

*Qualität ist kein Zufall;
sie ist immer das Ergebnis angestrengten Denkens.*

John Ruskin (1819–1900)

In diesem Kapitel werden Definitionen zur Qualität und Maßnahmen beschrieben, mit denen die Qualität von – nicht nur – webbasierten und mobilen Applikationen sichergestellt wird. Dazu werden vier wichtige Fragen der Qualitätssicherung beantwortet:

1. Was soll qualitätsgesichert werden?
2. Was ist Qualität?
3. Wie kann Qualität gemessen werden?
4. Wie kann Qualität erzeugt werden?

2.1 Was soll qualitätsgesichert werden?

Ein **Testobjekt** ist u. a. ein Programm, eine Komponente, ein Teilsystem oder das gesamte System, das einem Test unterzogen wird.

Die Differenzierung von Testobjekten spielt eine wichtige Rolle. Es ist ein Unterschied für die einzusetzenden Methoden, Tools und Ressourcen, ob eine Website, eine Mobile-App oder eine Web-App qualitätsgesichert werden muss. Daher ist es wichtig, die Typen der zu testenden Objekte voneinander abzugrenzen.

Es gibt Systemsoftware, systemnahe Software und Anwendungssoftware. Systemsoftware sorgt dafür, dass Rechner überhaupt funktionstüchtig sind, wie z. B. Betriebssysteme. Systemnahe Software wird benötigt, um Anwendungssoftware realisieren zu können,

dazu gehören z. B. Compiler und Testtools. In diesem Buch konzentrieren wir uns auf Anwendungssoftware.

► Alles ist App!

In der IT ist eine **Anwendung** oder Applikation oder kurz **App** ganz allgemein eine Software, die auf (irgend)einem Computer läuft und die einen Benutzer bei einer bestimmten Aufgabe unterstützt und ihm einen Nutzen bringt. Eine Anwendung alias App kann

- als Batch-Job (Stapelverarbeitung) ohne Benutzerinteraktion auf einem Rechner laufen,
- als Embedded Software ein technisches System regeln, steuern und überwachen oder
- dialogorientiert über eine Benutzeroberfläche mit dem Benutzer kommunizieren.

Web-Apps und Mobile-Apps, die – wie der Titel verspricht – in diesem Buch schwerpunktmäßig betrachtet werden, sind dialogorientierte Anwendungen.

Eine **Web-App** ist ein Computerprogramm, das auf einem Webserver ausgeführt wird, wobei eine Interaktion mit dem Benutzer ausschließlich über einen Webbrowser erfolgt. Hierzu sind der Computer des Benutzers (Client) und der Server über ein Netzwerk, wie das Internet oder ein Intranet, miteinander verbunden, so dass die räumliche Entfernung zwischen Client und Server unerheblich ist. Eine Web-App kann per Definition auch im Browser eines mobilen Gerätes laufen.

Bei einer **Client-Server-App** kommuniziert der Client ebenfalls mit einem Server in einem Netzwerk. Aber im Gegensatz zur Web-App ist eine Client-Server-Anwendung plattformabhängig und muss auf dem Client installiert werden.

Der Vollständigkeit halber: Eine **Desktop-App** läuft nur lokal auf einem Desktop-PC (oder Notebook) und muss dort installiert werden.

Eine mobile Applikation oder kurz **Mobile-App** ist eine Anwendungssoftware für mobile Endgeräte (wie Smartphones und Tablet-Computer) bzw. für mobile Betriebssysteme. Mobile-Apps werden unterschieden in Mobile-Web-Apps, Native-Apps und Hybrid-Apps.

Mobile-Web-Apps basieren auf Webtechnologien, kommunizieren mit einem Server und werden über den Webbrowser des Mobilgeräts abgerufen, sie müssen daher nicht installiert werden. Eine Mobile-Web-App kann prinzipiell im Browser eines Desktop-PCs laufen, sofern die Bedienung möglich ist und sie nicht für Desktops gesperrt ist.

Native-Apps werden für ganz spezielle Geräte für ein bestimmtes Betriebssystem entwickelt, sie sind hersteller- und plattformabhängig und müssen installiert werden.

Hybrid-Apps vereinen die Vorteile von Mobile-Web-Apps und Native-Apps, weil sie auf die Hard- und Software-Komponenten des mobilen Endgerätes zugreifen und gleichzeitig unterschiedliche Plattformen bedienen können. Zum Beispiel können Kalender- und Kamerafunktionen in eine Hybrid-App integriert werden. Hybrid-Apps müssen installiert werden.

