

Analytische Evaluation

Ilhan Aslan, Chi Tai Dang, Björn Bittner, Katrin Janowski,
Elisabeth André



Human Centered Multimedia

Institute of Computer Science

Augsburg University

Universitätsstr. 6a

86159 Augsburg, Germany

- **Empirische Evaluation** mit Nutzern
- **Analytische Evaluation** mit:
 - Usability Experten
 - Experten der Domäne
 - Mitgliedern des Projektteams

Analytische Evaluation

- Identifiziert grundlegende bzw. offensichtliche Probleme
- Arten von analytischen Evaluationstechniken:
 - **Inspektionsbasierte Evaluation:**
Durchlaufen der Tasks und Überprüfung anhand von Richtlinien und Checklisten
 - **Modellbasierte Evaluation:**
Durchlaufen der Tasks anhand von empirisch ermittelten Modellen

Analytische Evaluation: Inspektionstechniken



Human Centered Multimedia

Institute of Computer Science

Augsburg University

Universitätsstr. 6a

86159 Augsburg, Germany

- **Guideline-based:**
 - Überprüfung anhand von Richtlinien und Prinzipien (z.B. Heuristiken und Checklisten)
- **Walkthrough-based:**
 - Simulation einer Nutzerstudie
 - Durchlaufen der definierten Tasks durch die Experten
- **Kombination:**
 - Durchlaufen der definierten Tasks durch die Experten
 - Überprüfung anhand von Richtlinien und Prinzipien (z.B. Heuristiken und Checklisten)

- Inspektionen werden oft informell durchgeführt
 - Regelmäßig während der Entwicklungsphase (Design und Implementierung)
 - Typischerweise vor echten Nutzerstudien
- Hilfreiche inspektionsbasierte Evaluationen sind
 - klar und eindeutig dokumentiert!
 - wichtig, um die Qualität des Prototypen zu verbessern
- Hilfreiche inspektionsbasierte Evaluationen bieten
 - klare Beschreibungen vorhandener Probleme und eventuell auch deren Bedeutung
 - direkten Input für Änderungen und eventuell auch konkrete Vorschläge für Verbesserungen

- Beispiele:
 - Expertenreview
 - Heuristische Evaluation
 - Checklist Review
- Anwendung der Methoden je nach Budget und Nutzen in dem jeweiligen Projekt

- Werkzeug zum Finden von Problemen anhand von:
 - speziellen Prinzipien und Guidelines
 - Wissen und Erfahrung der Usability-Experten
- Strukturierter Ansatz
 - Hauptaufgabe ist es Probleme zu finden
 - Teilnehmer:
 - müssen offensiv sein
 - kommunizieren die Probleme
 - können Lösungen vorschlagen
 - Aber: Entscheidungen über Änderungen trifft das Team

Meist werden viele Experten eingesetzt, um alle Fehler zu finden

- Sehr umfassende Überprüfung der Nutzerschnittstelle
- Ausführliche Dokumentation mit Fehlerbeschreibung und möglichen Lösungsschritten
- Meistens teuer und aufwendig durchzuführen!

- Discountmethode des Expertenreviews
- Grundlegende Idee:
 - **5 Experten** reichen, um **75%** der Usability-Probleme zu finden!!!
 - Wenige Leute evaluieren das Interface und bewerten es anhand von Usability-Prinzipien (Heuristiken)
 - Entweder reine Inspektion anhand von Heuristiken oder auch task-basierter Durchlauf
 - Erzeugung einer Problemliste, deren Punkte bewertet werden

Beispiel: Zehn Usability Heuristiken von Nielsen

1. Sichtbarkeit des Systemstatus
2. Match zwischen System und realer Welt
3. Nutzerkontrolle und –freiheit
4. Konsistenz und Standards
5. Fehlervermeidung
6. Erkennen anstatt Erinnern
7. Flexibilität und Effizienz der Nutzung
8. Ästhetisches und minimalistisches Design
9. Hilfe für den Nutzer zur Erkennung, Bewertung und Behebung von Fehlern
10. Hilfe und Dokumentation

