Kompendium UX-Reminder

Inhaltsverzeichnis

[Fragen für das Rebriefing an den Kunden 2](#_Toc472088925)

[Grundsätze und Leitideen, die für die Konzeption bearbeitet/angedacht/gelöst werden sollten 5](#_Toc472088926)

[Project\_Canvas 6](#_Toc472088927)

[Konzeptphasen 8](#_Toc472088928)

[Wie legt man (in Axure) Wireframes an? 9](#_Toc472088929)

[Seitenbenennung in Axure 11](#_Toc472088930)

[Patternlibrary 12](#_Toc472088931)

[(Beispielhafte) Responsive Strategie 15](#_Toc472088932)

**Kandidaten für zukünftige Blogposts sind rot**

# Fragen für das Rebriefing an den Kunden

Teil 1: Der Briefing-Workshop

* Allgemeine Fragen zum Unternehmen des Kunde n:
  + In welchen Geschäftsfeldern ist das Unternehmen des Kunden tätig?
  + Positionierung des Unternehmens/der Produkte, Selbstverständnis
  + In welchem regionalen, nationalen oder internationalen geografischen Umfeld ist das Unternehmen aktiv?
  + Gibt es weitere Standorte/Filialen/Töchter/Mutterunternehmen des Unternehmens?
  + Wer sind die Konkurrenten des Unternehmens, wer die Marktführer?
  + Wer sind die Vorbilder/Leitbilder des Unternehmens?
  + Gibt es eine eigene Marketing-Abteilung im Unternehmen, die an der Konzeption des Webauftritts beteiligt ist?
  + Gibt es eine eigene EDV-Abteilung im Unternehmen, die an der Konzeption des Webauftritts beteiligt ist?
  + Welche Browser werden benutzt?
  + Wie viele Mitarbeiter hat das Unternehmen?
  + Wer sind die verantwortlichen Ansprechpartner/innen für das geplante Projekt?
* Fragen zur geplanten Web-Präsenz:
  + Wann soll Online-Start sein? Warum gerade zu diesem Zeitpunkt?
  + Wie ist die geplante Lebensdauer der neuen Web-Präsenz?
  + URL des Kunden, reservierte Domains vorhanden?
  + Gibt es bereits einen Provider, Hosting-Partner?
  + Stärken und Schwächen der bisherigen Site?
  + Ist die neue Website wesentlicher Bestandteil eines bestehenden Marketing-Mixes oder eine neue Strategie oder stand-alone?
  + Welche Multimedia-Projekte wurden schon mit wem und wann umgesetzt?
  + Umfang des bisherigen Webauftritts?
* URLs von Wettbewerbern?
* URLs von Vorbildern?
* Welche Ziele verfolgt das Unternehmen mit der geplanten Web-Präsenz?
  + Imagekommunikation, Transaktionen, Abverkauf, Kundengewinnung, Kundenbindung, Marketing, PR, Produktwerbung/Preisinformation, Entlastung des Supports, Rationalisierung, Web-Community, E-Commerce, Kampagnenbegleitung/Merchandising, Downloads, Online Magazin mit redaktionellen Inhalten, ...
* Mit welchen Inhalten und Features sollen die Ziele erreicht werden?
  + Selbstdarstellung des Unternehmens, Diskussionsrahmen für Kunden, Kontaktmöglichkeiten, Pressemeldungen/Veröffentlichungen, Kundensupport, Produkt-Präsentation, E-Commerce, Fun, Online Magazin, ...
* Messgrößen für den Erfolg?
  + Anzahl PIs/Visits, Downloads, Kontakte, verkaufte Produkte, Umzatzsteigerung, abgeschlossene Bestellvorgänge, sonstige Transaktionsanzahlen...
* Welche Zielgruppen sollen erreicht werden?
  + Vorhandene Kunden, B2B, Presse, Neukunden, Mitarbeiter/innen, Bewerber, Investoren, Consumer (demografische Definition), möglichst alle, ...
* In welcher Frequenz sollen die Inhalte der Website aktualisiert werden?
* Wer soll die Aktualisierungen durchführen?
* Ist ein Grobkonzept bereits vorhanden?
* Welche Vorgaben existieren?
  + Logo, Hausschrift/-farbe, CD, Technologie, Sicherheit ...
* Weitere Rahmenbedingungen
* Welches Material ist vorhanden? Sind die Rechte geklärt?
  + Content, Bilder, Texte, Kataloge/Broschüren, Audio/Video, ...
* Welche Features sind unbedingt gewünscht?
* Audio/Video, News, Chat, Kontaktformulare, Animationen, Spiele, Shop, Newsletter, WebCam, Volltextsuche, Gästebuch, passwortgeschützer Bereich, Dark Pages, Voting, ....
* Mehrsprachigkeit/mehrere Länder?
* Eingesetzte/geplante Hard- und Software?
* Budget für Realisierung?
* Sieht man die neue Website im Brot und Butter- oder im Premium-Segment? (Abgleich mit Budgetvorstellungen!)
* Budget für Hosting und Pflege?
* Personalressourcen während des Projekts?
* Personalressourcen in der Pflegephase?
* Stand des Projektes (beschlossen, in Vorbereitung, zweiter Versuch etc.)?
* Ist ein Zeit-/Projektplan vorhanden?

