



Steckbrief-Sammlung: Usability Engineering-Methoden

15 Methodenelemente für Analyse, Konzeption & Entwicklung, Evaluation

BRAINSTORMING

Aufwand	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	Schwierigkeit	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	Eignung mobil	<input checked="" type="checkbox"/> <input type="checkbox"/>
Abhängigkeiten	---	Projektphasen	Konzeption & Entwicklung	Beteiligte Rollen	Team

Kurzbeschreibung

Spontane Ideenäußerungen bezüglich einer bestimmten Problemstellung werden ohne das Anbringen von Kritik gesammelt. Erst nachdem eine große Anzahl Ideen zusammengetragen wurde, erfolgt eine Bewertung und Auswahl. (vgl. [1])



Ziel

- Ideen im Kollektiv der Gruppe generieren, Ideen der anderen für eigene Ideenfindung nutzen (vgl. [1])
- In kürzester Zeit möglichst viele Ideen generieren und anschließend brauchbare Ideen identifizieren (vgl. [1])

Ressourcen

Werkzeuge	<ul style="list-style-type: none"> • Stift, Whiteboard/Flip Chart etc. • Ggf. Textverarbeitungssoftware 	Teilnehmer	<ul style="list-style-type: none"> • 1 Leiter • mehrere Teilnehmer (vgl. [1]), im Optimalfall 5-7 (vgl. [2], S. 35) 	Zeit	<ul style="list-style-type: none"> • 30–120 min. (vgl. [2], S. 31)
------------------	---	-------------------	---	-------------	---

Vorgehen

Bevor mit der eigentlichen Brainstorming-Sitzung begonnen wird, sollte der Leiter eine Problemanalyse durchführen. Ist das zu behandelnde Problem sehr komplex, sollte es in kleinere Teilprobleme aufgesplittet werden. (vgl. [1])

Zu Beginn des Brainstormings stellt der Leiter den Teilnehmern die aus der Analyse hervorgegangene Fragestellung vor und nennt die vier wichtigsten Brainstorming-Regeln (vgl. [1]):

- 1) Quantität vor Qualität der Ideen.

- 2) Keine Kritik oder Kommentare.
- 3) In sehr kurzer Zeit (ca. 5-30 min.) möglichst viele Ideen.
- 4) Es ist erlaubt, Ideen von anderen Teilnehmern aufzugreifen.

Während des Brainstormings werden nun von den Teilnehmern spontan Ideen geäußert, die vom Leiter für alle sichtbar notiert werden. Der Leiter motiviert die Gruppe dabei nicht nur, sondern hilft auch festgefahrene Situationen aufzubrechen und neue Perspektiven einzubeziehen. (vgl. [1]) „Damit keine Vorschläge verloren gehen oder vergessen werden, während die Teilnehmer darauf warten, an die Reihe zu kommen, können Blätter verteilt werden, auf denen jeder seine spontanen Einfälle, Assoziationen und Gedanken notiert und diese ergänzt, wenn ihn die Äußerung eines Teamkollegen zu einer neuen Idee anregt.“ ([2], S. 32-34)

Nach Ende des eigentlichen Brainstormings erfolgt eine kurze Kreativ-Pause. Abschließend werden danach alle Ideen bewertet und geordnet. So können beispielsweise die brauchbarsten Ideen identifiziert werden. Das Bewerten und Sortieren kann entweder durch die Teilnehmer des Brainstormings aber auch durch andere Personen erfolgen. (vgl. [1])

Hilfsmittel



Templates • ---



Tools • ---

Das sollten Sie beachten

Regeln unbedingt einhalten

Um möglichst gute Resultate zu erzielen, sollten die 4 Brainstorming-Regeln (s. „Vorgehen“) unbedingt eingehalten werden. Besonders wichtig ist es, dass Punkt 2) eingehalten wird, denn so wird nicht unnötig Zeit verschwendet und „unbrauchbare“ Ideen bzw. deren Urheber werden nicht „verurteilt“ (vgl. [3]).

Vorgehen und Regeln vorab klären

Das geplante Vorgehen und die zugehörigen Regeln sollten allen Teilnehmern vorab mitgeteilt werden, so dass sie sich auf das Brainstorming vorbereiten können. Muss das Vorgehen zu Beginn des Meetings erst noch verhandelt und geklärt werden, kann dies zu einem ermüdenden Prozess für alle Teilnehmer werden. In der Regel wird dann – bezogen auf die eigentliche Fragestellung – ein weniger gutes Ergebnis erzielt. (vgl. [3])

Quellen

[1] Designmethodenfinder

<http://www.designmethodenfinder.de/brainstorming?s1> (Geprüft 01.08.2014)

[2] Boos., E. (2011). *Das große Buch der Kreativitätstechniken*. München: Compact Verlag GmbH.

[3] Spool, J. M. (2013). *Perspectives over Power: Habits of Collaborative Team Meetings*.

http://www.uie.com/articles/perspectives_over_power/ (Geprüft 01.08.2014)

CONTEXT MODELLING

Aufwand



1

Schwierigkeit



Eignung mobil



Abhängigkeiten

Nutzungskontext-
analyse (z. B.
Kontextuelles
Interview)

Projektphasen

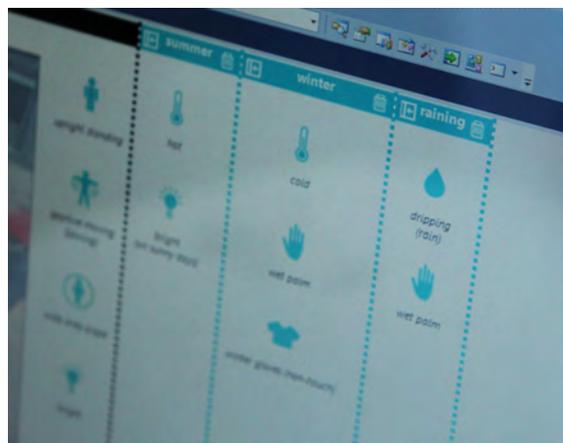
Analyse

Beteiligte Rollen

Software-
Gestalter,
Benutzer
(/Kunde)

Kurzbeschreibung

Anwendung einer Modellierungssprache, um beeinflussende Lokationsfaktoren und situationsbedingte Varianten des Nutzungskontexts einer mobilen Lösung systematisch und strukturiert zu ermitteln. (vgl. [1], S. 24)



Ziel

- Möglichst alle relevanten Informationen, die Einfluss auf die mobile Lösung haben können (erhobener Nutzungskontext: beeinflussende Lokationsfaktoren, situationsbedingte Varianten) mittels einfacher Bildsprache spezifizieren (vgl. [1])
- Durch die strukturierte Erfassung der relevanten Informationen (s. oben) „[...] ein gemeinsames Verständnis unter den Beteiligten für die lokations- und prozessbedingten Besonderheiten [...] [etablieren]“ ([1], S. 17)

Ressourcen

Werkzeuge

- Context Modeller

Teilnehmer

- 1 Kontext-Modellierer
- ggf. interne und/oder externe Personen zur Evaluation

Zeit

- ?²

¹ Eine genauere Einschätzung des Zeitaufwands ist noch nicht bekannt. Es ist aber davon auszugehen, dass der zeitliche Aufwand bei Einsatz des Context Modeller als Unterstützung nicht allzu hoch ausfallen wird.

² s. o.

Vorgehen

Beim Context Modelling werden mithilfe einer Modellierungssprache beeinflussende Faktoren (z. B. lokationsbedingte Besonderheiten) für die Gebrauchstauglichkeit einer mobilen Lösung strukturiert erfasst. (vgl. [1], S. 24)

Als Tool kann hierfür der Context Modeller eingesetzt werden. Er erleichtert das Vorgehen enorm, da er eine speziell für die Entwicklung mobiler Software definierte Modellersprache vorgibt. Außerdem wird ein Mehrwert durch die automatische Ermittlung von Anforderungen an situationsadäquate Geräte im Anschluss an die Spezifikation des Nutzungskontextes generiert (vgl. [1], S. 21).

Mit dem Context Modeller und der zugehörigen Modellierungssprache können drei maßgebliche *Umstandsklassen* abgedeckt werden:

- „*Interaktionseinschränkungen* sind im Wesentlichen durch die Art der Verrichtung gegeben und beschreiben Einschränkungen bezüglich der Interaktion mit einem potentiellen, mobilen Gerät – z. B. Körperhaltung, Handfreiheit, Handschuhe, etc.
- *Verfügbare Infrastruktur* beschreibt die nutzbare Geräteinfrastruktur vor Ort – beispielsweise Halterung, Ablagemöglichkeiten, Stromzufuhr, Datennetze, etc.
- *Umgebungseinschränkungen* sind Einschränkungen, die durch Umwelteinflüsse oder spezielle Raumkonfigurationen gegeben sind – z. B. Lichtverhältnisse, Extremtemperaturen, Geräuschpegel, etc.“ ([1], S. 19)

Insgesamt erfüllt der Modeller dabei folgende Anforderungen:

- „Verständliche Notation für die gesamte Zielgruppe
- An Entwurfsentscheidungen angepasster Abstraktions- und Detailgrad
- Breite Abdeckung mobiler Szenarien
- Berücksichtigung abweichender (Arbeits-)Situations
- Als Werkzeug: idealerweise bei einer Begehung oder im Workshop nutzbar
- Spezifikation in einem Standard Dokumentations-Format transformieren
- Regelsystem für Spezifikation erweiter- und anpassbar“ ([1], S. 17)

Hilfsmittel



Templates

- Checkliste Kontext-Profil



Tools

- Context Modeller

Quellen

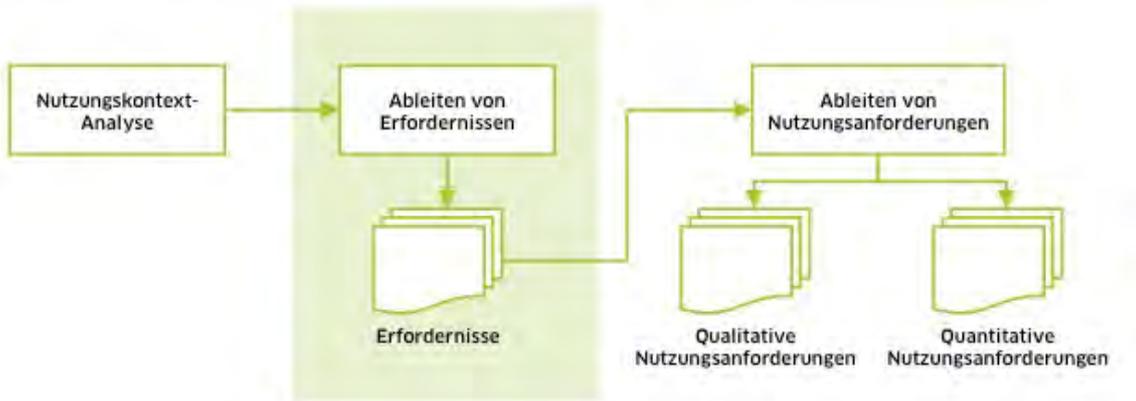
- [1] Ritz, T. & Strauch, J. (2012). Strukturierte Ermittlung beeinflussender Faktoren für mobile Softwarelösungen. In: Back, A., Bick, M., Breunig, M., Pousttchi, K. & Thiesse F. (Hrsg.) (2012). *MMS 2012 – Mobile und ubiquitäre Informationssysteme*. S. 13-26

ERFORDERNISSE

Aufwand	■ ■ □ □ ¹	Schwierigkeit	■ ■ □ □	Eignung mobil	+
Abhängigkeiten	Nutzungskontext-analyse (z. B. Kontextuelles Interview), Personas	Projektphasen	Analyse	Beteiligte Rollen	Software-Gestalter, Benutzer (/Kunde)

Kurzbeschreibung

Ein Erfordernis (engl. User Need) ist „[e]ine für einen Benutzer oder eine Benutzergruppe als notwendig identifizierte Voraussetzung, um ein Ziel innerhalb eines bestimmten Nutzungskontextes zu erreichen.“ [1] (S. 23)



Ziel

- Erkennen, welche die nötigen Voraussetzungen für die Zielerreichung innerhalb eines bestimmten Nutzungskontextes sind. vgl. [1] (S. 23)

Ressourcen

Werkzeuge	<ul style="list-style-type: none"> • Papier, Stift • Oder Textverarbeitungssoftware, PC 	Teilnehmer	<ul style="list-style-type: none"> • 1 Erfordernis-Ersteller • ggf. interne und/oder externe Personen zur Evaluation 	Zeit	• 1 PT ²
------------------	---	-------------------	--	-------------	---------------------

Vorgehen

„Da die Gebrauchstauglichkeit eines Produkts stets von den Erfordernissen des Nutzungskontexts abhängt, sind diese zunächst zu erfassen.“ ([2], S. 102) Daher wird bei während der Nutzungskontextanalyse (z. B. in Form eines

¹ Eine genauere Einschätzung des Zeitaufwands ist noch nicht bekannt. Es ist aber davon auszugehen, dass der zeitliche Aufwand nicht allzu hoch ausfallen wird.

