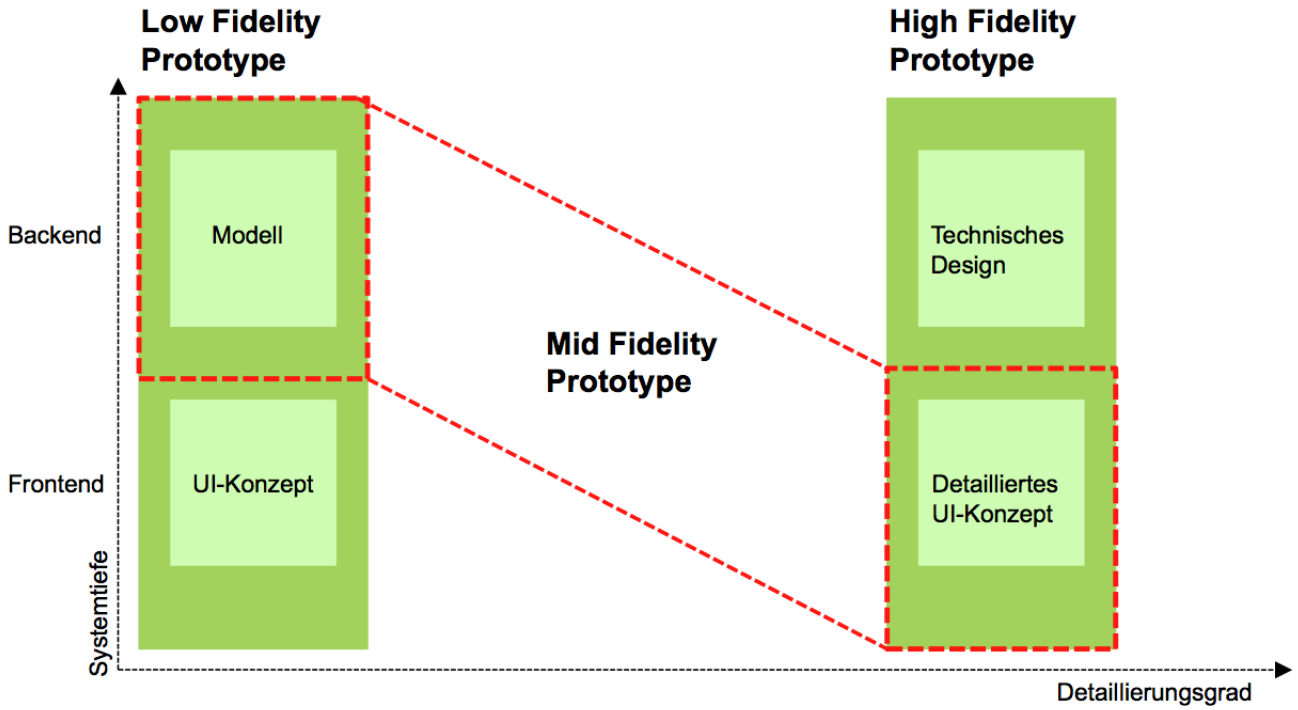
Usability & User Interface Design

Jan Fässler

Test 2 (FS 2012)

1 Low- & Hi-Fi Prototyping



1.1 Low Fidelity Prototype (Low-FI)

Papier, Storyboards, Skizzen

+ Geringe Entwicklungskosten-/zeit
+ Geeignet fur Grobkonzepte
+ Evaluation von Alternativen
+ Ideen kommunizieren
+ Kein technisches Know-How nötig
- Kaum Navigation und Interaktion
- Eingeschränkte Wiederverwendung
- Fehlerbehandlung nicht enthalten

1.1.1 Sketching

Schnelles Zeichnen mit Papier und Bleistift.
Ziel und Zweck:

- Ideen fur spätere Entwicklung sammeln
- Schnelle Visualisierung der Konzepte
- Einfaches Kommunikationsmittel
- Schnell verbreitet

1.1.2 Balsamiq Mockups

Tool zum zusammenklicken eines Prototypen. damit kann man schnell und einfach eine Oberflche skizieren.