Ablauf

- Nacheinander jede Heuristik einzeln betrachten und danach das System evaluieren.
- Zu jeder Heuristik kurz schildern welche Probleme in welchem Zusammenhang auftreten
- Beurteilung und Dringlichkeit der Probleme anhand vorgegebener Bewertungskriterien (siehe nächste Folien)
- Optional: Vorschlag einer Verbesserung oder Lösung

Bewertung der Probleme

- Drei Faktoren:
 1. Auftreten: Wo tritt das Problem auf? (z.B. Startseite)
 2. Häufigkeit: Wie oft tritt das Problem auf?
 3. Auswirkung: Wie wirkt sich das Problem aus? (z.B. Systemabsturz)
- Helfen für jedes einzelne Problem der Heuristik herauszufinden wie wichtig es ist es zu lösen
- Bieten direktes Feedback

Bewertung der Probleme

- Werte für die Beurteilung eines Problems:
 - 0: Kein Usability-Problem
 - 1: Kosmetisches Problem
 - 2: Geringes Problem: Geringe Priorität
 - 3: Grosses Problem: Wichtig zu beheben (hohe Priorität)
 - 4: Usability Katastrophe: Unbedingt vor dem Release zu beheben!!

- Heuristik 7: Flexibilität und Effizienz der Nutzung
- 1. Problem: umständlicher Funktionswechsel
 - a) Bewertungsfaktoren:
 - a) *Auftreten*: Um eine andere Funktion auszuwählen, musste umständlich der Weg über „Zurück“ -> „Zurück“ -> „andere Funktion“ gewählt werden.
 - b) *Häufigkeit*: Dies wurde bei beiden möglichen Funktionen festgestellt. Diese Funktionen werden häufig genutzt.
 - c) *Auswirkung*: Die Bedienung ist ineffizient. Dies kann bei häufigerer Nutzung zu Unzufriedenheit beim Nutzer führen.
 - b) Verbesserungsvorschlag:

Verwendung einer Roadmap => schnellere Navigation möglich, da z.B. auch die vorvorherige Seite ausgewählt werden kann.
 - c) Bewertung: **3**

Fazit:

- Billiger als Expertenreview
- Schnell und einfach anhand der Heuristiken
- Klares Feedback für das Team durch Bewertungen und eventuelle Lösungsvorschläge

Checklisten basieren auf Prinzipien oder Richtlinien

- Oft in Oberkategorien eingeteilt (z.B. Navigation, Content)
- Überprüfung der Einhaltung bzw. Nichteinhaltung
- Information zum Problem über Kommentare

GUI SCREEN DESIGN CHECKLIST				
	Yes	No	N/A	Rema
Window Components (including menus and screen titles)				
Is the correct window type used (e.g., primary window, dialog box, property sheet)?				
Are basic components (e.g., title bars, horizontal and vertical scroll bars) used consistently and according to established standards?				
Does the screen title identify the				

Navigation		
12. Main navigation is easily identifiable	✓	✓ x
13. Navigation labels are clear & concise	✓	✓ x
14. Number of buttons/links is reasonable	✓	✓ x
15. Company logo is linked to home-page	✓	✓ x
16. Links are consistent & easy to identify	✓	✓ x
17. Site search is easy to access	✓	✓ x
Content		
18. Major headings are clear & descriptive	✓	✓ x
19. Critical content is above the "fold"	✓	✓ x

78. Did you make sure that music and videoclips can be turned off at any time?
79. Did you inform the user about the size and length of your music and video?
Extra features
80. Did you make a custom 404 page?
81. Did you make a site map?
82. Did you make a rss feed? (if you add content regularly)
83. Did you make a high contrast version of your website?
84. Did you make good looking URL's (like this one)?
85. Did you make sure that there are no frames on your website?
86. Did you make sure that there are no pop ups on your website?
87. Did you make a contact form instead of just an email link?

Fazit:

- Sehr schnell anzuwendende Methode
- Wenig Fehler möglich
- Aber: Wenige Informationen über Gewichtungen und Lösungen der Probleme

Fazit:

- Ergebnisse hängen stark von Erfahrung und Wissen des Evaluators ab
- Schnelle Ergebnisse bzw. schnelles Feedback
- Aber: keine nutzerspezifischen Probleme in der realen Nutzung zu erkennen!

