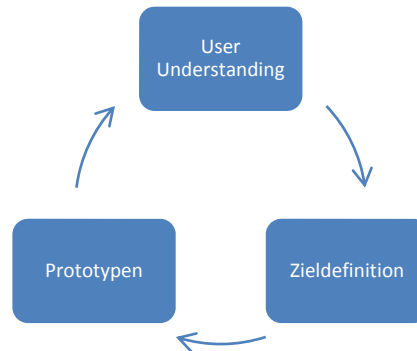


USABILITY

Voraussetzung: Iterativer Designprozess. User Centered Design (ständige Wiederholung, ständiges Testen, bis gewünschtes Ziel erreicht):



1) User Understanding

User Experience Research (Emotional Benchmarking 'need' 'like', Contextual Benchmarking 'can')

Empirische Verfahren (Erfahrung gewinnen, Beobachten), Durch Beobachtung, Durch Tests
Zur Recherche:

- a) Feldstudie Contextual Inquiry: Den Nutzer direkt vor Ort bei seiner Tätigkeit beobachten. Das Verhalten und das Umfeld des Benutzers kennen lernen. Durch Fragen möglichst viel von den Anforderungen an die neue Anwendung erfahren. (z.B.: Fly on the Wall)
- b) Feldstudie Ethnographic Research: Den Nutzer auf psychologischer/ kultureller und emotional/ kultureller Ebene durch Beobachtung kennen lernen. Dadurch Rückschlüsse auf die Art der Nutzung des neuen Systems führen. Zum Beispiel durch „A Day in the Life“-Studie.
- c) Fernstudie Self Reporting Logs: Der Benutzer protokolliert selbst seine Handlungen. Zum Beispiel mit vorgefertigten Formularen oder aber auch mit einer Kamera.
- d) Fernstudie Fragebögen: Teilnehmer füllen Fragebogen aus. Nachteil ist das sehr eingeschränkte Antwortenspektrum. Nur zu empfehlen, wenn es um die Entscheidung zwischen diversen Optionen geht.
- e) Moderation Focus Groups: Befragung der Teilnehmer im Interview oder in der Gruppe. Wichtig sind offene Fragestellungen und Offenheit gegenüber den Teilnehmern. Gut zum Testen von Zielgruppeneigenschaften, Zielgruppenwünsche (gut zur Ideenfindung), aber auch Konzepte/ eigene Ideen. Möglichkeiten: Card Sorting (wobei das eher beim Erstellen des Prototypen hilft), Cognitive Maps, Gruppengespräch, Collagen.
- f) Ergänzung: Nachgespräch mit den Teilnehmern führen um noch einmal kritische Punkte zu beleuchten oder noch neue Erkenntnisse zu gewinnen. Manche Teilnehmer äußern in einem Nachgespräch evtl. auch noch Punkte, die sie in der Gruppe nicht erwähnen wollten.

Aus den Erkenntnissen des UER leitet man nun eine Zieldefinition ab und kann daraufhin in Usabilitytests übergehen.

2) Zieldefinition

Nachdem man die Anforderungen des Nutzers an das System kennt, erstellt man eine Liste mit Zieldefinitionen. Diese Zieldefinitionen greift man auf, um einen ersten Prototypen zu entwickeln und daraufhin Usabilitytests durchzuführen. Die Zieldefinition bildet quasi die Messlatte an den Usabilitytest. Bestätigt der Usabilitytest, dass alle Zieldefinitionen erfüllt wurden, ist das System

Was sollte eine Zieldefinition beinhalten?

