

# Erstellen und Nutzen von Prototyping in der Softwareentwicklung: Vor- und Nachteile verschiedener Wiedergabetreue (Fidelity)

## Erfahrungen von UseTree und Ergebnisse einer Literaturrecherche

Charlotte Spang, Ronny Reckin

Fidelity (engl.: Wiedergabetreue) ist das Maß, das die Nähe eines (Design-) Prototyps zum tatsächlichen Endprodukt beschreibt, bezogen auf grafische Ausgestaltung und/oder Interaktivität. Dieser Beitrag beleuchtet die Erfahrungen von UseTree mit der Erstellung von Prototypen unterschiedlichster Wiedergabetreue. Diese werden ergänzt durch eine kurze Darstellung der in der Literatur diskutierten Vor- und Nachteile der unterschiedlichen Grade beim Erstellen und Nutzen von Prototypen.

Als Anschauungsbeispiel dient eine Eieruhr-App. Mit einer solchen App kann die optimale Kochzeit für Eier abhängig von deren Größe und des gewünschten Härtegrads bestimmt werden. Die App erinnert nach Ablauf dieser Kochzeit daran, die Eier aus dem Wasser zu nehmen. UseTree erstellte drei Prototypen einer solchen App mit jeweils unterschiedlicher Fidelity. Die hierbei gesammelten Erfahrungen illustrieren den folgenden Vergleich der verschiedenen Grade der Wiedergabetreue von Prototypen hinsichtlich Nutzen, Aufwand und Ressourceneinsatz.

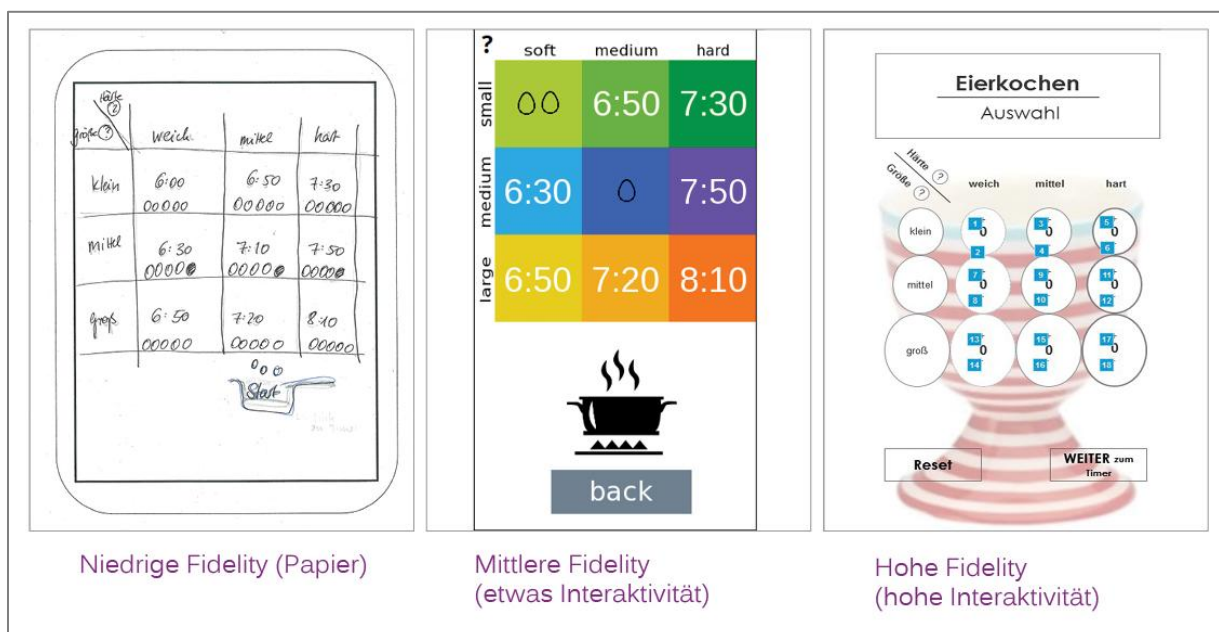


Abbildung 1: Drei Grade der Wiedergabetreue im Vergleich

### Niedrige Wiedergabetreue (Low Fidelity)

Der Prototyp mit der niedrigsten Fidelity der Applikation (siehe [Abbildung 2](#)) wurde mit Papier und Bleistift innerhalb einer halben Stunde erstellt, spezielle Vorkenntnisse waren nicht erforderlich. Veränderungen in Aussehen und Struktur der Anwendung sind durch den gewohnten Umgang mit Papier, Bleistift und Radiergummi leicht möglich. Ergebnisse werden also schnell erzielt, es gibt keine frustrierenden Erlebnisse.

Die Versionen der Prototypen können als hilfreiche Kommunikationsbasis innerhalb eines Teams dienen. Die eigene Hemmung „Ich kann nicht zeichnen“ ist oft das einzige Hindernis.

