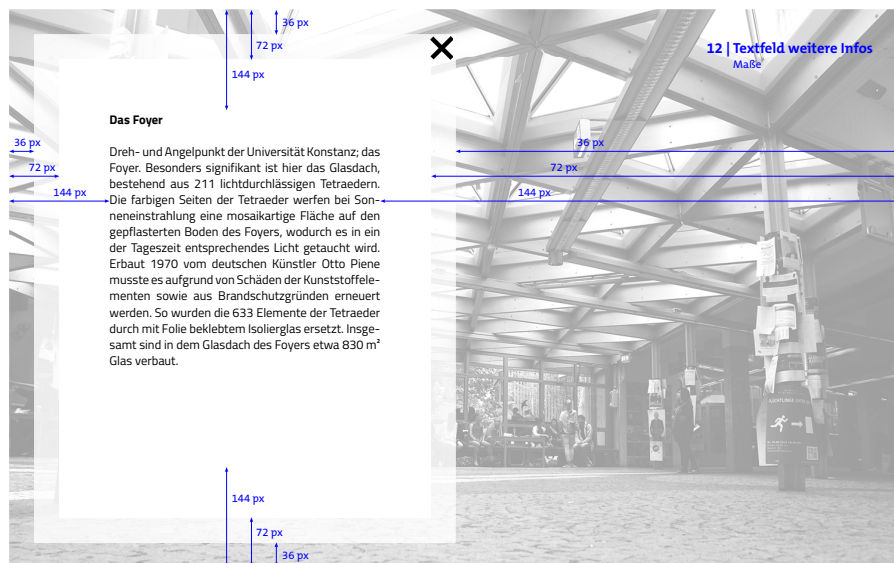
(a)

Inhalt

<p>01 Storyboard Navigation Navigationsprinzip Foyer Navigationsprinzip Foyer – Tour – Foyer Informationen während der Tour</p> <p>02 Farbwelt Grafik Anwendungen</p> <p>03 Typografie Anwendungen</p> <p>04 Maße Schutzraum</p> <p>05 Home-Button Foyer Anwendung Maße</p> <p>06 Übersichtskarten-Button Anwendung Maße</p> <p>07 Übersichtskarte Anwendung Maße</p> <p>08 Icon Foyer-Sequenz Anwendung Maße</p>	<p>09 Zeitleiste Foyer Anwendung Maße</p> <p>10 Eintrittskarte Anwendung</p> <p>11 Zeitleiste Sequenz Anwendung Maße</p> <p>12 Navigation im Raum Anwendung Maße</p> <p>13 Video-im-Video-Button Anwendung Maße</p> <p>14 Video-im-Video-Rahmen Anwendung Maße</p> <p>15 Close-Button Anwendung Maße</p>	<p>16 Info-Button Anwendung Maße</p> <p>17 Textfeld Weitere Infos Anwendung Maße</p> <p>18 Kurzportraits Anwendung Maße</p>
---	---	--

Styleguide Webdocumentary Universität Konstanz © AB_Studio für Gestaltung

(b)



(c)

Abbildung 18: Auszug aus dem Styleguide der Webdocumentary zum 50-jährigen Jubiläum der Universität

04

Vorwort

→
Das Corporate Design ist der Ausdruck und die visuelle Identität der archaischen Ausstellung. Es soll bei der Vermittlung der Inhalte helfen, durch ein einheitliches Auftreten eine Orientierung und einen hohen Wiedererkennungseffekt zu schaffen.
Damit die Ausstellung als starkes Ganzes wahrgenommen wird und sich behaupten kann, ist es notwendig, dass die Charakteristik möglichst oft und mit einer einheitlichen visuellen Sprache kommuniziert wird. Je konsequenter die Elemente des Erscheinungsbildes zusammen spielen, desto überzeugender gelingt dies. Auf diese Weise kann ein unverwechselbares Profil vermittelt werden, das Identifikation stiftet, Vertrauen bildet und einen hohen Wiedererkennungswert generiert.
Die vorlegend Dokumentation beschreibt die Grundregeln des visuellen Auftretens und stellt die Basiselemente für die visuelle Gestaltung im Einzelnen vor. Es soll den Umgang mit dem Corporate Design erleichtern und eine kleine Orientierungshilfe sein.

Inhalt	
A Basiselemente	
06-12	Typografie
13-14	Farben
15-27	Bildwelt
B Gestaltungsprinzip	
29-30	Raster
31-35	Layout
C Anwendungen	
36	Exponatsbezeichnung
37	Thementext
38	Website
39	Tablet-Application
40	Booklet

(a)

08

Typografie
Schriftensippe

Schriftschnitt	Anwendung	Merkmale
1 Optical BB Strata 01 Regular BB Strata 01 Bold	→ Fließtexte und Hervorhebungen Für längere Lesetexte und Hervorhebungen innerhalb von Textpassagen wird zur verbesserten Lesbarkeit eine optisch ausgeglichene Schrift verwendet.	äßl äßl
2 Headline BB Strata H1 Regular BB Strata H1 Bold	→ Überschriften Titel, die über die Schriftgröße der Fließtexte hinausgehen (z.B. Wandüberschriften) werden immer mit einer eigens entwickelten Schrift entworfen.	ößl ößl
3 MonoLine BB Strata M1 Skeleton BB Strata M1 TriLine BB Strata M1 BiLine BB Strata M1 Regular BB Strata M1 Bold	→ Exponattexte und Überschriften Die Schriftfamilie mit der gleichbleibenden Linienstärke wird für die Texte der Exponate eingesetzt. Darüber hinaus werden einzelne Schnitte für Überschriften (s. Anwendungen) der Thementexte verwendet.	ößl ößl ößl ößl

(b)

39

Tablet Application

1

2

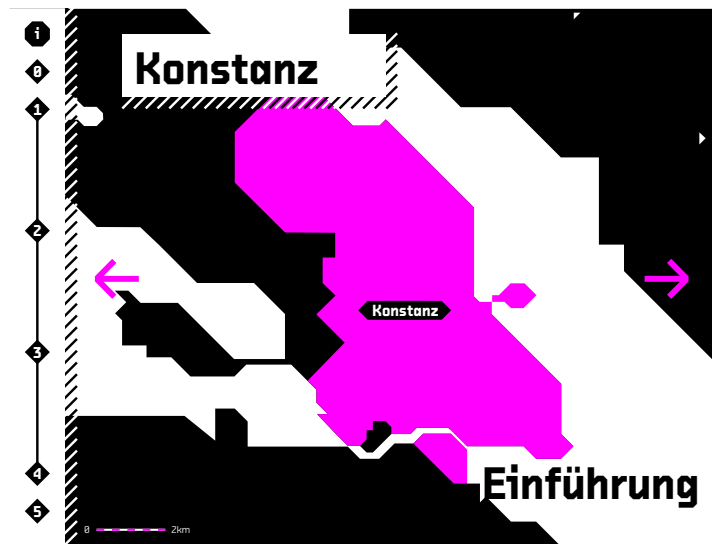
3

4

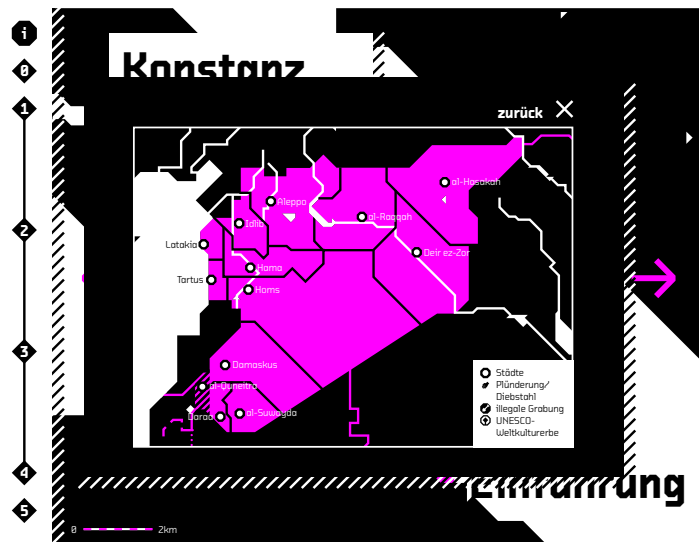
1 Format 2048 x 1536 Px (4:3)
Layouttemplate vorhanden

(c)

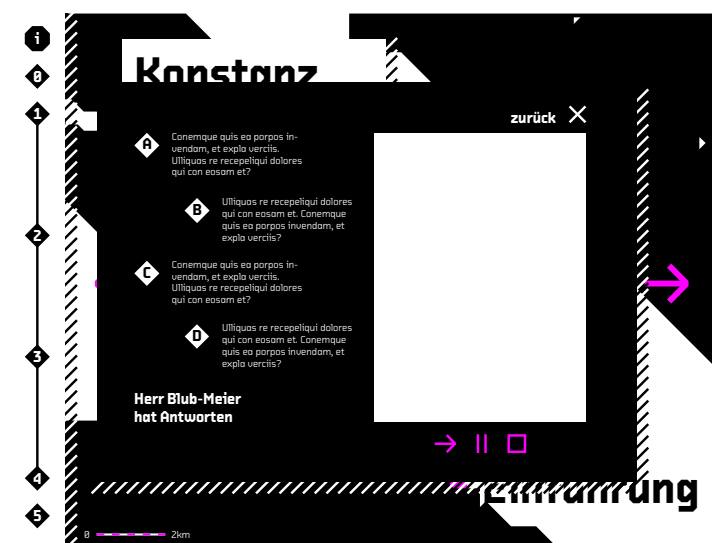
Abbildung 19: Auszug aus dem Styleguide (Corporate Design) der Ausstellung im Bildungstrum



(a)



(b)



(c)

Abbildung 20: Auszug aus dem Styleguide der iPad-Anwendung aus dem Bildungsturm

3.3 PROTOTYPINGPHASE

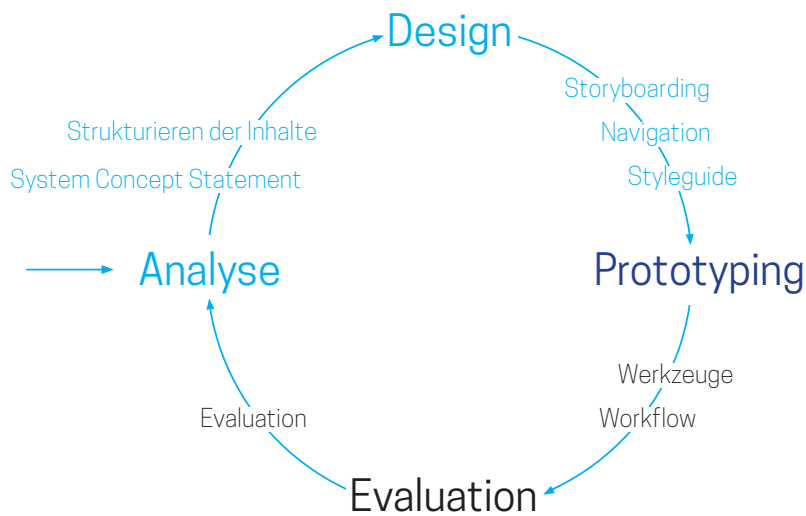


Abbildung 21: Konzeptionelle Umsetzung anhand des UX Lifecyclemodells.
Aktuelle Phase: Prototyping

Im UX-Lifecyclemodell nach Hartson und Pyla folgt nach der Designphase die Prototypingphase (Abbildung 21). Sie umfasst die Realisierung von Designalternativen durch Anwendung verschiedener Methoden, beispielsweise durch die Herstellung von Papierprototypen, welche die Funktionsweise der Applikation zeigen und das Testen dieser ermöglichen sollen. Hartson und Pyla drücken allerdings deutlich aus, dass sich Design- und Prototypingphase oftmals überlappen und der Prototyp in der Regel eine Repräsentation des Designs darstellt [8].

Für den Prototypen einer Webdocumentary ist nicht nur die reine Funktionalität oder eine Impression der späteren Designs wichtig. Vielmehr muss er die Möglichkeit bieten zu prüfen, ob Inhalte visuell gut repräsentiert werden, Handlungsstränge sinnvoll angeordnet sind und die Navigation für den Nutzer verständlich ist. Low-Fidelity Prototypen aus Papier und Zeichnungen eignen sich aus diesem Grund nicht, vielmehr wird eine erste Umsetzung der Webdocumentary als High-Fidelity Prototyp benötigt. Insbesondere die Webdocumentary zum 50-jährigen Jubiläum der Universität hat gezeigt, dass ein High-Fidelity T-Prototyp viel über die bereits erwähnten Punkte zeigen und gleichzeitig als Grundlage für den Ausbau zur finalen Anwendung dienen kann. Konkret wurde dieser T-Prototyp durch eine erste Version der Webdocumentary umgesetzt, bei welcher die erste Tour vollständig ausgearbeitet war. Im Fall beider Webdocumentaries wurde die erste Umsetzung mit Klynt immer wieder erweitert und

in mehreren Iterationsschritten verbessert. So war keine erneute, vollständige Programmierung der Anwendung notwendig, wie es in der Softwareentwicklung häufig der Fall ist. Der folgende Abschnitt beschreibt konzeptionell, welche Schritte notwendig sind, bevor mit der Implementation der Webdocumentary begonnen werden kann.

3.3.1 Werkzeuge

LEITLINIE 6:
Wähle deine Werkzeuge.