² s. o.

kontextuellen Interviews) ermittelt, welche Voraussetzungen für bestimmte Nutzer oder Nutzergruppen (welche z. B. in Form von Personas vorab bereits näher definiert wurden) zur Erreichung bestimmter Ziele innerhalb des relevanten Nutzungskontexts notwendig sind.

„Beispiele:

- Ein Vortragender (Benutzer) muss wissen, wie viel Zeit er noch hat (**Voraussetzung**), um den Vortrag während einer Präsentation mit einer festgelegten Zeitbeschränkung (Nutzungskontext) abschließen zu können (Ziel).
- Ein Finanzbuchhalter (Benutzer) muss die Anzahl der erhaltenen Rechnungen und ihre Beträge kennen (**Voraussetzung**) um die tägliche Rechnungsstellung (Ziel) als Teil der Cashflow-Kontrolle (Nutzungskontext) fertigstellen zu können.“ [1] (S. 23)

Die Erfordernisse werden im weiteren Projektverlauf schließlich in Nutzungsanforderungen, die den Nutzungskontext, Benutzerprioritäten sowie das Wechselspiel mit anderen Anforderungen und -Einschränkungen mit berücksichtigen, transformiert. (vgl. [1], S. 23)

Hilfsmittel



Templates

- Checkliste Kontext-Profil



Tools

- Context Modeller

Das sollten Sie beachten

Fundierte Datengrundlage

„Erfordernisse werden aufgrund verschiedener Ansätze identifiziert, unter anderem durch Interviews mit Benutzern, Beobachtungen, Befragungen, Evaluierungen, Expertenreviews usw.“ ([1], S. 23)

Keine zweifelhaften Erfordernisse

„Es ist besser, kein Erfordernis einzutragen als eines, das bezweifelt werden kann. Ein zutreffendes Erfordernis erkennt man daran, dass es einleuchtend und selbstverständlich erscheint, also zweifelsfrei ist.“ ([2], S. 134)

Unabhängigkeit von Lösungsansätzen

„Ein Erfordernis ist unabhängig von jeglicher vorgeschlagenen Lösung für dieses Erfordernis.“ ([1], S. 23)

Erfordernisse bei großer, heterogener Nutzergruppe:

Erfordernisse können von Nutzergruppe zu Nutzergruppe unterschiedlich sein. Daher bleiben bei einer sehr heterogenen Nutzerschaft ggf. nur recht wenig generell gültige Erfordernisse übrig.

Quellen

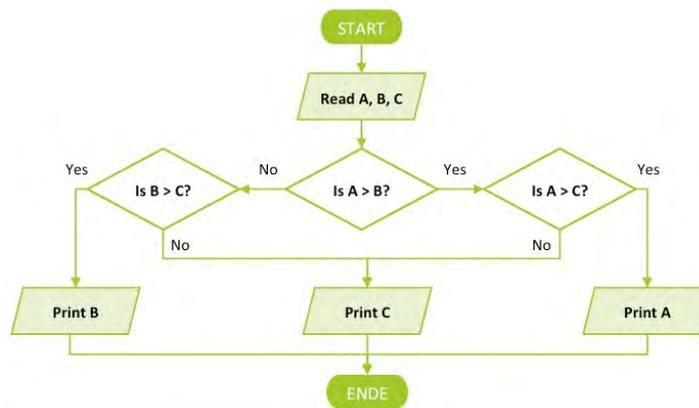
- [1] Arbeitskreis Qualitätsstandards der German UPA (2013). *CPUX F – Curriculum und Glossar*.
http://www.germanupa.de/data/mediapool/cpux-f__curriculum_und_glossar.pdf (Geprüft: 01.08.2014)
- [2] DakS – Deutsche Akkreditierungsstelle GmbH (2013). *Leitfaden Usability – Version 1.3*.
http://www.dakks.de/sites/default/files/71-SD-2-007_Leitfaden%20Usability%201.3.pdf (Geprüft: 01.08.2014)

FLOW ANALYSIS

Aufwand	■ ■ □ □	Schwierigkeit	■ ■ □ □	Eignung mobil	+ / -
Abhängigkeiten	Aufgabenanalyse (z. B. Kontextuelles Interview) oder Ideenfindung (z. B. Brainstorming)	Projektphasen	Analyse, Konzeption & Entwicklung	Beteiligte Rollen	Entwickler, Software-Gestalter, Benutzer (/Kunde)

Kurzbeschreibung

„Darstellung einzelner Schritte eines Ablaufes oder Informationsflusses innerhalb eines Prozesses oder Systems.“ [1]



Ziel

- Erkenntnisse über den Prozess-/ Informationsfluss innerhalb eines Systems erlangen und visualisieren (vgl. [1])
- oder Prozess-/ Informationsfluss innerhalb eines Systems modellieren und visualisieren (vgl. [3])

Ressourcen

Werkzeuge	<ul style="list-style-type: none"> • Stift, Papier/Kärtchen/ Whiteboard/Flip Chart etc. • Oder Grafikprogramm o. ä. (vgl. [2]) 	Teilnehmer	<ul style="list-style-type: none"> • 1 Diagramm-Ersteller • ggf. interne und/oder externe Personen zur Evaluation 	Zeit	• ? ¹
------------------	--	-------------------	---	-------------	------------------

Vorgehen

Zu Beginn der Flow Analysis wird festgelegt, wann und wo der zu analysierende Prozess anfängt und wann und wo er wieder aufhört. Außerdem wird definiert, wie detailgenau der Prozess schließlich dargestellt werden soll. (vgl. [1])

¹ Eine genauere Einschätzung des Zeitaufwands ist noch nicht bekannt. Es ist aber davon auszugehen, dass der zeitliche Aufwand nicht allzu hoch ausfallen wird.

Danach wird der Prozess visuell in Form eines Diagramms z. B. mithilfe von Kärtchen auf ein Whiteboard übertragen. Dazu werden zunächst alle Prozessbestandteile auf einzelnen Kärtchen notiert und danach in der Reihenfolge des Prozesses angeordnet. In sehr detaillierten Diagrammen werden die verschiedenen Arten von Prozessschritten durch spezielle Symbole dargestellt. (vgl. [1])

Das fertige Diagramm kann anschließend mit prozessbeteiligten Personen besprochen, überprüft und ggf. verbessert werden. Auch Problemstellen und Verbesserungsvorschläge können ermittelt werden. (vgl. [1])

Die Flow Analysis wird entweder dazu eingesetzt, Ergebnisse, die man während der Analyse z. B. von Arbeitsabläufen gewonnen hat, visuell darzustellen oder sie wird genutzt um Ideen bezüglich eines neuen oder verbesserten Systems zu modellieren und festzuhalten (vgl. [3]). Sie dient als Kommunikationsmittel zwischen verschiedenen Projekt-Beteiligten (z. B. Analysten, Entwicklern, Kunde etc.) [2].

Hilfsmittel



Templates

- Template Task Flow Sketchsheet
- Template Flow Analysis



Tools

- Workflow Modeller

Quellen

[1] Designmethodenfinder

<http://www.designmethodenfinder.de/flow-analysis?s1> (Geprüft: 01.08.2014)

[2] <http://asq.org/learn-about-quality/process-analysis-tools/overview/flowchart.html> (Geprüft: 01.08.2014)

[3] IDEO Method Cards

HEURISTISCHE EVALUATION (Usability Guidelines und Styleguides)

Aufwand	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> ¹	Schwierigkeit	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> (vgl. [2])	Eignung mobil	+/- ²
Abhängigkeiten	Lo-Fi/Hi-Fi Prototyping ³	Projektphasen	Analyse, Konzeption & Entwicklung, Umsetzung, Betrieb	Beteiligte Rollen	Software-Gestalter

Kurzbeschreibung

Analyse eines Softwareprodukts/eines Prototyps unter Berücksichtigung von Usability-Heuristiken und relevanten Styleguides. (vgl. [1], S. 71-79)



Ziel

- Überprüfung der Einhaltung von Usability-Regeln
- Überprüfung der Einhaltung verschiedener Styleguides, z. B. plattformspezifische Guides bei der Entwicklung mobiler Software

Ressourcen

Werkzeuge	<ul style="list-style-type: none"> • Eine oder mehrere Usability-Richtlinien und -Normen • Relevante Styleguides 	Teilnehmer	<ul style="list-style-type: none"> • 1 Evaluator 	Zeit	<ul style="list-style-type: none"> • 5 PT⁴
------------------	--	-------------------	---	-------------	--

¹ Eine genauere Einschätzung des Zeitaufwands ist noch nicht bekannt. Es ist zu beachten, dass auch die Suche und Auswahl von passenden Guidelines einen zeitlichen Aufwand bedeutet (1-2 PT). [1]

² Der spezielle Nutzungskontext mobiler Software wird mit diesem Methodenelement nicht berücksichtigt. Plattformspezifische Styleguides, welche besonders wichtig für die Erwartungskonformität mobiler Produkte sind, werden allerdings miteinbezogen.

³ Wenn noch keine fertige Software für die Analyse vorhanden ist, werden Prototypen benötigt.

⁴ s. Fußnote 1

Vorgehen

Bei der heuristischen Evaluation wird eine Software/ein Prototyp auf die Einhaltung verschiedener Usability-Richtlinien und -Regeln überprüft. Usability Guidelines beinhalten Regeln für die Gestaltung von Benutzeroberflächen. „Darunter fällt zunächst ein sehr breites Spektrum von Regelwerken. Die Palette reicht von globalen Grundsätzen bis hin zu detaillierten Vorgaben.“ ([1], S. 71-72) Es können dabei gesetzliche Verordnungen, Normen, Regelsammlungen, User Interface Patterns und Hersteller- oder plattformabhängige Styleguides unterschieden werden (vgl. [1], S. 72-74).

Folgende Richtlinien können bei einer Heuristischen Evaluation z. B. überprüft werden:

- DIN EN ISO 9241 ‚Ergonomie der Mensch-System-Interaktion‘ (vgl. [3])
- speziell: Grundsätze der Dialoggestaltung (vgl. [1], S. 72)
- DIN EN ISO 13407 ‚Benutzer-orientierte Gestaltung interaktiver Systeme‘ (vgl. [3])
- Bildschirmarbeitsverordnung (BildschArbV) und europäische Bildschirmrichtlinie EWG 90/270
- DAkks-Prüfhandbuch ‚Leitfaden-Usability‘ (vgl. [3])
- Usability-Heuristiken nach Nielsen & Molich (vgl. [3])

Ebenso sollte das Produkt auf die Einhaltung relevanter Styleguides überprüft werden. Das kann z. B. der Styleguide des Kunden sein, damit die entwickelte Anwendung an das Unternehmens-Corporate Design angepasst ist. Vor allem bei der Entwicklung mobiler Software ist aber ebenfalls die Berücksichtigung plattformspezifischer Styleguides wichtig, um ein erwartungskonformes Produkt erstellen zu können.

Hilfsmittel



Templates

- Android Styleguide
- iOS Styleguide
- WindowsPhone Styleguide
- 10 Usability Heuristiken nach Nielsen
- Dialogprinzipien nach ISO 9241-110
- Checkliste Heuristic Evaluation
- Usability-Checkliste für mobile
- Checkliste 7 Dialogprinzipien nach ISO 9241-110



Tools

- ---

Das sollten Sie beachten

Einheitliches und regelkonformes User Interface Design

„Usability Guidelines und Styleguides stellen in erster Linie Hilfsmittel für ein einheitliches und regelkonformes User Interface Design dar.“ ([1], S. 75)

Frühe Evaluation

Auch wenn man Heuristische Evaluation in allen Phasen der Produktentwicklung einsetzen kann, sollte man schon früh damit beginnen, Usability Guidelines und relevante Styleguides zu berücksichtigen. Je länger man damit wartet, desto aufwändiger wird die Behebung identifizierter Abweichungen.

