

Lehrplan

CPUE - SL

**Certified Professional for Usability and
User Experience Engineering
Specialist Level**

Version 1.3 DE

User Experience Quality Certification Center

UXQCC

Änderungsübersicht der deutschsprachigen Ausgabe

Version	Datum	Bemerkung
1.3	21. Jan. 2019	Überarbeitung Lean UX
1.2	18. Okt. 2018	Einbeziehung UXQCC Board
1.1	12. Nov. 2017	Grundlegende Überarbeitung und Erweiterung
1.0	01. Juli 2017	Release Version

Wissenschaftliches Gremium

Das wissenschaftliche Gremium von IBUQ sowie von UXQCC besteht aus renommierten Wissenschaftlern, Vertretern relevanter Organisationen und Unternehmen die sich mit Userexperience sowie Usability-relevanten Themenfeldern beschäftigen. Das Gremium unterstützt die Weiterentwicklung des Lehrplanes in didaktischer und inhaltlicher Sicht. Damit wird sichergestellt, dass die Inhalte aus wissenschaftlicher und aus berufspraktischer Sicht aktuell, relevant und anwendbar sind. Die aktuelle Zusammensetzung des Gremiums kann den Websites von IBUQ (<https://www.ibuq.org>) und UXQCC (<https://www.uxqcc.com>) entnommen werden.

Mitarbeit an der Gestaltung der Inhalte

Besonderen Dank gebührt an dieser Stelle Herrn Dominik Dolezal, MSc, Usability Labor der Fachhochschule Technikum Wien für seine wesentlichen Beiträge zum Inhalt.

Einführung

1) Zweck des Dokuments

Dieser Lehrplan definiert den Lehrplan zum Certified Professional for Usability and User Experience Engineering - Specialist Level. Das Programm wird vom User Experience Quality Certification Center (UXQCC) und vom International Board for Usability und User Experience Qualification (IBUQ) weiterentwickelt. Dieser Lehrplan wird akkreditierten Weiterbildungsanbietern zur Verfügung gestellt, damit sie eine Version in den jeweiligen Landessprachen erarbeiten und Kursunterlagen erstellen können. Die Lernenden bereiten sich anhand des Lehrplans auf die Zertifizierungsprüfung vor. Die angegebenen Literatur sollte für die Prüfungsvorbereitung ebenfalls benutzt werden.

Der Lehrplan setzt die Inhalte des Foundation-Levels und des Advanced-Levels voraus. Diese Inhalte werden nicht wiederholt.

2) Der Certified Professional for Usability und User Experience Specialist Level, Specialist Level

	Ziele
Neue Schlüsselqualifikationen erwerben	Mobile Applikationen, Softwareprodukte oder Webseiten müssen die ihnen zugedachten Ziele und Aufgaben erfüllen. Usability und User-Experience als Schlüsselkompetenzen garantieren die Zufriedenheit der Nutzer bei Verwendung des Produktes.
Ausbildungsstandard	Der <i>Certified Professional for Usability und User Experience Specialist Level</i> (CPUE SL) besitzt die notwendigen Fertigkeiten und das Wissen, um Softwareentwicklungsprozesse sowie weitere benötigte Prozesse zur Erzielung höchster Usability und optimaler User-Experience im Unternehmen einzuführen, argumentieren und gestalten zu können.
	Nutzen
Steigerung der Zufriedenheit Ihrer	Die Erfüllung von Leistungserwartungen und ihre Wahrnehmung des Kunden führt zu einer Kundenzufriedenheit. Die verbesserte Usability von

Kunden	Software- und Internetapplikationen führt zur Reduzierung von Diskrepanz zwischen erwarteter und wahrgenommener Leistung und fördert die Kundenbindung.
Folgekosten minimieren	Usability-Maßnahmen sollten nicht erst nach Launch oder Relaunch einer Webseite bzw. dem Verkaufsstart eines Softwareprodukts ergriffen werden. Dadurch wird ein Imageschaden oder Besucher- bzw. Kundenverlust vermieden und die Kosten für spätere Nachbesserungen und Korrekturen werden reduziert.
Wettbewerbsvorteile	Die Gewinnung der angestrebten Zielgruppen wird durch die Benutzerfreundlichkeit nicht nur erleichtert, sondern diese hebt die Produkte und Dienstleistungen des Anbieters auch von denen der Konkurrenz ab.
Vertrauensbildung	Die Bedürfnisse der Nutzer werden ernst genommen und diese fühlen sich mit dem Internet- oder Softwareangebot wohler. Dies stärkt die positive Einstellung gegenüber dem Anbieter und der Marke und sorgt für eine verbesserte Kundenbindung.
Schwerpunkte	
Der Weg zu effizienter Usability und zu effizientem User-Experience-Engineering	Was bedeutet gute Usability und durch welche Vorgehensweisen kann sie erreicht werden? Es wird zusätzlich auf gängige Irrtümer und deren Behebung eingegangen.
Prozesse im Unternehmen und im Software Engineering auf Usability und User Experience optimieren	Welche Prozesse sind notwendig, um Produkte so zu gestalten, dass die Anwender zufrieden sind? Wie können diese Prozesse in standardisierte Vorgehensmodelle integriert werden? Dieser wichtige Teil des Programms schlägt eine Brücke zwischen Requirements Engineering, Projektmanagement und User-Experience.
Übungen	Übungen verstärken das Verstehen und ermöglichen es Absolventinnen und Absolventen, das erworbene Wissen unmittelbar im Unternehmen umzusetzen.
Quantifizierung von Ergebnissen	Return of Investment. Aufwand, Kosten und Nutzen abschätzen lernen. Quantitative Messungen der Verbesserung der Usability durchführen.

Der Certified Professional for Usability und User Experience - Specialist Level, spricht alle in das Thema Entwicklung von Software- oder Internetapplikationen involvierten Personen und Berufsfelder an, beispielsweise Software-Entwickler, GUI-Programmierer, Projektleiter und -Mitarbeiter, Organisatoren, Führungskräfte, Mitarbeiter der Fachabteilungen, IT-Revisoren, Qualitätssicherungsbeauftragte.

Der Lehrplan ist auf modernen schlanken Methoden aufgebaut und beinhaltet die neuesten Entwicklungen auf dem Gebiet des Usability-Engineerings. Die unterrichteten Methoden werden von nahezu allen Unternehmen, die erfolgreich Produkte mit höchster Usability entwickeln, angewandt. Diese Methoden stellen sicher, dass Absolventinnen und Absolventen effizient und effektiv die Entwicklungsprozesse im Unternehmen in Hinblick auf Usability optimieren können.

Besonderes Gewicht wird auf die Vermittlung der notwendigen Fertigkeiten gelegt, Usability-Engineering in Unternehmen einzuführen bzw. optimieren zu können. Die Zertifizierung stellt sicher, dass das nötige Wissen über das Fachthema im Kontext des Entwicklungsprozesses angewandt werden kann. Die notwendigen Schnittstellen zum Requirements Engineering sowie zu Methoden des klassischen und agilen Softwareprojektmanagements werden ebenfalls behandelt.

Grundlegende Erfahrungen in der Entwicklung von technischen Produkten, wie mobile Applikationen, Software o. Ä. werden vorausgesetzt. Das Foundation-Level-Zertifikat und das Advanced-Level-Zertifikat sind Voraussetzung, um die Zertifikatsprüfung zum Usability Professional Specialist Level zu absolvieren.

Wie ist Usability messbar und welcher Aufwand ist notwendig, um Kunden tatsächlich zufrieden zu machen? Diese zentrale Frage können Absolventen des Specialist-Levels fundiert beantworten. Dies befähigt sie, die dauerhafte Implementierung der notwendigen Prozesse innerhalb des Unternehmens kompetent zu argumentieren und Skeptiker zu überzeugen.

3) Lernziele / Kognitive Stufen des Wissens

Jeder Abschnitt dieses Lehrplans ist einer kognitiven Stufe zugeordnet:

Stufen im Foundation- und im Advanced Level:

K1 Kenntnisse / Wissen: Kenntnisse konkreter Einzelheiten wie Begriffe, Definitionen, Fakten, Daten, Regeln, Gesetzmäßigkeiten, Theorien, Merkmale, Kriterien, Abläufe; Lernende können Wissen abrufen und wiedergeben.

K2 Verstehen: Lernende können Sachverhalte mit eigenen Worten erklären oder zusammenfassen; können Beispiele anführen, Zusammenhänge verstehen; können Aufgabenstellungen interpretieren. Dazu gehört, dass Inhalte von einer Darstellungsart in eine andere übertragen (z. B. Worte in eine Grafik), dass Inhalte erklärt und zusammengefasst und dass schließlich zukünftige Entwicklungen aus Inhalten abgeleitet werden können.

Stufen im Advanced Level:

K3 Anwenden: Transfer des Wissens, problemlösend; Lernende können das Gelernte in neuen Situationen anwenden und unaufgefordert Abstraktionen verwenden oder abstrahieren. Fähigkeit, den gelernten Stoff in neuen konkreten Situationen einzusetzen, indem z. B. bestimmte Regeln, Gesetze, Theorien etc. angewendet werden. So soll z. B. eine Informatikstudentin in der Lage sein, verschiedene Sortieralgorithmen in einer Assemblersprache zu programmieren, oder ein Mathematikstudent muss einen mathematischen Beweis nach den geltenden Regeln führen können.

K4 Analyse: Lernende können ein Problem in einzelne Teile zerlegen und so die Struktur des Problems verstehen; sie können Widersprüche aufdecken, Zusammenhänge erkennen und Folgerungen ableiten und zwischen Fakten und Interpretationen unterscheiden. Dazu gehört beispielsweise, die einzelnen Elemente zu identifizieren, die Beziehungen zwischen den Elementen festzustellen und die Gestaltungsprinzipien zu erkennen. Die Stufe Analysieren verlangt ein höheres Fähigkeitsniveau als Verstehen und Anwenden, weil sie voraussetzt, dass sowohl der Inhalt als auch die Struktur des Lernstoffes verstanden sind. So gehört etwa die Lernaktivität von Studierenden der Kunstgeschichte, die Stil bestimmenden Elemente eines Gemäldes zu eruieren und einer spezifischen kunstgeschichtlichen Epoche zuzuordnen, auf diese Stufe.

Stufen im Specialist Level:

K5 Synthese: Lernende können aus mehreren Elementen eine neue Struktur aufbauen oder eine neue Bedeutung erschaffen, können neue Lösungswege vorschlagen, neue Schemata oder begründete Hypothesen entwerfen.

K6 Beurteilung: Lernende können den Wert von Ideen und Materialien beurteilen und können damit Alternativen gegeneinander abwägen, auswählen, Entschlüsse fassen und begründen und bewusst Wissen zu anderen transferieren, z. B. durch Arbeitspläne.

4) Die Prüfung

Auf diesem Lehrplan basiert die Prüfung für das Specialist-Level-Zertifikat. Eine Prüfungsfrage kann Stoff aus mehreren Kapiteln des Lehrplans abfragen. Alle Abschnitte dieses Lehrplans können geprüft werden.

Das Format der Prüfung ist Multiple Choice.

Prüfungen können unmittelbar im Anschluss an einen akkreditierten Ausbildungslehrgang oder Kurs, aber auch unabhängig davon abgelegt werden. Prüfungen können über die WebSite von UXQCC gebucht werden (<https://www.uxqcc.org>) Der weltweit zugelassene Prüfungsanbieter ist gasq (Global Association for Software Quality).

5) Akkreditierung

Ausbildungsanbieter, deren Ausbildungsunterlagen entsprechend diesem Lehrplan aufgebaut sind, müssen durch UXQCC anerkannt und akkreditiert werden.

6) Detaillierungsgrad

Ziel des Lehrplans ist es, ein international konsistentes Lehren und Prüfen zu gestatten. Zur Erreichung dieses Zieles beinhaltet dieser Lehrplan folgende Bestandteile:

- Allgemeine Lernziele, welche die Intention des Specialist Levels beschreiben
- Inhalte, die zu lehren sind, mit einer Beschreibung und wo notwendig Referenzen zu weiterführender Literatur
- Lernziele für jeden Wissensbereich, welche das beobachtbare kognitive Ergebnis der Schulung und die zu erzielende Einstellung des Teilnehmers beschreiben
- Eine Liste von Begriffen, welche der Teilnehmer wiedergeben und verstehen soll
- Eine Beschreibung der wichtigen zu lehrenden Konzepte, inklusive der Quellen wie anerkannte Fachliteratur, Normen und Standards

Der Lehrplan ist keine vollständige Beschreibung des Wissensgebiets „Usability“. Er reflektiert lediglich den nötigen Umfang und Detaillierungsgrad, welcher für die Lehrziele des Specialist Levels relevant ist.

7) Lehrplanaufbau

Der Lehrplan besteht aus Hauptkapitel mit weiterer Unterteilung. Jeder Haupttitel eines Kapitels zeigt die Lernzielkategorie, welche mit dem jeweiligen Kapitel abgedeckt werden soll, und legt die Unterrichtszeit fest, welche in einem akkreditierten Kurs mindestens für dieses Kapitel aufgewendet werden muss.

Beispiel für den Lehrplanaufbau:

2**Prinzipien, Prozesse und Verfahren (K4)**

390 Minuten

Das Beispiel zeigt, dass in Kapitel 2 Lernziele K4 (ein Lernziel einer höheren Taxonomiestufe impliziert die Lernziele der tieferen Taxonomiestufen) und K1 - K3 (aber nicht K5) erwartet werden und 390 Minuten für das Lehren des Materials in diesem Kapitel vorgesehen sind.

Jedes Kapitel enthält eine Anzahl von Unterkapiteln. Jedes Unterkapitel kann wiederum Lernziele und einen Zeitrahmen vorgeben. Wird bei einem Unterkapitel keine Zeit angegeben, so ist diese im Oberkapitel bereits enthalten.

Lehrplanstruktur

Minimale Unterrichtszeit im Kurs:

1 Tag zu 8 h Unterrichtszeit (insgesamt 480 Minuten).

Empfohlene zusätzliche Selbstreflexionszeit zur Prüfungsvorbereitung:

1 Tag, 480 min

Diese Reflexion ist vor allem zur Festigung der erworbenen Fertigkeiten und des Wissens und nicht zuletzt zum Bestehen der Prüfung dringend empfohlen.

Tag 1 (480 Minuten)

1

Der Weg zu effizientem Usability-Engineering im Unternehmen (K3)

30 Minuten

1.1. Warum User-Experience wichtig ist (K3) – 2 LO (15 Minuten)

1.2. Prinzipien, die zu guter User-Experience führen (K3) – 2 LO (15 Minuten)

2

Prinzipien, Prozesse und Verfahren (K4)

150 Minuten

2.1 Schlanke Methoden (Lean UX) in den Entwicklungsprozess integrieren – 2 LO (20 Minuten)

2.2 Usability und Unternehmensziele – 3 LO (30 Minuten)

2.3 Nutzerbedürfnisse erfassen – 2 LO (30 Minuten)

2.4 Minimum-Viable-Products (MVPs) und Usability-Engineering – 2 LO (30 Minuten)

2.5 Usability-Engineering in Softwareentwicklungsprozesse integrieren – 2 LO (40 Minuten)

3

Vertiefende Übungen (K6)

60 Minuten

3.1 Kollaboratives Design – 1 LO (60 Minuten)

4 Quantifizierung von Ergebnissen (K5)

240 Minuten

4.1 Return on Investment (ROI)

4.2 Angaben zur objektive Zielerreichung

4.2.1 Statistische Aussagekraft von Testmethoden

4.2.2 Wann ist ein Design objektiv besser als ein konkurrierendes Design

4.3 Wie viele Test werden benötigt, um aussagekräftige Ergebnisse zu bekommen

4.4 Standardisierte Befragungsmethoden

Lehrplan

1

Der Weg zu effizientem Usability-Engineering im Unternehmen (K3)

30 Minuten

1.1. Warum User-Experience wichtig ist. (K3) – 2 LO (15 Minuten)

LO-1.1.1 User-Experience erzeugt Gefühle (K3)

LO-1.1.2 User-Experience und mentale Modelle (K2)

1.2. Prinzipien, die zu guter User-Experience führen (K3) – 2 LO (15 Minuten)

LO-1.2.1 Herausforderungen im Softwareengineering - Lean UX (K2)

LO-1.2.2 Grundlegende Prinzipien (K4)

1.1

Warum User-Experience wichtig ist. (K3)

15 Minuten

1.1.1 User-Experience erzeugt Gefühle (K3)

10 Minuten

Begriffe

User-Experience, Entstehung von Gefühlen, Analyse von Verwendungsszenarien, unbewusste/bewusste Kaufentscheidungen

Ein Produkt soll derart gestaltet sein, dass seine Benutzung ein positives Gefühl beim Benutzer hinterlässt. Einfache Beispiele zeigen, dass oft das Gegenteil der Fall ist. Die Verwendung des Gerätes oder der Software führt zur Frustration, weil die Ziele mit dem Wissen und den Fähigkeiten des Anwenders nicht erreicht bzw. nur unter Mühen und Plagen erreicht werden können.

