

Synchrone Internet-basierte Usability Tests - Mit wenig Aufwand den Nutzer frühzeitig verstehen

Tino Truppel
RapidRabb.it GmbH
Warschauerstr. 58a
10243 Berlin
tino.truppel@rapidrabb.it
<https://rapidrabb.it>

Abstract

Usability ist mittlerweile nicht nur ein verbreitetes Qualitätsmerkmal von Software, auch investieren immer mehr an der Softwareentwicklung beteiligte Firmen tatsächlich Kapital, um dieses Ziel zu erreichen. Methoden und Vorgehensweisen aus der akademischen Welt werden auch in der Industrie immer stärker aufgenommen, ausprobiert und den praktischen Erfahrungen angepasst. Oft beziehen sich solcher Art Anpassungen weniger auf die Effektivität der Methoden, sondern eher auf deren Effizienz. Denn der praktische Alltag ist selten frei von ökonomischen Zwängen.

Auch Usability Testing ist mit herkömmlichen Werkzeugen eine eher aufwändige Methode und muss vor Kunden und Geldgebern nicht selten immer wieder neu begründet werden. Aber gerade in der Frühphase der Konzeption kann Usability Testing sehr wertvoll sein, da Probleme einfacher und günstiger behoben werden können, je eher sie entdeckt werden. Dieser Beitrag stellt eine praktikable Lösung vor: web-basiertes Rapid Prototyping, welches schon frühzeitig erlaubt, potenzielle Nutzer an der Konzeption partizipieren zu lassen.

Keywords

Usability Test, Remote, Web-based, Rapid Prototyping, klickbare Wireframes

1.0 Einleitung

Die Entwicklung von Software ist ein dynamischer, kreativer und komplexer Prozess. Besonders deutlich wird dies während der Konzeption der Nutzerinteraktionen: Die Ideen müssen mit den Vorstellungen eines heterogenen Teams aus Entwicklern, Designern und anderen Akteuren zusammengebracht und auf eine für alle Beteiligten verständliche Art und Weise dokumentiert werden. Zusätzlich muss so früh wie möglich das Feedback der Zielgruppe eingeholt werden – gerade hier stellt die Kommunikation der Designideen eine Herausforderung dar.

Übliche Kommunikationsmittel sind statische, sogenannte *low-fidelity* Prototypen unterschiedlichsten Typs. Dies können Wireframes oder Mockups etwa in Form von Papierprototypen, Powerpoint-Folien, HTML-Klick-Dummys oder

Grafiken sein. Die genannten Formate haben eine weitere Eigenschaft gemeinsam: sie lassen sich schlecht verteilen und damit nur aufwändig mit potenziellen Nutzern evaluieren. Oft bleibt nur eine Evaluierung im eigenen Labor. Doch nicht in jeden Entwicklungsprozess ist das benötigte Zeit- und Kostenbudget dafür vorgesehen.

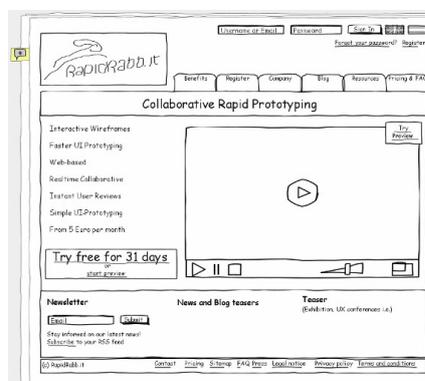


Abb. 1: Beispiel eines *low-fidelity* Prototypen, erzeugt mit RapidRabb.it Prototype Creator

Dieser Beitrag präsentiert kostengünstige Ergänzungen bzw. Alternativen zur Evaluierung von solcher Art *low-fidelity* Prototypen, indem verbreitete Evaluierungsmethoden aus der Konzeptionsphase diskutiert und die Möglichkeit diese verteilt über das Internet durchzuführen aufgezeigt werden. Methodischer Fokus liegt hierbei auf der direkten Kommunikation mit dem Nutzer, da nur qualitatives Feedback es ermöglicht, Probleme nicht nur aufzudecken, sondern auch Gründe für diese zu finden. Dieser Beitrag stellt keine neuen Methoden vor, sondern zeigt auf, mit welchen Werkzeugen bekannte und bewährte Methoden in der Evaluation von frühen Prototypen in das Web verlagert werden können.

