

Frage 96	2 richtige Antworten	LZ 2.5
Welche zwei der folgenden Aussagen über User-Interface-Elemente sind korrekt?		
A	Die Top-Level-Navigation einer Website ist ein Beispiel für ein User-Interface-Element.	
B	User-Interface-Elemente sind die Komponenten einer Benutzungsschnittstelle, die interaktiv sind.	
C	Das Standardisieren von User-Interface-Elementen über interaktive Systeme hinweg fördert die Erlernbarkeit für die Benutzer.	
D	User-Interface-Elemente werden in Form von Gestaltungsregeln definiert.	
E	Ein Tooltip ist ein Beispiel für ein User-Interface-Element.	
F	User-Interface-Elemente sind Lösungen für häufig auftretende Gestaltungsprobleme.	

## 7.2 Gestaltungslösungen finden und spezifizieren

Lernziele in diesem Abschnitt (die Nummerierung entspricht den Lernziel-Nummern im CPUX-F-Curriculum):

- 6.1.1 Verstehen, was ein Nutzungsszenario ist
- 6.1.2 Verstehen des Konzepts: Storyboard
- 6.1.4 Verstehen, was Card-Sorting ist
- 6.1.5 Verstehen, was Prototypen und Wireframes sind
- 6.1.6 Verstehen der Konzepte: Low-Fidelity-Prototyp und High-Fidelity-Prototyp sowie des Unterschieds zwischen Low-Fidelity- und High-Fidelity-Prototypen
- 6.1.8 Verstehen des Inhalts und des Zwecks der User-Interface-Spezifikation
- 6.1.9 Verstehen, was ethisches Design ist
- 6.1.10 Verstehen, was nachhaltiges Design ist

### 7.2.1 Nutzungsszenarien

Nutzungsszenarien (im Englischen »Use scenarios« oder auch »Scenarios of use«) beschreiben die Erledigung einer Aufgabe am interaktiven System durch den Benutzer. Sie fokussieren ausschließlich auf die Perspektive des Benutzers in der Nutzungssituation.

#### **Definition 7-5: Nutzungsszenario**

Eine erzählende, textuelle Beschreibung, wie ein Benutzer eine oder mehrere Aufgaben mit dem geplanten interaktiven System ausführen wird.

Soweit im Projekt Personas als Darlegungsform zur Beschreibung des Nutzungskontextes verwendet werden, können diese Personas natürlich auch die Benutzer in den Nutzungsszenarien repräsentieren.

In Tabelle 7–1 wird ein Beispiel für ein Nutzungsszenario gegeben:

<b>1. Einleitung</b>	
<b>Aufgabe, die im Nutzungsszenario betrachtet wird:</b>	Einen Behandlungstermin wahrnehmen, der verspätet beginnt
<b>Benutzergruppe:</b>	Patient
<b>Interaktives System:</b>	Terminvereinbarungs-App
<b>Persona:</b>	Martin Fischer (Details sind in der Persona-Beschreibung zu finden)
<b>Kontextuelle Vorbedingung(en):</b>	<ul style="list-style-type: none"> <li>■ Martin Fischer hat einen Behandlungstermin beim Arzt am heutigen Tag.</li> <li>■ Martin Fischer hat den Termin auf 17:30 Uhr gelegt, sodass er um 17:00 Uhr mit dem Bus vom Arbeitsplatz zur Arztpraxis fahren kann.</li> <li>■ Es ist gerade 16:00 Uhr.</li> </ul>
<b>Ziele (beabsichtigte Ergebnisse):</b>	<ul style="list-style-type: none"> <li>■ Martin Fischer ist kurz vor Beginn des Behandlungstermins beim Arzt.</li> <li>■ Er muss nicht länger als 10 Minuten im Wartezimmer warten, bis er drankommt.</li> </ul>

**2. Nutzungsszenario**

Martin Fischer hat heute einen Behandlungstermin bei seinem Hausarzt. Es ist ein Routinetermin, den er einmal im Jahr macht, um seinen Gesundheitszustand allgemein überprüfen zu lassen. Den Termin hat er vor Wochen mit seiner Terminvereinbarungs-App für heute 17:30 Uhr vereinbart. Es ist gerade 16:00 Uhr und Martin Fischer hat noch einiges zu tun. Er muss spätestens um 17:00 Uhr seinen Arbeitsplatz verlassen, um um 17:25 Uhr bei seinem Hausarzt einzutreffen.

Plötzlich vibriert sein Smartphone. Die Terminvereinbarungs-App teilt mit, dass der Behandlungstermin erst um 18:15 Uhr beginnen wird. Martin Fischer ist erleichtert. Er macht sich erst um 17:45 Uhr auf den Weg. Kurz bevor er sein Büro verlässt, schaut er noch einmal in seine Terminvereinbarungs-App. Sie zeigt an, dass der Termin um 18:15 Uhr beginnen wird. Die Angabe wurde zuletzt vor einer Minute aktualisiert.

Während Martin Fischer im Bus sitzt, schaut er noch einmal auf seine Terminvereinbarungs-App. Hier sieht er jetzt auch, dass noch ein Patient vor ihm dran ist. Er trifft um 18:10 Uhr in der Praxis ein. Die medizinische Fachangestellte in der Praxis begrüßt ihn freundlich und bittet um Entschuldigung für die Terminverspätung. Sie sagt, der Doktor sei gerade mit dem vorhergehenden Patienten fertig geworden und Martin Fischer könne direkt in das Behandlungszimmer zum Doktor gehen.

**Tab. 7–1** Beispiel für ein Nutzungsszenario (narrative Form)

Das CPUX-F-Curriculum kennt nur die Freitextform des Nutzungsszenarios, so wie in Tabelle 7–2 beschrieben. Nur diese Freitextform ist für die CPUX-F-Zertifizierungsprüfung relevant.

In der Literatur sind jedoch zahlreiche andere Formen für das Notieren von Nutzungsszenarios zu finden. Unter anderem wird im Beitrag »Modelling Usage: Techniques and Tools« [Riedemann & Freitag 2009] eine Form des Nutzungsszenarios vorgestellt, die sowohl das Aufgabenmodell (siehe Abschnitt 5.8.3 auf Seite 78) als auch die Interaktion des Benutzers am interaktiven System (Dialogschritte) und die Nutzungsanforderungen vereint. Diese tabellarische Form des Nutzungsszenarios wird im Curriculum Designing Solutions (CPUX-DS) [UXQB

CPUX-DS 2021] systematisch eingeführt und dort als »Interaktionsspezifikation« bezeichnet.