Ein Beispiel wäre der Anschluss eines digitalen Beamers an einen Computer. Jedes Beispiel kann in wichtige Komponenten zerlegt werden (Analyse). Eine Analyse soll die Diskrepanz zwischen Erwartung und Erleben aufzeigen.

Anwender verwenden technische Geräte und Software, weil sie etwas erreichen wollen.

Frage: Was wollte der Anwender mit dem Anschluss des Beamers erreichen?

Mögliche Antwort: Der Anwender wollte eine vorbereitete Präsentation einem größeren Publikum zugänglich machen.

Frage: Was muss der Anwender konkret tun, um sein Ziel zu erreichen?

Mögliche Antwort: Den Laptop mit einem geeigneten Kabel mit dem Beamer verbinden.

Frage: Was erlebt der Anwender?

Mögliche Antwort: Die Auflösung ist schlecht, der Ton funktioniert nicht, die Anordnung der Icons am Laptop wird verändert, kein Bild erscheint am Beamer.

Derartige Probleme erzeugen auf völlig unbewusster Ebene und damit unbeflussbar negative Gefühle. Diese negativen Gefühle werden automatisch verschiedenen Personengruppen zugeordnet. Mögliche Gruppen sind u. a. die Hersteller des Beamers, Personen die den Raum ausgerüstet haben, das Unternehmen, welches den Raum zur Verfügung gestellt hat, die Person, die den technischen Support übernommen hat.

Negative Gefühle erzeugen Ablehnung, bei Erkennen einer Alternative wird eine Kaufentscheidung in Zukunft mit beträchtlicher Wahrscheinlichkeit *zu Ungunsten* derjenigen Gruppe von Personen (= Unternehmen), auf die die negativen Gefühle projiziert wurden, ausfallen.

Herausragende positive Erlebnisse werden eine Kaufentscheidung in Zukunft mit beträchtlicher Wahrscheinlichkeit *zu Gunsten* der betroffenen Gruppe von Personen (= Unternehmen) ausfallen lassen.

Referenz

[1], [21],[22]

1.1.2 User-Experience und mentale Modelle (K2)

5 Minuten

Begriffe

Mentale Modelle, Erwartungen von Anwendern, User-Experience

Applikationen und Anwendungen müssen heute den Kundenerwartungen entsprechen und leicht und intuitiv nutzbar und verständlich sein. Dieser Satz lässt sich einfach umformulieren:

A user interface is well designed when the program behaves exactly how the user thought it would.

Sinngemäß übersetzt:

Ein Userinterface ist dann gut, wenn das Programm sich exakt so verhält, wie der Anwender angenommen hat, dass es sich verhalten wird.

Annahmen der Anwender sind ausschließlich in der komplexen psychischen Struktur von Anwendern zu suchen und nicht im guten oder schlechten Design von Interfaces. User-Experience ist demnach von den Anwendern abhängig.

Referenz

[1]

1.2	Prinzipien, die zu optimaler User-Experience führen (K3)	15 Minuten
1.2.1	Herausforderungen im Softwareengineering - Lean UX (K2)	5 Minuten

Begriffe

Softwareentwicklungsprozesse und Vorgehensmodelle

Modernes Softwareengineering unterscheidet sich in vielen Punkten von traditionellen Methoden.

- Meist Auslieferung der Software über das Internet
- Sehr kurze Release-Zyklen (oft Tage, manchmal Minuten)
- Agile Methoden in der Entwicklung
- Marktvorteile durch kurze Zyklen
 - o Schnelles Feedback durch Kunden
 - o Kurze Zeiten, um Feedback in das Produkt einfließen lassen
 - o Iteratives Vorgehen
 - o Extrem gestiegene Kundenerwartungen an die Flexibilität des Prozesses
 - o Minimierung von „requirement creep“

Beispiel:

Release-Zyklen von Android Apps, Feedback durch z. B. Beurteilung der Apps durch Kunden.

Nahezu jede beliebige App kann als Beispiel herangezogen werden. Z. B. die App der Fluggesellschaft SWISS.

Die Folgen:

- Keine Zeit für den traditionellen Ansatz, das Produkt völlig fertig zu spezifizieren, bevor mit der Entwicklung begonnen wird
- Notwendigkeit, auf Änderungswünsche kurzfristig kompetent reagieren zu können
- Spezifikationen werden oft während des Entwicklungsprozesses obsolet, da essentielle Kundenerfordernisse erst während des Prozesses bekannt werden

Ein Lösungsansatz, um gute Usability unter obigen Bedingungen zu erreichen, wird als Lean UX (UX = User-Experience) bezeichnet.

Referenz

[25], [26]

1.2.2 Grundlegende Prinzipien (K3)

10 Minuten

Begriffe

Teamstruktur, Ergebnisdefinition, unnötiger Ballast, GOOB

Um ein Softwareprodukt in dieser schnelllebigen Entwicklungswelt kundenorientiert entwickeln zu können, sind verschiedene Prinzipien notwendig. Diese Prinzipien leiten sich einerseits aus der Natur der Softwareentwicklungsprozesse, der psychologischen Eigenschaften von Menschen (Entwicklungsteam und Kunden) sowie der Struktur von Unternehmen ab.

1. Teamstruktur

- Notwendiger Mix von Fertigkeiten und Kenntnissen im Team („Cross-Functional Teams“)
 - Softwareentwicklung
 - Produkt-Management
 - Interaction-Design
 - Visual Design
 - Content-Strategie
 - Marketing
 - Qualitätssicherung
- Nicht mehr als 10 Kernpersonen
- Möglichst wenig gleichzeitige Projekte
- Möglichst alle Personen an einem Ort

2. Definition des Ergebnisses

- Ergebnisse des gesamten Prozesses sind das Erreichen von Geschäftszielen und nicht die Menge an Output
- Teams sollen ein Geschäftsproblem lösen und nicht ein Set an Features implementieren

3. Unnötigen Ballast vermeiden
 - Wenn es nicht unmittelbar zum Ergebnis beiträgt
 - Nur Designs verfolgen, die tatsächlich umgesetzt werden
 - Kontinuierliche Entdeckung von Anforderungen (Ziel ist es, den Anwender tatsächlich zu verstehen, und dies ist ein kontinuierlicher Prozess, der nie abgeschlossen ist)
4. GOOB („Get out of the building“)
 - Steve Blank, “Es gibt keine Fakten im Büro, also gehen Sie nach draußen und sprechen Sie mit den Kunden!“

Referenz

[25], [27], [28]

2

Prinzipien, Prozesse und Verfahren – Teil 1 (K4)

150 Minuten

2.1 Schlanke Methoden (Lean UX) in den Entwicklungsprozess integrieren – 2 LO (20 Minuten)

- | | |
|----------|--|
| LO-2.1.1 | Gemeinsames Verstehen im Team (K2) – 10 min |
| LO-2.1.2 | Eigenschaften kreativer Prozesse (K3) – 10 min |

2.2 Usability und Unternehmensziele – 3 LO (30 Minuten)

- | | |
|----------|---|
| LO-2.2.1 | Fundierte Annahmen als Basis zur Steigerung der Usability (K4) – 10 min |
| LO-2.2.2 | Annahmen als Ausgangsbasis (K4) – 10 min |
| LO-2.2.3 | Projekt-Faktoren (K4) – 10 min |

2.3 Nutzerbedürfnisse erfassen – 2 LO (30 Minuten)

- | | |
|----------|--|
| LO-2.3.1 | Requirements-Engineering und Usability (K2) – 20 Minuten |
| LO-2.3.2 | Usability-Engineering im Entwicklungsprozess (K2) – 10 Minuten |

2.4 Minimum-Viable-Products (MVPs) – 2 LO 30 Minuten

LO-2.4.1	Minimum-Viable-Products-Grundlagen (K2) – 15 Minuten
LO-2.4.2	Minimum-Viable-Products und Usability-Experimente (K4) – 15 Minuten

2.5 Usability-Engineering in Softwareentwicklungsprozesse integrieren – 2 LO 40 Minuten

LO-2.5.1	Usability und klassische Softwareentwicklungsverfahren (K4) – 20 Minuten
LO-2.5.2	Usability und agile Softwareentwicklungsverfahren (K4) – 20 Minuten

2.1

Schlanke Methoden (Lean UX) in den Entwicklungsprozess integrieren

20 Minuten

LO-2.1.1	Gemeinsames Verstehen im Team (K2) – 10 min
----------	---

Begriffe

Teamverständnis, Wissen teilen

Entwicklungsteams müssen ein gemeinsames Verständnis über das Produkt entwickeln. Dieses gemeinsame Verständnis entsteht in einem kontinuierlichen Prozess. Gemeinsames Verständnis entsteht per se nicht aus einer Spezifikation. Das gesamte Team benötigt Wissen über:

- Was wird genau getan?
- Warum wird es genauso gemacht?

Umfangreiche externe Dokumentation wird selten gelesen und noch seltener verstanden.

Häufig wird einfach an einem Detail so gearbeitet, wie „es immer gemacht wurde“. Gewohnheiten halten sich hartnäckig.

Gemeinsames Wissen kann durch gemeinsame Diskussionen gefördert werden. Auch ruhige Teammitglieder mit einbeziehen. Hier verbirgt sich oft tiefes Wissen. Es entstehen neue gemeinsame Ideen, die weit mehr sind als die Summe der Einzelteile. Damit wird verhindert, dass Wissen

- nur im eigenen Kopf vorhanden ist,

- nur in einer Dokumentation nachzulesen ist.

Referenz

[25]

LO-2.1.2 Eigenschaften kreativer Prozesse (K3) – 10 min

Begriffe

Kreative Prozesse, Fehler, Angst vor Fehlern, Notwendigkeit, Fehler zu machen, um zu lernen

Die Antwort auf viele Design-Fragen wird nicht vom Entwicklungsteam gefunden, sondern der Kunde wird sie beantworten. Um diese Antworten zu bekommen, müssen die Fragen konkretisiert werden, sodass der Kunde sie versteht und beantworten kann. Dies kann nicht in Beschreibungen erfolgen, sondern nur in einem Prototyp.

Kreative Prozesse benötigen Iterationen. Auf einen Versuch folgt eine rasche Verifikation.

Damit sie effizient ist, geschieht die Verifikation vor einem auf der Idee aufbauenden Folgeprozess.

Fehlerhafte Designs werden passieren und sind ein normaler Bestandteil des Prozesses. Die meisten Ideen werden zu dieser Kategorie gehören.

Häufige Fehler, die rasch korrigiert werden, sind ein wesentlicher Erfolgsfaktor für kreative Prozesse.

Referenz

[25], [29]

2.2

Usability und Unternehmensziele

30 Minuten

LO-2.2.1 Fundierte Annahmen als Basis zur Steigerung der Usability (K4) – 10 min

Begriffe

Analyse der Ist-Situation, Beschreibung des Problems und der Zielsetzung

Die Usability eines Produktes ist in vielen Fällen unmittelbar mit den Geschäftszielen verbunden. Soll die Usability verbessert werden, ist die tatsächliche Fragestellung bezüglich der Geschäftsziele präziser zu formulieren.

Gibt es bereits ein Produkt (z. B. Website) sind folgende Fragen eine gute Ausgangsbasis:

- Wie wird das Produkt benutzt und wie trägt die Benutzung zum Geschäftserfolg bei?
- Gab es Versuche zur Verbesserung? Wie war das Ergebnis?
- Wer ist die Zielgruppe (Personas)?

Analyse von Mitbewerbern. Sehr oft haben Mitbewerber zumindest einzelne Fragestellung ausgezeichnet gelöst. Es lohnt sich, diese genauer anzusehen.

In einer Gruppendiskussion (Zusammensetzung beachten) wird beschrieben:

- In welchen Bereichen des Produktes werden die Geschäftsziele nicht erreicht.
- Eine klare und präzise Formulierung der gewünschten Verbesserung.

Die Annahmen werden in einer Gruppe getroffen und schriftlich zusammengefasst.

Referenz

[25]

LO-2.2.2 Annahmen als Ausgangsbasis (K4) – 10 min

Begriffe

Analyse der Ist-Situation, Beschreibung des Problems und der Zielsetzung

Ergebnisbeispiel:

Unsere mobile Applikation erlaubt den Check-in ab 36 h vor Abflug bis 2 h vor Abflug. Ziel der Applikation ist es, den Check-in und die Sitzplatzwahl vom Kunden selbst durchführen zu lassen, um weniger Personaleinsatz in der Abfertigung vor dem Flug zu haben. Kundenfeedbacks weisen darauf hin, dass bei Nichtbestehen einer aktiven Internetverbindung (Roaming), Kunden nicht in der Lage sind, die mobile Boardkarte vorzuweisen. Dementsprechend wird das Personal am Flughafen unnötig mit Tätigkeiten wie Suchen der Buchung, Ausdrucken der Boardkarte u. Ä. belastet. Das Problem soll derart ge-

löst werden, dass nahezu alle Kunden mit der geöffneten Boardkarte am mobilen Gerät zum Check-in-Schalter kommen und die Boardkarte mit dem Barcodescanner eingescannt werden kann.

Die Teilnehmerinnen des Seminars erstellen ein ähnliches Beispiel.

Im Beispiel enthalten sein sollen:

- Ziel der Anwendung
- Wie soll die Anwendung die Geschäftsziele unterstützen?
- Warum werden die Geschäftsziele nicht erreicht (Vermutungen)?
- Was soll getan werden, um die Geschäftsziele zu erreichen?

Im Beispiel nicht enthalten sein soll ein konkreter Vorschlag, wie die Lösung aussieht.

LO-2.2.3 Projekt-Faktoren (K4) – 10 min

Begriffe

Projekt-Faktoren aus Sicht von UX

Aufbauend auf LO-2.2.2 erstellen die Teilnehmerinnen und Teilnehmer ein Arbeitsblatt, in dem die wichtigsten Faktoren der zu lösenden Fragestellung zusammengefasst sind.

Kunden

Meine Kunden

- sind definiert als (Personas)
- wollen mit dem Produkt folgendes Hauptziel erreichen
- erreichen das Ziel mit dem Produkt durch ...
- wollen mit dem Produkt folgende Nebenziele erreichen (maximal drei)
- erreichen die Nebenziele mit dem Produkt durch ...
- benutzen das Produkt in folgenden Umgebungen (mobile, Laptop, Internet verfügbar, Internet nicht verfügbar, Bildschirmgröße etc.)
- benutzen das Produkt bei folgenden Gelegenheiten

Geschäftsziele

Wie wirkt sich die Verwendung des Produktes auf die Geschäftsziele aus.

Risiken

Meine größten Risiken sind ...

Mitbewerber

Wir sind besser als die Mitbewerber, weil ...

Weitere Angaben

Zusätzliche Informationen sollen beinhalten:

- Wie werden die Kunden erreicht?
- Welche Mitbewerber gibt es und wie lösen diese die aktuelle Fragestellung?

2.3

Nutzerbedürfnisse erfassen – 2 LO

30 Minuten

LO-2.3.1 Requirements-Engineering und Usability (K2) – 20 Minuten

Begriffe: Requirements-Engineering, Definition von Anforderungen

Requirements-Engineering und Usability stehen in engem Zusammenhang.

Die Hauptaufgaben von Requirements-Engineering ist es, die Bedürfnisse von im Software-Entwicklungsprozess beteiligten Interessengruppen („Stakeholder“) standardisiert und hinreichend

- zu erfassen
- zu verwalten
- zu kommunizieren

Das Ziel dabei ist, das Projektteam derart mit Informationen zu versorgen, sodass es in der Lage ist, die Projektziele zu erreichen.

Die Interessengruppen sind i. A. Benutzer und Auftraggeber, fallweise auch weitere.

Die Zielvorstellung ist wesentlicher definierter Bestandteil und muss gemeinsam von allen Stakeholdern getragen werden.

Es ist ein wenig überraschend, dass konsequentes Requirements-Engineering alleine nicht wesentlich zu einer positiven Usability bzw. UX beiträgt.

Verschiedene Untersuchungen zeigen, dass der überwiegende Teil der tatsächlichen Interaktionen mit dem System (>50 %) von Entwicklern alleine festgelegt wird.

Der Grund dafür scheint zu sein, dass in den meisten Entwicklungsprozessen

- das Wissen des Teams nicht in den UI-Design-Prozess einfließt
- der Feedbackprozess (Usability-Test) nicht Teil des Prozesses ist
- kein oder nur wenig iteratives Vorgehen bei der Entwicklung von GUIs betrieben wird

Hinweise für derartige Probleme sind:

- Das Entwicklungsteam sieht die Implementierung von Funktionalitäten vorrangig („Abhaken von Punkten“)
- Es finden sich keine Arbeitspakete, die Usability-Tests vorsehen
- Es finden sich keine Arbeitspakete, die eine Korrektur von Usability-Problemen vorsehen.