Teil 2: Der Kickoff-Workshop mit Kunden

* Welche der besprochenen Inhalte haben höchste Priorität?
* Gibt es Dritte, die Inhalte liefern bzw. pflegen sollen oder müssen? Wer sind die Ansprechpartner?
* Welcher Teilprojekte/Contents werden vom Kunden selber realisiert?
* Wird es Banner-Werbung/eine Kampagne für die Promotion des Relaunchs geben? Ist diese von der Agentur zu entwickeln/zu schalten? Gibt es sonstige Relaunch-Kommunikationsmaßnahmen? Wie werden die User informiert und abgeholt?
* Wer liefert welche statischen Texte/Bilder? Wer ist hierfür Ansprechperson? Wird Texter- oder Grafikerleistung von der Agentur benötigt?
* In welchen Formaten werden Bilder und Texte angeliefert werden? Die Agentur geht davon aus, dass beides immer internetgerecht aufzuarbeiten ist und kalkuliert entsprechend.
* AGBs, Privacy Policy etc: die rechtliche Absicherung der Inhalte und Transaktionen müssen vom Kunden kommen!
* Mobile Vorstellungen?
* Gibt es Browsertechniken die anzuwenden/ nicht anzuwenden sind (Frames, Popups, Javascript, Flash, ActiveX, Java, Director...)?
* Welche Statistiken sind gefordert? In welchem Zeitintervall? Welche Aufbereitung?
* Welche Interaktion mit externen Systemen soll erfolgen?
* Welche Skalierbarkeit ist gefordert, welche Last muss das System im Durchschnitt/im Peak aushalten? Welche Lasttests sind durchzuführen? Was kann Peaks erzeugen? Welche Verfügbarkeit soll gewährleistet werden?
* Welche eigentlich statischen Seiten sollen pflegbar sein und benötigen somit Schnittstellen?
* Gibt es Ziele beim Suchmaschinenranking?
* Müssen die Seiten Traffic -optimiert werden (Verwendung von minimiertem
* HTML)?
* Barrierefreiheit?
* Welche Seiten müssen suchmaschinenoptimiert sein, welche bookmarkbar?
* Wer liefert die Inhalte der META-Tags?
* Bei CMS: Welche Funktionen müssen möglich sein, wie kompliziert darf die Bedienung sein? Werden die Bediener Aushilfskräfte oder Experten sein?
* Milestones und Zeitplan
* Ist Bannerplatz auf der Site einzukalkulieren? Welche Seiten, Positionen, wie viel,
* wie groß, wie oft?

