



5 User-Centered Design

5.5 Interaction Design Patterns

Problemstellung



- Ziel:
Erfahrung im Entwurf von
Benutzerschnittstellen vermitteln
 - Was einem ein alter Hase beim Mittagessen erzählt ...
- Probleme:
 - Guidelines sind wenig in konkreten Beispielen verwurzelt
 - Nur aus Ansehen existierender Projekte (Vorzüge / Fehler) zu lernen ist nicht sehr effizient
- Lösung:
Design Patterns (Entwurfsmuster)

Was ist ein Design Pattern?



- Ein paar Seiten Text und Skizzen
- Beschreibung einer bewährten Lösung eines wiederkehrenden Entwurfsproblems
- Ursprung in der Architektur
 - Christopher Alexander: The Timeless Way of Building. 1977.
 - Christopher Alexander: A Pattern Language. Towns • Buildings • Construction. Oxford, 1979.
 - 253 Patterns für "gute" Architektur



Folie 268

Beispiel: Intimacy Gradient



- Einerseits gibt es Räume mit verschiedener Privatsphäre (Flur vs. Schlafzimmer), andererseits will man Besucher je nach Bekanntheitsgrad unterschiedlich weit in die Privatsphäre eindringen lassen
- Lösung:
Vom Eingangsbereich bis zu den hintersten Räumen einen Gradienten zunehmender Privatsphäre schaffen.



Folie 269

Interaction Design Patterns



- Jedes Pattern beschreibt eine bewährte Lösung für ein Problem bei der Gestaltung von Benutzerschnittstellen
 - Name: Vokabular für das Designteam
 - Kontext: Was ist das Problem, welche Konflikte bestehen (Forces), Verweise auf „größere“ Patterns
 - Lösung: Wie löst man das Problem in diesem Kontext; oft mit Skizze
 - Beispiele: Bewährte Anwendungen des Patterns in existierenden Systemen
 - Konsequenzen: Verweise auf „kleinere“ Patterns



Folie 270

Patternbestandteil: Name



- Kurze, „selbsterklärende“ Bezeichnung für das Konzept, das das Pattern beschreibt
- Beispiel: „Intimacy Gradient“
- In der Diskussion im Entwurfsteam kann man mit einem Wort sagen, welches Konzept man einsetzen will
 - „Let’s arrange the rooms in an Intimacy Gradient“
- Ein Vokabular für Designer entsteht



Folie 271

Patternbestandteil: Kontext



- Wann / in welcher Design-Situation kann man das Pattern einsetzen?
- Bezieht sich auf höherliegende / "größere" Patterns
- Beschreibt konkurrierende Interessen (Kräfte, "Forces"), die das Design in unterschiedliche Richtungen "ziehen"



Folie 272

Patternbestandteil: Lösung



- Eine Lösung für das im Kontext beschriebene Designproblem
 - Balanciert die "Forces" aus
- Generisch
 - Durch Verwenden der Pattern-Lösung kann man für konkretes Design eine Lösung erzeugen
- Zentraler Teil jedes Patterns
- Oft mit graphischer Skizze



Folie 273

Patternbestandteil: Beispiele



- Nur eine Lösung, die bereits vielfach erfolgreich eingesetzt wurde, kann als Pattern bezeichnet werden
 - Rule Of Three
- Beispiele aus existierenden Systemen helfen dem Leser, das Pattern mit Bekanntem zu verknüpfen und so besser zu verstehen
- Oft mit Bildern



Folie 274

Patternbestandteil: Konsequenzen



- Folgen (positiv und negativ) des Einsatzes des Patterns
- Verweise auf "kleinere" Patterns, die verwendet werden sollten, um die Lösung zu implementieren



Folie 275

Patternhierarchie



- Patterns sind nur sinnvoll, wenn sie auf andere Patterns verweisen (Kontext + Konsequenzen)
- Ergebnis: Hierarchische Pattern Language
- Abstrakte Patterns für Probleme "im Großen" verweisen zu ihrer Umsetzung auf weitere Patterns, die Probleme "im Kleinen" lösen
- Führt den Designer beim Entwurf vom Abstrakten zum Konkreten