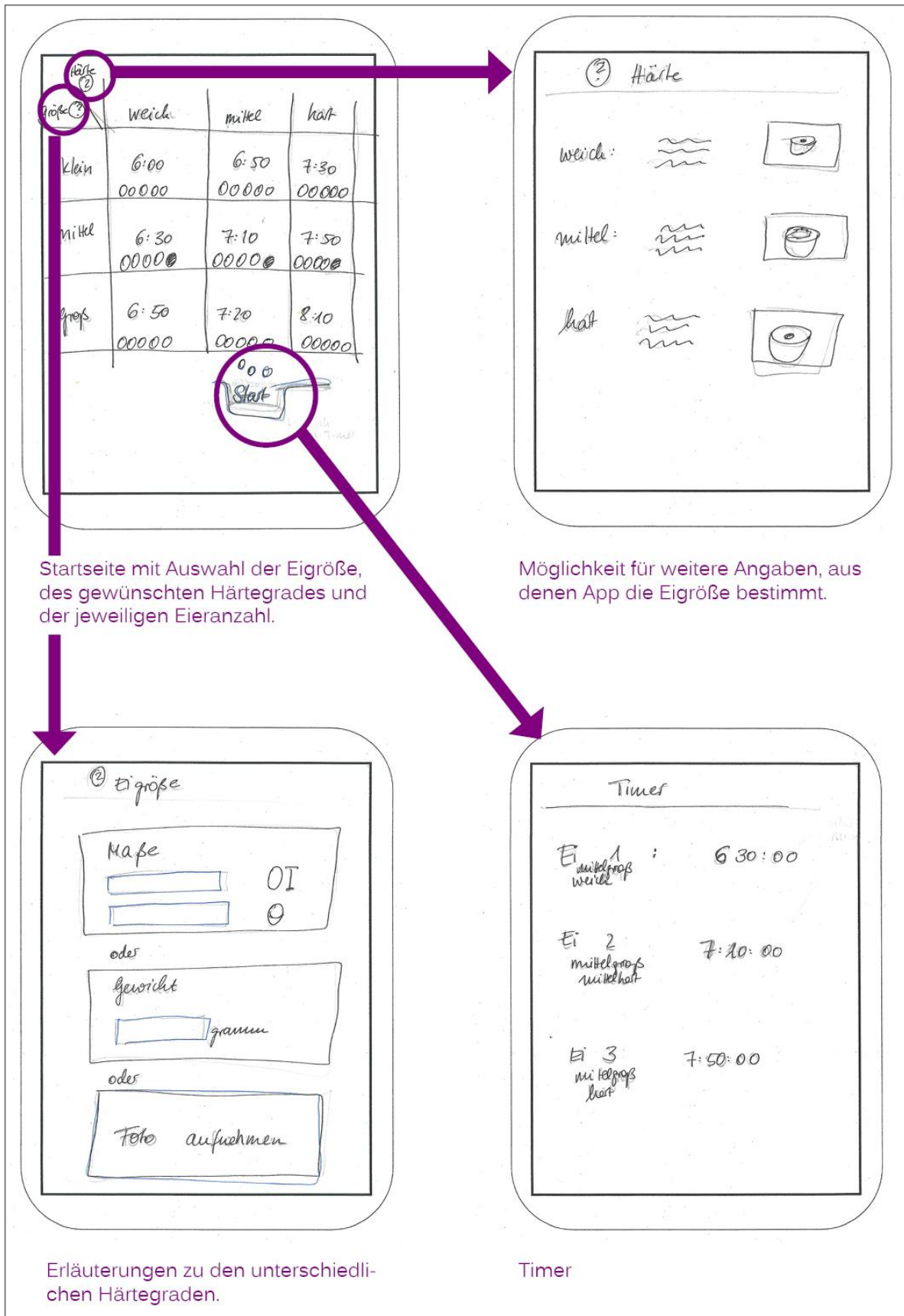


Abbildung 2: Papierprototyp der Eieruhr App (niedrige Fidelity)

Wie an unserem Beispiel in **Abbildung 2** zu sehen ist, sind keine großen künstlerischen Fähigkeiten vonnöten, um ein erstes Konzept zu skizzieren. Ohne größere Investitionen und Lernaufwand gibt es schnell eine gemeinsame Kommunikationsbasis, an der dann weiter

gearbeitet werden kann. Die grafische Gestaltung wird zunächst außen vor gelassen. Die Funktionalitäten unseres Beispiels ließen sich allesamt sehr einfach auf Papier skizzieren. Sobald die Funktionen allerdings komplexer werden, ist mehr Bastelaufwand und Kreativität erforderlich. Etwa ein Dropdown-Menü ist auf Papier nicht ganz so einfach umsetzbar. Andere Funktionen lassen sich auf Papier überhaupt nicht darstellen. Dies betrifft insbesondere Multitouchgesten und Swipen. Um das System mit anderen Personen durchzuspielen (zu testen), muss ein Testleiter das System „spielen“. Dies bedeutet, beispielsweise bei Druck auf einen gezeichneten Button zur nächsten Interaktion umzublättern, Geräusche zu machen usw. Die meisten Testpersonen interagieren mit einem Papierprototyp ohne sich eingeschränkt zu fühlen. Die Probanden sind viel toleranter gegenüber Fehlern des Systems oder noch nicht durchdachten Funktionalitäten und Verlinkungen. Hierdurch können das grobe Konzept und Anwendungsszenarien überprüft werden, ohne dass zu viel Aufmerksamkeit durch die Bewertung von Feinheiten verloren geht. Durch die Ferne des Papierprototyps von einem real funktionierenden System sind manche Verhaltensweisen und Aussagen jedoch nur eingeschränkt interpretierbar. Zudem ist es schwierig, mit einem Stapel Papier ein Testszenario „im Feld“ zu schaffen, also eine reale Situation bei der Interaktion mit einem System, wie beispielsweise Zeitungslesen auf einem Tablet morgens am Frühstückstisch. Manche Testpersonen und auch Angehörige anderer Unternehmensabteilungen stehen der einfachen Papierlösung zudem skeptisch gegenüber.

