


```
align-items: stretch|flex-start|flex-end|center|baseline|
first baseline|last baseline|start|end|self-start|self-end;
align-content: flex-start|flex-end|center|space-
between|space-around|space-evenly|stretch|start|end|
baseline|first baseline|last baseline; }
.item { order: 5; /* def. 0 */ flex-grow: 4; /* def. 0 */
flex-shrink: 3; /* def. 0 */
flex-basis: | auto; /* def. auto */
flex: none | grow shrink basis; align-self: auto|
flex-start|flex-end|center|baseline|stretch; }
```

Grid: 2-D Raster zur flexiblen Anordnung von Elementen

```
.container { display: grid|inline-grid;
grid-template-columns: 50px auto repeat(5, 1fr);
grid-template-rows: min-content 5px 1fr max-content;
grid-template-areas: "header header" "main . sidebar"
"footer footer";
grid-template: none|template-columns|template-rows;
column-gap: 1px; row-gap: 1px; gap: <row> <column>;
justify-items: start | end | center | stretch;
align-items: start | end | center | stretch;
justify-content: start|end|center|stretch|space-around|
space-between|space-evenly;
align-content: start|end|center|stretch|space-around|
space-between|space-evenly;
grid-auto-columns: 3px 4px ...;
grid-auto-rows: 3px 4px ...; }
.item { grid-area: header<rows><cols><rowe><cole>;
grid-column: <start> / <end> | <start> / span <value>;
grid-row: <start> / <end> | <start> / span <value>;
justify-self: start | end | center | stretch;
align-self: start | end | center | stretch; }
```

5 Accessibility

Rechtlich: UNO-Behindertenrechtskonvention (UNO-BRK), gleichberechtigte Möglichkeiten im Zugang & Benutzung von Informations- & Kommunikationssystemen
Häufigste Fehler: Text zu wenig Kontrast, kein alt Text, leere Links/Buttons/Doc-Sprache, fehlende Form Labels
Farbenblindheit: prüfen mit Dev-Tools (Rendering), Doppel-Codierung notwendig (Farbe & Form/Icon/Text), kritische Informationen nicht nur mit Text codieren, farbenblind-freundliche Farbpaletten wählen
Farbkontrast: prüfen mit Dev-Tools (Color Picker) & Lighthouse / Firefox Report, Zielgruppe erweitern mit WCAG Kontrast Level AA (5.7:1, 50+), AAA (15.9:1, 80+)
Zoombarkeit: Zoom nicht unterbinden, Falsch: user-scalable=0/no, maximum-scale=1
Animationen: sollten abstellbar sein, Verringerung der Ablenkung bei Konzentrationstörung, Verhinderung der Auslösung von Epilepsie und Migräne

```
@media (prefers-reduced-motion: reduce) { ... }
<link rel="stylesheet" href="animations.css" media="(prefers-reduced-motion: no-preference)">
```

Tastatur-Bedienbarkeit: alle wichtigen Input-Elemente sind in der richtigen Reihenfolge fokussierbar, verboten: float, flex-direction: *-reverse, Standard Input Controls nutzen: a. button, input[type="..."], textarea, zusätzliche Elemente können mit "tabindex=0" fokussierbar gemacht werden, Elemente jeder Seite können mit "autofocus" versehen werden

Screenreader-Optimierung: keine Headings Level auslassen, semantische Elemente richtig nutzen, Skip-Links am Anfang der Site, lang Attribut korrekt setzen

ARIA Attribute: lösen Accessibility Issues, die nicht mit nativem HTML gelöst werden können, heissen jeweils aria-*, wichtigste: hidden, label, expanded, current, labelledby, controls, haspopup, invalid, live, required
Tabellen: Heading für Rows & Columns mit <th> ausstatten, alternativ <td role="columnheader/rowheader">

Allgemeines: alt Tag bei Bildern, leer wenn dekorativ, logische Reihenfolge auch ohne CSS, semantische Struktur, Multimedia: 2-Sinne Prinzip, Anzeige-Art kann angepasst werden (Textgrösse, Animationen, Zeitlimits), Überschriften/Form Labels/Linktexte sind verständlich, auf Formatwechsel (z.B. PDF) wird hingewiesen, konsistente Navigation, valides HTML, kompatibel mit versch. I/O Geräten, Hilfestellung bei Interaktionen, PDFs auf PDF/UA Konformität validieren, keine Tabellen für Layout

