

Software Usability immer im Blick

Eine Einführung in den User Centered Design Process

Jonas Zimmermann, Mixed Mode GmbH

Der UCDP hilft mit einer systematischen Vorgehensweise dabei, sich benötigtes Wissen anzueignen, es auf die essentiellen Punkte zu komprimieren und daraus ein Produkt mit bestmöglichem Kundennutzen zu erzeugen. Im Mittelpunkt des User Centered Design Process (UCDP) stehen der Benutzer und die möglichst enge Zusammenarbeit mit ihm. Die vier Phasen sind durch den UCDP vorgegeben, das Team ist jedoch frei in der Wahl der Mittel.

„Eine Software mit tollen Funktionen nützt mir nichts, wenn ich diese Funktionen nicht finde oder verwenden kann.“ Die Meisten werden dieser Aussage wohl sofort zustimmen – und trotzdem gibt es immer noch zu oft Software, bei der man schnell bemerkt, dass die Usability bei der Entwicklung nicht im Fokus stand.

Eine hohe Usability ist – laut Definition in ISO 9241-210 – erreicht, wenn ein bestimmter Benutzer ein Produkt in einem bestimmten Nutzungskontext so nutzen kann, dass seine festgelegten Ziele effektiv, effizient und zufriedenstellend erreicht werden. Effektivität ergibt sich hierbei aus der Genauigkeit sowie Vollständigkeit, mit der dieser Benutzer bestimmte Ziele erreicht. Das Verhältnis der auf dem Weg eingesetzten Ressourcen (z.B. Aufwand, Kosten, Materialien) zu den erreichten Ergebnissen beschreibt die Effizienz. Wahrnehmungen und Reaktionen, die durch die Benutzung des Produktes entstehen, bestimmen den Grad der Zufriedenstellung des Benutzers.

Die Gründe für die Probleme der User mit einem bestimmten Produkt können vielfältig sein. Die Palette reicht hierbei von schwer zu findenden Funktionalitäten über solche, die nicht alle Aufgaben erfüllen können bis hin zu komplett fehlenden. Jedem ist klar, dass dies nicht die Absicht des Entwicklungsteams gewesen ist, sondern dass nach bestem Wissen und Gewissen gearbeitet wurde. In den meisten Fällen liegt die wirkliche Ursache in einem (unbewussten) Mangel an Wissen begründet. Was liegt also näher, als sich sein Wissen direkt an der Quelle - also beim Benutzer - anzueignen?

Schon Ende der 80er Jahre hat sich Don Norman (zusammen mit Jakob Nielsen einer der Urväter auf den Gebieten Usability und User Experience) mit User Centered Design beschäftigt. Bereits damals hat er darauf hingewiesen, dass man bei der Entwicklung von Produkten viel mehr auf den Benutzer und seine Bedürfnisse eingehen muss. Im Laufe der Jahre ist aus den ersten Empfehlungen eine eigenständige Vorgehensweise entstanden: der User Centered Design Process (UCDP). Er hilft mit einer systematischen Vorgehensweise dabei, sich das benötigte Wissen über Benutzer und Aufgaben anzueignen, es auf die essentiellen Punkte zu komprimieren und daraus ein Produkt mit bestmöglichem Kundennutzen zu erzeugen.