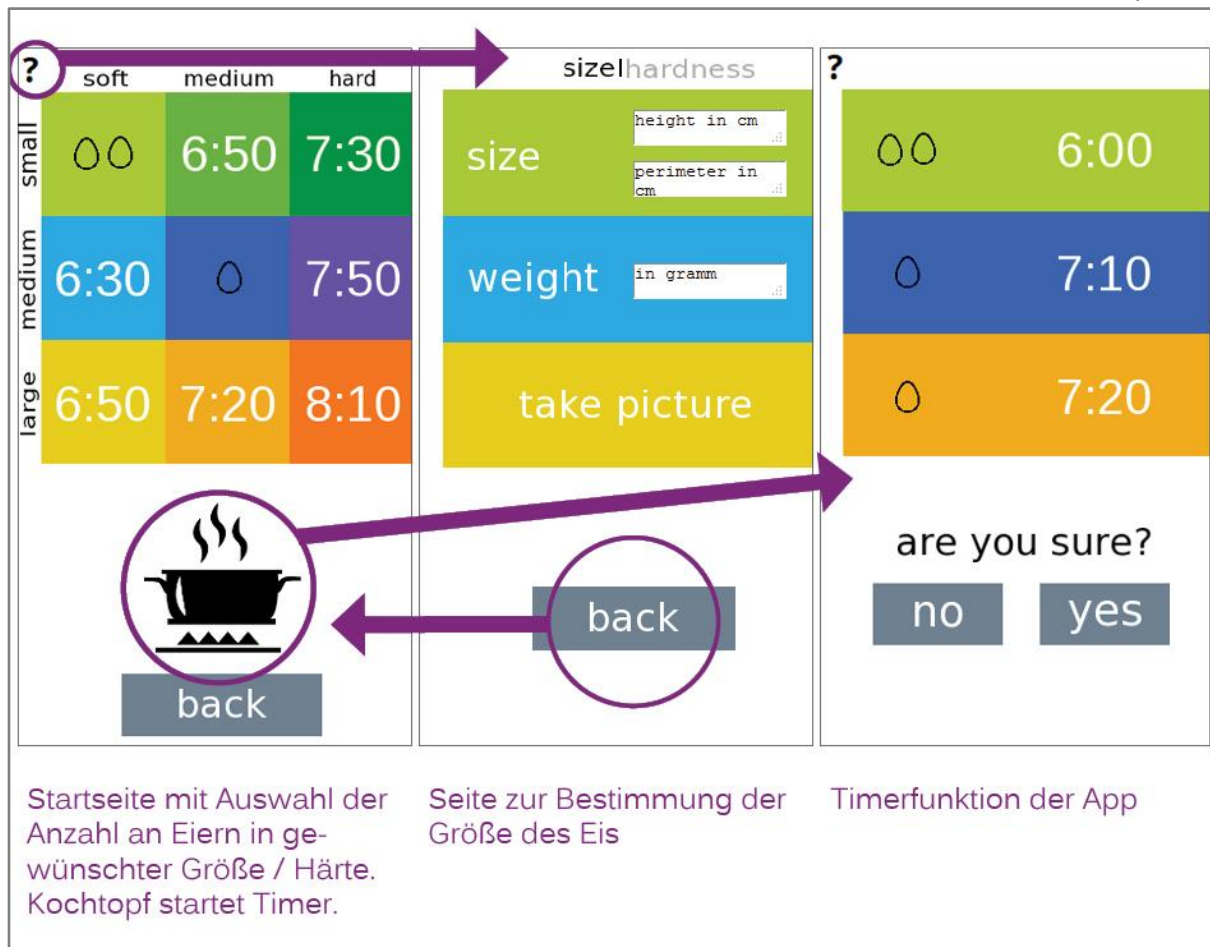
Um die Fidelity eines Papierprototyps um einfache Interaktionen und Verlinkungen zu erweitern, können geeignete Tools (beispielsweise [POP](#) oder [Marvel](#)) eingesetzt werden. Papierscreens werden hiermit fotografiert und Interaktionen digital ergänzt.

Papierprototypen sind häufig der erste Schritt bei der App-Entwicklung und helfen beim Erarbeiten einer Struktur. Auch für die Beispiel-App Eieruhr diente der Prototyp mit niedriger Fidelity als Basis für die Weiterentwicklung zu höheren Fidelity-Graden.

