



Dokumentation der Hausarbeit

Internetgrundlagen

an der

NORDAKADEMIE

Fachhochschule der Wirtschaft



Thema: Website „Bahamas“

Verfasser: Oliver Plehn (Matrikelnummer: 4361)
Steenwisch 8
22527 Hamburg

Kai Hufenbach (Matrikelnummer: 4355)
Kirchenweg 9c
22885 Barsbüttel

Abgabedatum: 06-09-2011

Erklärung der Selbstständigkeit

Hiermit versichern wir, die vorliegende Arbeit selbstständig verfasst
und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt
sowie die Zitate deutlich kenntlich gemacht zu haben.

Hamburg, den

Inhaltsverzeichnis

Erklärung der Selbstständigkeit.....	2
Abbildungsverzeichnis	5
Abkürzungsverzeichnis	5
1 Einleitung	6
1.1 Aufgabenstellung.....	6
1.2 Unsere Zielsetzung	6
2 Erstellungsprozess.....	6
2.1 Erste Schritte und Einrichten der Entwicklungsumgebung.....	6
2.1.1 Proportionskonzept.....	7
2.1.2 Farbkonzept.....	8
2.1.3 Schriftbild	10
2.1.4 Details zur Designentwicklung	10
2.1.5 Iterative Implementierungsphase.....	11
2.1.6 Abschließende Qualitätssicherung	12
3 Vorgehensweise	12
4 Aufteilung der Arbeit	13
5 Probleme	13
5.1 „Flickering Bug“ im IE8	14
5.1.1 Beschreibung	14
5.1.2 Lösungsansatz.....	14
5.2 IE8: Hover auf unsichtbare Boxen.....	14
5.2.1 Beschreibung	14
5.2.2 Lösungsansatz.....	14
5.3 Schachtelungsfehler durch XSLT im IE.....	14
5.3.1 Beschreibung	14
5.3.2 Lösungsansatz.....	15
5.4 Kein CSS ohne Angabe von Namespace in Javascript im FF	15
5.4.1 Beschreibung	15
5.4.2 Lösungsansatz.....	15
5.5 Umlautprobleme trotz UTF-8 im IE.....	15
5.5.1 Beschreibung	15
5.5.2 Lösungsansatz.....	15
5.6 XSLT „apply templates + parameter“	16
5.6.1 Beschreibung	16
5.6.2 Lösungsansatz.....	16

6	Abweichungen vom Grobkonzept.....	17
7	Aufbau des Webauftritt	18
7.1	Inhalt.....	18
8	Besonderheiten des technischen Rahmens	19
8.1	Namenskonventionen.....	20
8.2	Navigationsstruktur.....	20
8.2.1	Stickys	21
8.2.2	Blättern	21
8.3	Buchverhalten	22
8.4	Typisierung der Seiten	22
9	Besondere technische Aspekte einzelner Kapitel	23
9.1	Steckbrief	23
9.2	Quiz	25
10	Validität.....	26
11	Verzeichnisstruktur	26
12	Fazit.....	27
	Quellenverzeichnis	28
	Anhang 1	

Abbildungsverzeichnis

Abbildung 1: Plan mit Pixelangaben	7
Abbildung 2: Verwendung des Goldenen Schnitts beim Entwurf des Titelbilds	8
Abbildung 3: Verwendung des Goldenen Schnitts beim Layout des Fließtextes und der Ledermappe.....	8
Abbildung 4: Grober Farbwurf, auf Basis des Scribbles	9
Abbildung 5: Darstellung des Farbschemas in „Kuler“	9
Abbildung 6: Finales Design: Umsetzung des Farbkonzeptes auf der Startseite.....	10
Abbildung 7: Schriftmuster von „Courier New“	10
Abbildung 8: Erster Entwurf mit gleichbleibenden Logo	11
Abbildung 9: Zweiter Entwurf mit sich änderndem Logo	11
Abbildung 10: Finale Fassung des Seitenlayouts.....	11
Abbildung 11: Verteilung der Aufgabenbereiche innerhalb des Teams.....	13
Abbildung 12: Geschachtelte XSLT	16
Abbildung 13: WCAG 1.0 - A Auszeichnung	17
Abbildung 14: Mobile Kompatibilität 95% (W3C)	17
Abbildung 15: Technischer Rahmen (gestrichelt) im Verhältnis zum Rest.....	19
Abbildung 16: Einteilung des Seitenrandes der Stickys	21
Abbildung 17: Seite mit zwei Blättermöglichkeiten	21
Abbildung 18: Detailansicht auf Buchseiten	22
Abbildung 19: Schachtelung der Div-Elemente zur Realisierung der Hover-Effekte im Steckbrief.....	23
Abbildung 20: Detailansicht mit gefärbten Fillern	24
Abbildung 21: Detailansicht verschobener gefärbter Filler	24
Abbildung 22: Erzeugung der Quiz XHTML.....	25
Abbildung 23 XHTML 1.1 valide	26
Abbildung 24 CSS 2.1 valide: Schachtelung von Div-Elementen.....	26

Abkürzungsverzeichnis

Sticky	Lesezeichen
IE8	Microsoft Internet Explorer 8
FF5	Mozilla Firefox 5

1 Einleitung

1.1 Aufgabenstellung

Die Aufgabenstellung dieser Hausarbeit gibt vor, einen der 13 unabhängigen Inselstaaten der Karibik oder eine TV-Serie vorzustellen. Hierzu soll ein kleines Team eine Website zu dem gewählten Thema entwickeln und der Prozess dokumentiert werden. Die Inhalte der Website sollen anhand eines webbasierten Quiz' oder Spieles abgefragt werden. Der Einsatz von Codegenerierungsprogrammen ist nicht vorgesehen, die Website soll komplett selbstentwickelt werden.

1.2 Unsere Zielsetzung

Die Webseite soll Urlauber und an der Region „Bahamas“ interessierte Webnutzer ansprechen. Es sollen vom erfahrenen Surfer bis zum gelegentlichen Nutzer alle möglichen Besucher angesprochen werden.

Dem Besucher sollen sowohl Fachwissen über die Bahamas als auch ein verspielter und fröhlicher visueller Eindruck vermittelt werden. Dazu wird die grafische Aufbereitung in origineller Form an den zu vermittelnden Inhalten ausgerichtet.

Ein Buch lässt sich von jedem Teil der Zielgruppe gleichermaßen intuitiv bedienen. Sowohl für den Urlauber als auch den interessierten Webnutzer stellt es dennoch ein neuartiges Surferlebnis dar, welches das rustikale Inselflair bereits durch Design und Aufmachung vermittelt.

Die Homepage soll möglichst kompatibel zu den verschiedenen Benutzergruppen sein: Besucher mit langsamer Internetleitung, Besucher mit Handicap, Besucher mit Smartphone, et al. Es soll daher eine Standardversion, eine barrierefreie Version, eine Version für Mobiltelefone sowie eine Version zum Ausdrucken geben.

Eingesetzt werden nur Standards in der Webentwicklung. Um eine möglichst hohe Browserkompatibilität zu ermöglichen, wird (außer im Quiz) auf den Gebrauch von Javascript weitestgehend verzichtet. Das Design mithilfe von CSS 2.1 und XHTML 1.1 umgesetzt.

2 Erstellungsprozess

2.1 Erste Schritte und Einrichten der Entwicklungsumgebung

Unmittelbar nach der Teamfindung und Empfang der Aufgabenstellung wird folgende Absprache bezüglich des weiteren Vorgehens bis zum Samstag, den 6.8.2011 getroffen: Es wird vereinbart, dass jedes Teammitglied dem jeweils anderen zwei zuvor erdachte Themen vorschlägt, zu jeweils einem Film und zu einem Karibikstaat. Am 06.08.2011 wird nach einer kurzen Präsentation der Ideen (V – die Besucher, Bahamas, Barbados) der Themenwunsch gemeinsam auf „Bahamas“ festgelegt.

Kai übernimmt anschließend die Aufgabe, die technischen Anforderungen an die Homepage herauszuarbeiten, eine Vorlage für einen Projektplan bereitzustellen und dort den terminlichen Rahmen der Hausarbeit einzutragen. Außerdem befasst er sich mit der Bereitstellung des Versionsverwaltungssystem / Repository „GITHUB“ für beide Teammitglieder.

Oliver arbeitet währenddessen die Anforderungen an Design und Usability heraus, setzt eine Vorlage für die Dokumentation auf und erstellt eine Liste mit Validatoren für die verwendeten Sprachen.

Anschließend werden die zu verwendenden Websprachen festgelegt: XHTML 1.1, CSS2.1, Javascript, XML, XSD, XSLT, XHTML. Es erfolgt eine Einarbeitungsphase in die Entwicklungsumgebung Microsoft Visual Studio 2010 und die Versionsverwaltungsumgebung GITHUB. Hierbei handelt es sich um ein dezentrales Versionsverwaltungssystem, bei dem jedes Teammitglied zu jeder Zeit über eine lokale Kopie der gesamten Daten verfügen kann, um uneingeschränkt damit arbeiten zu können. Es wird außerdem ein Test der Grundfunktionen von GITHUB durchgeführt: „Commit, Push, Pull und Merge“¹. Anschließend wird das gleichzeitige Ändern einer Datei und anschließendes Zusammenführen in einen Branch in GITHUB getestet. Durch das Testen wird das Einrichten der gemeinsamen Entwicklungsumgebung abgeschlossen.

