Template-basierte UI Konzeption durchführen

Bereich
Planung und Design

Aktivität
Lösungskonzept entwerfen

schnell durchführbar ★★☆

einfach durchführbar ★★☆

agil einsetzbar ★★★

Ziele

- Entwicklung einer konsistenten Benutzeroberfläche
- Schnelle und konsistente Umsetzung von Änderungswünschen bezüglich des UI-Designs
- Erreichung einer Verfolgbarkeit von Änderungen im UI-Design
- Schaffung einer Grundlage für eine effektive und effiziente Implementierung von UI-Designs
- Verbesserung der Kommunikation zwischen UI-Designer und Programmierer

Motivation/Problemstellung

Das Aussehen der Benutzungsoberfläche (UI) einer Software-Anwendung unterliegt während der Entwicklung häufigen Änderungen, die prototypisch skizziert und als ausführbarer Code implementiert werden. Diese Änderungen haben nicht nur Einfluss auf einzelne Teile des UIs, sondern können Auswirkungen auf das komplette UI-Konzept haben. Konsistente Änderungen sind in der Regel sehr zeitintensiv, da es sehr aufwendig ist, Konsistenz sicherzustellen. Es muss nicht nur eine potentiell große Anzahl an Prototypen angepasst werden, sondern es müssen auch viele Änderungen am bereits implementierten Code vorgenommen werden. Eine Verfolgbarkeit von prototypisch skizzierten UI-Elementen zu deren Umsetzung in Code und die Verfolgung von Änderungen innerhalb von Prototypen über die Zeit hinweg sind dabei üblicherweise nicht möglich oder werden vernachlässigt. Eine schnelle und konsistente Umsetzung von Änderungswünschen bezüglich des UI-Designs ist dadurch nur schwer möglich. Durch eine fehlende Nachverfolgbarkeit von Code-Implementierungen zu bereits konzipierten Prototypen wird außerdem die Kommunikation zwischen Prototyp-Designer und Code-Entwickler erschwert, die die Grundlage für eine effektive und effiziente Implementierung von UI-Designs ist.

Kurzbeschreibung

Bei der template-basierten UI-Gestaltung geht es um die Erstellung und Verwendung von Vorlagen zur Gestaltung der Benutzeroberfläche. Diese Vorlagen definieren das Grundgerüst einer Anwendung durch instanziierbare UI-Elemente (z.B. Schaltflächen, Tabellen, Header, Navigationsmenüs), die innerhalb der Anwendung zum Einsatz kommen. Eine hierarchische Struktur dieser Elemente wird dabei festgelegt, um zum einen

die Zuordnung einzelner Elemente zu übergeordneten Elementen zu kennen, zum anderen die Einflüsse von Änderungen bestimmter Elemente auf untergeordnete, von ihnen abhängige Elemente von vornherein festzulegen und zu kennen. Weiterhin werden sämtliche Attribute jedes einzelnen Elements spezifiziert, um bei deren Implementierung keine unterspezifizierten oder nichtspezifizierten Aspekte auftreten zu lassen. Einzelne Dialoge der entwickelten Anwendung werden unter (ausschließlicher)Verwendung dieser Vorlagen gestaltet. Ähnlich einem Styleguide werden somit die Elemente für ein einzelnes Projekt, eine Produktlinie oder das gesamte Unternehmen spezifiziert. Durch die Verwendung von gleichen Identifikatioren in UI-Template und Source Code kann zusätzlivh zur Erstellung konsistenter Prototypen, auch die Nachverfolgbarkeit von prototypisch instanziierten UI-Elementen zu ausführbarem Code ermöglicht werden.