1.1.3 Papier-Prototypen

Kann verwendet werden, wenn die Funktionalitäten bekannt sind.
Ziel und Zweck:

- Visualisierung der Funktionalität
- Aufwändiger in der Erstellung als Sketches
- Einfach durch Nutzer testbar
- Machbarkeits-Test
- Unterschiedliches Material (Papier, Post-It

1.2 Mid Fidelity Prototype (Mid-FI)

HTML, Flash, UI Builders, Wizard of OZ

+ Look & Feel ähnlich Endprodukt
+ Interaktiv, Navigation
+ Weiterverwendung möglich
+ Detaillierte Nutzer-Rückmeldung
- Mittlere/Hohe Entwicklungskosten/-zeit
- Einschränkung der Kreativität
- Gefahr der Detailarbeit

1.2.1 Wizard of OZ

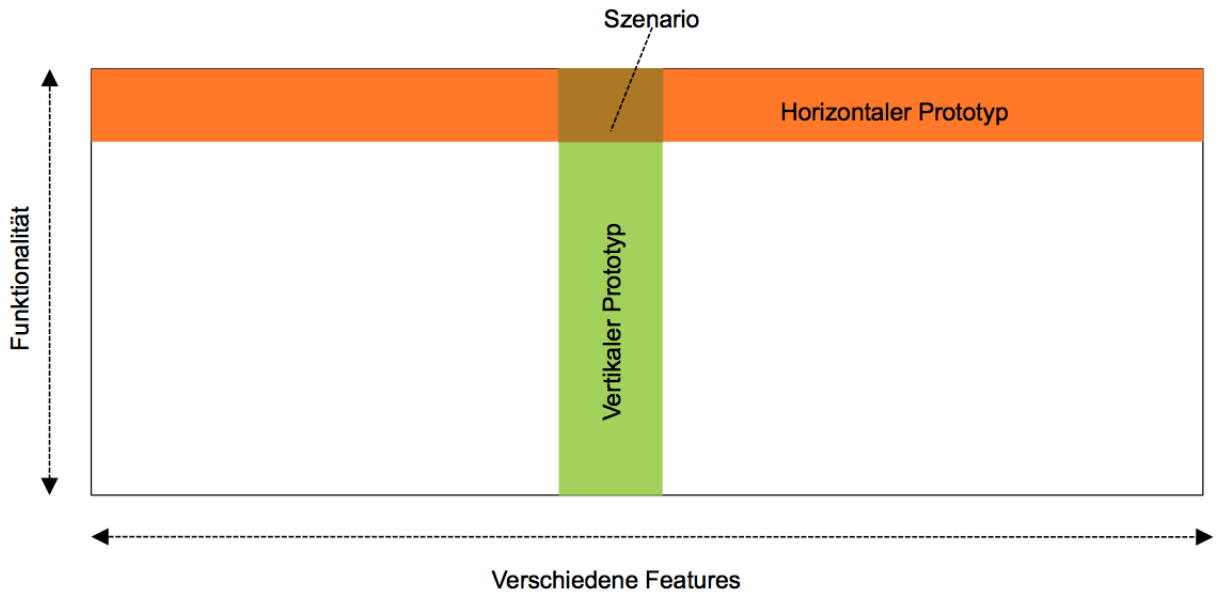
- Forschungs-Experiment
- Kelley, John F. (1980)
- System-Interaktion wird durch einen Menschen simuliert
- Beispiel: Sie möchten ein System testen, welches einem Nutzer einen Code per SMS schickt. Dieser Code muss vom Nutzer eingegeben werden, um eine Anmeldung fertig zu stellen. Der Code wird nicht vom System versandt, sondern von einem Menschen.

1.3 High Fidelity Prototype (HI-FI)

HTML, Flash, UI Builders, Programmierung

+ Look & Feel ähnlich Endprodukt
+ Realistisch, interaktiv, zeitecht
+ Weiterverwendung möglich
+ Detaillierte Nutzer-Rückmeldung
- Hohe Entwicklungskosten/-zeit
- Einschränkung der Kreativität
- Gefahr der Detailarbeit

2 Horizontale und vertikale Prototypen



2.1 Horizontaler Prototyp

Alle Features, dafür wenig Funktionalität. Beispielsweise das gesamte Interface.