Sowohl für die Entwicklung der Webdocumentary, als auch für die Produktion der Inhalte müssen entsprechende Werkzeuge verwendet werden, um die Arbeit zu erleichtern und erforderliche Formate zu erzeugen. Bricht man eine Webdocumentary auf ihre Medienstruktur herunter, so werden neben einem Tool für die Entwicklung der Webdocumentary selbst auch Werkzeuge für die Bearbeitung von Bilddateien, Texten, Videos und Audiodateien benötigt. Die folgenden Werkzeuge wurden für die Entwicklung der Webdocumentary im Bildungsturm und der Webdocumentary zum 50-jährigen Jubiläum der Universität verwendet:

Werkzeuge für die Medienproduktion:

- ADOBE INDESIGN CC 2015 für Layout, Styleguide, Setzen von Texten und Auspielen von Bilddateien im PNG-Format.
- ADOBE ILLUSTRATOR CC 2015 für die das Erzeugen von SVG-Dateien wie der Navigationskarte in der Webdocumentary zum 50-jährigen Jubiläum der Universität.
- ADOBE PHOTOSHOP CC 2015 für die Bearbeitung von Bildern.
- ADOBE PREMIERE PRO CC 2015 für Videoschnitt und das Auspielen der Videos.
- ADOBE AFTER EFFECTS CC 2015 für die Erstellung von Animationen, beispielsweise im Intro der Webdocumentary zum 50-jährigen Jubiläum der Universität.

Werkzeuge für die Entwicklung der Webdocumentary:

- KLYNT für die Implementation der Webdocumentary und dem Erstellen der webfähigen Version.
- SUBLIME TEXT für Code wie CSS, HTML oder Javascript der für Produktion und Optimierung der Webdocumentary notwendig war.
- IMAGEOPTIM für die Kompression von Bildern.

Werkzeuge für die Erstellung von iPad- und Desktopanwendungen, Testing und Dateiupload

- TRANSMIT für die Übertragung der Dateien auf einen FTP-Server.
- XCODE für die Entwicklung und Implementation der Applikation im Bildungsturm.
- NATIVEFIER für das Erzeugen von Desktopanwendungen aus einer Webdocumentary.
- GOOGLE CHROME für das Testen der Anwendung unter Benutzung der Chrome Developertools.

Die meisten der aufgeführten Tools sind Industriestandards. Sie zeigen aber nur eine Auswahl von möglichen Werkzeugen, die für die Entwicklung einer Webdocumentary verwendet werden können. Anhand der vorgestellten Werkzeuge wird deutlich, dass mit steigenden Anforderungen an eine Webdocumentary auch mehr Werkzeuge eingesetzt werden müssen. Praktisch ist es möglich, eine Webdocumentary nur unter Benutzung von Klynt zu erstellen, allerdings ist mit Einschränkungen bezüglich des Navigationskonzepts, des Designs und der Funktionalität verbunden.

3.3.2 *Workflow*

LEITLINIE 7:

Definiere einen Workflow für die Produktion und arbeite nach diesem.

Nachdem eine Auswahl der Werkzeuge für die Entwicklung getroffen wurde, kann ein Workflow für die Verwendung der Tools, die Produktion der Inhalte und die Implementation der Webdocumentary mit anschließender Kontrolle definiert werden. Hierbei ist anzumerken, dass sich die Produktion der Inhalte auf die notwendigen Schritte für den Import in das Erstellungstool für die Webdocumentary bezieht. Über die tatsächliche Produktion von Videos oder Texten kann an dieser Stelle keine Aussage getroffen werden, da diese komplett von LKM-Studierenden übernommen wurde. Auch bei der Webdocumentary im Bildungsturm wurden die Inhalte und Videos durch den Geschichtsstudenten angeliefert. Für die Umsetzung beider Projekte wurde als Werkzeug für die Erstellung der Webdocumentary die Software „Klynt“ verwendet, die momentan die Spitze der am Markt verfügbaren Storytellingtools darstellt. Kein anderes Werkzeug weist einen vergleichbaren Funktionsumfang auf. Klynt selbst wird im Projektbericht von Oliver Barwig [2] genauer beschrieben, alternative Werkzeuge in der vorangegangenen Seminararbeit [1].

Folgende Schritte präsentieren exemplarisch einen Workflow, wie er für die Entwicklung der Webdocumentary zum 50-jährigen Jubiläum der Universität verwendet wurde. Er basiert auf dem Workflow für die Entwicklung der Webdocumentary im Bildungsturm, wurde aber in einigen Punkten überarbeitet und verbessert. Als Grundlage dienen die fertig produzierten Videos, Audiodateien und Texte sowie der Styleguide. Ziel eines solchen Workflows ist es, effizient konsistente Dateiformate zu erzeugen, den Import zu erleichtern und die tatsächliche Implementation zu vereinfachen. Auch das anschließende Testen der Webdocumentary wird hierdurch vereinfacht. Die Schritte 5 bis 8 des stellen einen Iterationszyklus dar, der bis zur finalen Version der Webdocumentary angewendet wird.

1. **KONTROLLE DER INHALTE:** Videos, Audiodateien und Texte werden im Bezug auf Fehler (Videoruckler, Rechtschreibfehler) kontrolliert und gegebenenfalls direkt korrigiert.
2. **AUSSPIELEN:** Anhand des Styleguides erforderliche Interaktionselemente, Textboxen usw. als Bilddateien ausgespielt.
3. **NAMENSgebung:** Alle Elemente, die später in die Webdocumentary importiert werden, bekommen eine konsistente Namensgebung, beispielsweise nach Sequenznamen. Dies dient zum einen

der Übersicht, zum anderen ermöglicht es die eindeutige Zuordnung der Medien zu den entsprechenden Sequenzen.

4. **KOMPRESSION:** Um die Performance der Webdocumentary zu verbessern, werden alle Bilddateien komprimiert. Dies kann je nach Datei mehr als 50% der originalen Größe einsparen. Videodateien wurden bereits komprimiert angeliefert.
5. **IMPORT:** Alle erzeugten Dateien werden in Klynt importiert und können dort nun für das Erstellen der Webdocumentary verwendet werden.
6. **ERSTELLUNG:** Die Webdocumentary mit allen Medien und Sequenzen, ihren Verknüpfungen und der Integration von Navigation und Interaktionselementen wird entwickelt.
7. **KONTROLLE UND KORREKTUR:** Nach Abschluss der Erstellungsphase wird überprüft, ob alle Verknüpfen korrekt gesetzt wurden, Inhalte richtig und vollständig dargestellt werden und alle Interaktionselemente ihrer Funktion entsprechend arbeiten. Das Storyboard der Anwendung kann ein gutes Werkzeug sein, um strukturell alle Sequenzen durchzuarbeiten. Fehler werden gegebenenfalls direkt verbessert.
8. **UPLOAD:** Die aktuelle Version der Webdocumentary wird auf einen FTP-Server geladen und dort abermals getestet. Gegebenenfalls werden hier Funktionalitäten wie die in Unterabschnitt 4.2.4 beschriebene Skalierungsfunktion oder die Verlängerung der Tapdauer implementiert. Auftretende Fehler werden behoben.
9. **APP-ENTWICKLUNG:** Dieser Schritt ist optional, kann aber je nach Anwendungsfall notwendig sein – beispielsweise dann, wenn am Abspielort der Webdocumentary kein Internet vorhanden ist.

Der Workflow ist zum einen von den Anforderungen der Webdocumentary abhängig, zum anderen von den Werkzeugen, die verwendet werden. Keinesfalls sollte jedoch auf die Definition eines Workflows verzichtet werden. Er steigert die Effizienz der Produktionskette, erleichtert die Kontrolle über abgeschlossene Schritte und schafft dadurch Überblick im Entwicklungsprozess. Zudem ermöglicht er, Aufgaben auf verschiedene Mitglieder eines Entwicklungsteams zu verteilen.

3.4 EVALUATIONSPHASE

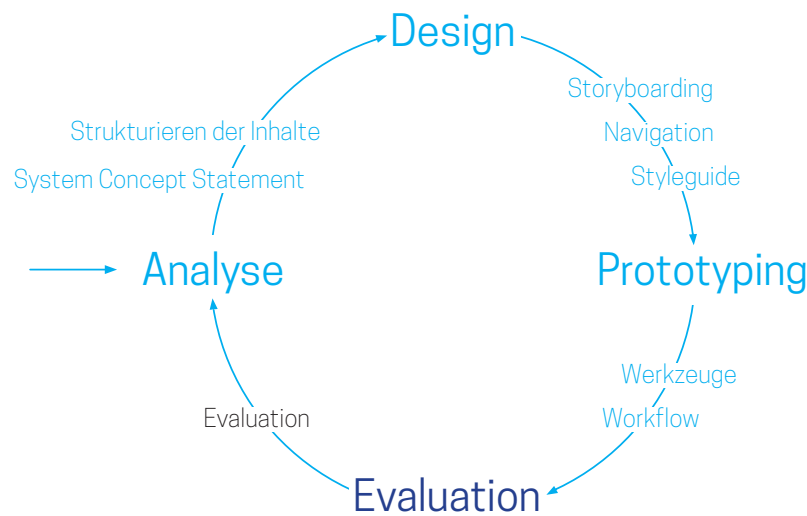


Abbildung 22: Konzeptionelle Umsetzung anhand des UX Lifecyclemodells.
Aktuelle Phase: Evaluation

Die Evaluation ist die letzte Phase des UX-Lifecyclemodells nach Hartson und Pyla. Dies wird in Abbildung 22 noch einmal visualisiert. Unter Verwendung verschiedener Methoden wird in diesem Schritt die Anwendung in Bezug auf Interaktionsdesign, Gebrauchstauglichkeit und Benutzerfreundlichkeit evaluiert [8]. Auf Basis der erhobenen Daten kann gegebenenfalls eine erneute Iteration über den UX-Lifecycle begonnen werden.

LEITLINIE 8:

Evaluieren Sie die Webdocumentary.

Sowohl die Webdocumentary als iPad-Anwendung im Bildungsturm, als auch die Webdocumentary zum 50-jährigen Jubiläum der Universität wurden nicht evaluiert, sondern nur in verschiedenen Entwicklungsstadien Funktionstests unterzogen. Dies liegt zum einen am enormen Zeitdruck, dem beide Projekte unterlagen, zum anderen daran, dass keine spezifische Informationen zur Evaluation einer Webdocumentary vorliegen. Die angesprochenen Funktionstests zeigten einige grundsätzliche Fehler in beiden Webdocumentaries auf, die bereits in dieser Arbeit, unter anderem in Unterabschnitt 3.2.2 angesprochen wurden und auch größtenteils gelöst werden konnten. Allerdings können reine Funktionstests keinesfalls eine Evaluation ersetzen. Auch wenn ein Navigationskonzept für die Entwickler verständlich ist oder Inhalte ohne große Mühe auffindbar sind, hat dies

keine Aussagekraft darüber, wie ein Endnutzer die Webdocumentary bedient. Hat er Spaß an der Benutzung? Kann er Inhalte finden, die ihn interessieren? Sind die Wege durch die Webdocumentary so angeordnet, dass der Nutzer sie versteht? Weisen die Interkationselemente genug Affordance auf, sodass er diese klickt oder tappt? Dies sind unter anderem Fragen, die eine Evaluation beantworten kann. Sie zeigen gleichzeitig auch, wie wichtig diese für den Erfolg einer Webdocumentary ist.

Die Webdocumentary stellt aufgrund ihrer medialen Komponenten, der Erzählstruktur und dem Erschließen der Inhalte besondere Anforderungen an eine Evaluation. Wie eine Webdocumentary unter diesen Aspekten evaluiert werden kann und welche Anforderungen außerdem an die Evaluation gestellt werden müssen, soll in Zukunft untersucht werden. Hierbei können aktuelle Techniken aus der Web- oder Anwendungsentwicklung wie die Heuristiken der Nielsen Norman Group⁷ die Grundlage bilden, auf welchen Methoden für die Evaluation einer Webdocumentary aufbauen. Weitere Vorschläge zu möglichen Evaluationsmethoden werden im abschließenden Kapitel dieser Arbeit „Reflexion und Ausblick“ vorgestellt.

⁷ <https://www.nngroup.com/articles/ten-usability-heuristics/> – zuletzt aufgerufen am 26.06.2016

3.5 ZUSAMMENFASSUNG DER LEITLINIEN

Der folgende Abschnitt fasst die in dieser Arbeit vorgestellten Leitlinien noch einmal zusammen und ergänzt sie mit einer kompakten Zusammenfassung der wichtigsten Punkte, die bei der Anwendung der Leitlinie zu beachten sind.

LEITLINIE 1:
Schreibe ein System Concept Statement.

Es sollte nicht länger als 150 Wörter sein und Fragen zum Zweck der Webdocumentary, der Nutzergruppe, dem Konzept und den Medien, die verwendet werden sollen, beantworten. Es soll keine technische Beschreibung sein, sondern allen Beteiligten dazu dienen, den Projektfokus und die Projektidee zu definieren.

LEITLINIE 2:
Strukturiere die Inhalte der Webdocumentary.

Schreibe dazu ein Inhaltsverzeichnis, das die Sequenzen der Webdocumentary grob strukturiert. Aus dem Inhaltsverzeichnis soll hervorgehen, welche Medien und Inhalte in einer Sequenz verwendet werden. Es hilft außerdem, Aufgaben zur Erstellung einzelner Inhalte direkt den Mitgliedern eines Entwicklungsteams zuzuweisen.

LEITLINIE 3:
Entwickle eine Storyboard.