Tabelle 7–2 zeigt beispielhaft diese Notation, die durch ihre Strukturiertheit als Basis für die Entwicklung von Low-Fidelity-Prototypen sehr gut geeignet ist. Die Einleitung des Nutzungsszenarios in Tabelle 7–1 kann als Einleitung zu Tabelle 7–2 verwendet werden, in der dann das Nutzungsszenario präzise mit den Teilaufgaben des Aufgabenmodells, umzusetzenden Nutzungsanforderungen sowie den Dialogschritten des Benutzers am interaktiven System beschrieben wird.

Teilaufgaben	Aktion (des Benutzers)	Reaktion (an der Benutzungsschnittstelle)	Nutzungsanforderungen (NA) aus dem Nutzungskontext
Entscheiden, wann man sich zur Praxis begibt	Auswählen: Liste der anstehenden Behandlungstermine	Anzeigen/Signalisieren: Liste der Behandlungstermine mit jeweiliger Angabe der Praxis und des vereinbarten Beginns der Behandlung	NA1 Der Benutzer muss am System erkennen können, wann sein vereinbarter Behandlungstermin beginnt. NA2 Der Benutzer muss am System erkennen können, wann er von seinem aktuellen Aufenthaltsort aufbrechen muss.
Die aktuelle Uhrzeit überwachen	Auswählen: –/–	Anzeigen/Signalisieren: –/–	–/–
Feststellen, ob sich der Behandlungstermin verzögert	Auswählen: –/–	Anzeigen/Signalisieren: Termin verspätet sich mit Angabe des verspäteten Beginns und der Uhrzeit, zu der man los muss vom aktuellen Aufenthaltsort.	NA3 Der Benutzer muss am System erkennen können, dass sein Behandlungstermin sich verzögert. Siehe auch NA2
Zum festgelegten Zeitpunkt aufbrechen	Auswählen: –/–	Anzeigen/Signalisieren: Instruktion: Jetzt aufbrechen, Angabe der Buslinie mit Angabe Abfahrzeit	Siehe NA2
In der Praxis ankommen	Auswählen: –/–	Anzeigen/Signalisieren: –/–	–/–
Warten, bis man dran ist	Auswählen: App	Anzeigen/Signalisieren: Beginn nächster Behandlungstermin mit Angabe der Anzahl von Patienten, die noch behandelt werden	NA4 Der Benutzer muss am System erkennen können, wie viele Patienten noch vor ihm dran sind.
Den Behandlungstermin beim Arzt beginnen	Auswählen: –/–	Anzeigen/Signalisieren: –/–	–/–

**Tab. 7–2** Beispiel für ein Nutzungsszenario (tabellarische Form)

In der Notation in Tabelle 7–2 werden zunächst die Teilaufgaben des Aufgabenmodells eingetragen und anschließend alle (im Rahmen der Analyse des Nutzungskontextes) identifizierten Nutzungsanforderungen der jeweiligen Teilaufgabe zugeordnet. Erst nachdem Teilaufgaben und Nutzungsanforderungen verortet sind, wird für jede Teilaufgabe spezifiziert, welche Aktionen des Benutzers am interaktiven System ermöglicht werden müssen und welche Informationen durch das interaktive System bereitgestellt werden müssen. Hierbei kommt es durchaus vor, dass das Aufgabenmodell noch einmal überarbeitet wird. Die Reihenfolge von Teilaufgaben kann sich ändern und es kann im Einzelfall sein, dass im Lichte der umzusetzenden Nutzungsanforderungen einzelne Teilaufgaben ganz wegfallen oder auch neue hinzukommen. Eine solche überarbeitete Version des Aufgabenmodells wird deshalb auch »Aufgabenmodell für die Gestaltung« genannt [UXQB CPUX-UR 2023].

Die tabellarische Form des Nutzungsszenarios ist nicht im CPUX-F-Curriculum enthalten und nicht relevant für die CPUX-F-Zertifizierungsprüfung. Sie ist nicht zu verwechseln mit einem Use Case. Im Gegensatz zu den Nutzungsszenarien beschreiben Use Cases (»Anwendungsfälle«) aus der Perspektive des Systems, wie Handlungen von Benutzern systemseitig verarbeitet werden. Die Notation (Freitext, Liste, Tabelle, Diagramm) ist hierbei zweitrangig. In Use Cases werden insbesondere auch technische Aktionen und Reaktionen des Systems beschrieben, die für den Benutzer unsichtbar sind. Hierzu gehören z. B. die Verifizierung des Benutzers oder technische Plausibilitätsprüfungen durch das interaktive System [UXPA BOK 2011; EduTechWiki 2016].

Jegliche technische Betrachtung des interaktiven Systems (wie sie z. B. in einem Use Case zu finden ist) bleibt im Nutzungsszenario außen vor. Nutzungsszenarien dienen als Basis für die Entwicklung von Low-Fidelity-Prototypen. Nutzungsszenarien sollen helfen, all das zu klären, was klar sein muss, bevor mit der Entwicklung von (User-Interface-)Prototypen begonnen wird.

### **Merksatz**

Das Nutzungsszenario beschreibt die konkrete Erledigung einer Aufgabe am zukünftigen System aus der Perspektive eines Benutzers. Hauptverwendungszweck des Nutzungsszenarios ist das Ableiten von Low-Fidelity-Prototypen.

Im Unterschied hierzu beschreibt das Ist-Szenario textuell den derzeitigen (»gegebenen«) Nutzungskontext für eine Benutzergruppe mit seinen Komponenten Benutzer(merkmale), Ziele und Aufgaben, Ressourcen und Umgebung im Zusammenspiel. Hauptverwendungszweck des Ist-Szenarios ist das Herleiten von Nutzungsanforderungen.

## 7.2.2 Storyboards als Visualisierung von Nutzungsszenarien

Als Storyboard bezeichnet man eine Serie von üblicherweise 6 bis 8 Abbildungen, innerhalb derer die Erledigung einer Aufgabe am interaktiven System visualisiert wird. Auf den Abbildungen, die häufig wie Comics dargestellt sind, ist typischerweise jeweils der Benutzer bei der jeweiligen Teilaufgabe zu sehen, die er gerade erledigt, das interaktive System mit relevanten Informationen für den Benutzer sowie die physische und soziale Umgebung, in der sich der Benutzer gerade befindet.

### Definition 7-6: Storyboard

Eine Folge visueller Bildschirmhalte, die das Zusammenspiel zwischen einem Benutzer und einem interaktiven System veranschaulicht.

Storyboards können als Illustration von Nutzungsszenarien helfen, die eigentlichen Nutzungsszenarien im Projektteam oder auch zu Stakeholdern anschaulich zu kommunizieren und Konsens zu finden darüber, wie die jeweilige Aufgabe am interaktiven System unterstützt werden soll.

