

Prototyping hoch Drei – interaktiv, dynamisch und konsequent (in der Zusammenarbeit)

Komplexe Industrie-Interfaces mit Prototypen nachhaltig entwickeln, optimieren und umsetzen

Moritz Fröhner

Micromata GmbH
Marie-Calm Str. 1-5
D-34131 Kassel
m.froehner@micromata.com

Julian Mengel

Micromata GmbH
Marie-Calm Str. 1-5
D-34131 Kassel
j.mengel@micromata.com

Sebastian Ammermüller

Micromata GmbH
Marie-Calm Str. 1-5
D-34131 Kassel
s.ammermueller@micromata.com

Michael Kühnel

Micromata GmbH
Marie-Calm Str. 1-5
D-34131 Kassel
m.kuehnel@micromata.com

Abstract

Softwareentwicklungsprojekte im Bereich Industrie-Interfaces haben nicht selten Laufzeiten über mehrere Jahre – von der Konzeption der Informationsarchitektur über die Ausgestaltung des User Interface bis hin zur Realisierung. Parallel werden die Bedienoberflächen von Softwareprodukten immer komplexer und dynamischer in den Interaktionsstrukturen und Bedienmechanismen. Das klassische Entwurfsvorgehen und die damit verbundenen Methoden im User Experience Design stehen vor neuen Herausforderungen. Das Entwerfen über Skizzen und Wireframe-Screens ist je nach Ausrichtung des geplanten User Interface oftmals zu statisch und zu langsam, um die dynamischen Interaktionsstrukturen sichtbar und für den Nutzer oder Kunden (Auftraggeber) erfahrbar und bewertbar zu machen. Neue Ansätze und Technologien im Bereich des Prototyping bieten indes gute Möglichkeiten den Entwurfsprozess für alle Beteiligten zu optimieren und zu beschleunigen.

Keywords

Industrie-Interfaces, Prototyping, Framework, UX-Designprozess, Interdisziplinär, Frontend.

Ausgangssituation und Herausforderung

Das Web hat sich in den letzten zehn Jahren stark verändert. Die Tools, mit denen Webseiten und Applikationen gestaltet werden, haben diesen Wandel sehr viel langsamer vollzogen.

Für die Entwicklung von Webapplikationen hat sich HTML5 als Bindeglied zwischen UI-Designern und Frontend-Entwicklung etabliert und bietet z. B. durch die native Unterstützung von Animationen neue Möglichkeiten in der Gestaltung von Benutzer-Interfaces. Dabei wird nicht mehr in einem statischen Grid und einer festen Breite gearbeitet, sondern mit flexiblen Designs auf einer Vielzahl von Endgeräten mit unterschiedlichen Auflösungen, wodurch die Möglichkeiten und damit die Komplexität von Webapplikationen stetig gestiegen sind. Diese Entwicklungen haben einen direkten Einfluss auf die Arbeit von Usability Professionals und stellen den klassischen Entwurfsprozess vor geänderte und neue Herausforderungen.

Das herkömmliche Entwurfsvorgehen über statische Wireframes stößt in komplexen Industrieprojekten oft an seine Grenzen. Dynamische Navigationsstrukturen und die Funktionen von Off-Canvas-Menüs sowie gesten- oder touchbasierte Interaktionen können dem Kunden und zukünftigem Anwender über statische Wireframe-Screens nur bedingt verdeutlicht werden.

Durch den Paradigmenwechsel zum Responsive Design wird das Entwerfen und Gestalten von Web-Applikationen mit herkömmlichen Entwurfs- und Designwerkzeugen zunehmend ineffizienter. Das Verhalten einer responsiven Bedienoberfläche auf unterschiedlichen Endgeräten in verschiedenen Anwendungskontexten kann dabei oft nur umständlich und mit großem Zeitaufwand veranschaulicht werden.

Seit dem Aufkommen des Material-Design-Ansatzes werden Interfaces nicht nur responsiv, sondern die Übergänge zwischen den einzelnen Kontexten einer Applikation verschwimmen immer häufiger. Animationen verbinden diese Kontexte in weichen Übergängen. Sie werden hier nicht zum Selbstzweck eingesetzt, sondern helfen dem Benutzer sich auf den jeweils veränderten Anwendungskontext zu fokussieren. Die Animation wird damit fester Bestandteil der Benutzerführung und User Experience.