Ein Storyboard soll die Sequenzen verknüpfen und Wege durch die Webdocumentary aufzeigen. Eine netzartige Struktur und die Verwendung von Bildern und kurzen Beschreibungen hilft, dem Storyboard mehr Aussagekraft zu geben und die Verknüpfungen besser zu verstehen. Gemeinsam mit dem Inhaltsverzeichnis bildet es die Grundlage für die weitere Entwicklung der Webdocumentary.

LEITLINIE 4:
Entwickle ein intuitiv zu benutzendes Navigationskonzept.

Die Navigation spielt eine Schlüsselrolle in der Webdocumentary. Das Navigationskonzept soll die Inhalte und Struktur der Webdocumentary aufnehmen und für den Benutzer zugänglich machen. Sie soll gut sichtbar positioniert sein, dem Nutzer anzeigen, wo er sich gerade befindet, und ihm die Möglichkeit geben, zu für ihn interessanten Inhalten zu navigieren.

LEITLINIE 5:
Entwurf einen Styleguide.

Der Styleguide soll Farben, Formen, Typographie und Interaktionselemente wie Buttons definieren, visualisieren und ausführlich beschreiben. Er dient als Grundlage für die Erstellung von Interaktionselementen und Inhalten und sollte die Möglichkeit bieten, diese auch direkt zu produzieren. Eine dem Projekt angepasste Struktur des Styleguides hilft bei der Zuordnung der Elemente und schafft Übersicht.

LEITLINIE 6:
Wähle deine Werkzeuge.

Die Werkzeuge zu Entwicklung einer Webdocumentary beeinflussen den Entwicklungsaufwand und das Ergebnis maßgeblich. Deshalb sollten auf das Projekt angepasste Werkzeuge gewählt werden, die den Entwicklungsprozess unterstützen und vereinfachen.

LEITLINIE 7:
Definiere einen Workflow für die Produktion und arbeite nach diesem.

Ein Workflow ermöglicht es, konsistente Dateiformate zu erzeugen, den Import von Dateien zu erleichtern und die Implementation zu vereinfachen. Schritte in der Entwicklung werden hierdurch nachvollziehbar und können einzelnen Personen des Teams zugewiesen werden.

LEITLINIE 8:
Evaluere die Webdocumentary.

Die Abschließende Evaluation hilft, Fehler im Navigations- und Interaktionskonzept aufzudecken, die durch reine Funktionstests während der Entwicklung nicht gefunden oder übersehen wurden. Die Evaluation hilft außerdem, die Benutzererfahrung und Benutzerfreundlichkeit der Webdocumentary zu testen. Die Ergebnisse dienen gegebenenfalls als Basis für eine erneute Iteration des UX-Lifecycles.

Es liegen keine spezifischen Informationen zur Evaluation einer Webdocumentary vor. Deshalb ist es notwendig, diese Leitlinie in der Zukunft auszuarbeiten und genauer zu untersuchen.

Das folgende Kapitel beschreibt zum einen die Entwicklung einer Webdocumentary als Offline-Anwendung in Form einer iPad- und Desktopapplikation, zum anderem Möglichkeiten, um die Funktionalität der Software „Klynt“ zu erweitern. Die vorgestellten Punkte basieren größtenteils auf Praxisbeispielen und Problemstellungen aus der Entwicklung von zwei Webdocumentaries und werden als Schritt-für-Schritt Anleitung mit instruktivem Charakter präsentiert, um die einzelnen Schritte nachvollziehbar und reproduzierbar zu machen. Für die Umsetzung der Arbeiten wird jedoch grundlegendes Wissen über die aktuellen Webtechnologien und Erfahrung im Umgang mit diesen benötigt.

4.1 DIE WEBDOCUMENTARY ALS APP

Es existieren Anwendungsfälle, welche die Entwicklung einer Webdocumentary als Offline-Anwendung erfordern. Dies ist beispielsweise dann der Fall, wenn am Ausstellungsort kein Internet vorhanden ist oder Anforderungen an die Webdocumentary gestellt werden, die nicht ohne weiteres durch die Verwendung von Standardsoftware (Browser) erfüllt werden können. Die folgenden Abschnitte zeigen zwei Möglichkeiten einer technischen Umsetzung hierzu: Die Entwicklung einer Webdocumentary als iPad-Anwendung, wie sie in der Ausstellung im Bildungsturm gezeigt wurde und die Umsetzung einer Webdocumentary als Desktop-Anwendung. Dieser spezielle Fall basiert auf keinem konkreten Beispiel, kann aber analog zur iPad-Anwendung für einige Anwendungsfälle nützlich sein.

4.1.1 *iPad-Anwendung*

Der folgende Abschnitt erläutert die Entwicklung einer iPad-Anwendung mit einer aus Klynt exportierten Webdocumentary unter Verwendung einer für MacOSX optimierten Entwicklungsumgebung. Der vorgestellte Arbeitsablauf kann nur mit gleicher Konfiguration repliziert werden. Dies umfasst die Verwendung von Klynt als Editor und das Vorliegen der gesamten Webdocumentary als aus Klynt exportiertes Web-Projekt und die Verwendung von XCode als Entwicklungsumgebung. Für den konkreten Anwendungsfall wurde ein iPad Air 2 mit Retinaauflösung verwendet. Teile des folgenden Abschnitts basieren auf dem Projektbericht von Oliver Barwig [2].

Die Webdocumentary der Ausstellung im Bildungsturm – zu sehen in Abbildung 23 – repräsentiert einen klassischen Anwendungsfall für die Implementation einer Webdocumentary als iPad-Anwendung. Dies war zunächst jedoch nicht geplant. Die Webdocumentary sollte ursprünglich über den integrierten Safari-Browser des iPads aufgerufen werden. Schnell wurde jedoch klar, dass dies nicht ohne weiteres umsetzbar ist. Beispielsweise konnte die Tableiste im oberen Bereich des Browsers nicht entfernt werden, was dazu geführt hätte, dass Nutzer trotz „geführten Zugriffs“ eine beliebige URL in die Suchleiste eingeben und so die Applikation verlassen können. Ein Lösungsversuch für dieses Problem war die Verwendung eines anderen, speziell für das Prototyping ausgelegten Browsers¹, der Apps im Fullscreenmodus aufrufen, und die URL-Eingabeleiste ausblenden kann. Allerdings konnte auch hier die Tableiste mit der Eingabemaske für eine Webadresse durch eine spezielle Geste aufgerufen werden. Obwohl es recht unwahrscheinlich erschien, dass ein Nutzer diese Geste ausführt, war dieser Anwendungsfall jedoch nicht auszuschließen.

Der nächste Lösungsansatz bestand in der Implementation einer iPad-Anwendung, welche die Webdocumentary lokal aufruft und anzeigt. Nach mehreren Tests lief diese App auf einem iPad im Netzwerk der Universität perfekt und wurde auf alle vier iPads aufgespielt. Bei der Vernissage der Ausstellung stellte sich jedoch heraus, dass die Internetverbindung öfters abbrach oder stark in der Geschwindigkeit schwankte. Dies hatte zur Folge, dass Videos oftmals nicht schnell genug luden und dadurch das Benutzererlebnis trübten. Überdies hinaus wurde festgestellt, dass Buttons nicht flüssig reagierten, was oft zur Frustration der Nutzer führte.

Das Problem der schlechten Verbindung konnte durch die Verwendung lokaler Daten gelöst werden, d.h. alle Daten der Webdocumentary wurden auf das iPad übertragen und von dort aus lokal aufgerufen. Da Klynt nur standardisierte Webformate wie HTML-, CSS- und Javascriptdateien exportiert, entstehen in dieser Hinsicht keinerlei Probleme. Diese Methode lässt sich auch optimal nutzen, wenn am Abspielort der Webdocumentary kein Internet vorhanden ist. Das Problem der träge reagierenden Buttons zu lösen war schwierig, da sich dies nicht über das iPad, beziehungsweise die App regulieren lässt. Vielmehr wurde nach langer Suche das Problem in einem Gesten-Unterstützungsframework (Hammer.js²) gefunden, das Klynt verwendet. Dort sind Taps auf Mobilgeräten nicht explizit definiert, wodurch ein zu kurzer Tap oder zu langes Drücken auf einen Button kein Ereignis auslösen. Außerdem implementieren die meisten Browser ein 300ms Delay auf Buttons oder klickbare Elemente, was zu einer wei-

¹ <https://stakes.github.io/Frameless/> – zuletzt aufgerufen am 12.06.2016

² <http://hammerjs.github.io/> – zuletzt aufgerufen am 12.06.2016

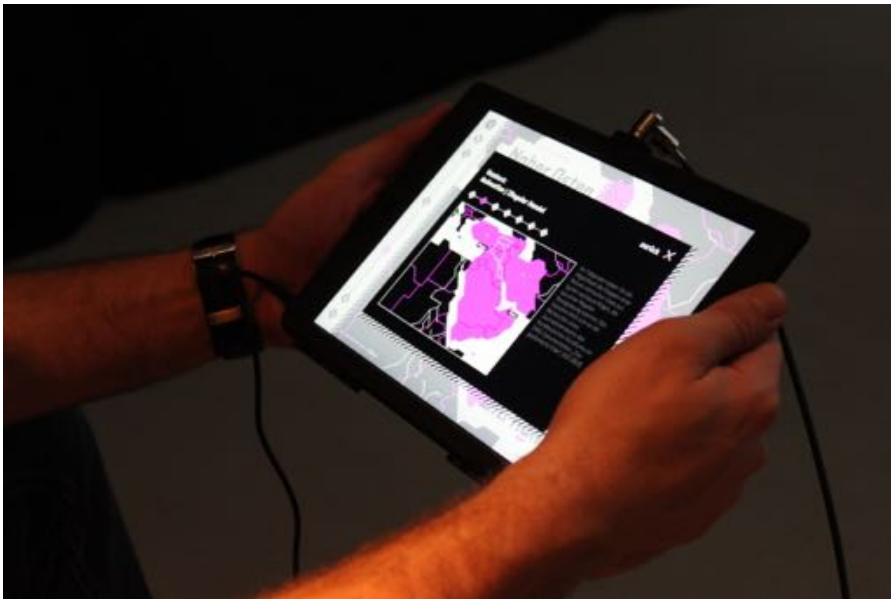


Abbildung 23: Die Webdocumentary „Kulturgüter in Gefahr!“ als Apple iPad-Anwendung in der Ausstellung des Bildungsturms

teren Verzögerung führt. Im weiteren Verlauf der Ausstellung war festzustellen, dass sich die App manchmal nicht mehr starten ließ – in der Regel dann, wenn sie nicht korrekt über den Homebutton und das anschließende „Hinausstreichen nach oben“ beendet worden war. In einem anderen, von der Umsetzung der beiden Webdocumentaries unabhängigen Projekt wurde das Problem gefunden und behoben. Hierzu wird eine Funktion implementiert, welche die App schon bei einem einfachen Klick auf den Homebutton komplett terminiert und dadurch komplett schließt.

Der nächste Abschnitt zeigt den Lösungsansatz mit kommentierten Codebeispielen unter Verwendung einzelner Arbeitsschritte. Der Arbeitsablauf setzt voraus, dass die Webdocumentary in Klynt im 4:3 Format mit 1024×768 Pixeln implementiert wurde. Für eine bestmögliche Auflösungen sollten Dateien jedoch im Format 2048×1536 Pixel in Klynt importiert werden. Diese zunächst verwirrende Diskrepanz hängt mit dem Retinadisplay des iPads und der Exportfunktion von Klynt zusammen, welche nicht nativ ins Retinaformat exportieren kann. Die Verwendung des angegebenen Workarounds löst dieses Problem allerdings. Für die Umsetzung muss außerdem ein Apple-Entwickler Account vorhanden sein, mit dem gültige Zertifikate ausgestellt werden können. Unter Verwendung einer AppleID können zwar ebenfalls Zertifikate ausgestellt werden, allerdings besitzen diese nur eine Gültigkeitsdauer von einer Woche. Da die Zertifikate immer wieder neu ausgestellt werden können, ist dies jedoch eine gute, vorübergehende Alternative, um die App zu testen.

1. Export des Projekts aus dem Klynt-Editor mit der Funktion: *File* → *Package for Web*.
2. Erstellen eines Xcode Projekts mit den Folgenden Vorgaben: *Single View Application*, *Projektname*, *Language: Swift*, *Devices: iPad*, keine Häkchen setzen bei *Use Core Data*, *Include Unit Tests* und *Include Ui Tests*. Die Daten können einfach beim Öffnen eines neuen XCode Projekts angegeben werden.
3. Setzen der korrekten Einstellungen im Hauptmenü der Applikation. Hier ist es wichtig, anzugeben, ob die App beispielsweise nur im Landscape-Modus verwendet werden kann. Außerdem muss das *Deployment Target* auf die Software Version des iPads angepasst, oder das iPad zur entsprechenden Version geupdated werden.
4. Import des Klynt-Projekts. Hierbei wird einfach der komplette Projektordner per Drag&Drop in die linke Seite des XCode-Projekts gezogen. Ein Umbenennung des Ordners in „webapp“ vor dem Import vereinfacht später das Handling.
5. Erstellen der WebView. Dafür im *Main Storyboard* in der Auswahlleiste ganz unten rechts *WebView* auswählen und per Drag & Drop in den *ViewController* in der Mitte des Bildschirms ziehen. Anschließend oben rechts die korrekte Größe von 1024×768 Pixeln definieren.
6. Verknüpfen von *WebView* und *ViewController*. Dafür im *Main Storyboard* in der linken Seitenleiste die *WebView* per Rechtsklick auswählen und dort den kleinen, runden Kreis unter *Referencing Outlets* mit einer Mausbewegung auf den *ViewController* ziehen.
7. Anpassung der *ViewController.swift* Datei nach Listing 6.
8. Anpassung einer Funktion in der *AppDelegate.swift* Datei nach Listing 7.
9. Anpassung der zeitlichen Einstellungen in der Datei „webapp/-Player/js/libs/jquery.hammer.min.js“. Dort nach „tap“ suchen und *time* auf mindestens 1200 setzen.
10. Aufspielen der App auf das iPad. Dazu iPad anschließen, in der Xcode Hauptansicht oben links das Gerät auswählen und den *Playbutton* klicken. Die Inhalte werden nun auf das iPad übertragen. Nach dem Übertragen den *Stopbutton* klicken und das Gerät vom Computer entfernen – das iPad mit der App kann nun verwendet werden.