Nicht automatisch eine brauchbare Lösung

„Selbst wenn Sie alle relevanten Regeln einhalten würden, könnten Sie noch immer eine für die Anwender unbrauchbare Lösung entwerfen. Mehr und mehr setzt sich die Erkenntnis durch, dass Zielgruppe und Nutzungskontext über die tatsächliche Qualität einer Benutzerschnittstelle entscheiden.“ ([1], S. 74) Eine Heuristische Evaluation sollte also auf keinen Fall den User Test ersetzen, sondern ihn eher ergänzen.

Heuristiken nicht blind einsetzen

Die Heuristiken, die zur Evaluation eingesetzt werden, sollten immer dem spezifischen Produkt angepasst sein (Regeln für Website sind z. B. für Desktop Software nicht immer passend etc.). „Die blinde Einhaltung von Vorgaben kann sonst schnell zu einem unnötigen „Klotz am Bein“ werden und gute Usability schlussendlich sogar verhindern.“ ([1], S. 75)

Quellen

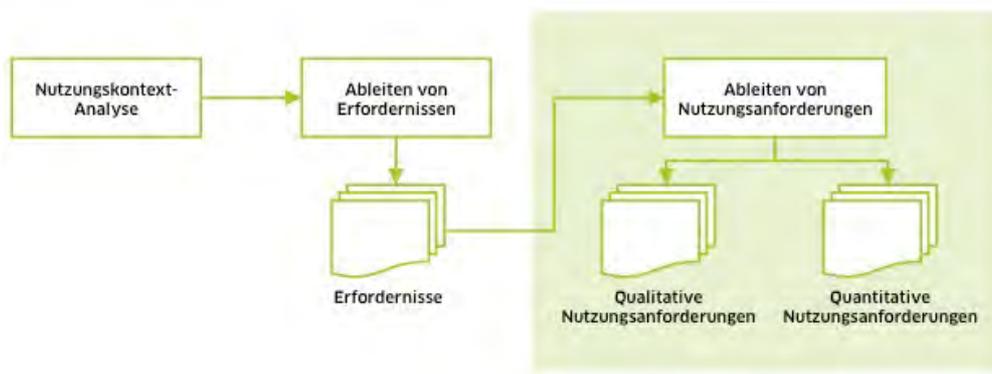
- [1] Richter, M. & Flückiger, M. D. (2013). *Usability Engineering kompakt – Benutzbare Software gezielt entwickeln*. 3. Auflage. Berlin: Springer Verlag
- [2] German UPA – Usability Quartett
- [3] http://www.eresult.de/leistungen/methoden_verfahren/expertenbasierte_evaluation.html (Geprüft 01.08.2014)

KONTEXTUELLES INTERVIEW

Aufwand	■ ■ □ □ ¹	Schwierigkeit	■ ■ ■ □	Eignung mobil	+
Abhängigkeiten	Erfordernisse	Projektphasen	Analyse	Beteiligte Rollen	Software-Gestalter, Benutzer (/Kunde)

Kurzbeschreibung

Nutzungsanforderungen beschreiben, was ein interaktives System aus Sicht des Benutzers ermöglichen soll. Sie können qualitativ oder quantitativ sein und werden von Erfordernissen abgeleitet. (vgl. [1], S. 31)



Ziel

- Erkennen, welchen Anforderungen das zu entwickelnde Produkt gerecht werden muss (vgl. [1], S. 31)
- Grundlage für Produkt-Evaluation schaffen (vgl. [1], S. 31)

Ressourcen

Werkzeuge	<ul style="list-style-type: none"> • Stift, Papier • oder Textverarbeitungssoftware, PC 	Teilnehmer	<ul style="list-style-type: none"> • 1 Anforderungs-Erheber • ggf. interne und/oder externe Personen zur Evaluation 	Zeit	• 3,5 PT ²
------------------	---	-------------------	---	-------------	-----------------------

Vorgehen

Zuvor aufgestellte „Erfordernisse werden [...] in Nutzungsanforderungen, die den Nutzungskontext, Benutzerprioritäten sowie das Wechselspiel mit anderen Anforderungen und -Einschränkungen mit berücksichtigen“ ([1], S. 23), transformiert. Dabei wird zwischen qualitativen und quantitativen Nutzungsanforderungen unterschieden. (vgl. [1], S. 31)

¹ Eine genauere Einschätzung des Zeitaufwands ist noch nicht bekannt. Es ist aber davon auszugehen, dass der zeitliche Aufwand nicht allzu hoch ausfallen wird.

² s. o.

Qualitative Nutzungsanforderungen

Eine qualitative Nutzungsanforderung ist „[e]ine Beschreibung, was Benutzer während der Durchführung einer Aufgabe mit dem interaktiven System erkennen, auswählen oder eingeben müssen [...] Qualitative Nutzungsanforderungen sind keine Funktionen, sondern stellen die Basis für Funktionen dar.“ ([1], S. 31)

„Beispiele für sinnvolle qualitative Nutzungsanforderungen:

- »Der Benutzer muss die unterschiedlichen Fahrzeuge, die für einen bestimmten Preisbereich auf der Autovermietungswebseite verfügbar sind, vergleichen können. «
- »Der Benutzer muss ein Fahrzeug mit automatischer Übertragung auf der Autovermietungswebseite auswählen können. «
- »Der Benutzer muss die Öffnungszeiten einer bestimmten Autovermietungsstelle sehen können. «“ ([1], S. 31)

„Beispiele falsch formulierter Nutzungsanforderungen:

- »Die Benutzungsschnittstelle muss gebrauchstauglich sein und alle Benutzeraufgaben unterstützen. « (zu allgemein)
- »Die Benutzungsschnittstelle muss einen großen roten Button „Auto mieten“ haben. « (zu detailliert)“ ([1], S. 31)

Quantitative Nutzungsanforderungen

Eine quantitative Nutzungsanforderung ist die „[g]eforderte Leistung, die die Basis für Design und die Evaluierung eines interaktiven Systems darstellt, mit dem Ziel, identifizierte Erfordernisse zu befriedigen [...] Quantitative Nutzungsanforderungen stellen Akzeptanzkriterien für die Implementierung einer Benutzungsschnittstelle dar [...] [sie] erlauben, ein Design zu evaluieren, mit dem Ziel festzustellen, ob Nutzungsanforderungen effizient erfüllt wurden.“ ([1], S. 31)

„Beispiel:

- »80% der Benutzer, die die Autovermietungswebseite zumindest zweimal benutzt haben, müssen in der Lage sein, innerhalb von 5 Minuten einen Kleinwagen ab Flughafen Frankfurt (Deutschland) für zwei Tage beginnend ab morgen 09:00 Uhr zu mieten.«“ ([1], S. 31)

Sind die Nutzungsanforderungen aufgestellt, können sie im weiteren Projektverlauf dazu herangezogen werden, um das entstehende Produkt jederzeit darauf zu überprüfen, ob es den beabsichtigten Anforderungen gerecht wird.

Hilfsmittel



Literatur-Empfehlung

- <http://de.slideshare.net/guest24d72f/8-characteristics-of-good-user-requirements-presentation>

Quellen

- [1] Arbeitskreis Qualitätsstandards der German UPA (2013). *CPUX F – Curriculum und Glossar*.
http://www.germanupa.de/data/mediapool/cpux-f__curriculum_und_glossar.pdf (Geprüft: 04.08.2014)

KONTEXTUELLES INTERVIEW

Aufwand	■ ■ ■ □ ¹	Schwierigkeit	■ ■ ■ □	Eignung mobil	+
Abhängigkeiten	---	Projektphasen	Analyse	Beteiligte Rollen	Software-Gestalter, Benutzer

Kurzbeschreibung

Das Kontextuelle Interview (engl. Contextual Inquiry) ist eine „Erhebung im Umfeld der Benutzer“. Die Bedürfnisse der Anwender werden untersucht, indem Letztere bei ihren Tätigkeiten beobachtet und befragt werden. (vgl. [1], S. 30-38)



Ziel

- Wissen über den Nutzer/Nutzungskontext einer Software etc. erlangen
- Ableiten von Anforderungen an die Software, die sich aus dem konkreten Nutzungskontext ergeben

Ressourcen

Werkzeuge	<ul style="list-style-type: none"> • Stift, Papier • Ggf. Fotokamera 	Teilnehmer	<ul style="list-style-type: none"> • min. 1 Observator/Interviewer, besser: 2 verschiedene Personen • 1 Testperson pro Benutzergruppe 	Zeit	• 8 PT ²
------------------	--	-------------------	---	-------------	---------------------

¹ Der Zeitaufwand ist sehr schwer einzuschätzen. Er hängt stark von der Komplexität des Projekts und der Anzahl verschiedener Benutzergruppen ab. Mindestens ist mit 3 Personentagen zu rechnen, bei umfangreicheren Analysen kann der Aufwand aber auch bis zu 12 PTs betragen [vgl. 1].

Nimmt man eine gründliche Kontextanalyse zu Beginn eines Projekts vor, kann diese außerdem aufwändiger sein als spätere – vermutlich eher ergänzende – weitere Analysen.

² Wenn noch keine fertige Software für das Testing vorhanden ist, werden Prototypen für das Testing benötigt.

Vorgehen

„Mit [...] [dem Kontextuellen Interview] sollen ausgewählte Fragestellungen beantwortet werden. [1] (S. 31) Bevor ein Analyst mit Benutzern spricht, stellt sich das Projektteam die Frage, was in Bezug auf das neue System wichtig zu wissen ist. „Lohnenswerte Fragen zielen auf den Einsatz und das Umfeld von heute benutzten Produkten ab [...] [Das Kontextuelle Interview] fokussiert auf die Tätigkeiten der späteren Benutzer und das Umfeld der Anwendung [...] Die Auswahl der Interviewpartner muss nicht repräsentativ im Sinne der Statistik sein, doch sie sollte ein breites Spektrum an Meinungen und Bedürfnissen abdecken. Es lohnt sich, auf eine gewisse Streuung bezüglich Alter, Geschlecht, Position, Arbeitsort, Erfahrung, Fachwissen, kulturellem Hintergrund und mehr zu achten.“ ([1], S. 31-33)

„Die Untersuchung findet vor Ort und während der Arbeit statt. Der Analyst beobachtet den Interviewpartner und stellt gezielt Fragen über das Beobachtete. Der Interviewpartner soll die eigene Handlungsweise reflektieren und so angewandtes Expertenwissen aufdecken [...] Der Analyst und der Interviewpartner diskutieren, ausgehend von einer gerade vorgeführten Arbeitstätigkeit, über Probleme, fachliche Zusammenhänge und Verbesserungsmöglichkeiten [...] Bei neuartigen Produktentwicklungen sollten erste Entwürfe, Prototypen oder existierende vergleichbare Produkte in die Interviews einbezogen werden. Je näher die Erhebungssituation an die Realität der geplanten Anwendung heranreicht und je lebensechter der Kontext dieser Anwendung vermittelt werden kann, desto wertvoller sind die Rückmeldungen der Benutzer.“ ([1], S. 34)

„In kaum einem Umfeld ist der gesamte Nutzungskontext in nur einer Interviewrunde erfassbar. Die Erfahrung zeigt, dass die Fragestellungen zu Beginn breit und unscharf sind. Im Verlauf des Projektes kennt das Analyseteam den Kontext immer besser, und die Fragestellung wird konkreter und enger gefasst. Es lohnt sich deshalb, mehrere Iterationen durchzuführen. Die Resultate befruchten jeweils die Fragestellung der nächsten Iteration.“ ([1], S. 32)

Die einzelnen Schritte:

1. Vorbereitung/Planung

- Festlegen, welche Fragestellungen beantwortet werden sollen
- Interview-Leitfaden entwickeln
- Probanden akquirieren

2. Durchführung des Tests

- Probanden instruieren/informieren
- Interview starten
- Beobachtungen notieren
- Ggf. Fotos machen

3. Nachbereitung

- Auswertung der Notizen/Testergebnisse

Hilfsmittel



Templates

- Fragestellungen für das Kontextuelle Interview
- Beispiel-Leitfaden Contextual Inquiry
- Checkliste Kontext-Profil



Tools

- ---

Literatur-Empfehlung

- <http://www.uxmatters.com/mt/archives/2012/06/why-are-contextual-inquiries-so-difficult.php>
-

Das sollten Sie beachten

Im realen Kontext befragen

Die Befragung sollte unbedingt im realen Nutzungskontext erfolgen, da „[...] Menschen viel Wissen, das sie in einer bestimmten Situation anwenden, nicht einfach abrufen können [...] Die Kombination von Beobachtung und Befragung ermöglicht daher einerseits, das wirkliche Geschehen im Detail zu erfassen, und andererseits, die Gründe und Zusammenhänge dahinter zu durchleuchten. Diese Informationen sind wertvoll, um den notwendigen Informationsgehalt, die Benutzerführung und die Funktionalität des geplanten Systems abzuleiten.“ ([1], S. 31)

Früh gründlich analysieren:

Analysiert man den Nutzungskontext bereits früh im Projekt gründlich, hat man eine gute Grundlage für erste Konzepte geschaffen. Später kann man die Ergebnisse bei Bedarf mit kleineren Analysen ergänzen.