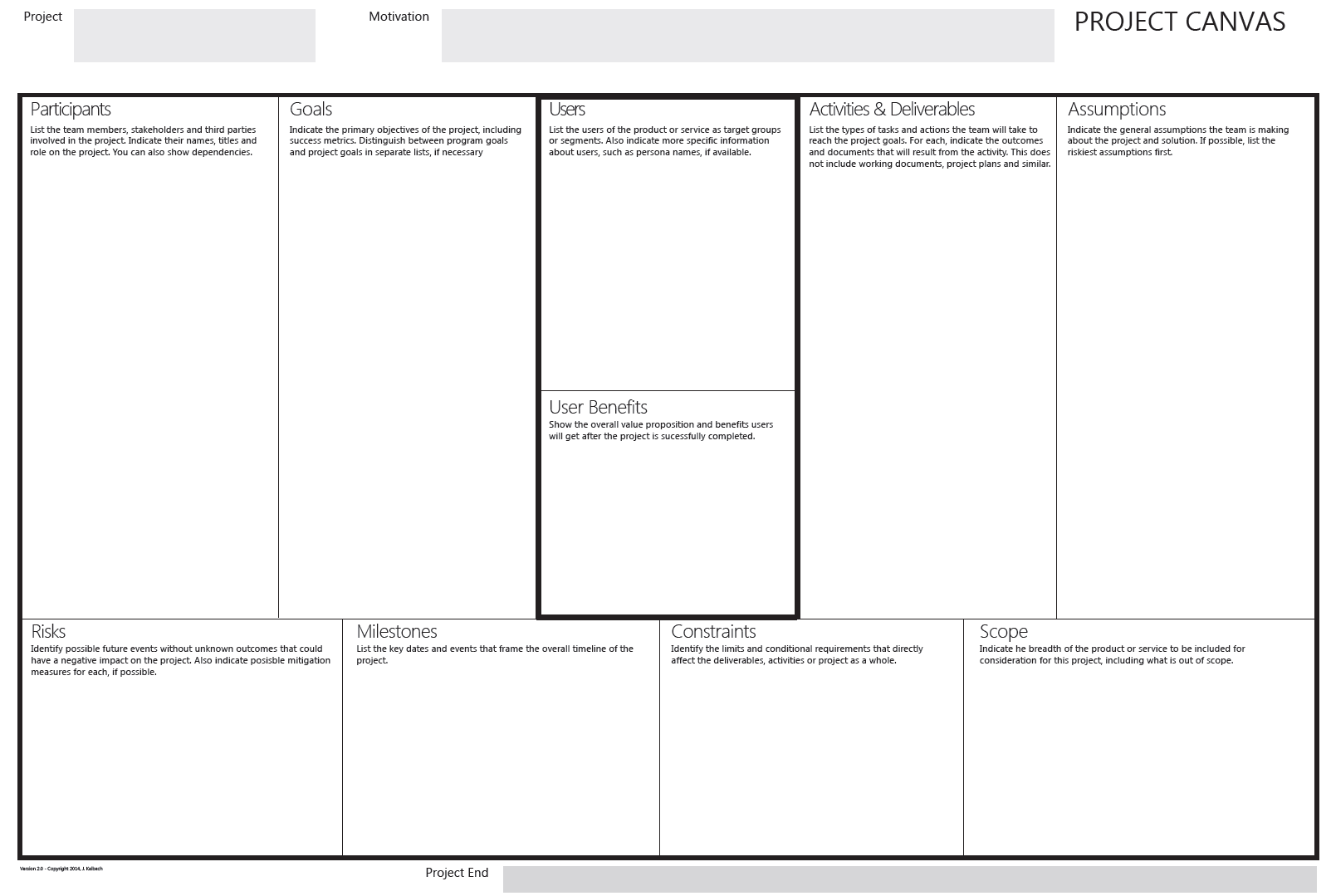
# Grundsätze und Leitideen, die für die Konzeption bearbeitet/angedacht/gelöst werden sollten

* Unser Anspruch an die Vision (die Website) - z.B. eine oder mehrere der folgenden Fragen:
  + Main Idea
  + Auf die Website zu gehen bedeutet...
  + Sie macht folgenden Eindruck...
  + Unser Anspruch ist...
  + Die Website ist...
  + Im Vergleich zum Wettbewerb haben wir...
  + Unser Vorbild ist...
  + Unser Negativbeispiel ist…
  + Folgende Adjektive passen...
* Look & Feel
  + Emotionen?
  + Begeisterung (wodurch?)
  + Joy of Use (Stellenwert)
  + Onepager vs. Tabs vs. Einzelseiten vs. Akkordeon-Box
  + großzügig vs. kleinteilig
  + Tonality, Siezen/Duzen, Ihre/Meine/Neutral
  + Wieviel passt auf eine Seite? Ungefähre Seitenbreite der Vision in px (800/1000/1200/responsiv/fluid?) - Bevorzugen wir große oder kleine Schriften, wo ist der Fold?
  + Raster/Spaltigkeit der Wireframes (=Raster/Spaltigkeit der Website?)
* Fachliches Vorgehen
  + Welche Research-Daten haben wir, was brauchen wir?
  + Personas/Mindsets?
  + Welche Annahmen treffen wir (ggf. absegnen lassen)
  + Welche zu überprüfenden Hypothesen stellen wir auf?
  + Mobile first? Mobile Strategie (responsive, adaptive, eigene Website)
  + No frills / MVP-Ansatz vs. maximale Feature-Dichte
  + Einfachheit vs. Schnelligkeit vs. Vollständigkeit (der Lösung, nicht der Dokumentation)
  + Wie können/werden wir validieren und testen? Wie dokumentieren?
  + Wie/wo unterscheiden oder dokumentieren wir must-have und nice-to-have Features?
  + Welche Edge-Cases müssen berücksichtigt werden?
  + Welche Prozesse und Zustände gibt es?
  + Gibt es Rollen&Rechte?
  + Gibt es einen Power-User Modus?
  + Welche Dokumentation brauchen wir? Lean-UX?
  + One Site fits all oder 2-4 verschiedene Portale für die Personas (Entscheidung ist auch visionärer Natur und nicht nur von den Anforderungen der einzelnen Personas abhängig)
* Wireframe-Formalien siehe Kapitel „Wie (in Axure) Wireframes anlegen?“
* Konzeption
  + Patternlibrary erstellen (u.a. für Widgets/Module) siehe Kapitel „Patternlibrary“
  + Templates/Atomic Design
  + Checkout ohne Navigation?
  + Zielgruppen? (z.B. Alter 25 vs. 50, Männer vs. Frauen, Bildungsgrad, Online-Affinität)
  + Responsives Verhalten/Spalten
  + Stickyness
  + Drag&Drop
  + Informationsarchitektur, Informationstiefe, Navigationsprinzip (IA), Navigationsmechanik
  + Unterschiedliche Zugangsmöglichkeiten
  + Berücksichtigung von SEO
  + Einbindung und Verlinkung Web-Universum des Kunden / Kampagnen
  + Registrierung/Accountmigraion/Trial-Nutzung/Unverifizierte Nutzung
  + Berücksichtigung Customer Lifecycle
  + User Journey
  + Welche Spezifikation/Dokumentation wird benötigt?
* Inhalte
  + Storytelling
  + Kontakt & Beratung / Hilfe
  + Inspiration
  + Mehrfachverwertung von Contents (möglich im CMS?) / How-To Datenbank / FAQ
  + Verwendung/Migration Altcontents?
  + Research Offline Purchase Online
  + Loyalty-Programm
  + Personalisierung/Targeting
  + Recommendations (automatisch oder Bewertungen)
  + Cross-Selling
  + Trust
  + SoMe
  + Suche und Suchergebnis
* Organisaorisches
  + Wer sind die Stakeholder
  + Ziele des Kunden (offen kommuniziert, vermeintlich, Agenda, noch nicht entdeckt oder benannt)
  + Zielerreichung messbar machen?
  + Deadlines und Milestones (offen/verdeckt)
  + Dokumentation
  + Abnahmeprozess
  + Team und Verfügbarkeit, PTs