Referenzen

[30]

Richter & Flückinger [31]

LO-2.3.2 Usability-Engineering im Entwicklungsprozess (K2) – 10 Minuten

Begriffe: Software-Entwicklungsprozess, Schnittstelle zu UX

Entwicklungsprozesse sind geradezu dramatisch unterschiedlich. Es ist nicht sinnvoll möglich, ein einziges Vorgehensmodell für Usability-Engineering, welches auf alle anzutreffenden Situationen passt, zu beschreiben.

Sinnvoll ist es, die Prinzipien, die zu guter Usability führen, in den jeweiligen Softwareentwicklungsprozess zu integrieren. Ziel ist es, das Beste aus beiden Welten zu erhalten.

Diese Vorgehensweise bedingt, dass sowohl der Softwareentwicklungsprozess als auch die Methoden des Usability-Engineerings verstanden und kombiniert werden können.

2.4

Minimum-Viable-Products (MVPs) – 2 LO

30 Minuten

LO-2.4.1 Minimum-Viable-Products-Grundlagen (K2) – 15 Minuten

Begriffe: Minimum-Viable-Product, Lean UX

Minimum-Viable-Products (MVPs) sollen helfen, Annahmen mit sehr geringem Arbeitsaufwand zu testen.

- Welche Features sollen weiterentwickelt werden?
- Wird die eingeschlagene Taktik zum Geschäftserfolg beitragen?

Ziele:

- Begrenzte Ressourcen sinnvoll einsetzen

- Sinnlose Arbeit minimieren

Zwei Haupteinsatzgebiete:

1. Herausfinden, was der Markt benötigt

Z. B. Ein Unternehmen möchte wissen, ob es Sinn macht, einen Newsletter zu versenden. Ein MPV kann in diesem Fall eine Möglichkeit zur Anmeldung zu einem nicht existierenden Newsletter sein. Falls sich ein genügender (vorher Annahmen treffen!) Prozentsatz der Besucher anmeldet, dann macht es Sinn, den weiteren Aufwand zum Design des Newsletters zu betreiben.

2. Herausfinden, wie der Markt etwas benötigt

Z. B.: MPV erstellen, der verschiedene Features wie Content-Strategie, Design etc. des Newsletters testet.

Z. B.: MPV erstellen, der eine bestimmte Methode der Darstellung von verfügbaren Fahrrädern in Standorten in der ganzen Stadt auf Verständnis testet.

Referenz

[25], [30], [31], [32]

LO-2.4.2	Minimum-Viable-Products und Usability-Experimente (K4) – 15 Minuten
----------	---

Begriffe: Minimum-Viable-Product, Usability-Test

Wenn ein MPV erstellt wird, sind folgende Prinzipien im Auge zu behalten:

- Priorisierung: Aus den vielen Ideen rasch die wichtigsten auswählen. Ideen kommen und gehen und sind ein flüchtiges Element
- Agilität: Rasches Reagieren auf neue Erkenntnisse muss möglich sein - > Auswahl des Mediums des MVPs
- Handlungen messen. Handlungen von Usern sind wichtiger als Aussagen
- Call to action verwenden: Wenn Sie wissen wollen, ob Anwender etwas tun wollen, dann müssen diese Anwender auch aufgefordert werden, es zu tun. Z. B.: „Jetzt anmelden“

Für viele Fragen ist ein Prototyp die Methode der Wahl.

Methoden, um einen MPV als Prototyp zu realisieren:

- Paper-Prototypen
- Klickbare Wireframes (Tools)
 - o Balsamiq
 - o Visio

- OmniGraffel (Mac)
- Power Point
- Fluid Designer
- High-End-Prototypen (Tools)
 - Axure R P
 - Adobe Fireworks
- Lauffähige Prototypen

Für viele Fragen ist kein Prototyp notwendig, es existieren bessere Möglichkeiten.

Nicht prototypische MVPs:

- Ein Button, der nichts bewirkt, sondern nur Klicks zählt
- Landing Pages. Jeder Anwender, der die angebotenen Tasks durchführt zählt
- Google Ad Words. Herausfinden, wie groß das Interesse an verwendeten Wörtern ist

Referenz

[25], [30], [31], [32]

2.5

Usability-Engineering in Softwareentwicklungsprozesse integrieren – 2 LO

40 Minuten

LO-2.5.1

Usability und klassische Softwareentwicklungsverfahren (K4) – 20 Minuten

Begriffe: klassische Softwareentwicklungsmethoden, Wasserfallmodell, V-Modell, Spiralmodell.

Überblick über klassische Methoden der Softwareentwicklung und Erklärung, in welchen Bereichen Usability ein Thema ist.

- Wasserfallmodell
- V-Modell
- Spiralmodell

Projektmanagement für Softwareprojekte und Erklärung, in welchen Phasen wie auf Usability zu achten ist.

LO-2.5.2 Usability und agile Softwareentwicklungsverfahren (K4) - 20 Minuten

Begriffe: agile Methoden, agile Manifesto, Scrum, Scrummaster, Scrumowner

Überblick über agile Verfahren zur Softwareentwicklung. Erklärung, wie in welchen Bereichen auf Usability geachtet werden muss.

3 Vertiefende Übungen (K6)

60 Minuten

3.1 Kollaboratives Design – 1 LO (K6)

60 Minuten

LO-3.1.1 Übung: Kollaboratives Design beherrschen (K6) – 60 Minuten

In UX Projekten arbeiten i. A. mehrere Personen aus unterschiedlichen Gruppen zusammen. Kommunikation zwischen unterschiedlichen Menschen ist eine der größten Herausforderungen in Projekten. Eine standardisierte, in UX-Teams bewährte Methode damit umzugehen, wird als „Design Studio“ bezeichnet. Dabei werden zwei Ziele verfolgt:

- Erste UX-Designs zu bekommen
- Vorstellungen, Zielvorstellungen und Lösungsideen stimulieren und vereinheitlichen

Vorgehensweise

- Definition des Problems, der Rahmenbedingungen und der Limitationen
- Generation von individuellen Ideen
- Präsentation und Kritik der Ideen
- Weiterentwicklung der Ideen und Zusammenführung der Ideen
- Generation von Ideen, zu denen das gesamte Team die Zustimmung geben kann

Teamgröße

Die Teams sollen in etwa 7 Personen umfassen, aber nicht größer als 10 Personen sein.

Referenz

[25] und [36]

7 Quantifizierung von Ergebnissen (K4)

240 Minuten

7.1. Return on Investment (ROI) (K2) – 3 LO (50 Minuten)

- LO-7.1.1 Den Return on Investment von Usability sowie die Vorgehensweise zu dessen Berechnung kennen (K2) – 15 min
- LO-7.1.2 Anhand von Beispielen den wirtschaftlichen Nutzen von Usability beschreiben können (K2) – 25 min
- LO-7.1.3 Die Kosten von Usability kennen (K2) – 10 min

7.2. Messung von Usability (K4) – 3 LO (115 Minuten)

- LO-7.2.1 Unterschiedliche Metriken zur Messung von Usability kennen (K2) – 35 min
- LO-7.2.2 Statistische Methoden zum Vergleich verschiedener Designs kennen (K2) – 35 min
- LO-7.2.3 Ergebnisse eines Usability-Tests auswerten und interpretieren (K4) – 45 min

7.3. Stichprobengröße – wie viele Tests werden benötigt (K2) – 3 LO (40 Minuten)

- LO-7.3.1 Unterschiedliche Arten von Usability-Studien kennen (K2) – 10 min
- LO-7.3.2 Stichprobengröße für formative Studien abschätzen können (K2) – 15 min
- LO-7.3.3 Stichprobengröße für summative Studien abschätzen können (K2) – 15 min

7.4. Standardisierte Befragungsmethoden (K3) – 3 LO (35 Minuten)

- LO-7.4.1 Unterschiedliche Befragungsmethoden kennen (K2) – 15 min
- LO-7.4.2 Den System-Usability-Scale (SUS) auswerten können (K3) – 20 min

7.1**Return on Investment (ROI) (K2)**

50 Minuten

- LO-7.1.1 Den Return on Investment von Usability sowie die Vorgehensweise zu dessen Berechnung kennen (K2) – 15 min

Begriffe

Return on Investment, Metriken, interner ROI, externer ROI

Die Frage nach dem Nutzen oder ROI von Usability ist wohl fast so alt wie die moderne Usability selbst. Usability-Maßnahmen kosten Ressourcen, welche in

der Regel vom Management bzw. der Projektleitung bereitgestellt werden müssen, weshalb es zuerst notwendig ist, diese Entscheidungsinstanz von dem unbestreitbaren Nutzen der Maßnahmen des Usability-Engineerings zu überzeugen. Daher beschrieb Jakob Nielsen 1993 in seinem Buch „Usability Engineering in der Executive Summary“ bereits einige Fallbeispiele, wie Großunternehmen durch Usability-Maßnahmen erfolgreich Kosten einsparen konnten, beispielsweise durch eine erheblich geringere Fehlerrate bei der Ausfüllung eines Formulars durch Kunden und Kundinnen. Nielsen merkte jedoch auch gleich berechtigterweise an, dass der Nutzen guter Usability nicht immer sofort ersichtlich ist und sich unterschiedlich, oft schwer messbar manifestiert.

Je nach Metrik kann sich die Wirkung einer Usability-Maßnahme kurzfristig, mittelfristig oder langfristig entfalten:

- **Kurzfristig:** Die Veröffentlichung des optimierten Produkts wirkt sich sofort bzw. innerhalb weniger Tage auf die jeweilige Metrik aus. Einige Beispiele sind:
 - Verwendung von Funktionen (z. B. User sollen mehr Feedback zur Qualität eines Online-Artikels geben)
 - Produktivität bzw. User-Performance (z. B. Benutzer und Benutzerinnen einer Intranet-Anwendung ersparen sich täglich 10 Minuten, um die Stunden in die Zeitverwaltungssoftware einzutragen)
 - Aufwand für Support (z. B. Anzahl an Anrufen beim Helpdesk halbiert sich nach Optimierung der Webseite)
 - Conversion-Rate (z. B. Prozent der User, die sich innerhalb einer festgelegten Zeitspanne für den Newsletter einer Nachrichtenwebseite anmelden, steigt von 1 % auf 2,5 %)
- **Mittelfristig:** Die Verbesserung des Produkts benötigt eine gewisse Anlaufzeit, typischerweise einige Wochen oder Monate, um ihre Wirkung zu zeigen. Dazu gehören beispielsweise:
 - Seitenbesuche (z. B. Besucherzahlen einer e-Commerce-Webseite erhöhen sich um 50 %, nachdem die Ladezeiten sowie die Navigation verbessert wurden)
 - Verkaufszahlen (z. B. Umsatz steigt um 20 %, nachdem der Bestellprozess vereinfacht wurde)
- **Langfristig:** Diese Metriken ändern sich nur auf lange Sicht, typischerweise innerhalb einiger Jahre, und sind daher schwer messbar bzw. schwer direkt in Bezug zur Usability-Maßnahme zu setzen, allerdings sind diese Metriken auch die nachhaltigsten, weshalb sie eher auf der Meta-Ebene das langfristige Ziel eines gut etablierten Usability-Prozesses darstellen. Dazu gehören:
 - Kundenzufriedenheit (Wie zufrieden sind meine Kunden und Kundinnen im Allgemeinen mit allen meinen Produkten?)
 - Firmen- oder Markenimage (Was ist mein Ruf am Markt?)
 - Qualität des Produkts (Nehmen Kunden und Kundinnen mein Produkt als hochqualitativ wahr?)

- Wiederverkaufszahlen (Wie viele Kunden und Kundinnen bleiben mir treu und nehmen weiterhin meine Dienste in Anspruch?)

Des Weiteren können ROI-Metriken folgendermaßen eingeteilt werden:

- **Interner ROI:** Diese Gruppe beinhaltet ROI-Metriken, die durch eingesparte Aufwände innerhalb der Organisation entstehen, wie z. B.:
 - Erhöhte Produktivität der Nutzerinnen und Nutzer
 - Verringerte Anzahl von Fehlern, die bei der Bedienung gemacht werden
 - Eingesparte Schulungskosten zur Nutzung des Produkts
 - Einsparungen durch Früherkennung von nötigen Änderungen, anstatt in den finalen Phasen des Produktentwicklungszyklus Änderungen einbauen zu müssen
 - Geringere Supportaufwände
- **Externer ROI:** In diese Kategorie fallen jene Metriken, die einen direkten profitablen Nutzen bzw. Einsparungen nach außen hin bringen, beispielsweise:
 - Erhöhter Umsatz durch mehr Kundinnen und Kunden bzw. erhöhter Umsatz pro Kunde und Kundin
 - Geringere Schulungskosten, falls solche vereinbart wurden
 - Einsparungen durch Früherkennung von nötigen Änderungen bei Auftragsprojekten
 - Verringerte Supportkosten für Kundinnen und Kunden

Natürlich korrelieren manche Metriken miteinander und sind nicht getrennt voneinander zu betrachten. Außerdem gibt es noch weitere, kaum messbare Faktoren, wie zum Beispiel zusätzlicher Gewinn für Softwaredienstleistungsunternehmen, weil diese den Zuschlag für ein Auftragsprojekt erhalten, da sie aufgrund eines etablierten Usability-Prozesses günstiger anbieten können als die Konkurrenz. Auch eingesparte Zeit, Kosten und Aufwände durch früh erkannte Änderungen und Kundenwünsche sind zwar definitiv erstrebenswert, aber kaum messbar und lediglich nur abschätzbar – es ist einfach unmöglich nachzustellen, wie ein Entwicklungsprojekt verlaufen wäre, wenn keine Usability-Maßnahmen gesetzt worden wären.

Die Vorgehensweise bei der Messung des ROI eines Usability-Projekts ist folgende:

1. Definition der relevanten, eindeutig messbaren **Metriken** (z. B. Conversion-Rate, Anzahl an Bestellungen pro Monat ...)
2. **Erhebung** der aktuellen Zahlen
3. **Anwendung** der Usability-Methode(n)
4. **Anpassung** des Designs anhand der Ergebnisse der Untersuchung
5. **Veröffentlichung** des neuen, optimierten Designs
6. **Erhebung** der neuen Zahlen, sobald ein von der Metrik abhängiger sinnvoller Zeitraum verstrichen ist

7. Gegenüberstellung, Evaluierung und Auswertung der erhobenen Zahlen

Da es für das Management oft eindrucksvoller ist, hauseigene ROI-Zahlen zu sehen, anstatt Zahlen anderer Unternehmen präsentiert zu bekommen, kann es durchaus sinnvoll sein, die Messung des Nutzens von Usability-Maßnahmen selbst in die Hand zu nehmen. Dadurch fällt auch für das nächste Projekt die Argumentationsbasis weg, dass die Situation des eigenen Unternehmens „einzigartig“ sei und daher aus bestimmten Gründen auf jegliche Usability-Maßnahme ruhigen Gewissens verzichtet werden könne. Sieht das Management allerdings einen konkreten Nutzen – z. B. die Verkaufszahlen stiegen durch die letzten Optimierungen der Webseite um 20 % –, ist auch mit einem entsprechenden Commitment des Managements zu rechnen.

Referenz

[41], [42], [43]

LO-7.1.2 Anhand von Beispielen den wirtschaftlichen Nutzen von Usability beschreiben können (K2) – 25 min

Um jedoch ein erstmaliges Commitment für den Einsatz von Usability-Maßnahmen einholen zu können, ist es oftmals notwendig, den durchschnittlichen zu erwartenden Nutzen aufzuzeigen und zu argumentieren. Genau aus diesem Grund begann Nielsen bzw. die NN Group, Niensens Usability-Consulting-Unternehmen, bereits sehr früh mit dem Sammeln von zahlreichen Fallbeispielen, um einen konkret messbaren durchschnittlichen Nutzen von Usability, kategorisiert nach Metrik, zu erheben. Obwohl ROI-Daten von Unternehmen relativ schwer zu bekommen sind, da sie teilweise sensible Daten sind und auch niemand gerne zugibt, zuvor auf Usability verzichtet zu haben, umfasst die Studie mittlerweile 72 Fallbeispiele aus den unterschiedlichsten Branchen. Im Folgenden seien einige interessante Beispiele zu den verschiedenen Metriken genannt.