Je eher ein Problem identifiziert wird, umso günstiger ist die Behebung dessen. Dieser Grundsatz gilt besonders bei der Konzeption und Entwicklung von

gebrauchstauglicher Software. Sind grundlegende Konzepte, etwa der Interaktion, nicht konsistent oder unvollständig und wird dieser Mangel erst sehr spät, zum Beispiel erst während der Implementierung, festgestellt, so ist die Beseitigung nicht selten so teuer, dass sie nur oberflächlich durchgeführt werden kann. Das Resultat ist dann anhand der Qualität im Endprodukt sichtbar.

Viele grundlegende konzeptionelle Entscheidungen von Software werden in frühen Entwürfen getroffen. Deshalb ist es gerade in dieser Phase essenziell, diese Entscheidungen zu evaluieren. Ein Ansatz hierfür ist das Rapid Prototyping.

2.0 Rapid Prototyping

Rapid Prototyping während der Entwicklung von Software zeichnet sich durch folgende drei Punkte aus:

- Wesentlich ist die Nutzung von

Prototypen und das schon vor dem Schreiben der ersten Zeile Quellcode. Hierbei können Prototypen unterschiedlich abstrakt und vollständig in Bezug zum eigentlichen Produkt sein.

- Alle relevante Beteiligte sollen in die Evaluation der Prototypen einbezogen werden. Beteiligte können Entwickler (geben Feedback zur technischen Machbarkeit), Auftraggeber bzw. das Management (geben Feedback zu wirtschaftlichen Aspekten), externe Berater (z.B. aus Marketing, Design etc.) und potenzielle Nutzer (geben Feedback zur Gebrauchstauglichkeit) sein.
- Das gewonnene Feedback fließt in einem neuen Prototyp ein, welcher wiederum evaluiert wird und anschließend dem Feedback angepasst wird. Dies wird solange iterativ fortgesetzt, bis das Evalu-

ierungsergebnis den gesetzten Qualitätskriterien entspricht. Wesentlich hierbei ist, dass die Iterationszyklen häufig und in kurzen Abständen erfolgen, auch wenn dadurch die einzelnen Fortschritte scheinbar klein sind.

Abbildung 2 zeigt den Prozess und die möglichen Beteiligten.

2.1 Frühphase der Konzeption

Je eher Nutzer in den Designprozess einbezogen werden, umso eher können Ideen an das Feedback von Nutzern angepasst werden. Eine Methode der Partizipation des Nutzers im Designprozess ist das Usability Testing. Umso eher damit begonnen wird, desto kostengünstiger sind Änderungen des Konzeptes.

Nach Beaudouin und Lafon (2003) werden gerade in dieser frühen Phase eher *low-fidelity* bzw. Papier-Prototypen genutzt. Die wesentlichen Vorteile solcher Art Prototypen sind, dass sie schnell und mit wenig Aufwand erzeugt werden und daher auch schnell verworfen oder geändert werden können.

Das Ergebnis sind schnelle Iterationen, wobei unterschiedlichste Designideen exploriert werden können.

Zusätzlich können diese einfachen Prototypen potenziell von allen Beteiligten erstellt werden, also auch von Personen ohne technisches Vorwissen. So können unterschiedlichste Rollen ihre Anforderungen, Ideen und Erwartungen auf gleicher Basis zum Ausdruck bringen und auch die Ideen der anderen verstehen.

Der grundlegende Nachteil solcher Prototypen ist jedoch, dass diese nur mit erheblichem Aufwand mit einer verteilten Zielgruppe bzw. remote getestet werden können.