Insbesondere sind Storyboards hilfreich, wenn sich der Benutzer während der Aufgabenerledigung an verschiedenen Orten aufhält bzw. wenn bei den einzelnen Teilaufgaben andere Personen beteiligt sind (wechselnde soziale Umgebung). Physische Umgebungen und Menschen lassen sich schlichtweg gut grafisch repräsentieren.

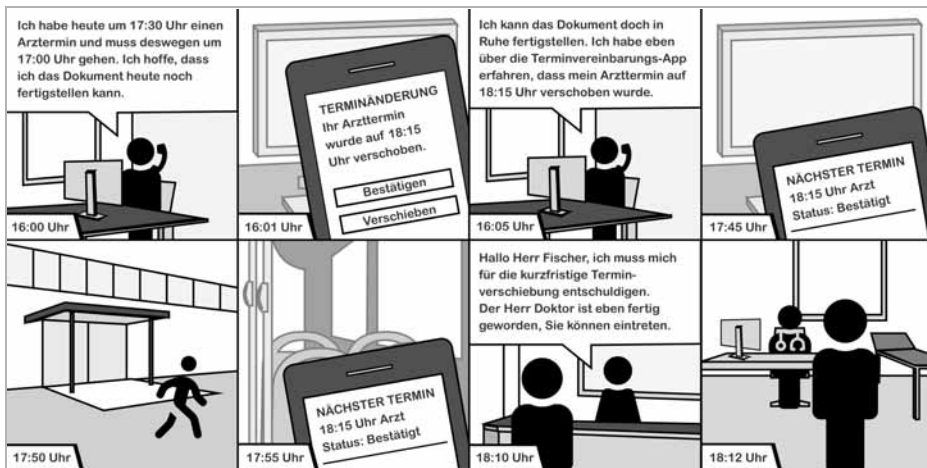


Abb. 7-13 Storyboard für die Aufgabe »Einen Arzttermin wahrnehmen, der sich verzögert«<sup>8</sup>

8. Quelle: Simon Birke, Senior Consultant Human-centred Design bei der ProContext Consulting GmbH.

### 7.2.3 Card-Sorting als Methode zur Strukturierung von Information

Als Card-Sorting (Sortierung von Karten) bezeichnet man eine Aktivität zum Strukturieren von Information, bei der ein Moderator mit (meist mehreren) Benutzern zusammenarbeitet. Das Ziel der Zusammenarbeit beim Card-Sorting ist es, mithilfe von Benutzern eines interaktiven Systems z.B. eine Navigationsstruktur zu entwickeln, in der diese dann die benötigten Bedienfunktionen des Systems systematisch und effizient auffinden können.

Dies ist insbesondere hilfreich für z.B. betriebswirtschaftliche Software, die sehr viele Bedienfunktionen hat, oder Websites, die sehr viele unterschiedliche Informationen bieten.

So kann man für ein Programm, für das man die Bedienfunktionen auf Basis der spezifizierten Nutzungsanforderungen identifiziert hat, jede Bedienfunktion auf eine Karte schreiben und diese dann durch die Benutzer sortieren lassen.

#### **Definition 7-7: Card-Sorting**

Eine Methode zum Strukturieren von Information – wie beispielsweise Menüs in einer Navigationsstruktur –, bei der Kernbegriffe auf verschiedene Karten geschrieben werden und Benutzer aufgefordert werden, diese Karten in Gruppen zu sortieren.

Man unterscheidet offenes Card-Sorting und geschlossenes Card-Sorting.

Beim offenen Card-Sorting werden die Teilnehmer gebeten, die Karten, die aus ihrer Sicht zusammengehören, zu gruppieren und für jede der erarbeiteten Gruppen einen Titel zu vergeben. Beim geschlossenen Card-Sorting stehen die Titel der Gruppen bereits fest und die Teilnehmer werden gebeten, die vorhandenen Karten den vorhandenen Titeln zuzuordnen. Das geschlossene Card-Sorting eignet sich besonders, wenn das Entwicklungsteam bereits eine Hypothese hat, wie z.B. Menüpunkte benannt sein sollten, und diese Hypothese überprüfen möchte.

Abbildung 7-14 zeigt das Ergebnis einer offenen Card-Sorting-Sitzung für die Navigationsstruktur eines Navigationssystems.



Abb. 7-14 Ergebnis einer offenen Card-Sorting-Sitzung<sup>9</sup>

Es ist wichtig für den Moderator, genau zuzuhören, was in der Gruppe während des Sortierens gesprochen wird, um Diskrepanzen (»Mismatches«) zwischen dem, was das Entwicklungsteam glaubt, und dem, was die Benutzer glauben, zu identifizieren.

### 7.2.4 Prototypen

Prototypen, so wie sie im Rahmen von menschenzentrierter Gestaltung verstanden werden, sind immer »User-Interface-Prototypen«. Der Begriff Prototyp wird in der Produktentwicklung auch für »technische Prototypen« verwendet, bei denen es darum geht, ob eine bestimmte Technologie das leistet, was man von ihr erhofft. Hierum geht es bei der menschenzentrierten Gestaltung jedoch nicht.

Ein (User-Interface-)Prototyp ist also immer eine »Vorschau« auf die Benutzungsschnittstelle für das interaktive System. In der Regel repräsentiert ein Prototyp zunächst nur Teile der Benutzungsschnittstelle.

**Definition 7-8: Prototyp**

Eine Repräsentation von Teilen oder des gesamten interaktiven Systems, die in Grenzen für Analyse, Design und Usability-Evaluierung benutzt werden kann.

9. Quelle: Ausbildung zum Usability & User Experience Professional bei artop GmbH, Berlin 2018.

Das vorrangige Ziel bei der Erarbeitung von Prototypen ist, dass man vor der eigentlichen technischen Implementierung einer Benutzungsschnittstelle diese bereits mit zukünftigen Benutzern auf ihre Usability und User Experience hin evaluieren kann. So gibt es inzwischen zahlreiche Werkzeuge zur Erzeugung von Prototypen, auch Prototyping-Tools genannt, mit denen man ohne eine Zeile Programmiercode schreiben zu müssen, bereits die Benutzungsschnittstelle eines interaktiven Systems »interaktiv erlebbar« machen kann. Prominente Beispiele für Prototyping-Tools sind Balsamiq, Axure, ForeUI, Sketch und Adobe XD.