Das amerikanische Telekommunikationsunternehmen **ADC** beschloss im Jahr 2001, den Online-Shop als Vertriebskanal zu forcieren. Bisher diente adc.com eher dazu, (potentielle) Kunden und Kundinnen mit Informationen zu versorgen. Obwohl es zwar die Möglichkeit gab, Produkte online zu bestellen, kamen die Online-Verkäufe nie so richtig in Schwung. Daher beauftragte ADC ein Redesign der Webseite, wodurch u. a. einerseits Online-Bestellungen ohne Anmeldung eines Benutzerkontos ermöglicht wurden und andererseits auch der gesamte Bestellprozess auf vier einfache Schritte reduziert wurde. Die Wirkung zeigte sich bereits im nächsten Jahr: Die **Anzahl an Bestellungen in diesem Jahr** erhöhte sich von 157 auf 704, während der **Umsatz über den Online-Shop** von \$170.000 auf \$750.000 anstieg, was einem imposanten Zuwachs von **348 %** bzw. **341 %** entspricht. Natürlich sind diese Zahlen auch kritisch zu hinterfragen – gerade in diesem Zeitraum gab es viele

(infra-)strukturelle Änderungen, welche eine allgemeine Verbreitung von e-Commerce bedeuten konnten. Des Weiteren könnte der enorme Zuwachs auch durch eine Verschiebung des Absatzkanals entstanden sein: Während die Verkäufe im Online-Shop anstiegen, gingen eventuell dafür weniger Kunden und Kundinnen in den Shop vor Ort. Trotzdem würde dies nichts an der Tatsache ändern, dass durch die optimierte Usability mehr Kunden und Kundinnen über den Online-Shop erreicht werden konnten.

ADC	
Metriken	<ul style="list-style-type: none"> • Umsatz über Online-Bestellungen pro Jahr • Anzahl an Online-Bestellungen pro Jahr
Vorher (2001)	<ul style="list-style-type: none"> • \$170.000 Umsatz • 157 Bestellungen
Nachher (2002)	<ul style="list-style-type: none"> • \$750.000 Umsatz • 704 Bestellungen
Zuwachs	<ul style="list-style-type: none"> • 341 % mehr Umsatz • 348 % mehr Bestellungen

Die Webseite der Online-Plattform **OpenTable** ermöglicht den Benutzerinnen und Benutzern das Finden und direkte Reservieren von Tischen in unterschiedlichen Restaurants. Sowohl die Online-Reservierungen als auch die telefonischen Tischreservierungen werden direkt im System eingetragen, weshalb es wünschenswert ist, dass möglichst viele Gäste die Reservierungen selbstständig online vornehmen. Daher beschloss OpenTable, einige Usability-Verbesserungen im Rahmen eines Redesigns vorzunehmen, wodurch u. a. die Suchfunktion verbessert, das Design modernisiert und aktuelle Sonderangebote prominenter dargestellt wurden. Innerhalb von drei Monaten konnte die Gesamtanzahl an monatlichen **Tischreservierungen**, die per Telefon und online getätigt wurden, von 12.142 auf 26.621 und die monatliche Anzahl an **Account-Registrierungen** von 8.461 auf 16.917 gesteigert werden. Dies bedeutet eine Steigerung von **119 %** bzw. **100 %**. Die Zahlen dieses Beispiels sind robuster als jene des vorherigen, da die Zeitspanne wesentlich kürzer war und auch die Gesamtanzahl an Reservierungen – online und telefonisch – berücksichtigt wurde, weshalb eine Verschiebung des Reservierungsweges nicht infrage kommt.

OpenTable	
Metriken	<ul style="list-style-type: none"> • Anzahl an Tischreservierungen pro Monat (telefonisch plus online) • Anzahl an Account-Registrierungen pro Monat
Vorher (Nov. 2001)	<ul style="list-style-type: none"> • 12.142 Reservierungen

Nachher (Feb. 2002)	<ul style="list-style-type: none"> • 8.461 Account-Registrierungen • 26.621 Reservierungen • 16.917 Account-Registrierungen
Zuwachs	<ul style="list-style-type: none"> • 119 % mehr Reservierungen • 100 % mehr Account-Registrierungen

eBags gehört zu den größten Verkaufsunternehmen von Taschen und Accessoires in den Vereinigten Staaten. Durch die Reduktion der Ladezeiten der Webseite und die Vereinfachung des Bestellprozesses konnte die **Conversion-Rate**, gemessen als Anteil an Besucherinnen und Besuchern, die etwas auf der Seite kaufen, von 1 % auf 2,5 % erhöht werden, was einem Zuwachs von 150 % entspricht.

ebags.com	
Metriken	<ul style="list-style-type: none"> • Conversion-Rate (Kauf pro Seitenbesuch)
Vorher (1998)	<ul style="list-style-type: none"> • 1 % der Besucher/Besucherinnen kauften etwas
Nachher (2001)	<ul style="list-style-type: none"> • 2,5 % kauften etwas
Zuwachs	<ul style="list-style-type: none"> • 150 % mehr Käufe pro Besuch

Ein äußerst interessantes Fallbeispiel verdeutlicht die Wichtigkeit von kleinen Details im Bereich der Usability – selbst sehr kleine Details können äußerst große Unterschiede in der Bedienbarkeit ausmachen. Auf der Webseite von **Microsoft Office** finden sich Hilfsartikel zur Bedienung von Microsoft Office, welche von Besucherinnen und Besuchern bewertet werden können. Da Microsoft ein erhöhtes Interesse an der **Verwendung der Bewertungsfunktion** hat, wurden drei verschiedene Bewertungsvarianten implementiert und im Sinne des A/B-Testings mehr als eine Million Mal unterschiedlichen Benutzerinnen und Benutzern präsentiert. Die erste Variante beinhaltete eine 5-Sterne-Bewertung und ein optionales Textfeld, welches nach einer zusätzlichen Begründung fragt. Die zweite Variante präsentierte ebenfalls eine 5-Sterne-Bewertung, allerdings erschien das optionale Textfeld erst, wenn man sich für eine Sternebewertung entschied. Die dritte Variante beinhaltete die Frage „War diese Information für Sie hilfreich?“ mit drei Schaltflächen für Ja / Nein / Weiß nicht, wobei je nachdem, welcher Button gedrückt wurde, ein Textfeld mit einer angepassten Nachricht erschien („Wie konnte Ihnen die Information weiterhelfen?“, „Wie können wir die Informationen verbessern?“, „Was versuchen Sie zu tun?“). Die genauen Rücklaufanzahlen sind leider geheim, allerdings sind die Verhältnisse zwischen den drei Varianten bekannt. Den letzten Platz belegte die 5-Sterne-Bewertung mit dem fixen optionalen Textfeld, codiert als 100 %. Die 5-Sterne-Bewertung mit dem erscheinenden Textfeld konnte 220 % und somit mehr als das Doppelte an Antworten errei-

chen. Mit Abstand konnte jedoch die „Ja / Nein / Weiß nicht“-Variante gewinnen – im Vergleich zur ersten Variante haben **795 %** und somit fast achtmal so viele Leute geantwortet. Dies ist ein perfektes Beispiel für Niensens Regel „Details matter“ – einen solchen enormen Unterschied würde man kaum erwarten und ohne Usability-Maßnahmen nur schwer entdecken. Im Nachhinein ist es natürlich leicht zu erklären: Leserinnen und Leser eines Artikels denken nach dem Durchlesen eines Artikels nicht in einem 5-Sterne-System – der Artikel konnte ihnen entweder helfen oder eben nicht, weshalb eine „Ja / Nein / Weiß nicht“-Antwort genau dem mentalen Modell entspricht und ein einfacher Klick auf „Ja“ oder „Nein“ einen sehr geringen kognitiven Aufwand bedeutet.

Microsoft Office Online	
Metriken	<ul style="list-style-type: none"> • Anzahl an abgeschickten Reviews für Artikel
Erste Möglichkeit	<ul style="list-style-type: none"> • 100 %
Zweite Möglichkeit	<ul style="list-style-type: none"> • 220 %
Dritte Möglichkeit	<ul style="list-style-type: none"> • 795 %

Wie wichtig es ist, Online-Texte webtauglich zu verfassen, zeigt das Beispiel eines Whitepapers für Java von **Sun Microsystems**. Indem das informative Whitepaper zusammengefasst wurde, durch Aufzählungszeichen das Scannen erleichtert und das Whitepaper objektiver (statt verkaufsfördernd) gestaltet wurde, konnte einerseits die durchschnittliche **Lesedauer** von 637 Sekunden auf 315 Sekunden mehr als **halbiert** werden. Andererseits wurde aber auch gleichzeitig die **Einprägsamkeit**, welche mithilfe von Multiple-Choice-Tests überprüft wurde, **verdoppelt**.

Sun Microsystems	
Metriken	<ul style="list-style-type: none"> • Task-Time in Sekunden • Einprägsamkeit
Vorher	<ul style="list-style-type: none"> • 637 Sekunden • 33 % Einprägsamkeit
Nachher	<ul style="list-style-type: none"> • 315 Sekunden • 65 % Einprägsamkeit
Zuwachs	<ul style="list-style-type: none"> • 102 % verbesserte Task-Time • 97 % bessere Einprägsamkeit

Dass durch Usability-Maßnahmen auch der Aufwand und die Entwicklungszeit von Projekten reduziert werden können, zeigt das Beispiel eines anonymen **Netzwerkanbieters**. Durch frühe Usability-Maßnahmen konnte erhoben wer-

den, dass ein sehr aufwändiges Feature, nämlich eine userbasierte Personalisierung von den zukünftigen Benutzerinnen und Benutzern gar nicht benötigt wird. Dadurch konnte die **Entwicklungszeit** von 13 Wochen auf 10 Wochen reduziert werden, was einer Verbesserung von **30 %** entspricht.

Netzwerkanbieter	
Metriken	<ul style="list-style-type: none"> • Entwicklungszeit
Vorher	<ul style="list-style-type: none"> • 13 Wochen
Nachher	<ul style="list-style-type: none"> • 10 Wochen
Zuwachs	<ul style="list-style-type: none"> • 30 % Entwicklungszeit eingespart

Die ausgewählten Fallbeispiele zeigten, wie der ROI unterschiedlicher Metriken berechnet werden kann. Die nachstehende Tabelle zeigt die zusammengefassten Gesamtergebnisse von allen 72 Fallbeispielen, wobei manche Metriken in gemeinsame Kategorien zusammengefasst wurden:

Metrik	Durchschn. Zuwachs
Umsatz / Conversion-Rate	87 %
Traffic / Besucherzahlen	91 %
Produktivität	112 %
Verwendung von Funktionen	174 %

Es kann also behauptet werden, dass im Durchschnitt – unabhängig von Metrik – die Verbesserung ungefähr 100 % beträgt, wenn Usability-Maßnahmen eingesetzt werden. Hervorzuheben ist außerdem, dass es vor allem bei der Produktivität und der Funktionsverwendung sehr viel Potenzial zur Verbesserung gibt.

Referenz

[39]

LO-7.1.3 Die Kosten von Usability kennen (K2) – 10 min

Begriffe

Usability-Projektbudget

Bisher wurde über die möglichen Verbesserungen durch den Einsatz von Usability-Maßnahmen gesprochen. Wie viel soll allerdings in Usability investiert werden, um den optimalen Nutzen zu erzielen? Kann auch zu viel in Usability investiert werden?

Die von der NN Group untersuchten Projekte investierten ungefähr 10 % bis 13 % des Projektbudgets in Usability. Jakob Nielsen empfiehlt daher als aktuelle „Best Practice“, ungefähr 10 % des Budgets für Usability-Maßnahmen zu reservieren, um den optimalen Nutzen zu erzielen. Bei weiteren durchgeführten praktischen Untersuchungen und Firmenumfragen (2000, 2002, 2006, 2007), welche von der NN Group analysiert wurden, haben sich die 10 % immer wieder als gute Näherung bestätigt. Anzumerken ist jedoch, dass das aufgewandte Usability-Budget nicht linear mit der Projektgröße wächst – bei sehr großen Projekten sinkt der prozentuelle Anteil des Usability-Budgets. Wächst das Projekt um den Faktor 10, so wird nur ungefähr viermal so viel in Usability investiert.

Generell ist zur zeitlichen Entwicklung des Usability-Budgets zu sagen, dass sich Usability immer mehr als wesentlicher Bestandteil eines Projekts etabliert. War der ungefähre Anteil, der in Usability investiert wurde, 1971 noch ca. 3 %, stieg er 1989 auf 5 %, 1994 auf ca. 6 % und hat sich Anfang des 20. Jahrhunderts bei seinen 10 % ungefähr stabilisiert. Der gesamte Aufwand für Usability ist jedoch trotzdem noch dabei zu steigen, da sich Usability aktuell vor allem in Großprojekten als fixer Bestandteil integriert, welche zuvor noch nicht so präsent waren

Betrachtet man die räumlichen Unterschiede von Usability, so zeigt sich kaum ein Unterschied zwischen den USA und Europa. Vergleicht man jedoch die Usability-Aufwände mit Australien, so zeigt sich, dass hier bei Großprojekten wesentlich weniger in Usability investiert wird als in den USA und in Europa. Dies lässt sich eventuell dadurch erklären, dass hier die Großunternehmen noch eher „traditionell“ unterwegs sind und es noch etwas an Reife fehlt.

Referenz

[42]

7.2 Messung von Usability (K4)

115 Minuten

LO-7.2.1 Unterschiedliche Metriken zur Messung von Usability kennen (K2) – 35 min

Begriffe

Leistungsmetriken, problembasierte Metriken, Selbsteinschätzungsmetriken, Verhaltensmetriken

Dass Usability-Methoden einen messbaren Nutzen haben und welche ROI-Metriken gemessen werden können, ist jetzt klar. Nun stellt sich jedoch die Frage, wie Usability bzw. User-Experience selbst tatsächlich numerisch festgehalten, gemessen bzw. geschätzt werden kann. Bisher wurde eher eine

qualitative Herangehensweise gewählt – wir führen Usability-Maßnahmen durch, wie zum Beispiel Usability-Tests, und vertrauen darauf, dass die eingebauten Änderungen zum gewünschten positiven Effekt, wie zum Beispiel zufriedener Benutzerinnen und Benutzer, führen.

Die Einführung von quantitativen Elementen bringt viele Vorteile:

- Design- und Evaluierungsprozess werden **strukturiert**, da wichtige (Design-)Entscheidungen nicht aufgrund von Bauchgefühlen, sondern aussagekräftigen Daten getroffen werden
- Entscheidende Instanzen werden mit **Informationen** versorgt
- Der **Schweregrad** von Usability-Herausforderungen kann abgeschätzt werden
- Der **ROI** kann teilweise 1:1 abgeleitet werden
- **Verbesserungen** werden von Iteration zu Iteration aufgezeigt, was auch für das Entwicklungsteam sehr bestärkend wirkt
- Teilweise **schwierig zu entdeckende Probleme**, wie zum Beispiel zeitintensive Formulareingaben, die nun mal schon immer so waren, können aufgedeckt werden
- Hat man sich daran gewöhnt, können Metriken sehr **einfach gesammelt und ausgewertet** werden – manchmal sogar automatisch
- Usability-Metriken sind – entgegen mancher Vorurteile – **aussagekräftig und nützlich**, wenn man die richtigen Methoden kennt und weiß, was die Ergebnisse aussagen
- Metriken helfen auch beim Auffinden von **Ursachen**, sofern sie richtig miteinander kombiniert werden, und decken somit nicht nur das „Was“, sondern auch das „Warum“ ab
- Der wahrscheinlich wichtigste Vorteil ist, dass gesammelte Metriken vom **Management** verstanden und geschätzt werden und daher als Basis für wichtige (Ressourcen-)Entscheidungen dienen

Grundsätzlich unterscheiden wir zwischen vier verschiedenen Kategorien von UX-Metriken, die natürlich nach Belieben miteinander kombiniert werden können. Generell verraten uns UX-Metriken, wie die Benutzerinnen und Benutzer mit dem System interagieren, wie sie es wahrnehmen und wie sie dazu stehen. UX-Metriken können unterteilt werden in: Leistungsmetriken, problembasierte Metriken, Selbsteinschätzungsmetriken und Verhaltensmetriken.

Leistungsmetriken

Das beobachtbare Verhalten der Benutzerinnen und Benutzer in Kombination mit Aufgabenstellungen setzt den Grundstein für Leistungsmetriken („Performance Metrics“). Anhand dieser können sowohl die **Effektivität** als auch die **Effizienz** des Produkts, also zwei von drei Faktoren der ISO-Definition von Usability, evaluiert werden. Leistungsmetriken helfen auch stark dabei, den Schweregrad von Problemen einzuschätzen bzw. besonders schwierige Auf-

gaben zu identifizieren: Wenn die Mehrheit aller Probanden und Probandinnen für eine bestimmte Aufgabe, welche in 2 Minuten erledigt werden kann, 10 Minuten benötigt, stimmt etwas nicht mit der Bedienbarkeit. Des Weiteren schenkt das Management Leistungsmetriken üblicherweise Beachtung, da sie guten Aufschluss über die allgemeine Usability eines Systems geben und auch direkt in Geld im Sinne von Einsparungen und Umsatzsteigerung umgewandelt werden können. Es gilt jedoch zu beachten, dass uns Leistungsmetriken zwar das „Was“ liefern, also beispielsweise Tasks, die Probleme machen, aber nicht das „Wieso“. Daher sollten Leistungsmetriken immer mit anderen Metriken kombiniert werden, wie zum Beispiel qualitativer Input durch Lautdenken oder Freitextfragen.