## Project\_Canvas

Warum machen wir das Projekt, was sind die Rahmenbedingungen und Annahmen:

<http://experiencinginformation.wordpress.com/2012/08/05/the-project-canvas/>



# Konzeptphasen

1. Einarbeitung, Scopeklärung, Benchmarking, Messkriterienerstellung, Machbarkeitscheck grob
2. Kundenbedürfnisse verstehen, Research, Personas
3. Zieldefinition, Lösungspitch, High-Level-Anforderungen, Vision, Service Design, Design Thinking
4. Grobkonzeption, Lösungsfindung, Machbarkeitscheck fein, Methoden, Journeys
5. Konzeptvalidierung, Prototyping, Testing
6. Feinkonzeption
7. Entwicklung/QS
8. Ausrollen, Messen, Verbessern, Iterieren

# Wie legt man (in Axure) Wireframes an?

Definition der Leserschaft: UX’ler, Entwickler inhouse/extern, Management, Spezifikation, Ideenabstimmung, Usability-Test. Danach Entscheidung der Granularität und vor allem des Detailgrads.

Definition des Ziels: Spezifikation, Klickdummy für Spec, Klickdummy für Abstimmung, Klickdummy für Usability-Test. Wenn Spezifikation, dann Format überlegen: Word oder HTML mit Annotations auf AxShare?

Eine Übersichtsseite mit allen Atomen, Molekülen, Organismen und Pattern soll deren richtige /exemplarische Verwendung zeigen. Evtl. Do’s und Don’ts dort beschreiben.

Entscheidung, ob Responsiveness-Feature von Axure genutzt oder mobile Varianten autark angelegt werden.

Mobile Seiten direkt (z.B. als Variante) an die Desktop-Variante hängen - sonst ist bei Änderungen die Sucherei zu groß. Oder als Axure-Referenz verlinken. Allerdings ist beim separaten Rausrendern dann mehr zu klicken.

Eine verlinkte Sitemap erstellen, ggf. verlinktes Flowchart, Fertigstellungsgrad an Flowchart verzeichnen.

Prozesse als Flows/Übersichtscharts aufzeichnen und mit Screens verlinken, ggf. verkleinerte Axure-Snapshots - Abfolgen von Screens alleine sind unverständlich und decken nicht alle Edge-Cases ab

Axure-Master verwenden, ggf. Atomic Design. Widget Library nur für ganz allgemeine Elemente, Projektübergreifendes, da diese bei Änderungen an der Library nicht upgedatet werden. Dynamic Panels eher nicht verwenden. Nur für Interaktionen.

Blindtext: Lorem Ipsum vs. horizontale Striche oder nur Pre-Echttext?

Wenn die Wireframes in einem sehr frühen Stadium sind und hauptsächlich noch der Abstimmung und Ideenfindung dienen, kann man überlegen, ob man

* die Seitennamen auf den Screen schreibt
* Varianten auf der gleichen Seite nebeneinander darstellt
* Annotations mit Callouts direkt auf den Screen schreibt
* Mit einem (unsichtbaren und überall gleich positionierten) weißen Fleck auf weißem Grund für alle Ausdrucke die gleiche Zoom-Stufe herstellt
* Sketchiness nutzt
* Grundsätzlich noch zu Klärendes und offene Fragen eher als Text auf Screens/in Callouts schreiben. In Annotations bleibt das relativ unscheinbar und kann übersehen werden.