Im Anschluss wird ein Scribble angefertigt und das Grobkonzept für die Website festgelegt.

Nach der Genehmigung des Grobkonzeptes wird das Gesamtkonzept erstellt. Hierzu gehören neben der Aufteilung der einzelnen Arbeitspakete unter den Teammitgliedern und der Zeitplanung auch ein Proportionskonzept, ein Farbkonzept und das Festlegen eines Schriftbildes. Der Projektplan ist im Anhang zu finden.

2.1.1 Proportionskonzept

Dem Besucher soll ein „Ringbuch“ angeboten werden. Als Vorgabe für die Aufteilung einer „Ringbuchseite“ werden folgende Maße festgelegt:

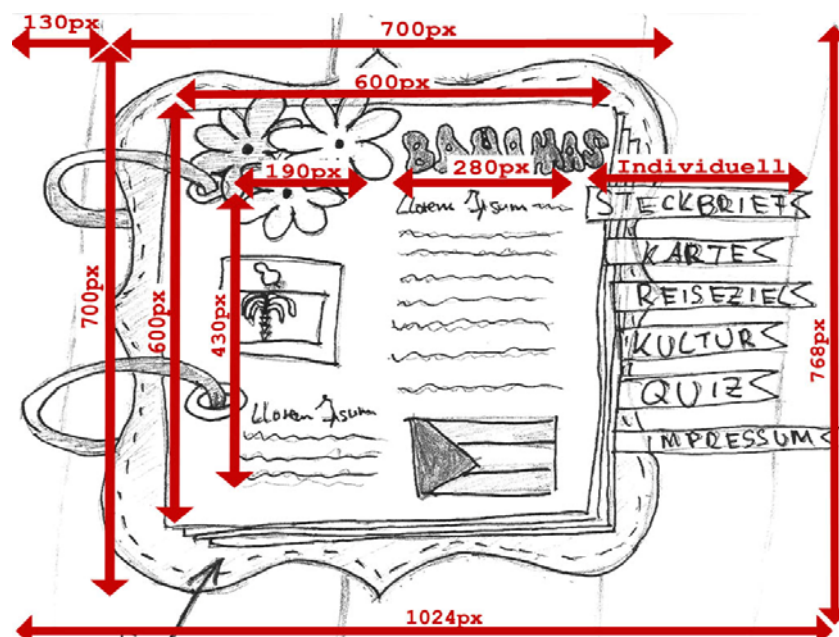


Abbildung 1: Plan mit Pixelangaben

Die Website soll statisch aufgebaut werden, d.h. im Layout-Bereich wird größtenteils mit absoluten Größenangaben gearbeitet. Egal, auf welchem Browser die Seite angezeigt wird, sie liegt immer oben links in der Ecke. Auf Bildschirmen mit einer Auflösung von bis zu 1700 x 1200 Pixeln ist eine fehlerfreie Darstellung möglich. Somit ist die Seite praktisch auf allen

¹ Weitere Infos: <http://b.lesseverything.com/2008/3/25/got-git-howto-git-and-github>

aktuellen Geräten fehlerfrei darstellbar. Vom linken Bildschirmrand bis zum eigentlichen Ringbuch wird eine 130 Pixel große Lücke gelassen, um das „Aufblättern“ des Buches realisieren zu können; hier liegen, sobald man die Startseite verlässt, die umgeblätterten Buchseiten.

Beim Entwurf des Titelbilds wird auf klassische Gestaltungsraster wie z.B. den Goldenen Schnitt zurückgegriffen, um dem Benutzer von Anfang an ein vertrautes, ansprechendes Erscheinungsbild zu präsentieren.



Abbildung 2: Verwendung des Goldenen Schnitts beim Entwurf des Titelbilds

Auch die Proportion der beiden Fließtextelemente wird am Goldenen Schnitt ausgerichtet. Diesem Muster entspricht außerdem das Verhältnis zwischen den Wendepunkten der geschwungenen Form der Ledermappe, auf der die Seiten liegen.

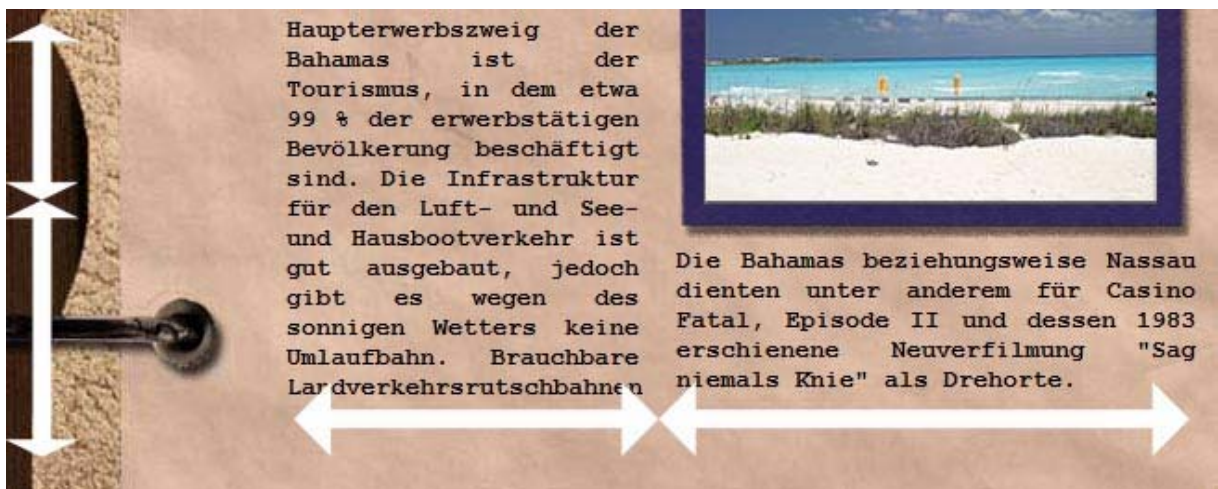


Abbildung 3: Verwendung des Goldenen Schnitts beim Layout des Fließtextes und der Ledermappe

2.1.2 Farbkonzept

Da die Entscheidung auf das Thema „Bahamas“ gefallen ist, stellt sich nun die Frage nach dem geeigneten Farbschema. Das Design einer Website lebt vor allem von stimmigen räumlichen Proportionen und zum Thema passenden Farben – einem Vertrauten Umfeld. Bestimmte Assoziationen und Empfindungen werden mit gewissen Farben verbunden. Da die Bahamas Inseln in der Karibik sind, ist es unsere Intention, den Eindruck von Sonne, Strand und Meer zu vermitteln. Daher fällt die Wahl auf die folgenden Farben:



Abbildung 4: Grober Farbwurf, auf Basis des Scribbles

Die Grundfarbe ist ein zurückhaltendes Ocker, die Farbe von Sand und eben von altem Papier. Darauf aufbauend werden die Farben der bahamaischen Flagge verwendet: Das Orange erinnert an die Farbe der Sonne und hat dadurch eine heitere und auflockernde Wirkung. Blau gilt als Farbe der Ferne und der Weite und wird mit Himmel sowie Wasser assoziiert.

Wie aus der Auflistung der für die Website zu verwendenden Farbtöne in „Kuler“ leicht zu erkennen ist, wird stark in Richtung der rot-bräunlichen Farbtöne gewichtet.

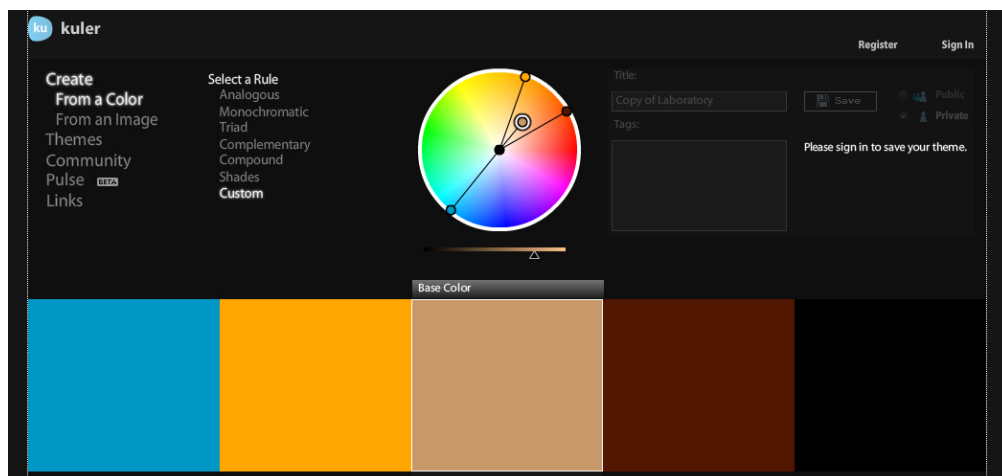


Abbildung 5: Darstellung des Farbschemas in „Kuler“

Durch deren farbliche Nähe wird ein zusammengehöriges und harmonisches Gesamtbild geschaffen. Die Farben Orange, Ocker und Braun sind sehr harmonisch in der Kombination. Dies lässt sich auch anhand der Symmetrie der Verteilung im Farbkreis leicht erkennen.