- Interaktive Prototypen auf Basis von HTML5 können hierbei mehrfach Abhilfe schaffen – zum einen bei der Gestaltung der Interaktionsstrukturen und Mikrointeraktionen auf den unterschiedlichen Endgeräten, zum andern bei der frühen Bewertbarkeit und verständlichen Kommunikation der Applikationen mit Nutzern, Kunden und Entwicklern.

UI-Design: Standardvorgehen und Methoden

Heute existieren vielfältige Entwurfsansätze, Vorgehensmodelle und Methoden für das klassische User Interface Design und den nutzerzentrierten Designprozess. Im Bereich der Visualisierung von UI-Entwürfen haben sich einige Standardmethoden etabliert, über die im Folgenden ein kurzer Überblick inklusive deren Vor- und Nachteile aufgezeigt wird.

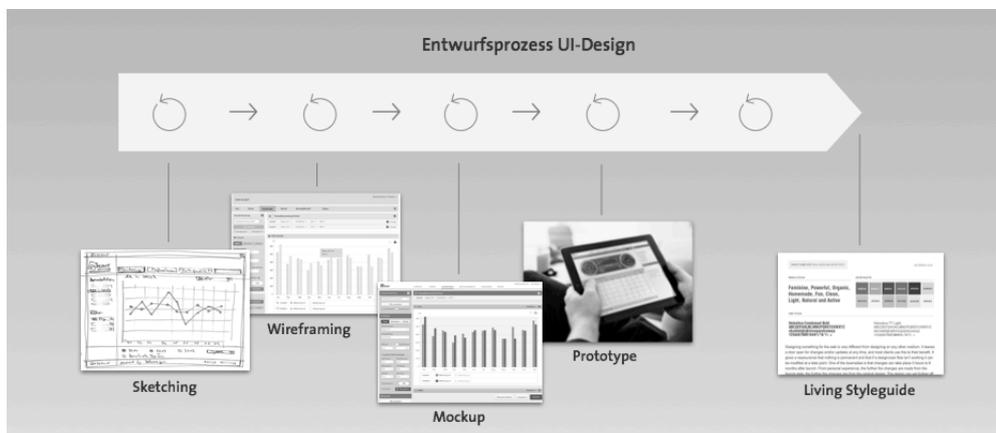


Abbildung 1: Übersicht und Zusammenspiel der Visualisierungsmethoden im klassischen Entwurfsprozess

Sketching – „schnelle Skizzen“

Erste schnelle Pen&Paper-Skizzen dienen der Ideenfindung und Exploration ohne Anspruch auf Konsistenz. Sketching erlaubt es in der frühen Entwurfsphase schnell und einfach Ideen auszutauschen und zu diskutieren. UX-Designer, Projektmanager und Entwickler können so Ideen gezielt erkunden und überprüfen ohne zu viel Zeit und Aufwand zu investieren.

Wireframing – „Flächenbelegung und Klickflows“

Beim Wireframing steht die Benutzerführung und Flächenbelegung im Fokus. In diesem Designschritt werden die grundlegende Informationsarchitektur, das Navigationskonzept, die Funktionen und Verortung der Interaktionselemente sowie die visuelle Struktur der geplanten Anwendung entwickelt und festgelegt. Wireframes bilden die relevanten Anwendungsfälle exemplarisch ab, um Kunden und Stakeholdern eine Rückversicherung zu geben, dass sämtliche Anforderungen abgebildet werden können bzw. berücksichtigt worden sind. Diese Phase dient auch dem Kunden, seine Anforderung auf Vollständigkeit zu überprüfen und hilft die oft komplexen Anforderungsdokumente besser zu verstehen.

Mockups – „Visuelles Prototyping“

Unter Mockups werden im UI-Designprozess visuelle Prototypen bzw. nicht funktionale Designsimulationen verstanden. Mockups oder Design-Mockups verlieren im zielführenden und effektiven Entwurfsprozess zunehmend an Bedeutung. Eingesetzt werden sie vor allem in exemplarischen Designscreens, um sicherzustellen, dass die Corporate-Design-Vorgaben des Kunden (Auftraggeber) korrekt umgesetzt wurden.