Dies führt dazu, dass man die Screens für Workshops gut ausdrucken, an die Wand hängen und diskutieren kann. Die Nutzung für einen interaktiven Test ist dann aber mit den ganzen sichtbaren Anmerkungen schlecht. Hierfür blendet man alle Anmerkungen aus. Eine globale Variable wird gesetzt ("testmode"), die auf den Testmodus hinweist. Alle Anmerkungen werden dann mit einer onLoad-Action ausgestattet, die die globale Variable überprüft und sich ggf. dann selber versteckt (OnLoad: if onloadvariable equals "testmode" hide this()).

Üblicherweise sollte das Zielformat in der finalen Phase ein HTML-Klickdummy sein (in Ausnahmefällen Word-Dokumentation) und die Standard-Axure-Annotations für Elemente und Seiten sollten genutzt werden. Annotationen gruppieren!

Ein Grid nutzen, um auf allen Screens angeglichene Positionierungen zu erreichen. Gerne auch horizontale Abstände definieren (die einzuhalten wegen mangelndem Grid aber Aufwand bedeutet, den man sich nicht unbedingt machen muss).

Spaltigkeit und responsives Verhalten sollten möglichst früh geklärt werden, sonst weichen Wireframes zu stark von späteren Designs ab und der Abstraktionsgrad für die Bearbeiter bzw. Leser ist hoch.

Wireframes ohne Gründzüge des späteren Designs zu erstellen bedeutet, dass entweder das spätere Design stark eingeschränkt wird, da es zu den Wireframes, die vorher erstellt wurden, passen muss, oder, dass die Wireframes überhaupt nicht zum Design passen können (Breite, Spaltigkeit, Schriftgröße, Navigationsanordnung). Was eine intuitive Übersetzung von Wireframe in Design extrem erschwert.

Möglichst oft Custom Styles verwenden, damit Änderungen einfach sind. Möglichst wenig verschiedene Custom Syles verwenden.

Die Seitennavigation (und der Footer) sollte auf allen Seiten gezeigt werden, aber nur grob vereinfacht als Master, damit man nicht immer den korrekten Aktiv-Zustand zeigen muss und keine Arbeit hat, wenn sich die echte Navigation in der Höhe ändert.

Sprache festlegen. Wenn englischsprachige Leser zu erwarten sind, dann gleich in Englisch aufsetzen. Trifft oft auf Programmierer von Fremdfirmen zu.

HTML Buttons in Axure werden nur für Formulare verwendet - wenn überhaupt, da sie nur eingeschränkt gestyled werden können.

Wie Varianten bzw. unterschiedliche Zustände von Screens anlegen, die sich nur leicht voneinander unterscheiden?

* Klickbare (interaktive) HTML-Varianten auf einer Seite bauen (ggf. mit Dynamic Panels und einer nebenstehender Klickanleitung, um alle Varianten im Preview-Mode sehen zu können)
* Alles auf Einzelseiten bauen (ggf. mit Snapshot, auf dem dann nur ein Teil mit der Variante überlagert wird)
* Alles nebeneinander auf einer Axure-Seite zeigen (ggf. nur die abweichenden Teile darstellen)

Alle diese 3 Möglichkeiten haben Vor- und Nachteile, je nachdem, wie oft man es ändern muss, wer es lesen soll, ob man als finalen Output einen Klickdummy oder eine Word-Spezifikation braucht…

## Seitenbenennung in Axure

Seiten sinnvoll benennen und mit System nummerieren. Bei den Nummern kann man mit 10er-Blöcken arbeiten, damit man einfach noch Seiten dazwischen bekommt.

Beispiel für Minimalvariante: 2.1\_Produktdetail

Beispiel für einfache Variante: PROD\_20.10\_Produktdetail

Beispiel für ausführliche Variante: W\_PROD\_20.10.70\_a\_LI\_Produktdetail

(==> FrontendWeb\_BereichPRODukte\_SeitennumerMitZehnerblöcken\_ZustandA\_Status:LoggedIn\_BeschreibenderSeitenname)

Frontend, Zustand, Status werden nur verwendet, wenn es im jeweiligen Projekt Sinn macht.

Auf Sortierbarkeit der Seitennamen wird verzichtet, weil dann führende Nullen und einheitliche Ziffernblöcke (01.02.00.00) notwendig würden, die dann wieder in der Anzahl der Seiten unflexibel wären.

Variante einer Seite, über die noch entschieden werden soll: W\_PROD\_20.10.70\_VARIANTE\_Produktdetail

# Patternlibrary

Für wiederkehrende Elemente werden Grundsätze festgelegt und Beispiele in Axure abelegt.