Folie 276

Beispielpattern: DynamicDescriptor



- Auch bekannt als
 - BalloonHelp (Mac OS)
 - Tool Tips (Windows)
 - Short Description (Tidwell)
- Kontext
 - Oberfläche, mehrere Objekte
 - Nicht alle gleichzeitig erklärbar
 - Benutzer ist Anfänger
- Kräfte
 - Präsenz: Kontext verlassen unbequem
 - Ausführlichkeit: Platz geometrisch/ästhetisch begrenzt



Folie 277

DynamicDescriptor



DynamicDescriptor



Onlinehilfe



Handbuch

Präsenz

Ausführlichkeit



Folie 278

DynamicDescriptor



▪ Lösung

- Objekte knapp repräsentieren (Icon, Name)
- Weitere Informationen dynamisch einblenden
- In räumlicher / zeitlicher Nähe
- Auf Benutzerinitiative hin (Mouse-Over, ...)

▪ Konsequenzen

- Zwischenstufe zwischen Ausgangsrepräsentation und Zielobjekt (verwirrend?)
- Drei-Phasen-Interaktionsobjekte (unruhig?)



Folie 279

DynamicDescriptor



- Querverweise
 - Realisiert:
ExplorableInterface, SimpleAppearance
 - Verwendet:
3-State-Button oder vergleichbares
- Weitere Einsatzbeispiele
 - URL-Linktexte [Nielsen98]
 - WorldBeat-Auswahlseite



Folie 280

Jenifer Tidwells Pattern Language



- Umfangreiche Sammlung (über 50 Patterns)
- Geordnet nach Fragestellungen
 - Wie wählt man die Grundform des Inhalts?
 - Wie wählt man die Grundform der Interaktion?
 - Wie präsentiert man den Inhalt?
 - Wie gestaltet man Navigation?

http://www.mit.edu/~jtidwell/interaction_patterns.html



Folie 281

Nutzen von Design Patterns



- Vokabular für Entwickler
- Einlernen neuer Mitglieder im Designteam
- Schulung
- Bewahren von Erfahrung aus vergangenen Projekten
 - Corporate Memory
 - Experience Factory Ansatz
- Auch verständlich für Nicht-Designer: Softwareentwickler, Benutzer!



Folie 282

Interdisziplinärer Einsatz

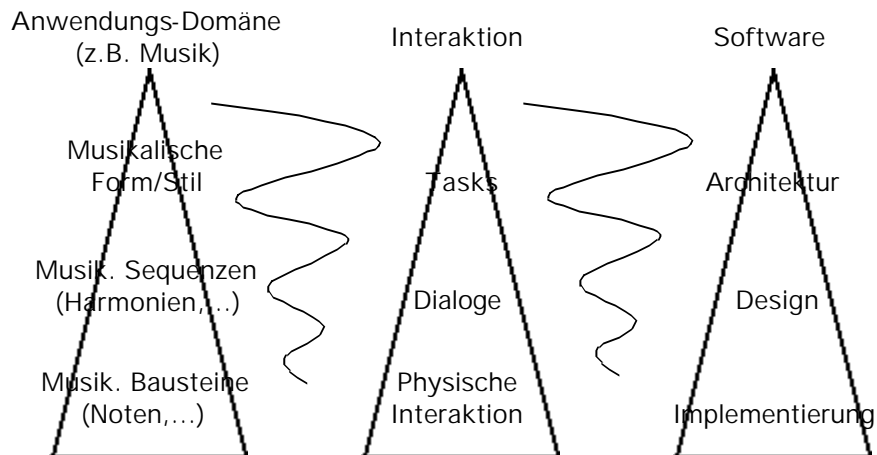


- Design Patterns können auch andere Themen als User Interface Design beschreiben
 - Erfolgreich und verbreitet im Software Engineering
 - Gang-of-Four Buch: Gamma, Helm, Johnson, Vlissides: Design Patterns, 95.
 - Einsetzbar in der Anwendungsdomäne der Benutzer
- Ein interdisziplinäres Designteam (Entwickler, Designer, Benutzer) kann sein Wissen als Patternsatz formulieren und so besser kommunizieren