Die gängigsten Leistungsmetriken sind:

Task-Success

- Durchschnittlicher Anteil an Testpersonen, die eine Aufgabe erfolgreich abgeschlossen haben (z. B. 70 % der Leute haben es geschafft, Aufgabe 1 zu absolvieren)
- Misst die Effektivität eines Produkts
- Kann für jede Usability-Studie eingesetzt werden, die Aufgaben verwendet
- Einfach zu verstehen
- Es sind eindeutige, klare Erfolgskriterien notwendig, daher muss es auch ein wohldefiniertes Ziel geben. (Gut: „Finden Sie den jährlichen Gewinn von XYZ“, schlecht: „Erforschen Sie Möglichkeiten, um für die Pension vorzusorgen“)
- Entweder binärer Task-Success (erfolgreich / nicht erfolgreich bzw. Task-Success / Task-Failure) oder über Erfolgsstufen (z. B. partieller Erfolg, mit Hilfe) – jedenfalls sind eindeutige Kriterien pro Stufe nötig
- Breite Toolunterstützung (z. B. Morae)
- Typische Visualisierungen: Balkendiagramm gruppiert nach Aufgaben, gestapeltes Balkendiagramm für Erfolgsstufen

Time on Task

- Durchschnittliche Dauer einer Aufgabe, gemittelt über alle Testpersonen (z. B. 125 Sekunden für Aufgabe 1)
- Misst die Effizienz eines Produkts
- Üblicherweise gilt: Je kürzer eine Aufgabe dauert, desto besser bzw. effizienter (Ausnahme z. B. bei Spielen)
- ROI kann relativ einfach abgeleitet werden
- Es muss klar sein, wann die „Stoppuhr“ gestartet, pausiert und gestoppt wird
- Anfällig für Ausreißer (z. B. Testperson beantwortet Telefonanruf), daher manuelle Kontrolle notwendig

- Entweder nur erfolgreiche Testdurchläufe für die Berechnung verwenden („verschönert“ das Ergebnis allerdings extrem) oder auch nicht erfolgreich absolvierte Aufgaben (inklusive Task-Failure, kann das Ergebnis allerdings ebenfalls unnatürlich verschlechtern – eventuell ein Maximum setzen)
- Kontroverse darüber, ob man den Testpersonen sagen soll, dass die Zeit gemessen wird
 - o Sagt man es ihnen, verhalten sie sich eventuell unnatürlich und sind gestresst oder nervös
 - o Sagt man es ihnen nicht, könnten sie sich „verplaudern“ oder auch zu explorativ verhalten (das Produkt erst einmal „durchklicken“ o. Ä.)
 - o Jedoch ist es schließlich auch das qualitative Feedback der Probandinnen und Probanden, welches ausschlaggebend dafür ist, Usability-Probleme aufzudecken
 - o Kompromisslösung: Den Testpersonen sagen, dass sie sich beim Erfüllen der Aufgaben möglichst gut konzentrieren (während die Uhr läuft), und anschließend – zwischen den Aufgaben – bespricht man die vorherige Aufgabe und sammelt Feedback
- Breite Toolunterstützung (z. B. Morae)
- Typische Visualisierungen: Balkendiagramm gruppiert nach Aufgaben, Balkendiagramme mit Grenzwerten (z. B. Task-Success-Anteil innerhalb von zwei Minuten)

Fehler

- Durchschnittliche Anzahl an gemachten Fehlern pro Aufgabe
- Unterscheidung zwischen Fehler und Problem
 - o Ein Problem ist die zugrundeliegende Ursache (z. B. verwirrende Navigation)
 - o Fehler sind die Konsequenzen von Problemen (z. B. Testpersonen klicken auf den falschen Link in der Navigation)
 - o Als Usability-Expertinnen und -Experten ist es unsere Aufgabe, Fehler zu beobachten und auf die Probleme zu schließen
- Fehler sind daher Fehlverhalten, das eventuell zu nicht erfolgreich absolvierten Aufgaben führt (Task-Failure)
- Das tatsächliche Messen von Fehlern macht allerdings nur selten Sinn, da es sehr aufwändig ist, und es ist nur sinnvoll, wenn es um kritische Anwendungen geht (z. B. Krankenhaussoftware, Pilotensoftware, sehr teure Konsequenzen)
- Da es keine allgemeingültige Definition von Fehlern gibt, ist es hier umso wichtiger, klare Kriterien zu verwenden, was einen Fehler ausmacht
- Am besten im Vorhinein pro Aufgabe definieren, welche Fehler gemacht werden können

- Es kann auch Sinn machen, ein paar einzelne Fehler herauszupicken (z. B. „falsches Passwort eingegeben“), sich auf diese zu beschränken und dann pro Design auszuwerten
- Typische Visualisierung: Balkendiagramm gruppiert nach Design für jeden Fehler

Effizienz

- Inkludiert andere Abschätzungen des physischen oder kognitiven Aufwands, z. B. Anzahl an Klicks, Mauspfadlänge, Tastenanschläge, nötige Seitenbesuche, Anzahl an erfolgreichen Tasks pro Zeiteinheit (auch als Standard ISO/IEC 25062:2006), Lostness (vor allem für Webseiten)
- Vor allem sinnvoll, wenn Aufgaben ungefähr dasselbe Ausmaß an Aufwand erfordern, da von Natur aus komplexere Aufgaben sonst unnötig „schlecht“ dargestellt werden

Erlernbarkeit

- Lernen passiert nicht augenblicklich, sondern erfordert Zeit
- Man möchte daher messen, wie lange es dauert, bis die Testpersonen effizient mit dem Produkt arbeiten können
- Die Erlernbarkeit ist vor allem für häufig verwendete Produkte (z. B. Zeiterfassungssoftware) relevant
- Die grundsätzliche Herangehensweise beinhaltet das Messen von anderen Leistungsmetriken (üblicherweise Task-Time oder Effizienz) in mehreren, zeitlich voneinander getrennten Testdurchläufen mit denselben Testpersonen
- Idealerweise entspricht die Zeit zwischen den Testdurchläufen jener, die auch in der Realität zwischen den Verwendungen des Produkts liegt, was allerdings nicht immer möglich ist (z. B. Software für Jahresabschluss) – es sollte allerdings zumindest ein Tag dazwischenliegen, um Vergessen wenigstens minimal zu berücksichtigen
- Bei der Analyse sieht man typischerweise eine sich abflachende Lernkurve
- Sobald die Testpersonen kaum mehr effizienter werden, kann die Analyse beendet und können Vergleiche gemacht werden (z. B. anfangs benötigen Benutzerinnen und Benutzer doppelt so lange, um Aufgabe 1 zu bewältigen)
- Typische Visualisierung: Liniendiagramm über die Testdurchläufe

Problembasierte Metriken

Das Identifizieren von Usability-Problemen und das Herausarbeiten von Verbesserungsvorschlägen sind wahrscheinlich die wichtigsten Aufgaben von Usability-Expertinnen und -Experten, da sie Produkte einfach besser bedienbarer machen wollen – problembasierte Metriken („Issue-based Metrics“) setzen genau darauf auf. Das Hauptwerkzeug zum Identifizieren von Usability-

Problemen ist das **Beobachten des Verhaltens** der Testpersonen, beispielsweise jene Verhaltensweisen, die zum nicht erfolgreichen Abschluss von Aufgaben (Task-Failure) führen, die Effizienz erheblich beeinträchtigen (auf den falschen Weg leiten) oder auf Frustration schließen lassen. Usability-Spezialistinnen und Spezialisten **interpretieren** nun diese beobachteten Verhaltensweisen und leiten die zugrundeliegenden Usability-Probleme ab, um schließlich Verbesserungsvorschläge herausarbeiten zu können.

Beispielsweise ist die Beobachtung „Etliche Testpersonen klicken sehr viel und fast wahllos in der Toplevel-Navigation auf der Suche nach der gewünschten Information herum“ interpretierbar als „Die Toplevel-Navigation verwirrt die Testpersonen“, was auf das Usability-Problem „Die Toplevel-Navigation ist verwirrend / beinhaltet nicht eindeutige Kategorien“ schließen lässt. Beim Schreiben des Berichts sollten jedenfalls Beispiele (am besten mit Screenshots) gegeben werden und auch schon erste Hinweise inkludiert werden, wo die Ursache des Problems liegen könnte, um dem Entwicklungsteam Arbeit abzunehmen.

Es ist jedoch meist kontraproduktiv, dem Entwicklungsteam eine Liste mit Hunderten Usability-Problemen – eventuell sogar alle mit der Priorität „hoch“ – „an den Kopf zu werfen“ und es als vollendete Tatsache zu betrachten. Sich auf das Wesentliche und die größten Usability-Probleme zu beschränken ist oft eine Herausforderung, jedenfalls sollten aber zumindest eine sinnvolle Priorisierung vorgenommen und auch **positive Dinge** der aktuellen Version hervorgehoben werden. Dadurch fühlt sich das Entwicklungsteam einerseits positiv bestärkt, andererseits stellt man aber auch sicher, dass jene Elemente, die gut funktionieren (z. B. die Seitensuche), bei der nächsten Iteration nicht „kaputtgemacht“ werden (z. B. komplette Überarbeitung der Suche).

Doch wie genau werden nun Usability-Probleme identifiziert? Die zwei Hauptmethoden zur **Identifikation von Problemen** beim Usability-Testing sind das **Beobachten des Verhaltens** von Testpersonen sowie **Input durch die Probandinnen und Probanden** mithilfe von Varianten wie Lautdenken. Basierend auf diesen Daten ist es nun die Aufgabe von Usability-Spezialistinnen und -Spezialisten, Usability-Probleme abzuleiten und zu messen. Dies ist natürlich besonders herausfordernd, wenn kein direkter Zugriff auf die Testpersonen möglich ist, wie es meist bei Online-Studien der Fall ist. Dadurch, dass man nicht direkt mit den Probandinnen und Probanden interagieren kann und ihnen keine (Rück-)Fragen stellen kann, ist es schwieriger, bestimmte Verhaltensweisen zu interpretieren. Daher müssen oft die gesamten Video- und Tonaufnahmen angesehen bzw. analysiert werden und die Antworten auf offene Fragen interpretiert werden.

Oft stellt sich die Frage, was mit **Einzelfällen** gemacht werden soll (z. B. 1 von 10 Testpersonen hatte ein Problem mit dem Wording). Ob es sich bei einer solchen Beobachtung um ein Usability-Problem handelt oder nicht, hängt von der zu erwartenden Reproduzierbarkeit des Problems in der Gesamtpopulation der Benutzerinnen und Benutzer ab. War das Verhalten der Testperson durchaus schlüssig, logisch und folgte es einer nachvollziehbaren Story, so könnte es sich tatsächlich um ein Usability-Problem handeln. Kann die Testperson jedoch nicht wirklich erklären, was bzw. warum sie etwas tat, so wird es sich höchstwahrscheinlich um einen Ausreißer handeln.

Das Verwenden von **Schweregraden** (Severity) bei Usability-Problemen ist oft essenziell. Der Schweregrad hängt meist von dem Ausmaß der Beeinträchtigung der User-Experience ab. Ein gängig verwendetes System ist ein dreistufiges nach folgender Einteilung:

- **Niedrig**: Geringe Beeinträchtigung, das Problem „nervt“ die Benutzerinnen und Benutzer in geringem Ausmaß, Effizienz und Zufriedenstellung sind nicht oder nur schwach beeinträchtigt
- **Mittel**: Hat einen signifikanten Einfluss auf die Schwierigkeit der Aufgabe, Effizienz und Zufriedenstellung sind beeinträchtigt, aber das Problem führt nicht zur Nichtbewältigung (Task-Failure) der Aufgabe
- **Hoch**: Das Problem führt zur Nichtbewältigung der Aufgabe (Task-Failure), Effektivität, Effizienz und Zufriedenstellung sind massiv beeinträchtigt

Der Schweregrad kann natürlich auch anhand verschiedener Faktoren berechnet werden, beispielsweise aus einer Mischung von Beeinträchtigung der User-Experience, geschätzter Häufigkeit des Auftretens, Auswirkungen auf Unternehmensziele und Kosten zur Behebung des Problems. Wie auch immer der Schweregrad berechnet wird, es ist wichtig, dass alle Parteien ein klares Verständnis darüber haben, was die einzelnen Schweregradkategorien bedeuten. Beispiele pro Schweregradkategorie können hier sehr hilfreich sein.

Die gängigsten Arten zur **Analyse** von problembasierten Metriken sind:

Häufigkeit von einzelnen Usability-Problemen

- Gesamtanzahl einzigartiger Usability-Probleme über alle Testpersonen hinweg, gruppiert nach Design/Iteration
- Kein doppeltes Zählen von aufgetretenen Problemen (z. B. Testperson A stolperte über 10 Probleme, Testperson B über 12 Probleme, allerdings wurden 6 davon bereits bei Testperson A entdeckt → 16 Probleme)
- Besonders hilfreich, um den Fortschritt der Iterationen aufzuzeigen
- Schweregrade können inkludiert werden

Häufigkeit von Usability-Problemen

- Durchschnitt von gesamten Usability-Problemen über alle Testpersonen hinweg, gruppiert nach Design/Iteration
- Doppeltes Zählen von aufgetretenen Problemen
- Schweregrade können inkludiert werden

Anzahl an Testpersonen, die bestimmtem Usability-Problem begegnen

- Anzahl oder Prozentsatz aller Testpersonen, die einem bestimmten Problem begegnen
- Hilfreich, um die Usability von bestimmten Elementen zu überprüfen

Probleme nach Kategorie

- Kategorisierung von Usability-Problemen (z. B. Design / Navigation / Inhalt)
- Kategorisierung sollte für das Zielpublikum Sinn machen

Probleme nach Aufgabe

- Usability-Probleme pro Aufgabe aufschlüsseln
- Schnelle Identifizierung von Aufgaben, die zu den meisten Problemen führen
- Schweregrade können inkludiert werden

Selbsteinschätzungsmetriken

Einen guten Einblick in die Zufriedenstellung des Produkts geben Selbsteinschätzungsmetriken („Self-reported Metrics“), welche gemessen werden können, indem die Testpersonen zur Usability des Systems einfach befragt werden. Sie werden absichtlich nicht „subjektive Metriken“ genannt, da dies einen falschen Eindruck vermittelt. Zwar entstehen diese Metriken durch subjektive Einschätzungen der Benutzerinnen und Benutzer, allerdings ist User-Experience erstens per Definition schon etwas Subjektives, da jede Benutzerin und jeder Benutzer ein anderes Erlebnis des Systems hat. Zweitens sind für uns Usability-Expertinnen und -Experten diese Daten sehr wohl objektiv – die Summe der Einschätzungen gibt eine saubere, objektive Messung der allgemeinen User-Experience des Systems.

Eine klassische Möglichkeit, Einschätzungen über das System zu erhalten, sind **Likert**-Skalen, bei welchen die Testpersonen üblicherweise den Grad der Zustimmung zu einem vorgegebenen Statement angeben sollen, beispielsweise:

1. Stimme zu
2. Stimme eher zu
3. Weder noch
4. Stimme eher nicht zu
5. Stimme nicht zu

Klassischerweise werden fünf oder sieben Optionen geboten, wobei es immer ein „Ankerwort“ pro Option gibt. Generell sollten Adverbien und Adjektive vermieden werden (z. B. „sehr“, „absolut“, „extrem“, „überhaupt nicht“), da Menschen üblicherweise nicht gerne Extrempositionen einnehmen.

Eine andere häufig verwendete Möglichkeit ist das **semantische Differential**. Der Hauptunterschied besteht darin, dass es nur bei den äußeren Optionen ein „Ankerwort“ gibt und die Optionen dazwischen in gleichmäßigen Abständen voneinander entfernt liegen, wodurch der Eindruck einer gleichmäßig verteilten Bewertungsskala entsteht, beispielsweise:

schwach	<input type="radio"/>	stark				
unprofessionell	<input type="radio"/>	professionell				
freundlich	<input type="radio"/>	unfreundlich				

Auch hier sollten Adjektive vermieden werden. Außerdem ist es nicht trivial, wahre „Gegensätze“ zu finden. Auch die Wortwahl kann erhebliche Folgen haben – eine Skala von „freundlich“ bis „unfreundlich“ liefert andere Ergebnisse als eine von „freundlich“ bis „feindlich“ oder „freundlich“ bis „nicht freundlich“.

Abgesehen von der verwendeten Skalenart spielt auch die **Zeit**, wann die Testpersonen gefragt werden, eine wesentliche Rolle. Üblicherweise nutzt man kurze, schnelle „Post-Task“-Ratings (Bewertungen direkt nach Aufgabenstellungen), um die Schwierigkeit von Aufgaben abzufragen, da sich die Testpersonen hier noch am besten an die Aufgabenstellungen erinnern. Eine ausführlichere Evaluierung der generellen User-Experience des Gesamtsystems macht hingegen eher am Ende des Usability-Tests Sinn, da die Probandinnen und Probanden zu diesem Zeitpunkt das gesamte System kennen und gut einschätzen können.