Transparenz

„Niemand wird gerne bei der Arbeit beobachtet, wenn nicht klar ist, wozu die Analyse dient. Nur eine transparente, offene, interessierte und partnerschaftliche Haltung führt zum Ziel.“ ([1], S. 37) Wenn Sie Fotos machen möchten, fragen Sie vorher unbedingt um Erlaubnis.

Fragestellung im Auge behalten

„Behalten Sie während der Analyse die ausgewählte Fragestellung im Auge. So manches Projektteam hat sich schon in der Unzahl der Informationen in der Analyse verloren.“ ([1], S. 37)

Festhalten, worauf Erkenntnisse zurückzuführen sind

„Halten Sie auch fest, worauf Ihre Erkenntnisse zurückzuführen sind. Irgendwann kommt ein Auftraggeber und will wissen, weshalb ein bestimmtes Feature eingebaut wurde.“ ([1], S. 38)

Quellen

- [1] Richter, M. & Flückiger, M. D. (2013). *Usability Engineering kompakt – Benutzbare Software gezielt entwickeln*. 3. Auflage. Berlin: Springer Verlag

PERSONAS

Aufwand


1

Schwierigkeit

Eignung mobil

Abhängigkeiten

Analyse der Nutzer
(z. B. Kontextuelles
Interview)

Projektphasen

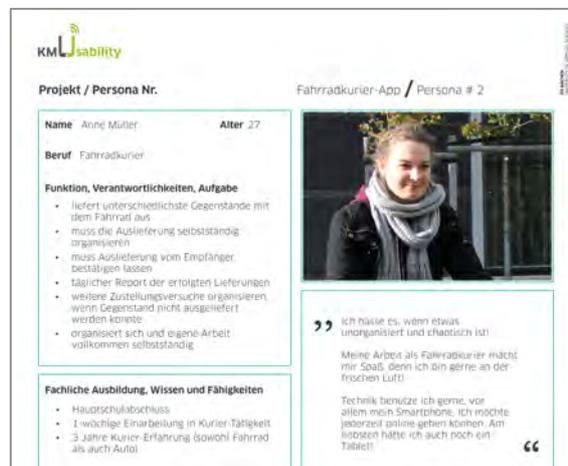
Analyse, Betrieb
(vgl. [3])

Beteiligte Rollen

Software-
Gestalter,
Benutzer
(/Kunde)

Kurzbeschreibung

„Personas beschreiben die wichtigsten Benutzer-Prototypen in ihren Charaktereigenschaften und Verhaltensweisen mit dem interaktiven System.“ [4]



Ziel

- Erhalten eines Einblicks in die potentielle Nutzergruppe bzw. Verständnis für die Nutzer (vgl. [2])
- Hineinversetzen in die Benutzerperspektive um user-centered entwickeln zu können (vgl. [2])

Ressourcen

Werkzeuge

- Stift, Papier
- oder digitale Aufbereitung und Gestaltung
- Bildmaterial zur Visualisierung

Teilnehmer

- 1 Persona-Ersteller
- weitere Beteiligte zur Validierung: z. B. Auftraggeber, Teammitglieder (Entwickler etc.) (vgl. [1], S. 47)

Zeit

- min. 2 PT² (vgl. [1], S. 47)

¹ Der Zeitaufwand kann nicht genau geschätzt werden. Je nach Produktkomplexität und Anzahl Personas, die erstellt werden muss, sind auch bis zu 6 Personentage denkbar. (vgl. [1], S. 47)

² s. o.

Vorgehen

„Personas stellen prototypische Benutzer dar und verkörpern ihre unterschiedlichen Ziele, Verhaltensweisen und Eigenschaften, die im Hinblick auf das zu entwickelnde Produkt relevant sind [...] Personas werden aufgrund von Informationen über die zukünftigen Benutzer eines Systems erarbeitet. Dazu dienen beispielsweise Ergebnisse aus Workshops mit Benutzern, [...] [Kontextuellen Interviews], Fragebögen oder Usability Walkthroughs mit bestehenden Systemen. Der Analyst entwirft Vorschläge und validiert diese mit den Beteiligten, oder die Personas werden in einem gemeinsamen Workshop erstellt. Eine Persona sollte schließlich die für das Produktdesign relevanten Eigenschaften der Benutzer widerspiegeln.“ ([1], S. 39)

„Eine Persona kann über folgende Eigenschaften Auskunft geben:

- Ziele der Benutzer,
- Beruf, Funktion, Verantwortlichkeiten und Aufgaben,
- Fachliche Ausbildung, Wissen und Fähigkeiten,
- Verhaltensmuster und Vorgehensweisen,
- Werte, Ängste, Sehnsüchte, Vorlieben,
- Allgemeine Computerkenntnisse,
- Kenntnisse über verwandte Produkte, Vorgängersysteme, Konkurrenzprodukte,
- Verbesserungspotenzial in der heutigen Situation,
- Erwartungen an eine neue Lösung.“ ([1], S. 40-41)

„Der erstellte Charakter soll einprägsam sein. Seine Eigenschaften sollen einfach verinnerlicht werden können. Dazu kann eine Persona mit zusätzlichen passenden Informationen zum Leben erweckt werden:

- Name, Alter, Geschlecht,
- Markige Charakterzüge,
- Bild, Skizze, Porträt,
- Passende Zitate aus Interviews,
- Ein Tag im Leben von ...

Insbesondere die weichen Kriterien wie Ziele, Werte und Ängste erscheinen einem Ingenieur im ersten Moment zwecklos. Die Handlungsweise von Menschen wird indessen stark von genau solchen Faktoren beeinflusst.“ ([1], S. 41)

Hilfsmittel



Templates • ---



Tools • ---

Literatur-Empfehlung

- <http://de.slideshare.net/guest24d72f/8-characteristics-of-good-user-requirements-presentation>
-

Das sollten Sie beachten

Fundierte Datengrundlage:

„Wenn immer möglich sollten Personas aufgrund von Erkenntnissen über die (zukünftigen) Benutzer abgeleitet werden, beispielsweise aus Ergebnissen von Interviews, Kontextuellen Interviews, Beobachtungen oder Befragungen. Es besteht sonst die Gefahr, dass aufgrund falscher Vorstellungen an der Zielgruppe vorbeientwickelt wird.“ ([1], S. 46)

Personas \neq Zielgruppe:

„Personas sind keine quantitativen Zielgruppenbeschreibungen. Personas verkörpern die für die Entwicklung relevanten Aspekte der Benutzer, während sich Zielgruppen auf die für Marketing und Verkauf wesentlichen Aspekte konzentrieren.“ ([1], S. 46)

Personas \neq Marktsegmente:

„Personas sind keine Marktsegmente. Segmente teilen die potenziellen Käufer entsprechend ihrer Eigenschaften in verschiedene Bereiche ein (z. B. alle Kunden im Alter von 18–35). Die Eigenschaften von Personas dagegen weisen keine Bereiche auf und erfüllen einen anderen Zweck. Sie spiegeln Bedürfnisse einer Benutzergruppe wider und fokussieren auf die Interaktion mit dem zukünftigen Produkt.“ ([1], S. 46)

Quellen

- [1] Richter, M. & Flückiger, M. D. (2013). *Usability Engineering kompakt – Benutzbare Software gezielt entwickeln*. 3. Auflage. Berlin: Springer Verlag
- [2] Designmethodenfinder
<http://www.designmethodenfinder.de/personas?s1> (Geprüft: 04.08.2014)
- [3] eResult – Methoden und Verfahren
http://www.eresult.de/leistungen/methoden_verfahren/personas.html (Geprüft: 11.2013)
- [4] German UPA – Usability Quartett

STORYBOARDING

Aufwand


1

Schwierigkeit

Eignung mobil


2

Abhängigkeiten

Nutzungskontext-
analyse (z. B.
Kontextuelles
Interview), Szenarios

Projektphasen

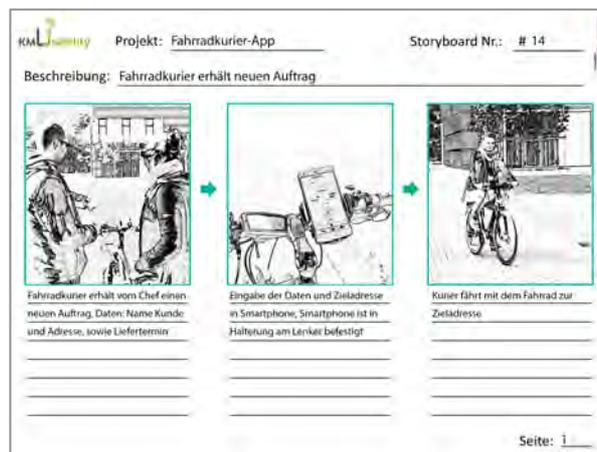
Analyse, Konzeption
& Entwicklung,
Betrieb

Beteiligte Rollen

Software-
Gestalter,
Grafikdesigner

Kurzbeschreibung

„Unterstützung eines Konzeptentwurfs, durch skizzenhafte grafische Darstellung und Beschreibung.“ [3]



Ziel

- Eine Idee oder ein Konzept visuell vereinfacht darstellen und vermitteln (vgl. [3])
- Nutzungskontext visualisieren und vermitteln
- Software-Einführung und Ausbildung unterstützen (vgl. [1], S. 51)

Ressourcen

Werkzeuge

- Papier, Stift (vgl. [3])

Teilnehmer

- 1 Storyboarder (vgl. [1], S. 52)
- weitere Beteiligte zur Validierung: z. B. Auftraggeber, Teammitglieder (Entwickler etc.) (vgl. [1], S. 52)

Zeit

- min. 1 PT pro Storyboard (vgl. [1], S. 52)

¹ Der Zeitaufwand kann nicht genau geschätzt werden. Hängt von Produktkomplexität und Anzahl Storyboards, die erstellt werden muss, ab.

² Einige Kontextfaktoren (z. B. Geräusche) sind schwierig zu zeichnen, können aber beispielsweise durch Symbole im Storyboard dargestellt werden.

Vorgehen

„Ein Storyboard zeigt mithilfe der Benutzerschnittstelle, wie ein System oder Produkt verwendet wird. Es stellt wichtige Aspekte der Anwendung bildlich dar und dient damit der Kommunikation zwischen allen Beteiligten. Im Wesentlichen handelt es sich dabei um die Visualisierung eines Szenarios.“ ([1], S. 48)

„Abhängig vom Kommunikationszweck kann ein Storyboard in unterschiedlichen Ausprägungen erstellt werden. Die Palette reicht von skizzenartigen oder realistisch gestalteten Abfolgen der Benutzerschnittstelle (User Interface Storyboard) bis zu Bildergeschichten, die auch Kontext und handelnde Personen darstellen.“ ([1], S. 48) Letzteres ist besonders für die Entwicklung mobiler Software von Bedeutung, da der oft komplexe Nutzungskontext so miteinbezogen werden kann.