Folie 283

Interdisziplinäre Pattern Language



Folie 284

Zusammenfassung



- Interaction Design Patterns beschreiben bewährte Lösungen wiederkehrender Entwurfsprobleme bei der Gestaltung interaktiver Systeme
- Name, Kontext (Forces), Lösung, Beispiele, Konsequenzen
- Hierarchische Pattern Language
- Nutzen: Gut geeignetes Format für
 - Vokabular, Design Rationale, Corporate Memory
- Beispiel: Tidwell

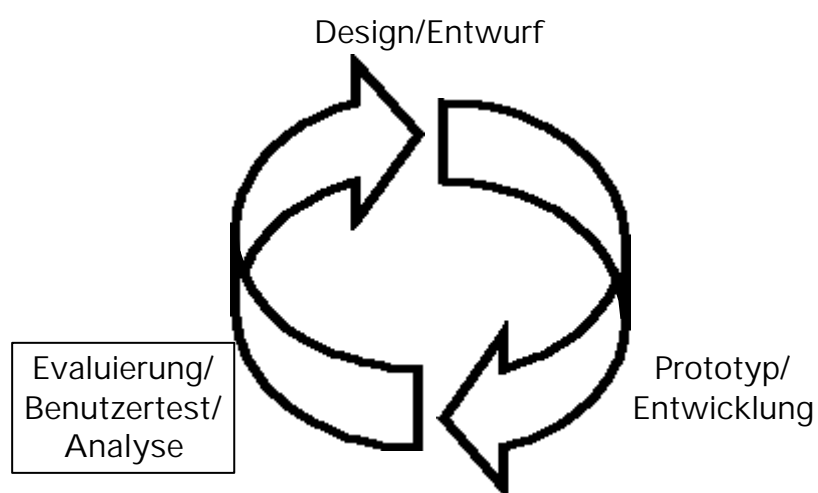
Folie 285



5 User-Centered Design

5.6 Evaluation

Nochmal: Der Entwurfszyklus



Wozu Evaluierung?



- Evaluierung soll sicherstellen, dass das System den Anforderungen der Benutzer entspricht
- Evaluierung ist auch noch nötig, wenn Entwurf bereits benutzerorientiert (Umfragen, ...) war!
- Evaluierung sollte während der gesamten Softwareentwicklung erfolgen
 - im Entwurf häufig durch Designer (analytisch-informell)
 - in Implementierung noch mehr durch Benutzer (experimentell-formell)



Folie 288

Ziele der Evaluierung



- Funktionalität des Systems bewerten
 - Erleichtert das System die Tasks der Benutzer?
 - Sind die richtigen Funktionen vorhanden, gut erreichbar und erwartungskonform präsentiert?
- Auswirkungen auf den Benutzer bewerten
 - Wie gut ist das System erlern- und bedienbar?
 - Welche Einstellungen entwickeln Benutzer zum System?
 - Gibt es Bereiche, die Benutzer überlasten?
- Spezifische Probleme aufdecken
 - Entstehen unerwartete/verwirrende Situationen?



Folie 289

Evaluierungsformen 1: Laborstudien



(allgemeine Formen, nicht konkrete Techniken)

- Mit oder ohne Benutzer
- Vorteile:
 - Ausstattung vorhanden (Audio/Video-Aufzeichnung, halbdurchlässiger Spiegel, spezielle Rechner)
 - Keine Unterbrechungen, Ruhe
- Nachteile:
 - Umfeld fehlt (Aktenschränke, Wandkalender, ...)
 - Unnatürliche Situation → Relevanz?
- Einzige Methode, wenn realer Einsatz gefährlich/entfernt (Raumstation, ...) oder kontrollierte Situation nötig



Folie 290

Evaluierungsformen 2: Feldstudien



- Untersuchungen in der Arbeitsumgebung der Benutzer
- Vorteile:
 - Realistischer (auch wegen Unterbrechungen)
 - Situationen und Verhalten natürlicher
 - Besser für Langzeituntersuchungen geeignet
- Nachteile:
 - Lärm, Unruhe, Unterbrechungen, unterbricht Arbeit
 - Testsituation trotzdem gegeben



Folie 291

Evaluierungsformen 3: Entwurfsbeteiligung



- Auch: „Participatory Design“
- Betrifft nicht nur Evaluierung, sondern gesamten Entwicklungsprozess
- Benutzer als Kontext-Experte Teil des Entwurfsteams