Auch die **Art**, wie die Testpersonen gefragt werden, spielt eine Rolle. Werden sie mündlich befragt, so ist dies zwar vielleicht am bequemsten für die Testperson, allerdings wird hierdurch eine ungewollte Verzerrung der Ergebnisse durch die soziale Erwünschtheit herbeigeführt, da Menschen bei direktem Kontakt dazu tendieren, jene Antworten zu wählen, von denen sie glauben, dass sie von dem Fragesteller bzw. von der Fragestellerin gehört werden wollen – es ist ihnen unangenehm, schlechte Bewertungen abzugeben. Die handschriftliche Ausfüllung eines Fragebogens (ohne dass der Testleiter bzw. die Testleiterin zusieht) schafft hier zwar Abhilfe, allerdings können dafür Fehler beim Lesen der Handschrift auftreten. Wenn nichts dagegen spricht, sind daher Fragebögen am Computer zu bevorzugen, da sie auch leichter auszuwerten sind und nicht erst digitalisiert werden müssen. Moderne Usability-Test-Tools (wie z. B. Morae) unterstützen dies meist von Haus aus.

Schließlich spielt auch noch das **Design** des Fragebogens bzw. der Fragen eine wichtige Rolle. Nach ein und demselben Attribut (z. B. visuelles Ansprechen) kann auf unterschiedliche Arten gefragt werden, um das Ergebnis zu triangulieren, zu mitteln und somit robuster zu gestalten. Es spielt außerdem eine Rolle, ob eine ungerade oder gerade Anzahl an Antwortmöglichkeiten verwendet wird, da durch letztere Option die Testpersonen dazu gezwungen werden, sich für eine Tendenz in eine der beiden Richtungen zu entscheiden. Dieses Thema ist nicht nur unter Usability-Expertinnen und -Experten sehr kontrovers. Besteht kein außerordentlicher Grund, die Testpersonen zu einer solchen Entscheidung zu zwingen, so ist eine neutrale Antwort üblicherweise kein Nachteil und für die Testpersonen bequemer.

Für die **Analyse** werden den Antwortmöglichkeiten üblicherweise Zahlen zugeordnet und wie gewohnt mit diesen gerechnet, obwohl es sich gerade bei der Likert-Skala eher um eine Ordinalskala handelt und dies daher streng genommen nicht ganz korrekt ist. Weitere Möglichkeiten sind sogenannte Box-Scores (z. B. 60 % der Testpersonen bewerteten die Webseite als (eher) ansprechend). Für die generelle User-Experience des Systems empfehlen sich standardisierte Fragebögen wie der SUS (siehe späteres Kapitel).

Verhaltensmetriken

Probandinnen und Probanden machen während eines Usability-Tests mehr, als nur Aufgaben zu erfüllen und Fragen zu beantworten – sie lachen, grinsen, seufzen, sind frustriert, gestresst, überrascht, aufgeregt und betrachten das Interface. Diese Kategorie von Metriken beschäftigt sich mit möglichst objektiv messbaren Verhaltensmetriken und physiologischen Metriken („Behavioral / physiological Metrics“).

Eine erste, relativ einfache Möglichkeit, Verhalten festzuhalten, ist das Codieren und Klassifizieren von (unaufgeforderten) **Kommentaren** der Testpersonen, beispielsweise:

- **Negativ:** „Uff, das ist schwer“, „Hm, das Design gefällt mir nicht“
- **Positiv:** „Das gefällt mir“, „Das war einfach“
- **Neutral:** „Interessant“, „Verstehe“

Die neutrale Kategorie dient als „Referenzwert“ für die anderen beiden. Natürlich bedeutet das Codieren einen erheblichen Mehraufwand, weshalb es nicht immer sinnvoll ist. Für die Auswertung können die Kommentare als gestapeltes Balkendiagramm über mehrere Iterationen hinweg präsentiert werden.

Mithilfe von **Eye-Tracking**-Systemen kann das Blickverhalten von Testpersonen aufgezeichnet werden – eventuell kennt man Bilder von „Heatmaps“ aus der Werbebranche. Für die Auswertung sind Heatmaps jedoch ungeeignet, da auf Bildern keine statistischen Designanalysen durchgeführt werden können. Was allerdings gemacht werden kann, ist das Definieren von Areas of Interest

(AOI) für interessante Bereiche (z. B. Logo, Navigation), zu welchen anschließend automatisch Metriken berechnet werden können, beispielsweise:

- Durchschnittliche Verweildauer in der AOI
- Anzahl an Fixierungen
- Sequenz/Reihenfolge
- Durchschnittliche Zeit bis zur ersten Fixierung
- Revisits
- Trefferquote

Diese Metriken können wieder wie gewohnt ausgewertet werden. Eye-Tracking-Systeme zeichnen zwar üblicherweise auch den Pupillendurchmesser auf, welcher theoretisch Rückschlüsse auf Emotionen zulässt, allerdings ist dieser von zu vielen Variablen abhängig, als dass er für sinnvolle Vergleiche verwendet werden könnte.

Für spezialisierte Produkte, bei denen die User-Experience die wichtigste Rolle spielt, sind auch noch weitere Verfahren im Einsatz bzw. in Forschung:

- Gesichtsmimik-Erkennung
- Hautleitwertmessung
- EEG
- Ausgeübter Druck auf die Maus bzw. auf den Sessel
- Herzfrequenz

Diese Metriken sind allerdings noch zu teuer bzw. instabil, um von der breiten Masse verwendet werden zu können, und sind daher nur für Spezialfälle sinnvoll.

Referenz

[43]

LO-7.2.2 Statistische Methoden zum Vergleich verschiedener Designs kennen (K2) – 35 min

Begriffe

Skalenniveau, deskriptive Statistik, schließende Statistik, Konfidenzintervall, t-Test

Um fundierte und aussagekräftige Vergleiche zwischen Designs zu treffen, bedient man sich als Usability-Expertin bzw. -Experte grundlegender Methoden der Statistik. Die Verfügbarkeit der jeweiligen Methoden hängt davon ab, welches Format die gesammelten Daten haben. Grundsätzlich wird zwischen folgenden **Skalenniveaus** bzw. Messniveaus unterschieden:

- Qualitative Daten

- **Nominalskala:** Hierbei handelt es sich um ungeordnete Gruppen bzw. Kategorien. Das Einzige, was man weiß, ist, dass sie voneinander verschieden sind – es gibt keine Rangordnung und auch keine Distanz zwischen den einzelnen Kategorien. Beispiele hierfür wären Geschlecht, Nationalität oder Designvarianten. Diese Daten erlauben nur einfache Methoden der Statistik, wie zum Beispiel Häufigkeiten (z. B. 60 % der Testpersonen waren weiblich) und den Modalwert. Die Berechnung weiterführender Kennzahlen wie zum Beispiel arithmetisches Mittel oder Standardabweichung ist nicht möglich.
- **Ordinalskala:** Auch hier handelt es sich lediglich um Gruppen bzw. Kategorien, allerdings existiert hier eine natürliche Rangfolge, z. B. „besser“, „größer“, „schneller“. Es existieren nach wie vor keine Distanzen zwischen den Kategorien. Beispiele hierfür sind die Antworten auf eine Likert-Skala oder Schulnoten. Auch hier stehen nur einfache statistische Methoden zu Verfügung, allerdings kann hier bereits der Medianwert berechnet werden.
- Quantitative Daten (diskret oder stetig)
 - **Intervallskala:** Hier kann nun eine sinnvolle Distanz zwischen den einzelnen Werten berechnet werden. Beispiele hierfür sind Temperatur in Celsius oder die Antworten auf ein semantisches Differential. Bei solchen Daten steht fast das gesamte Repertoire an statistischen Möglichkeiten zu Verfügung (Mittelwert, Konfidenzintervalle, t-Tests ...).
 - **Verhältnisskala:** Verhältnisdaten besitzen zusätzlich auch noch einen natürlichen Nullpunkt, sodass multiplikative Vergleiche wie „doppelt so viel“, „halb so lang“ etc. angestellt werden können. Hier kann noch zusätzlich das geometrische Mittel gebildet werden, was für Usability-Testing allerdings nicht unbedingt relevant ist.

Somit sind wir als Usability-Expertinnen und -Experten vor allem daran interessiert, Intervalldaten zu messen, da diese sehr gut ausgewertet werden können.

Grundsätzlich kann die Statistik in die deskriptive Statistik und die schließende Statistik unterteilt werden.

Deskriptive Statistik

Mithilfe der deskriptiven Statistik werden die gemessenen Daten, wie zum Beispiel Time on Task, beschrieben, es werden allerdings noch keine Vorhersagen bezüglich der Population, also der Grundgesamtheit aller Benutzerinnen und Benutzer des Produkts, gemacht. Die wichtigsten Maße hierfür sind Ortsmaße und Streuungsmaße.

Ortsmaße dienen dazu, eine Reihe an gesammelten Daten anhand einer einzelnen charakteristischen Zahl zu beschreiben. Die drei relevantesten sind:

- **Arithmetisches Mittel**
 - Der durchschnittliche Wert der Daten
 - Wird berechnet als die Summe aller Werte dividiert durch die Anzahl
 - Gut geeignet für symmetrische Datensätze (symmetrische Häufigkeitsverteilung)
 - Nicht robust gegenüber Ausreißern
 - Excel-Funktion: MITTELWERT(Werte)
- **Median**
 - Mittlerer Wert des Datensatzes, sodass 50 % der Daten unterhalb und 50 % der Daten oberhalb des Werts liegen
 - Findet sich bei einer ungeraden Anzahl an Werten genau in der Mitte der sortierten Werte, bei einer geraden Anzahl wird der Mittelwert zwischen den zwei mittleren Werten gebildet
 - Robust gegenüber Ausreißern
 - Excel-Funktion: MEDIAN(Werte)
- **Modalwert**
 - Entspricht dem am häufigsten vorkommenden Wert
 - Nicht immer eindeutig, da Werte gleich oft vorkommen können
 - Für Usability-Testing nur bei Nominalskalen bzw. Ordinalskalen sinnvoll, da bei Intervallskalen und Verhältnisskalen zu viele verschiedene Werte möglich sind
 - Excel-Funktion: MODUS.EINF(Werte)

Komplett unterschiedliche Verteilungen können allerdings denselben Mittelwert aufweisen – eine Normalverteilung kann denselben Mittelwert wie eine Gleichverteilung haben. Daher macht es Sinn, zusätzlich noch ein Maß für die **Streuung** einzuführen. Die gängigsten Maße sind:

- **Spannweite**
 - Berechnet sich aus Minimum minus Maximum
 - Gibt einen groben Überblick über die Verteilung und kann zum Detektieren von Ausreißern verwendet werden
 - Excel-Funktion: MAX(Werte) - MIN(Werte)
- **Stichproben-Varianz**
 - Berechnet sich als Summe der quadratischen Abstände zum Mittelwert, dividiert durch die Stichprobengröße minus 1
 - Ist sehr gut für symmetrische Verteilungen geeignet, allerdings nicht robust gegenüber Ausreißern
 - Weist aufgrund des Quadrierens eine andere Einheit als die Ausgangsdaten auf (z. B. Quadratsekunden), weshalb die Varianz schwer vorstellbar ist
 - Das Quadrieren verhindert einerseits negative Werte und bestraft andererseits Ausreißer mehr als Werte, die nahe beim Mittelwert liegen

- Excel-Funktion: VAR.S(Werte)
- Stichproben-**Standardabweichung**
 - Ist die Quadratwurzel aus der Varianz
 - Besitzt dieselbe Einheit wie die Ausgangsdaten und ist daher besser mit diesen vergleichbar
 - Bei einer Normalverteilung können mit der Standardabweichung interessante Aussagen gemacht werden, z. B. innerhalb von plus/minus drei Standardabweichungen liegen 99,7 % aller Werte
 - Bietet die Basis für Konfidenzintervalle
 - Beim Usability-Testing wird meist davon ausgegangen, dass es sich um eine Normalverteilung handelt (auch wenn dies nicht immer, z. B. bei Time on Task, ganz korrekt ist)
 - Excel-Funktion: STABW.S(Werte)

Schließende Statistik

Meist reicht es jedoch nicht, einfach nur die genannten Kennzahlen zu berechnen. Um fundierte Designentscheidungen treffen zu können, sind wir als Usability-Spezialistinnen und -Spezialisten daran interessiert, Schätzungen abzugeben, ob sich die gemessenen Effekte wahrscheinlich auch in der Grundgesamtheit aller Benutzerinnen und Benutzer so verhalten – nur weil der gemessene Mittelwert einiger Tests mit zwei verschiedenen Designs unterschiedlich ist, heißt das nicht automatisch, dass dies für die Grundgesamtheit auch automatisch zutrifft. Der **Zufall** bei der Auswahl von Benutzerinnen und Benutzern spielt hier also eine wichtige Rolle. Es ist daher oft notwendig zu sagen, ob ein gemessener Effekt (z. B. unterschiedliche Mittelwerte bei verschiedenen Designs) größer ist als das, was wir vom Zufall erwarten würden – mit anderen Worten: ob ein gemessener Effekt signifikant ist.

Die **Signifikanz** kann für unsere Zwecke grundsätzlich über zwei verschiedene Methoden ermittelt werden:

- Mithilfe von Konfidenzintervallen
 - Wenn sich die Konfidenzintervalle zweier Designs nicht überlappen, so besteht ein signifikanter Unterschied
 - Wenn sich die Konfidenzintervalle leicht überlappen, so sollte am besten die zweite Methode verwendet werden, um eine genauere Schätzung zu erhalten
 - Wenn sich die Konfidenzintervalle sehr breit überlappen, so besteht wahrscheinlich kein signifikanter Unterschied
- Mithilfe von statistischen Tests
 - Statistische Tests schätzen ab, mit welcher Wahrscheinlichkeit der gemessene Effekt zufällig entstehen kann (und daher in Wahrheit gar kein Unterschied zwischen den Designs besteht)
 - Diese Zufallswahrscheinlichkeit wird auch p-Wert genannt
 - Ein Unterschied ist signifikant, wenn der p-Wert geringer als ein bestimmter Grenzwert ist, auch Signifikanzniveau genannt

- Gängige Signifikanzniveaus sind für unsere Zwecke 1 %, 5 % und 10 % (wobei 10 % bereits sehr „riskant“ sind)
- Wählen wir also beispielsweise ein Signifikanzniveau von 5 % und ein statistischer Test liefert einen p-Wert von 2 %, so bezeichnet man den Unterschied als signifikant
- Liefert ein statistischer Test allerdings einen Wert größer als 5 %, so ist der Unterschied nicht signifikant

Zu beachten ist jedoch, dass Signifikanz lediglich aussagt, dass der Unterschied „höchstwahrscheinlich nicht 0 ist“. Ungeachtet dessen, dass die meisten Menschen mit „signifikant“ Wörter wie „bedeutsam“, „wichtig“ o. Ä. verbinden, sagt Signifikanz daher nichts über die praktische Relevanz aus. Ein 10-Minuten-Task kann bei zwei Designs signifikant unterschiedlich sein, auch wenn der Unterschied nur 10 Sekunden beträgt. Außerdem ist so ziemlich jeder Unterschied signifikant, wenn genügend Testpersonen getestet werden. Daher ist es immer wichtig, den praktischen Nutzen nicht aus den Augen zu verlieren.

Konfidenzintervalle sind sehr hilfreiche Werkzeuge, um die „Unsicherheit“ bei einer berechneten Kennzahl (z. B. Mittelwert von Time on Task) auszudrücken. Sie sind geschätzte Intervalle, innerhalb deren sich die „wahre“ Kennzahl, also z. B. der Mittelwert der Gesamtpopulation aller Benutzerinnen und Benutzer des Systems, mit einer bestimmten Sicherheit (Konfidenz) befindet. Die gewünschte Sicherheit wird über das Konfidenzlevel festgelegt und je sicherer man sich sein möchte, dass die Intervalle den wahren Wert inkludieren, desto größer werden auch die Intervalle. Konfidenzintervalle können verringert werden, indem mehr Testpersonen getestet werden bzw. sich die Streuung verringert (worauf wir aber keinen Einfluss haben). Gängige Konfidenzlevel sind 99 %, 95 % und 90 %.