Abbildung 33 und Abbildung 34 im Appendix dieser Arbeit zeigen Screenshots aus einem XCode-Projekt, die beim Zuordnen der einzelnen Arbeitsschritte helfen können.

4.1.2 Desktop App

WICHTIGE ANMERKUNG: Der folgende Abschnitt erläutert die Entwicklung einer Desktopapplikation mit einer aus Klynt erzeugten Webdocumentary, die unter einer URL im Internet verfügbar ist. Die erläuterten Arbeitsschritte wurden an einem Computer mit MacOSX Betriebssystem ausgeführt.

Unter Umständen verlangen einige Anwendungsfälle die Entwicklung einer Desktopapplikation. Mittlerweile gibt es Werkzeuge, welche dies unter Verwendung von Internettechnologien ermöglichen. Eines davon ist Electron³, ein Framework, durch das man mit HTML, CSS und Javascript native Desktopapps erzeugen kann. Weil bei einer aus Klynt exportierten Webdocumentary all diese Dateien bereits vorliegen, gestaltet sich die Umsetzung im Vergleich zur vorgestellten Entwicklung einer iPad-Anwendung vergleichsweise simpel. Hierzu existiert ein *Commandlinetool* (Nativefier⁴), welches unter Angabe der URL der Webdocumentary die benötigten Daten herunterlädt und in eine Electron-App verpackt. Mit entsprechender Konfiguration lassen sich so Apps erzeugen, die auf den Betriebssystem MacOSX, Windows und Linux laufen.



Abbildung 24: Die Webdocumentary „Kulturgüter in Gefahr!“ als MacOSX Desktopanwendung

³ <http://electron.atom.io/> – zuletzt aufgerufen am 13.06.2016

⁴ <https://github.com/jiahaog/nativefier> – zuletzt aufgerufen am 13.06.2016

Abbildung 24 zeigt die Webdocumentary aus dem Bildungsturm als MacOSX Desktopapplikation. Die Inhalte sind komplett heruntergeladen worden, eine Internetverbindung ist zum Anzeigen der Applikation nicht notwendig. Das Projekt wird folgendermaßen umgesetzt:

1. Installation von `node.js`⁵, eine Javascript-Laufzeitumgebung. Hierzu die Website aufrufen, die Software herunterladen und installieren.
2. Installation von *Nativefier* über das Terminal mit:
`$ npm install nativefier -g`
3. Erzeugen der Desktop-App über das Terminal mit:
`$ nativefier http://url/der/Webdocumentary/`

In der Regel verwendet Nativefier Icon und Namen der App nach deren HTML-Metatags. Um sicher zu gehen, können dem Befehl zum Erzeugen der App direkt beim Aufruf Parameter, sogenannte *Flags*, mitgegeben werden, welche Namen, Größe, App-Icon und Distributionsform bestimmen.

- Größe: `--width <Zahlenwert1>, height <Zahlenwert2>`
- Name: `--name <NameDerApp>`
- Icon: `--icon <Pfad zum Icon>`, idealerweise als `.png` Datei.
- Betriebssystem: `--platform <windows, linux, osx>`

Die Desktopapplikation der Webdocumentary aus dem Bildungsturm aus Abbildung 24 kann beispielsweise durch folgende Terminaleingabe erzeugt werden:

```
$ nativefier http://tell-genderes.de/webdoc
--name AntikenInGefahr
--platform osx --width 1024 --height 768
--icon /Users/oliverbarwig/Desktop/icon-1.png
```

Je nachdem, von welchem Verzeichnis aus der Befehl aufgerufen wurde, befindet sich die App nun in einem Ordner im entsprechenden Verzeichnis. Sie kann jetzt per Doppelklick auf das Icon gestartet werden.

⁵ <https://nodejs.org/en/> – zuletzt aufgerufen am 13.06.2016

4.2 ERWEITERUNGEN FÜR KLYNT

Klynt erlaubt es, große und komplexe Projekte umzusetzen, stößt in speziellen Fällen aber an seine Grenzen. Die Software bietet allerdings einige Möglichkeiten zur Erweiterung. Beispielweise lassen sich Elementen im Klynt Editor CSS-Klassen zuweisen. Dies ist insbesondere dann sinnvoll, wenn Objekte animiert werden sollen – Klynt bietet nativ nur wenig Animationsmöglichkeiten an. Durch die Verwendung von eigenem CSS lässt sich außerdem das Erscheinungsbild des Klynt-Players verändern, wodurch man unter anderem den Footer oder das Aussehen eines der Widget-Menüs verändern kann. Überdies hinaus erlaubt Klynt das Einbinden von IFrames – eine gute Möglichkeit, eigene Inhalte einzufügen oder komplexe Navigationen zu erstellen. Die Übersichtskarte der Webdokumentation zum 50-jährigen Jubiläum der Universität wurde auf diese Weise implementiert. Allerdings lassen sich durch Umwege und Workarounds auch Funktionen einbinden, die Klynt nicht nativ unterstützt. Dies umfasst unter anderem die Verwendung eigener Buttondesigns im Klynt-Editor oder die Abfrage der aktuellen Sequenz in IFrames durch Manipulation der Klynt API. Die folgenden Abschnitte geben hilfreiche Anweisungen für die Umsetzung der vorgestellten Beispiele. Weitere Informationen zur Arbeit mit Klynt, dem Erstellen von Sequenzen und Overlays und der Funktionsweise des Klynt-Editors finden sich im Projektbericht von Oliver Barwig [2].

4.2.1 *Verwendung eigener Buttons*

Obwohl sich die in Klynt vorhandenen Buttons durch den integrierten Editor in Farbe oder Deckkraft verändern lassen, bleibt deren ursprüngliches Erscheinungsbild in Bezug auf die Form erhalten. In der aktuellen Klynt-Version 3.4.400 sind Buttons wie Richtungspfeile, ein Hamburger-Icon oder ein Play-Button enthalten. Abbildung 25 zeigt einige der in Klynt enthaltenen Buttons im Button-Editor.

Der folgende Ablauf beschreibt eine Methode, um die Buttons im Klynt-Editor durch eigene zu ersetzen, die anschließend an Stelle der ursprünglichen Buttons im Projekt verwendet werden können. Leider wurde diese Methode zu spät entdeckt, insbesondere Im Fall der Webdocumentary 50-jährigen Jubiläum der Universität hätte sie die Arbeit deutlich erleichtert. Wie bereits erwähnt, wurden alle Interaktionselemente wie Buttons als Bilddateien aus InDesign ausgespielt. Hovereffekte (Der Button verändert beim Hovern die Deckkraft von 80% auf 90%) mussten hierdurch aufwändig und für jedes Element manuell durch die Advanced-Interactions-Funktion des Klynt-Editors eingefügt werden. Durch der Verwendung eines eigenen Buttondesigns hätte man diesem Button entsprechende Eigen-

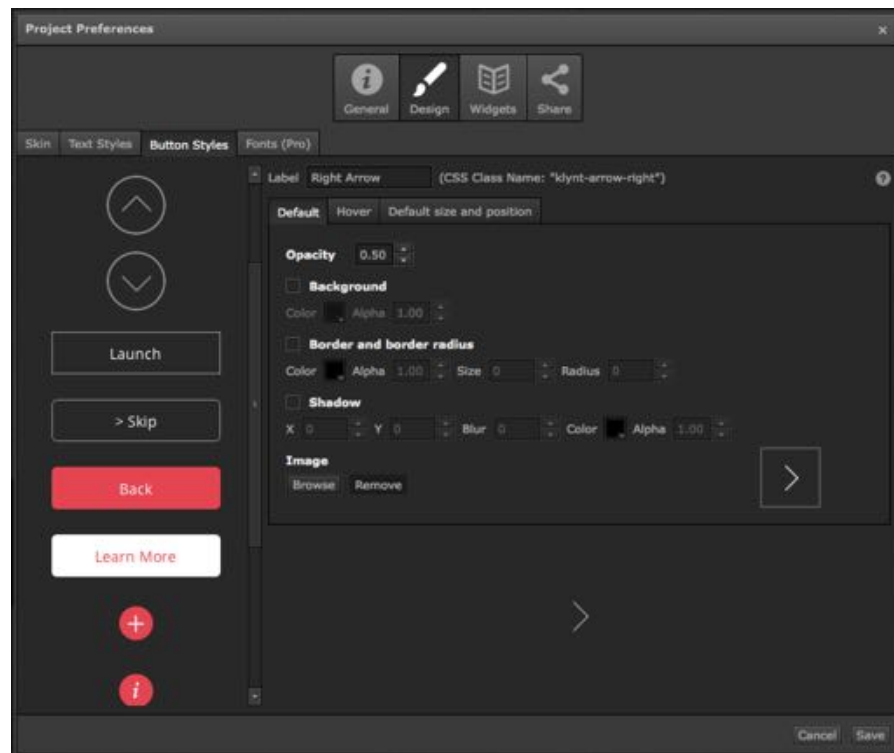


Abbildung 25: Der Button-Editor in Klynt

schaften bereits implementieren, anschließend wäre lediglich noch das Einfügen und Positionieren des Buttons nötig gewesen. Im Folgenden wird beispielhaft der Button für die Anzeige von Textboxen in der Webdocumentary zum 50-jährigen Jubiläum der Universität implementiert, der den Launch-Button in Klynt ersetzt. Um die Sichtbarkeit zu verbessern, wurde die Farbe Rot verwendet. Das Endergebnis ist in Abbildung 26 zu sehen – der reguläre Button (rot) und dessen Hoverzustand (weiß) sind hier zur besseren Übersicht nebeneinander platziert.

1. Erstellen der Buttons als .svg Datei, wahlweise .png oder .jpg Datei. SVG Dateien sind kleiner in der Dateigröße, immer scharf und deshalb zu bevorzugen. Für dieses Beispiel wurde Adobe Illustrator für die Erstellung der .svg-Dateien verwendet, siehe Abbildung 27. Gegebenenfalls sollten im gleichen Zuge auch Buttons für den Hoverstate des Buttons erstellt werden (In diesem Fall die gleiche Datei, nur mit einer anderen Hintergrundfarbe).
2. Hinzufügen der Dateien zu Player/css/editor/img/. Wahlweise kann hier die alte Datei ersetzt, oder die neue Datei unter einem anderem Namen eingefügt werden.

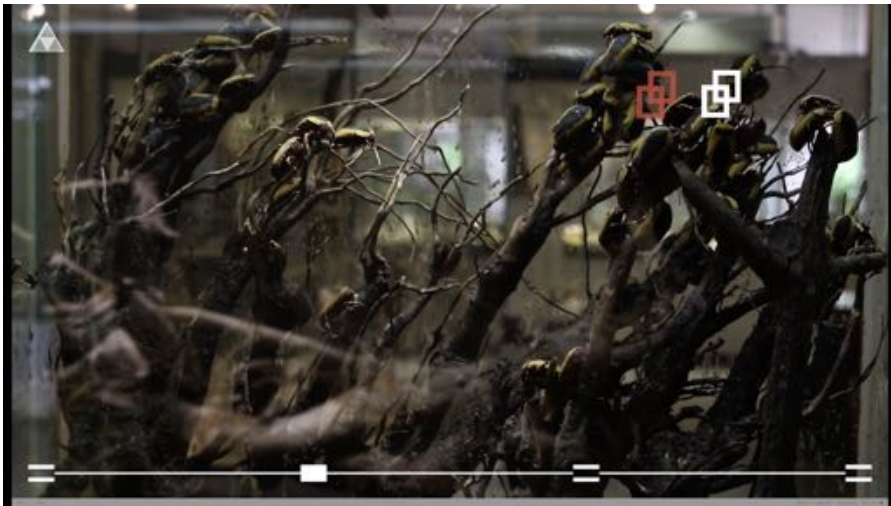


Abbildung 26: Button (rot) und dessen Hoverstate (weiß) nebeneinander

3. Austauschen der Background-Images in `Player/css/editor/buttons.css` für den entsprechenden Button. Listing 1 zeigt das vorgehen. Man muss sich im Klaren darüber sein, dass man hier die originalen Klynt Buttons ersetzt – diese sind anschließend nicht mehr im Editor verfügbar. Stattdessen taucht dort der eigene Button unter der ursprünglichen Klynt-Namensgebung auf.
4. Gegebenenfalls muss das Label des Buttons entfernt werden, je nach dem, ob es in Klynt ursprünglich für den ausgetauschten Button vorgesehen war. In diesem Fall wird das Label versteckt, siehe Listing 1.
5. Der Button kann jetzt in Klynt verwendet werden. Dort wird er im Klynt Editor nicht mehr visuell als Button angezeigt, sondern nur noch durch einen roten Rahmen repräsentiert. Sobald die Sequenz jedoch im Browser aufgerufen wird, ist der Button sichtbar.
6. Im Klynt Editor können dem Button direkt Standardproportionen und Text mitgegeben werden. In diesem Fall wird der Text nicht angezeigt, weil das Label versteckt wurde, allerdings ist die Angabe dennoch von Vorteil, um eine Zuordnung im Klynt-Editor zu gewährleisten.