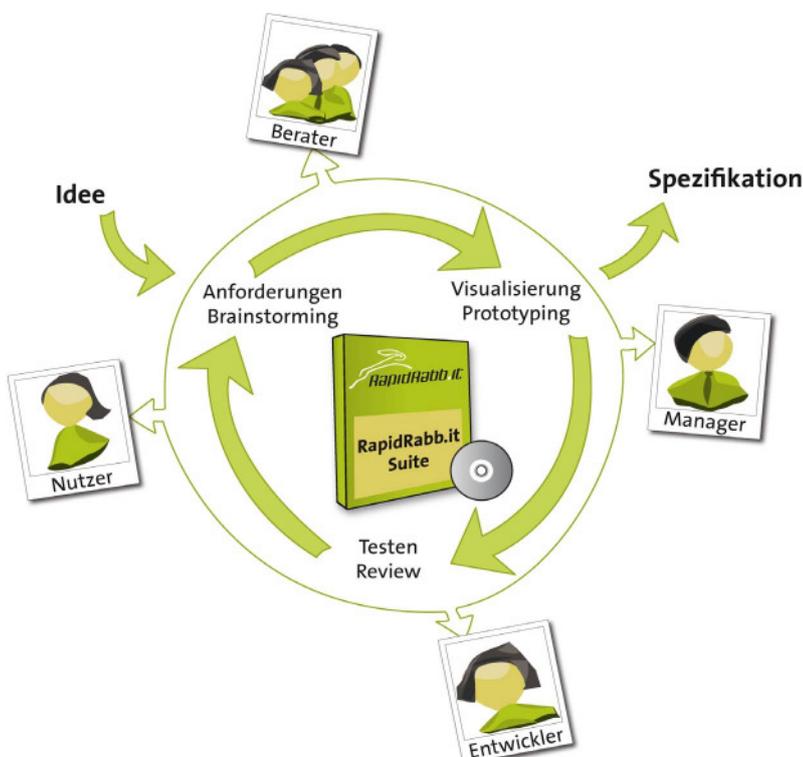


Abb. 2: Rapid Prototyping: iteratives Vorgehen mit allen Beteiligten

2.2 Remote Usability Testing

Remote Usability Tests werden durchgeführt, ohne die Probanden in ein Usability Labor einzuladen oder sich physisch anderwärtig zu treffen. Testleiter und Proband befinden sich demnach an unterschiedlichen Orten. Dies kann, dank weit verbreiteten Breitbandanschlüssen und mächtigeren Kollaborationsapplikationen, im zunehmenden Maße gelebte Praxis werden.

Um den Nutzer tatsächlich zu verstehen, beinhaltet ein Usability Test auch immer eine qualitative Komponente. Diese Komponente ist gerade in der Frühphase des Prozesses notwendig. Qualitative Daten können besonders gut mit synchronen Testverfahren gewonnen werden.

2.3 Synchrones Usability Testing

Synchrone Usability Tests sind Tests, an denen der Testleiter und meist nur ein Proband gleichzeitig teilnehmen.

Der Fokus solcher Tests liegt eher auf die Erhebung von qualitativen Daten, als quantitativen. Die Ergebnisse hängen stark von den Aussagen der Probanden und den Fragen des Testleiters ab. Die Methoden müssen flexibel genug sein, damit der Testleiter auftretende Probleme schon im Test näher explorieren kann. Aber nicht nur das Vorgehen, sondern auch die Interpretation hängt stark von den Fähigkeiten und Erfahrungen des Testleiters ab.

Synchrones Usability Testing ist gut geeignet low-fidelity Prototypen zu evaluieren, da diese Art Prototypen oft Grenzen besitzen, welche dem Probanden nur durch Kommunikation mit dem Testleiter verständlich gemacht werden können. Beispielsweise wird in der direkten Kommunikation geklärt, ob dem Prototyp Eigenschaften fehlen, weil der Prototyp zu abstrakt ist oder weil sie nicht vorgesehen sind.