Beispielhafte Ausprägung ist grau.

* Tonality  
  "Sie" + formal. „Ihre“... / „Meine“ (... oder neutral?)
* Auszeichnung der Linkabstufungen (CTA/1./2./3. Ebene)
* Breadcrumb/Brotkrume  
  oben-links, 1. Ebene (Produktübersicht): nein, ab 2. Ebene (Wartung): "<< Produktübersicht | Wartung".  
  Ausnahme: Infoseiten, die keine Modalfenster sind (z.B. im FAQ-Bereich angesiedelte Seiten) haben History-back "< zurück"-Buttons, wenn sie nicht aus den FAQ aus aufgerufen werden.  
  Seiten, die aus einem fremden Bereich angesprungen werden (z.B. Shop-Updates, die aus der Produktwartung angesprungen werden) haben ebenfalls History-back "< zurück"-Button statt der Brotkrume. wie flexibel ist die Programmierung?
* "Nach oben"-Link  
  Auf Desktop nein, auf Tablet/Mobile evtl. als HTML-Back-To-Top Floatbutton (TBD - wie lang ist eigentlich die längste Seite?)
* Navigation  
  2. und 3. Navigationsebene ausklappbar? , Mega-Flyouts/Hamburger?, Hauptmenüpunkte verlinkt oder nicht?
* Wizard/schrittweiser Prozess   
  Progress-Bar, Content zentriert, vor/zurück, statt Breadcrumb. Wizard mit "Abbrechen" verlassen
* Lange Listen  
  Keine Verwendung von "Frames mit eigenen Scrollbalken" innerhalb von Seiten wg. schwieriger mobile-Umsetzung; jede lange Liste muss eine eigene Seite haben
* Ankernavigation auf Sub-Contents einer Seite  
  Bis jetzt nicht geplant
* Button-Feedback: Hover-, Mousedown-, Klick-/Touch-Effekte, CTA-/Synchronisierungsfeedback  
  Ja, möglichst auf jedem interaktiven Element
* Latenzzeiten-Feedback/Widget-Ladezeit-Anzeiger  
  Lade-Spinner + ggf. abgesoftetes Design für Content
* Fehlermeldungen (Formular + anderswo (z.B. CTA-Klick)):
  + Formular-Inlinevalidierung  
     Ja, hinter/unter dem Feld
  + Sync-Fehler  
    s. CTA-/Synchronisierungsfeedback
  + Normaler Formularfehler nach Absenden  
    oben in rot – oder am Button?
  + Berechtigungsfehler/sonstiger Fehler  
    TBD
* Tooltips  
  Info-Layer auf "i" (Klick oder Hover) bei Desktop, auf Mobile als Modalfenster. Content sollte kurz sein, so dass Scrollen nie nötig wird
* FAQ  
  Als eigenständige Sektion (mit Suche?) und kontextbezogen auf passenden Seiten
* Stickyness  
  Wo nötig (To-Top-Link, evtl. den Header verkleinern beim Scrollen?)
* Modale Fenster/Lightboxen (wann, wie, wie lang, wie mobil?)  
  Ja, als Feedback bei "wichtigen" Prozessen, evtl. für "Zusatzinformationen". Weitere TBD
* Drag&Drop (mobil?)  
  Nein!
* Mobile Tabellen mit mehr als 3 Spalten (normale Tabellen, Vergleichstabellen, Diagramme (ggf. interaktiv)  
  Darstellung wie bei Desktop. Tabellenaufbau bleibt mobil unverändert, kann horizontal im Viewport verschoben werden. Evtl. öffnen in modalem Fenster .  
  Theoretische Möglichkeiten: Bild zoombar, Tabelle horizontal verschiebbar, mobiler Umbau (immer 2-spaltig mit Wiederholung der Spaltentitel)
* Verwendung von Templates  
  Wir haben evtl. sich ähnelnde Seiten, entwickeln aber nicht anhand von Templates, jede Seite individuell
* Formulare  
  Inline-Editing + sticky Hinweis, dass noch gespeichert werden muss, der nach erster Änderung z.B. am Viewport-Bottom erscheint - auch wenn nur Toggle-Buttons auf der Seite sein sollten! Es gibt keine Abbrechen-Buttons zum Rückgängig-machen von versehentlichen Formular-Änderungen. Oder doch? Warum eigentlich nicht?  
  Sonderform: Initial leere Formulare haben diesen Hinweis nicht  
  Ausnahme: Upload von Bildern muss NICHT explizit gespeichert werden, sondern ist nach Upload automatisch erfolgt
  + Inline-Validierung  
    Ja
  + Feldbeschriftung  
    links, mobile oben
  + Felderklärung  
    Nicht als eigene Spalte, ggf. Infos-i's
  + Speichern / CTA  
    Ja, unten rechts
  + Abbrechen  
    Ja, links neben dem CTA
* Microtransitions  
  Grundsätzlich ja, wo sinnvoll. s. Button-Feedback, Ladeverzögerung, ggf. Wizard-Slider, Message-Center: Slide-In Effekt
* Paginierungsmechanik / Lazy Loading  
  Kein klassisches Lazy Loading (on-scroll-Loading). Evtl. "Weiter>"-, "1|2|3" oder "Mehr sehen"-Button
* Bestätigung von kritischen Aktionen (Account löschen etc.)  
  Modalfenster!
* Buttonbeschriftungen  
  Sprechende Beschriftungen ohne separate Frage (=längere Buttons) oder separate Frage + kurze Ja/Nein-Buttons? Formulierungen wie "Zum Formular" oder "Zur Übersicht" vermeiden, stattdessen eher "Jetzt anmelden" oder "Alle Einträge" verwenden  
  Theoretische Möglichkeiten: Buttonbeschriftungen und Headlines sind neutral, auffordernd, seriös, kumpelhaft, kurz, beschreibend, Icons, superkurz (OK/Ja/Nein/Cancel), Verben, Substantive, Imperative etc.pp.
* Anzeige von Zahlen und Daten eher als Tabellen oder eher als Diagramme (z.B. Tortendiagramm)?  
  Diagramme!
* Schreibweisen  
  Produktname: Wenn Nutzer keinen Namen für sein Produkt vergeben hat: Typ-Kurzbezeichnung. Wenn Name vergeben wurde: <Name> + " - " + <Typ-Kurzbezeichnung>  
  Nutzername:  
  Uhrzeit/Datum/Termin/verstrichene Zeit: Termine der fernen Zukunft oder Vergangenheit in Deutsch (mehr als 7 Tage): "17. September 2016", ggf. mit Uhrzeit: "17. September 2016", 17:46 Uhr". Termine der nahen Zukunft oder Vergangenheit in Deutsch: "in 3 Sekunden", "vor 45 Minuten", "in 2 Stunden", "vor 4 Tagen".
* Suche  
  Mit Type Ahead. Mit Maus und Tastatur bedienbar. Gibt es eine Volltextsuche der Seite? Top-Treffer / Facettierung?
* Wiederkehrende Elemente
  + Lange Listen mit / ohne Button(s) in jeder Zeile
  + Maps  
    Suchfeld über der Karte + Type Ahead
  + Formulare  
    s.o.
  + Ampelsignale
  + Nutzernamen / Avatare  
    Keine Avatarbilder!
  + Akkordeon-Box oder Tabs?  
    Wenn nötig - lieber nicht - bisher auch nicht gebraucht. Wenn, dann eher Tabs.
  + Icons
  + ...
* Ajax-Verhalten (dynamisches Austauschen von Seitenteilen ohne Reload)  
  Ja, wo sinnvoll
* Funktionen, die auf allen Seiten verfügbar sind (Drucken, Schriftvergrößerung, ...?)  
  Erst mal nicht