Input

- Anforderungen
- Epics
- User Stories

Output

- Templates
- UI-Konzept
- UI-Element-Hierarchie

Rahmenbedingungen

Ausführender
Designer, Entwickler, UI-Engineer, Architekt

Werkzeuge, Hilfsmittel
Papier, Stift

Sketching-Tool

HiFi-Tool

Vorkenntnisse/Erfahrungen
Prototyping-Kenntnisse, Kenntnisse im Bereich Objektorientierung

Ort/Umgebung
Bildschirm-/Büroarbeitsplatz

Weitere Teilnehmer
-

Voraussichtliche Dauer
Ca. 240 Minuten pro Screen

Vorgehensweise

Vorbereitung
Die bereits spezifizierten Anforderungen, vorhandenen Epics und User Stories müssen analysiert werden. Als Ergebnis dieser Analyse erhält man einen Überblick über die gewünschte Funktionalität der zu entwickelnden Software-Anwendung und ggf. auch Hinweise zum zu erstellenden Interaktionskonzept und -design. Aus diesen Informationen lassen sich während der Durchführung dieser Best Practice die benötigten und gewünschten UI-Elemente ableiten.

Durchführung
Legen Sie auf Basis der analysierten Anforderungen, Epics und User Stories zunächst fest, welche UI-Elemente im Verlauf Ihres Projekts verwendet werden sollen. Neben Basis-Elementen wie Text, Schaltfläche oder Eingabefeld sollten Sie vor allem komplexe, zusammengesetzte UI-Elemente festlegen (vgl. Abbildung 1). Zusammengesetzte UI-Elemente sind beispielsweise Navigationsmenüs (zusammengesetzt aus Text, ggf. Grafiken und Schaltflächen), Suchfelder (zusammengesetzt aus einem Eingabefeld mit einer

Schaltfläche, ggf. Label und Grafik) oder ein Header (beispielsweise zusammengesetzt aus Text, Logo, Navigationsmenü und Suchfeld).

Spezifizieren Sie diese Elemente mit Hilfe der Festlegung ihrer Attribute. Dazu gehören beispielsweise die Form, Größe und Farbe des UI-Elements. Überlegen Sie sich anschließend, wie die hierarchische Struktur der Elemente aussieht und wie sich Elemente gegenseitig beeinflussen, welche Elemente also von übergeordneten Elementen beeinflusst werden, welche Attribute sie von diesen erben, welche untergeordneten Elemente sie beeinflussen und welche ihrer Attribute sie an welche untergeordneten Attribute vererben. Ein Suchfeld ist beispielsweise eine spezielle Form eines Eingabefelds, ist hierarchisch also unter dem UI-Element Eingabefeld angeordnet und erbt dessen Attribute. Da ein Suchfeld ein zusammengesetztes UI-Element ist, erbt es weiterhin die Attribute des UI-Elements Schaltfläche und ggf. diejenigen des UI-Elements Grafik.

Zur Identifizierung von benötigten UI-Elementen und zur Erstellung der UI-Element-Hierarchie können Sie zunächst mit der Festlegung des Layouts einzelner Screens oder Dialoge Ihrer Software-Anwendung beginnen. Skizzieren Sie Bereiche für bestimmte UI-Elemente (beispielsweise Navigationsmenü, Inhaltsbereich, Statusleiste) ohne bereits Details der einzelnen UI-Elemente festzulegen. Sie schaffen sich so das erste Template für Ihre Software-Anwendung. Anschließend können Sie für jedes Element in den festgelegten Bereichen dessen Details spezifizieren, beispielsweise aus welchen konkreten Elementen das Navigationsmenü bestehen soll. Weiterhin legen Sie Templates für die spezifizierten Elemente fest, d. h. Sie geben den Attributen der Elemente Werte, legen beispielsweise die Text- und Hintergrundfarbe von Schaltflächen fest und geben ihnen eine bestimmte Form. Ihr Corporate Design ist gut geeignet, um die Attribute zu instanziieren.