3.0 Anforderungen an Synchrones Remote Usability Testing

Um also nun Nutzer frühzeitig in den Designprozess zu integrieren, stehen optimalerweise remote verfügbare low-fidelity Prototypen zur Verfügung (bspw. erzeugt mit dem RapidRabb.it Prototype Creator, siehe Gersabeck 2008), welche durch synchrones remote Usability Testing evaluiert werden. Ein allgemeiner Aufbau sollte folgende Punkte beinhalten:

- Testleiter und Proband sind via Internet verbunden, haben die gleiche Sicht auf den Prototypen, können via Audio kommunizieren und der Testleiter sieht die Aktionen des Probanden.
- Die Daten der Sitzung (Audio-kommunikation, Videomitschnitt der Aktionen, Mousetrace etc.) werden automatisch aufgezeichnet und können anschließend zur Auswertung abgespielt werden.

Weitere optionale Bestandteile erhöhen die Qualität der Sitzung:

- Informationen, Aufgaben und Hinweise können den Probanden eingeblendet werden, dies reduziert Nachfragen des Probanden.
- Der Testleiter kann Fragebögen vor und nach dem Test einbinden, welche der Proband ohne Beobachtung des Testleiters ausfüllen kann.

- Der Testleiter kann den Prototypen während der Testsitzung verändern. Dies erlaubt das flexible Reagieren auf Nutzungsproblemen und ermöglicht die freie Exploration von spontanen Lösungen.
- Der Testleiter kann während der Sitzung Kommentare, Notizen und Annotationen (etwa bezüglich Usability Problemen) hinzuzufügen.
- Der Proband kann während der Sitzung den Prototypen zu verändern. Dies sollte sich jedoch auf einfache, weniger komplexe Manipulationen beschränken (etwa markieren oder verschieben von Elementen, skizzieren von Ideen).
- Proband und Testleiter können nicht nur via Audio, sondern auch per Video verbunden sein. Dies erlaubt dem Testleiter den Kontext des Probanden besser zu verstehen und schafft Vertrauen.

4.0 Beteiligte – Internationale Probanden

Usability Tests sind unter anderem deshalb so populär, weil nur wenige Probanden ausreichen, um die meisten Usability Probleme zu entdecken. Tests mit mehr als fünf bis acht Testnutzern werden schnell unproduktiv, da zunehmend dieselben Probleme aufgedeckt werden (Dumas & Fox 2003). Das Ziel sollte sein, pro Iteration nur einige Tests durchzuführen, dafür aber die Anzahl der Iterationen zu erhöhen.

Software adressiert oft eine internationale Zielgruppe. Dies verlangt aber nach internationalen Usability Tests, da kulturelle Unterschiede häufig über die Übersetzung des Textes hinausgehende Anpassungen benötigen. Diese Anpassungen können sogar die Struktur der Applikation betreffen. Remote Usability Tests helfen hier die Kosten erheblich zu reduzieren.

5.0 Methoden

Der allgemeine beschriebene Testaufbau (Abschnitt 3) erlaubt die Durchführung folgender Methoden:

- *Thinking Aloud*: der Proband wird aufgefordert seine Gedanken (Probleme bei der Nutzung, was erwartet er wo, was fehlt ihm, ...) zu äußern. Der Testleiter kann so den Kontext des Probanden besser verstehen und darauf reagieren.
- *Wizard of Oz Prototyping*: der Prototyp kann während des Tests vom Testleiter geändert werden. Dies ermöglicht Exploration alternativer Designideen schon während der Sitzung.
- Gleichzeitiges Beobachten im Team: das Designteam beobachtet live den Test und sammelt schon während des Tests Ideen.
- Usability Probleme können leichter identifizieren und behoben werden, da qualitative Daten gesammelt werden (z.B. Fehler, Zögern oder Frustration von Probanden oder der Proband benötigt Hilfe).

- Es können formale und informale Tests durchgeführt werden.
- Testauswertung durch Highlight-video/Videoschnipsel, Klickpfadanalyse, Headmaperstellung etc.
- Interview per Fragebögen

6.0 Werkzeuge (Beispiele)

Dieser Abschnitt stellt eine Auswahl von Werkzeugen vor, mit denen die in Abschnitt 2 beschriebenen Methoden remote durchgeführt werden können.