„Der Analyst setzt Storyboards in Situationen ein, wo Text alleine nicht ausreicht. Für eine solche Visualisierung spricht:

- In Bildern können Aspekte vermittelt werden, die mit Text nicht oder nur schwer auszudrücken sind, beispielsweise neuartige Konzepte, für die es noch keine Begriffe gibt.
- Mit der visuellen Umsetzung können Erlebnisse, die für die Anwendung von Bedeutung sind, besser in die Welt des Zielpublikums transportiert werden.“ ([1], S. 48-49)

„Ein Storyboard eignet sich deshalb, um folgende Gesichtspunkte aufzuzeigen:

- Dialogabläufe der Benutzerschnittstelle,
- Schwer verständliche Konzepte oder Sachverhalte,
- Wichtige Aspekte des Anwendungskontexts,
- Spezielle oder komplexe Umgebungen, in denen das System eingesetzt wird.“ ([1], S. 48)

„Storyboards können in verschiedenen Situationen und für unterschiedliche Zwecke eingesetzt werden:

- Zur Diskussion einer Idee oder einer ausgearbeiteten Lösung mit Benutzern und weiteren Stakeholdern.
- Um das korrekte Verständnis der Bedürfnisse und der fachlichen Zusammenhänge zu prüfen und Missverständnisse auszuräumen.
- Zur Diskussion von Vor- und Nachteilen verschiedener Varianten.
- Um über Neuerungen zu informieren und damit beispielsweise Akzeptanz für das neue Werkzeug zu erzeugen.
- Um neugierig auf das Neue zu machen.
- Um Führungskräfte darüber zu informieren, wie ihre Vision durch die neue Lösung Realität wird.
- Um Entwicklern die relevanten Anforderungen der Benutzung näherzubringen und zu zeigen, warum gewisse Entscheidungen getroffen wurden.
- Um Benutzern im Rahmen einer Ausbildung einen Überblick über das System zu geben.
- Für Projektmarketing bei Auftraggebern, Geschäftsleitung und Benutzern.“ ([1], S. 50-51)

Hilfsmittel



Templates

- Template Storyboarding 1 interaktiv
- Template Storyboarding 3 interaktiv



Tools

- s. Toolliste Storyboarding

- Template Storyboarding 6
interaktiv

Das sollten Sie beachten

Auch „unsichtbare“ Faktoren berücksichtigen:

Der Nutzungskontext mobiler Software kann unglaublich komplex sein. Daher sollte er nach Möglichkeit in die Storyboards miteinbezogen werden, damit er visuell festgehalten und somit nicht so schnell aus den Augen verloren wird. Einige kontextuellen Aspekte sind allerdings nur schwer bildlich festzuhalten (z. B. Geräusche). Da aber gerade sie bei der Entwicklung schnell in Vergessenheit geraten können, sollten sie z. B. durch den Einsatz geeigneter Symbole (z. B. Musiknoten) in die Storyboards integriert werden. (vgl. [2], S. 5-6)

Quellen

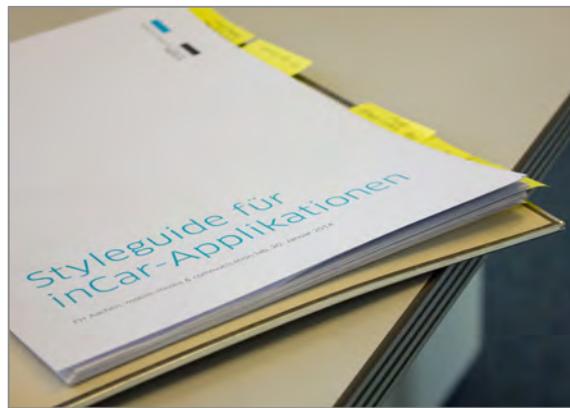
- [1] Richter, M. & Flückiger, M. D. (2013). *Usability Engineering kompakt – Benutzbare Software gezielt entwickeln*. 3. Auflage. Berlin: Springer Verlag
- [2] Keller, C., Kühn, R. & Schlegel, T. (2011). *Von modellbasierten Storyboards zu kontextsensitiven Interaction-Cases*.
http://tu-dresden.de/die_tu_dresden/fakultaeten/fakultaet_informatik/smt/seus/dateien/publikationen_dateien/i-com_2011 (Geprüft: 11.2013)
- [3] Designmethodenfinder
<http://www.designmethodenfinder.de/storyboarding?s1> (Geprüft: 04.08.2014)

STYLEGUIDES

Aufwand	■ ■ ■ □ ¹	Schwierigkeit	■ ■ ■ □	Eignung mobil	+ ²
Abhängigkeiten	---	Projektphasen	Konzeption & Entwicklung	Beteiligte Rollen	Software-Gestalter, Grafikdesigner

Kurzbeschreibung

Styleguides enthalten „konkrete Vorgaben für die visuelle Gestaltung und das Layout einer bestimmten Benutzeroberfläche. [...] [Sie] beschreiben Aussehen und Verhalten (Look&Feel) von User-Interface-Elementen, abhängig von der eingesetzten Technologie.“ ([1], S. 72)



Ziel

- Eine einheitliche und regelkonforme Gestaltung eines Interface festlegen und/oder gewährleisten (vgl. [1], S. 75)
- Kommunikationsmittel zwischen Designern, Entwicklern und anderen an der Softwareentwicklung beteiligten Personen (vgl. [1], S. 78)
- Vor allem im Bereich der mobilen Software ein erwartungskonformes, plattformspezifisches Design entwickeln und gewährleisten

Ressourcen

Werkzeuge	• Grafiksoftware, PC • oder bereits vorhandene Styleguides	Teilnehmer	• 1 Designer	Zeit	• ? ³
------------------	---	-------------------	--------------	-------------	------------------

¹ Eine genauere Einschätzung des Zeitaufwands ist nicht möglich, da er davon abhängig ist, in welcher Form Styleguides verwendet werden. Soll ein eigener Styleguide für das zu entwickelnde Produkt erarbeitet werden, ist von mindestens 10 PT auszugehen. Werden während der Produktentwicklung Styleguides Dritter einbezogen um z. B. ein erwartungskonformes Design zu gewährleisten, ist der Aufwand eher geringer. (vgl. [1], S. 79).

² Vor allem im Bereich der mobilen Software ist es unabdingbar plattformspezifische Styleguides einzubeziehen, um ein erwartungskonformes Design entwickeln und gewährleisten zu können.

³ s. Fußnote 1

Vorgehen

In der Softwareentwicklung können Styleguides auf zwei verschiedene Arten genutzt werden (vgl. [1], S. 71-79):

Zum einen können Sie dazu verwendet werden, um Gestaltungs- und Interaktionsmuster und andere relevante Vorgaben festzuhalten. Dies hilft dabei, eine konsistente Software zu entwickeln. Dafür wird für das zu entwickelnde Produkt ein eigener Styleguide erarbeitet, welcher z. B. die Verwendung bestimmter Steuerelemente und Farben festhält.

Zum anderen können aber auch bereits vorhandene Styleguides Dritter (z. B. Unternehmens-Styleguide) zur Entwicklung und Prüfung eines Produkts heran gezogen werden.

Eigenen Styleguide entwickeln

Folgende Informationen sollte ein Styleguide enthalten:

- *„Technologische Rahmenbedingungen und Zielgruppe:* Auf welche Systeme bezieht sich der Styleguide? Wer ist der Empfänger des Regelwerks?
- *Software-Ergonomie:* Allgemeingültige Regeln, die im konkreten Fall zu berücksichtigen sind (z. B. Anzahl Menü-Einträge) sowie auch Regeln bezüglich der spezifischen Zielgruppe und Anwendung (z. B. vollständig über die Tastatur bedienbare Dialoge).
- *Grundsätzlicher Aufbau:* Vorgaben für den Aufbau einer Applikation, z. B. Titelleiste, Ribbons, Navigation-, Arbeits-, Hilfe- und Statusbereich, sowie Dialogtypen.
- *Anzeigegeräte und Anordnung:* Definition, wie der Aufbau abhängig von Bildschirmgröße und Bildschirmauflösung zu gestalten ist.
- *Eingabemedien:* Welche Eingabemedien wie Tastatur, Touch, Stift, Gesten und Maus verwendet werden und für welchen Zweck.
- *Anwendungsregeln:* Welche GUI-Elemente für welche Situation verwendet werden. Dies gilt sowohl für die Basis-Elemente des Betriebssystems (z. B. wann werden Radio Buttons, wann Listboxes eingesetzt) als auch für zusammengesetzte Elemente (z. B. sortierbare Tabellen, Wizards) und insbesondere für neu definierte Elemente (z. B. Kalender-Pop-up zur Datumsauswahl).
- *Verhalten der GUI-Elemente:* Beschreibung der Reaktion des Systems (z. B. Selektion eines Eintrags, Deaktivierung von Controls).
- *Navigation:* Beschreibung der Navigationselemente (z. B. Einsatz von Menus, Links, Buttons).
- *Visuelles Design:* Farbschema, Kontraste, Schriften, Layout, Abstände, Icons etc. Hier werden auch Corporate-Design-Aspekte referenziert.
- *Technische Umsetzbarkeit:* Hinweise auf die technische Umsetzung, z. B. Referenz auf verfügbare GUI-Komponenten.
- *Terminologie:* Begriffe und Bezeichnungen der Benutzeroberfläche, wie wird der Benutzer angesprochen, welche Fachbegriffe werden verwendet, Formulierung von Fehlermeldungen.
- *Tastaturbedienung:* Shortcuts, Kommandos, Tabulator-Reihenfolge, Default Buttons in einem Dialog.“ ([1], S. 77-78)

Nutzung Styleguides Dritter

Bei der Entwicklung mobiler Software wird angeraten unbedingt plattformspezifische Styleguides bzw. die Vorgaben in diesen bei der Entwicklung eines eigenen Styleguides zu berücksichtigen. Nur so können erwartungskonforme Anwendungen entstehen.

Daneben kann aber auch z. B. vom Kunden verlangt werden, dass der unternehmenseigene Styleguide oder der Styleguide für eine bestimmte Produktlinie bei der Softwareentwicklung einbezogen werden sollen.

Hilfsmittel



Templates

- Styleguide Android
- Styleguide iOS
- Styleguide WindowsPhone
- Checkliste für den Einsatz von Styleguides



Tools

- ---

Quellen

- [1] Richter, M. & Flückiger, M. D. (2013). *Usability Engineering kompakt – Benutzbare Software gezielt entwickeln*. 3. Auflage. Berlin: Springer Verlag

SZENARIOS

Aufwand	■ ■ □ □ ¹	Schwierigkeit	■ ■ ■ □	Eignung mobil	+
Abhängigkeiten	Personas, Anforderungserhebung	Projektphasen	Analyse, Konzeption & Entwicklung, Umsetzung, Betrieb	Beteiligte Rollen	Software-Gestalter, Entwickler

Kurzbeschreibung

Szenarien stellen mittels realistischer Beispiele dar, wie ein Nutzer mit dem geplanten System interagiert wird. Dabei wird aus Benutzersicht ein konkreter Ablauf unter Berücksichtigung des Nutzungskontexts beschrieben. (vgl. [1], S. 42)



Ziel

- Durch die Darstellung des Nutzungskontexts z. B. einer Software auf Anforderungen, die sich aus dem Kontext ergeben, aufmerksam werden
- Unter Berücksichtigung der ermittelten Nutzungskontext-Anforderungen die Benutzbar- und Umsetzbarkeit der Software überprüfen (vgl. [2])

Ressourcen

Werkzeuge	• Papier, Stift • oder Textverarbeitungssoftware, PC	Teilnehmer	• 1 Szenario-Ersteller	Zeit	• min. 3 PT ² (vgl. [1], S. 47)
------------------	---	-------------------	------------------------	-------------	---

Vorgehen

Als Basis für Szenarios dienen zuvor erstellte Personas der (potentiellen) Zielgruppe (vgl. [2]). So können verschiedene Szenarios für verschiedene Benutzergruppen entwickelt werden. Szenarios entstehen auf der Basis der Anforderungen an das neue System. (vgl. [1], S. 43)

¹ Der Zeitaufwand kann nicht genau geschätzt werden. Je nach Produktkomplexität und Anzahl Szenarios, die erstellt werden muss, sind auch bis zu 10 Personentage denkbar. (vgl. [1], S. 47)

² s. o.

Bei der Erstellung eines Szenarios wird also ein konkreter, für eine Benutzergruppe realistischer Anwendungsfall einer Software etc. Schritt für Schritt formuliert. In leicht verständlicher Sprache werden alle Handlungsschritte des Benutzers aus dessen Sicht und unter Einbezug des Benutzungskontextes notiert. (vgl. [1], S. 42)

Durch das Erstellen möglicher Szenarios für eine Software werden die Anforderungen an die Software modelliert (vgl. [1], S. 43).