In der nachfolgenden Tabelle finden sich praktische Hinweise zur Berechnung:

Konfidenzintervalle (Normalverteilung)	
Berechnung	$\hat{m} \pm z_{(1-\frac{\alpha}{2})} \frac{\hat{s}}{\sqrt{N}}$ <p> \hat{m}... Mittelwert z... Quantil α... Alphalevel (1 - Konfidenzlevel) \hat{s}... Standardabweichung N... Stichprobengröße </p> <p>Nachschlagen des Quantils:</p>

	Konfidenzlevel	α	$Z_{(1-\frac{\alpha}{2})}$
	99 %	0.01	2.58
	95 %	0.05	1.96
	90 %	0.1	1.64
Excel-Funktion	KONFIDENZ.NORM(Alphalevel; Standardabweichung; Stichprobengröße)		
Beispiel	Testperson	Task Time (sec)	
	P1	30	
	P2	40	
	P3	40	
	P4	44	
	P5	65	
	Konfidenzintervall-Breite (95%)	12.6	
	Untere Grenze	29.2	
	Obere Grenze	54.4	
	Unter der Annahme einer Normalverteilung (und einer repräsentativen Stichprobe) liegt daher die Task-Time der Grundgesamtheit mit einer 95%igen Wahrscheinlichkeit zwischen 29,2 Sekunden und 54,4 Sekunden. Die Angabe eines Intervalls verdeutlicht die mit der Schätzung verbundene Unsicherheit, statt eine Punktschätzung des Mittelwerts von 41,8 abzugeben. Das Intervall kann verringert werden, indem mehr Testpersonen getestet werden.		
Signifikanz	Zwei Designs sind signifikant unterschiedlich, wenn sich die Konfidenzintervalle nicht überlappen.		

Genauere Tests können mithilfe von **t-Tests** durchgeführt werden, welche nicht direkt auf der Normalverteilung, sondern auf der studentischen t-Verteilung aufbauen, die vor allem für kleine Stichproben gut geeignet ist. Die t-Verteilung ist bei kleineren Stichproben etwas „misstrauischer“ und überschätzt die Wahrscheinlichkeiten eines signifikanten Unterschieds weniger als die Normalverteilung. Bei einer Stichprobengröße gegen unendlich konvergieren die beiden Verteilungen.

t-Tests ermitteln die Zufallswahrscheinlichkeit von Variablen (Metriken) zwischen zwei Optionen (Designs) und sie unterscheiden je nach Testsetup zwischen abhängigen (bzw. gepaarten) Stichproben und unabhängigen Stichproben. Bei **abhängigen** Stichproben testet dieselbe Menge an Testpersonen beide Designs, d. h., für jede Probandin und jeden Probanden liegt ein gemessener Wert für beide Designs vor. Bei einem solchen Testsetup muss unbedingt sichergestellt werden, dass das zuerst getestete Design alterniert wird, also bei der Hälfte aller Tests zuerst Design A und bei der anderen Hälfte zuerst Design B. Dies dient der Minimierung des „Carryover“-Effekts, also des Lerneffekts zwischen den zwei Designs. Bei abhängigen Stichproben hat man grundsätzlich den Vorteil, dass Unterschiede in den Messungen direkt auf Unterschiede in den Designs zurückgeführt werden können – jede Testperson ist sozusagen ihre eigene „Kontrollgruppe“.

In der nachfolgenden Tabelle finden sich praktische Hinweise zur Berechnung:

Abhängige t-Tests																									
Berechnung	$t = \frac{\hat{D}}{\frac{\hat{s}_D}{\sqrt{N}}}$ <p>t..... Teststatistik (t-Quantil) \hat{D}..... Mittelwert der Unterschiede \hat{s}_D..... Standardabweichung der Unterschiede N..... Stichprobengröße</p> <p>Nachschlagen des p-Werts anhand der Teststatistik t in einer t-Tabelle oder in Excel</p>																								
Excel-Funktion	<p>Mit T.VERT.2S(t; $N - 1$) kann der p-Wert nachgeschlagen werden, wenn man t händisch ausgerechnet hat. Die Funktion T.TEST(Werte1;Werte2;2;1) führt den gesamten t-Test durch und liefert p zurück (inkl. Berechnung von t und Nachschlagen von p).</p>																								
Beispiel	<table border="1"> <thead> <tr> <th rowspan="2">Testperson</th> <th colspan="2">Task</th> <th rowspan="2">Differenz (sec)</th> </tr> <tr> <th>Time 1 (sec)</th> <th>Task Time 2 (sec)</th> </tr> </thead> <tbody> <tr> <td>P1</td> <td>30</td> <td>44</td> <td>-14</td> </tr> <tr> <td>P2</td> <td>30</td> <td>45</td> <td>-15</td> </tr> <tr> <td>P3</td> <td>40</td> <td>65</td> <td>-25</td> </tr> <tr> <td>P4</td> <td>44</td> <td>31</td> <td>13</td> </tr> </tbody> </table>			Testperson	Task		Differenz (sec)	Time 1 (sec)	Task Time 2 (sec)	P1	30	44	-14	P2	30	45	-15	P3	40	65	-25	P4	44	31	13
Testperson	Task		Differenz (sec)																						
	Time 1 (sec)	Task Time 2 (sec)																							
P1	30	44	-14																						
P2	30	45	-15																						
P3	40	65	-25																						
P4	44	31	13																						

P5	65	65	0
Mittelwert	41,8	50	-8,2
Std. Abw.	14,4	14,8	14,8
t	-1,23704		
p	0,2837		

Signifikanz

Die Wahrscheinlichkeit liegt also bei 28,37 %, dass die Unterschiede nur zufällig gemessen wurden und zwischen den Designs gar kein Unterschied besteht. Dieses Ergebnis ist also nicht signifikant.

Zwei Designs sind signifikant unterschiedlich, wenn der p-Wert unterhalb des Signifikanzlevels liegt, z. B. 10 %, 5 %, 1 %.

Für die berechneten Differenzen können wiederum Konfidenzintervalle, so wie sie vorhin vorgestellt wurden, berechnet werden, sodass für den gemessenen Unterschied auch eine Analyse des praktischen Nutzens möglich ist. Inkludiert das berechnete Konfidenzintervall die 0, so besteht typischerweise kein signifikanter Unterschied.

Bei **unabhängigen Stichproben** werden zwei verschiedene Mengen an Testpersonen verwendet, um die zwei Designs zu testen, d. h., jede Testperson testet auch nur ein Design. Hier können gemessene Unterschiede daher zusätzlich auch noch durch Unterschiede bei den Testpersonen selbst bedingt sein. Die Stichprobengrößen müssen nun dafür nicht zwangsweise gleich groß sein.

In der nachfolgenden Tabelle finden sich praktische Hinweise zur Berechnung:

Unabhängige t-Tests

Berechnung

$$t = \frac{\hat{x}_1 - \hat{x}_2}{\sqrt{\frac{\hat{s}_1^2}{N_1} + \frac{\hat{s}_2^2}{N_2}}}$$

t Teststatistik (t-Quantil)

\hat{x}_1, \hat{x}_2 Mittelwerte der zwei Stichproben

\hat{s}_1, \hat{s}_2 Standardabweichungen der zwei Stichproben

N_1, N_2 Stichprobengrößen

Nachschlagen des p-Werts anhand der Teststatistik t

Excel-Funktion

in einer t-Tabelle oder in Excel

Mit $T.VERT.2S(t; N_1 + N_2 - 2)$ kann der p-Wert nachgeschlagen werden, wenn man t händisch ausgerechnet hat.

Die Funktion $T.TEST(Werte1;Werte2;2;2)$ führt den gesamten t-Test durch und liefert p zurück (inkl. Berechnung von t und Nachschlagen von p).

Beispiel

	Task Time Design A (min)	Task Time Design B (min)
	3	4
	2	3
	4	5
	2	6
	3	5
	5	4
	3	5
	4	5
	5	4
Mean	3,4	4,6
Std. Dev.	1,1	0,9
t	-2,324953	
p	0,0336	

Die Wahrscheinlichkeit liegt also bei 3,36 %, dass die Unterschiede nur zufällig gemessen wurden und zwischen den Designs gar kein Unterschied besteht. Dieses Ergebnis ist also signifikant, wenn ein Signifikanzniveau von 5 % gewählt wird.

Signifikanz

Zwei Designs sind signifikant unterschiedlich, wenn der p-Wert unterhalb des Signifikanzlevels liegt, z. B. 10 %, 5 %, 1 %.

Für die Berechnung von Konfidenzintervallen bei unabhängigen Stichproben wird eine leicht abgewandelte Formel verwendet: $(\hat{m}_1 - \hat{m}_2) \pm z_{(1-\frac{\alpha}{2})} \sqrt{\frac{\hat{s}_1^2}{N_1} + \frac{\hat{s}_2^2}{N_2}}$

Um kategorische Daten auszuwerten, können χ^2 -Tests durchgeführt werden. Diese sind relativ traditionell und gehen von unabhängigen Stichproben aus. Die passende Excel-Funktion lautet CHIU.TEST(Beobachtete Werte; Erwartete Werte). Leider sind χ^2 -Tests relativ ungenau für kleine Stichprobengrößen, weshalb lieber Alternativen wie der „Fisher exact test mit Yates Korrektur“, „ $N - 1 \chi^2$ Test“ oder der „ $N - 1$ two Proportion Test“ verwendet werden sollten.

Um Proportionen (z. B. Task-Success) auszuwerten, können Wald-Konfidenzintervalle verwendet werden, welche anhand der Formel $\hat{p} \pm z_{(1-\frac{\alpha}{2})} \sqrt{\frac{\hat{p}(1-\hat{p})}{N}}$ berechnet werden können. Für eine höhere Genauigkeit bei kleinen Stichprobengrößen sollten allerdings lieber „Adjusted Wald“-Konfidenzintervalle verwendet werden.

Mit den vorgestellten Methoden können nun beliebige Metriken zweier Designs gegenübergestellt und ausgewertet werden. Um mehrere Metriken zu kombinieren und dann gesamtheitlich gegenüberzustellen, können diese in einen einheitlichen UX-Score umgewandelt werden. Hierfür gibt es verschiedene Verfahren:

- **Basierend auf Zielen**
 - Hier wird für jede Metrik ein Zielwert festgelegt
 - Pro Testperson wird nun berechnet, ob dieses Ziel erreicht wurde
 - Beispiel: Testpersonen sollen mind. 70 % aller Aufgaben erfolgreich erfüllen können und pro Aufgabe nicht mehr als zwei Minuten benötigen
 - Schließlich kann ein Prozentsatz berechnet werden, wie oft dieses Ziel insgesamt von allen Testpersonen erreicht wird
- **Basierend auf Prozenten**
 - Hier werden für jede Metrik ein Maximum und ein Minimum definiert
 - Nun kann für jede Testperson ein Score in Prozent berechnet werden
 - Dieser Score kann über alle Metriken und Testpersonen gemittelt werden
- **Single Usability Metric (SUM)**
 - Hierbei handelt es sich um eine standardisierte Methode zum Berechnen einer UX-Metrik
 - Basiert auf Task-Success, Task-Time, Fehleranzahl und User-Bewertung
 - Online-Rechner und Details: <http://usabilityscorecard.com/>
- **Usability Scorecards**

- Usability Scorecards haben das Ziel, die allgemeine Usability grafisch darzustellen
- Verwendet häufig die Prozentmethode als Basis
- Typische Diagramme: Netzdiagramm, kombiniertes Balkendiagramm / Liniendiagramm

Referenz

[43], [44]

LO-7.2.3	Ergebnisse eines Usability-Tests auswerten und interpretieren (K4) – 45 min
----------	---

Den Seminarteilnehmerinnen und -teilnehmern werden zwei verschiedene Datensätze zu Verfügung gestellt, welche mit den gelernten Methoden und unter aktiver Hilfe der bzw. des Vortragenden ausgewertet werden sollen. Hierfür bieten sich beispielsweise die Berichte der Comparative Usability Evaluation (CUE) an, welche öffentlich zugänglich sind und thematisch gut passen würden, z. B. CUE8. Die CUE-Reports beschäftigen sich damit, inwiefern Usability-Methoden selbst reproduzierbar sind.

7.3

Stichprobengröße – wie viele Tests werden benötigt (K2)

40 Minuten

LO-7.3.1	Unterschiedliche Arten von Usability-Studien kennen (K2) – 10 min
----------	---

Begriffe

Formative Studie, summative Studie

Um sich mit Stichprobengrößen auseinandersetzen zu können, müssen zuerst die Einteilungen von Usability-Studien vorgestellt werden. Grundsätzlich kann zwischen formativen und summativen Studien unterschieden werden:

- Formative Studien

- Metapher: Koch probiert während des Zubereitens regelmäßig die Speise, bevor sie serviert wird
- Die Usability-Spezialistin bzw. der -Spezialist prüft das Produkt regelmäßig und erstellt Verbesserungsvorschläge
- Dieser Prozess wird wiederholt, um die Usability des Produkts zu maximieren
- Die Leitfragen sind hier:
 - Was sind die schwerwiegendsten Usability-Probleme, welche unsere Benutzerinnen und Benutzer davon abhalten, Aufgaben erfolgreich und effizient abzuschließen?
 - Was frustriert Benutzerinnen und Benutzer?
 - Was sind die häufigsten Fehler, die gemacht werden?
 - Wird das Design von Iteration zu Iteration verbessert?

- Welche Usability-Probleme bleiben bestehen, wenn das Produkt veröffentlicht wird?
- **Summative Studien**
 - Metapher: Ein Kritiker evaluiert und bewertet ein fertiges Gericht
 - Ein Produkt wird dahin gehend evaluiert, inwiefern es seinen Zweck erfüllt
 - Verschiedene Produkte werden miteinander verglichen
 - Nachbereitungsaufgaben werden definiert (z. B. Design verbessern)
 - Die Leitfragen sind hier
 - Haben wir die Usability-Ziele unseres Projekts erreicht?
 - Was ist die generelle Usability des Produkts?
 - Wie ist das Produkt im Vergleich zu Konkurrenzprodukten?
 - Wird das Produkt von Release zu Release verbessert?

Abhängig vom Typ der Studie ist die optimale Stichprobengröße anders zu berechnen.

Referenz

[43], [44]

LO-7.3.2	Stichprobengröße für formative Studien abschätzen können (K2) – 15 min
----------	--

Begriffe

Magic Number 5

In der Statistik hört man oft, dass man mit einer Stichprobengröße von unter 30 keine sinnvollen statistischen Evaluierungen machen kann. Angesichts der hohen Kosten für Usability-Tests ist dies jedoch nicht realistisch, weshalb es ein wichtiger Punkt ist, die optimale Stichprobengröße für die durchzuführende Studie korrekt abzuschätzen.

Formative Studien benötigen weniger Testpersonen, da hier das Ziel nicht das „Beweisen“ bestimmter Sachverhalte ist – das Produkt soll einfach nur verbessert und Usability-Probleme sollen aufgedeckt werden, weshalb hier die Herangehensweise eher qualitativ als quantitativ ist. Die Berechnung der nötigen Anzahl an Testpersonen bei formativen Studien basiert auf simpler Wahrscheinlichkeitsrechnung:

$$P = 1 - (1 - p)^n$$

Die Gesamtwahrscheinlichkeit P, dass ein bestimmtes Problem nach n Testpersonen mindestens einmal aufgetreten ist, basiert auf der geschätzten Problemwahrscheinlichkeit p: p sagt aus, wie wahrscheinlich ein Problem bei einem einzelnen Usability-Test auftritt. Verständlicherweise hat der Wert für p einen erheblichen Einfluss darauf, wie viele Testpersonen benötigt werden –

nach Nielsens Untersuchungen liegt dieser Wert bei ungefähr 0,31, wodurch sich auch Nielsens „Magic Number 5“ ableiten lässt: $P = 1 - (1 - 0.31)^5 = 0.8436$.

Nach dieser Rechnung deckt man daher mit fünf Testpersonen ungefähr 85 % aller Usability-Probleme eines Systems auf (natürlich in Abhängigkeit von den gewählten Aufgabenstellungen etc.).

Der Wert von p kann über verschiedene Faktoren beeinflusst bzw. verbessert werden:

- Testleiterin bzw. Testleiter sollte möglichst erfahren sein
- Mehrere Testleiterinnen bzw. Testleiter verwenden
- Vor allem eher unerfahrene Testpersonen gut beobachten
- Möglichst unterschiedliche Aufgabenstellungen verwenden (komplexe und einfache)
- Usability-Spezialistinnen bzw. -Spezialisten mit fachlichem Wissen einsetzen

Abhängig von der Schätzung für p und der gewünschten Wahrscheinlichkeit, Usability-Probleme aufzudecken, kann also der optimale Wert für n berechnet werden.