Prototypen – „funktional & interaktiv“

Unter Prototypen verstehen wir funktionale, interaktive Prototypen (Low- und High-Fidelity) ohne Backend-Anbindung. Für die zunehmende Interaktivität und die Verwendung von

Animationen als Gestaltungsmittel bietet der Einsatz interaktiver Prototypen eine gute Möglichkeit, Kunden und Nutzern die Funktionalitäten besser verständlich zu machen.

Prototyping mit HTML-Frameworks

Ansatz und Idee

Für viele Kunden und Anwender ist die Bedienoberfläche einer Software die eigentliche Applikation. Durch die Entwicklung mit Prototypen sind beide in der Lage die Applikation mit ihrer Bedienlogik frühzeitig vor der Fertigstellung zu bewerten, zu testen und zu optimieren.

In der Softwareentwicklung versteht man unter dem Begriff Framework eine Sammlung von Konventionen und Vorgaben, eine sogenannte Rahmenstruktur für die Programmierung. Unter dem hier verwendeten Begriff des (CSS-)Frameworks verstehen wir eine auf HTML5 (HTML, JavaScript und CSS3) basierende Sammlung von Gestaltungsvorlagen für Formulare, Buttons, Tabellen, Typografie, Grid-Systeme, Navigations- und weitere Oberflächengestaltungselemente.

Die Einbindung eines solchen Frameworks in den Designprozess bietet die Möglichkeit, in relativ kurzer Zeit mit vertretbarem Aufwand interaktive Prototypen zu erstellen, die den erweiterten UI-Anforderungen in den Punkten Komplexität und Dynamik gerecht werden. Die vorgefertigten Komponenten wie Buttons, Navigationskomponenten oder Formularelemente ermöglichen das schnelle Erreichen von Ergebnissen.

Dabei entsteht ein komplexes Tool-Set an UI-Komponenten, die relativ einfach, schnell und mit wenigen HTML-Kenntnissen zusammengebaut werden können. Auf diesem Weg können sinnvolle Werkzeuge entstehen, die Usability Professionals in die Lage versetzen, erste Designideen und Lösungen schnell und kostengünstig in Prototypen zu überführen und so die Klickpfade einzelner oder aller User Stories für die Projektbeteiligten einfach nutzbar und erfahrbar zu machen.

HTML-Frameworks bieten gegenüber Standard Prototyping Tools u.a. folgende Vorteile.

- HTML-/CSS-Frameworks sind für interdisziplinäre Teams der kleinste gemeinsame Nenner, um gemeinsam eine konsistente Umsetzung der Oberfläche zu erreichen.
- Das entstandene Markup kann im gesamten Prozess wieder- bzw. weiterverwendet werden. Es entstehen keine „Abfallprodukte“ bis zur finalen Implementierung.
- Die zusätzliche Entwicklung eines Design-Style-Guides entfällt, da die visuellen Designdefinitionen direkt in HTML/CSS-basierten Designbibliotheken vorliegen.

Arbeiten mit HTML-Frameworks im UX-Design Prozess

Die Detailarbeit beim Erstellen eines HTML-Prototyps unterscheidet sich abhängig vom gewählten Framework marginal. Die grundsätzliche Vorgehensweise lässt sich jedoch unabhängig vom gewählten Framework in fünf Schritte unterteilen.

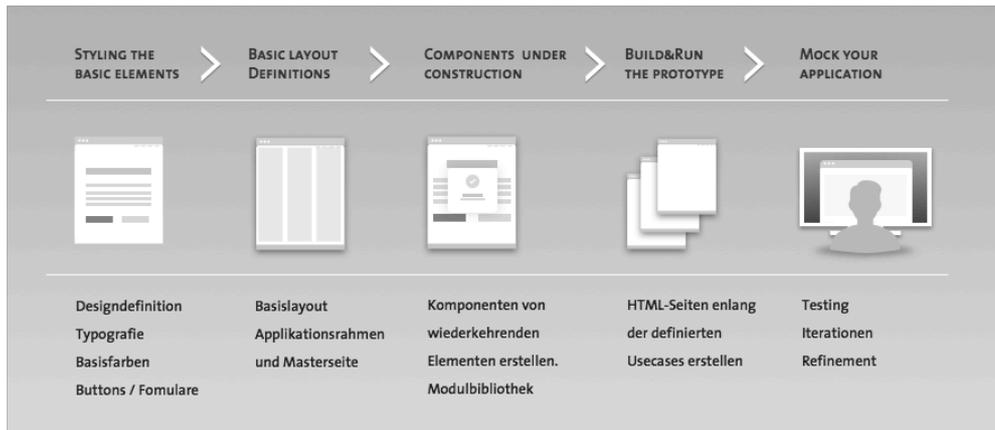


Abbildung 2: Prototyping in HTML-Frameworks – Schritt für Schritt zum funktionalen Prototyp