Prototypen helfen, mit zukünftigen Benutzern des interaktiven Systems möglichst frühzeitig ins Gespräch zu kommen. Es kann leichter sein, ein Nutzungsszenario anhand eines Prototyps mit Benutzern zu evaluieren, als Benutzer mit dem textuellen Nutzungsszenario zu konfrontieren. Auch helfen Prototypen dabei, den Stakeholdern (wie z.B. Vorgesetzten von Benutzern) frühzeitig zu zeigen, woran man arbeitet und dass etwas entsteht. Prototypen erzeugen »Sichtbarkeit« in Entwicklungsprojekten.

Man unterscheidet Low-Fidelity-Prototypen und High-Fidelity-Prototypen.

**Definition 7-9: Low-Fidelity-Prototyp**

Eine preiswerte Veranschaulichung eines Designs oder eines Konzepts, die benutzt wird, um Feedback von Benutzern und anderen Interessenvertretern während früher Phasen der Entwicklung einzuholen.

Ein wesentliches Merkmal eines Low-Fidelity-Prototyps ist, dass er mit sehr geringem Aufwand erstellt und geändert werden kann. Das bedeutet meistens, dass visuelle Details weggelassen werden und der Fokus eines Low-Fidelity-Prototyps darauf liegt, was im User Interface zu sehen ist und wie die Interaktion abläuft, und noch nicht, wie Dinge im Detail visualisiert werden. Einerseits reduziert das den Aufwand in der Herstellung des Prototyps massiv, andererseits erreicht man so, dass Benutzer, mit denen man den (Low-Fidelity-)Prototyp evaluiert, sich wirklich darauf konzentrieren, ob dieser die Informationen enthält, die bei der Aufgabenerledigung benötigt werden, und die Schritte ermöglicht, die während der Aufgabenerledigung gemacht werden müssen. Anderenfalls diskutiert man über Formen und Farben, was bei frühen Evaluierungen vom eigentlich wichtigen Evaluierungsgegenstand ablenkt.

Low-Fidelity-Prototypen müssen sich schnell ändern lassen, wenn man im Rahmen von Evaluierungen den Änderungsbedarf erkennt, damit ihr Einsatz effektiv ist. Manche Low-Fidelity-Prototypen werden demgemäß auf Papier erstellt. Dies ist insbesondere nützlich, wenn man die Prototypen in einem Team erstellt, in dem »jeder Hand anlegt«. Es gibt zu »Paper Prototyping« sogar eigene Bücher (z.B. [Snyder 2003]).

Die inzwischen am Markt vorhandenen Prototyping-Tools eignen sich auch zur Erstellung von Low-Fidelity-Prototypen. Sie erfordern keinerlei Program-



merkmale und sind schnell erlernbar. Sie erzeugen meist Benutzungsschnittstellen, die ganz bewusst so aussehen, als wären sie gezeichnet worden.

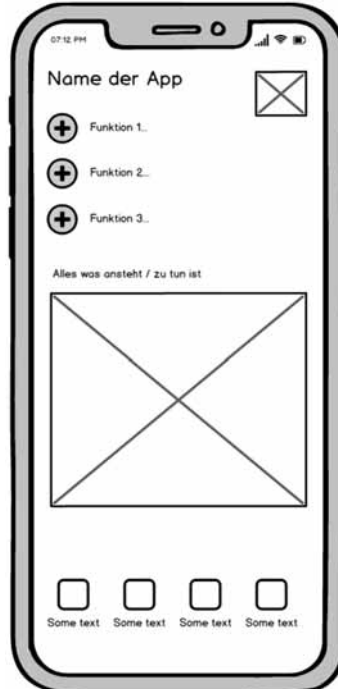
Jedoch gilt auch hier: »A fool with a tool is still a fool.« Wer Prototypen entwickelt, sollte dies nicht tun, ohne Klarheit über die zu unterstützenden Aufgaben in Form von Aufgabenmodellen zu haben und Klarheit über die Interaktion am System in Form von Nutzungsszenarien. Des Weiteren ist es für den »Prototyper« wichtig zu wissen, welche User-Interface-Elemente sich für welchen Zweck eignen und für welchen nicht.

Eine einzelne »Draufsicht« (z. B. eine Bildschirmseite) auf einen Low-Fidelity-Prototyp bezeichnet man auch als »Wireframe«.

#### **Definition 7–10: Wireframe**

Ein Screen oder eine Seite in einem Low-Fidelity-Prototyp für eine grafische Benutzungsschnittstelle, bestehend aus Linien, rechteckigen Kästen und Text, der das zukünftige Interaktionsdesign repräsentiert.

In Abbildung 7–15 ist ein solcher Wireframe für die Terminvereinbarungs-App abgebildet. Dieser Wireframe ist noch sehr rudimentär, da er zunächst nur darstellt, welche Informationen an welcher Stelle des Bildschirms dargestellt werden sollen.



**Abb. 7–15** Wireframe eines Low-Fidelity-Prototyps

In Abbildung 7–16 sieht man bereits zwei Wireframes der Terminvereinbarungs-App, die zeigen, wie man von der Startseite ausgehend einen neuen Behandlungstermin beim Arzt vereinbaren kann. Dies kann man bereits als Low-Fidelity-Prototyp bezeichnen, da eine Interaktion mit dem Prototyp bereits möglich ist. Mit einem Prototyping-Tool lässt sich diese Interaktion so herstellen, dass der Benutzer eigenständig vom linken Wireframe auf den rechten Wireframe kommt, indem er auf das Pluszeichen oder den Text »Arzttermin jetzt vereinbaren ...« klickt und dann das Bildschirmformular zur Terminauswahl erscheint.



**Abb. 7–16** Low-Fidelity-Prototyp der Terminvereinbarungs-App

Im Gegensatz zu Low-Fidelity-Prototypen sehen High-Fidelity-Prototypen optisch bereits so aus, wie die jeweilige Anwendung auf der »Zielplattform« nach der Entwicklung aussehen wird. Auch hierfür gib es Prototyping-Werkzeuge, die z. B. »originalgetreu« Windows-Anwendungen oder iPhone-Apps visualisieren.

#### **Definition 7–11: High-Fidelity-Prototyp**

Ein Prototyp, der die Benutzungsschnittstelle eines interaktiven Systems so genau wie möglich abbildet und oft pixelgenaue User-Interface-Elemente und visuelles Design aufweist.

Abbildung 7–17 zeigt eine Benutzungsschnittstelle einer Terminvereinbarungs-App, die sich auf diese Weise mit einem High-Fidelity-Prototyping-Tool herstellen und interaktiv erleben lässt, ohne dass sie technisch implementiert ist.