### Mittlere Wiedergabetreue (Medium Fidelity)

Als zweites wurde ein Prototyp mittlerer Fidelity erstellt. Diese Prototypen werden meist digital erstellt und zeigen bereits die grundlegende Struktur der App. Zudem besitzen sie schon einige Funktionalitäten die von Testpersonen ausprobiert werden können, auch wenn weder der gesamte Funktionsumfang noch die grafische Ausgestaltung abgedeckt sind. Für das Erstellen digitaler Prototypen mittlerer Fidelity gibt es verschiedene Programme, die bereits vorgefertigte Icons, Interaktionsflächen, Buttons, Textfelder usw. anbieten. Im Gegensatz zu Papierprototypen müssen Buttons, Icons und sonstige interaktive Elemente damit nicht mehr selbst gezeichnet werden, sondern können aus einer Menüleiste auf die Arbeitsfläche gezogen werden. Dadurch kann die „Ich kann nicht zeichnen“-Hemmung minimiert werden. Funktionen können klickbar miteinander verlinkt werden, was einen besseren Eindruck vom interaktiven System ermöglicht.

Mit ca. 8 Stunden dauerte das Erstellen dieses digitalen Prototyps etwas länger als die des Papierprototyps, mit zunehmender Erfahrung mit der Software ist jedoch eine erhebliche Reduzierung des Zeitaufwands zu erwarten. Die grundlegenden Funktionen der Programme sind meist einfach und schnell zu erlernen und erfordern keine besonderen Vorkenntnisse. Voraussetzung für eine Einarbeitung sind normale Computerkenntnisse und die Bereitschaft, sich mit dem Programm auseinanderzusetzen.



**Abbildung 3: Ausgewählte Funktionen des mit pidoco erstellten Prototypes der Eieruhr App (mittlere Fidelity)**

Für allererste Ideenskizzen dürfte diese Erstellung etwas zu aufwendig sein, aber für die Konkretisierung und Kommunikation von Ideen und deren weitere schrittweise Ausarbeitung im Team eignen sich solche digitalen Prototypen mittlerer Fidelity sehr gut. Von besonderem Vorteil ist, dass Veränderungen in einem solchen Prototypen viel leichter vorzunehmen sind als bei der Papierversion. Auch das Testen von Konzept und Funktionen kann detaillierter erfolgen. Testpersonen können bereits Funktionalitäten ausprobieren, welche im späteren System vorhanden sein werden. Dabei werden sie weder von Papierstapeln abgelenkt, noch ist ein zusätzlicher Testleiter zur Darstellung des Systems vonnöten. Dadurch kann der Eindruck der Testpersonen in mancher Hinsicht aussagekräftiger sein als bei einem Papierprototyp.

Geeignete Programme sind häufig browserbasiert oder einfach zu installieren und können mit relativ geringen Kosten professionell genutzt werden (ab ca. 10 € pro Monat). Beispiele für solche Software sind [balsamiq](#), [lumzy](#), [wireframe.cc](#) oder das von uns ausprobierte [pidoco](#). Typische und beabsichtigte Eigenschaft dieser Programme ist, dass sich klickbare Prototypen einfach und schnell erstellen lassen, sie hierbei viele Möglichkeiten zur Modellierung von Funktionalitäten bieten, aber wenig Mittel zu seiner ästhetischen Gestaltung. Diese reduzierte Funktionalität erleichtert einerseits das Erlernen und macht die Programme leicht verständlich. Andererseits können diese Einschränkungen aber auch Komplikationen bei der Erstellung unterschiedlicher Workflows mit sich bringen. Beispielsweise ist das Einführen von Bedingungen, um verschiedene Workflow-Varianten zu testen, nicht möglich. Für jede Version muss ein neuer Prototyp erstellt werden. Die

allermeisten Funktionen eines Softwarekonzepts lassen sich mit einem solchen Prototyp mittlerer Fidelity jedoch sehr gut erstellen und vorführen. Dem Entwicklerteam kann damit erleichtert werden, sich das fertige System hinsichtlich seiner funktionalen Ausgestaltung besser vorzustellen, für eine weitere Verbesserung der Zusammenarbeit ermöglichen einige Programme einfaches Teilen und Kommentieren der digitalen Prototypen. Auch für die Präsentation der erarbeiteten Konzepte sind solche funktionalen, digitalen Prototypen bei vielen Zielgruppen eher anerkannt als Papierprototypen.

### Hohe Wiedergabetreue (High Fidelity)

Der höhere Grad an Wirklichkeitstreue erhöht den Aufwand für das Erstellen von Prototypen mit hoher Fidelity wesentlich. Es ist mehr Zeit und etwas teurere, komplexere Software erforderlich. Die notwendige Prototyping-Software mit dem großen Funktionsumfang erfordert auch eine längere Einarbeitungszeit. Trotz logischer und guter Strukturierung kann der Einstieg aufgrund der Komplexität und Funktionstiefe am Anfang etwas frustrierend sein. Hilfreich sind jedoch die von den Softwareherstellern häufig angebotenen Tutorials und Foren. Sobald die erste Hürde genommen ist, macht die Arbeit mit dem Programm Spaß, da sich wirklich fast alles wunschgemäß abbilden lässt. Bei der Beispiel-App war das ein Grund dafür, dass der Prototyp mit seiner hohen Fidelity schon fast wie eine richtige App anfühlt.