Schritt 1 – Styling The Basic Elements

Zu Beginn der Arbeit mit einem Framework (wie z. B. Bootstrap) wird im ersten Schritt das grundlegende Aussehen der Applikation bestimmt. Dabei unterstützt Bootstrap das von Luke Wroblewski ausgerufene Paradigma des „Mobile-First“-Ansatzes voll umfänglich. Neben den grundlegenden HTML-Elementen, wie z. B. Typographie, Buttons, Tabellen und Formularen wird auch ein Gestaltungsraster definiert.

Ergebnisse nach diesem Arbeitsschritt sind ein robustes und für mobile Endgeräte optimiertes Gestaltungsraster. Alle Elemente, wie z. B. Typographie, Formulare, Tabellen etc. sind nach den Designvorgaben angepasst ebenso wie sämtliche Basiskomponenten des Frameworks.

Schritt 2 – Basic Layouts Definition

Zur Erstellung eines Prototyps wird mindestens eine Masterseite benötigt, in der die grundlegende Flächennutzung definiert ist. In diesem Template wird der Aufbau der Applikation festgelegt, wie Kopf- und Fußzeile, Navigationsbereich/-strukturen und der eigentliche Inhalts-/Arbeitsbereich. Das Spaltenlayout kann zusätzlich individuell eingerichtet werden. Je nach Detaillierungsgrad der Corporate-Design-Vorgaben und dem Anspruch eines High-Fidelity-Prototypen ist der Umsetzungsaufwand in diesem Bereich unterschiedlich hoch. Die Masterseiten bilden die Vorlage für den späteren Prototyps.

Mit Abschluss dieses Arbeitsschritts ist das Applikationslayout festgelegt. Die Masterseiten für die verschiedenen Applikationsbereiche sind definiert und dienen als „Blaupause“ für die Erstellung des Prototyps.

Schritt 3 – Components under Construction

Wiederkehrende Elemente werden analysiert und daraus HTML-Komponenten erstellt. Typische Beispiele für Komponenten sind z. B. Navigationsleiste, Buttonbars, modale Dialoge, Tabellen, Feedback- und Login-Panel. Je nach verwendetem Framework werden die Basiskomponenten bereits mitgeliefert und sind individuell modifizierbar. Zusätzlich können eigene Komponenten entwickelt werden. Dafür sind erweiterte HTML-Kenntnisse hilfreich.

Nach Schritt 3 stehen die wiederkehrenden Elemente als HTML-Komponenten (Baukasten) zur Verfügung und können für die Erstellung weiterer Seiten kopiert und eingefügt werden. Zudem wurden die Corporate-Design-Vorgaben der Komponenten in das CSS überführt. Durch die Trennung von Code und Layout können Designer und Entwickler parallel am Prototyp weiterarbeiten.

Schritt 4 – Build & run the Prototype

Mit den HTML-Elementen und den Komponenten wird entsprechend den Skizzen und Wireframes das UI nachgebaut. Für jeden Schritt eines Click Flows wird eine eigene HTML-Seite gebaut. Durch die logische Verlinkung der einzelnen Seiten werden die Use Cases durch interaktive Click Flows für den Benutzer erfahrbar gemacht. Die Inhalte sind zu diesem Zeitpunkt noch statisch im HTML-Code integriert und bestehen aus fiktiven Daten.

Mit Fertigstellung dieses Arbeitsschrittes liegen alle Seiten und Dialoge der Applikation als HTML-Seiten vor. Gleichzeitig sind sämtliche Schritte eines Use Cases als einzelne HTML-Seiten vorhanden. Die Bedienlogik der Applikation kann von allen Beteiligten anhand der Click Flows nachvollzogen und bewertet werden.