Man sollte jedoch im Projekt immer zunächst Low-Fidelity-Prototypen erstellen und diese so oft (iterativ) mit Benutzern evaluieren, bis man ausreichend Sicherheit hat, dass der Interaktionsablauf erwartungskonform ist und Klarheit über die Informationen herrscht, die dem Benutzer in jedem Dialogschritt präsentiert werden müssen. Darauf aufbauend ist das Erstellen von High-Fidelity-Prototypen sinnvoll, insbesondere wenn die Umsetzung der Benutzungsschnittstelle auf der Zielplattform von anderen Personen durchgeführt wird als denen, die die Low-Fidelity-Prototypen entwickelt haben, bzw. wenn die Umsetzung durch ein hierfür beauftragtes Unternehmen erfolgt.



**Abb. 7–17** Mobile App für die Arzttermin-Vereinbarung<sup>10</sup>

### Merksatz

Auf der Basis von Nutzungsszenarien sollten immer zunächst Low-Fidelity-Prototypen erstellt werden und so lange evaluiert werden, bis aus Benutzersicht keine neuen Erkenntnisse mehr für eine Überarbeitung gewonnen werden können. Diese Iterationen sind sehr viel kostengünstiger als Iterationen an teilweise entwickelter Software.

10. Quelle: jameda GmbH, [www.jameda.de](http://www.jameda.de).

Typische Verwendungen von High-Fidelity-Prototypen sind:

- Als Teil der Spezifikation des User Interface als Vorgabe für die Entwicklung
- Bei Marketing und Sales als Instrument für Demos
- Zum Testen von Usability-Aspekten, die von der visuellen Gestaltung und anderen Details abhängen, z.B. Attraktivität, Affordance, Selbstbeschreibungsfähigkeit

### 7.2.5 User-Interface-Spezifikation

Spezifikationen werden immer formuliert, um denjenigen, die etwas liefern oder umsetzen sollen, einerseits klarzumachen, was geliefert bzw. umgesetzt werden soll, und andererseits Unsicherheiten und Diskussionsbedarf auf das wirklich notwendige Maß zu reduzieren.

Eine User-Interface-Spezifikation dient demzufolge dazu, dass diejenigen, die eine Benutzungsschnittstelle implementieren sollen, alle Informationen verfügbar haben, die hierzu notwendig sind. Natürlich wird es immer wieder Rückfragen und Kommunikationsbedarf geben zwischen denen, die die User-Interface-Spezifikation erstellt haben, und denjenigen, die diese umsetzen sollen. Aber alleine durch das Vorhandensein der User-Interface-Spezifikation wird dieser Kommunikationsbedarf auf wirklich nützlichen Austausch statt »Guesswork« beschränkt.

#### **Definition 7–12: User-Interface-Spezifikation**

Eine Beschreibung des Aussehens und Verhaltens der Benutzungsschnittstelle eines interaktiven Systems.

In der User-Interface-Spezifikation muss einerseits das dynamische Verhalten der Benutzungsschnittstelle beschrieben sein und andererseits alle Bedienfunktionen und Benutzerunterstützung.

Aus der Erfahrung der Autoren heraus beinhaltet eine User-Interface-Spezifikation folgende Punkte:

- Alle Ansichten (Screens) des interaktiven Systems mit allen Inhalten und dem gesamten Informationsfluss zwischen ihnen (UI-Elemente, Texte, Bilder, Videos, Warn- und Fehlermeldungen, Benutzerunterstützung, Übersetzungen etc.)
- Das Verhalten der Benutzungsschnittstelle bei Interaktionen
- Low-Fidelity-Prototypen (in der Regel annotiert zwecks Spezifikation des Verhaltens)
- Gegebenenfalls High-Fidelity-Prototypen, wenn es keine standardisierte Zielplattform und/oder keinen Styleguide gibt
- Informationsarchitektur und Navigationsstruktur

- Beschreibung aller Bedienfunktionen (Aufgabenobjekte und ausführbare Funktionen)
- Beschreibung aller Elemente der Benutzerunterstützung – Meldungen, Statusinformationen, Instruktionen, Onlinehilfe und Benutzerdokumentation (sofern bereits verfügbar)
- Nutzungsszenarien

### 7.2.6 Ethisches Design und nachhaltiges Design

Der Begriff »ethisch« kommt von »Ethik«. »Ethik ist die Lehre bzw. Theorie vom Handeln gemäß der Unterscheidung von gut und böse. Gegenstand der Ethik ist die Moral« [Gabler Wirtschaftslexikon].

#### **Definition 7–13: Ethisches Design**

Ein Verhaltensprinzip, das Erfordernissen [und Zielen] [der Benutzer] Vorrang vor individuellen [Zielen einzelner Stakeholder] oder organisatorischen Zielen [des Herstellers/Auftraggebers] gibt.

Ethisches Design geht davon aus, dass bei der Gestaltung von interaktiven Systemen die Erfordernisse der Benutzer im Vordergrund stehen statt der Vorlieben einzelner Stakeholder oder der Ziele der Organisation, die das interaktive System herstellt bzw. in Verkehr bringt. Natürlich will ein Hersteller eines interaktiven Systems Profite erzielen. Dies soll jedoch gelingen, indem die Erfordernisse der Benutzer des interaktiven Systems befriedigt werden, statt eine positive Antizipation der Nutzung zu erzeugen, die dann während und nach der Nutzung negative Erlebnisse verursacht.

Ethisches Design ist immer auf langfristigen Produkterfolg ausgerichtet und bestrebt, Gutes für die Benutzer zu tun.

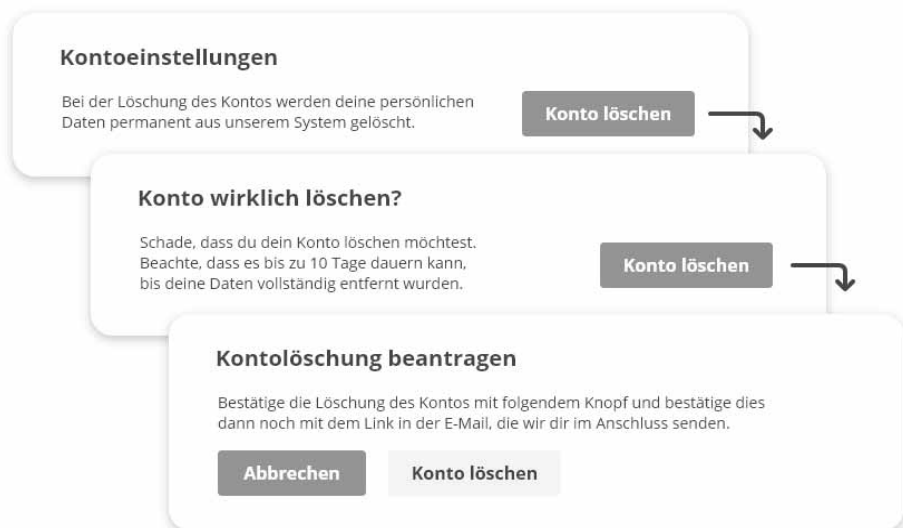
Abbildung 7–18 zeigt ein Beispiel für ethisches Design einer Benutzungsschnittstelle. Bei einem Fahrzeug wird dem Fahrer die Statusinformation angezeigt, dass das Fahrzeug Müdigkeit beim Fahrer erkannt hat, und eine Fahrpause empfohlen wird. Dies ist gut zu wissen für den Fahrer, der ermuntert wird, eine Pause zu machen, und es dient der Sicherheit aller Verkehrsteilnehmer.