Für die Eieruhr-App wurden ca. 40 Arbeitsstunden mit der spezialisierten Software [Axure RP](#) aufgewendet, die sich nach Einarbeitung vermutlich auf ca. 30 Stunden reduzieren lassen. Weitere passende Softwareanwendungen sind beispielsweise [justinmind](#), [proto.io](#) oder [Origami](#) (freeware). Das Ausprobieren solcher Programme ist problemlos möglich, weil üblicherweise Testversionen oder Accounts mit beschränktem Funktionsumfang kostenfrei zur Verfügung gestellt werden. Großer Vorteil dieser Anwendungen ist der hohe Funktionsumfang und die Grafikfähigkeiten, die eine Gestaltung und Detaillierung bis hin zur realistischen Simulation von Webseiten, Apps oder Programmen ermöglichen. Es lassen sich beispielsweise dynamische Inhalte, Variablen und Bedingungslogiken, Animationen, Touch-Interaktionen oder Drag-and-Drop mit originalen Grafikdateien oder User Interface-Elementen (z.B. Buttons und Icons aus iOS oder Android) erstellen. Die erstellten Prototypen lassen sich dann auf dem jeweiligen Endgerät als Simulation testen, mit realistischer Interaktion ohne spürbaren Unterschied zu einer ausprogrammierten Version. Die Entscheidung, statt schneller und eher konzeptueller Arbeit auf detaillierteres Prototyping mit hoher Fidelity zu setzen, wird beeinflusst von der verfügbaren Zeit sowie insbesondere von den jeweils verfolgten Zielen und der gerade aktuellen Entwicklungs- und Designphase. Durch den hohen Realitätsgrad lassen sich, vorzugsweise in späteren Entwicklungsphasen, Details und Feinheiten testen. Prototypen mit hoher Wiedergabetreue können beeindrucken und haben eine starke Überzeugungskraft, z.B. auf Entscheider und Unternehmen. Der - im Vergleich zu Prototypen mittlerer Fidelity - große Aufwand sollte eine bewusste Entscheidung sein, z.B. wenn es notwendig ist, mehr geplante Details, Funktionalitäten oder sogar das angedachte Layout zu zeigen. In frühen Phasen mit ersten Entwürfen und dem Erarbeiten des Interaktionsdesigns sind Prototypen geringerer und mittlerer Fidelity meist wesentlich sinnvoller und selbst in späten Entwicklungsphasen und als Vorlage für die Entwickler für die Programmierung ist die wirklichkeitsgetreue Abbildung sämtlicher Details und Funktionalitäten oft nicht notwendig. Die hierfür zusätzlich benötigten Ressourcen könnten ggf. anderweitig mehr Nutzen bringen.