Fassen wir etwas salopp zusammen:

- ▶ Eine App mit „Web“ im Namen läuft auf browserfähigen Geräten, ist plattform-unabhängig und muss (abgesehen von eventuell benötigten Plugins) nicht installiert werden.
- ▶ Eine App ohne „Web“ im Namen ist plattformabhängig, läuft nicht im Browser, besitzt daher eine eigene Benutzeroberfläche und muss auf dem Client installiert werden.

Websites liefern dem Benutzer Informationen, dienen der Kontaktaufnahme und können Anwendungen wie z. B. Webshops im Netz bereitstellen. Sie sind quasi abgespeckte Web-Apps.

Eine **Website** (Webpräsenz oder Webauftritt) ist die Gesamtheit aller Seiten eines Webangebotes, d. h. alle Webseiten bilden zusammen eine Website. Eine **Webseite** (Webpage) ist eine einzelne Internetseite, d. h. ein Teil einer Website. Eine **Homepage** ist die Startseite einer Website.

Die drei Testobjekte Website, Webseite und Homepage werden manchmal fälschlicherweise synonym verwendet. Als Merksatz zur Unterscheidung der drei Begriffe dient die Qualitätsanforderung¹:

- ▶ Jede Webseite einer Website muss einen Link auf die Homepage enthalten.

Bezogen auf die Zielplattform können Webseiten weiter differenziert werden. Ist eine Webseite für Desktop-PCs und Notebooks programmiert, handelt es sich um eine **Desktop-Webseite**.

Eine mobile Webseite oder **Mobile-Webseite** ist speziell für die Darstellung von Inhalten auf mobilen Endgeräten entwickelt. Gewöhnlich stellt eine mobile Webseite als Ergänzung des Webauftritts den Inhalt einer Desktop-Webseite in grafisch abgespeckter, für mobile Geräte optimierter Form dar.

Geschafft! Wir sind bei der Antwort auf die Frage, welche Typen von Testobjekten in diesem Buch schwerpunktmäßig betrachtet werden. Es sind die Testobjekttypen, die der Benutzer „vordergründig“ auf seiner Anwendungsoberfläche, seinem **Front-End** sieht:

- Website
- Web-App
- Mobile-Web-App

Im Normalfall bestehen diese Testobjekte nicht alleine, sie sind Bestandteil eines mehr oder weniger umfangreichen Gesamtsystems. Denn im Hintergrund kommunizieren sie mit einer Vielzahl von Systemkomponenten, dem **Back-End**. Dazu können unter anderem

¹ Qualitätsanforderungen s. Kap. 2.3.

Datenbanken, Firewalls, Endgeräte, Netzwerke, Server, Schnittstellen und Webservices gehören. Alle diese Objekte müssen „hintergründig“ in die Tests miteinbezogen werden.

2.2 Was ist Qualität?

Was Qualität von Software ist und welche Merkmale die Qualität von Software bestimmen, wird in mehreren Normen festgelegt. Die für den Inhalt dieses Buches relevanten Normbegriffe werden im Folgenden vorgestellt.² Zudem werden spezielle Qualitätsmerkmale für die im Kap. 2.1 definierten Testobjekte des Front-Ends beschrieben.

2.2.1 Normenbestimmte Qualitätsmerkmale

Die ISO/IEC 25000 (Software-Engineering – Software product Quality Requirements and Evaluation (SQuaRE) [ISO/IEC 25000]) ist eine internationale, mehrteilige Norm. Sie ersetzt die bis 2005 gültige ISO/IEC 9126 (Software-Engineering – Qualität von Software-Produkten) und legt nun die Begriffe zur Software-Qualität fest:

- ▶ „Software-Qualität ist die Gesamtheit der Merkmale und Merkmalswerte eines Software-Produkts, die sich auf dessen Eignung beziehen, festgelegte oder vorausgesetzte Erfordernisse zu erfüllen.“

Die Normen unterscheiden die sechs **Qualitätsmerkmale**

- Funktionalität,
- Zuverlässigkeit,
- Benutzbarkeit,
- Effizienz,
- Änderbarkeit,
- Übertragbarkeit,

wobei die letzten fünf die **nicht-funktionalen Qualitätsmerkmale** einer Software sind.