Herkömmliche Werkzeuge beschränken sich meist nur auf einzelne Methoden, so dass nur durch eine Kombination verschiedener Werkzeuge alle Methoden angewendet werden können.

6.1 Online-Konferenzsysteme

Diese Art Werkzeuge erlauben die Freigabe von Bildschirmen (Screensharing) und der Steuerung von Maus und Tastatur über das Internet. Beispiele sind Cisco Systems WebEx, Microsoft Netmeeting oder ähnliche. Der Testleiter kann lokal den Prototyp ausführen (z.B. als HTML-Klickdummy) den Probanden dann die Steuerung über seinen Rechner überlassen und den Probanden dabei beobachten, wie dieser den Prototypen bedient.

Grundsätzlich können mit dieser Art Werkzeug sehr flexible Testsitzungen durchgeführt werden. Jedoch lassen sich einige Nachteile identifizieren: (1)

Der Proband muss eine Clientsoftware auf seinen Rechner installieren, was zwar zunehmend einfacher wird, aber dennoch bestimmte Zielgruppen ausschließen kann. (2) Diese Art Software wird teilweise durch Firewalls geblockt (etwa 5% der Probanden haben dieses Problem (Dumas, 2003)). (3) Der Prototyp kann nicht während der Sitzung verändert werden. (4) Der Testleiter kann seinen Rechner während dessen nicht für andere Aufgaben (z.B. Notizen führen) verwenden. (5) Da der Proband einen fremden Rechner über das Internet bedient, kann es einerseits zu Latenzen kommen und andererseits befindet sich der Proband dadurch nicht vollständig in seiner gewohnten Umgebung (wenn man den Rechner des Probanden als seine Umgebung betrachtet).

6.2 Screencapture-Programme

Mit dieser Art Programm kann der Bildschirm virtuell abgefilmt werden (z.B. mit TechSmith Camtasia). Dadurch kann eine Testsitzung aufgezeichnet werden. Es sollte jedoch darauf geachtet werden, dass die Software eine leistungsfähige Hardware voraussetzt, vor allem, wenn parallel ein Online-Konferenzsystem und weitere Software betrieben wird.

6.3 Videokonferenzsysteme

Um mit den Probanden per Audio und Video verbunden zu sein, können Videokonferenzsysteme (wie etwa Skype) eingesetzt werden. Ein Nachteil ist, dass der Proband ebenfalls eine kompa-



Abb. 3: Usability Testing mit RapidRabb.it. Links: mobiles Endgerät des Probanden. Rechts: Sicht des Moderators auf denselben Prototypen.

tible Software installiert haben muss.

6.4 Präsentationsprogramme

Präsentationsprogramme (wie Microsoft Powerpoint etc.) können verwendet werden, um den Probanden vorab mit Kontextinformationen zu versorgen.

6.5 Videoschnittprogramme

Mit Hilfe dieser Programme (z.B. Microsoft Movie Maker, Adobe Premiere oder andere) können aus den Mitschnitten Highlightvideos erstellt werden, jedoch ist die Erstellung nicht ganz unaufwändig.

6.6 Online Fragebögen

Fragebögen (z.B. Google Form, SurveyMonkey) kann der Proband vor und nach der Testsitzung ausfüllen. Dies macht Sinn, um weitere Kontextinformationen (Alter, Beruf, Affinität etc.) vom Probanden zu erfragen. Jedoch muss die Zuordnung zur Testsitzung manuell geschehen.

6.7 RapidRabb.it Suite

RapidRabb.it ist eine echtzeit-kollaborative Onlineplattform, auf der Einzelne oder Teams interaktive Web- und Softwareprototypen schnell und einfach erstellen, diskutieren, testen und optimieren können. Die web-basierte Software ermöglicht es, ortsunabhängig alle Beteiligte in die Web- bzw. Softwareentwicklung einzubeziehen und beschleunigt so die Kommunikation und damit den Prozess erheblich. Dank der Verwendung von Webtechnologien entstehen keine technischen Hürden beim Probanden (keine Firewallprobleme, keine Installation), es wird lediglich ein Browser benötigt.