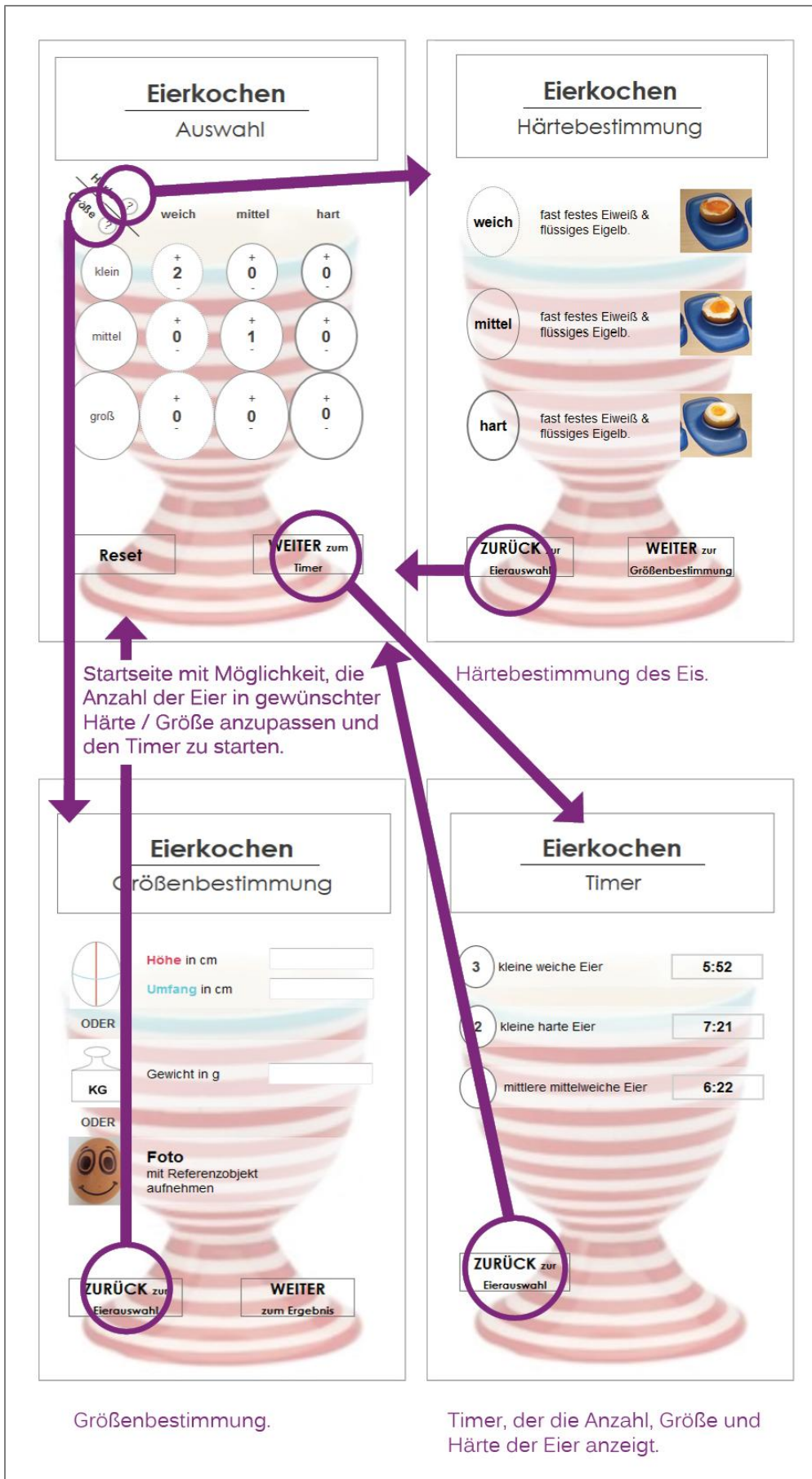


Abbildung 4: Mit Axure RP erstellter Prototyp der Eieruhr App (hohe Fidelity)

**Tabelle 1: Subjektive Bewertung nach eigener Erfahrungen bei der Erstellung von Prototypen mit verschiedener Fidelity**

| <b>Fidelity Medium</b>  | <b>Zeit-aufwand</b> | <b>Material / Vorbereitung</b>  | <b>Vorkenntnisse</b>   | <b>Frust-faktor</b> | <b>Spaß-faktor</b> | <b>Vorteile</b>   | <b>Nachteile</b>  |
|---|---------------------|---|--|---------------------|--------------------|---|---|
| <i>niedrig</i><br>Papier  | 0,5 h               | * Papier & Stift  | * Keine (nur Mut)  | *                   | ***                | + sehr schnell einsetzbar<br>+ keine Investitionen notwendig<br>+ kein Lernaufwand<br>+ schnell veränderbar                                   | - Hemmungen, einfach loszuzeichnen, "ich kann nicht zeichnen"<br>- manche Sachen nicht oder nur schwer vorstellbar und mit hohem Einsatz umsetzbar (z. B. Interaktionen, Feedback des Systems)                    |
| <i>mittel</i><br>z.B. balsamiq, lumzy, wireframe.cc oder pidoco | 8 h                 | ** 30 Tage Testversion kann online abgerufen werden (sonst ab 10 € / Monat) | *** Grundsätzliches Programmverständnis  | **                  | *****              | + Angst vor dem Zeichnen wird überwunden<br>+ spielerischer Ansatz, herumprobieren<br>+ relativ geringer Lernaufwand für technikaffine Nutzer | - Es fehlen Funktionalitäten im Tool, einiges nicht umsetzbar<br>- ein Prototyp kann im Testing nur genau einen bestimmten Workflow abbilden<br>- sehr wenig Möglichkeiten, Farbgebung / Schriftarten zu arbeiten |
| <i>hoch</i><br>z. B. justinmind, proto.io, Origami, Axure       | 40 h                | ***** Programm herunterladen (30 Tage kostenlos testen / dann 289 \$)       | ***** Programm lernen, Tutorials ansehen, Funktionen herausfinden, Logik verstehen | ****                | *****              | + optisch ansprechende Ergebnisse<br>+ fast alle Funktionen sind möglich<br>+ fühlt sich wirklich wie funktionierende App an                  | - sehr hoher Lernaufwand<br>- stellenweise frustrierend   |

**Legende: \*kein(e), \*\* wenig/ niedrig, \*\*\* mittelmäßig, \*\*\*\*viel/ hoch, \*\*\*\*\* sehr viel/ sehr hoch**

## Vor- und Nachteile verschiedener Fidelity-Grade

Ein Gesamtüberblick über die Erfahrungen beim Erstellen der drei Prototypen mit unterschiedlichem Fidelity-Grad wird in [Tabelle 1](#) dargestellt. Ergänzend lassen sich bei der Diskussion von Vor- und Nachteilen in Fachliteratur und in Blogartikeln Argumente für den Nutzen der verschiedenen Prototypen finden. Neben den Erkenntnissen, die allein durch das Erstellen gewonnen werden, spielt auch die Anwendung der Prototypen nach Fertigstellung hierfür eine große Rolle, z.B. das Testen mit Nutzern und zur Kommunikation im Team und mit Kunden.