Es ist nicht zwingend notwendig, die Buttons komplett zu ersetzen. Natürlich können in der entsprechenden CSS Datei unter `Player/css/editor/buttons.css` für einen entsprechenden Button auch nur die visuellen Eigenschaften wie Rahmen, Farbe oder Textanzeige verändert werden. Diese Methode funktioniert analog zum vorgestellten Arbeitsablauf. Auch dort sind die Änderungen anschließend im Button-Editor sichtbar.

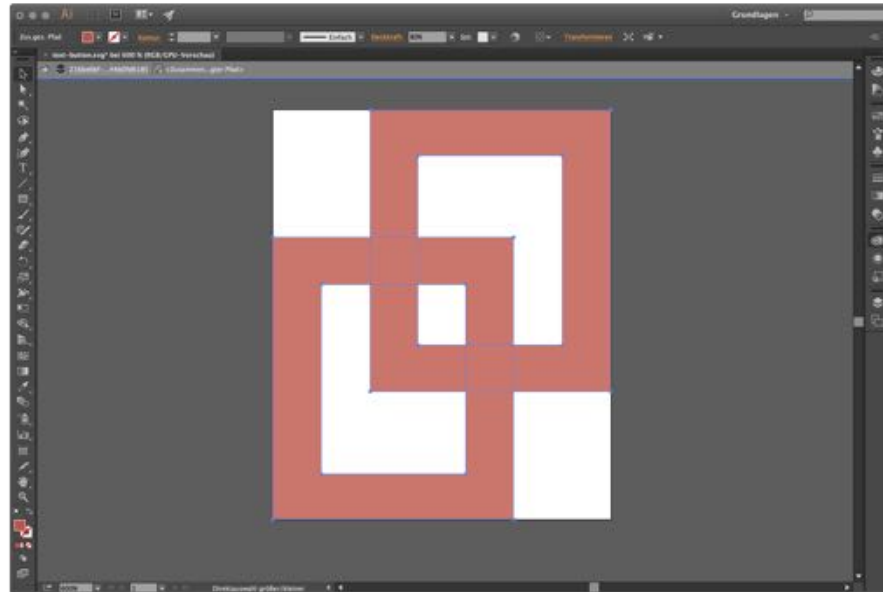


Abbildung 27: Ausgangsbasis für die Erstellung eigener Buttons: Erstellen der Datei in Adobe Illustrator

```

1  /*Ursprüngliches CSS für Button:*/
   .klynt-button-1 {
       color: rgba(255,255,255,1);
       font-family: Open Sans,sans-serif;
5   font-size: 14px;
       font-weight: regular;
       text-align: center;
       text-decoration: none;
9   border: 1px solid rgba(255,255,255,0.5);
       border-radius: 0px;
   }
   .klynt-button-1:hover {
13  background-color: rgba(255,255,255,1);
       color: rgba(0,0,0,1);
   }
   /*Wird ersetzt durch :*/
17  .klynt-button-1 {
       background-image: url("img/text-button.svg");
   }
   .klynt-button-1:hover {
21  background-image: url("img/text-button-hover.svg");
   }
   /*Gegebenenfalls muss das Label entfernt werden:*/
   .klynt-button-1 > .label-container > span {
25  display: none;
   }
   /*Weitere Eigenschaften müssen nicht hinzugefügt werden, weil das SVG
       bereits alle notwendigen Daten enthält.*/

```

Listing 1: CSS-Code für die Einbindung eines eigenen Buttondesigns

4.2.2 Verwendung eigener CSS-Klassen

Klynt erlaubt die Einbindung eigener CSS-Klassen und die anschließende Zuordnung dieser zu Elementen des Klynt Editors. Auf diese Art und Weise können diesen unter anderem Animationen hinzugefügt werden, die Klynt nativ nicht unterstützt. Die Webdocumentary zum 50-jährigen Jubiläum der Universität implementiert beispielsweise einen pulsierenden Button (Abbildung 28, unten, mittig). Für eine bessere Nachvollziehbarkeit und zur Visualisierung des Effekts, wird der Aufruf der Webdocumentary unter `uni.kn/50jahre/webdoku` empfohlen. Der folgende Arbeitsablauf beschreibt die Erstellung einer entsprechenden CSS-Klasse und das Hinzufügen dieser zum Button, der als Bilddatei in Klynt importiert wurde.

1. Erstellen des Buttons als PNG- oder JPG Datei. Leider können hierfür keine SVG Dateien, wie in Unterabschnitt 4.2.1 beschrieben, verwendet werden, weil Klynt dieses Medienformat nicht unterstützt, beziehungsweise im Editor verarbeiten kann.
2. Implementierung einer CSS-Klasse in `Ressources/css/player.css`. In diesem Beispiel wird der pulsierende Effekt durch eine Animation mit Keyframes erzeugt. Zwei Sekunden nach Erscheinen des Buttons startet die Animation, welche den Button im Zeitraum von drei Sekunden um 10% vergrößert und wieder verkleinert. Hierdurch entsteht der pulsierende Effekt. Der entsprechende CSS Code hierfür wird in Listing 2 gezeigt.
3. Die in diesem Fall erstellte Klasse *pulsar* kann nun im Klynt Editor verwendet und einem Element zugewiesen werden. Hierzu wird das Element aufgerufen und im Einstellungsmenü unter dem Menüpunkt *Advanced* die Klasse hinzugefügt. Allerdings ohne vorstehenden Punkt, wie es in der Regel bei der Angabe von CSS Klassen üblich ist. Sollen mehr CSS Klassen hinzugefügt werden, werden diese einfach nur mit Leerzeichen getrennt hintereinander aufgeführt.



Abbildung 28: Pulsierender Button in der Webdocumentary zum 50-jährigen Jubiläum der Universität. Rechteck, unten mittig.

```

1  /*Pulser*/
   @keyframes pulse {
     from {
       -webkit-transform: scale3d(1, 1, 1);
5     transform: scale3d(1, 1, 1);
     }

     50% {
9     -webkit-transform: scale3d(1.1, 1.1, 1.1);
       transform: scale3d(1.1, 1.1, 1.1);
     }

13    to {
       -webkit-transform: scale3d(1.0, 1.0, 1.0);
       transform: scale3d(1.0, 1.0, 1.0);
     }
17  }

.pulsar{
  -webkit-animation: pulse 3s ease-in-out;
21  animation: pulse 3s ease-in-out;
  /*transform-origin: initial;*/
  position: absolute;
  -webkit-animation-iteration-count: infinite;
  animation-iteration-count: infinite;
25  visibility: visible;
  overflow: hidden;
  -webkit-animation-delay: 1s;
29  animation-delay: 1s;
}

```

Listing 2: CSS-Code für einen pulsierenden Button

4.2.3 Verwendung eigener Inhalte durch IFrames

Spätestens bei komplexen Navigationen oder komplett animierten Inhalten stößt Klynt an die Grenzen – es bietet nicht die nötige Funktionalität, um derartige Funktionen zu erstellen. Allerdings liefert Klynt von nativ die Funktion mit, IFrames in Sequenzen zu integrieren. Hierdurch haben insbesondere Entwickler die Möglichkeit, eine Webdocumentary um diverse Funktionalitäten zu erweitern.

Am konkreten Anwendungsfall der Navigationskarte in der Webdocumentary zum 50-jährigen Jubiläum der Universität wird die Entwicklung der Karte, das Hinzufügen von Animationen und die Benutzung der Klynt-API erläutert. Sie steht beispielhaft für eine von vielen Möglichkeiten, IFrames in Klynt zu nutzen, um dessen Funktionalität zu erweitern und einige Inhalte, die durch die alleinige Nutzung Klynts nicht implementierbar wären, einzufügen. Die Angabe von Arbeitsschritten macht den Prozess besser nachvollziehbar und bietet die Möglichkeit, so auch ähnliche Modelle zu reproduzieren.

1. Erstellen der Karte in Adobe Illustrator und anschließender Export im SVG Format. In diesem Fall wurde die Datei von den Kommunikationsdesignern erstellt und in Kooperation von einer LKM-Studentin und einem Informatikstudenten in Bezug auf Namensgebung und der Positionsanpassung einiger Elemente optimiert – die Datei ist in Abbildung 29a zu sehen.
2. Kontrolle der SVG Datei und korrekte Benennung der Elemente. Die Umfasst zum einen die Korrektur verschobener Zeichen, Zum anderen das Zuweisen von Klassennamen zu den entsprechenden Dreiecken und Texten – die Chrome Entwicklertools können hier bei der Zuordnung helfen (Abbildung 29b). Zur Verdeutlichung hierfür zeigt Listing 3 einen Auszug aus dem SVG, in welchem alle Dreiecke korrekt benannt wurden, Abbildung 29c das SVG vor den Korrekturen. Jedes Dreieck ist durch drei Punkte mit jeweils zwei Koordinaten (x,y) definiert. Die Zuweisung aller Dreiecke zur Klasse *.dreiecke*, hilft, später alle Dreiecke auf einmal anzusprechen. Die Zuweisung einer eindeutigen *ID* ist notwendig, um die Dreiecke später auch gezielt bezüglich Animationen und Verlinkungen anzusprechen zu können. Der *Name-Tag* bezeichnet den Namen der Sequenz, wie er in Klynt verwendet wird. Dies wird benötigt, um später über die Klynt-API die korrekte Sequenz anzusteuern.
3. Nachdem die Dreiecke und Texte korrekt benannt wurden, kann mit der Implementation der Animationen und den Verknüpfungen der einzelnen Elemente begonnen werden. Im konkreten Fall dieser Karte wurden folgende Anforderungen gestellt:

- Beim Hovern über Text oder Dreiecke sollen sich die entsprechenden Elemente blau einfärben und vergrößern.
- Beim Klick auf eines der Elemente, soll der Klynt Player zur ausgewählten Sequenz wechseln.

Umgesetzt wurden diese Anforderungen durch Javascript und CSS. Eine Funktion iteriert über alle Dreiecke und weist diesen *EventListener* für *mouseover*, *click* und *mouseout* zu. Die *EventListener* „mouseover“ und „mouseout“ rufen selbst wiederum Funktionen auf, welche den ausgewählten Elementen bestimmte CSS Eigenschaften zuweisen, oder diese wieder entfernen. „Click“ ruft eine Funktion der Klynt API auf, schließt das Overlay und wechselt zu der in den Funktionsparametern mitgegebenen Sequenz. Listing 4 zeigt den entsprechenden Javascriptcode anhand der Dreiecke, er lässt sich jedoch analog für die Texte reproduzieren. Der CSS Code, der den Elementen beim „mouseover“ zugewiesen wird, basiert auf dem gleichen Effekt wie der pulsierende Button aus Abschnitt Unterabschnitt 4.2.2, lediglich der Text wird nicht vergrößert, sondern verändert seine Stärke zu *bold*.

4. Nachdem alle notwendigen Funktionalitäten implementiert wurden, kann die Karte einer Sequenz in Klynt hinzugefügt werden. Hierzu wird in der Sequenz ein *IFrame* angelegt, in welchem auf den entsprechenden Code referenziert wird, in diesem Beispiel:

```
<iframe src='Ressources/navi.html' width='1920' height='1080'></iframe>
```

Die Karte ist nun in Klynt verfügbar und kann beispielsweise durch die Verlinkung mit einem Button als Overlay aufgerufen werden.

Natürlich ist dies nur ein möglicher Anwendungsfall. Er zeigt allerdings, dass *IFrames* in Klynt, insbesondere in Kombination mit der Klynt-API ein mächtiges Werkzeug sind. Die Klynt-API implementiert neben dem Wechseln zu anderen Sequenzen unter anderem auch das Schließen von Overlays oder Starten und Stoppen des Klynt Players. Eine vollständige Dokumentation der Klynt API findet sich im Github-Repository des Klyntplayers⁶. Nichtsdestotrotz weist dies auf einen deutlichen Nachteil hin: Ohne Entwickler im Team ist eine Implementation solch komplexer Funktionalitäten nicht möglich. Es ist verständlich, dass Klynt nicht alle Bereiche abdecken kann, doch gerade im Bereich der Navigation wäre es wünschenswert, zumindest unter einigen, anpassbaren Vorlagen auswählen zu können.

⁶ <https://github.com/Klynt/Klynt-Player/wiki/iFrameAPI>
– zuletzt aufgerufen am 22.06.2016


```

1 <g id="Dreiecke">
  <polygon class="dreiecke" id="zoo" name="Zoo" points="608.6,365
    623.4,390.7 593.8,390.7"/>
  <polygon class="dreiecke" id="hockgraben" name="Hockgraben" points="
    777.4,960.5 792.2,986.1 762.6,986.1"/>
  ...
5 <tspan x="0" y="0" class="st9 st2"> - Tour 1: Tiere und Pflanzen</tspan>
  <tspan x="0" y="32.7" id="bienen-text" class="text">Bienenstoecke</
    tspan>
  <tspan x="169" y="32.7" id="zoo-text" class="text">Zoologische Sammlung
    </tspan>
9 ...