Folie 292

Evaluierungsformen 3: Entwurfsbeteiligung



- Ursprung in Skandinavien, dort gesetzlich verankert
- Ansonsten weniger verbreitet, da aufwendig und Probleme mit Machthierarchien beim Kunden
- Techniken zur Kommunikation im Team:
 - Brainstorming, Storyboarding, Workshops (Fragestunden/Rollenspiele), Papier-Prototypen



Folie 293

Evaluierungstechnik für den Entwurf

E1: Cognitive Walkthrough



- Ohne Benutzer;
Experte= Designer oder
kognitiver Psychologe
- Ziel: Vor allem Erlernbarkeit beurteilen
- Design jeder Task schrittweise durchgehen
- Fragen:
 - Wie wirkt die Interaktion auf den Benutzer?
 - Welche kognitiven Prozesse benötigt er?
 - Gibt es Probleme beim Erlernen dieses Schritts?



Folie 294

Cognitive Walkthrough



- ➔ Hilft das System dem Benutzer, von
Tasks zu Zielen und Aktionen zu
kommen?
- ➔ Benötigt Interface-Beschreibung,
Task-/Ziele-/ Aktionenbeschreibung
(GOMS) und Benutzerprofil



Folie 295

Cognitive Walkthrough



- Vorgehen:
 - Wähle Task – beschreibe Ziele – bestimme Aktionen
 - Analysiere den Entscheidungsprozeß (obige Fragen)
- Formulare mit Fragen repräsentieren psychologisches Wissen und leiten den Durchführenden
- Analytische Methode, für frühes Design oder existierende Systeme geeignet
- Zeitaufwendig



Folie 296

Methode E2: Heuristische Evaluierung



- Naivere Variante des Cognitive Walkthrough
- Satz von Benutzbarkeitskriterien (Heuristiken, z.B. 9 Goldene Regeln, Patterns) aufstellen
- Dann Tasks durchgehen und Einhaltung der Kriterien prüfen
- Vorteile: schnell und billig
- Nachteil: subjektiv!
 - ➔ Besser durch mehrere Entwickler unabhängig



Folie 297

Methode E3: Recherche



- Viele Forschungsergebnisse über Benutzerschnittstellendesign publiziert
- Idee:
Suche in Fachliteratur nach Evidenz für (oder gegen) Aspekte des Designs
- Vorteil: Eigene Experimente erspart
- Nachteil: Ergebnisse nur bei sehr ähnlichem Kontext übertragbar
 - Versuchspersonen, Annahmen, ...



Folie 298

Methode E4: Modellbasierte Evaluierung



- Es gibt Modelle, die einen Rahmen für Design und Evaluierung bieten
- Beispiele:
 - GOMS, KLM
 - Design Rationale
 - Historie begründeter Entwurfsentscheidungen und -alternativen
 - Pattern Framework



Folie 299

Evaluierung für die Implementierung



- Bisherige Methoden E1-E4 dienen der Evaluierung eines Entwurfs und wurden durch den Entwickler (ohne Benutzer) durchgeführt
- Spätestens sobald eine Implementierung (Prototyp oder fertiges System) vorhanden ist, müssen Benutzer eingebracht werden
- Die folgenden Methoden dienen der Evaluierung von Implementierungen mit Benutzern



Folie 300

Methode I1: Kontrollierte Experimente



- Quantitativ-empirische Methode
- Ablauf:
 - Hypothese formulieren
 - Experiment mit zu variierenden & zu messenden Größen entwickeln
 - Personen auswählen
 - Tests durchführen
 - Ergebnisse interpretieren, um Hypothese verwerfen oder akzeptieren zu können



Folie 301

I1: Kontrollierte Experimente (2)



- Testpersonen
 - Gleich (oder ähnlich) späteren Benutzern in Alter, Ausbildung, Computerkenntnissen, Systemkenntnissen, Expertise im Anwendungsbereich
 - Faustregel: Minimum = 10 Testpersonen
- Variablen
 - Unabhängige: werden kontrolliert verändert
 - z.B. Anzahl der Menüeinträge
 - Abhängige: werden gemessen
 - z.B. Ausführungszeit, Fehlerrate, Vorlieben(!)