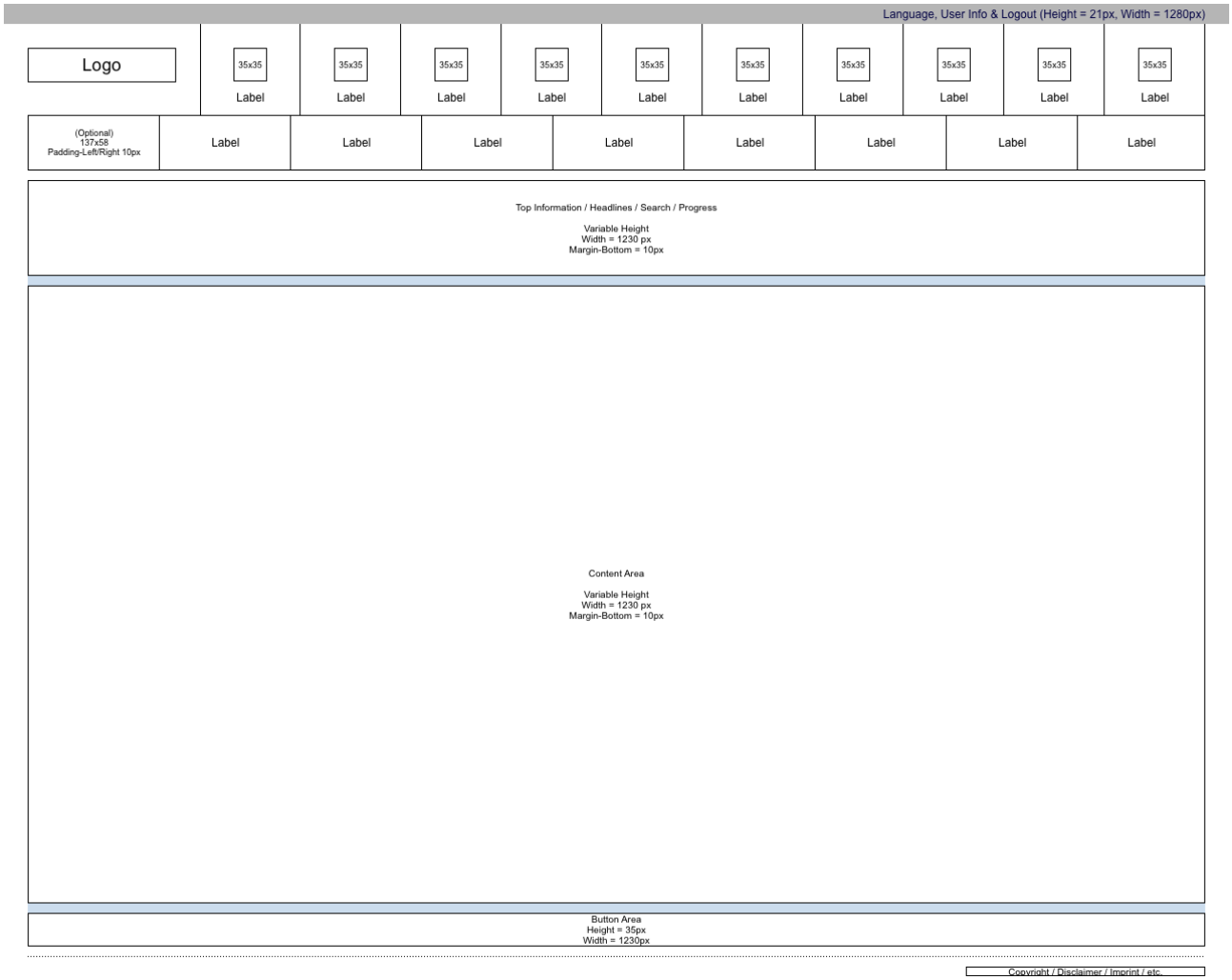

Abbildung 1: Zusammengesetzter Screen bestehend aus Navigation, Header, Content und Footer