Die recherchierten Beiträge in der Literatur unterscheiden nicht so feingranular zwischen den verschiedenen Graden der Wiedergabetreue. Vielmehr wird der Kontrast zwischen Prototypen hoher und niedriger Fidelity herausgearbeitet. Die Expertenmeinungen bestätigen die eigenen Erfahrungen im Großen und Ganzen. Auch in der Literatur wurde berichtet, dass Papierprototypen sehr schnell erstellt sind, sie keine finanziellen Investitionen erfordern und nur geringen Zeitaufwand. Zudem kann in ersten Tests der Usability und User Experience (UUX) mit ihnen schon sehr brauchbares Feedback gesammelt werden. Beispielsweise trauen sich die Probanden, einfach selbst zum Stift zu greifen und Vorschläge für eine Umgestaltung zu machen. Es stehen die Funktionalitäten der Applikation im Vordergrund, das Design wird erst einmal vernachlässigt. Die Funktionalitäten in einem Nutzertest zu simulieren kann manchmal allerdings zu „Herumgewurstel“ mit Papier führen und bedarf auch guter Vorbereitung und ein bisschen Übung. Manche der Experten bezweifelten auch die Eignung des Papierprototyps für Nutzertests und dessen Tauglichkeit zum Fehlerfinden. Ein Prototyp mit hoher Fidelity hingegen ist auf jeden Fall eindrucksvoll und kann zur späteren Umsetzung ohne umfangreiche, zusätzliche Erläuterungen an die Entwickler übergeben werden. In einem Nutzertest wird kaum Phantasie seitens der Probanden benötigt. Der Nutzer kann zumeist frei damit interagieren und die Applikation explorieren. Dafür ist die Erstellung sehr viel aufwändiger und im Nutzertest bestehen oft Hemmungen, ein scheinbar so ausgereiftes und aufwändiges Produkt zu kritisieren. Eine Zusammenstellung der Vor- und Nachteile der beiden Prototypen aus der Literatur finden Sie in [Tabelle 2](#).

**Tabelle 2: Vor- und Nachteile von Prototypen mit hoher und niedriger Fidelity auf Basis von Literaturrecherche**

|                 | niedrige Fidelity   | hohe Fidelity   |
|-----------------|---|---|
| <b>Vorteile</b> | <ul style="list-style-type: none"> <li>+ Berührungsangst der Probanden sinkt.</li> <li>+ Probanden greifen selbst zum Stift.</li> <li>+ Probanden trauen sich zu kritisieren.</li> <li>+ Wenig Anstrengungen und Lernaufwand investiert.</li> <li>+ Schnell.</li> <li>+ Sehr brauchbares Feedback insbesondere für Anfangsphase eines Systems.</li> <li>+ Gute Gesprächsbasis, weil alle sehen, wovon gesprochen wird.</li> </ul> | <ul style="list-style-type: none"> <li>+ Keine „Zettelwirtschaft“.</li> <li>+ Auch Design kann schon berücksichtigt werden.</li> <li>+ Für Präsentationen sehr eindrucksvoll.</li> <li>+ (Fast) volle Funktionalität.</li> <li>+ Auch Interaktionen möglich.</li> <li>+ Inzwischen durch Prototyping-Programme nicht mehr so aufwändig wie früher.</li> <li>+ Auch für Laien sofort verständlich.</li> <li>+ Hohe Glaubhaftigkeit.</li> </ul> |



|                         |  |   |
|-------------------------|--|---|
|                         | <ul style="list-style-type: none"> <li>+ Verbesserungen geschehen, bevor aufwändig implementiert wird.</li> <li>+ Es können mehrere Konzepte leicht miteinander verglichen werden.</li> <li>+ Nützlich, um Marktanforderungen herauszufinden.</li> <li>+ Kann als proof-of-concept dienen.</li> <li>+ Kann in Gruppenarbeit generiert werden.</li> <li>+ Jeder kann mitmachen, kein Vorwissen erforderlich.</li> <li>+ Design / Gestaltung werden zunächst außen vor gelassen, es geht um Funktionalität.</li> <li>+ keine Einschränkung in den Design-Möglichkeiten durch beschränkte Auswahl an Interaktionen im Prototyping Tool.</li> </ul>  | <ul style="list-style-type: none"> <li>+ Nutzer hat volle Handlungsmacht.</li> <li>+ Für Marketing und Sales viel eindrucksvoller.</li> </ul>   |
| <p><b>Nachteile</b></p> | <ul style="list-style-type: none"> <li>- Interaktion ist beschränkt (z.B. kein mouseover darstellbar).</li> <li>- Bearbeitungszeiten aus Nutzertest mit Papierprototyp sind nicht aussagekräftig.</li> <li>- „Herumgewurstel“ mit Papier.</li> <li>- Sieht u.U. unprofessionell aus.</li> <li>- Phantasie seitens Probanden benötigt.</li> <li>- Fehler werden nur bedingt gefunden.</li> <li>- Für Usability-Tests aufgrund der geringen Übertragbarkeit auf funktionierendes System nur bedingt geeignet.</li> <li>- Taugt nur bedingt als Vorlage für die Programmierung.</li> <li>- Für Veränderungen muss radiert werden, es können keine Funktionen „herumgeschoben“ werden.</li> <li>- Extra Versuchsleiter notwendig, der das System simuliert.</li> <li>- Beeinflusst von demjenigen, der das System spielt.</li> </ul> | <ul style="list-style-type: none"> <li>- Erstellung dauert meist mehrere Tage.</li> <li>- Hohe Hürde, noch einmal von vorne zu beginnen.</li> <li>- Kann im Testprozess kaum noch modifiziert werden.</li> <li>- Aussehen tritt im Test oft ungewollt in den Vordergrund.</li> <li>- Nutzer können nicht selbst am Prototyp verbessern.</li> <li>- Für Nutzertest müssen fiktive Daten generiert werden, um realitätsnah testen zu können</li> <li>- Hohes Know-How für die Erstellung notwendig.</li> <li>- Hoher Zeit- und Ressourceneinsatz bei hohem Risiko.</li> <li>- Lohnt sich nicht nur um Anforderungen zu sammeln.</li> <li>- Kann nicht in der Gruppe erstellt werden.</li> </ul> |