Nach diesen sechs Qualitätsmerkmalen sind die Kapitel des zweiten Teiles dieses Buches gegliedert, in denen die Tests zur Überprüfung eben dieser Merkmale beschrieben werden.

Funktionalität (Functionality) Vorhandensein von Funktionen, die festgelegte und vorausgesetzte Erfordernisse erfüllen.

² Die vollständigen Normtexte sind beim Beuth-Verlag [URL: Beuth] erhältlich.

Teilmerkmale: Angemessenheit, Interoperabilität³, Ordnungsmäßigkeit^[GL], Richtigkeit, Sicherheit⁴

Zuverlässigkeit (Reliability) Fähigkeit einer Software/eines Systems, ihr/sein Leistungsniveau unter festgelegten Bedingungen in einem festgelegten Zeitraum zu halten.

Teilmerkmale: Fehlertoleranz, Reife, Wiederherstellbarkeit

Benutzbarkeit (Usability) Fähigkeit einer Software, unter festgelegten Bedingungen für einen Benutzer verständlich, erlernbar, anwendbar und attraktiv zu sein.

Teilmerkmale: Attraktivität, Bedienbarkeit, Erlernbarkeit, Verständlichkeit

Effizienz (Efficiency) Verhältnis zwischen dem Leistungsniveau der Software und dem Umfang der eingesetzten Betriebsmittel unter festgelegten Bedingungen.

Teilmerkmale: Verbrauchsverhalten, Zeitverhalten (Performanz)

Änderbarkeit⁵ (Maintainability) Aufwand, der zur Durchführung vorgegebener Änderungen notwendig ist. Änderungen können Korrekturen von Fehlern, Anpassungen an veränderte Anforderungen oder eine veränderte Systemumgebung und die Verbesserung der Wartung einschließen.

Teilmerkmale: Analysierbarkeit, Modifizierbarkeit, Prüfbarkeit, Stabilität

Übertragbarkeit⁶ (Portability) Leichtigkeit, mit der eine Software von einer Umgebung in eine andere übertragen werden kann. Umgebung kann die organisatorische Umgebung, die Hardware-Umgebung oder die Software-Umgebung bedeuten.

Teilmerkmale: Anpassbarkeit, Austauschbarkeit, Installierbarkeit, Koexistenz^[GL]

Konformität ist ein Teilmerkmal, das zusätzlich alle sechs Qualitätsmerkmale besitzen.

Konformität ist der Grad, in dem die Software Normen oder Vereinbarungen zum jeweiligen Qualitätsmerkmal erfüllt.

Neben der ISO/IEC 25000 definiert die DIN EN ISO 9241 („Ergonomie der Mensch-System-Interaktion“) Software-Qualitätsmerkmale. Für den Tester von Anwendungen mit einer Benutzerschnittstelle sind deren Teile 11 „Anforderungen an die Gebrauchstauglichkeit (Allgemeine Leitsätze)“, 110 „Grundsätze der Dialoggestaltung“, 151 „Leitlinien zur Gestaltung von Benutzungsschnittstellen für das World Wide Web“ und 171 „Richtlinien für die Zugänglichkeit von Software“ relevant.⁷

Wenn es darum geht, welche Inhalte auf einer Website erscheinen sollen, wie die Kundenzufriedenheit sichergestellt und die Anwendung optimal bedient werden kann, ist die

³ Definition von Interoperabilität s. Kap. 9.11.

⁴ Definition von Sicherheit s. Kap. 9.10.

⁵ Änderbarkeit und Wartbarkeit werden synonym verwendet.

⁶ Übertragbarkeit und Portabilität werden synonym verwendet.

⁷ Siehe [ISO 9241-11], [ISO 9241-110], [ISO 9241-151], [ISO 9241-171].

ISO 9241 Teil 110 hilfreich. Sie belegt die „Grundsätze der Dialoggestaltung“ mit folgenden Merkmalen:

Aufgabenangemessenheit „Ein interaktives System ist aufgabenangemessen, wenn es den Benutzer unterstützt, seine Arbeitsaufgabe effektiv und effizient zu erledigen, d. h., wenn Funktionalität und Dialog auf den charakteristischen Eigenschaften der Arbeitsaufgabe basieren, anstatt auf der zur Aufgabenerledigung eingesetzten Technologie.“

Eine Webanwendung ist zum Beispiel aufgabenangemessen, wenn keine aufgabenfremden, unnötigen Eingaben gefordert werden, wie z. B. irrelevante, persönliche Angaben zu einer Bestellung.