**Abb. 7-18** Müdigkeitserkennung bei Pkws von VW<sup>11</sup>

Sogenannte »Dark Patterns« bzw. Deceptive Patterns sind prominente Beispiele für unethisches Design. Es handelt sich hier um Benutzungsschnittstellen-Designs, die bewusst so gestaltet sind, dass der Benutzer zu Handlungen verleitet wird, die dessen Interessen entgegenlaufen [Deceptive Patterns]. Inzwischen wird hier auch der Begriff »Deceptive Design« (trügerisches Design) verwendet.

Abbildung 7-19 zeigt ein Beispiel für unethisches Design in Form eines »Dark Pattern«. Bei der dritten Rückmeldung des Systems im Verlauf des Löschens eines Benutzerkontos ist die Schaltfläche »Abbrechen« visuell identisch dargestellt mit der Schaltfläche »Konto löschen« bei der ersten und der zweiten Rückmeldung. Der Benutzer wird verleitet, unabsichtlich die Löschung des Kontos abzubrechen.



**Abb. 7-19** Absichtliche Irreführung beim Löschen eines Benutzerkontos<sup>12</sup>

11. Quelle: <https://www.volkswagen-newsroom.com/de/muedigkeitserkennung-3932>.

12. Quelle: <https://www.redim.de/blog/dark-patterns>.

Die Begriffe »Dark Pattern« und »Honest Design« sind im CPUX-DS-Curriculum definiert und dort tiefergehend beschrieben. Für die CPUX-F-Prüfung sind diese Begriffe nicht relevant.

Neben dem Begriff »ethisches Design« gibt es den Begriff »nachhaltiges Design«. Beide Begriffe sind im CPUX-F-Curriculum definiert und somit prüfungsrelevant.

**Definition 7–14: Nachhaltiges Design**

Ein Gestaltungsansatz, der den Menschen und dem Planeten Vorrang einräumt, indem die für die Nutzung interaktiver Systeme erforderlichen Ressourcen minimiert werden.

Nachhaltiges Design zielt darauf ab, dass interaktive Systeme (und Produkte insgesamt) so gestaltet werden, dass diese insbesondere bei der Nutzung, aber auch bei der Herstellung und Entsorgung so wenige Ressourcen wie möglich verbrauchen, um so den Erhalt des natürlichen Lebens auf der Erde nicht zu gefährden.

In Bezug auf die Gestaltung schließt Ressourcenminimierung hierbei unter anderem Folgendes ein:

- Minimaler Einsatz von seltenen Erden in Produkten
- Minimaler Stromverbrauch während der Nutzung
- Minimaler Schadstoffausstoß bei der Nutzung
- Recyclingfähigkeit von verwendeten Materialien

So lässt sich leicht begreifen, dass die Nutzung von Streamingdiensten hohen Strom für den Betrieb von Serverfarmen erfordert und diese wiederum gekühlt werden müssen, was wiederum zu vermehrtem CO<sub>2</sub>-Ausstoß in die Atmosphäre führt.

Jegliche Gestaltungen an internetbasierten Anwendungen, die die Interaktion zwischen Benutzer und System erhöhen, sei es durch überflüssige Schritte bei Benutzern oder durch häufige Behebung von Benutzungsfehlern, führen zur mehr Datenverkehr über Datenleitungen und sind somit umweltschädlich.

Effiziente Gestaltung der Interaktion zwischen Benutzer und System liefert somit einen Beitrag zum nachhaltigen Design.

**7.2.7 Prüfungsfragen zu diesem Abschnitt**

Die folgenden Fragen dienen zur Vertiefung der Inhalte des Abschnitts und zur Vorbereitung auf die CPUX-F-Zertifizierungsprüfung. Die Fragen entsprechen in ihrer Form und Schwere den Prüfungsfragen für die CPUX-F-Zertifizierungsprüfung. Es handelt sich aber nicht um tatsächliche Prüfungsfragen, sondern die Fragen wurden von den Autoren speziell für dieses Buch erstellt.

Die Lösungen zu den Fragen befinden sich in Anhang A »Lösungen zu den Prüfungsfragen«.

Frage 97	2 richtige Antworten	LZ 6.1.9
Welche zwei der folgenden Aussagen beschreiben eine Gestaltung eines interaktiven Systems, die den Grundsätzen des ethischen Designs NICHT folgt?		
A	Beim Einwilligungsbanner auf einer Website (auch »Cookie-Banner« genannt) ist die Standardeinstellung beim Bestätigen durch den Benutzer die bzgl. Datenschutz für ihn beste Einstellung. Dies wird auch durch die DSGVO gesetzlich gefordert.	
B	Während der Benutzung einer App wird der Benutzer wiederholt darauf aufmerksam gemacht, was die App alles leisten kann und warum er die App weiterbenutzen sollte.	
C	In einer Fitness-App kann der Benutzer durch Sport und Bewegung Punkte sammeln und damit Preise gewinnen. Wenn der Benutzer sich nicht genug bewegt, wird er von der App erinnert und es werden keine weiteren Punkte vergeben.	
D	Eine Social-Media-Website bietet dem Benutzer an, einen Newsletter zu abonnieren mit der Formulierung »Abonnieren Sie unseren kostenlosen Newsletter«. Der Benutzer muss zum Abonnieren seinen Namen und seine E-Mail-Adresse angeben sowie weitere persönliche Angaben.	
E	Eine mobile Zahlungsanwendung zur Benutzung im Restaurant setzt eine Option »Trinkgeld« als Standard. Der Benutzer kann den voreingestellten Trinkgeld-Betrag auswählen oder die Option abwählen.	
F	Ein Online-Streamingdienst kann einen Monat gratis getestet werden. Der Benutzer wird eine Woche vor Ablauf der Gratis-Zeit informiert und erhält die Möglichkeit, sein Abonnement fortzusetzen, indem er seine Kreditkartendaten zur Bezahlung des Abonnements hinterlegt.	