Der bewusste Einsatz des Komplementärkontrastes mithilfe der Farbe Blau, zum Beispiel in der Navigationsleiste, setzt darüber hinaus schmückende Akzente. Es sollen deutliche, jedoch nicht zu extreme Farbkontraste zwischen Vorder- und Hintergrund herrschen. Obwohl die Liste der eingesetzten Farben relativ lang ausfällt, wirkt deren Kombination weder zu bunt noch überladen. Die verwendete Farbkombination (Blau: Himmel, Gold: Sonne, Braun: Erde) ist in dieser Form in der freien Natur ubiquitär, somit wird auf Anhieb ein vertrauter Eindruck erweckt.



Abbildung 6: Finales Design: Umsetzung des Farbkonzeptes auf der Startseite

2.1.3 Schriftbild

ABCDEFGHIJKLMNOPQRSTUVWXYZ ; [ÜÖÄ]
 abcdefghijklmnopqrstuvwxyz : (üöä ß)
 0123456789

Abbildung 7: Schriftmuster von „Courier New“

Über die gesamte Webseite hinweg wird nur eine Schriftart verwendet: „Courier New“. Sie gehört zu den websicheren, auf den meisten gängigen Computern vorhandenen Schriftarten. Die Wahl fällt auf Courier New, da diese Schriftart dafür bekannt ist, früher viel auf analogen Schreibmaschinen zum Einsatz gekommen zu sein, daher passt sie sehr gut zum Thema. Weiterhin erleichtert die Tatsache, dass es sich bei Courier New um eine Nicht-Proportionalschrift handelt, die Arbeit mit Fließtext im Blocksatz.

2.1.4 Details zur Designentwicklung

Um einen Wiedererkennungswert und Zusammenhang zwischen den Seiten zu erreichen, soll der Aufbau der einzelnen Seiten stets ähnlich sein. Dennoch nehmen die Titelseite und das Impressum, orientiert man sich am Design eines gewöhnlichen Buches, eine Sonderrolle ein.

Im Folgenden wird die Entwicklung des Aufbaus der regulären Seiten beschrieben.



Abbildung 8: Erster Entwurf mit gleichbleibenden Logo

Der Erste Entwurf sah ein Seitenlayout mit gleichbleibendem Logo vor (Blumen + „Bahamas“). Dieses sollte dann je nach Kapitel mit einer Kapitelüberschrift versehen werden. Ein gleichbleibendes Logo erzeugt eine gewisse Monotonie und nimmt dem eigentlichen Inhalt den Platz weg, daher wurde diese Idee wieder verworfen.



Abbildung 9: Zweiter Entwurf mit sich änderndem Logo

Daraufhin wurde ein platzsparenderes Logo verwendet und dieses -je nach Kapitel- mit einer kleinen Zeichnung, einem sogenannten „Doodle“, illustriert. Diese Zeichnungen wirkten jedoch unpassend und wurden daraufhin durch einfache Schriftzüge, unterlegt mit der gelben Grundfarbe, ersetzt.



Abbildung 10: Finale Fassung des Seitenlayouts

Um den Inhalt der Seite in maximalem Umfang präsentieren zu können, das aktuelle Kapitel darstellen zu können und trotzdem noch ein Verbindendes Element zu haben, fiel die Entscheidung auf dieses reduzierte, aber prägnantere Design.

Die Menüpunkte werden in Form von „Stickies“, welche zwischen den Buchseiten stecken, realisiert. Anschließend wird in Photoshop die Grundlage für einen Mouseover-Effekt geschaffen, indem die vorhandenen Stickies dupliziert und leicht horizontal nach rechts verschoben wieder eingefügt wurden. Desweiteren wird für die aktiven Seiten jeweils ein Sticky „auf“ die aktive Seite gelegt.

Abschließend werden sämtliche Bildelemente der Website durch „Slicing“ aus Photoshop exportiert.

2.1.5 Iterative Implementierungsphase

Die weitere Vorgehensweise zeichnet sich durch ein abwechselndes Fertigstellen von HTML-Dokumenten für die einzelnen Kapitel und durch die Anpassung der dazugehörigen

CSS-Datei aus. Zwischendurch wird nach Abschluss jeder Arbeitseinheit validiert, dokumentiert und das Ergebnis ins gemeinsame „GITHUB“-Repository hochgeladen.

Die HTML-Vorlage für die Steckbrief-Seite beinhaltet beispielsweise eine Karte der Bahamas, die mithilfe der CSS-Pseudoklasse `:hover` dafür sorgt, dass bei einem Mouseover über die Inselgruppen diese durch eine Farbveränderung hervorgehoben werden. Zusätzlich dazu erscheint rechts neben der Karte eine dazugehörige Steckbriefstabelle, die Informationen bezüglich der „gehoverten“ Inselgruppe beinhaltet. Dies wird durch die Schachtelung von Divs und Pseudoklassen ermöglicht. Technische Details lassen sich im zugehörigen Kapitel nachlesen.

Die HTML-Vorlagen werden im Anschluss modularisiert und mithilfe von XSLT-Templates realisiert. Das Quiz wird mithilfe von Javascript implementiert.

Daraufhin werden die XML-Seiten für die einzelnen Kapitel erstellt. Dies beinhaltet u.a. das Erstellen von Kapitelüberschriften (PNG), Bild- und Textelementen, die im XML mithilfe von `<paragraph>` und `<image>` eingefügt werden. Die Bildelemente werden aus Designgründen jeweils auf ein Post-It „geklebt“.

Anschließend wird die Print-CSS erstellt. Diese ermöglicht es dem User, allein den Content zu drucken. Das Ziel des Drucklayouts ist, dass keine Designelemente auf der fertig ausgedruckten Seite erscheinen.

2.1.6 Abschließende Qualitätssicherung

Abschließend wird ein Refactoring der kompletten Standard-CSS vorgenommen. Sie wird hierbei von Doppelungen und sonstigen unnötigen Fragmenten befreit und mit Kommentaren versehen. Es werden außerdem diverse feingranulare Änderungen an Fließtextformatierung, Layout und Usability vorgenommen.

Zum Beispiel wird nachträglich ein weiteres Feature realisiert, welches die intuitive Bedienbarkeit der Website unterstützt: das Blättern durch das Buch wurde durch ein Abnehmen der Seitenzahl auf der rechten Buchseite und eine Zunahme der Seiten auf der linken Seite visualisiert. Außerdem wird implementiert, dass die Stickies nach einmaligem Anklicken nicht mehr im Buch stecken, sondern auf dem Tisch liegen. Diese Anpassung erfolgt aus Gründen der Logik – normalerweise müssten die Stickies ja nach dem Umblättern mit der umgeblätterten Seite aus dem linken Bildrand verschwinden.

3 Vorgehensweise

Der Erstellungsprozess der Website lässt sich grob in vier Phasen unterteilen:

- **Planungsphase:** Erstellung eines Gesamtkonzeptes inklusive Zeitplan, Arbeitspaketen und deren Aufteilung (siehe: Projektplan im Anhang)
- **Designphase:** Erstellung der Website als „ein Bild“ in „Photoshop“
- **Iterative Implementierungsphase:** Umsetzen des Gesamtkonzeptes in eine fertige Website mit parallel laufender Dokumentation, Validierung und Qualitätssicherung. Das Grobkonzept dient hierbei als ein omnipräsenter Leitfaden.
- **Abschließende Qualitätskontrolle:** Refactoring und Feinschliff

4 Aufteilung der Arbeit

Innerhalb des Teams findet eine strikte Aufgabenteilung statt.

Oliver ist verantwortlich für das Design, das Layout und die Struktur der Website. Dies beinhaltet zum Einen das Erstellen sämtlicher benötigter Grafiken mithilfe von Photoshop, das Exportieren der Grafiken in PNG- oder JPEG-Dateien und deren Namensgebung. Oliver ist zuständig für das Anfertigen sämtlicher HTML-Vorlagen und des Stylesheets: die Aufteilung der Seitenelemente in DIV-Container und die Findung von geeigneten Layouteigenschaften wie z. B. Positionierung und Größe, Hintergrundgrafik und Mouseover-Effekten. Darüber hinaus sucht er nach passenden Inhalten in Form von Schrift und Bild und kümmert sich um deren sinnvolle Einbindung im XML.

Kai ist zuständig für die Konzeptionierung und Umsetzung des technischen Aufbaus in XSD und XSLT, sowie der Modularisierung des Inhalts. Inhalte werden auf diese Weise formalisiert, standardisiert und in Templates gegossen. Ebenfalls implementiert er das Javascript zur Darstellung des Quiz und bereitet die Schemadefinition der einzelnen XML Dateien vor.