- Anforderungen an die Gestaltung der Arbeitsaufgabe (Aufgabenerfordernis)
z.B.: Das System muss eine Rückmeldung geben, ob eine Tätigkeit ausgeführt wurde
- Anforderungen an den Durchführungsrahmen der Arbeitsaufgabe (Organisationserfordernis)
z.B.:
- Anforderungen an die vom Benutzer ausgeführte Tätigkeit (Kognitionserfordernis)
- Anforderungen an das Dialogsystem (Dialog oder Produkthanforderung)

3) Prototypenbau

Zu Unterscheiden:

- Prototyp von Recherche zu Konzeption (User Experience Design) ist explorativ, initialisierend. Trägt noch zur Ideenfindung bei. (zwingend Lo-Fi)
- Prototyp von Konzeption zu Produktion (Usability Testing) ist gezielt, bestätigend. (evtl Hi-Fi)

Wichtig: Informationsarchitektur wird oftmals unterschätzt, da sie „unsichtbar“ ist. Allerdings entstehen Usability- Probleme gerade aus dem Grund, weil sich eine Anwendung nicht normkonform verhält. Informationsarchitektur ist also zwingend nötig und sollte zusammen mit dem Nutzer erarbeitet werden.

Zur Informationsarchitektur gehören:

- Inhalte Sammlung (Content-Map)
- Inhalte Strukturierung (Struktogramm, Contentogramm, Diagramme allgemein)
- Handlungsabfolge Gestaltung (Struktogramm, Contentogramm)
- Einsatz von Sprache, Verwendung richtiges Vokabulara

Informationsarchitektur = Content + Informationsstruktur + Navigationsstruktur + Präsentationsstruktur + Präsentationsdesign

Um die Informationsarchitektur zu bestimmen, kann man Card Sorting (z.B: Online-Ticketkauf, Was erwartet der Benutzer zuerst? Was möchte er erst angeben? Was ist für ihn unwichtig?) anwenden.

Nachdem man die Architektur festgelegt hat, sollte man erste Prototypen bauen. Lo-Fi Prototypen, damit das Augenmerk auf die Benutzung und nicht auf Details wie Farbe etc gelenkt wird.

Hintergrundinformation: Kognetik (Die Rolle des Gedächtnisses):

Die kognitiven Fähigkeiten eines Menschen sind: Aufmerksamkeit, Erinnerung, Lernen und Kreativität.

Bilder werden von unserem Gehirn schnell und unmittelbar wahrgenommen. Sie wirken emotional intensiv und direkt. Dadurch werden sie gut erinnert und haben eine hohe Effizienz für das Gedächtnis.

Texte hingegen müssen Wort für Wort gelesen werden und wirken daher eher verzögert. Sie sind typischerweise keine Blickfänger und wirken emotional weniger intensiv

Im Idealfall stellt man Bild und Wort gemeinsam dar (Duale Codierung, spricht beide Gehirnhälften an), was die Erinnerungsleistung steigert. Ebenso sind konkrete Begriffe, die man sich bildhaft vorstellen kann wesentlich einfacher zu merken als abstrakte.

Prinzipiell ist es so, dass Sachverhalte, die auf mehreren Wahrnehmungskanälen kommunizieren besser gemerkt werden können. Der Mensch merkt sich:

20% von dem was er hört

30% von dem was er sieht

50% von dem was er hört und sieht
70% worüber er selbst spricht
90% von dem was er selbst ausführt

Man kann das Speichervermögen steigern durch

- Chunking-Strategie (Brockenbildung) 12-34-56 statt 123456
- Penetration (Wiederholung desselben Icons)
- 7 +-2 oder 4+-1 Strategie (maximale Anzahl an Elementen)

Folgende Faktoren beeinflussen außerdem unser Langzeitgedächtnis:

- Emotional geladene Situationen werden besser erinnert (Schockwerbung)
- Blitzlicht-Erinnerungen (Erschießung J.F. Kennedy, Mondlandung)
- Negative Emotionen können blockieren. Ein Blackout tritt ein
- Erinnerung ist am Besten, wenn der Kontext bei der Wiedergabe der Selbe ist wie bei der Aufnahme (Icon auf rotem Hintergrund)

Man kann unterscheiden zwischen dem Neu erlernen (Kurzzeitgedächtnis, Arbeitsgedächtnis) und dem Abrufen (Langzeitgedächtnis) von mentalen Modellen.