„Folgende Eigenschaften zeichnen ein Szenario aus:

- Es wird für eine bestimmte Benutzergruppe entworfen, berücksichtigt ihre Eigenschaften und erfüllt ihre Bedürfnisse.
- Es stellt einen konkreten Fall aus der Anwendung dar.
- Es zeigt, wie die Benutzer die neue Software in ihrem realen Umfeld einsetzen werden.
- Es illustriert die für die Entwicklung der neuen Lösung relevanten Aspekte.
- Es beschränkt sich nicht auf den Schönwetterfall, sondern beschreibt auch exemplarisch wichtige Ausnahme- und Fehlersituationen.“ ([1], S. 43)

„Szenarien können zu unterschiedlichen Zeitpunkten in der Entwicklung einer neuen Lösung und für verschiedene Ziele eingesetzt werden:

- *Erhebung und Validierung von Anforderungen:* Die Reflektion am konkreten Beispiel erlaubt es Auftraggebern und Benutzern, Anforderungen in der konkreten Anwendungssituation zu vergegenwärtigen, zu überprüfen und zu ergänzen. Szenarien können als erste Prototypen eines neuen Systems betrachtet werden.
- *Spezifikation:* Szenarien illustrieren die Anwendung im realen Kontext und dienen als Ergänzung des Use-Case-Modells. Sie vermitteln den Entwicklern ein Verständnis der Abläufe und Zusammenhänge. In agilen Projekten sind Szenarien eine gute Quelle für die Erstellung von User Stories. Szenarien bringen die konkrete Anwendung in die Diskussion ein.
- *User-Interface-Konzept:* Design-Szenarien dienen dazu, die Abläufe der Benutzerschnittstelle zu beschreiben. Damit kann die Interaktion modelliert und mit Benutzern optimiert werden. Die technischen Anforderungen können von Entwicklern überprüft werden.
- *Usability-Testszenarien:* Szenarien dienen als Basis für die Evaluation eines Systems oder eines Prototypen zusammen mit Benutzern.
- *Testszenarien:* Aus den erstellten Szenarien können Testszenarien für die Prüfung der entwickelten Software abgeleitet werden.
- *Schulung:* Szenarien dienen zur Schulung von Benutzern und als Basis für die Erstellung von Anleitungen.“ ([1], S. 43-44)

Hilfsmittel



Templates

- Template Szenario-Storyboard



Tools

- ---
-

Quellen

- [1] Richter, M. & Flückiger, M. D. (2013). *Usability Engineering kompakt – Benutzbare Software gezielt entwickeln. 3. Auflage.* Berlin: Springer Verlag
- [2] Designmethodenfinder
<http://www.designmethodenfinder.de/scenarios?s1> (Geprüft: 04.08.2014)

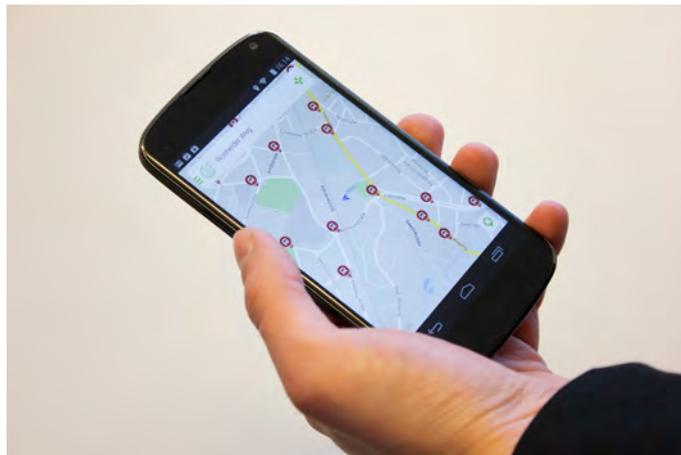
USER INTERFACE PROTOTYPING, HI-FI

Aufwand	■ ■ ■ □ ¹	Schwierigkeit	■ ■ ■ □ ²	Eignung mobil	+/-
Abhängigkeiten	Lo-Fi Prototyping	Projektphasen	Konzeption & Entwicklung	Beteiligte Rollen	Software-Gestalter, Grafikdesigner

Kurzbeschreibung

„User Interface Prototyping (UI Prototyping) wird im Usability Engineering eingesetzt, um Aspekte der Benutzerschnittstelle zu entwerfen, zu evaluieren und zu verbessern, noch bevor ein lauffähiges System vorhanden ist.“ ([1], S. 52)

Im Gegensatz zu einem Lo-Fi-Prototyp ist ein Hi-Fi-Prototyp detailreicher und bietet oft bereits einen gewissen Funktionsumfang bzw. Interaktionsmöglichkeiten. Er kommt dem finalen Produkt in der Regel sehr nah.



Ziel

- Erarbeitung und Optimierung der Benutzerschnittstelle inkl. der Details
- Grundlage für User Interface-Evaluation (z. B. Usability Test, Heuristische Evaluation) schaffen
- Finalisierung von Konzept und Entwurf eines Produkts → User Interface-Spezifikation (für die Übergabe in die Umsetzung)

Ressourcen

Werkzeuge	Verschiedene Möglichkeiten (vgl. [1], S. 61):	Teilnehmer	• 1 Designer	Zeit	• ? ³
	• Bildbearbeitungsprogramm				

¹ Ein pauschaler Zeitaufwand und Schwierigkeitsgrad sind sehr schwer/nicht einzuschätzen, da dies nicht nur von der Produktkomplexität und dem geplanten Umfang des Prototyps abhängt, sondern auch vom verwendeten Tool oder der Detailtreue, die angestrebt wird. (vgl. [1], S. 63)

² s. o.

³ s. o.

- oder Grafikprogramm
 - oder UI Prototyping-Werkzeug
 - oder Multimediawerkzeug
 - oder Programmierwerkzeug
- Bildermaterial zur Visualisierung

Vorgehen

User Interface-Prototypen eignen sich zum einen zur Konzeption, Entwicklung und Evaluation der Benutzerschnittstelle einer zu entwickelnden Software, zum anderen können sie eingesetzt werden, um das entwickelte, getestete und optimierte Design anschaulich festzuhalten und in die Umsetzung zu übergeben.

Hi-Fi-Prototypen werden dabei eingesetzt, um auf Basis bereits evaluierten und verbesserten Lo-Fi-Prototypen ein Produkt weiterzuentwickeln und sich dem finalen Produkt anzunähern. Dazu können bzw. sollten die Hi-Fi-Prototypen ebenfalls evaluiert und auf Basis des erhaltenen Feedbacks verbessert werden.

UI-Prototyp zur User Interface-Entwicklung

Mittels User Interface-Prototypen wird ein für das zu entwickelnde Produkt geeignetes User Interface-Konzept ausgearbeitet (vgl. [1], S. 55). Dieses Konzept sollte schließlich folgende Aspekte beinhalten:

- „Grundsätzlicher Aufbau und Screen-Layout
- Anzeige- und Eingabegeräte
- Aufteilung und Struktur von Informationen
- Verwendung und Verhalten von Fenstern
- Wichtige Bedienelemente
- Navigation mittels Menüs, Schaltflächen und Links
- Prüfung von Eingaben und Anzeige von Fehlermeldungen
- Konzepte für das Speichern von Informationen und Zuständen
- Rückgängig machen und erneut ausführen
- Interaktionsprinzipien wie Gesten, direkte Manipulation, Drag&Drop oder Kontextmenüs“ ([1], S. 56)

„Dabei muss auch die Technologie der Zielplattform berücksichtigt werden, beispielsweise Eingabe- und Ausgabemedien, Betriebssystem, Bildschirmgrößen und Auflösung. Besonders bei der Entwicklung mobiler Software ist dies von besonderer Bedeutung.“ ([1], S. 56)

„Abhängig vom Ziel, das verfolgt wird, können unterschiedliche Arten von UI-Prototypen zum Einsatz kommen. Um den geplanten Prototypen näher zu charakterisieren, lassen sich die folgenden Dimensionen unterscheiden:

- *Funktionsumfang*: Wie viel der vorgesehenen Funktionalität der Benutzerschnittstelle soll im Prototyp gezeigt werden? Sind dies ausgewählte Ausschnitte, oder geht es darum, den gesamten Umfang darzustellen?
- *Funktionstiefe*: Wie detailliert sollen die einzelnen funktionalen Elemente wiedergegeben werden? Sollen beispielsweise mehrstufige Berechnungen nur angedeutet werden, oder sind die Zwischenschritte und ihre Resultate entscheidend?
- *Darstellungstreue*: Wie ähnlich soll der Prototyp dem Endprodukt in Bezug auf Aussehen der Benutzeroberfläche (Look & Feel) sein?

- *Interaktivität:* Wie interaktiv soll der Prototyp sein? Braucht es lauffähige Beispiele, um komplexe Abläufe wiederzugeben, oder genügen statische Darstellungen der Benutzerschnittstelle?
- *Datengehalt:* Sollen reale Daten zum Einsatz kommen, genügen realistische Beispiele oder gar Platzhalter für Bezeichnungen und dargestellte Informationen? Wie relevant ist die dargestellte Menge an Informationen?
- *Technische Reife:* Wie viel der endgültigen User-Interface-Technologie soll im Prototypen verwendet werden? Muss der Prototyp mit der Entwicklungsumgebung der Zielplattform entwickelt werden, oder sind einfache Zeichnungswerkzeuge ausreichend?“ ([1], S. 52-53)

Prototypen können dabei mit den verschiedensten Tools erstellt werden. In Frage kommen: Stift und Papier etc., Office-Anwendungen, Bildbearbeitungsprogramme, Grafikprogramme, UI Prototyping-Werkzeuge, Multimediawerkzeuge oder Programmierwerkzeuge. (vgl. [1], S. 61)

Nach der Erstellung des Prototyps kann dieser mit verschiedenen Testpersonen (am besten mit Endnutzern im realen Nutzungskontext) im Rahmen von Usability Tests evaluiert werden.

UI-Prototyp für die User Interface-Spezifikation

„Mit einem User Interface-Prototypen können viele Aspekte der Benutzerschnittstelle auf anschauliche Weise festgehalten werden [...] Im Rahmen der Spezifikation kann ein Prototyp für z. B. für folgende Zwecke eingesetzt werden:

- Illustration des Funktionsumfangs
- Verdeutlichung der Funktionsweise
- Spezifizieren der User-Interface-Elemente
- Aufzeigen der Navigation und Interaktion
- Visualisierung der geplanten Lieferung
- Abschätzung des Realisierungsaufwands durch die Entwickler“ ([1], S. 58)

„In der Regel sind im Rahmen der User-Interface-Spezifikation die Darstellung des Funktionsumfangs und für ausgewählte Aspekte auch eine realistische Funktionstiefe gefordert. Eine gewisse Interaktivität ist ebenfalls hilfreich, da so verschiedene Zustände des User Interface visualisiert werden können. Ein solcher Prototyp wird deshalb in vielen Fällen mit einer Technologie erstellt, welche der Zieltechnologie nahekommt. Bei einfacheren Produkten genügen auch aber Grafik- und Bildbearbeitungsprogramme.“ ([1], S. 58)

Hilfsmittel



Templates

- Styleguide Android
- Styleguide iOS
- Styleguide WindowsPhone
- Template Hi-Fi iOS 7 iPhone 5 (Photoshop)
- Template Hi-Fi WindowsPhone 8 (Photoshop)
- Template Hi-Fi Android 4.4 KitKat Nexus 5 (Photoshop)



Tools

- s. Toolliste
-

Das sollten Sie beachten

Kein zu früher Perfektionismus:

„User Interface Prototyping ist eine iterative Tätigkeit. Investieren Sie nicht zu viel Zeit in die Perfektionierung, bevor Sie Feedback einholen.“ ([1], S. 62) In der Regel ist es ratsam nicht mit Hi-Fi Prototyping zu beginnen, sondern dieses erst einzusetzen, wenn bereits iterativ verbesserte Lo-Fi-Prototypen vorliegen.

Ohne großen Aufwand Änderungen:

„Es lohnt sich, ein Werkzeug zu wählen, mit dem ohne großen Aufwand verschiedene Varianten ausgearbeitet und Änderungen umgesetzt werden können. Häufig wird zu früh auf Programmierwerkzeuge für die Erstellung von UI-Prototypen gesetzt.“ ([1], S. 62) Besonders zu Beginn eines Projekts bzw. zu Beginn der Ausarbeitung von Ideen müssen noch viele Änderungen vorgenommen werden oder Prototypen werden komplett verworfen. Daher lohnt es sich vor allem an dieser Stelle möglichst aufwandsarme Lo-Fi Prototypen (z. B. mit Papier und Stift) zu entwickeln.

Endgerät beachten:

„Bei Geräten sollten Prototypen die Software in Kombination mit der Hardware zeigen. Anzahl, Größe und Anordnung von Bedienelementen beeinflussen das Konzept auf dem Display maßgeblich.“ ([1], S. 62)

Kompromiss finden:

„Jeder Prototyp stellt einen Kompromiss zwischen notwendigem Aufwand und Zweck dar. Bevor Sie mit UI Prototyping loslegen, sollten Sie sich deshalb im Klaren sein, welche Fragestellungen Sie verfolgen. Daraus lässt sich ableiten, welche Art von UI-Prototyp geeignet ist.“ ([1], S. 53)

Quellen

- [1] Richter, M. & Flückiger, M. D. (2013). *Usability Engineering kompakt – Benutzbare Software gezielt entwickeln*. 3. Auflage. Berlin: Springer Verlag

USER INTERFACE PROTOTYPING, LO-FI

Aufwand	 ¹	Schwierigkeit	 ²	Eignung mobil	
Abhängigkeiten	<p>Nutzungskontext-analyse (z. B. Kontextuelles Interview) und deren Ergebnisse (z. B. Personas), Nutzungsanforderungen, Szenarios, Ideenfindung (z. B. Brainstorming)</p>		Projektphasen	<p>Konzeption & Entwicklung</p>	
			Beteiligte Rollen	<p>Software-Gestalter</p>	

Kurzbeschreibung

„User Interface Prototyping (UI Prototyping) wird im Usability Engineering eingesetzt, um Aspekte der Benutzerschnittstelle zu entwerfen, zu evaluieren und zu verbessern, noch bevor ein lauffähiges System vorhanden ist.“ ([1], S. 52)

Im Gegensatz zu einem Hi-Fi-Prototyp ist ein Lo-Fi-Prototyp weniger detailreich und bietet oft noch keine Interaktionsmöglichkeiten.