Schritt 5 – Mock Your Applikation

Die vorhandenen Seiten sind bis auf die Verlinkung einzelner Arbeitsschritte noch ohne dynamische Interaktion. Um das Benutzererlebnis noch stärker an die Realität zu koppeln können die statischen Inhalte durch dynamische Daten ersetzt werden. Dazu werden reale Daten aus zukünftigen Schnittstellen so integriert, als ob diese Schnittstellen bereits bestehen würden. Somit sind weitere Benutzerinteraktionen, wie z.B. die Facettierung eines Suchergebnisses möglich. Dadurch entsteht ein realistisches Look & Feel der zukünftigen Applikation.

Nach dem letzten Arbeitsschritt ist der Prototyp scheinbar voll funktionsfähig und kann von Nutzern getestet und optimiert werden. Die Mikrointeraktionen, die den „Joy of Use“ der Applikation steigern, sind integriert. Alle Projektbeteiligten bekommen ein vollständiges Bild der Funktionsweise der zukünftigen Applikation.

Ausblick – Prototype goes Production

Der final abgestimmte Prototyp kann nun in die Implementierung gegeben werden. Abhängig davon, wie viel Detailarbeit in die Entwicklung des Prototyps investiert wurde, können bis zu 100% des Codes wiederverwendet werden. Neben der Zeitersparnis in der Entwicklung ist der Prototyp ein wichtiges Kommunikationsmittel zwischen UX Professionals und Entwicklern. Die Entwickler können das Verhalten auf unterschiedlichen Endgeräten, die Interaktionsmöglichkeiten und Animationen jederzeit anschauen und ihre eigene Implementierung überprüfen.

Best Practice – Praxisbeispiele für Prototyping

Die folgenden Beispiele geben einen kleinen Überblick über Einsatzmöglichkeiten von HTML-Prototypen im UI-Designprozess. Nicht immer, müssen ganze Applikationen in Prototypen überführt werden, manchmal sind es nur einzelne Details, sogenannte „Microinteractions“ – die durch einen Prototyp verdeutlicht werden können.

Volkswagen Werbebriefprogramm

Das Volkswagen Werbebriefprogramm ist eine Web-Applikation, auf der Volkswagen Partnerbetriebe Aktionsbriefe individualisieren und über einen Lettershop an ihre Kunden verschicken können. Die Anwendung wird kontinuierlich weiterentwickelt - mit steigender Komplexität im User Interface. Die Herausforderung bestand in der Konsolidierung der Informationsarchitektur und Harmonisierung des UI-Designs unter Berücksichtigung der Anforderungen aus Usability Tests und unter Einbeziehung der wichtigsten Stakeholder.

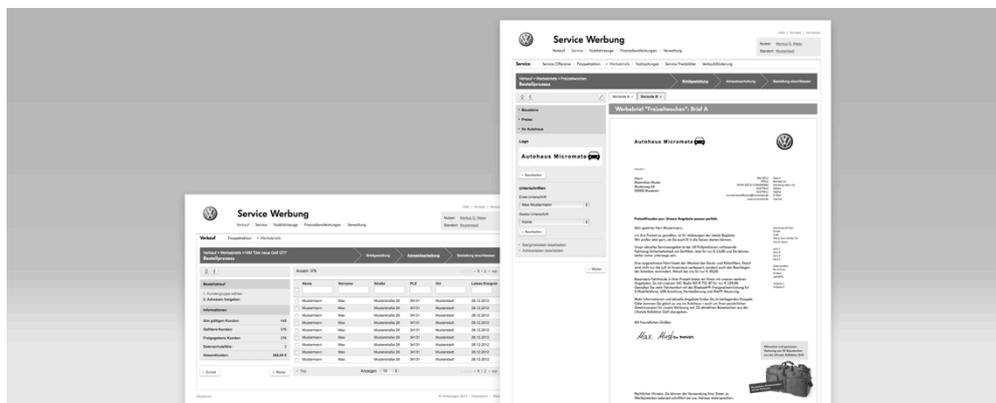


Abbildung 3: Volkswagen Werbebriefprogramm – Prototyping zur kontinuierlichen Weiterentwicklung

Der Prototyp diente dabei als Kommunikationsmittel zur gemeinsamen Abstimmung mit den beteiligten Fachbereichen und weiteren externen Dienstleistern. Die Moderation und Kommunikation im Entwurfsprozess konnte so effektiv und effizient gestaltet werden. Zudem wurde der Prototyp zur Durchführung von User Acceptance Test bereits in der Konzeptionsphase als fester Bestandteil integriert werden.