Erkenntnis: Standards sind aus Sicht der Benutzerfreundlichkeit immer zu bevorzugen. Suchzeiten und die Orientierung in interaktiven Anwendungen werden optimiert, wenn der Benutzer die Elemente an der „richtigen“ Stelle findet. Also dort, wo er sie erwartet. Standards vertiefen die Wiederholung (Penetration). Allerdings verhindern sie auch ein Überraschungselement und Innovation.

Hintergrundinformation: Das Mentale Modell:

Jeder Mensch hat eine innere Modelldatenbank mit der eine gewisse Erwartungshaltung verknüpft ist. Sieht der Benutzer einen Button, geht er davon aus, dass dieser gedrückt werden kann und eine Aktion ausgelöst wird.

Das mentale Modell ist also eine Vorstellung darüber, was ein Objekt darstellt und wie es funktioniert. Diese Vorstellung sollte bedient werden um keinen Bruch der Erwartungshaltung zu erzielen!

Folgende mentale Modelle kann man bedienen:

- Physische Modelle (1:1 Abbildung eines realen Gegenstandes, Illustration eines realen Gegenstandes.)
- Organisationsmodelle (Ordneranzeige, Level-Navigation, Baumstruktur)
- Dialogmodell (Eingabemaske)
- Gestische Modelle (Illusion, dass Mauszeiger der verlängerte Arm ist)

4) Usability Testing

Mögliche empirische Verfahren zum Usability-Testing (Idealerweise kombinierbar mit Think Aloud-Methode):

- a) Empathy Tool. Eine Zielgruppe Nachstellen/ Imitieren. Zum Beispiel Bedienung eines Handys mit „dicken“ Händen. Bedienung einer Webseite mit Farbblindheit u.ä.
- b) Paper Prototypes. Nachbilden der Anwendung mit Papierprototypen. Die Prototypen sollten Lo-Fi Prototypen sein, damit man sich nicht an Details auslässt, sondern sich ausschließlich auf die Benutzbarkeit konzentriert
- c) Ergänzung: Nachgespräch mit den Teilnehmern führen um noch einmal kritische Punkte zu beleuchten oder noch neue Erkenntnisse zu gewinnen. Manche Teilnehmer äußern in einem Nachgespräch evtl. auch noch Punkte, die sie in der Gruppe nicht erwähnen wollten.

Mögliche heuristische Verfahren:

- a) Usability Inspection (Experten Evaluation). Der Test wird durchgeführt von Usability-Experten, Entwicklern und erfahrenen Endbenutzern. Es wird geprüft nach Einhaltung von Richtlinien (Normen) und Konventionen. Nach der Vermeidung von Fehlern und Bedienung und nach der Verringerung des Lern- und Interaktionsaufwands. Durch die Beteiligung von Experten können die Probleme direkt zusammengefasst und priorisiert werden. Anwendungsabhängige Probleme bleiben allerdings oft unerkannt und können besser durch empirische Tests (Empathy Tool, Paper Prototypes) festgestellt werden.

Zusatzinformation: Wie erstellt man einen Usabilitytest?

Das DATech-Prüfhandbuch Gebrauchstauglichkeit gilt als Leitfaden für Prüfer die ein Softwareprodukt auf Normkonformität mit DIN EN ISO 9241-11 und -110 durchführen wollen. Es dient dazu um eigene Prüfanweisungen, etwas für ein Usabilitylabor, zu entwickeln.

Bei der Usability zu beachten:

DIN EN ISO 9241-11: Anforderungen an die Gebrauchstauglichkeit.

Der Benutzer muss Ziele effektiv (vollständig und genau), effizient (wenig Aufwand) und zufrieden stellend (positive Einstellung gegenüber der Nutzung) erreichen!