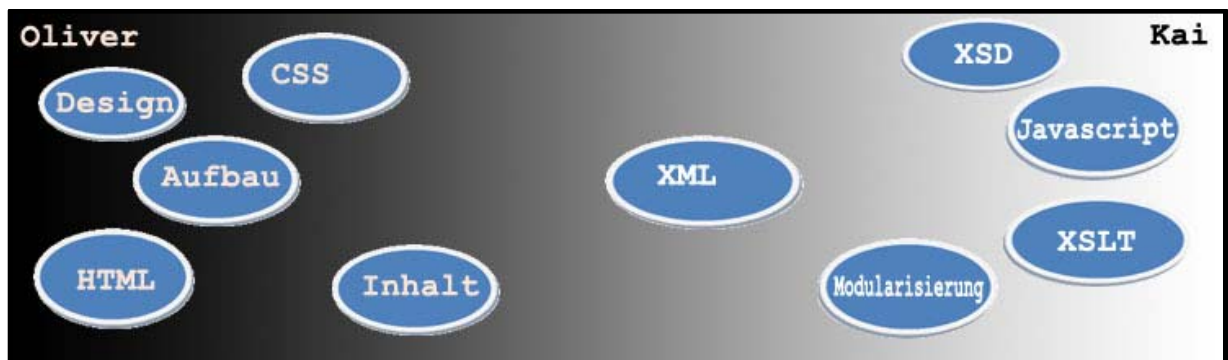


Abbildung 11: Verteilung der Aufgabenbereiche innerhalb des Teams

5 Probleme

Im folgenden Kapitel werden technische Probleme und deren Lösungsansätze beschrieben. Dazu wurde im Laufe der Arbeit folgendes Schema entwickelt, um mit Problemen umzugehen:

- Problem erkannt und nicht innerhalb von 30 min unter Zuhilfenahme von Google selbst lösbar?
 - Rat suchen bei Teampartner
- Problem gemeinsam nicht innerhalb von 30 min lösbar?
 - Rat suchen in der Zenturie
- Problem nicht nach weiteren 60 min lösbar und zum Erreichen des Ziels relevant?
 - Dozent fragen

Der erste Schritt dieses Schemas ist besonders wichtig, denn das Problem muss eingegrenzt, reproduziert und möglichst einfach beschrieben werden. Dennoch ist eine Zeitbegrenzung notwendig, da die Zeit beim Suchen schnell „aus den Augen verloren wird“.

5.1 „Flickering Bug“ im IE8

5.1.1 Beschreibung

Beim Anzeigen von Webinhalten im IE von einer lokalen Quelle wird stets der gesamte Inhalt einer Seite neu geladen. Das führt dazu, dass beim Navigieren nicht alle Elemente gleichzeitig geladen werden, sondern erst nacheinander. Bevor das eigentliche Bild angezeigt wird, wird eine Füllfläche angezeigt. Deren Farbe kann zwar festgelegt werden, dennoch entsteht beim Laden ein kurzes „Aufblitzen“ („flickering“).

5.1.2 Lösungsansatz

Die ausprobierten Lösungsansätze bestanden im wesentlichen darin, durch gewisse Steuer-tags zu versuchen, dem IE das „Cachen“ von Dateien anzugewöhnen. Der User könnte das Problem durch diverse Einstellungen innerhalb des Browsers vermeiden, jedoch ist mit einem derartigen Verhalten normalerweise nicht zu rechnen.

Bei einer Veröffentlichung der Website in Internet träte dieses Problem aufgrund von bestimmten, vom Webserver gesendeten Headern nicht mehr auf.

Für die lokal gespeicherte Version ist uns allerdings kein Lösungsansatz für die Konstellation Win 7 + IE 8 bekannt.

5.2 IE8: Hover auf unsichtbare Boxen

5.2.1 Beschreibung

Auf nicht sichtbaren Elementen, die z.B. durch das Style Attribut „**visibility:hidden;**“ gekennzeichnet sind, kann nicht mit der Maus drüber gefahren („gehovert“) werden. Es ist somit nicht möglich auf Pseudklassen wie z.B. „**:hover**“ zuzugreifen. Div-Boxen werden ebenfalls dann „unsichtbar“, wenn sie über keinen oder unsichtbaren Inhalt verfügen (ist leider auch ein unsichtbarer Inhalt). Weder Positionierung, noch die Größe der Box spielt dabei eine Rolle.

5.2.2 Lösungsansatz

Durch Hinzufügen eines neuen „Filler“-Elementes, das eine Hintergrundgrafik enthält, wird das Element „sichtbar“, nun kann es „angehovert“ werden. Dazu wird ein 1x1px großes, transparentes PNG-Bild als Hintergrund für ein „Filler“-Element eingebunden. Die Größe dieses „Fillers“ wird auf die Dimension des „anzuhovernden“ Bereiches gesetzt.

Das Problem tritt in Kombination mit dem „Schachtelungsfehler“ auf.

5.3 Schachtelungsfehler durch XSLT im IE

5.3.1 Beschreibung

Wird im Internetexplorer via XSLT eine Div-Box (auch getestet mit „<a>“ ohne Inhalt) erstellt, so wird diese automatisch als verkürztes Element: „<div/>“ interpretiert. Dies sorgt dafür, dass alle zukünftigen Elemente fälschlicherweise unterhalb des Elements angeordnet werden. Ein explizites Angeben im XSLT Stylesheet („<div></div>“) sorgt für dieselben Probleme.

5.3.2 Lösungsansatz

Eine grundsätzliche Vermeidung leerer Tags ist bei dynamischem Aufbau der Seite nicht an allen Stellen möglich. Gerade für das Navigationskonzept ist es notwendig, <a> - Tags ohne Inhalt zu definieren.

Aus diesem Grund müssen diese Tags mit Inhalt gefüllt werden. Dieser Inhalt wird mit: „**text-indent: -9999px;**“ verschoben, sodass er nicht mehr im sichtbaren Bereich des Bildschirms liegt. Zusätzlich wird bei Links noch: „**outline:none;**“ gesetzt, um die Unterstreichung zu entfernen.

5.4 Kein CSS ohne Angabe von Namespace in Javascript im FF

5.4.1 Beschreibung

Verwendet man im Javascript zur Erzeugung von HTML-Objekten im Doctype XHTML 1.1 das Kommando: `document.createElement(„...“)`, so werden die Elemente zwar Erzeugt und im Quelltext angezeigt, dazugehörige CSS Styles werden jedoch im FF nicht eingebunden.

5.4.2 Lösungsansatz

Um die Differenz zwischen IE und FF zu überwinden, ist eine Browserweiche von Nöten. Diese kann realisiert werden mit: **if (navigator.appName == "Microsoft Internet Explorer") {*//Code Für IE*}else {*//Code für FF*}**. Im Firefox ist es nötig, zur Erzeugung der Elemente auch den korrekten Namespace zu übergeben, das geschieht mit folgendem Kommando: `document.createElementNS("http://www.w3.org/1999/xhtml", name);`

5.5 Umlautprobleme trotz UTF-8 im IE

5.5.1 Beschreibung

Sonderzeichen werden im IE nicht richtig ausgegeben, obwohl folgendes Tag gesetzt wurde:

```
<xsl:output encoding="utf-8" method="xml" version="1.0" indent="yes" doctype-  
public="-//W3C//DTD XHTML 1.1//EN" doctype-  
system="http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd"/> .
```

Ebenfalls wurde in der XML Datei im <head>-Bereich folgendes gesetzt: **<?xml version="1.0" encoding="utf-8"?>**.

Die Encodierung der Datei ist UTF-8. Im Browser zeigt sich jedoch, dass die Kodierung nicht auf UTF-8 gesetzt wird.

5.5.2 Lösungsansatz

Zu diesem Problem wurde keine plausible Erklärung gefunden. Es stellte sich jedoch heraus, dass beim Entfernen des `encoding="utf-8"` (rot markiert) die Zeichenkodierung richtig erkannt wurde und das Problem somit als gelöst erscheint.

5.6 XSLT „apply templates + parameter“

5.6.1 Beschreibung

```
<xsl:apply-templates>
<xsl:with-param.../>
<xsl:apply-templates/>
```

Das beschriebene Konstrukt ist laut W3C spezifiziert² und soll Parameter übergeben. Dies ist jedoch in dieser Form nicht von den XSLT-Engines der Browser vorgesehen und funktioniert nicht.

5.6.2 Lösungsansatz

Geplant ist ein Dictionary, in dem der Name eines Kapitels einer Technischen ID zugeordnet wird und weitere Parameter herausgesucht werden. Um dies zu realisieren wurden drei Lösungsansätze erprobt:

- Das Dictionary könnte über ein spezielles Template an die XML Dateien der Seite „angehängt“ werden:

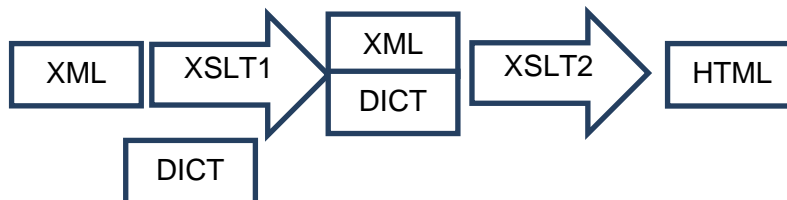


Abbildung 12: Geschachtelte XSLT

Diese Lösung wird grundsätzlich durch gängige XSLT Transformatoren unterstützt, nicht jedoch von denen der Webbrowser.

- Loslösung vom funktionalen Programmierkonzept der XSLT und Nutzung von imperativen Sprachelementen. Das führt in diesem Falle zu einer Code-„Duplette“, die in Form eines Snippets immer wieder genutzt werden kann:

```
<xsl:for-each select="$chapters/chapters/chapter">
  <xsl:if test="@name = $chapter">
    <xsl:variable name="chapterNr" select="@nr"/>
    ...
  </xsl:if>
</xsl:for-each>
```

Da in diesem Falle nur der Zugriff auf ein Template nötig war, wurde diese Variante gewählt. Dieses Codesnippet lässt sich ebenfalls nicht in eine Variable auslagern, da diese zum Zeitpunkt der Erstellung noch keinen Konkreten Wert für eine Ausgabe annehmen kann.