Bei der Detailierung der Applikation konnte durch den Prototyp die optimale Flächenbelegung einzelner Elemente, wie die Spaltenbezeichnung von Tabellen, auf die eingeschränkte maximale Breite der Anwendung unter realen Bedingungen geprüft und abgestimmt werden.

Zum Ende der Spezifikation entstand so ein High-Fidelity-Prototyp, der alle Funktionen der zukünftigen Anwendung detailliert dokumentiert und als Style Guide fungierte. Dadurch konnte eine hohe Qualität der Corporate Usability der Anwendung zwischen allen beteiligten Dienstleistern erreicht werden.

POLYAS Login

Polyas ist der führende Anbieter für Online-Wahlen in Deutschland. Die gleichnamige SAAS-Software ermöglicht es Unternehmen, Institutionen aber auch Einzelpersonen sichere Online-Wahlen individuell zu konfigurieren und an die eigene Bedürfnisse anzupassen.

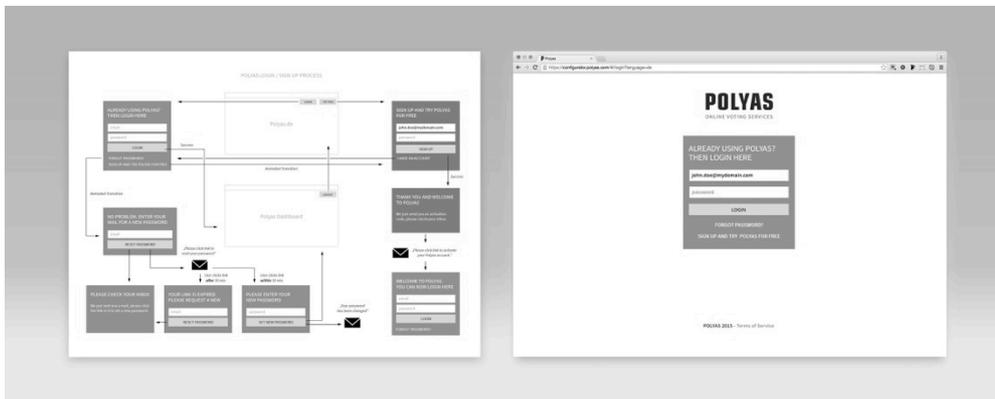


Abbildung 4: Polyas Login – Prototyping zur Gestaltung von Microinteractions

Um den Login- und Registrierungsprozess möglichst einfach und attraktiv zu gestalten, entstand die Idee, beide Formulare nahtlos über eine Animation zu verknüpfen. Die Herausforderung dieses Entwurfs war die exakte Dokumentation der Animationsvorgaben und des genauen Timings der Mikrointeraktionen für die Entwicklung. Die Entscheidung einen Prototyp zu entwickeln, brachte den Vorteil, dass die Animationsvorgaben der UI-Designer nicht aufwändig dokumentiert werden mussten, sondern 1:1 von den Entwicklern in die Applikation übernommen werden konnten.

Kirchlicher Arbeitsplatz ECKD GmbH

Die von der ECKD GmbH entwickelte Software KirA („Kirchlicher Arbeitsplatz“) wird von zirka 3.000 Benutzern in den Kirchengemeinden mehrerer Landeskirchen eingesetzt. Die ECKD GmbH zählt seit mehr als 25 Jahren zu den führenden IT-Dienstleistungsunternehmen für Informationsverarbeitung im kirchlichen Umfeld. Die Herausforderung im Projekt bestand zunächst in der Überführung einer Windows-Applikation in eine mobile-friendly Webapplikation. Zudem sollte eine applikationsübergreifende, konsistente

Benutzerführung und Usability gestaltet und sichergestellt werden unter Berücksichtigung der Nutzergruppe 50+ und deren besonderen Bedürfnisse an die Software-Ergonomie.

Im Rahmen der Umstellung der Anwendung wurde im ersten Schritt eine Experten-Evaluation der Bestandsapplikation durchgeführt. Für die Neuentwicklung standen zusätzlich zwei zentrale Anforderungen im Mittelpunkt: Zum einen sollte die bestehende Ribbon-Navigation, bekannt aus Windows Office-Anwendungen, beibehalten werden, zum anderen sollte die Applikation vollumfänglich auch auf einem Tablet nutzbar sein. Zur Überprüfung des Navigationskonzepts und der Interaktionsmodule wurde ein Prototyp entwickelt, der es den Projektbeteiligten ermöglichte, ein Gefühl für die zukünftige Tablet-Version von KirA zu bekommen und das Look & Feel zu bewerten.