Die web-basierte RapidRabb.it Suite besteht aus zwei integrierte Modulen: den Prototype Creator und den Remote Usability Tester, welche das Rapid Pro-

totyping optimal unterstützen, ohne dass ein „Tool“-Zoo von Nöten ist. Mit dem Prototype Creator können sehr einfach und schnell *low-fidelity* Prototypen erstellt werden, welche mit dem Remote Usability Tester iterativ evaluiert werden können (Beispiel siehe Abbildung 3).

Der Remote Usability Tester ermöglicht es, einen Prototyp unabhängig vom Aufenthaltsort des Probanden zu testen. Dabei kann durch aktuelle Screencastingtechnik der Testleiter den Bildschirm und die Interaktionen der Testperson beobachten und durch Videotelefonie ein Testinterview durchführen, genau so, als ob der Proband im selben Raum sitzen würde. Im Gegensatz zu existierenden Konferenzprodukten bleibt es den Teilnehmern erspart, eine zusätzliche Software zu installieren. Zudem wird dem Testleiter spezielle Funktionen zur Aufnahme, Bearbeitung und Wiedergabe der Testsitzung zur Verfügung gestellt. Dies umfasst auch die Möglichkeit spezielle Situationen für die spätere Auswertung zu markieren und mit kurzen Kommentaren zu versehen. Der Remote Usability Tester, unterstützt damit Methoden wie *Wizard-of-Oz* Prototyping und *Think Aloud* Tests. Screensharing, Screenshot, Audio- und Videomitschnitt geschehen automatisch.

7.0 Fazit

Auch in der Frühphase der Konzeption ist es dank zunehmend besserer Software möglich, die Nutzer per synchrone remote Usability Tests partizipieren zu lassen. Herkömmliche Werkzeuge können zwar prinzipiell dafür eingesetzt werden, bieten jedoch bis dato keine integrierte Lösung. Dies kann nicht nur zu Kompatibilitätsproblemen und aufwändigen Setup führen,

sondern auch bestimmte Techniken und Methoden (z.B. *Wizard-of-Oz* Prototyping, Manipulation des Prototypen durch den Probanden, Annotationen während der Testsitzung durch den Testleiter) nur unvollkommen unterstützt.

RapidRabb.it bietet einen neuen über den gesamten Konzeptionsprozess integrierten Ansatz: Prototyping, Evaluation, Auswertung und Anpassung der Prototypen mit einer web-basierten Suite, welche ohne Installation und aufwändigen Setup genutzt werden kann.

8.0 Literatur

Beaudouin-Lafon, M; Mackay, W (2003) Prototyping Tools and Techniques. In: Jacko, J; Sears, A (2003): The Human-Computer Interaction Handbook. Lawrence Erlbaum Associates. S. 1017-1039.

Burger, S; Burmester, M; Selzer, A (2008): Formatives Remote Usability Testing. I-com Heft 1/2008. S. 47-49.

Dumas, J. (2003): Usability evaluation from your desktop. Association for Information Systems (AIS) SIGCHI Newsletter 2, S. 7-8

Dumas, J; Fox, J (2003): Usability Testing: Current Practice and Future Directions. In: Jacko, J; Sears, A (2003): The Human-Computer Interaction Handbook. Lawrence Erlbaum Associates. S. 1129-1149.

Gersabeck, V; Golega, S; Truppel, T (2008): Web-based Paper Prototyping mit RapidRabb.it – Ein Ansatz zum kollaborativen Prototyping. Usability Professionals 2008. S. 157-159

Houck-Whitaker, J: Remote Testing versus Lab Testing. <http://boltspeters.com/articles/versus.html> (Zugriff: 2009-06-02).

Lorenzen-Schmidt, O; Nufer, S (2008): From a Distance: Usability Testing aus der Ferne. I-com Heft 1/2008. S. 44-46.

Sieker Andreasen, M; Villemann Nielsen, H; Ormholt Schroder, S; Stage, J (2007): What Happened to Remote Usability Testing? An Empirical Study of Three Methods. CHI 2007 Proceedings.