Die Vermeidung unnötiger Eingaben dient der Aufgabenangemessenheit. Dazu gehört, dass Standardwerte automatisch für den Benutzer gesetzt werden, damit er sie nicht selber eintragen muss, z. B. Tagesdatum oder Ortsname zur eingegebenen Postleitzahl.

Erwartungskonformität „Ein Dialog ist erwartungskonform, wenn er den aus dem Nutzungskontext heraus vorhersehbaren Benutzerbelangen sowie allgemein anerkannten Konventionen entspricht.“

Eine Webanwendung ist erwartungskonform, wenn Funktionscodes und -tasten in allen Masken und Menüs gleich verwendet werden oder wenn unterstrichene Wörter stets *Hyperlinks* sind.

Fehlertoleranz „Ein Dialog ist fehlertolerant, wenn das beabsichtigte Arbeitsergebnis trotz erkennbar fehlerhafter Eingaben entweder mit keinem oder mit minimalem Korrekturaufwand seitens des Benutzers erreicht werden kann.“

Eine Webanwendung ist zum Beispiel fehlertolerant, wenn die Eingaben in ein Formular vor dem Abschicken überprüft werden und nur die fehlerhaften Eingaben erneut erfasst werden müssen.

Oder ein Eingabefeld erkennt eine fehlerhafte Eingabe automatisch und teilt dies dem Benutzer mit. Trotzdem kann der Benutzer seine Arbeit erst einmal fortsetzen.

Individualisierbarkeit „Ein Dialog ist individualisierbar, wenn Benutzer die Mensch-System-Interaktion und die Darstellung von Informationen ändern können, um diese an ihre individuellen Fähigkeiten und Bedürfnisse anzupassen.“

Eine Webanwendung ist zum Beispiel individualisierbar, wenn der Nutzer die Inhalte für seinen persönlichen Newsletter bestimmen oder sein persönliches Benutzerprofil speichern kann. Abschaltbare bzw. erweiterbare Symbolleisten oder Menüs gehören ebenfalls zur Individualisierung.

Lernförderlichkeit „Ein Dialog ist lernförderlich, wenn er den Benutzer beim Erlernen der Nutzung des interaktiven Systems unterstützt und anleitet.“

Durchgängige Konzepte bei der Strukturierung von Dialogen sind lernförderlich, z. B. existiert ein Menüpunkt, in dem alle Funktionen aufgelistet sind, und für jeden Funktionsaufruf wird stets der Anfangsbuchstabe mit der Strg-Taste als Shortcut verwendet.

Eine Webanwendung ist zum Beispiel lernförderlich, wenn eine Guided Tour durch die Anwendung existiert und Testbuchungen vorgenommen werden können.

Selbstbeschreibungsfähigkeit „Ein Dialog ist in dem Maße selbstbeschreibungsfähig, in dem für den Benutzer zu jeder Zeit offensichtlich ist, in welchem Dialog, an welcher Stelle im Dialog er sich befindet, welche Handlungen unternommen werden können und wie diese ausgeführt werden können.“

Eine gute selbstbeschreibende Website gibt dem Benutzer Antworten auf die großen Wo-Fragen:

- Wo komme ich her?
- Wo bin ich?
- Wo kann ich hin?
- Wo finde ich Hilfe?

Auf Websites mit vielen Seiten sind Breadcrumbs^[GL] (Brotkrumen) ein wichtiger Bestandteil der Selbstbeschreibung. EineBreadcrumb-Navigation zeigt dem Nutzer seinen aktuellen Standort in der Webanwendung. Die Brotkrumen reduzieren benutzerfreundlich die Anzahl der Aktionen eines Website-Besuchers, die ihn auf eine Seite einer höheren Ebene bringen – schönen Gruß von Hänsel und Gretel.

Beispiele für Selbstbeschreibung

Ein Beispiel einer selbstbeschreibenden Website findet man auf <https://www.deutschland.de>, wo neben dem aktuellen Standort (Breadcrumb) auf der Website auch die Anzahl der Beiträge auf der aktuellen Webseite angezeigt wird, s. Abb. 2.1.