2.1.1 Evaluation

- der Navigation, des UI-Gesamtkonzeptes
- des Zugangs zum System

2.1.2 Verwendung

- In Low-FI und HI-FI
- in frühen Designphasen, um das Feature-Set und das Grobkonzept zu überprüfen

2.2 Vertikaler Prototyp

Volle Funktionalität fur ein paar wenige Features

2.2.1 Evaluation

- einer einzelnen Funktionalität
- des spezifischen Nutzer-Verhaltens

2.2.2 Verwendung

- In HI-FI und (Low-FI)
- in frühen Designphasen um verschiedene Designs einzelner Funktionalitäten zu prüfen überprüfen
- in späteren Designphasen, um spezifische Funktionalität zu optimieren

3 UI Builder

- Software, die einem ermöglicht, eine grafische Benutzeroberfläche zu entwickeln.
- Standardisierte Widgets werden mittels Drag und Drop auf das Interface- Fenster verschoben und dort individuell weiter bearbeitet.
- Die Software erzeugt Programmcode.
- Eigenständige Applikation oder Teil eines Application Development Systems
- GUI-Builder sind ein wesentlicher Bestandteil der modernen Programmierung

3.1 Vorteile

- Wirtschaftlich: Entwicklung, Test und Pflege kann effizienter sein
- Qualitat des UIs kann profitieren
  - Weniger Code schreiben
  - Verbesserte Modularisierung (Visualisierung von Logik separiert)
  - Programmier-Expertise im Team darf variieren
  - Zuverlässigkeit des UIs erhöht (Code-Generierung)

3.2 Nachteile

- Wirtschaftlich: Entwicklung, Test und Pflege kann erschwert werden
- Qualität des UIs kann leiden wenn Builder
  - Manuelle Code-Änderungen verhindern
  - Hersteller-spezifische Files verwenden
  - Die generierte Struktur nicht aufgebrochen werden kann oder nicht zufriedenstellend ist

4 Qualitätskriterien

Informationsarchitektur
Info fur den User schnell auffindbar? Logisch aufgebaut?

Navigation
Logische Struktur, Ebenenwechsel, links, oben, klare Titel, Hilfe rechts

Informationsdarstellung
Klarheit, Konsistenz, Unterscheidbarkeit, Lesbarkeit

Kompaktheit
Nicht zu kompakt fur das Auge, aber dennoch gutes Mass

Konsistenz
Ähnliche Funktionen sich gleich aufrufbar

Farbgestaltung
Klassifizierung, Auffindbarkeit, Wahrnehmung

Schriftgestaltung
Zeilenlange, Zeilenabstand, Buchstabenabstand, Schriftart

Bildsprache
Blickrichtung, Format, goldene Schnitt, Linien, Kontraste

Benutzerfuührung
Aufgabenbez., wählbar, gut Formuliert, Ruckmeldung, Info

5 Grundsätze der Dialoggestaltung

Aufgabenangemessenheit
Ein Dialog ist aufgabenangemessen, wenn er den Benutzer unterstutzt, seine Arbeitsaufgabe effektiv und effizient zu erledigen.

Selbstbeschreibungsfähigkeit
Ein Dialog ist selbstbeschreibungsfähig, wenn jeder einzelne Dialogschrittdurch Rückmeldung des Dialogsystems unmittelbar verständlich ist oder dem Benutzer auf Anfrage erklart wird.

Steuerbarkeit
Ein Dialog ist steuerbar, wenn der Benutzer in der Lage ist, den Dialogablauf zu starten sowie seine Richtung und Geschwindigkeit zu beeinflussen, bis das Ziel erreicht ist.

Erwartungskonformität:
Ein Dialog ist erwartungskonform, wenn er konsistent ist und den Merkmalen des Benutzers entspricht, z. B. seinen Kenntnissen aus dem Arbeitsgebiet, seiner Ausbildung und seiner Erfahrung sowie den allgemein anerkannten Konventionen.

Fehlertoleranz
Ein Dialog ist fehlertolerant, wenn das beabsichtigte Arbeitsergebnis trotz erkennbar fehlerhafter Eingaben entweder mit keinem oder mit minimalem Korrekturaufwand seitens des Benutzers erreicht werden kann.

Individualisierbarkeit
Ein Dialog ist individualisierbar, wenn das Dialogsystem Anpassungen an die Erfordernisse der Arbeitsaufgabe sowie an die individuellen Fähigkeiten und Vorlieben des Benutzers zulässt.

Lernförderlichkeit
Ein Dialog ist lernförderlich, wenn er den Benutzer beim Erlernen des Dialogsystems unterstützt und anleitet.