```

Listing 3: Codeauszug aus der HTML Datei für die Entwicklung der Navigationskarte in der Webdocumentary zum 50-jährigen Jubiläum der Universität

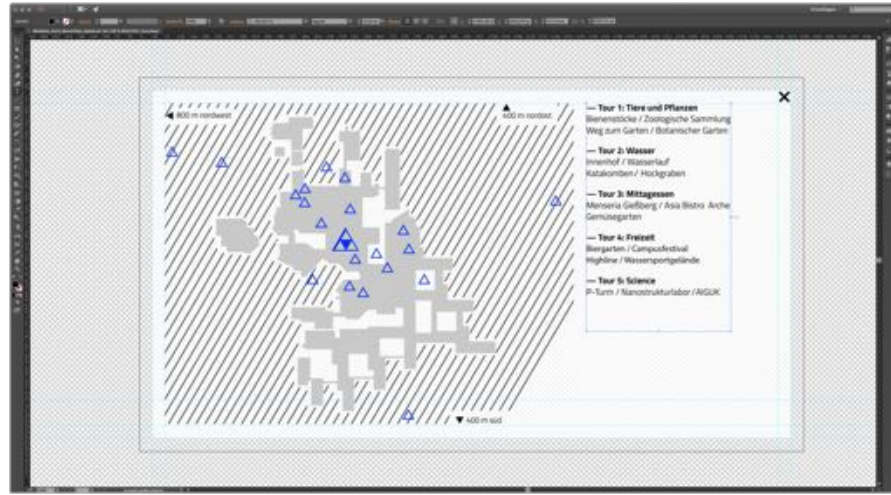
```

// Alle Dreiecke in ein Array, alle Texte der Touren in ein Array
var alltriangles = document.getElementsByClassName("dreiecke");
3 var foyerIcon = document.getElementsByClassName("st14");
var alltours = document.getElementsByClassName("text");

// Allen Elementen Eventlistener hinzufügen
7 for (var i = alltriangles.length - 1; i >= 0; i--) {
  alltriangles[i].addEventListener('mouseover', mouseOver);
  alltriangles[i].addEventListener('mouseout', mouseOut);
  alltriangles[i].addEventListener('click', clicker);
11 }
// Beim Mouseover werden Dreiecke und Texte blau
function mouseOver(){
  triangle = this.id + '-text';
15 this.classList.add("filler");
  document.getElementById(triangle).classList.add("filler");
}
// CSS Klasse bei Mouseout wieder entfernen
19 function mouseOut(){
  triangle = this.id + '-text';
  // console.log(triangle);
  this.classList.remove("filler");
23 document.getElementById(triangle).classList.remove("filler");
}
// Bei Klick Sequenz wechseln
function clicker(){
27 target = this.getAttribute("name");
  console.log(target);
  // console.log(this.name.innerHTML);
  KlyntAPI.commands.openSequence({
31   targetSequence: target,
   transition: 'fade'});
}

```

Listing 4: Codeauszug: Javascript für die Entwicklung der Navigationskarte in der Webdocumentary zum 50-jährigen Jubiläum der Universität



(a)



(b)



(c)

Abbildung 29: Abbildungen verschiedener Arbeitsschritte für die Entwicklung der Navigationskarte in der Webdocumentary zum 50-jährigen Jubiläum der Universität

4.2.4 Skalierung des Klynt-Players

Heute ist es besonders wichtig, Inhalte im Web plattformunabhängig auf den unterschiedlichsten Geräten anzeigen zu können. Aktuelle Statistiken belegen, dass bereits 2016 mehr als 50% der deutschen Bevölkerung mobile Geräte nutzen, um im Internet zu surfen (Abbildung 30). Eine andere Statistik, zu sehen in Abbildung 31, prognostiziert, dass im Jahr 2019 rund 56 Millionen Deutsche mobil auf das Internet zugreifen werden. Dies bekräftigt, dass auch die Web-documentary plattformunabhängig und von den unterschiedlichsten Gerätegrößen aus aufrufbar sein sollte.

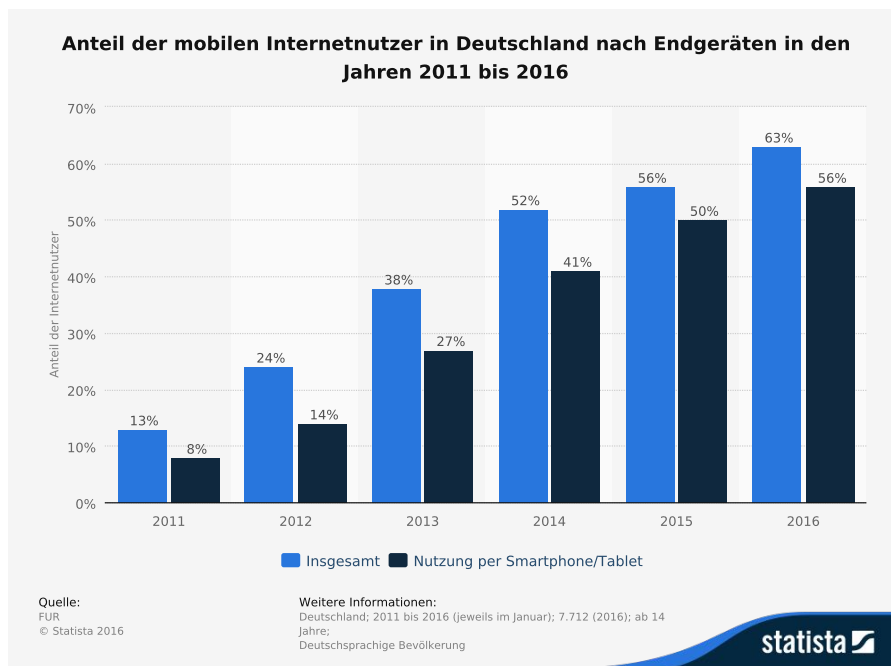


Abbildung 30: Verband Internet Reisevertrieb. (n.d.). Anteil der mobilen Internetnutzer in Deutschland nach Endgeräten in den Jahren 2011 bis 2016. In Statista – Das Statistik-Portal⁷.

Obwohl Klynt einen *Responsive Mode* (aktuell noch in der Betaphase befindlich) zur Darstellung der Webdocumentary auf mobilen Endgeräten anbietet, ist die Arbeit mit diesem kaum möglich. Die Ergebnisse im Browser sind nicht konsistent, oftmals behalten Elemente ihr Proportionen nicht bei, oder Hotspots, die Bildelemente klickbar machen sollen, verschieben sich. Unter Verwendung des *Responsive Modes* in Klynt müssen für jedes Element und jede Gerätegröße (Desktop, Tablet, Smartphone) die Eigenschaften separat gesetzt werden.

⁷ <http://de.statista.com/statistik/daten/studie/181973/umfrage/genutzte-mobilgeraete-fuer-mobilen-internetzugang-in-deutschland/>
– zuletzt aufgerufen am 27.06.2016

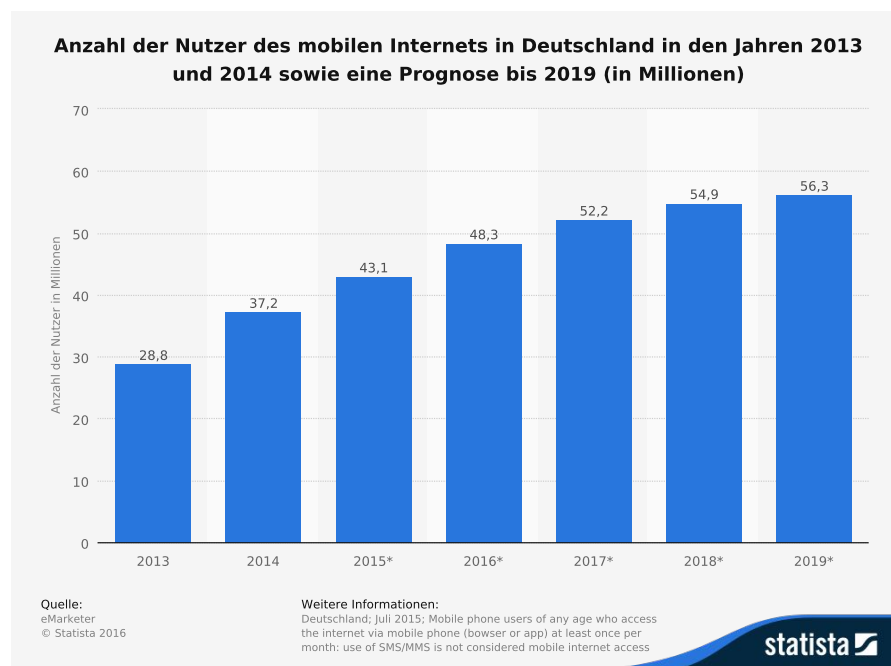


Abbildung 31: AP. (n.d.). Anzahl der Nutzer des mobilen Internets in Deutschland in den Jahren 2013 und 2014 sowie eine Prognose bis 2019 (in Millionen). In Statista - Das Statistik-Portal⁸.

Dies umfasst unter anderem Positionierung und Skalierung der Elemente und nimmt viel Zeit in Anspruch, was die Arbeit in diesem Modus enorm erschwert. Aus diesem Grund wurde ein Lösungsansatz entwickelt, der auf Basis der *CSS Scale Property* die Webdocumentary entsprechend der Gerätegröße auf die gesamte Breite des Viewports skaliert. Natürlich ist diese Lösung nur ein Workaround, der nicht einbezieht, dass Buttons oder Texte auf Mobilgeräten gegebenenfalls eine andere Größe aufweisen müssen. Allerdings zeigten Tests bei der Webdocumentary zum 50-jährigen Jubiläum der Universität, dass diese zumindest auf Tablets mit etwa 10 Zoll Displaygröße noch benutzbar ist. Gegebenenfalls kann im Browser überprüft werden, mit welchem Endgerät der Benutzer die Webdocumentary aufruft. Sollte dieser ein Smartphone nutzen, kann man ihn darauf hinweisen, dass die Webdocumentary nicht für diese Form der Darstellung ausgelegt ist und empfehlen, die Seite nochmals mit einem Tablet oder Desktop-PC zu besuchen.

Die technische Umsetzung für die Skalierung basiert auf Javascript. Beim initialen Aufrufen der Webdocumentary und beim Skalieren des Browserfensters wird ein Multiplikator aus aktueller Bildschirmbreite und Webdocumentarybreite errechnet, der notwendig ist, um die Webdocumentary an die Bildschirmbreite anzupassen. Anschlie-

⁸ <http://de.statista.com/statistik/daten/studie/180578/umfrage/anzahl-der-nutzer-des-mobilen-internets-in-deutschland-seit-2005/>
– zuletzt aufgerufen am 27.06.2016

ßend wird dieser Multiplikator mit der *Scale Property* auf den Container, in dem die Webdocumentary abläuft, angewendet. Allerdings skaliert die Webdocumentary nicht über deren originalen Breite von 1920 Pixeln – sie kann jedoch über den *Fullscreenbutton* in der unteren Menüleiste im Vollbildmodus abgespielt werden. Listing 5 zeigt die Umsetzung mit Javascript. Die Javascript Datei wird in das gleiche Verzeichnis wie die *index.html* Datei der Webdocumentary gelegt und in dieser entsprechend verlinkt. Anschließend ist die Skalierung wirksam.

```

function sizePlayer(){
  // Aktuelle Fensterbreite und Multiplikator errechnen
3  var windowWidth = $(window).width();
  var multiplier = windowWidth / 1920;

  // Ist der Multiplaktor kleiner als 1, also der Viewport kleiner als
  // 1920px,
7  // wird die Scale Property angewendet
  if (multiplier <= 1) {
    playerContainer = $('#player');
    playerContainer.css('transform-origin', 'top left');
11  playerContainer.css('transform', 'scale(' + multiplier + ')');
    playerContainer.css('bottom', 'auto');
  } else {
    playerContainer.css('transform', 'scale(1)');
15  }

  var windowHeight = $(window).height();
  var playerHeight = playerContainer.height()*multiplier;
19

  // Anpassungen an der Positionierung des Players, der Mittig im Browser
  // liegen soll
  if (playerHeight > 0) {
23  var marginTop = (windowHeight - playerHeight)/2;
    playerContainer.css('margin-top', marginTop+"px");
  } else {
    window.setTimeout(sizePlayer, 100);
27  }
  }

$(window).resize(sizePlayer);
31 $(window).load(sizePlayer);

```

Listing 5: Javascript Code für die Skalierung des Klynt Plyers

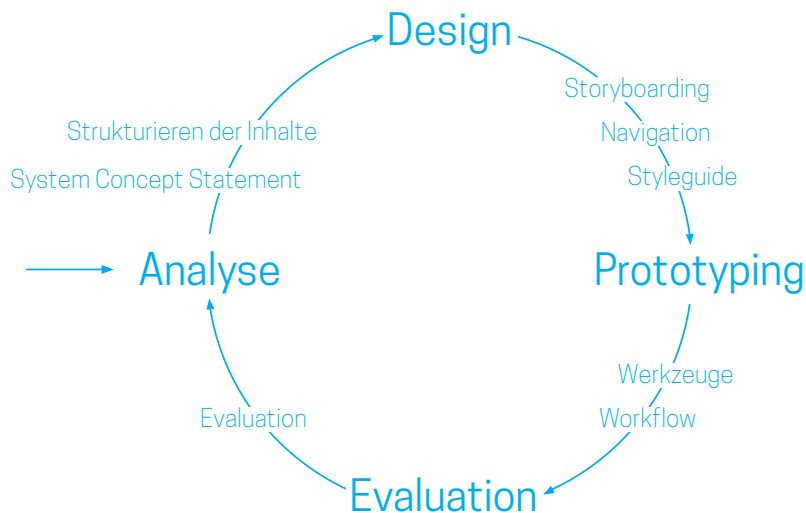


Abbildung 32: Konzeptionelle Umsetzung anhand des UX Lifecyclemodells zur Extraktion der Leitlinien