Eine Webanwendung ist selbstbeschreibend, wenn die Anzahl der Treffer nach einer Suche wie in Abb. 2.2 angezeigt wird oder wenn das Eingabeformat eines Datumfeldes vorgegeben ist.

Startseite > Wirtschaft > Innovation & Technik

67 Beiträge

Abb. 2.1 Breadcrumbs

Abb. 2.2 Anzeige Suchtreffer

Suchergebnisse

Suchen nach:

Webtesting

[suchen]

Suche nach "**webtesting**"

Anzeige der Ergebnisse **1 bis 10** von insgesamt **14**

[Seite 1](#) [Seite 2](#) [Nächste >](#)

Ein reales Beispiel für „Nicht-Selbstbeschreibung“

Und so sollte man es nicht machen: Ich erfasse in einem Webformular meine Telefonnummer mit Leerzeichen nach der Vorwahl <0999 888888>, denn es sind keine Angaben zur Erfassung auf der Seite gemacht. Ich erlebe eine Überraschung:

```
-----
Fehler beim Ändern der Kundendaten

Hinweis: Bitte ergänzen bzw. berichtigen Sie
folgende Angaben:
Fehl-SC = „61“
Fehl-SC-Text = „Schreibweise fehlerhaft“
-----
```

Was habe ich falsch gemacht? Vielleicht das Leerzeichen rausnehmen: <0999888888>? Aha! Jetzt wird mir mitgeteilt, in welchem Format die Telefonnummer eingegeben werden muss:

```
-----
Datenänderung

Hinweis: Bitte ergänzen bzw. berichtigen Sie
folgende Angaben:
Tel. privat ist nicht korrekt (Bitte Format
Vorwahl/Rufnummer verwenden)
-----
```

Wie schön wäre es doch, stünde im Label „Tel. privat (Vorwahl/Rufnummer)“ oder wären zwei separate Felder für Vorwahl und Rufnummer festgelegt.

Steuerbarkeit „Ein Dialog ist steuerbar, wenn der Benutzer in der Lage ist, den Dialogablauf zu starten sowie seine Richtung und Geschwindigkeit zu beeinflussen, bis das Ziel erreicht ist.“

Eine Webanwendung ist zum Beispiel steuerbar, wenn für Suchvorgänge die Anzahl der anzuzeigenden Treffer pro Seite festgelegt, Download-Prozesse unterbrochen und Funktionen mit Tastenkürzeln statt über Menüs aufgerufen werden können.

Ein Beispiel für Steuerbarkeit

Ein positives Beispiel für Steuerbarkeit zeigt Abb. 2.3: Steuerbarkeit. Dort kann der Benutzer unter <Optionen> – <Sucheinstellungen> unter anderem selbst festlegen, ob Sucheingaben vervollständigt werden, wie viele Treffer pro Seite und wo sie angezeigt werden sollen.

Abb. 2.3 Steuerbarkeit.

(Quelle: www.google.de/preferences?hl=de)

☒ Sofort-Ergebnisse nie anzeigen

Ergebnisse pro Seite

10 20 30 40 50 100
Schneller Langsamer

Öffnen von Ergebnissen

☒ Jedes ausgewählte Ergebnis in einem neuen Browserfenster öffnen

Gebrauchstauglichkeit Die Gebrauchstauglichkeit (Usability) einer Anwendung wird in der ISO 9241 – Teil 11 [ISO 9241-11] definiert:

„Das Ausmaß, in dem ein Produkt durch bestimmte Benutzer in einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und zufriedenstellend zu erreichen.“

Und weiter erklärt die Norm

- **Effektivität** als die Genauigkeit und Vollständigkeit, mit der Benutzer ein bestimmtes Ziel erreichen,
- **Effizienz** als den im Verhältnis zur Genauigkeit und Vollständigkeit eingesetzten Aufwand, mit dem Benutzer ein bestimmtes Ziel erreichen,
- **Zufriedenstellung** als die Freiheit von Beeinträchtigung und positive Einstellung gegenüber der Nutzung des Produkts.

Die aufgeführten Anforderungen an die Dialoggestaltung einer Software und die Gebrauchstauglichkeit sind dem Qualitätsmerkmal Benutzbarkeit zuzuordnen. Ihnen wird insbesondere im Usability-Test (s. Kap. 10.3) Rechnung getragen.