DIN EN ISO 9241-110: Grundsätze der Dialoggestaltung.

DIN EN ISO 13407: Benutzer-orientierte Gestaltung interaktiver Systeme

Hintergrundwissen:

Das klassische Qualitätsmanagement bei Software besteht aus: Funktionalität, Zuverlässigkeit, Wartbarkeit und Übertragbarkeit (ISO 9000 ff.)

Da bei diesem Qualitätsmanagement nur die Qualität der Software, nicht aber die Benutzerfreundlichkeit überprüft wird, prüft man im ganzheitlichen Qualitätsmanagement auch noch die Gebrauchstauglichkeit (ISO 9241 / ISO 13407).

Checkliste nach DIN 9241-110: Ist das System...

- Lernförderlich? Können Arbeitsaufgaben mit minimalem Lernaufwand erledigt werden? Gibt es eine geeignete Lernunterstützung für bestimmte Bereiche, Buttons, Anwendungen?
- Erwartungskonform? Greift die Webseite mentale Modelle auf? Wird vertrautes Vokabular benutzt. Ist die Seite natürlich strukturiert und organisiert? Ist sie konsistent?
- Individualisierbar? Kann der Benutzer die Darstellung seinen persönlichen Wünschen und Fähigkeiten anpassen. Ist die Barrierefreiheit gewährleistet?
- Fehlertolerant? Können Fehler schnell rückgängig gemacht werden? Ist die Fehlerkorrektur einfach (z.B. bei Formularen)? Gibt es eine automatische Korrektur (Disign = Design)? Werden Fehler vernünftig ausgewiesen?
- Aufgabenangemessen? Werden keine unnötigen Informationen angezeigt? Steht die Anforderung des Nutzers oder das System im Vordergrund?
- Selbstbeschreibungsfähig? Ist es dem Benutzer immer offensichtlich wo er sich gerade befindet? Gibt es interne Erläuterungen (Mouseover o.ä) oder muss der Benutzer Handbücher zu Hilfe nehmen? Werden Zustandsänderungen (z.B. erwartete Eingabe) kenntlich gemacht?
- Steuerbar? Können Dialoge in der Richtung beeinflusst werden? Können Schritte rückgängig gemacht werden? Können Suchergebnisse sortiert (z.B. aufsteigend, absteigend) und auf andere Arten dargestellt werden (Baumansicht, Hierarchisch)?

Auswertung der Checkliste im Idealfall mit einer Soll-Ist-Analyse mit Spinnennetz-Diagramm.

Checkliste für Barrierefreies-Webdesign basierend auf WCAG1:

Priorität 1:

- Gibt es Textalternativen für alle Elemente die keine Texte sind? (Video, Bilder, Applets usw)
- Sind alle Informationen die in Farbe vorliegen auch in Graustufen interpretierbar (+ und -, statt rot und schwarz)?
- Kann das Dokument auch mit abgeschaltetem Stylesheet gelesen werden?
- Existiert Bildschirmflackern durch kritische Farbkombinationen (komplementäre Farben)?
- Sind alle Texte für jedermann verständlich?
 - o Kurze Sätze
 - o Kein Konjunktiv
 - o Keine Fremdwörter
 - o Abkürzungen müssen erklärt werden
- Sind Tabellen richtig gekennzeichnet (Spalten und Reihenüberschriften)
- Liegen Audiobeschreibungen für Multimediainhalte vor (für Sehbehinderte)?
- Werden Synchronisationen angeboten (Untertitel, Gebärdensprache)

Priorität 2:

- Bietet Vorder- und Hintergrundfarbe genügend Kontrast für Menschen mit Farbwahrnehmungsproblemen?