## Fazit

Aus unserer Sicht bieten Prototypen mit niedriger Fidelity aufgrund ihres guten Kosten-Nutzen-Verhältnisses bereits viele Möglichkeiten für die meisten Anwendungsfälle mit denen sich kleine und mittlere Unternehmen am Entwicklungsbeginn konfrontiert sehen: Sie können sehr schnell erstellt werden, dienen als Kommunikationsbasis im Team und können sogar für Nutzertests verwendet werden.

Im weiteren Verlauf der Produktentwicklung kann sich dann das Erlernen eines Prototypenprogramms lohnen, mit dem sich einfache klickbare Prototypen mit mittlerer Fidelity erstellen lassen. Diese Prototypen sind oftmals bei Entscheidern im Unternehmen anerkannter als Papierprototypen. Der relativ geringe Lernaufwand lohnt sich, wenn mehr Interaktionen abgebildet werden sollen, als es ein Prototyp mit niedriger Wiedergabetreue erlauben würde. Auch diese Prototypen schaffen ein gemeinsames Verständnis im Team. Die digitalen Prototypenprogramme erlauben zudem meist auch Kommentare und gemeinsame Arbeit am selben Objekt. Buttons können in den weiteren Iterationen herumgeschoben werden, ohne radieren zu müssen. Sehr komplexe Benutzeroberflächen mit tiefen Hierarchien können allerdings aufgrund der Einfachheit der Prototyping-Programme mit mittlerer Fidelity nicht zufriedenstellend abgebildet werden. Zudem bieten die Programme meist nur sehr eingeschränkte Möglichkeiten zur realistischen grafischen Ausgestaltung.

Dies können Prototypen mit sehr hoher Fidelity leisten und damit als genaue Vorlage für Entwickler dienen. Erläuterungsaufwand und Kosten durch Missverständnisse werden dadurch minimiert. Das Erstellen von High Fidelity Prototypen lohnt sich aus unserer Sicht jedoch nicht für die Usability-bezogenen Aufgaben (wie etwa Usability-Tests, heuristische Evaluationen), mit denen sich kleine und mittlere Unternehmen erfahrungsgemäß konfrontiert sehen. Die Investitionen für das Erlernen des Programmes und die aufwändige Erstellung von sehr detailgetreuen Prototypen zahlen sich deshalb meist nicht aus.

**Grundsätzlich gilt: ein Prototyp sollte genau so detailgetreu sein, wie es zum Kommunizieren, Vorführen oder Testen erforderlich ist und lediglich die dafür benötigten Funktionalitäten abbilden.**

## Quellen

- Cerejo, L. (Juni 2006): Design Better and Faster With Rapid Prototyping.  
<http://www.smashingmagazine.com/2010/06/16/design-better-faster-with-rapid-prototyping/> (Abrufdatum: 13.04.2015).
- Doleatto, K. (Dezember 2014). High Fidelity vs. Low Fidelity Prototyping in Web Design and App Development. <http://www.atlargeinc.com/insights/high-Fidelity-vs-low-Fidelity-prototyping-web-design-and-app-development> (Abrufdatum: 13.04.2015).
- Farnum, C. (Juli 2002): What You Should Know About Prototypes for User Testing.  
<http://boxesandarrows.com/what-you-should-know-about-prototypes-for-user-testing/> (Abrufdatum: 13.04.2015).
- Rudd, J., Stern, K., & Isensee, S. (1996). Low vs. high-Fidelity prototyping debate. *interactions*, 3(1), 76-85. Online:  
<http://www.it.uu.se/edu/course/homepage/avint/vt09/prototype/lowfi.pdf> (Abrufdatum: 13.04.2015).
- Waloszek, G., & Carignan, J.(July 2004). Using Prototypes. abgerufen von:  
[http://www.sapdesignguild.org/community/design/print\\_prototypes.asp](http://www.sapdesignguild.org/community/design/print_prototypes.asp) (Abrufdatum: 13.04.2015).