Folie 302

I1: Kontrollierte Experimente (3)



- Hypothese
 - Vorhersage des Ergebnisses des Experiments
 - Behauptet, dass Änderung unabhängiger Variablen die Werte der abhängigen Variablen beeinflusst
 - Ziel des Experiments: Hypothese bestätigen
 - Ansatz: Nullhypothese (Gegenteil der Hypothese, d.h. „kein Zusammenhang“) widerlegen



Folie 303

I1: Kontrollierte Experimente (4)



- Zwischen-Gruppen-Test:
 - Jede Person wird nur einer Variante des Experiments ausgesetzt
 - Es gibt mindestens 2 Varianten
 - manipulierte Form + Kontrollgruppe, um Wirkung der Manipulation zu isolieren
 - Kein Lerneffekt über Varianten, braucht aber mehr User



Folie 304

I1: Kontrollierte Experimente (5)



- In-Gruppen-Test:
 - Jede Person macht jede Variante des Experiments mit
 - Braucht weniger User, individuelle Unterschiede ausgeglichen, aber Lerneffektgefahr bei bestimmten Experimenten



Folie 305

I1: Kontrollierte Experimente (6)



- Analyse der Ergebnisse
 - Statistische Auswertung mit Hilfe definierter Testverfahren
 - Beispiele: Students t-Test, ANOVA (Analysis of Variance), Regressionsanalyse, Wilcoxon- oder Mann-Whitney-Test, χ^2 -Test
 - Wahl: je nach Kontinuität, angenommener Verteilung der unabhängigen+ abhängigen Variablen und gesuchter Aussageform
 - Aussageformen: Unterschied ja/nein, Größe der Differenz, Konfidenz der Schätzung



Folie 306

Methode I2: Modellextraktion



- Designer zeigt Benutzer Prototyp oder Bildschirmschnappschüsse
- Benutzer versucht Elemente und ihre Funktion zu erklären
- Gut, um Anfängerbild des Systems zu begreifen
- Schlecht, um Erlernen nachzuvollziehen



Folie 307

I3: Stumme Beobachtung



- Designer beobachtet Benutzer im Labor oder im Feld bei Bearbeitung einer der Tasks
- Keine Kommunikation während der Beobachtung
- Gut, um grobe Probleme zu entdecken
- Aber:
Kein Einblick in Entscheidungsprozesse oder Einstellung des Benutzers



Folie 308

I4: Think Aloud



- Wie I3,
aber Benutzer wird gebeten zu sagen,
 - was er glaubt, dass gerade passiert (Zustand),
 - was er zu erreichen versucht (Ziele),
 - warum er etwas Bestimmtes tut (Aktionen)
- Häufigste Methode in der Industrie
- Nachteile:
 - Unangenehm für viele Benutzer (laut denken ist nicht normal!)
 - Bewusstes Denken kann Verhalten ändern
 - Reden schwierig, wenn man sich gerade auf das Problem konzentriert



Folie 309

I5: Konstruktive Interaktion



- Zwei Personen arbeiten gemeinsam an Task
 - Normale Konversation wird aufgezeichnet
 - Nicht so unangenehm wie Think-Aloud
- Variante: Ungleiche Partner
 - Halbexperte als „Trainer“ + Neuling als „Schüler“
 - Schüler benutzt Interface und fragt
 - Trainer antwortet
 - Gibt Einsicht in Denkvorgänge bei Neulingen und etwas erfahreneren Usern gleichzeitig



Folie 310

Beobachtungen aufzeichnen



- Papier + Bleistift
 - Evaluierer notiert Ereignisse, Interpretationen, sonstige Beobachtungen
 - Billig, aber schwer bei vielen Details
 - Schreiben ist langsam
 - Formulare können helfen



Folie 311

Beobachtungen aufzeichnen



- Audioaufzeichnung
 - Gut für Sprache bei Think-Aloud & Konstruktiver Interaktion
 - Aber schwer mit Programmzustand zu verbinden
- Video
 - Ideal: 2 Kameras (Benutzer + Bildschirm) in 1 Bild
 - Beste Erfassung, aber anfangs evtl. zu aufdringlich