- Nutzung eines Imperativen Konzepts für das vom XSLT vorgenommene Pattern-Matching. Auf diese Weise behält man die Kontrolle und kann nach Benutzung dieses Snippets voll funktional fortfahren. Daher kommt diese Lösung neben der zuvor beschriebenen ebenfalls zum Einsatz:

² Siehe: http://www.w3schools.com/xsl/el_apply-templates.asp

```

<xsl:for-each select ="descriptionBox/*">
  <xsl:if test="name() = 'table'">
    <xsl:call-template name="table">
      <xsl:with-param name="classTable">...</xsl:with-param>
    </xsl:call-template>
  </xsl:if>
  <xsl:if test="name() = 'picture'">
    <xsl:call-template name="picture"/>
  </xsl:if>
  <xsl:if test="name() = 'paragraph'">
    <xsl:call-template name="paragraph"/>
  </xsl:if>
</xsl:for-each>

```

Dieses Snippet muss Aufgrund des manuellen Matchings bei hinzukommenden Elementen stets erweitert werden.

6 Abweichungen vom Grobkonzept

Von der Entwicklung einer barrierefreien Version und einer Version für Mobiltelefone wurde abgesehen, nachdem die Überprüfung auf „Web Content Acessibility“³ den Level „A“ erhielt und die Konformitätsprüfung für mobile Endgeräte⁴ ein Ergebnis von 95 % lieferte.



Abbildung 13: WCAG 1.0 - A Auszeichnung



Abbildung 14: Mobile Kompatibilität 95% (W3C)

Von den zuvor definierten Anforderungen an Inhalt, Quiz, Ziele, Zielgruppe, Technische Mittel und Darstellungsform musste nicht abgewichen werden.

³ Siehe: <http://achecker.ca/checker/index.php>

⁴ Siehe: <http://validator.w3.org/mobile/>

7 Aufbau des Webauftritts

7.1 Inhalt

Der Inhalt ist in folgende Themen gegliedert:

- **Steckbrief:**
Auf einer Seite wird ein kurzer Überblick über die Bahamas präsentiert.
- **Karte:**
Auf dieser Seite werden die einzelnen Inseln gezeigt jeweils Informationen über deren Eigenschaften (u.A. Größe, Lage, Einwohnerzahl, Tourismus, Geschichte, Kultur, Sehenswürdigkeiten ...) vermittelt.
- **Reiseziel:**
Hier werden sowohl visuelle Eindrücke als auch Informationen für Touristen gezeigt.
- **Kultur:**
Exemplarisch werden Besonderheiten der Kultur in schriftlicher und visualisierter Form dargestellt.
- **Quiz:**
In Bezug auf die einzelnen Themen werden Fragen in Form eines Multiple-Choice-Tests an den Besucher gestellt. Der Besucher kann sich die Fragen auswerten lassen und bekommt am Ende eine Bewertung („Einheimischer“, „Interessierter Tourist“ oder „Handtuchkrieger“) und eine Korrektur der Antworten.

Die Startseite trägt –vergleichbar zu einem Buchcover- keinen wirklichen Inhalt außer dem Titelbild. Dies würde das Erscheinungsbild stören und die Startseite „überladen“.

Blättert man weiter zum Menüpunkt „Steckbrief“, findet man eine Karte wieder, unter der in tabellarischer Form Inhalte dargestellt werden. Der Besucher wird darauf hingewiesen, dass er, sobald er mit der Maus über die Karte fährt, weitere inselspezifische Informationen angezeigt bekommt.

Die nächsten beiden Unterkapitel „Reiseziel“ und „Kultur“ beinhalten Fließtext und inhaltsbezogene Bilder. Innerhalb dieser Kapitel lässt sich durch einen Klick auf die untere rechte Ecke der Seite weiterblättern. Dies wird durch ein Eselsohr, welches sich vergrößert, sobald man mit der Maus darüber fährt, verdeutlicht. Somit wird dem Benutzer eine intuitive Steuerung angeboten.

Ein direkter Sprung zu einem weiter unten gelegenen Kapitel, zum Beispiel von der Startseite zum Quiz, wird durch Anklicken eines der Lesezeichen („Sticky“), die auf der rechten Seite der Buchseiten „eingeklebt“ sind, ermöglicht. Sobald man weiterblättert, liegt der Sticky, welcher zuvor noch oben auf der angezeigten Seite lag, anschließend auf dem Tisch. Dieses Verhalten entspricht dem Verhalten eines Buches, da man beim Weiterblättern eigentlich davon ausgehen müsste, dass der Sticky „mit-umgeblättert“ würde. Da die linke Seite des Buches nicht komplett angezeigt wird, müsste der Sticky eigentlich „aus dem linken Bildrand geblättert“ werden. Damit auch nach dem Weiterblättern eine Navigation möglich ist, haben wir uns dafür entschieden, den Sticky stattdessen auf den Tisch zu „fallen zu lassen“. So wird erreicht, dass weiterhin eine Navigation in der gewohnten Reihenfolge möglich ist, ohne Abstriche bei der Logik machen zu müssen.

Die erste Seite des Kapitels verfügt über ein Eselsohr zum Weiterblättern, die zweite Seite darüber hinaus über eines zum zurückblättern. Auf der letzten Seite befindet sich lediglich

ein Eselsohr zum zurückblättern. So wird signalisiert, dass es sich um das Ende dieses Kapitels handelt.

Das nächste Kapitel des Buches ist das Quiz. Hier wird dem Benutzer die Möglichkeit gegeben, sich bezüglich der zuvor vermittelten Inhalte eines kleinen Wissenstests zu unterziehen. Dieser soll jedoch nicht immer derselbe sein, sondern aus einem Fundus von Fragen generiert werden. Das Quiz lässt sich auswerten und der Benutzer wird durch einen „Rang“ bewertet.

Anschließend folgt ein Impressum mit Adresse des Seiteninhabers und rechtlichen Hinweisen.

8 Besonderheiten des technischen Rahmens

Der technische Rahmen ist die Gesamtheit sämtlicher allgemeinen Grundfunktionalitäten, die nicht seitenpezifisch programmiert oder eingestellt werden müssen. Sie gelten für alle Seiten und verhalten sich für den Seitenpfleger transparent. Durch Modularisierung dieser Komponenten ist es möglich, die Inhalte getrennt voneinander zu erweitern und frei von Nebenwirkungen anzupassen.

Durch den technischen Rahmen soll ermöglicht werden, dass der Inhalt der Seite streng vom Layout getrennt wird. Dabei werden die Einflussmöglichkeiten auf Stiltransformationen trotzdem durch die Möglichkeit des Customizing maximal gehalten.

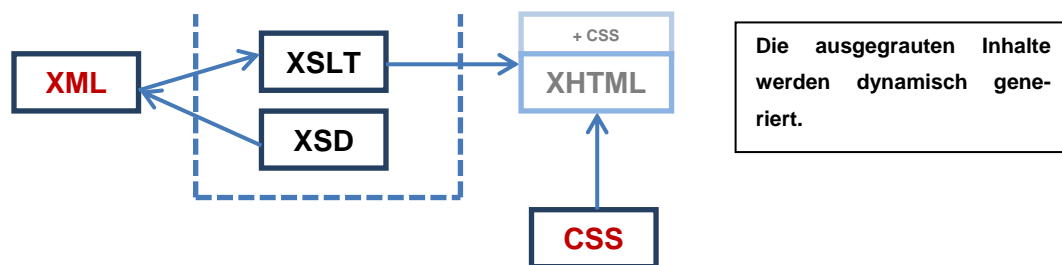


Abbildung 15: Technischer Rahmen (gestrichelt) im Verhältnis zum Rest

Für den Seitenpfleger sind nur die rotgekennzeichneten Elemente von Relevanz, um die Seite anpassen und Pflegen zu können. Die Definition des XML-Schemas gibt den Aufbau der XML-Datei vor, die vom XSLT dann transformiert wird. Eine XML-Datei enthält lediglich den Inhalt einer Seite.

Im CSS macht der Seitenpfleger Angaben zum Design der Seite. Um nicht jeden „Trivialinhalt“ (z.B. Namen oder Hintergrundbilder für die Navigation oder Logos im Kopfbereich) im CSS pflegen zu müssen, ist dieser über Namenskonventionen festgelegt. Diese Inhalte werden vom XSLT bei Bedarf dynamisch an die XHTML Datei angehängt.

Dieses Konzept gibt die Möglichkeit, theoretisch jede beliebige Ausgabe nutzen zu können (SGML, PDF, etc...). Die Struktur der XML Datei ist unabhängig von der Ausgabe. Ebenso kann der Aufbau der Seite mit geringfügigen Änderungen angepasst werden.

Genauso bietet sich die Möglichkeit, statt der XML-Datei ein Content Management System als Datenquelle zu nutzen.

Im Folgenden werden einzelne Module des technischen Rahmens genauer beschrieben. Die technische Realisierung ist im Quelltext ergänzend kommentiert.

8.1 Namenskonventionen

Da für dieses Projekt viele gleichartige Bilder benötigt werden, wurde insbesondere für Bild-dateien eine Namenskonvention festgelegt:

„genereller Zusammenhang“_„zugehöriges Kapitel“_„Typ“.Dateiendung.