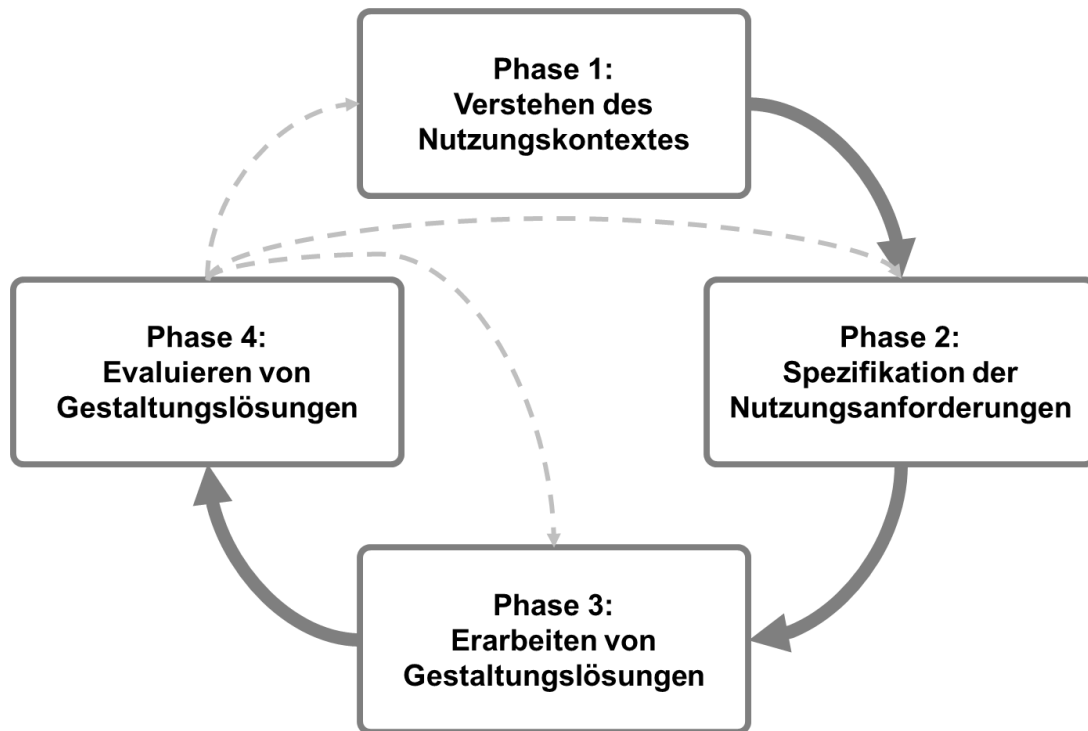


Abbildung 1: Der User Centered Design Process mit seinen vier Phasen

Der UCDP schafft den Spagat zwischen Flexibilität auf der einen und klarer Struktur auf der anderen Seite. Obwohl es vier deutlich voneinander abgegrenzte Phasen mit klaren Zielen gibt, hat das Entwicklungsteam eine große Freiheit sowohl in der Wahl des Softwareentwicklungsprozesses als auch in der Auswahl der Methoden innerhalb der einzelnen Phasen.

Im Mittelpunkt steht der Benutzer. Obwohl in jeder der vier Phasen andere Aufgaben zu lösen sind, orientiert sich das Team jederzeit an den Benutzern und ihrer ganz spezifischen Situation.

Wichtig beim User Centered Design Process: Iteratives Vorgehen, d. h. die vier Phasen werden in der Regel nicht nur einmal durchlaufen. Nach der Evaluation wird je nach Ergebnis eine der drei anderen Phasen bearbeitet – abhängig von den Mängeln, die festgestellt worden sind. Dieser Ablauf wird so oft wiederholt, bis die Evaluation zu einem befriedigenden Ergebnis kommt.

Phase 1: Verstehen des Nutzungskontextes

In der ersten Phase beschäftigt sich das Team mit den verschiedenen Benutzergruppen seines zukünftigen Produktes. Mit Hilfe verschiedener Techniken (z.B. Beobachtung, kontextuelle Interviews, Schüler-Meister-Modell, ...) sammelt das Team Erkenntnisse über den Nutzungskontext, um mindestens die folgenden Fragen beantworten zu können:

Wer sind die Stakeholder, d.h. wer benutzt das System oder ist von den Ergebnissen der Nutzung betroffen? Welche Benutzer gibt es und welche Ziele sowie Aufgaben haben sie? Welche Daten werden während der Nutzung verarbeitet? Wie ist die

Ausstattung der Benutzer? In welcher Umgebung (aus technischer, physischer und sozialer Sicht) arbeiten sie?

Die Antworten auf diese Fragen sollten sich am Ende dieser Phase in einer Nutzungskontextbeschreibung sowie Sammlungen von Szenarien und Persona finden lassen.

Man könnte in Versuchung geraten, diese Phase ohne direkte Interaktion mit Benutzern zu durchlaufen, allerdings würde das den Verzicht auf einen Großteil der zu entdeckenden Informationen bewirken. Doch Vorsicht: Benutzer werden schnell betriebsblind und formulieren Bedürfnisse oft als konkrete Erweiterungen der bestehenden Lösung. Die viel wertvollere Erkenntnis ist das Problem, das der Benutzer mit seinem Vorschlag lösen möchte. Sobald sich das Entwicklungsteam aller Probleme und Aufgaben bewusst ist, kann es passende Lösungen und Abläufe entwickeln, die oft weit über die geäußerten Wünsche der Benutzer hinausgehen.