Die spezifizierten Elemente können Sie nun verwenden, um Ihre UIs für das aktuelle Projekt prototypisch zu implementieren. Dafür instanziieren Sie die Elemente, d. h. Sie füllen beispielsweise Tabellen mit Inhalt, legen Textinhalte fest, beschriften Schaltflächen, etc. Durch die Templates haben die Elemente über Ihre Software-Anwendung hinweg immer dasselbe Erscheinungsbild und dieselbe Anordnung. Bei der Erstellung von Prototypen müssen Sie sich also nur überlegen, was Sie wo einsetzen möchten, die Ausgestaltung und benötigte Größe der einzelnen Elemente werden bereits durch die Templates geliefert. Die Erstellung der Prototypen geht daher wesentlich schneller vonstatten als ohne einen Template-basierten Ansatz mit festgelegter UI-Element-Hierarchie. Weichen Sie bei der Erstellung des Prototypen vom Template ab, sollte es immer einen berechtigten Grund dafür geben und geprüft werden, ob das Template angepasst werden sollte.

Üblicherweise erstellen Sie eine Vielzahl von Prototypen, um diverse Aspekte des UIs in alternativen Versionen abzustimmen, zu verfeinern oder zu validieren. Versehen Sie jeden Prototypen und jede relevante Gruppe von Elementen innerhalb des Prototyps immer mit einem eindeutigen Identifier (ID), so dass darauf Bezug genommen werden kann und die Kommunikation innerhalb des Entwicklungsteams, insbesondere zwischen UI-Designer und Programmierer erleichtert wird. Weiterhin erleichtert die Vergabe von IDs die Nachverfolgbarkeit der Entstehungsgeschichte bestimmter zusammengesetzter Elemente über Prototypen hinweg. Vergessen Sie daher nicht, anzugeben, auf welchen Prototypen bzw. Elemente neu erstellte Prototypen und instanziierte Elemente basieren bzw. welche

Prototypen und instanziierte Elemente sie erweitern.

Im Lauf eines Projekts können sich Hierarchien oder Attribute von Elementen verändern, so dass die Erstellung von Templates iterativ erfolgt. Bereits implementierte Instanzen der Elemente werden automatisiert angepasst, sobald sich das entsprechende Template ändert, der Inhalt bleibt dabei unberührt. Beispiel: Führen Sie eine Änderung des Layouts einer Tabelle durch, so werden die Layout-Änderung auf sämtliche Anwendungen, die das entsprechende Tabellen-Template verwenden, übertragen. Der Inhalt der ursprünglichen Instanz der Tabelle bleibt erhalten, wird aber auf das neue Template übertragen.

Wird ein neues Element im Lauf eines Projekts wegen der Erhebung neuer Anforderungen oder des Erstellens neuer Epics oder User Stories in eine vorhandene Hierarchie eingebunden und spezifiziert, muss darauf geachtet werden, dass die übrigen Elemente in der korrekten Relation zu dem neuen Element stehen. Dies bedeutet, dass Einflussfaktoren des neuen Elements auf alte Elemente festgelegt werden, Einflüsse alter Elemente wieder richtig zugeordnet werden, wenn das neue Element auf einer Hierarchieebene zwischen zwei alten Hierarchieebenen eingefügt wird und Attribute des neuen Elements auf alte, untergeordnete Elemente vererbt werden und somit Attribute alter Elemente ersetzen.

Fügen Sie schließlich die IDs der instanziierten Elemente Ihrem Quellcode hinzu (beispielsweise als Kommentar im Header). Somit wird die Traceability zwischen den prototypisch gestalteten UI-Elementen und der Implementierung sichergestellt. Die Kommunikation zwischen UI-Designer und Programmierer wird erhöht, da die Templates bzw. UI-Elemente als Diskussionsgrundlage dienen. Hierfür soll der Entwickler stets prüfen, ob für ein zu implementierendes UI-Element bereits ein Template vorliegt. Ist dies nicht der Fall, sollte der UI-Designer das Element zunächst prototypisch instanziieren. Liegt bereits eine prototypische Implementierung des Elements vor, kann sie entsprechend als Code implementiert werden.