Zum Beispiel lautet also der Name der Mouseover-Version des Stickies „Kultur“, welches der Navigationsleiste zugehörig ist: „nav_kultur_hover.png“.

Generell wird die Übereinkunft getroffen, dass sprechende Namen abstrakten Abkürzungen vorzuziehen sind.

Somit ist es für den Seitenpfleger möglich, auf einfache Weise Bilder zu ändern, ohne speziellen Code modifizieren zu müssen. Auf der anderen Seite ist beim Hinzufügen eines neuen Kapitels daran zu denken, dass im Layout-Ordner der Bilder ein neues Sticky für die Zustände: „Aktiv“, „Normal“, „Hover“, „Tisch“, „Tisch-Hover“ mit entsprechenden Dateinamen zu erstellen sind. Diese lassen sich aber aufgrund des schematischen Aufbaus selbst ohne Dokumentation recht schnell herausfinden.

Die Anwendungsbereiche der Namenskonvention erstrecken sich über fast alle Komponenten des Rahmenwerkes und finden ebenfalls Anwendung in der Komponente des Steckbriefes sowie in `<picture>`-Tags.

8.2 Navigationsstruktur

Ziel ist es, die Navigationsstruktur durch einen technischen Rahmen abzubilden, bei dem der Seitenpfleger sich nicht um Implementationsdetails oder Logik der (komplexen) Navigationsstruktur kümmern muss.

Die Navigationsstruktur besteht technisch aus zwei Ebenen:

- Stickys zum Navigieren in einzelne Kapitel
 - Eselsohren in den Seiten, um innerhalb der Kapitel durch Blättern navigieren zu können

Navigationselemente bestehen aus einzelnen Bildern mit Linkfunktion. Diese verfügen zusätzlich über einen Hover-Effekt. Realisiert wurde dies mit einem `<a href=“...”`

`alt=“...”>` Tag mit Hintergrundbild und zusätzlich `display:block` zu dem eine `a: hover` Pseudoklasse im CSS Stylesheet definiert wurde. Dieses Tag muss zwar technisch über einen Inhalt verfügen, der wird jedoch über das CSS ausgeblendet.

Um eine Zuordnung einer Seite zu einem Kapitel treffen zu können, wird zum einen eine Namenskonvention bei der Erstellung von Seiten benutzt, zum anderen ein Steuertag:

```
<pageInfo nr="3" lastPage="true" chapter="Kultur" typ="normal" />
```

Es wird in diesem Steuertag für die Seite angegeben, welche Nummer sie innerhalb des Kapitels trägt (zweiter Bestandteil des Dateinamens) und ob diese Seite die letzte Seite des Kapitels ist. Somit ist es nicht notwendig, im Vorfeld zu wissen, wie viele Seiten es zu einem Kapitel insgesamt geben soll.

Desweiteren muss für die Navigation noch das Kapitel, zu dem die Seite gehört, angegeben werden (erster Bestandteil des Dateinamens). Diese Attribute sind Pflichtattribute und müssen gesetzt werden.

8.2.1 Stickys



Die allgemeinen Attribute der Stickys werden im CSS Stylesheet definiert (Größe, Anfangsposition, position: absolute, display: block). Somit entsprechen diese den auf dem Bild gezeigten Slices mit gleichbleibender Größe und Abstand. Über die Namenskonventionen werden durch das XSLT-Stylesheet die Hintergrundbilder und die relative Position als CSS Attribut gesetzt.

Abbildung 16: Einteilung des Seitenrandes der Stickys

Ein Sticky kann 5 Zustände haben:

- **Normal:** Die ID des aktuellen Kapitels ist kleiner als die ID des Stickys (Zuordnung über chapters.xml)
- **Hover:** Die Maus wird über ein Kapitel im Status Normal gefahren
- **Aktiv:** Die ID des aktuellen Kapitels entspricht der ID des Stickys
- **Tisch:** Die ID des Stickys ist kleiner als die ID des aktuellen Kapitels oder das Sticky befindet sich im Zustand Aktiv und die Seitenzahl des Kapitels ist größer als 1. (Info kommt aus dem Dictionary chapters.xml)
- **Tisch-Hover:** Ein Sticky, das auf dem Tisch liegt wird mit der Maus angefahren.

Diese fünf Zustände werden durch das XSLT Stylesheet erkannt und gesetzt.

8.2.2 Blättern

Um innerhalb einer Seite Blättern zu können, kann diese bis zu zwei „Eselsohren“ besitzen:

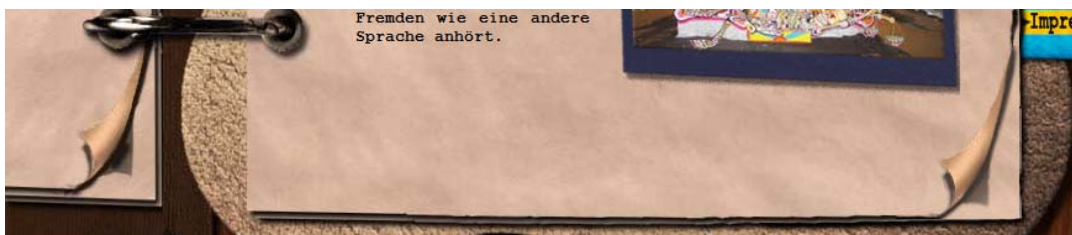


Abbildung 17: Seite mit zwei Blättermöglichkeiten

Eine Seite kann also über 4 Zustände verfügen:

- Keine Möglichkeit zum Blättern: Besteht, wenn die Aktuelle Seitenzahl = 1 und `lastPage = true`
- Nur Vorblättern möglich: `lastPage = false`
- Nur Zurückblättern möglich: `lastPage = true` und Seitenzahl > 1
- Vor-/ und Zurückblättern möglich: `lastPage = false` und Seitenzahl > 1

Technisch wurde das Einbinden der „Eselsohren“ genauso gelöst wie das Einfügen der Stickys. Die Verlinkungen werden dynamisch nach dem Prinzip „**Kapitel_Seitennummer.xml**“ gesetzt.

8.3 Buchverhalten

Damit sich die Homepage wie ein Buch anfühlt, gehört es dazu, dass die Seiten beim Umblättern auf einer Seite „mehr“- und auf der anderen Seite „weniger“ werden.

Dieses Verhalten wurde ähnlich der Links, nur mit Boxen im Hintergrund mit Hintergrundbild, dass nach Namenskonventionen benannt wurde, gelöst. Mit Hilfe der chapters.xml und dem Namen der Seite wird die Kapitelnummer herausgefunden und das entsprechende Hintergrundbild eingeblendet.

```
<chapter nr="3" name="Reiseziel" xConstraint="10" yConstraint="5"/>
```

Aus einem Eintrag im Dictionary wird die Nummer des Kapitels herausgelesen. Dieser Nummer sind die entsprechenden Hintergrundbilder für den linken und rechten Seitenstapel zugeordnet.

Zusätzlich kann ein Parameter („Constraint“) angegeben werden, der den gesamten Inhalt der Seite um angegebene Koordinaten verschiebt.



Abbildung 18: Detailansicht auf Buchseiten

Aus dieser Detailansicht wird ersichtlich, dass die einzelnen Seiten aus perspektivischen Gründen nicht direkt, sondern versetzt übereinander liegen. Würde der Inhalt der Seite konstant auf einer Stelle des Bildschirms stehen, so würde beim Weiterblättern das Logo nicht mehr oben rechts in der Seitenecke positioniert sein. Ebenso würden Eselsohren und Inhalt verrutschen. Aus diesem Grund ist es notwendig, für jedes Kapitel eine Verschiebung konfigurieren zu können.

Hierbei wird der Inhalt der Seite relativ um die entsprechenden Koordinaten verschoben. Da das Layout mit absoluten Positionsangaben arbeitet, ist die Verschiebung über dynamisch hinzugefügte „**margin-top: ...**“ Style-Elemente im XSLT Stylesheet realisiert worden.

8.4 Typisierung der Seiten

Vom Aufbau gesehen, sind die Seiten „Reiseziel“ und „Kultur“ in sich relativ gleich, unterscheiden sich aber völlig von den anderen Seiten. Aus diesem Grunde wurden die Seiten in verschiedene Typen kategorisiert. Die Kategorisierung fand sowohl nach Inhaltlichen wie auch Style-Gesichtspunkten statt. Auf diese Weise kann unabhängig vom Inhalt ein besonderes Verhalten je nach Seitentyp definiert werden.

Die Typen der Seiten werden im **<pageInfo>**-Element definiert. Folgende Ausprägungen sind vorgesehen:

- „**Normal**“: Normaler Seitenaufbau, 2 Spalten, Seitenlogo
- „**Start**“: Startseite mit einem Logo, weiterblättern möglich

- „Quiz“: Kein Inhalt, Einbindung von Javascript-Inhalten möglich, sowie onLoad-Funktion im Body
- „Impressum“: Ähnlich zu Startseite, kein Weiterblättern möglich
- „Overview“: Einspaltiges Layout: Aufteilung in Steckbriefbild (mit „Hovermöglichkeit“) und Infotabelle darunter

9 Besondere technische Aspekte einzelner Kapitel

9.1 Steckbrief

Um sicherzustellen, dass der Steckbrief von einer möglichst großen Benutzergruppe genutzt werden kann, werden die Hover-Effekte der Karte rein durch die Verwendung von CSS implementiert. Auf den Gebrauch von Javascript o.Ä. wird verzichtet. Der Effekt beruht grundsätzlich auf der Schachtelung von mehreren Div-Elementen. Auf oberster Ebene befindet sich ein Div-Element mit der ID „AlleInselnAufDerKarte“, welches die gesamte Karte und den Bereich der Tabellen, die bei Mouseover über die Inselgruppen erscheinen (im Folgenden „Mouseover-Tabelle“ genannt), beinhaltet. Innerhalb dieses Div-Elements sind drei weitere Div-Elemente enthalten: „InselObenLinks“, „InselUntenLinks“ und „InselUntenRechts“. Diese Drei enthalten jeweils einen weiteren Div-Container für die verschiedenen Mouseover-Tabellen.