Diese Arbeit hat zum einen konzeptionelle Leitlinien für das Design und die Produktion von Webdocumentaries vorgestellt, zum anderen spezifische, technische Erweiterungsmöglichkeiten und Lösungsansätze für Probleme, die bei der Entwicklung und der späteren Veröffentlichung auftreten können. Die konzeptionellen Leitlinien wurden entlang des UX-Lifecycles nach Hartson und Pyla [8] erstellt, konnten aber nicht immer ohne weiteres übernommen werden, weil das Format der Webdocumentary vergleichsweise neu ist und kaum Informationen oder Instruktionen zur Erstellung existieren. Aus diesem Grund wurden einige Schritte des UX-Lifecyclemodells auf Basis von Erfahrungswerten aus der Entwicklung zweier Webdocumentaries angepasst. Abbildung 32 visualisiert noch einmal den UX-Lifecycle und zeigt an, welche Schritte darin aufgenommen worden sind. Die Erfahrungswerte, auf welchen die Schritte angepasst und die Leitlinien extrahiert wurden, beruhen sowohl auf positiven, als auch auf negativen Erfahrungen und reflektieren so eine realistische Basis für die Umsetzung zukünftiger Projekte. Dies zeigt aber auch deutlich, wie sehr sich die Entwicklung einer Webdocumentary von der einer Website oder Softwareanwendung unterscheidet. Zu beiden finden sich im Internet und in Fachbüchern Instruktionen, Guidelines und Best-Practices für die Umsetzung. Weitere Untersuchungen im Bereich der

Entwicklung einer Webdocumentary sind deshalb erforderlich, um die vorgestellten Leitlinien in zukünftigen Projekten zu überprüfen und um sinnvolle Punkte zu erweitern.

Die Entwicklung der Webdocumentary als iPad-Anwendung für die Ausstellung *Tell Genderes - 20 Meter Menschheitsgeschichte* war das erste Projekt dieser Art, welches von Studenten der Universität Konstanz umgesetzt wurde. Zu diesem Zeitpunkt stand zunächst die Idee im Raum, die Inhalte der Ausstellung im dritten Obergeschoss des Bildungsturms durch einen zusätzlichen Informationskanal in Form einer Webdocumentary zu erweitern. Auf Basis der Seminararbeit von Oliver Barwig wurde ein Werkzeug für die Erstellung der Webdocumentary festgelegt, anschließend die Idee, für die Entwicklung nach dem UX-Lifecyclemodell zu arbeiten. Dieser Ansatz erwies sich als tragfähig, weil viele Prozesse auf die Entwicklung der Webdocumentary angewendet werden konnten, unter anderem die Idee des Storyboardings und Prototypings. Allerdings stellte sich auch heraus, dass einige der Schritte speziell auf die Webdocumentary angepasst werden müssen, weil diese im UX-Lifecycle nicht existieren oder nicht direkt übertragbar sind. So wurde beispielsweise die Erstellung eines Inhaltsverzeichnisses eingefügt, welches die einzelnen Sequenzen der Webdocumentary aufführt, deren Inhalte benennt und so das Storyboarding und die spätere Implementation vereinfacht. Weiterhin wurden bei diesem Projekt wertvolle Erfahrungen mit der Software „Klynt“ gesammelt, welche im Projektbericht von Oliver Barwig [2] festgehalten wurden und die Grundlage für die Entwicklung der Webdocumentary zum 50-jährigen Jubiläum der Universität bildeten. Dieses Projekt war weitaus größer angelegt als die iPad-Anwendung im Bildungsturm und hatte zudem den Anspruch, die am Projekt beteiligten LKM-Studierenden zu befähigen, große Teile der Webdocumentary mit Hilfe von Klynt selbst umzusetzen. Aus diesem Grund wurden Workshops für die Entwicklung einer Webdocumentary mit Klynt gegeben und regelmäßige Treffen eingeführt, bei welchen sich die Projektbeteiligten austauschten und den aktuellen Stand ihrer Arbeit vorstellten. Auch in diesem Projekt wurde der UX-Lifecycle angewendet, mit den Erfahrungswerten aus der Entwicklung der ersten Webdocumentary erweitert und auf die Entwicklung der Webdocumentary zum 50-jährigen Jubiläum der Universität angepasst. Unter anderem wurde hier der bereits angesprochene Workflow eingeführt, um Aufgaben besser verteilen zu können und einen strukturierten Ablauf zu ermöglichen.

Die Umsetzung der Webdocumentary zum 50-jährigen Jubiläum zeigte aber auch, dass trotz eines strukturierten Workflows, viel Kommunikation, der Zuteilung von Aufgaben und regelmäßigen Treffen nicht immer alles nach Plan läuft. So wurden zeitlich gesetzte Zie-

le selten eingehalten, weil Diskrepanzen bezüglich Produktion des Filmmaterials und der Implementierung des ersten Prototypen auftraten – kurzum: Das Filmmaterial war noch nicht fertig, obwohl der erste Prototyp es bereits sein sollte. Dies legt nahe, bei zukünftigen Webdocumentary-Projekten eine Person zu ernennen, die nur für das Projektmanagement zuständig ist. Sie sollte den zeitlichen Rahmen im Visier haben, Aufgaben zuteilen und das Projekt auf allen Ebenen koordinieren. Es bedarf weiterer, zukünftiger Untersuchungen um herauszufinden, wie das Projektmanagement auf die Entwicklung einer Webdocumentary zugeschnitten sein sollte. Es wäre jedoch sinnvoll, dann eine Leitlinie bezüglich des Projektmanagements mit in die vorgestellte Sammlung aufzunehmen.

Ein weiterer Punkt, der in Zukunft genauer untersucht werden sollte, ist die Evaluation einer Webdocumentary. Wie bereits im entsprechenden Abschnitt erwähnt, liegen hierzu keine spezifischen, auf die Webdocumentary angepassten Informationen vor. Nichtsdestotrotz ist es notwendig, vor der Veröffentlichung herauszufinden, wie oder ob ein Nutzer die Anwendung bedienen kann und ob sie die Anforderungen an Benutzerfreundlichkeit und Gebrauchstauglichkeit erfüllt. Weil eine Webdocumentary in manchen Punkten einer Website ähnlich ist, könnten hier die Heuristiken der Nielson Norman Group¹ als Grundlage dienen und entsprechend auf die Webdocumentary angepasst werden. Eine Evaluation könnte beispielsweise auch in unterschiedlichen Stadien der Entwicklung stattfinden, wie Hartson und Pyla im UX-Book vorschlagen [8]. Wie diese Evaluation in den entsprechenden Phasen aussehen könnte, muss jedoch genauer untersucht werden, beispielsweise könnte das RITE-Modell auf dem High-Fidelity-Prototyp angewendet werden. Nach Hartson und Pyla umfasst dies ein schnelles, kollaboratives Testen bezüglich der Benutzererfahrung der Anwendung mit Teammitgliedern und Testpersonen, bei welchem gefundene Fehler direkt verbessert werden [8]. Das Modell basiert ursprünglich auf einer von Medlock et al. [11] entwickelten Evaluationsmethode.

Ein anderer Aspekt, der die Entwicklung der Webdocumentary oftmals erschwerte oder Probleme verursachte, war die Software *Klynt*. Obwohl diese momentan ein nahezu alternativloses Werkzeug zur Erstellung von Webdocumentaries ist, fehlen dieser einige Funktionalitäten, die im Projekt notwendig gewesen wären. An dieser Stelle werden die fehlenden Funktionalitäten aufgeführt und gemeinsam mit Verbesserungsvorschlägen präsentiert:

- **PROJEKT-MERGING:** Klynt kann Projekte nicht zusammenfügen, was dazu führt, dass mehrere Parteien auf einer Version der

¹ <https://www.nngroup.com/articles/ten-usability-heuristics/> – zuletzt aufgerufen am 26.06.2016

Webdocumentary arbeiten müssen. Diese wird nach abgeschlossener Arbeit auf einen Fileserver oder USB-Stick gelegt und dann an die andere Person weitergegeben. Angesichts der Dateigröße eines Webdocumentary-Projekts ist der Zeitaufwand beim Kopieren der Daten meist sehr hoch. Eine mögliche Verbesserung hierfür wäre die Integration eines Versionsverwaltungstools wie *GitHub*² oder *Subversion*³, wie es in der Softwareentwicklung bereits lange verwendet wird.

- **DESIGN-KITS:** Klynt bietet lediglich eine kleine Auswahl vordefinierter Buttons und anderer Interaktionselemente. Eigene Designs müssen aufwändig auf Codebasis hinzugefügt werden. Eine Alternative bietet der Import von Bilddateien, die jedoch oftmals eine schlechte Affordance besitzen und nicht hinreichend mit *Click-* oder *Tapaktionen* belegbar sind. Eine mögliche Verbesserung wäre die Integration einer Auswahlmöglichkeit für verschiedene Designs, die heruntergeladen werden können oder direkt mitgeliefert werden.
- **RESPONSIVENESS:** Wie bereits in Unterabschnitt 4.2.4 erwähnt, ist der *Responsive Mode* von Klynt schwer benutzbar. Aus diesem Grund wurde auch die Skalierungsfunktion für den Player implementiert, weil es heute unabdingbar ist, Webanwendungen nicht nur auf Desktops, sondern auch auf Smartphones und Tablets anzeigen zu können. Eine mögliche Verbesserung wäre die Überarbeitung des *Responsive Modes*, der anschließend die Möglichkeit bietet, Gesten für mobile Geräte zu nutzen und selbstständig skaliert, ohne dass für jedes Element spezifische Einstellungen gesetzt werden müssen.
- **PERFORMANCE:** Aus Klynt exportierte Projekte können sehr groß werden, wenn viele Videos enthalten sind. Dies kann zu langen Ladezeiten führen. Eine mögliche Verbesserung hierfür wäre eine native Klynt-Funktion, welche Videos – abhängig von der Internetgeschwindigkeit des Nutzers – komprimiert und diesem so eine für ihn ohne große Pufferzeiten nutzbare Version zur Verfügung stellt. Unter anderem verwendet die Videoplattform *youtube*⁴ diesen Ansatz. Überdies hinaus exportiert Klynt Projekte, ohne den HTML-, CSS- und Javascriptcode zu komprimieren, beziehungsweise zu minifizieren, wie es heute für Webanwendungen üblich ist. Die Einbindung einer Code-Minifizierungsfunktion könnte das Problem lösen und so zu schnelleren Ladezeiten und einer besseren Performance der Webdocumentary im Browser führen.

2 <https://github.com/> – zuletzt aufgerufen am 26.06.2016

3 <https://subversion.apache.org/> – zuletzt aufgerufen am 26.06.2016

4 <https://www.youtube.com/> – zuletzt aufgerufen am 27.06.2016

Wie diese Arbeit gezeigt hat, existieren keine Patentrezepte für die Erstellung einer Webdocumentary. Dies liegt zum einen an dem modernen Medium selbst, zu dem bisher wenig wissenschaftliche Informationen vorliegen, zum anderen an der Komplexität, welche durch die Verschmelzung verschiedener Medien unter Verwendung moderner Webtechnologien herbeigeführt wird. Durch die Umsetzung der beiden Webdocumentaries hat sich jedoch gezeigt, dass sich viele bestehende Methoden aus Web- und Softwareentwicklung übertragen und auf die Entwicklung einer Webdocumentary anpassen lassen. Zusammen mit dem Potential, welches die Webdocumentary als neue und moderne Präsentationsform bietet, entsteht so die Motivation, weitere Untersuchungen im Bereich der Webdocumentary und deren Entwicklung anzustellen und die vorgestellten Leitlinien zu ergänzen.

Aus diesem Grund wird angestrebt, gemeinsam mit einigen Mitwirkenden beider Projekte, weitere Leitlinien bezüglich Storytelling und Film- und Audioproduktion zu generieren und diese anschließend in einem e-Book zu veröffentlichen. Dies soll Interessierten und motivierten Personen ermöglichen, selbst vom Format der Webdocumentary Gebrauch zu machen und sie dabei unterstützen, *Ihre* Geschichte zu erzählen.