Referenz

[43], [44]

LO-7.3.3 Stichprobengröße für summative Studien abschätzen können
(K2) – 15 min

Bei summativen Studien wollen sehr wohl bestimmte Sachverhalte „bewiesen“ werden und nicht einfach nur Probleme aufgedeckt werden, weshalb hier naturgemäß mehr Testpersonen getestet werden müssen. Die Idee zur Herleitung der optimalen Stichprobengröße basiert auf Umformen der Formel für t-Tests:

$$t = \frac{d}{\frac{s}{\sqrt{N}}} \Rightarrow N = \frac{t^2 s^2}{d^2}$$

Dadurch erhalten die Variablen neue Bedeutungen:

- d ist nun die gewünschte Mindestdifferenz, die detektiert werden soll
- s ist die Standardabweichung der Differenzen, welche unbekannt ist und geschätzt werden muss
- t ist die Teststatistik der gewünschten Konfidenz, welche wiederum von N abhängt

Die grundsätzliche Vorgehensweise zur Berechnung von N ist nun folgende:

1. Geeigneten Wert für d wählen (z. B. mich interessieren nur Task-Time-Unterschiede größer als 2 Minuten)

2. s anhand einer der folgenden Methoden schätzen:
 - Anhand von Erfahrungen (z. B. vorige Designiteration)
 - Bei gut abschätzbarem Maximum und Minimum die Daumenregel $s = \frac{\max - \min}{6}$ verwenden
 - Statt konkreten Wert für d zu wählen, d in Abhängigkeit von s wählen, um die Beziehung aufzulösen, z. B.: Wir interessieren uns nur für Unterschiede größer als eine halbe Standardabweichung: $d = 0.5s$
3. Eine erste Schätzung für N unter Verwendung der Normalverteilung berechnen: $N = \frac{z^2 s^2}{d^2}$
4. Aktuelles N als Input für t verwenden und neues N berechnen: $N = \frac{t^2 s^2}{d^2}$ und aufrunden
5. Alten und neuen Wert für N vergleichen
 - Hat sich an N nichts geändert, hat man bereits die optimale Stichprobengröße berechnet
 - Springt N zwischen zwei oder mehreren Werten hin und her, mittelt man die Werte, rundet auf und hat die optimale Stichprobengröße berechnet
 - Erhält man ein komplett neues N , so wiederholt man Schritt 4

Folgendes Beispiel demonstriert diesen Prozess:

- Wir interessieren uns für Differenzen größer als ein Drittel der Standardabweichung ($d = s/3$) mit einer Konfidenz von 80 %
- Die erste Schätzung anhand der Normalverteilung liefert $N = \frac{z^2 s^2}{d^2} \Rightarrow N = \frac{1.282^2 s^2}{(0.3s)^2} = 14.8 \approx 15$
- Dieses N wird für das Nachschlagen des t -Werts verwendet und die nächste Iteration liefert $N = \frac{t^2 s^2}{d^2} \Rightarrow N = \frac{1.345^2 s^2}{(0.3s)^2} = 16.3 \approx 17$
- Verwendet man dieses N abermals zum Nachschlagen von t , so bestätigt sich das N von 17

Dieses Beispiel zeigt also: Um Differenzen von einem Drittel der Standardabweichung mit einer 80%igen Konfidenz detektieren zu können, werden 17 Testpersonen benötigt.

Referenz

[44]

7.4

Standardisierte Befragungsmethoden (K3)

35 Minuten

LO-7.4.1

Unterschiedliche Befragungsmethoden kennen (K2) - 15 min

Begriffe

Post-Task-Rating, Post-Session-Rating

In der Usability unterscheidet man zwischen „Post-Task“-Ratings und „Post-Session“-Ratings. **Post-Task-Ratings** werden verwendet, um besonders schwierige bzw. komplizierte Tasks durch Befragung der Testpersonen schnell und einfach identifizieren zu können. Einige gängige Methoden hierfür sind:

- Ease of Use Ratings
 - o Die Testpersonen werden nach einer Aufgabenstellung gebeten, die Schwierigkeit der Aufgabe auf einer oder mehreren Likert-Skalen oder über ein semantisches Differential zu bewerten
 - o Beispiel „Single Ease Question“:
„Insgesamt fand ich diese Aufgabe ...“
Sehr schwierig o o o o o o o Sehr einfach
 - o Diese Methode hat sich als besonders effektiv herausgestellt
- After-Scenario Questionnaire (ASQ)
 - o Bewertung einer Aufgabenstellung anhand von drei Fragen auf einer klassischen Likert-Skala
 - o Zielt auf die drei ISO-Faktoren Effektivität, Effizienz und Zufriedenstellung ab
 - o Deutsche Übersetzung: „Ich bin mit der Schwierigkeit der Aufgabenstellung zufrieden“, „Ich bin mit der benötigten Zeit der Aufgabenstellung zufrieden“, „Ich bin mit den verfügbaren Informationen (Hilfe, Nachrichten, Dokumentation) für diese Aufgabenstellung zufrieden“
- Expectation Measure
 - o Idee: Vergleich der erwarteten und tatsächlichen Schwierigkeit von Aufgaben
 - o Die Testpersonen bewerten vor der Aufgabe die erwartete Schwierigkeit auf einem Siebenpunkt-Differential
 - o Die Testpersonen bewerten nach der Aufgabe die tatsächliche Schwierigkeit
 - o Durch die Gegenüberstellung der Bewertungen können relativ schnell die höchstproblematischen Tasks identifiziert werden (niedrige erwartete Schwierigkeit und hohe tatsächliche Schwierigkeit)

In breit angelegten Untersuchungen mit über 1.000 Testpersonen wurden verschiedene Post-Task-Rating-Möglichkeiten miteinander verglichen. Die Ergebnisse: Bei großen Stichprobengrößen konvergieren alle Methoden und es besteht kein großer Unterschied, während bei kleinen Stichprobengrößen die erste Variante (Ease of Use Rating) zu bevorzugen ist, da hier mit sehr wenigen Befragungen bereits gut reproduzierbare Ergebnisse erzielt werden können.

Post-Session-Ratings werden hingegen verwendet, um die generelle Usability des Gesamtsystems sinnvoll bewerten und vergleichen zu können. Auch hier gibt es viele verschiedene standardisierte Möglichkeiten:

- System Usability Scale (SUS)
 - Sehr weit verbreiteter standardisierter Fragebogen
 - Schnell und einfach zu beantworten
 - Inkludiert 10 Statements, zu denen die Befragten ihre Zustimmung auf einem sematischen Differential von 1 (strongly disagree) bis 5 (strongly agree) angeben
 - Die Hälfte der Statements ist positiv formuliert, die andere Hälfte negativ
 - Bei der Auswertung ergibt sich eine SUS-Score von 0 bis 100
 - Generelle Interpretation der Score:
 - Geringer als 50: inakzeptabel
 - 50 bis 70: grenzwertig
 - Über 70: akzeptabel
- Computer System Usability Questionnaire (CSUQ)
 - 19 Statements zur Bedienung des Systems
 - Testpersonen geben den Grad der Zustimmung auf einer Skala von 1 (strongly disagree) bis 7 (strongly agree) an
 - Alle Statements sind positiv formuliert
 - Vier Hauptkategorien (Nützlichkeit, Informationen, Interface und Zufriedenheit)
- Questionnaire for User Interface Satisfaction (QUIS)
 - 27 Statements, die in fünf Kategorien unterteilt sind (Reaktion, Oberfläche, Terminologie/Information, Lernbarkeit, Systemfähigkeiten)
 - Die Testpersonen entscheiden sich zwischen gegensätzlichen Attributen auf einer Skala von 0 bis 9
- Usefulness, Satisfaction and Ease-of-Use Questionnaire (USE)
 - 30 Statements, die in vier Kategorien unterteilt sind (Nützlichkeit, Zufriedenstellung, Einfachheit und Lernbarkeit)
 - Testpersonen geben den Grad der Zustimmung auf einer Likert-Skala von 1 bis 7 an
- Product Reaction Cards
 - Ein komplett anderer Ansatz von Microsoft, bei welchem sich die Testpersonen von 118 Karten mit Eigenschaftswörtern die Top 5 Karten auswählen, die das System am besten beschreiben
 - Qualitativer Input, indem die Testpersonen erklären, wieso sie die Karten ausgewählt haben
 - Quantitativ auswertbar, indem die Häufigkeiten der gewählten Wörter gezählt werden

Auch hier wurden Untersuchungen zur Reproduzierbarkeit verschiedener Post-Session-Rating-Methoden angestellt. Basierend auf Tests mit 123 Testpersonen hat sich herausgestellt, dass die Ergebnisse des SUS bei so ziemlich

allen Stichprobengrößen überlegen sind – konkret wurde gemessen, wie hoch die Wahrscheinlichkeit ist, das bekannte „richtige“ Design auszuwählen, wenn ein Subset der Beantwortungen der 123 Testpersonen ausgewählt wird und als Grundlage für diese Entscheidung genommen wird.

Referenz

[43]

LO-7.4.2	Den System Usability Scale (SUS) auswerten können (K3) – 20 min
----------	---

Immer wieder hat sich der SUS als beste und einfachste Möglichkeit herausgestellt, Post-Session-Ratings einzuholen. Ob dies an der Mischung aus positiven und negativen Statements oder an der guten Wahl der Fragen liegt, ist nicht ganz bekannt – das Wichtigste ist allerdings die Erkenntnis, dass der SUS funktioniert.

Die nicht übersetzten Fragen des SUS lauten:

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

Nachdem die Testpersonen ihren Grad an Zustimmung auf einer Skala von 1 (strongly disagree) bis 5 (strongly agree) angegeben haben und alle Daten gesammelt wurden, kann die Auswertung beginnen. Der Gesamtscore pro Fragebogen berechnet sich folgendermaßen:

- Für die positiven Fragen 1, 3, 5, 7 und 9 entspricht der Fragenscore der angekreuzten Skalenposition minus 1, z. B. „strongly agree“ = $5 - 1 = 4$
- Für die negativen Fragen 2, 4, 6, 8 und 10 entspricht der Fragenscore 5 minus der angekreuzten Skalenposition, z. B. „strongly disagree“ = $5 - 1 = 4$
- Die Fragenscores werden aufsummiert und schließlich mit 2,5 multipliziert, damit der Gesamtscore auf einen Wertebereich von 0 bis 100 skaliert wird

Nun liegt für jede Testperson ein SUS-Score vor, welcher wie gewohnt als normale Metrik behandelt und zum Vergleich verschiedener Designs herangezogen werden kann.

Übung: Die Seminarteilnehmerinnen und -teilnehmer bekommen vorbereitete Datensätze oder bewerten selbst zügig zwei Systeme, welche gemeinsam bestimmt werden. Der SUS-Score wird für beide Systeme berechnet und dient als Grundlage für die Entscheidung, welches System zu bevorzugen ist bzw. wie der Gesamtscore zu interpretieren ist. Eventuell kann auch ein Template zur Auswertung und Visualisierung verwendet werden.

Referenz

[43]

Referenzen und Literatur

- [1] Jacob Nielsen, Designing Web Usability: The Practice of Simplicity, Indianapolis, 1999
- [2] Michael Herczeg, Software-Ergonomie: Grundlagen der Mensch-Computer-Kommunikation, Bonn, 1994
- [3] Weimarer Erklärung,
http://www.udgermany.de/html/ud/g/Espacio_Vital_2010/charta_UD_14nov092.pdf
- [4] S. Schubert & C. Eibl, Die 3 Gestaltgesetze, Universität Siegen, Didaktik der Informatik und E-Learning
- [5] Franz Docekal, <http://www.lenslens.ch/>, Adliswil
- [6] Dr. Dirk Struve, Design Disaster und Usability: Einführung in die Gebrauchstauglichkeit, 2005
- [7] Prof. Dr. Schmidt, Prof. Dr. Butz, Vorlesung „Mensch-Maschine-Interaktion, 2003/2004
- [8] Philip Zimbardo, Psychologie, Berlin, 1999
- [9] DIN EN ISO 9241: Ergonomie der Mensch-System-Interaktion – Teil 110
- [10] Wolfgang Schneider, Ergonomische Gestaltung von Benutzungsschnittstellen: Kommentar zur Grundsatznorm DIN EN ISO 9241-11, 2008
- [11] Verordnung über Sicherheit und Gesundheitsschutz bei der Arbeit an Bildschirmgeräten (Bildschirmarbeitsverordnung – BildscharbV, <http://bundesrecht.juris.de/bundesrecht/bildscharbv/gesamt.pdf>
- [12] World Wide Web Consortium, www.w3.org
- [13] E-Government-Handbuch: Qualitätskriterien für einen bürgerfreundlichen und sicheren Web-Auftritt, https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Egovernment/4_Qualit_pdf.pdf?__blob=publicationFile
- [14] Gesetz zur Gleichstellung behinderter Menschen / Behindertengleichstellungsgesetz (BGG) <http://www.gesetze-im-internet.de/bgg/index.html>
- [15] Verordnung zur Schaffung barrierefreier Informationstechnik nach dem Behindertengleichstellungsgesetz (Barrierefreie Informationstechnik-Verordnung - BITV), <http://www.gesetze-im-internet.de/bitv/BJNR265400002.html>
- [16] Jan Eric Hellbusch, <http://www.barrierefreies-webdesign.de/richtlinien/global/weltweit.html>
- [17] Informationspool Computerhilfsmittel für Blinde und Sehbehinderte, <http://www.incobs.de/>
- [18] Deutsche Akkreditierungsstelle (DAkkS), Leitfaden Usability, http://www.dakks.de/sites/default/files/71-SD-2-007_Leitfaden%20Usability%201.3.pdf
- [19] Prof. Dr. rer. pol. Thomas Urban, FH Schmalkalden

- [20] Prof. Dr. Christian Stary, Hannes Gotthartsleitner, Ing. Mag. Peter Eberle, Zur Verschränkung von User Experience und Usability Engineering: Merkmale, Prinzipien und Vorgehensmodelle, Linz, 2009
- [21] Steve Krug, Don't make me think! Web Usability: Das intuitive Web, 2. Auflage 2006, Redline Heidelberg
- Michael Richter, Markus Flückiger, Usability Engineering kompakt, 2. Auflage 2010, Spektrum Verlag Heidelberg.
- [22] HW Hunziker, Im Auge des Lesers: vom Buchstabieren zur Lesefreude; foveale und periphere Wahrnehmung, 2006, transmedia verlag, Zürich
- [23] S.M. Weinschenk, 100 Dinge, die jeder Designer über Menschen wissen muss, 2011, Addison Wesley Verlag
- [24] Alan Cooper, The Inmates Are Running the Asylum: Why High-tech Products Drive Us Crazy and How to Restore the Sanity, 2004, SAMS
- [25] Jeff Gothelf with Josh Seiden, LEAN UX. Applying Lean Principles to Improve User Experience. O'Reilly Media, 2013
- [26] <https://play.google.com/store/apps/details?id=com.yoc.swiss.swiss&hl=de> (01.08.2105)
- [27] Steve Blank, Das Handbuch für Stardups, O'Reilly Media 2013, ISBN 9783955618124
- [28] Steve Blank, <http://steveblank.com/tools-and-blogs-for-entrepreneurs/> (01.08.2105)
- [29] Derek Sivers, Why You Need to Fail, <https://www.youtube.com/watch?v=HhxcFGuKOys> (01.08.2105)
- [30] Dan Olsen, The Lean Product Playbook: How to Innovate with Minimum Viable Products and Rapid Customer Feedback. Erscheint 17. Juni. 2015
- [31] L. Trenner, The Politics of Usability: A Practical Guide to Designing Usable Systems in Industry (Practitioner Series), Springer; Auflage: Softcover reprint of the original 1st ed. 1998
- [32] Clayton M. Christensen, The Innovator's Dilemma: The Revolutionary Book That Will Change the Way You Do Business, 1997, Harvard Business School
- [33] John P. Kotter: Das Pinguin-Prinzip. Wie Veränderung zum Erfolg führt. Verlag Droemer/Knauer, November 2006
- [34] Axel Kaune: Moderne Organisationsentwicklung – ein Konzept zur mitarbeiterorientierten Gestaltung von Veränderungsprozessen. In: Axel Kaune (Hrsg.): Change Management mit Organisationsentwicklung – Veränderungen erfolgreich durchsetzen. Berlin 2004, S. 11-58
- [35] Thomas Lauer: Change Management - Grundlagen und Erfolgsfaktoren Berlin 2010, ISBN 978-3642043390
- [36] Warfel 2015, <https://vimeo.com/37861987> (01.08.2105)

- [37] Nielsen [Norman 2003]. Paperprototyping <https://vimeo.com/43774624> (01.08.2015)
- [38] Steve Krug. Rocket Surgery Made Easy. New Riders 2010, ISBN 0-321-65729-2
- [39] Jeffrey Rubin und Dana Chisnell, Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests. Wiley Publishing 2008, ISBN 978-0-470-18548-3
- [40] Carol M. Barnum. Usability Testing Essentials. Ready, Set...Test!, Elsevier 2011, ISBN 978-0-12-375092-1
- [41] J. Nielsen, Usability Engineering, 1993
- [42] J. Nielsen et al., Usability Return on Investment (ROI), 4th ed., 2012
- [43] T. Tullis und B. Albert, Measuring the User Experience, 2nd ed., 2013
- [44] J. Sauro und J. R. Lewis, Quantifying the User Experience, 2012