- Nutzung von Szenarien für die Inspektion
- Oft in Kombination mit Walkthrough-Methoden:
 - Simulation einer empirischen Evaluation
 - Durchlaufen der Nutzerschnittstelle anhand von Tasks
 - Hineinversetzen in die Nutzer
 - Führt zu realistischeren Ergebnissen
 - Beispiel:
 - Cognitive Walkthrough

- Einer oder mehrere Evaluatoren durchlaufen verschiedene Tasks
- Durchführung:
 1. Definition der grundlegenden Informationen
 - Wer werden die Nutzer des Systems sein?
 - Personas
 - Welche Tasks sollen analysiert werden?
 - Szenarios
 - Was sind die korrekten Aktionssequenzen für diese Tasks?
 - Taskmodellierung (z.B. Hierarchische Taskanalyse)
 - Wie ist die Nutzerschnittstelle definiert?
 - Prototyp

2. Während des Walkthroughs an jeder Stelle Beantwortung der folgenden Fragen:
- Wird der Nutzer erkennen welche Aktionsmöglichkeiten vorhanden sind?
 - Wird der Nutzer wissen welche Aktionsmöglichkeit er nutzen muss, um dem Ziel einen Schritt näher zu kommen?
 - Wird der Nutzer den möglichen Aktionen die richtigen Auswirkungen zuordnen können?
 - Wenn die richtige Aktion ausgeführt wurde, wird der Nutzer sofort den Systemfortschritt erkennen können und dass er dem Ziel einen Schritt näher gekommen ist?

Vergleiche 7 Handlungsschritte von Norman!!

Fazit:

- Man muss sich in den Nutzer hineinversetzen können
- Viel Wissen über den Nutzer nötig
- Fehlinterpretationen unausweichlich
- Während der Entwicklung von neuen Features/Tasks sinnvoll

- Analytische Methoden sind VOR den Nutzertests sinnvoll
 - Erkennen und Beheben grober Probleme der Usability
 - Nutzer in empirischen Studien nicht unnötig abgelenkt
- Analytische Methoden sind typischerweise billiger und schneller durchzuführen als empirische Methoden.

ABER!!

- Kein Ersatz für Nutzertests, sondern nur Ergänzung
- Tatsächliche nutzerspezifische Probleme NUR durch Tests mit Nutzern aus der Zielgruppe erkennbar
- Allerdings lieber analytische Methoden als gar keine Usability-Tests

Evaluation mit Nutzern und Experten

-

Pluralistic Walkthrough



Human Centered Multimedia

Institute of Computer Science

Augsburg University

Universitätsstr. 6a

86159 Augsburg, Germany

Grundlagen:

- Gruppendiskussion mit Usability-Experten, Entwicklern und Nutzern
- Durchlaufen der Nutzerschnittstelle anhand von Tasks
- Fragestellungen ähnlich wie bei Cognitive Walkthrough
- Meist mit Papierprototypen oder Storyboards während Konzept- bzw. Entwicklungsphase
- Teilweise auch als Qualitätskontrolle oder bei der Überarbeitung bestehender Systeme (Re-Design)

Ablauf:

1. Usability-Experten übernehmen Moderation
2. Entwickler geben kurze Einleitung in die Anwendung
3. Nutzer führen alleine einen vorgegebenen Task mit dem Prototypen aus (Aufschreiben der Aktionen)
4. Entwickler „simulieren“ Hilfefunktionen
5. Usability-Experten präsentieren nach jedem Durchlauf die möglichen „Lösungen“
6. Anschließende Diskussion:
 - a) Nutzer präsentieren und diskutieren aufgetretene Probleme
 - b) Entwickler dürfen erst gegen Ende der Diskussion einsteigen
 - c) Usability-Experten können Problemlösungen vorstellen
7. Nächster Task

Fazit:

- Evaluation in einer sehr frühen Entwicklungsphase möglich
- Kein interaktiver Prototyp vorausgesetzt
- Entwickler erhalten bessere Einsicht in die tatsächliche Nutzung ihrer Ideen bzw. Konzepte
- Viele sehr qualifizierte Erkenntnisse
- Auftretende Fragen können direkt geklärt werden.
- Änderungen können direkt vor Ort und im Konsens besprochen werden
- Entwickler müssen offen gegenüber Kommentaren sein!