LITERATUR

- [1] Oliver Barwig. „Webdocumentary – Mediale Eigenschaften, Technische Hintergründe und Autorenwerkzeuge“. Seminararbeit. 2015.
- [2] Oliver Barwig. „Webdocumentaries mit Klynt – Eine Toolbox mit Anleitungen, Praxisbeispielen und Erweiterungen im Kontext einer multimedialen Ausstellung“. Projektbericht. 2016.
- [3] Judith Aston und Sandra Gaudenzi. „Interactive documentary: setting the field“. In: *Studies in Documentary Film* 6.2 (2012), S. 125–139. DOI: doi:10.1386/sdf.6.2.125_1.
- [4] Christian Eichinger und Michael Schrefl. „Navigation Consistency in Web Site Families“. In: *Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services. iiWAS '09*. Kuala Lumpur, Malaysia: ACM, 2009, S. 121–129. DOI: 10.1145/1806338.1806367.
- [5] Dayna Galloway, Kenneth B. McAlpine und Paul Harris. „From Michael Moore to JFK Reloaded: Towards a working model of interactive documentary“. In: *Journal of Media Practice* 8.3 (2007), S. 325–339. DOI: 10.1386/jmpr.8.3.325_1.
- [6] Jeff Gothelf und Josh Seiden. *Lean UX: Applying Lean Principles to Improve User Experience*. Sebastopol: O'Reilly Media, Incorporated, 2013.
- [7] Saul Greenberg, Sheelagh Carpendale, Nicolai Marquardt und Bill Buxton. *Sketching User Experiences: The Workbook*. San Francisco: Morgan Kaufmann, 2011.
- [8] Rex Hartson und Pardha Pyla. *The UX Book: Process and Guidelines for Ensuring a Quality User Experience*. San Francisco: Morgan Kaufmann, 2012.
- [9] Paul Issing Ludwig J. und Klimsa. *Information und Lernen mit Multimedia und Internet: Lehrbuch für Studium und Praxis*. 3. Aufl. Weinheim: Beltz, 2002.
- [10] Daniel Kehlmann. *Ruhm: ein Roman in neun Geschichten*. Reinbek: Rowohlt, 2009.
- [11] MC Medlock, D Wixon, M McGee und D Welsh. „The rapid iterative test and evaluation method: Better products in less time“. In: *Costjustifying usability: An update for the Internet age* (2005), S. 489–517.
- [12] Rolf Molich und Jakob Nielsen. „Improving a Human-computer Dialogue“. In: *Commun. ACM* 33 (1990), S. 338–348. DOI: 10.1145/77481.77486.

- [13] Kate Nash. „Modes of interactivity: analysing the webdoc“. In: *Media, Culture & Society* 34.2 (2012), S. 195–210. DOI: 10.1177/0163443711430758.
- [14] Jakob Nielsen. „User Interface Design for the WWW“. In: *CHI '97 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '97. Atlanta, Georgia: ACM, 1997, S. 140–141. DOI: 10.1145/1120212.1120312.
- [15] Sharon Oviatt. „Ten Myths of Multimodal Interaction“. In: *Commun. ACM* 42.11 (1999), S. 74–81. DOI: 10.1145/319382.319398.
- [16] Sheizaf Rafaeli. „Interactivity: From new media to communication“. In: *Sage annual review of communication research: Advancing communication science* 16 (1988), S. 110–134.
- [17] Robert W. Root und Kathy M. Uyeda. „A Headsup on GUI Styleguides: Report on the CHI'92 Styleguide SIG“. In: *SIGCHI Bull.* 25 (Juli 1993), S. 32–35. DOI: 10.1145/155786.155792.

APPENDIX – DIGITALE ANLAGEN

Dieser Arbeit ist ein USB-Stick mit den folgenden Dateien angehängt:

- Diese Arbeit als:
„Bachelorthesis_OliverBarwig.pdf“
- Die dieser Arbeit vorangegangene Seminararbeit als:
„Seminararbeit_OliverBarwig.pdf“
- Der dieser Arbeit vorangegangene Projektbericht als:
„Projektbericht_OliverBarwig.pdf“
- Der Styleguide für die Webdocumentary zum 50-jährigen Jubiläum der Universität als:
„Styleguide_Webdocumentary_UniversitaetKonstanz_2016.pdf“
- Die Corporate-Design-Dokumentation der Ausstellung im Bildungsturm als:
„Tell_Genderes_CDD.pdf“
- Das Screendesign der iPad-Anwendung im Bildungsturm als:
„Tablet_Application_Screendesign.pdf“

b

APPENDIX

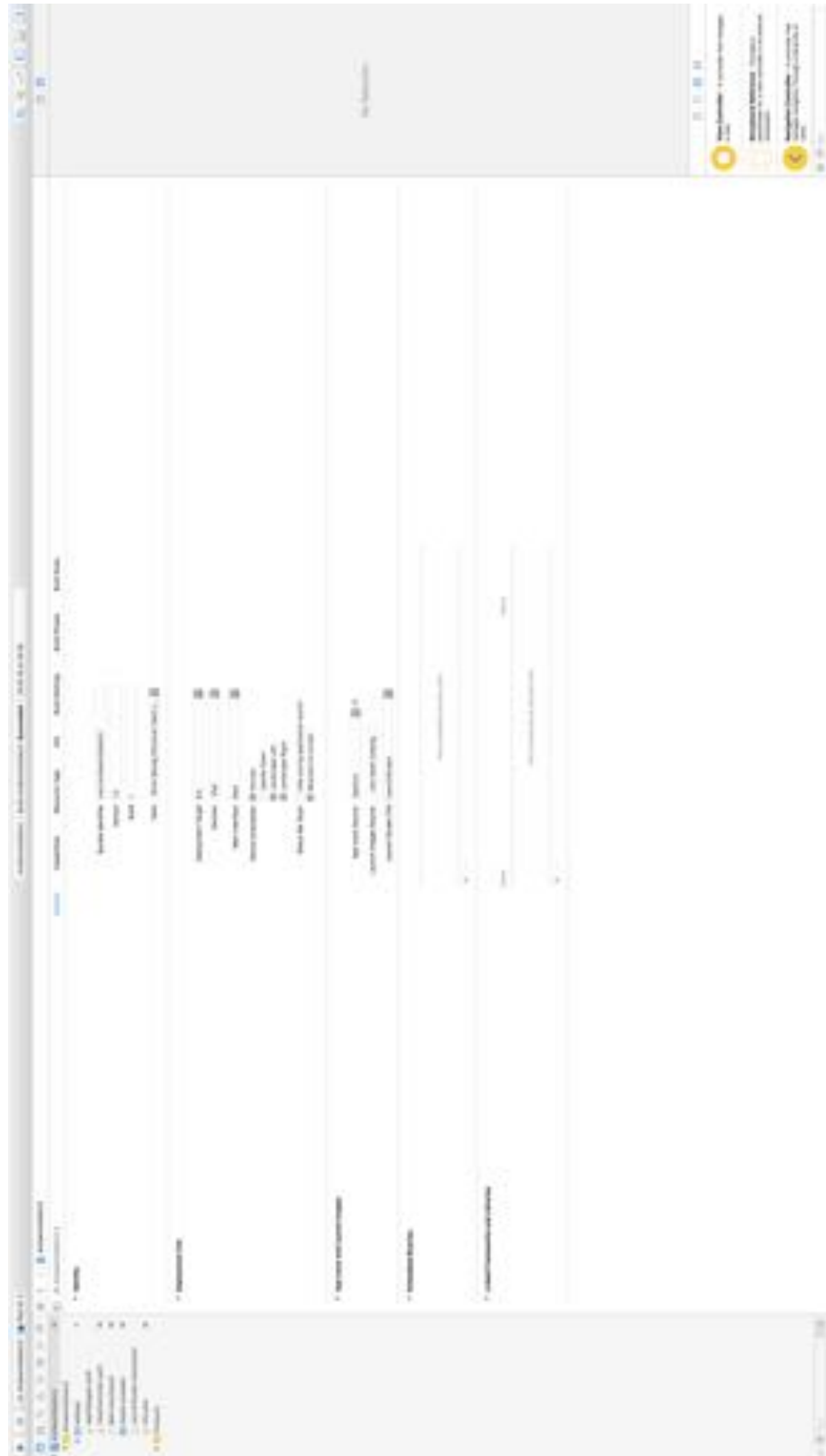


Abbildung 33: Xcode – Übersicht Hauptmenü

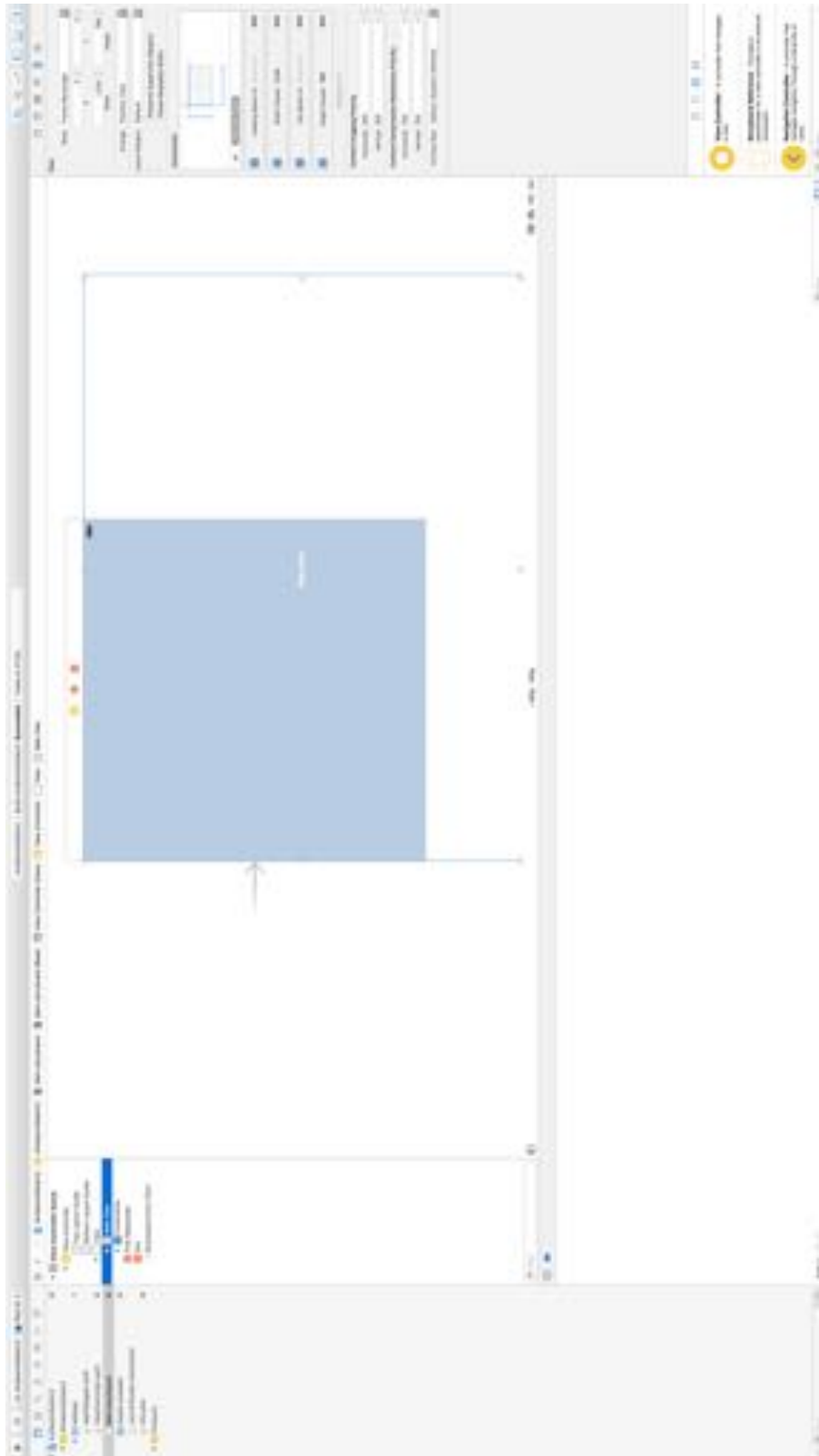


Abbildung 34: Xcode – Übersicht Main Storyboard

```

1 //
  // ViewController.swift
  // AntikenInGefahr2
  //
5 // Created by Oliver Barwig on 22.12.15.
  //

import UIKit
9 import WebKit

class ViewController: UIViewController {

13   @IBOutlet var webView: UIWebView!

  override func viewDidLoad() {
    super.viewDidLoad()

17    // Definieren, wo die index.html des Projekts liegt
    // index.html ist immer der Startpunkt jedes Projekts
    let html = NSBundle.mainBundle().pathForResource("index.html",
21    ofType: "", inDirectory: "webapp")

    // URL-Objekt aus dem oben genannten Pfad generieren
    let url = NSURL.fileURLWithPath(html!)

25    // Request-Objekt aus URL-Objekt, das der WebView übergeben wird
    let request = NSURLRequest(URL: url)

29    // Übergabe des URL-Objekts an die WebView
    self.webView.loadRequest(request)

  }

33  override func didReceiveMemoryWarning() {
    super.didReceiveMemoryWarning()
  }

37  override func prefersStatusBarHidden() -> Bool {
    return true;
  }

41 }

```

Listing 6: Code mit Kommentaren aus der ViewController.swift Datei der iPad-Anwendung

```

//
2 // AppDelegate.swift
// ZumutungenDesHoerens
//
// Created by Oliver Barwig on 25.05.16.
6 //

import UIKit
import CoreData
10

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

14     var window: UIWindow?

    func application(application: UIApplication,
didFinishLaunchingWithOptions launchOptions: [NSObject: AnyObject]?)
-> Bool {
    // ...
18     return true
}

    func applicationWillResignActive(application: UIApplication) {
22     // ...
}

    func applicationDidEnterBackground(application: UIApplication) {
26     // Use this method to release shared resources, save user data,
invalidate timers, and store enough application state information to
restore your application to its current state in case it is
terminated later.
// If your application supports background execution, this method
is called instead of applicationWillTerminate: when the user quits.

// Untenstehende Zeile einfügen. Sie terminiert die App beim
Klick auf den Homebuuton.
30     exit(0);
}

//...
34 }
}

```

Listing 7: Code mit Kommentaren für AppDelegate.swift Datei