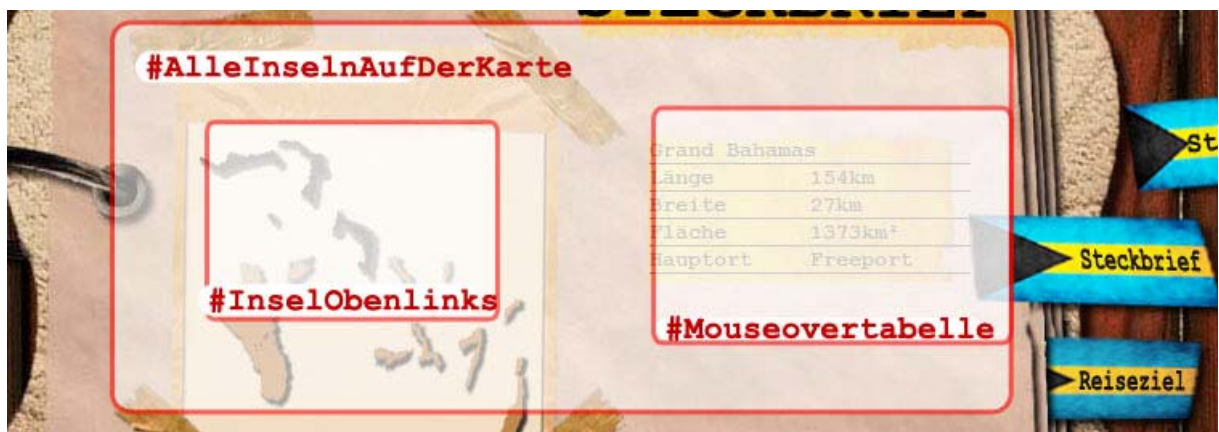


Abbildung 19: Schachtelung der Div-Elemente zur Realisierung der Hover-Effekte im Steckbrief

Diese zunächst kompliziert anmutende Schachtelung ist notwendig, um ein gleichzeitiges Ansteuern von zwei Elementen zu ermöglichen: der Inseln und der Mouseover-Tabelle. Im CSS werden die einzelnen Inseln mit der Pseudoklasse „:hover“ versehen, bei der über die Einbindung einer Hintergrundgrafik die Hervorhebung bei Mouseover entsteht. Die gleichzeitige Sichtbarmachung der Mouseovertabelle geschieht durch das Setzen von „display:block;“

in der Pseudoklasse

„#alleinselnaufderkarte #inseluntenrechts:hover .infobox“.

Die Positionierung der Elemente #InselObenlinks geschieht absolut und stellt somit den Bereich dar, der vom Mouseover erfasst werden kann. Die Position der #Mouseovertabelle ist jedoch relativ und stellt den Abstand zur oberen linken Ecke der #InselObenlinks, #InselUntenLinks bzw. #InselUntenRechts Box dar.

Gleichzeitig liegt ein gleich großer „Filler“ in #InselObenlinks, um die Box auf einer bestimmten Größe zu halten und es für den IE zu ermöglichen, dass diese Box „gehovert“ werden kann.

Der Seitenpfleger soll sich um diese technischen Aspekte nicht kümmern müssen.

```
<overviewPictureArea>
  <standardPicture>
    <picture file="steckbrief_karte.png" name="Karte"/>
  </standardPicture>
  <hoverArea target="330" target="90">
    <picture file="steckbrief_hoverbild_start.png" name="infos"/>
    <hoverPart posx="54" posy="91" sizex="116" sizey="85">
      <picture file="steckbrief_insel_oben.png" name="oben"/>
      <descriptionBox>
        ...
      </descriptionBox>
    </hoverPart>
  </hoverArea>
</overviewPictureArea>
```

Um die eigentliche, relative Positionierung kümmert sich das XSLT Stylesheet, sowie die Erzeugung der Filler-Elemente. Diese werden anhand von Namenskonventionen angelegt und über XSLT im Stylesheet mit den notwendigen Angaben Größe und Position aus den angegebenen Informationen angereichert.

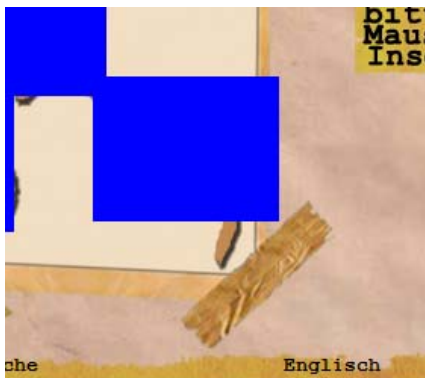
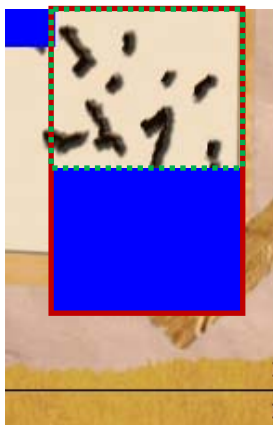


Abbildung 20: Detailansicht mit gefärbten Fillern

Diese Filler werden von verschiedenen Browsern unterschiedlich behandelt: Der IE8 verschiebt bei Mouseover die Fillerbox nach unten. Somit würde der Bereich, über den ein Mouseover möglich ist, vergrößert werden.



Das führt dazu, dass der Hover Bereich nun größer wird (**roter Rahmen**). Dies erscheint für den Benutzer als sehr verwirrend. Gelöst wurde dieses Problem mit einer weiteren CSS Definition `.filler:hover{ visibility:hidden;}`.

Die Wirkung davon ist für den IE 8 die Folgende: Unsichtbare Elemente können keinen Hover mehr haben, das bedeutet, sobald die Filler Ebene direkt gehovert wird, wird sie unsichtbar, verliert den Hover und der eigentliche Hover Bereich schrumpft wieder auf die Originalgröße. (grüner Bereich).

Abbildung 21: Detailansicht verschobener gefärbter Filler

9.2 Quiz

Das Konzept des Quiz ist, dass dieses vollständig durch Eingabe im XSLT ohne Programmierkenntnisse konfiguriert werden kann. Dazu werden die einzelnen Elemente des Quiz vom XSLT eingelesen und daraus Javascript Code generiert. Daraus ergibt sich der Vorteil, dass das Javascript für eine Inhaltsänderung des Quiz nicht mehr verändert werden muss.

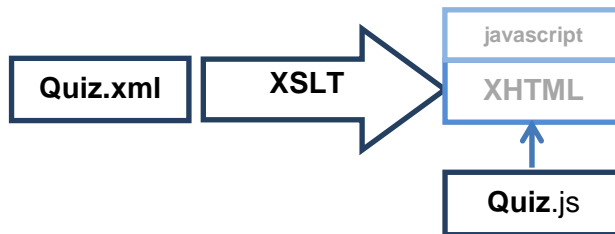


Abbildung 22: Erzeugung der Quiz XHTML

Das hieraus generierte Javascript dient zum Festlegen bestimmter Variablen als Parameter, sowie bestimmte Inhalte, die als Objekte in Literaldarstellung von Arrays erzeugt werden. Objekte dienen als logische Zusammenfassung von verschiedenen Informationen. Folgende Objekte sind von Relevanz:

Question	Rank
Str - Question	Str - Graphic
Str - Answer	Str - Text
Picture - Picture	Int - From
Str[] - Options	Int - To

Folgende XML Datei wird genutzt, um die Fragen zu definieren:

```

<question question="Was machen die da?">
  <picture name="Party" file="quiz_bild_2.png"/>
  <option rightAnswer="true">Junkanoo Karneval</option>
  <option >Ramalam Ding Dong</option>
  <option >Reeperbahn Festival</option>
</question>
  
```

Wird zu:

```

var questionPool = new Array(new question("Was machen die
da?","Junkanoo Karneval",new Array("Junkanoo Karneval","Ramalam Ding
Dong","Reeperbahn Festival"), new Pic-
ture("quiz_bild_2.png","Party")),...)
  
```

Somit besteht das Javascript aus einem generischen Teil, der die reine Verarbeitung der Eingaben übernimmt und einem dynamisch erzeugten Teil, der die XML Daten in Javascript darstellt. Auf diese Weise entfällt ein aufwendiger externer Zugriff, Die folgenden Schritte werden im Javascript durchgeführt (deren genaue Funktionalität ist ebenfalls im Quelltext dokumentiert):

- Auswahl von zufälligen Fragen aus dem Fragenpool

- Bereitstellung eines Formular Objekts zur Verwaltung der Eingaben und Auswertung der Fragen
- Generierung des HTML Quellcodes (Somit wird sichergestellt, dass das Formular nur dann angezeigt wird, wenn Javascript auch aktiviert ist. Es kommt nicht zur Darstellung eines nicht voll funktionsfähigen Formulars)
- Reaktion auf Klicks über spezielle Methoden (z.B. Ankreuzen einer Antwortbox)

Um das Design des Javascriptes so umsetzen zu können, war es notwendig, komplett eigene Objekte mit eigenem Verhalten zu definieren. Es lassen sich Formular Elemente (insbesondere Radio Boxen) nicht in ausreichendem Maße grafisch anpassen.