6 Security

Stored XSS: Server kann so manipuliert werden, dass JS Code an die Browser der Opfer ausgeliefert & ausgeführt wird, **Massnahmen:** User-Input muss beim Output encodet werden, z.B. mit 3x {} in Handlebars, oder User-Input wird sanitized, z.B. mit xss oder dompurify Library
JS Injection: dem Server kann JS Code gesendet werden, welcher ihn im Code mit eval()/setTimeout()/setInterval()/Function() ausführt, **Massnahmen:** stattdessen parseXXX()/JSON.parse() verwenden, globale Scopes & Variablen reduzieren, rechen-intensive Tasks mit child-process.spawn auslagern, Node kein Root-Prozess
Insecure direct object references: durch Manipulation der URL lassen sich sensible Daten ohne Authentifizierung oder Autorisierung aufrufen, **Massnahmen:** bei allen Sites mit sensiblen Daten sicherstellen, dass der User eingeloggt und berechtigt ist

```
const u = findUser(req.body.name);
req.session.user_id = u.id; // if user exists & PW correct
function login(req, res, next) {
  if user.isAuthorized(req.session.user_id, req.url) {
    next(); } else { res.send(401, "not authorized"); } }
app.all("/api/*", login, function(req, res, next) { next(); });
```

Cross Site Request Forgery: Angreifer bringt Nutzer mit gültiger Session dazu, auf einer Site ein gefälschtes Formular an die richtige Site zu submitten, **Massnahmen:** Formular-Submission-Requests (GET/POST) müssen geprüft werden, ob sie von einem vom Server frisch ausgelieferten Formular stammen (mit CSRF Token)

```
const csrf = require("tiny-csrf");
app.use(csrf({ cookie: true }));
router.route("/form").get(function(req, res, next) {
  res.render("form", { csrfToken: req.csrfToken() }); })
.post(function(req, res, next) {
  if(checkAndRecordUnseenCSRFToken(req.body._csrf)){
    res.send("good")} else { res.send("bad") } });
```

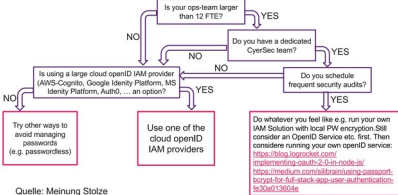
Replay attacks: z.B. bei Quiz wird mit Formular Aufgabe und User-Input-Lösung submitted, wenn richtig gibt es einen Punkt, das kann replayed werden, sollte aber nicht gehen, da der User dann unrechtmässig Punkte erhält, **Massnahmen:** CSRF Token 1x gültig, bereits gelöste Aufgaben speichern, letzte Aufgabe in Session speichern
Cryptographic failures: Crypto schlecht eingesetzt, **Massnahmen:** https nutzen, keine geheimen Infos in Query-Params (Leaks in Server Logs oder Browser History), Authentication Service nutzen

Identification & Authentication failures: z.B. ist Session Timeout zu lang, User bleibt auf öffentlichem PC eingeloggt, **Massnahme:** Session Timeout sinnvoll setzen
CORS Header: ermöglicht Laden von Ressourcen (Bilder, Scripts, usw.) von anderem Ursprungsort als eigener Origin, man beschränkt, welche Websites auf seine Ressourcen zugreifen dürfen

Access-Control-Allow-Origin: https://test.ch | *
Access-Control-Allow-Methods: GET, POST

CSP Header: Content Security Policy, Beispiel schränkt Kommunikation auf Source Domain/api.ch ein, externe & Inline-Scripts sind verboten, aktivieren mit unsafe-inline
Content-Security-Policy: default-src 'self' *.api.ch

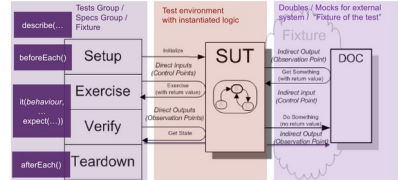
Weiteres: DNS Prefetching & Client Caching deaktivieren, Public Key Pinning Headers, Strict-Transport-Security Header (HSTS) setzen, X-XSS-Protection Header, Safe-Regex verwenden (DDoS), Parameter pollution
Cookies: Secure Attribut: Cookie wird nur über HTTPS versendet, nicht HTTP (ausser localhost), HttpOnly Attribut: deaktiviert Zugriff auf Cookies aus JS



Quelle: Meinung Stölze

7 Testing

Unit Test: einzelne Klassen/Module (meist ein File) werden getestet, einfache Automation, **Herausforderungen:** Units isolieren, asynchrone Operationen, Testdaten-Generierung (z.B. Fuzzing)