Ziel

- Erarbeitung und Optimierung der Benutzerschnittstelle inkl. der Details
- Grundlage für User Interface-Evaluation (z. B. Usability Test, Heuristische Evaluation) schaffen

Ressourcen

Werkzeuge Verschiedene Möglichkeiten **Teilnehmer** • 1 Designer/Gestalter **Zeit** • 13,5 PT³

¹ Ein pauschaler Zeitaufwand und Schwierigkeitsgrad sind sehr schwer/nicht einzuschätzen, da dies nicht nur von der Produktkomplexität und dem geplanten Umfang des Prototyps abhängt, sondern auch vom verwendeten Tool oder der Detailtreue, die angestrebt wird. (vgl. [1], S. 63)

² s. o.

³ s. o.

(vgl. [1], S. 61):

- Stift, Papier
- oder Office-Anwendung
- oder UI Prototyping-Werkzeug

Vorgehen

User Interface-Prototypen eignen sich zum einen zur Konzeption, Entwicklung und Evaluation der Benutzerschnittstelle einer zu entwickelnden Software, zum anderen können sie eingesetzt werden, um das entwickelte, getestete und optimierte Design anschaulich festzuhalten und in die Umsetzung zu übergeben.

Lo-Fi-Prototypen werden dabei eingesetzt, um besonders in frühen Projektstadien Konzeptideen etc. zu visualisieren, evaluieren und anschließend zu verbessern. Später können dann auf Basis bereits evaluierten und verbesserten Lo-Fi-Prototypen Hi-Fi-Prototypen erstellt werden, um ein Produkt weiterzuentwickeln und sich dem finalen Produkt anzunähern.

UI-Prototyp zur User Interface-Entwicklung

Mittels User Interface-Prototypen wird ein für das zu entwickelnde Produkt geeignetes User Interface-Konzept ausgearbeitet (vgl. [1], S. 55). Dieses Konzept sollte schließlich folgende Aspekte beinhalten:

- „Grundsätzlicher Aufbau und Screen-Layout
- Anzeige- und Eingabegeräte
- Aufteilung und Struktur von Informationen
- Verwendung und Verhalten von Fenstern
- Wichtige Bedienelemente
- Navigation mittels Menüs, Schaltflächen und Links
- Prüfung von Eingaben und Anzeige von Fehlermeldungen
- Konzepte für das Speichern von Informationen und Zuständen
- Rückgängig machen und erneut ausführen
- Interaktionsprinzipien wie Gesten, direkte Manipulation, Drag&Drop oder Kontextmenüs“ ([1], S. 56)

„Dabei muss auch die Technologie der Zielplattform berücksichtigt werden, beispielsweise Eingabe- und Ausgabemedien, Betriebssystem, Bildschirmgrößen und Auflösung. Besonders bei der Entwicklung mobiler Software ist dies von besonderer Bedeutung.“ ([1], S. 56)

„Abhängig vom Ziel, das verfolgt wird, können unterschiedliche Arten von UI-Prototypen zum Einsatz kommen. Um den geplanten Prototypen näher zu charakterisieren, lassen sich die folgenden Dimensionen unterscheiden:

- *Funktionsumfang*: Wie viel der vorgesehenen Funktionalität der Benutzerschnittstelle soll im Prototyp gezeigt werden? Sind dies ausgewählte Ausschnitte, oder geht es darum, den gesamten Umfang darzustellen?
- *Funktionstiefe*: Wie detailliert sollen die einzelnen funktionalen Elemente wiedergegeben werden? Sollen beispielsweise mehrstufige Berechnungen nur angedeutet werden, oder sind die Zwischenschritte und ihre Resultate entscheidend?
- *Darstellungstreue*: Wie ähnlich soll der Prototyp dem Endprodukt in Bezug auf Aussehen der Benutzeroberfläche (Look & Feel) sein?

- *Interaktivität:* Wie interaktiv soll der Prototyp sein? Braucht es lauffähige Beispiele, um komplexe Abläufe wiederzugeben, oder genügen statische Darstellungen der Benutzerschnittstelle?
- *Datengehalt:* Sollen reale Daten zum Einsatz kommen, genügen realistische Beispiele oder gar Platzhalter für Bezeichnungen und dargestellte Informationen? Wie relevant ist die dargestellte Menge an Informationen?
- *Technische Reife:* Wie viel der endgültigen User-Interface-Technologie soll im Prototypen verwendet werden? Muss der Prototyp mit der Entwicklungsumgebung der Zielplattform entwickelt werden, oder sind einfache Zeichnungswerkzeuge ausreichend?“ ([1], S. 52-53)

Prototypen können dabei mit den verschiedensten Tools erstellt werden. In Frage kommen: Stift und Papier etc., Office-Anwendungen, Bildbearbeitungsprogramme, Grafikprogramme, UI Prototyping-Werkzeuge, Multimediawerkzeuge oder Programmierwerkzeuge. (vgl. [1], S. 61)

Nach der Erstellung des Prototyps kann dieser mit verschiedenen Testpersonen (am besten mit Endnutzern im realen Nutzungskontext) im Rahmen von Usability Tests evaluiert werden.

UI-Prototyp für die User Interface-Spezifikation

„Mit einem User Interface-Prototypen können viele Aspekte der Benutzerschnittstelle auf anschauliche Weise festgehalten werden [...] Im Rahmen der Spezifikation kann ein Prototyp für z. B. für folgende Zwecke eingesetzt werden:

- Illustration des Funktionsumfangs
- Verdeutlichung der Funktionsweise
- Spezifizieren der User-Interface-Elemente
- Aufzeigen der Navigation und Interaktion
- Visualisierung der geplanten Lieferung
- Abschätzung des Realisierungsaufwands durch die Entwickler“ ([1], S. 58)

„In der Regel sind im Rahmen der User-Interface-Spezifikation die Darstellung des Funktionsumfangs und für ausgewählte Aspekte auch eine realistische Funktionstiefe gefordert. Eine gewisse Interaktivität ist ebenfalls hilfreich, da so verschiedene Zustände des User Interface visualisiert werden können. Ein solcher Prototyp wird deshalb in vielen Fällen mit einer Technologie erstellt, welche der Zieltechnologie nahekommt. Bei einfacheren Produkten genügen auch aber Grafik- und Bildbearbeitungsprogramme.“ ([1], S. 58)

Hilfsmittel



Templates

- Styleguide Android
- Styleguide iOS
- Styleguide Windowsphone
- Lo-Fi Wireframes Beschreibung
- Lo-Fi Paper Prototyping Beschreibung
- Template Lo-Fi iPhone sketchy (OmniGraffle)
- Template Lo-Fi iPad sketchy (OmniGraffle)
- Template Lo-Fi iPhone (Photoshop)



Tools

- s. Toolliste

- Template Lo-Fi Wireframe iPhone (GoogleDocs)
- Templates Paper Prototyping Android
- Templates Paper Prototyping iOS
- Templates Paper Prototyping WindowsPhone

Das sollten Sie beachten

Kein zu früher Perfektionismus:

„User Interface Prototyping ist eine iterative Tätigkeit. Investieren Sie nicht zu viel Zeit in die Perfektionierung, bevor Sie Feedback einholen.“ ([1], S. 62) In der Regel ist es ratsam nicht mit Hi-Fi Prototyping zu beginnen, sondern dieses erst einzusetzen, wenn bereits iterativ verbesserte Lo-Fi-Prototypen vorliegen.

Ohne großen Aufwand Änderungen:

„Es lohnt sich, ein Werkzeug zu wählen, mit dem ohne großen Aufwand verschiedene Varianten ausgearbeitet und Änderungen umgesetzt werden können. Häufig wird zu früh auf Programmierwerkzeuge für die Erstellung von UI-Prototypen gesetzt.“ ([1], S. 62) Besonders zu Beginn eines Projekts bzw. zu Beginn der Ausarbeitung von Ideen müssen noch viele Änderungen vorgenommen werden oder Prototypen werden komplett verworfen. Daher lohnt es sich vor allem an dieser Stelle möglichst aufwandsarme Lo-Fi Prototypen (z. B. mit Papier und Stift) zu entwickeln.

Endgerät beachten:

„Bei Geräten sollten Prototypen die Software in Kombination mit der Hardware zeigen. Anzahl, Größe und Anordnung von Bedienelementen beeinflussen das Konzept auf dem Display maßgeblich.“ ([1], S. 62)

Kompromiss finden:

„Jeder Prototyp stellt einen Kompromiss zwischen notwendigem Aufwand und Zweck dar. Bevor Sie mit UI Prototyping loslegen, sollten Sie sich deshalb im Klaren sein, welche Fragestellungen Sie verfolgen. Daraus lässt sich ableiten, welche Art von UI-Prototyp geeignet ist.“ ([1], S. 53)

Quellen

- [1] Richter, M. & Flückiger, M. D. (2013). *Usability Engineering kompakt – Benutzbare Software gezielt entwickeln*. 3. Auflage. Berlin: Springer Verlag

USABILITY TEST IM FELD

Aufwand	 ¹	Schwierigkeit		Eignung mobil	
Abhängigkeiten	Lo-Fi/Hi-Fi Prototyping ²	Projektphasen	Analyse, Konzeption & Entwicklung, Umsetzung, Betrieb	Beteiligte Rollen	Software-Gestalter, Benutzer

Kurzbeschreibung

Bei einem Usability-Test im Feld bedienen (potentielle) Nutzer eine Software oder einen Prototypen einer Software und führen vorgegebene Tasks aus. Besonders ratsam ist es dabei – besonders bei der Entwicklung mobiler Software – das Testing unter Berücksichtigung des eigentlichen Nutzungskontexts durchzuführen. (vgl. [5], S. 86)

Bei diesem qualitativen Ansatz wird die Gesamtheit der Erfahrungen, die die Zielgruppe bei der Nutzung einer interaktiven Anwendung macht, ermittelt. (vgl. [4]) Zusätzlich wird ein besonderer Fokus darauf gelegt, Bedienschwierigkeiten – vor allem auch solche, die sich durch die Anwendung der Software in einem bestimmten Nutzungskontext ergeben – zu identifizieren.



Ziel

- Identifikation von Usability-Problemen einer Software etc. besonders auch in Bezug auf den Nutzungskontext
- Betrachtung der gesamten User Experience (z. B. Spaß bei der Nutzung) (vgl. [4])
- Erkennen von Entscheidungs- und Informationsverhalten der Nutzer

Ressourcen

Werkzeuge	• Endgerät mit zu testender SW, oder Prototyp jeglicher Art (z.	Teilnehmer	• 1 Observator • 3(-5) Testpersonen	Zeit	• min. 3 Tage ³ (vgl. [2])
------------------	---	-------------------	--	-------------	---------------------------------------

¹ Vor allem der Zeitaufwand ist sehr schwer einzuschätzen. Er hängt stark vom Projekt/Prototypumfang/Nutzungskontext etc. ab. Es sollte aber mindestens 1 Tag für die Vorbereitung, 1 Tag für das Testing und 1 Tag für die Nachbereitung eingeplant werden. [2]

² Wenn noch keine fertige Software für das Testing vorhanden ist, werden Prototypen für das Testing benötigt.

³ s. Fußnote 1

- B. Paper Prototype)
- Stift, Papier (vgl. [1])

(Zielgruppe) (vgl. [1])

Vorgehen

Beim Usability-Test werden Testsessions mit einzelnen (potentiellen) Nutzern durchgeführt. Die Anwender bekommen verschiedene Aufgaben, die sie mit der Software (oder dem Prototypen) ausführen sollen. Besonders ist dabei, dass die Tests im eigentlichen Nutzungskontext durchgeführt werden (z. B. Test eines Navigationssystems im Auto während des Fahrens). Handelt es sich bei dem Testobjekt um einen Papier-Prototyp, kann die Methode „Wizard-of-Oz“ zum Einsatz kommen, um geplante Interaktionen zu testen.