# (Beispielhafte) Responsive Strategie

Maxime definieren: Welche Inhalte stehen auf welchen Devices zur Verfügung? Alle auf allen?

Prinzipien:

Desktop first, Smartphone wird aber immer mitgedacht. Tablet leitet sich aus Desktop ab und soll im Idealfall genauso wie Desktop funktionieren. Nach Möglichkeit ändern sich Schrift- und Bildgrößen zwischen Tablet und Desktop nicht oder nur für wenige Elemente (Design-Aufgabe).

2 Haupt-Breakpoints: Smartphone -> Tablet -> Desktop

Die Smartphone-Variante zeigt eine Hamburger-Navigation, ggf. auch die Tablet-Variante (ggf. auch Design-Entscheidung).

Hover-Effekte können für Desktop geplant werden und müssen dann für Touchdevices in Tap-Actions transformiert werden. Sie sollten deswegen sparsam eingesetzt werden.

Grundsätzliche Überlegungen:

Muss die mobile Variante responsiv (fluides Grid) sein oder kann sie auch adaptiv (festes Grid, auf Desktops können Ränder entstehen) sein? Auch eine separate mobile Website mit dem gleichen Content wie Desktop ist eine valide Lösung bzgl. Userexperience.

Kompakte Erläuterungen zu den Unterschieden "responsive/adaptive/fluide/elastic" usw.:

<https://blog.kulturbanause.de/2012/11/adaptive-website-vs-responsive-website/>

<https://blog.kulturbanause.de/2012/10/die-layout-typen-einer-website-fixed-fluid-elastic/>

Wenn Entscheidung "Responsiv":

* Smartphone:

Einspaltig - fluide (bei Wechsel von Portrait auf Landscape bleiben Bilder und Schrift gleich groß und fließen neu - es passt (theoretisch) genauso viel Content auf den Screen. Bilder über die volle Breite sind elastic(??)). Pinch-Zoom ist möglich. Kein horizontales Verschieben der Website über die Viewportränder hinaus möglich. Sehr breite Inhalte müssen horizontal innerhalb des Viewports gescrollt werden (Bsp.: Tabellen, Diagramme).