- Werden Stylesheets verwendet um das Aussehen der Seite zu gestalten?
- Werden relative statt absolute Angaben für Attribut und Style-Eigenschaften verwendet?
- Werden Überschriften-Elemente verwendet um das Dokument richtig zu strukturieren?
- Werden die richtigen Elemente zur Textauszeichnung benutzt (Listen, Zitate)
- Sind dynamische Inhalte zugänglich, oder existieren Alternativen?
- Sind die Ziele der Links deutlich benannt?
- Existiert eine Sitemap/ Inhaltsverzeichnis
- Ist die Navigation konsistent
- Werden Tabellen zum Layouten benutzt?
- Sind Formularfelder erkenntlich betitelt und mit dem zugehörigen Formular verknüpft?

Checkliste für Barrierefreies-Webdesign basierend auf WCAG2:

Basierend auf dem Entwurf der WCAG2, werden die Kriterien der WCAG1 übernommen und in vier Prinzipien unterteilt:

- Wahrnehmbarkeit
 - o Gibt es zu jedem Element, das selbst kein Text ist, ein Alternativtext?
 - o Gibt es synchronisierte Alternativen für Multimedia?
 - o Ist die Information, Struktur und Funktion auch ohne Stylesheet erkenntbar?
 - o Sind Vordergrundinhalte (Text) und Hintergrundelemente (Grafiken) klar unterscheidbar?
- Bedienbarkeit
 - o Sind alle Funktionalitäten auch per Tastatur bedienbar?
 - o Ist die Plattformunabhängigkeit gegeben?
 - o Gibt es Mechanismen, die den Benutzern beim Finden von Inhalten, bei der Orientierung innerhalb des Inhalts und bei der Navigation durch den Inhalt helfen?
 - o Wird der Benutzer dabei unterstützt Fehler zu vermeiden, bzw. besteht die Möglichkeit Fehler einfach zu korrigieren?
- Verständlichkeit
 - o Sind alle Texte in einfacher Sprache?
 - o Sind alle Inhalte in konsistenter Weise angeordnet?
- Robustheit
 - o Wurden für alle Techniken die entsprechenden Spezifikationen (W3C) berücksichtigt?
 - o Sind Schnittstellen verfügbar oder gibt es Alternativen?

Usability- Checkliste (basierend auf Jakob Nielsen und Bruce Tognazzini) (Usability Inspection)

- Weiß der Benutzer immer, was das System gerade macht?
- Können Aktionen rückgängig gemacht werden?
- Ist die Konsistenz auf allen Seiten gegeben? (Sowohl intern als auch extern (siehe mentales Modell))

- Sind Funktionen, die nicht benutzt werden ausgegraut? Unterstützt das System den Benutzer bei der Fehlervermeidung?
- Gibt es Shortcuts? Bookmarkfunktionen?
- Gibt es unnötige Features?
- Liegen Fehlermeldungen in einfacher Sprache vor? Gibt es konstruktive Lösungsvorschläge?
- Existiert eine einfache, klar beschriebene Hilfe zum System?
- Erkennt die Anwendung die Wünsche des Benutzers und stellt beispielsweise Werkzeuge fallbezogen dar?
- Steht die Produktivität des Benutzer im Vordergrund (nicht die des Systems)? Effizienz kann gesteigert werden durch kurze Wartezeiten, Systemrückmeldungen, knappe Hilfetexte, gute Beschriftungen.
- Wird die Wahrnehmung der Wartezeit verringert? (Durch Animation, Information)
- Werden alle Eingaben Zwischengespeichert? Die Arbeit des Benutzers darf nie verloren gehen! (z.B bei Formularen)
- Ist die Lesbarkeit und die Barrierefreiheit gegeben?
- Existieren mehrere Wege um ein Ziel zu erreichen? (Suche, Sitemap, Inhaltsverzeichnis) Und wird der beste Weg hervorgehoben?
- Existiert eine Suchfunktion? Ist diese leicht zu bedienen? Und werden die Suchergebnisse sinnvoll dargestellt?
- Ist es immer klar, wo der Benutzer herkam, wo er ist, und wo er hin kann?