Frage 98	2 richtige Antworten	LZ 6.1.8
Welche zwei der folgenden Aussagen über User-Interface-Spezifikationen sind korrekt?		
A	Der Zweck einer User-Interface-Spezifikation ist, die Einhaltung aller relevanten Anforderungen überprüfen zu können.	
B	Eine allgemeingültige Regelung, welche Farbe wofür verwendet werden darf, ist Bestandteil einer User-Interface-Spezifikation.	
C	Es kann sein, dass eine User-Interface-Spezifikation nur aus einem annotierten Prototyp besteht.	
D	Meldungen (Fehlermeldungen, Warnungen, Hinweise) gehören zur Informationsarchitektur und nicht zur User-Interface-Spezifikation.	



Frage 98	Fortsetzung	LZ 6.1.8
E	Welche User-Interface-Elemente wofür verwendet werden dürfen, ist Bestandteil einer User-Interface-Spezifikation.	
F	Der primäre Adressat einer User-Interface-Spezifikation sind die Entwickler.	

<b>Frage 99</b>	<b>1 richtige Antwort</b>	<b>LZ 6.1.10</b>
Welcher eine der folgenden Aspekte ist der primäre Fokus von nachhaltigem Design?		
A	Das nachhaltige Erreichen der Ziele der Benutzer durch die Verwendung des interaktiven Systems	
B	Das Minimieren des kognitiven Aufwands für die Bedienung des interaktiven Systems	
C	Das Minimieren des Energieaufwands für die Verwendung des interaktiven Systems	
D	Die Vermeidung von Umweltverschmutzung und Energieverschwendung bei der Verwendung des interaktiven Systems	
E	Die dauerhafte Erfüllung aller relevanten Anforderungen durch das interaktive System	
F	Das Minimieren der erforderlichen Ressourcen für die Nutzung des interaktiven Systems	

<b>Frage 100</b>	<b>3 richtige Antworten</b>	<b>LZ 6.1.2</b>
Welche drei der folgenden Aussagen sind korrekt in Bezug auf Storyboards?		
A	Storyboards sind etwas Ähnliches wie Nutzungsszenarien.	
B	Storyboards werden in der Gestaltungsphase erstellt.	
C	Ein Storyboard ist ähnlich einem Kanban-Board, nur dass es keine Arbeitspakete enthält, sondern User Stories.	
D	Ein Storyboard ist eine comicartige Darstellung des Zusammenspiels zwischen einem Benutzer und einem interaktiven System.	
E	Ein Storyboard ist eine umfassende Sammlung aller Nutzungsszenarien für ein zu erstellendes interaktives System.	
F	Storyboards werden von Benutzern im Rahmen von Usability-Evaluierungen erstellt.	

<b>Frage 101</b>		<b>3 richtige Antworten</b>	<b>LZ 6.1.5</b>
Welche drei der folgenden Aussagen über Prototypen und Wireframes sind korrekt?			
A	Ein Wireframe ist ein Bestandteil eines Prototyps.		
B	Ein Prototyp ist eine Form von Wireframe.		
C	Bei der Evaluierung von Prototypen können Informationen über den Nutzungskontext ermittelt werden.		
D	Prototypen werden immer nur in der Designphase und der Evaluierungsphase verwendet.		
E	Prototypen können als Teil der Spezifikation für die Implementierung der Benutzerschnittstelle eines interaktiven Systems dienen.		
F	Wireframes zeigen nicht das visuelle Design des interaktiven Systems, wohl aber das exakte Layout von allen UI-Elementen in einem Screen oder einer Seite.		

<b>Frage 102</b>		<b>1 richtige Antwort</b>	<b>LZ 6.1.6</b>
Für eine interne Demonstration hat einer Ihrer Kollegen ein neues Bestellsystem auf einigen Blättern Papier skizziert. Jedes Blatt stellt einen Screen dar. Der Inhalt der Screens ist handgezeichnet und unvollständig. Ihr Kollege wechselt die Screens, sobald man auf den Button »klickt«, indem man mit einem Bleistift das Blatt berührt.			
Welcher eine der folgenden Begriffe beschreibt diese Sammlung von Skizzen am besten? (Quelle der Frage: [UXQB CPUX-F 2023b])			
A	Gestaltungsregel		
B	Low-Fidelity-Prototyp		
C	High-Fidelity-Prototyp		
D	Wireframe		
E	Styleguide		
F	Storyboard		

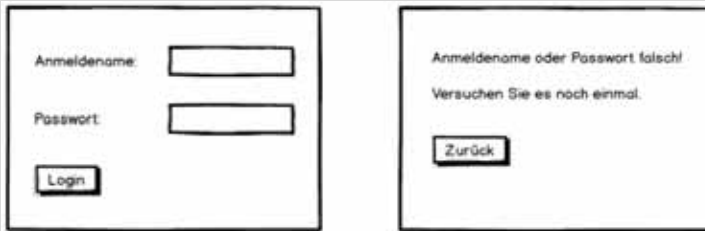
<b>Frage 103</b>		<b>2 richtige Antworten</b>	<b>LZ 6.1.5</b>
Welche zwei der folgenden Aussagen beschreiben wichtige Merkmale eines Wireframe?			
A	Ein Wireframe besteht aus Formen wie zum Beispiel Linien, Kästen und Text.		
B	Ein Wireframe beinhaltet Aspekte des visuellen Designs des interaktiven Systems.		
C	Ein Wireframe zeigt immer das exakte visuelle Design des interaktiven Systems.		
D	Ein Wireframe ist eine Seite / ein Screen in einem Low-Fidelity-Prototyp.		
E	Ein Wireframe beinhaltet Aspekte des Layouts der User-Interface-Elemente.		
F	Ein Wireframe zeigt immer das exakte Layout der User-Interface-Elemente.		