Zwei hervorragende Varianten, die sich simplifiziert mit „über die Schulter gucken“ beschreiben lassen, sind die Beobachtung oder das Meister-Schüler-Modell. In beiden Fällen sollte das Sammeln der Informationen am Arbeitsplatz des Benutzers stattfinden und von einem Teammitglied durchgeführt werden. Bei der Beobachtung ist die Hauptaufgabe des Teammitglieds das Aufnehmen der Aufgaben und Abläufe. Die Kommunikation mit dem Benutzer sollte auf ein Minimum beschränkt sein, so dass nur in seltenen Fällen eine klärende Frage gestellt wird, wenn das Verhalten des Benutzers nicht verstanden wird. Der große Vorteil liegt darin, dass der Benutzer tatsächlich unbeeinflusst seinem täglichen Workflow folgt und der Beobachter so einen sehr realitätsgetreuen Eindruck der Arbeit bekommt. Das Meister-Schüler-Modell geht noch einen Schritt weiter: Das Teammitglied versucht hierbei wie in einem Crash-Kurs das Handwerk des Benutzers zu erlernen.

Phase 2: Spezifikation der Nutzungsanforderungen

Das Ziel der zweiten Phase ist es, die bisherigen Erkenntnisse in Nutzungsanforderungen niederzuschreiben. Eine systematische Vorgehensweise sieht vor, den Nutzungskontext zuerst auf Erfordernisse („User Needs“) zu untersuchen und sie in einem weiteren Schritt in Nutzungsanforderungen zu konvertieren. Ein Erfordernis entspricht hierbei einer Voraussetzung, um ein Ziel innerhalb eines bestimmten Nutzungskontexts zu erreichen und lässt sich aus den Szenarien ableiten, die in der ersten Phase herausgearbeitet wurden.

Das immer wieder genannte Paradebeispiel ist der Vortragende, der jederzeit wissen muss, wie viel Zeit er noch hat, um seinen Vortrag im festgelegten Zeitrahmen zu beenden. Eine Anforderung, die sich aus diesem Erfordernis ableiten lässt, ist die ständige Sichtbarkeit der verbleibenden Zeit in der Referentenansicht einer Präsentationssoftware.

Phase 3: Erarbeiten von Gestaltungslösungen

In der „kreativen Phase“ geht es schließlich um die Umsetzung der Anforderungen in einen Prototypen. Hierbei ist nicht festgelegt, wie gut er ausgearbeitet werden muss. Gerade zu Beginn ist es in den meisten Fällen ratsam, mit simpleren Versionen zu arbeiten: einfache Skizzen, Wireframes oder auch Designs aus Haftnotizen sind schnell erzeugt und reichen für grundlegende Diskussionen vollkommen aus.

Es geht hierbei nicht nur darum, das Aussehen der Oberfläche zu definieren, sondern vor allem um Struktur und Verhalten der Software. Im ersten Schritt sollte man sich also Gedanken darüber machen, mit welchen Interaktionen die Aufgaben gelöst werden sollen und wie sie zusammenhängen. Eine einfache Technik zur Strukturierung (z.B. für eine Menüstruktur) ist das Card Sorting. Hierbei bekommt ein Testkandidat einen Stapel Karten mit den vorhandenen Begriffen und soll diese in für ihn sinnvolle Gruppen bringen.

Der User Centered Design Process hat natürlich auch keine Wunderwaffe in der Hinterhand, die zwangsläufig hervorragend aussehende und bestens bedienbare Oberflächen produziert. Es gibt aber durchaus einige Grundsätze, die man bei der Gestaltung immer im Hinterkopf haben sollte. Hierbei sind vor allem die Grundsätze der Dialoggestaltung nach ISO 9241-110 (Aufgabenangemessenheit, Selbstbeschreibungsfähigkeit, Erwartungskonformität, Lernförderlichkeit, Steuerbarkeit, Fehlertoleranz, Individualisierbarkeit) und die Nielsen-Heuristiken [1] zu nennen. Vereinfachte Beispiele für Heuristiken sind etwa „Beachte Konventionen der verwendeten Plattform“, „Hilf den Usern dabei Fehler zu erkennen, sie zu verstehen und schlage eine Lösung für das Problem vor“ oder auch „Informiere den User durch angemessenes Feedback über den aktuellen Zustand“.

Phase 4: Evaluieren von Gestaltungslösungen

Je nach Situation gibt es unterschiedliche Techniken, um die erstellten Designs zu evaluieren. Dank der großen Unterschiede hinsichtlich Zeit, Aufwand und Kosten sollte sich entsprechend der Gegebenheiten immer eine passende Variante finden lassen: die Bandbreite reicht von einer Heuristischen Inspektion über Benutzerbefragungen bis hin zum Usability Test.