Um die einzelnen, dynamisch erzeugten Elemente vom Javascript aus im Imperativen Programmierstil ansprechen zu können, werden IDs vergeben, die aus dem Fragetext gebildet werden. Auf diese Weise muss intern keine eigene ID-Vergabe der Objekte realisiert werden. Um dafür zu sorgen, dass die so erzeugten IDs auch einem validen Schema entsprechen (insb. Keine Sonderzeichen, oder Zahlen am Anfang der Zeichenkette) wird eine externe Bibliothek genutzt, um einen „MD5 Hash“⁵ aus einer Zeichenkette zu bilden. Wird diesem Hash noch ein Zeichen (,x') vorangestellt, erfüllt diese immer den Anforderungen an eine gültige ID. Die Entwicklung hat gezeigt, dass ein vollständig objektorientierter Ansatz weniger Aufwand in der eigentlichen Entwicklung, jedoch mehr konzeptionellen Aufwand erfordert.

Die aktuelle Lösung ist praktikabel und für diese Projektgröße vollständig ausreichend.

10 Validität

Ein Ziel war es, dass die Homepage nach der Transformation durch „validator.w3.org“ als valide angesehen wird. Dazu wurden die Ergebnisse jeder Seite validiert. Folgende Auszeichnungen vergab der Validator für jede Seite:

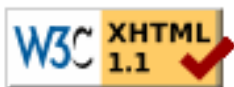


Abbildung 23 XHTML 1.1 valide



Abbildung 24 CSS 2.1 valide: Schachtelung von Div-Elementen

Ebenfalls wurde der XML Quelltext⁶ validiert.

11 Verzeichnisstruktur

Die Verzeichnisstruktur ist in folgender Tabelle beschrieben. Dabei wurden im Dokumentationszweig u.a. einzelne Stadien der Entwicklung abgebildet. Der Zweig „Homepage“ könnte in dieser Form auf einem Webserver liegen.

⁵ Quelle: <http://www.webtoolkit.info/>

⁶ Dazu wurde <http://www.xmlvalidation.com> genutzt

Ordner	Beinhaltet...
/	Stammverzeichnis
/Dokumentation	Dokumentationszweig
/Dokumentation/Plan	Dateien für die Planung der Homepage
/Dokumentation/Quellen	Quellen der Bilder/Inhalte, nach Themen sortiert
/Dokumentation/Screenshots	Screenshots von relevanten Entwicklungsschritten
/Homepage	Inhalt der eigentlichen Homepage, beinhaltet index
/Homepage/css	style.css u. print.css
/Homepage/img	Bilder (aufgeteilt in layout/inhalt)
/Homepage/js	Javascript Dateien (relevant für Quiz)
/Homepage/xml	XML Dateien einzelner Seiten und chapters.xml
/Homepage/xml/xsd	XML Schemata für chapters.xml und Seiten
/Homepage/xml/xslt	Page.xslt um aus XML zu XHTML zu transformieren

12 Fazit

Während der Arbeit an der Website konnten wir viele Erfahrungen im Webdesign sammeln. Nach der Überwindung anfänglicher Einstiegsbarrieren fiel es uns, je mehr wir uns mit der Materie beschäftigten, umso leichter, das gesammelte Wissen miteinander zu verknüpfen und auf diese Weise neue Probleme eigenständig zu bewältigen.

Für zukünftige Projekte haben wir einige Erkenntnisse mitnehmen können:

- Ein Schlüssel zum Erfolg ist es, bereits zu Beginn des Projektes eine Vision zu haben. Sie führt wie ein Roter Faden durch den Erstellungsprozess und kann grobe (Design-)Fehler verhindern.
- Erfahrungen aus diesem Projekt, insbesondere Techniken zur gemeinsamen Aufbereitung eines Projektes werden in Zukunft für ähnliche Projekte nutzbar sein und die Bearbeitungszeit dieser deutlich verkürzen.
- Das Prinzip der wechselseitigen Kontrolle ermöglicht es, den Überblick zu behalten und Detailproblemen offener begegnen zu können.
- Code-Refactoring ist wichtig. Es „reinigt“ nicht nur den Code, sondern eröffnet auch neue Wege und ist für ein dynamisches, iteratives Vorgehen notwendig. Ab einem bestimmten Punkt ist aber auch ein Schlusstrich („Code-Freeze“) zu ziehen.

Das Projekt in Zahlen:

- Ca. 28 Manntage (Berechnet anhand der Aufzeichnungen im Repository)
- 300 Tassen Kaffee (Zählungen aus dem Kaffeeautomaten)
- 1,5 m² Pizza (Rechnungen vom Pizzadienst)

Quellenverzeichnis

- **Layout**
 - Sticky 1:
<http://us.123rf.com/400wm/400/400/tupungato/tupungato1003/tupungato100300158/6685408-green-sticky-note-pinned-to-an-office-notice-board-nurses-wanted--medical-career-opportunity-and-rec.jpg>
 - Sticky2: http://image.yaymicro.com/rz_1210x1210/0/55e/red-sticky-note--you-are-fired---55ee86.jpg
 - Papier: http://www.blackracket.com/Textures/13-229450373_ff8fd8e0ff_o.jpg
 - Flagge:
http://th07.deviantart.net/fs71/PRE/i/2011/092/f/2/bahamas_flag_grunge_by_think0-d3d2fa5.jpg
 - Leder: <http://www.photos-public-domain.com/wp-content/uploads/2011/02/light-brown-leather-texture.jpg>
 - Mappe:
http://2.bp.blogspot.com/_PKrqhojXazM/TNmqq6vIIVI/AAAAAAAAADs/HUY-VFI_ii8/s1600/clodaghs+album+001.jpg
 - Eselsohr: <http://www.designtrax.de/wp-content/uploads/2011/02/nextpage/nextpage-preview-small.jpg>
 - Hintergrund: <http://amazingtextures.com/textures/data/media/14/DSCF0324.jpg>
 - Courier New Schriftmuster:
http://upload.wikimedia.org/wikipedia/commons/9/93/Courier_Bitstream.png
 - Post-It:
<http://us.123rf.com/400wm/400/400/argus456/argus4560810/argus456081000419/3640228-alt-besch-digt-polaroid-auf-wei-em-hintergrund.jpg>
- **Reiseziel**
 - Hotel Atlantic Bahamas: <http://www.thehawaii.org/wp-content/uploads/2011/08/Bahamas-Atlantis.jpg>
 - Strand:
http://de.wikipedia.org/w/index.php?title=Datei:Emerald_bay_great_exuma_bahamas.jpg&filetimestamp=20051011004757
 - Hafen:
<http://de.wikipedia.org/w/index.php?title=Datei:Nassau02.jpg&filetimestamp=20051129143821>
 - Tempel:
http://de.wikipedia.org/w/index.php?title=Datei:Bahamas_Parliament.jpg&filetimestamp=20060515043423
 - Boot: http://www.google.de/imgres?imgurl=http://www.aycalifeclub.com/wp-content/uploads/bahamas-photos-7.jpg&imgrefurl=http://www.aycalifeclub.com/bahamas.html/bahamas-photos-7&usq=__EQjnjbrc-LmbAtTN5ObGFmZR7Uc=&h=798&w=1200&sz=211&hl=de&start=41&zoom=1&id=1WgxmyERR7GDUM:&tbnh=146&tbnw=193&ei=avFfTu99yviyBtHs_JYO&prev=/search%3Fq%3Dbahamas%26um%3D1%26hl%3Dde%26sa%3DN%26rls%3Dcom.microsoft:de:IE-Sear-Box%26rlz%3D1I7ACAW_deDE368%26biw%3D1280%26bih%3D814%26tbm

%3Disch&um=1&itbs=1&biw=1280&bih=814&iact=rc&dur=98&page=3&ndsp=20
&ved=1t:429,r:3,s:41&tx=133&ty=71

- **Kultur**

- Bild 1: http://www.meine-bahamas-reise.com/image-files/Straw_Market.jpg
- Bild 2: <http://www.meine-bahamas-reise.com/image-files/Obeah.jpg>
- Bild 3: <http://www.meine-bahamas-reise.com/image-files/Junkanoo3.jpg>
- Bild 4: <http://www.meine-bahamas-reise.com/image-files/Junkanoo1.jpg>
- Bild 5: <http://upload.wikimedia.org/wikipedia/commons/d/dc/Junkanoo.jpg>

- **Quiz**

- Tourist: http://www.victorialodging.com/files/images/event/tourist_town_victoria.jpg?0
- Handtuch: <http://www.go4celebrity.com/wallpapers/Diane-Kruger/Diane-Kruger-010.jpg>
- Einheimischer: <http://www.solagracia.eu/wp-content/uploads/5-einheimischer-paddelt-uns-zu-austernbank.jpg>

Anhang

Projektplan Homepage