Ein Beispiel für Gebrauchsuntauglichkeit

Nicht gerade gebrauchstauglich ist der Onlineshop (m) eines Hardware-Händlers:

Ich interessiere mich für Monitore. In die engere Auswahl kommen drei Exemplare. Damit ich diese schnell wiederfinde, möchte ich sie mir in einer Wunschliste speichern. Ich weiß, dass es die Wunschliste gibt, ich kann mich noch vom letzten Einkauf daran erinnern. Ich suche vergeblich den entsprechenden Button; erst nach langem Rumprobieren finde ich heraus, dass der Weg zur Wunschliste über den Warenkorb führt.

Also: jeden der drei Artikel in den Warenkorb stellen und von dort aus in die Wunschliste. Effizienz ist was anderes! Aber der Onlineshop hat noch ein paar Macken:

- Das Passwort für das Login zum Onlineshop ist nicht im Firefox-Passwortmanager abrufbar, obwohl ich es dort gespeichert habe. Ist das aufgabenangemessen?

- Nachdem ich dann doch mein Passwort manuell erfasst habe und angemeldet bin, kann ich mich nicht so einfach wieder abmelden, denn der Button zum <Abmelden> befindet sich nicht auf jeder Webseite. Nein, den gibt es nur auf der Seite „Mein Konto“ und die muss ich erst mit der Funktion <Anmelden/Registrieren> aufrufen. Da musst Du drauf kommen!
 - Und um das noch zu toppen ist die Funktion <Anmelden/Registrieren> auf jeder Seite am unteren Ende... und die Seiten sind gaaanz laaaang. Das will „erscrollt“ sein!
- ☹ Zufriedenheit strahle ich nach diesem Einkaufsvorgang nicht aus...

2.2.2 Spezielle Qualitätsmerkmale

Die bisher beschriebenen Qualitätsmerkmale müssen für jede Applikation getestet werden, sei es im Großrechner-, im Client-/Server-, Web- oder Mobile-Umfeld. Für web-basierte und mobile Anwendungen kommen zusätzliche Qualitätsmerkmale zum Tragen.

Eine Website muss den zusätzlichen Anforderungen der Auffindbarkeit, Barrierefreiheit und Rechtskonformität genügen. Diese drei Qualitätsmerkmale werden der in der DIN 66272 [DIN 66272] geforderten Benutzbarkeit zugeordnet und dementsprechend durch die im Kap. 10 beschriebenen Testarten geprüft.

Für eine mobile Anwendung, sei es eine Mobile-Webseite oder eine Mobile-App, müssen die beiden spezifischen Qualitätsmerkmale Mobiltauglichkeit und Mobilfähigkeit eingefordert werden.

Auffindbarkeit Eine Website muss leicht auffindbar sein. Das bedeutet zum einen, dass die Webadresse einen sprechenden, leicht zu merkenden Namen hat und zum anderen, dass sie von Suchmaschinen unter den Top-Treffern angezeigt wird.

Der Auffindbarkeitstest (s. Kap. 10.5) stellt die Auffindbarkeit einer Website sicher.

Barrierefreiheit (Accessibility) Soll eine Webanwendung auch für behinderte Personen gebrauchstauglich sein, so muss sie barrierefrei realisiert werden. Barrierefreiheit nach § 4 des Behindertengleichstellungsgesetzes (BGG) ist wie folgt definiert (s. [URL: BGG]):

„Barrierefrei sind bauliche und sonstige Anlagen, Verkehrsmittel, technische Gebrauchsgegenstände, Systeme der Informationsverarbeitung, akustische und visuelle Informationsquellen und Kommunikationseinrichtungen sowie andere gestaltete Lebensbereiche, wenn sie für behinderte Menschen in der allgemein üblichen Weise, ohne besondere Erschwernis und grundsätzlich ohne fremde Hilfe zugänglich und nutzbar sind.“

Barrierefreiheit (Accessibility) entspricht somit dem in der ISO 9241 definierten Qualitätsmerkmal Gebrauchstauglichkeit (Usability) für einen Benutzerkreis, der behinderte Menschen mit einschließt.

Handbuch zum Testen von Web- und Mobile-Apps

Testverfahren, Werkzeuge, Praxistipps

Franz, K.

2015, XXI, 303 S. 56 Abb., Hardcover

ISBN: 978-3-662-44027-8