Ändern sich Anforderungen an das UI, die Auswirkungen auf diverse bereits erstellte oder sogar als Code implementierte Prototypen haben, müssen nun nicht mehr sämtliche Prototypen überarbeitet werden, sondern nur einzelne Elemente. Ändert sich beispielsweise das Layout des Navigationsmenüs einer Ihrer Anwendungen, so muss nur das entsprechende Navigationsmenü-Element angepasst werden; diese Änderung überträgt sich automatisch auf alle Prototypen, die das Navigationsmenü-Element einsetzen. Der Inhalt des in den Prototypen instanziierten Navigationsmenü-Elements bleibt unberührt und wird auf das neue Layout angepasst, sofern dies möglich ist. Die Möglichkeit dafür hängt von der korrekten Spezifikation der Attribute des Navigationsmenü-Elements und dessen korrekter Einordung in die UI-Element-Hierarchie ab. Die Nachverfolgbarkeit der Änderungen und der Entstehung implementierter Elemente bleibt gewährleistet.

Nachbereitung/Auswertung
Prüfen Sie nach der Erstellung ihrer Templates und Prototypen, ob damit alle Anforderungen an die Gestaltung des UIs erfüllt werden und ob alle UI-Elemente die potentiell wiederverwendet werden könnten als Templates vorliegen.

Gütekriterien/Empfehlungen

Da die Erstellung einer Hierarchie der UI-Elemente zunächst zeitaufwändig ist, eignet sich diese Best Practice vor allem für große Entwicklungsprojekte. Auch wenn die Vorteile der Best Practice auch in kleinen Entwicklungsprojekten greifen, könnte der Aufwand für die

Erstellung der UI-Element-Hierarchie hier nicht in einem günstigen Verhältnis zu dem dadurch erzeugten Nutzen stehen, da ein beträchtlicher Teil des Gesamtbudgets in die Erstellung der Hierarchie fließen könnte. Das Nachziehen von Änderungen und die damit verbundene konsistente Gestaltung von Prototypen ist in kleinen Entwicklungsprojekten noch relativ einfach manuell durchzuführen. In großen Entwicklungsprojekten hingegen ist der Einsatz der Best Practice sehr viel wichtiger, da hier nicht nur mit einer wesentlich höheren Anzahl verwendeter UI-Elemente zu rechnen ist, sondern durch eine größere Anzahl erstellter Prototypen und eine größere Menge implementierten Codes eine Nachverfolgung von Änderungen sowie eine konsistente Erstellung von Prototypen schwer möglich ist. Es empfiehlt sich, in großen Entwicklungsprojekten die UI-Element-Hierarchie toolgestützt zu erstellen, um eine Unterstützung beim Auffinden von Auswirkungen von Änderungen auf bestimmte UI-Elemente und Prototypen zu erfahren, die diese UI-Elemente verwenden.

Risiken

-

Einordnung in den agilen Referenzprozess

Mögliche Vorgänger

- Architektur\Grob- und Detailplanung bei der Implementierung nutzen
- Requirements Engineering\Anforderungen kontinuierlich priorisieren
- Requirements Engineering\Anforderungen mit Hilfe von Prototypen erheben
- Requirements Engineering\Anforderungen reviewen
- Requirements Engineering\Experimentelles und exploratives Prototyping durchführen
- User Experience\Informationsarchitektur erstellen

Mögliche Nachfolger

- Requirements Engineering\Anforderungen mit Hilfe von Prototypen erheben

Mögliche Alternativen, verwandte Praktiken

-

Einordnung in das PQ4Agile-Qualitätsmodell

Beeinflusste Qualitätsmerkmale und -teilmerkmale: Wartbarkeit (insbesondere Modularität und Wiederverwendbarkeit), Anpassbarkeit und Gebrauchstauglichkeit

Schlagworte

Benutzungsschnittstelle, UI, Gestaltung, Interaktionsgestaltung, Interaction Design, Prototyping, Sketching, Template

Weiterführende Informationen

Informationen im Internet

-

Literatur

-

Das Projekt PQ4Agile wird vom Bundesministerium für Bildung und Forschung im Rahmen der Maßnahme KMU-innovativ: IKT (01|S13032) gefördert.