* Tablet:

Für manche Elemente zweispaltig - fix (= natives Browserverhalten. Bei Wechsel von Landscape auf Portrait skalieren Bilder und Schrift auf die neue Breite runter und werden dadurch kleiner - es passt mehr Content auf den Screen). Pinch-Zoom ist möglich. Kein horizontales Verschieben der Website über die Viewportränder hinaus möglich.

* Desktop:

Für manche Elemente zweispaltig, ab weiterem (Sub-)Breakpoint, drei- oder vierspaltig. Dazwischen sind die Elemente fluide. Bilder über die volle Breite sind elastic(??). Ab einer zu definierenden Maximalbreite werden die Elemente fix (nicht mehr fluide).

Entscheidungen zur Breakpoint-Anpassung von Schriftgrößen müssen anhand Design und Devices durchgeführt werden!

Zusammenfassung:

Responsive – fluides Grid – 2 Breakpoints

Maxime: Im Web stehen alle Inhalte auf allen Devices zur Verfügung

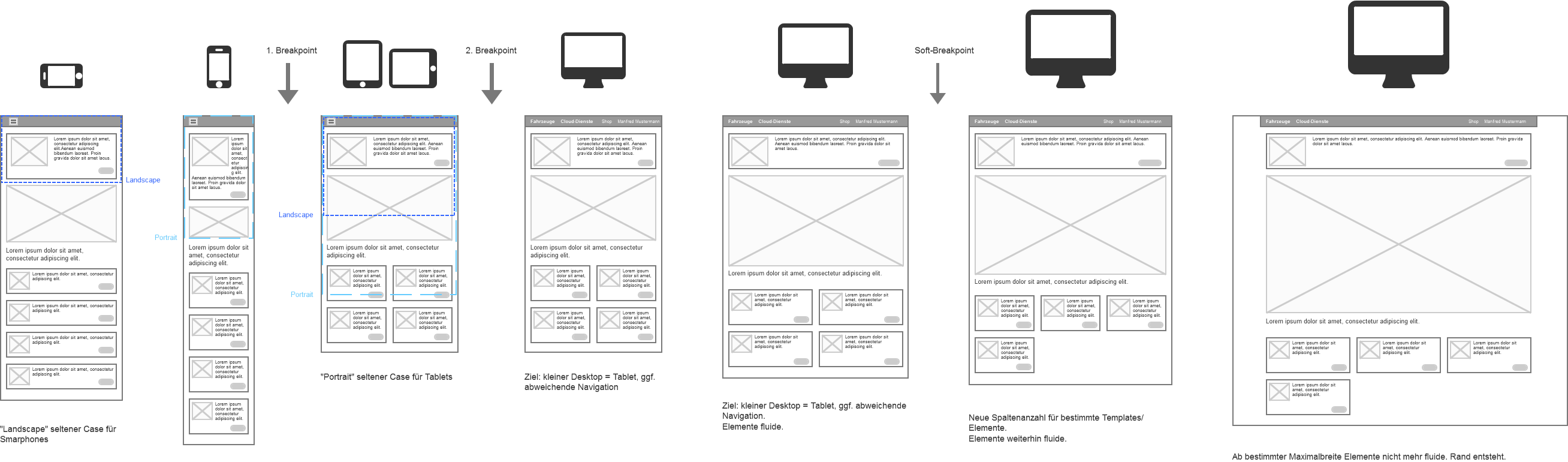
Smartphone: Einspaltig fluide.

Tablet: Hauptsächlich zweispaltig (bei Wechsel von Landscape auf Portrait skalieren Bilder und Schrift auf die neue Breite runter und werden dadurch kleiner).

Desktop: Hauptsächlich zweispaltig und fluide. Einzelne Elemente sind ggf. elastic (ändern Breite und Höhe). Ab einer zu definierenden Maximalbreite werden die Elemente fix (nicht mehr fluide).

Responsive Prototype:

Das responsive Verhalten wird nicht im Axure-Prototyp angelegt. Mit 3 exemplarischen Screens sollte das responsive Verhalten frühzeitig im Blockdiagramm-Stil als echter HTML-Prototyp getestet, optimiert und dokumentiert werden (für fluides Verhalten ist Axure vermutlich ungeeignet). Sobald Design existiert, soll dieses auf den Responsive-Prototyp angewendet werden, um Design und responsives Verhalten zu validieren und im Einklang miteinander zu optimieren.



*Skizze siehe BlockdiagrammPatternsResponsiveStrategie.rp*