Folie 312

Beispiel - Testlabor



Folie 313

I6: Retrospektives Testen



- Zusatzmaßnahme nach einer Beobachtung
- Benutzer und Evaluierer sehen sich anschließend gemeinsam die Videoaufzeichnung an
- Benutzer kommentiert seine Aktionen in Nachhinein
- Bietet guten Ausgangspunkt für anschließendes Interview
- vermeidet Verfälschung der Erinnerungen
- Liefert oft konkrete Verbesserungshinweise



Folie 314

Andere Evaluierungsmethoden



- Vor und während des Designs mit Benutzern
 - Fragebögen (bereits behandelt)
 - Interviews (persönlich)
- Nach Abschluss der Entwicklung:
 - Kummerkästen
 - Email-Formular im Programm, ...
 - Hotlines
 - Rückblickende Interviews und Fragebögen
 - Feldbeobachtungen
 - existierendes System im realen Einsatz beobachten



Folie 315

Ein paar Worte zum Umgang



- Tests sind unangenehm für Benutzer
 - Leistungsdruck
 - Fehler
 - Konkurrenzdenken
- Daher:
Testpersonen („Tester“) stets mit Respekt behandeln!
 - Vor, während und nach dem Test



Folie 316

Vor dem Test



- Keine Zeit des Testers verschwenden
 - Pilottests vorab, alles fertig, wenn Benutzer kommt
- Sorge dafür, dass sich Tester wohlfühlt
 - Betone, dass das System, nicht er getestet wird!
 - Bestätige, dass das System noch Fehler haben kann
 - Lass Tester wissen, dass er jederzeit aufhören kann
- Gewährleiste Verschwiegenheit
 - Erkläre, dass individuelle Testergebnisse absolut vertraulich behandelt werden



Folie 317

Vor dem Test



- Informiere den Tester
 - Erkläre alle Beobachtungsformen
 - Audio/Video?
 - Beantworte alle anderen Fragen
 - aber Beeinflussung vermeiden
- Verwende nur Freiwillige
 - Einverständniserklärung



Folie 318

Während des Tests



- Keine Zeit des Testers verschwenden
 - Nie unnötige Tasks erledigen lassen
- Sorge dafür, dass sich Tester wohlfühlt
 - Frühes Erfolgserlebnis ermöglichen
 - Entspannte Atmosphäre im Raum
 - Pausen, Kaffee, etc.
 - Testaufgaben einzeln nacheinander ausgeben
 - Nie Unzufriedenheit mit Verhalten des Testers zeigen
 - Unterbrechungen vermeiden (kein Handy, ...)
 - Test abbrechen, falls er zu unangenehm wird



Folie 319

Während des Tests



- Gewährleiste Verschwiegenheit
 - Lasse nie den Chef des Testers (oder andere) zusehen!



Folie 320

Nach dem Test



- Sorge dafür, dass sich Tester wohlfühlt
 - Betone, dass Tester beim Auffinden von Verbesserungsmöglichkeiten mitgeholfen hat
- Informiere den Tester
 - Beantworte Fragen, die vor dem Test das Ergebnis hätten verändern können
- Gewährleiste Verschwiegenheit
 - Nie Ergebnisse veröffentlichen, die einzelnen Benutzern zuzuordnen sind
 - Zeige Aufzeichnungen außerhalb der Forschergruppe nur mit Genehmigung des Testers



Folie 321

Zusammenfassung



- Evaluierung beurteilt Funktionen, Erlern- und Bedienbarkeit eines Systems
- Laborstudien, Feldstudien, Entwurfsbeteiligung
- Entwürfe qualitativ ohne Benutzer evaluieren: Cognitive Walkthrough, Recherche, etc.
- Implementierungen qualitativ mit Benutzer evaluieren: Think-Aloud, Konstruktive Interaktion, etc.



Folie 322

Zusammenfassung



- Implementierungen quantitativ mit Benutzern evaluieren:
Kontrollierte Laborexperimente
 - Hypothese, Nullhypothese, unabh. + abh. Variablen, Benutzer, statistische Testverfahren
- Testpersonen mit Respekt behandeln



Folie 323