Während des Tests werden die Testpersonen von einem Observator, der sich Notizen macht, beobachtet. Sind vorab quantitative Usability-Ziele aufgestellt worden, können diese während des Tests auf ihre Erreichung überprüft werden. Nach dem der Proband die Test-Aufgaben durchgeführt hat, sollte er noch einmal zum Produkt befragt werden (z. B. Interview oder Fragebogen).

Die einzelnen Schritte

1. Vorbereitung/Planung (Siehe Steckbrief: *Usability Test Planning*)
2. Durchführung des Tests
 - Probanden instruieren
 - Test starten
 - Beobachtungen notieren
 - Test beenden
 - Probanden befragen
3. Nachbereitung
 - Auswertung der Notizen/Testergebnisse

Hilfsmittel



Templates

- Template Usability-Test Report
- Template Usability-Test Task
- Anleitung Wizard-of-Oz-Methode
- Fragebogen IsometrS
- Fragebogen System Usability Scale



Tools

- ---

Das sollten Sie beachten

Realer Nutzungskontext

Berücksichtigen Sie nach Möglichkeit immer den realen Nutzungskontext (Benutzer, Arbeitsaufgaben, Arbeitsmittel (Hardware, Software und Materialien) sowie die physische und soziale Umgebung) des zu entwickelnden Produkts.

Vor allem bei der Entwicklung mobiler Software ist dies wichtig. Nur im Notfall (z. B. bei Ressourcenknappheit) sollten die Tests z. B. im Büro durchgeführt werden.

Kunde \neq Endnutzer

Denken Sie vor diesem Hintergrund auch daran, dass der Kunde oft nicht (einer) der Endnutzer ist. In der Regel kennen nur tatsächliche/potentielle ihre eigenen Arbeitsabläufe etc. en détail.

Fokus auf qualitativen Ergebnissen:

Die Tests mit relativ wenig Probanden (min. 3) durchzuführen ist vollkommen ausreichend, da vor allem qualitative Ergebnisse von Belang sind. Die Erhebung quantitativer Ergebnisse sollte bei Usability Tests nicht im Vordergrund stehen. (vgl. [3])

Thinking aloud

Instruieren Sie die Probanden während des Tests laut zu denken. Oftmals erfahren Sie nur so, was z. B. die konkreten Probleme der Testpersonen sind oder was sie fühlen. (vgl. [1])

Einfaches Equipment

Es ist völlig ausreichend sich während des Tests Notizen zu machen und ggf. die Zeiten zu stoppen. Aufwändigeres Equipment wie z. B. Videokameras kann zwar während des Tests zunächst praktisch erscheinen, allerdings ist so nicht nur der Testaufbau – vor allem im Feld – aufwändiger, sondern auch die Nachbereitung anschließend ist mühsamer. (vgl. [1])

Früh testen

Auch wenn Usability Tests in allen Phasen der Softwareentwicklung eingesetzt werden können, empfiehlt es sich dringend mit dem Testing schon sehr früh im Projekt zu beginnen. Je länger man wartet, desto kostenintensiver wird die Behebung identifizierter Fehler und Probleme. (vgl. [6], S. 106)

Quellen

- [1] Nielsen, J. (1994). Guerrilla HCI: Using Discount Usability Engineering to Penetrate the Intimidation Barrier. <http://www.nngroup.com/articles/guerrilla-hci/> (Geprüft: 05.08.2014)
- [2] Nielsen, J. (2009). Anybody Can Do Usability. <http://www.nngroup.com/articles/anybody-can-do-usability/> (Geprüft: 05.08.2014)
- [3] Nielsen, J. (2009). Discount Usability: 20 Years. <http://www.nngroup.com/articles/discount-usability-20-years/> (Geprüft: 05.08.2014)
- [4] Designmethodenfinder <http://www.designmethodenfinder.de/usability-testing?s1> (Geprüft: 05.08.2014)
- [5] Richter, M. & Flückiger, M. D. (2013). Usability Engineering kompakt – Benutzbare Software gezielt entwickeln. 3. Auflage. Berlin: Springer Verlag
- [6] Rudlof, C. (2006). Handbuch Software-Ergonomie – Usability Engineering. 2. Auflage. <http://www.ukpt.de/pages/dateien/software-ergonomie.pdf> (Geprüft: 05.08.2014)

USABILITY TEST PLANNING

Aufwand	■ ■ ¹	Schwierigkeit	■ ■ ²	Eignung mobil	+ / -
Abhängigkeiten	Nutzungsanforderungen (z. B. User Stories)	Projektphasen	Konzeption & Entwicklung	Beteiligte Rollen	Projektmanager, Software-Gestalter, Kunde

Kurzbeschreibung

Beim Usability Test Planning entstehen die Akzeptanzkriterien für die Endabnahme. Das Resultat dieser Maßnahme ist die Erstellung einer strukturierten Beschreibung des am Ende einer Entwicklungsperiode anstehenden Usability-Tests. Das Dokument beinhaltet die wichtigsten Fakten wie Umfang des Tests, dessen Ziele, eingesetzte Metriken, adressierte Zielgruppe, Aufgaben etc.

Ziel

- Planung des anstehenden Nutzertests
- Orientierungshilfe für Entwickler und Gestalter, damit diese von der Zielsetzung getrieben die Umsetzung ausüben
- Erstellung von messbaren Akzeptanzkriterien

Ressourcen

Werkzeuge	<ul style="list-style-type: none"> • Textverarbeitungssoftware • Stift, Paper 	Teilnehmer	<ul style="list-style-type: none"> • 1 Projektmanager • 1 Software-Gestalter • 1 Kunde 	Zeit	<ul style="list-style-type: none"> • min. 1 PT³ (vgl. [3])
------------------	---	-------------------	---	-------------	--

¹ Vor allem der Zeitaufwand ist sehr schwer einzuschätzen. Er hängt stark vom Projekt/Prototypumfang/Nutzungskontext etc. ab. Es sollte aber mindestens 1 Tag für die Vorbereitung eingeplant werden. [3]

² s. o.

³ s. o.

Vorgehen

Ein Usability Test sollte wohlüberlegt geplant und durchgeführt werden. Sowohl das Herstellerunternehmen, welches den Test durchführt, als auch der Kunde/das Unternehmen, das die Probanden bereitstellt, möchten ihre Investition/Ausgaben sinnvoll eingesetzt wissen. Die Erarbeitung eines standfesten Plans sollte für die Kommunikation mit dem Kunden bestehen. Zur Erstellung des Plans sollten folgende Maßnahmen durchgeführt werden.

Zunächst einmal sollte gemeinsam der Umfang, den es zu Testen gilt erarbeitet werden. Dazu gehören einerseits prinzipielle Angaben zum Produkt und dem zu testenden Bereich z. B. Navigation, bestimmte Features o. ä. und andererseits die Angabe ob es sich um einen Prototyp, ein fertiges Produkt o. ä. handelt. Ein guter Startpunkt für den Umfang ist die Betrachtung der ausgewählten User Stories für die anstehende Entwicklungsperiode.

Zudem ist es essentiell die richtigen Zielgruppen für den Test zu identifizieren. Dessen Eigenschaften sollten schriftlich festgehalten werden und dienen für die Akquirierung von Probanden. Hierbei könnte man sich z. B. an ggf. erstellten Personas orientieren. Pro Zielgruppe sollten für einen qualitativen Test 5 Personen vorhanden sein. Bei quantitativen Tests gibt jeder Proband, der zusätzlich am Test teilnimmt, eine höhere Sicherheit bzgl. der Genauigkeit der Resultate, daher sind für quantitative Tests auch mehr Personen pro Zielgruppe einzuplanen (siehe [4]).

In einem darauffolgenden Schritt sollten die Ziele des Tests definiert werden. Diese können entweder generelle oder auch spezifische Fragestellungen behandeln. Aufbauend darauf sollten entsprechende Aufgaben und Szenarien erstellt werden. Diese sollten den Nutzer bzw. Probanden durch ihre realitätsnähe das Gefühl vermitteln im realen Nutzungskontext zu sein.

Je nach Fragestellung und verbundener Aufgabe sollten Metriken zu den Aufgaben erstellt werden. Beispielsweise können sich unter den quantitativen Metriken die *Successful Task Completion*, *Critical Errors*, *Non-Critical Errors*, *Error-Free Rate*, *Time On Task*, *Amount of Clicks* etc. befinden. Qualitative Metriken können subjektive Fragestellungen, *Likes*, *Dislikes*, Empfehlungen und Fragebögen beinhalten.

Neben den Metriken ist es zudem wichtig Schwellenwerte zu ermitteln, welche als Akzeptanzkriterien dienen. Ohne diese würden die Messwerte keinerlei Effekt haben. Weder ließe sich genau sagen, was noch verbessert werden muss noch was sich in einem abnahmefähigen Zustand befindet.

Durch die vorher behandelten und bestimmten Themen ergeben sich auch Anforderungen hinsichtlich des benötigten Equipments, dass zum Einsatz kommt. Dieses sollte in jedem Fall schriftlich festgehalten werden, wodurch eine persistente Erinnerung besteht – Nichts ist peinlicher als das Vergessen von benötigten Geräten für einen Test.

Bevor die Nutzer akquiriert werden, ist es notwendig weitere organisatorische Rahmenbedingungen geklärt zu haben. So müssen Ablauf der Sessions, der Ort wie auch der Zeitplan für die Tests feststehen – keiner mag es zu warten.

Für die interne Organisation ist es zudem wichtig einzuteilen, wer welche Aufgaben vor, während und nach dem Test übernimmt.

In den letzten Schritten der Vorbereitung sollte der Plan mit dem Kunden durchgegangen werden sowie Probanden rekrutiert werden.

Hilfsmittel



Templates

- Checkliste Usability Test Plan
- Leitfaden Probanden-Akquise
- Template Usability-Test Tasks



Tools

- ---
-

Literatur-Empfehlungen

- <http://www.nngroup.com/articles/how-many-test-users/>
- <http://www.usability.gov/get-involved/blog/2013/09/measurable-usability-goals.html>

Das sollten Sie beachten

Easy to use vs. easy to learn

Überlegen Sie, ob bei dem zu entwickelnde Produkt der Fokus auf easy to use (z. B. komplexere Betriebssoftware) oder easy to learn (z. B. Software im CarSharing-Auto) liegen soll.

Bei umfangreicheren easy to use-Prototypen kann eine kurze Schulung/Einführung der Probanden in das Produkt nötig sein. Werden allerdings Prototypen, die easy to learn sein sollen, getestet, sollte die Erklärung des Produkts eher knapper ausfallen, damit durch den Test auch festgestellt werden kann, ob das Produkt tatsächlich einfach zu erlernen ist.

Realer Nutzungskontext

Berücksichtigen Sie nach Möglichkeit immer den realen Nutzungskontext (Benutzer, Arbeitsaufgaben, Arbeitsmittel (Hardware, Software und Materialien) sowie die physische und soziale Umgebung) des zu entwickelnden Produkts.

Vor allem bei der Entwicklung mobiler Software ist dies wichtig. Nur im Notfall (z. B. bei Ressourcenknappheit) sollten die Tests z. B. im Büro durchgeführt werden.

Kunde \neq Endnutzer:

Denken Sie vor diesem Hintergrund auch daran, dass der Kunde oft nicht (einer) der Endnutzer ist. In der Regel kennen nur tatsächliche/potentielle ihre eigenen Arbeitsabläufe etc. en détail.

Fokus auf qualitativen Ergebnissen:

Die Tests mit relativ wenig Probanden (min. 3) durchzuführen ist vollkommen ausreichend, da vor allem qualitative Ergebnisse von Belang sind. Die Erhebung quantitativer Ergebnisse sollte bei Usability Tests nicht im Vordergrund stehen. (vgl. [2])

Quellen

[1] Usability.gov – *Improving the User Experience*

<http://www.usability.gov/how-to-and-tools/methods/planning-usability-testing.html> (Geprüft: 05.08.2014)

[2] Nielsen, J. (2009). *Discount Usability: 20 Years.*

<http://www.nngroup.com/articles/discount-usability-20-years/> (Geprüft: 05.08.2014)

[3] Nielsen, J. (2009). *Anybody Can Do Usability.*

<http://www.nngroup.com/articles/anybody-can-do-usability/> (Geprüft: 05.08.2014)

[4] Nielsen, J. (2012). *How Many Test Users in a Usability Study?*

<http://www.nngroup.com/articles/how-many-test-users/> (Geprüft: 05.08.2014)