Abbildung 5: KirA – Prototyping zur Anwendungsoptimierung auf Desktop und Tablet und Erstellung „Living Style Guide“

Aus dem Prototyp wurde im Verlauf des Projekts ein „Living Style Guide“ entwickelt. Dieser ist vergleichbar mit einem HTML/CSS-Baukasten, in den UI-Designer und Entwickler wiederverwertbare Elemente einstellen und entnehmen können. Durch das „Baukastenprinzip“ konnte die Konsistenz der Applikation gesteigert und gleichzeitig der Entwicklungsaufwand für das Frontend reduziert werden.

Fazit

Dynamisches Prototyping bietet eine Reihe von Vorteilen bei der Konzeption, Ausgestaltung, Definition und Implementierung von Softwareprodukten insbesondere im Rahmen der Zusammenarbeit mehrerer Disziplinen im Entwurfs- und Entwicklungsprozess. Die folgenden Aspekte können dabei herausgestellt werden.

HTML ist leicht für Nicht-Entwickler zu lernen und ist der kleinste gemeinsame Nenner. Bereits nach wenigen Tagen sind auch Usability Professionals ohne ausgeprägten Entwicklungshintergrund in der Lage, ihre ersten Prototypen umsetzen.

Das dynamische Verhalten von Menüs, Dialogen und Interaktionselementen kann durch Prototypen schneller und besser für Kunden und Nutzer erfassbar und bewertbar gemacht werden. Eine frühe Optimierung der Bedienstrukturen ist das Ergebnis.

Informationsarchitektur, Interaktionsdesign und visuelles Design können parallel und unabhängig voneinander entwickelt werden, das Styling erfolgt unabhängig von der HTML-Struktur durch CSS-Anweisungen. Kollaboration von Design und Entwicklung werden durch die vorgegeben Strukturen erleichtert und fokussiert.

Die Zusammenarbeit von Kunden, UX-Professionals, UI-Designern, Requirement Engineers, Frontend- und Backend-Entwicklern wird durch den Einsatz von Prototypen- Frameworks verbessert, gleichzeitig können spätere Korrekturen oder Anpassungen am User Interface auf ein Minimum reduziert werden.

Der gesamte Entwicklungsprozess von der Idee über die Konzeption und Ausgestaltung des User Interface bis hin zur Umsetzung und Implementierung kann je nach Komplexität und Dynamik der Anwendung deutlich beschleunigt werden – insbesondere bei Industrieprojekten mit vielen Stakeholdern.

Lessons Learned

„Lessons learned“ aus dem frühen Einsatz interaktiver Prototypen im Entwurfsprozess:

[Effektive Planung und Optimierungsmanagement] Die Korrekturen des User Interfaces werden bei interaktiven Prototypen im Projektverlauf nach vorne verlagert und können so schneller und „kostengünstiger“ umgesetzt werden.

[Vermeidung von redundanten Korrekturen] Änderungen an Navigationspunkten oder am „Wording“ führen bei klassischen Prototypen oft zu zeitaufwendigen Änderungen an zahlreichen Screens. In webbasierten Prototypen können diese leicht und automatisiert geändert bzw. im Vorfeld vermieden werden, indem wiederkehrende Elemente nur einmal erstellt aber mehrfach referenziert werden.

[Stärkere Kundenintegration und Partizipation] Für viele Kunden ist das Frontend die eigentliche Anwendung. Kunden und auch Nutzer können durch interaktive Prototypen zu einem frühen Zeitpunkt das finale Produkt erfahren und bewerten.

[Bessere Problemlösung] Viele kleine, oft unscheinbare Usability-Probleme lassen sich an interaktiven Prototypen besser erkennen und darüber hinaus auch besser lösen. Die User Experience der zu entwickelnden Anwendung kann dabei spürbar gesteigert werden.

[Detaillierte und vollumfängliche Spezifikation] Steigerung der Anforderungsdokumentation durch die Parallelisierung von Konzeption und UI-Design durch Usability Professionals und Spezifizierung der Software durch Requirements Engineers.