Frage 104	2 richtige Antworten	LZ 6.1.1
Welche zwei der folgenden Punkte stellen einen Ausschnitt eines validen Nutzungsszenarios dar für die Website eines Pizza-Lieferservice und die Aufgabe »Artikel bestellen«?		
A	Hanna geht auf die Website des Pizza-Lieferservice und da sie früher schon einmal dort bestellt hat, wird sie direkt mit Namen begrüßt und gebeten, ihr Benutzerpasswort einzugeben. Nachdem sie das getan hat, sieht sie die Artikel aufgelistet, die in ihren letzten Bestellungen enthalten waren, wo auch ihre selbst zusammengestellte Lieblingspizza dabei ist. Sie wählt diese aus und freut sich, dass sie nicht den Aufwand treiben muss, die Pizza erneut mühsam zusammenzustellen.	
B	<ol style="list-style-type: none"> <li>1. Benutzer identifizieren, entweder automatisch oder durch Angabe des Benutzernamens</li> <li>2. Benutzer authentifiziert sich durch Eingabe eines Passworts</li> <li>3. Artikel der letzten Bestellungen auflisten, neueste zuerst</li> <li>4. Benutzer wählt aus der Liste aus und legt pro Artikel die Anzahl fest</li> </ol>	
C	Der Benutzer gibt sein Passwort ein. Er sieht dann die Artikel seiner letzten Bestellungen in der Reihenfolge »Neueste zuerst«, auch selbst zusammengestellte Pizzen, wenn welche dabei waren. Er kann beliebige Artikel aus der Liste auswählen und diese einfach erneut bestellen.	
D	Der Benutzer Nr. 3 gibt Benutzernamen und Passwort ein und bekommt dann die Artikel seiner letzten Bestellungen aufgelistet in der Reihenfolge »Neueste zuerst«. Er verzieht das Gesicht und sagt »Warum kann ich denn nicht meine Lieblingsartikel festlegen? Das wäre doch viel praktischer, als nur die letzten Bestellungen zu sehen, wo ja bestimmte Artikel immer wieder aufgelistet werden. Jetzt muss ich durch die ganze lange Liste durchscrollen, um meine selbst zusammengestellte Pizza zu finden, die ich wieder bestellen möchte!«	
E	Der Benutzer wird identifiziert über ein Feld »Benutzername« (wird wenn möglich automatisch vorausgefüllt mittels Nutzung eines Cookies) und authentifiziert über ein Feld »Passwort«. Alternativ kann er sich auch neu registrieren, ein Bestellen ohne Login soll nicht erlaubt sein. Nach Anklicken des Buttons »Los geht's« und erfolgreicher Prüfung der Login-Daten bekommt der Benutzer die Artikel seiner letzten Bestellungen aufgelistet in der Reihenfolge »Neueste zuerst«, auch selbst zusammengestellte Pizzen, wenn welche dabei waren.	
F	Hanna bestellt mindestens zweimal pro Woche eine selbst zusammengestellte Pizza bei ihrer Lieblingspizzeria. Hier kann sie die Pizza dann 15 Minuten später abholen. Ihre Kombinationen von Zutaten sind im Pizzamenü der Pizzeria nicht als Standardpizza enthalten. Manchmal kommt sie beim Anruf durcheinander und vergisst eine Zutat zu nennen, die sie eigentlich haben wollte. Das ärgert sie dann, wenn sie später zu Hause die Pizza genießen will.	

## Frage 105

1 richtige Antwort

LZ 6.1.6



Welcher eine der folgenden Begriffe beschreibt die Abbildung am besten?

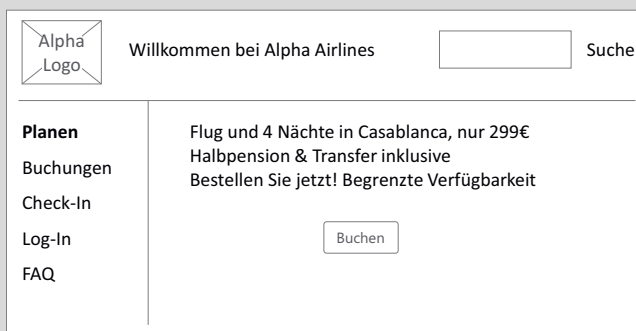
- |   |                       |
|---|-----------------------|
| A | Design Pattern        |
| B | Nutzungsszenario      |
| C | Wireframe             |
| D | Bildschirmaufteilung  |
| E | Storyboard            |
| F | Low-Fidelity-Prototyp |

## Frage 106

1 richtige Antwort

LZ 6.1.5

Welcher eine der folgenden Begriffe beschreibt die unten stehende Abbildung am besten?



(Quelle der Frage: [UXQB CPUX-F 2023b])

- |   |                  |
|---|------------------|
| A | Design Pattern   |
| B | Persona          |
| C | Nutzungsszenario |
| D | Wireframe        |
| E | Styleguide       |
| F | Storyboard       |

Frage 107	3 richtige Antworten	LZ 6.1.4
Welche drei der folgenden Aussagen sind korrekt in Bezug auf den Begriff »Card-Sorting«?		
A	Card-Sorting ist eine Methode zum benutzerorientierten Strukturieren von Informationen.	
B	Card-Sorting kann sowohl zum Erstellen neuer Navigationsstrukturen als auch zum Verbessern von bestehenden Navigationsstrukturen verwendet werden.	
C	Es werden zwei Arten von Card-Sorting unterschieden: neutral und vordefiniert.	
D	Die Teilnehmer an einem Card-Sorting sollten immer Usability Professionals sein.	
E	Die Teilnehmer an einem Card-Sorting sollten immer Benutzer des betreffenden interaktiven Systems sein.	
F	Beim Card-Sorting sind die Karten und deren Beschriftungen immer von einem Usability Professional vorgegeben.	

Frage 108	3 richtige Antworten	LZ 6.1.6
Welche drei der folgenden Punkte beschreiben am besten den Unterschied zwischen einem Low-Fidelity-Prototyp und einem High-Fidelity-Prototyp?		
A	Low-Fidelity-Prototypen veranschaulichen immer nur Konzepte, wogegen High-Fidelity-Prototypen weit mehr umfassen.	
B	Low-Fidelity-Prototypen können mit sehr geringem Aufwand geändert werden, High-Fidelity-Prototypen in der Regel nicht.	
C	Ein High-Fidelity-Prototyp kann als Teil der Spezifikation der Benutzungsschnittstelle dienen, ein Low-Fidelity-Prototyp nicht.	
D	Ein High-Fidelity-Prototyp ähnelt dem fertigen interaktiven System, ein Low-Fidelity-Prototyp in der Regel nicht.	
E	High-Fidelity-Prototypen sollten mit dem zu nutzenden technischen Framework abgestimmt sein, damit nicht etwas in dem Prototyp enthalten ist, das gar nicht umsetzbar ist. Bei Low-Fidelity-Prototypen spielt das keine Rolle.	
F	Sowohl Low-Fidelity- als auch High-Fidelity-Prototypen sollten immer aufgabenorientiert gestaltet sein, also eine sinnvolle Aufgabenerledigung demonstrieren können.	

### 7.3 Prinzipien und Richtlinien für die Gestaltung von Benutzungsschnittstellen

Lernziele in diesem Abschnitt (die Nummerierung entspricht den Lernziel-Nummern im CPUX-F-Curriculum):

- 6.2.1 Verstehen des Konzepts: Interaktionsprinzip
- 6.2.2 Verstehen aller sieben Interaktionsprinzipien
- 6.2.3 Verstehen des Konzepts: Heuristik