Bei der Heuristischen Inspektion (auch Heuristisches Review genannt) wird der Designvorschlag mit einer Liste von Heuristiken verglichen. Die Inspektion wird idealerweise von einem oder mehreren Usability-Experten durchgeführt und dient somit hauptsächlich zur Überprüfung hinsichtlich der Usability (und nicht so sehr der Nützlichkeit). Die Vorzüge dieser Technik liegen sowohl in der leichten Durchführbarkeit als auch dem vergleichsweise geringen Aufwand – aber eben auf Kosten der Vielfalt möglicher Erkenntnisse.

Benutzerbefragungen mit Hilfe von Fragebögen sind sowohl in der Vorbereitung als auch in der Interpretation mit deutlich mehr Aufwand verbunden, versprechen dafür aber auch einen deutlich größeren Informationsgewinn. Hierzu trägt neben der größeren Reichweite auch die konkrete Kommunikation mit dem Benutzer bei. In dieser Hinsicht ist übrigens zu beachten, dass der Fragebogen selber auch auf seine Usability hin optimiert sein sollte. Wenn die Benutzer Schwierigkeiten beim Ausfüllen des Fragebogens haben und Fragen bzw. Antworten falsch interpretieren, so kann es leicht zu fehlerhaften Schlüssen bei der Auswertung kommen.

Die Königsdisziplin der Evaluation ist wohl der Usability Test. Bei jeder Testsitzung versucht ein Testteilnehmer, vorher definierte Aufgaben mit dem System (oder einem Prototypen) zu lösen, während er im Optimalfall „laut denkt“, also seine Überlegungen ausspricht. Ein Moderator führt den Teilnehmer durch die Testsitzung, ohne jedoch bei der Bearbeitung der Aufgaben zu helfen. Im Hintergrund können Beobachter und Protokollant direkt erkennen, welche Aufgaben intuitiv lösbar sind

und welche nicht. Um den Testteilnehmer nicht unnötig abzulenken, sollte außer dem Moderator keine andere Person sichtbar sein. Die Möglichkeiten zur „heimlichen“ Beobachtung sind auch hier vielfältig: z.B. ein Beobachtungsraum, der über einen halbdurchlässigen Spiegel vom Testraum getrennt ist oder eine Webcam in Verbindung mit einer Kopie dessen, was auf dem Monitor des Teilnehmers passiert. Ein zusätzlicher Eye-Tracker kann sogar Informationen liefern, die dem Teilnehmer selbst gar nicht bewusst sind – nämlich die Elemente, die dem Teilnehmer zuerst ins Auge fallen.

Missverständnisse: Nur agil und viel zu teuer?

Der User Centered Design Process mag auf den ersten Blick so wirken als wäre er nur in einem agilen Umfeld verwendbar – tatsächlich kann man ihn aber auch in ein klassisches Wasserfallmodell einbetten. Ein Beispiel hierfür kann man sich auf der Website des U.S. Department of Health & Human Services [2] ansehen.

Auch die Entwicklungskosten explodieren nicht automatisch durch die Konzentration auf den User. Zum einen lassen sich durch die Wahl der verwendeten Techniken die Kosten skalieren und zum anderen bekommt man einen konkreten Mehrwert in vielerlei Form – von ansonsten unentdeckten Bedürfnissen der Benutzer über Bedienungsprobleme bis hin zu einem deutlich besseren Produkt und somit einer besseren Position im Wettbewerb.

Weiterführende Links

- [1] Jakob Nielsen, Ten Usability Heuristics
<https://www.nngroup.com/articles/ten-usability-heuristics/>
- [2] User-Centered Design Process Map
<https://www.usability.gov/how-to-and-tools/resources/ucd-map.html>

Autor



Herr Zimmermann (B.Sc Informatik und Multimedia) bringt Erfahrung in der Softwareentwicklung für grafische Benutzeroberflächen seit 2009 mit. Seit Beginn seiner Laufbahn beschäftigt er sich intensiv mit den Themen Requirements-Engineering und Usability von Software Systemen. Herr Zimmermann ist sowohl “IREB Certified Professional for Requirements-Engineering” als auch “Certified Professional Usability and UX”.

Kontakt

Email: jonas.zimmermann@mixed-mode.de

Mixed Mode GmbH, Lochhamer Schlag 17, 82166 Gräfelfing bei München