Integration Test: Zusammenspiel von 2 oder mehr Units wird getestet, Automation meist möglich, **Herausforderungen:** gleiche wie bei Unit Tests, Simulation Browser & Benutzerinteraktion, Test mit Datenbank

End-to-End Test: Integration Test über alle Layer
Anders: Smoke, Static (TypeScript), Regression, Load, Performance, Endurance, Chaos, Security, Usability

Test-Runner: Rahmen, der Tests entgegennimmt, ausführt und Resultate anzeigt, z.B. Mocha, Cypress
Assertion Library: gut lesbare Annahmen schreiben, die nicht verletzt werden sollten, z.B. Assert, Chai

Mocking Library: Separierung von Units, Erstellung von Test-Doubles insb. Mocks, z.B. Proxyquire, Sinon.js

```
const { expect } = require("chai");
describe('Array', function() {
  describe('#indexOf()', function() {
    beforeEach(function() { this.testArray = [1, 2, 3]; });
    it('should return -1 when the value is not present', function() {
      expect(this.testArray.indexOf(4)).to.equal(-1); }); }); });
```

Test Double Pattern: Bei Unit Tests hat man zum Teil ein Dependent On Component (DOC) an der Unit, mit Sin kann man einen vorbestimmten Input (Stub/Fake) fürs SUT und überprüfbaren Output (Spy/Mock) liefern
Eigenschaften guter Unit Tests: Output statt Internes testen, jeder Test hat voraussagbare & deklarative Struktur, nur eine Aktion und der Output davon testen, keine Daten mit anderen Tests teilen, alles Nötige zum Verstehen der Absicht muss im Test sein

Unit Test Smells: Hard-to-Test Code, viele Bugs bei formalem Testing & Produktion, Tests fallen wenn SUTs so

verändert werden, dass die Tests nicht beeinflusst werden sollten, Mal-so-mal-so Tests, Conditionals in Tests, Test-Code Duplikate, langsame Tests, schwierig zu verstehende Tests, Test-Logik im Prod-Code, schwer zu sagen welches Assert im Test failed, keine Tests

8 Internationalization

Internationalisierung: Software-Entwicklungsmethode, damit alle Benutzerausgaben per Spracheinstellung austauschbar und nicht hard-coded sind
Lokalisierung: Inhalt den sprachlichen & kulturellen Eigenheiten bestimmter Zielkulturen/Länder anpassen
Übersetzung: Inhalt übersetzen, Arbeit der Übersetzer
Locale: String, der Region (Sprache & Land) bestimmt
Unterschiede Sprachregionen (nicht halbautomatisch anpassbar): kulturelle Unterschiede (Bilder, Symbole, Farben), regulatorische Anforderungen (Daten-/Verbraucherschutz), sprachliche Feinheiten (Dialekte, Idiome, Redewendungen), UX und Usability (Navigation, Benutzerführung, Leserichtung), Markt- & zielgruppenspezifische Anpassungen (Produktangebote, Werbung, Marketing), kulturelle Sensibilitäten (politische & soziale Themen, Feiertage), Titel & Anreden, anderes Konzept als Vorname/Nachname, Schreibweise von Adressen & Telefonnummern, Satzzeichen, Masseneinheiten (Umrechnung), Kalender, Sounds, Layout, Steuern

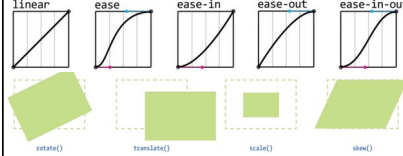
```
const rn = new Intl.DisplayNames(['de'], {type: 'region'});
console.log(regionNames.of('CH'));
const cg = new Intl.Collator('de').compare('ä', 'z');
c.log((cg)); // negativ, 'ä' kommt nach 'z'
const d = new Date(Date.UTC(2022, 4, 17, 13, 30, 0));
c.log(new Intl.DateTimeFormat('en-US').format(d));
const rtf_english = new Intl.RelativeTimeFormat('en', {
  localeMatcher: "best fit", // oder: "lookup"
  numeric: "always", // oder: "auto"
  style: "long" }); // oder: "short" / "narrow"
c.log(rtf_english.format(-1, 'day'));
const m = 123456.789;
c.log(new Intl.NumberFormat('de-CH').format(m));
const colors = [ 'red', 'blue', 'pink', 'orange' ];
const formatter = new Intl.ListFormat('en', {
  style: 'long', type: 'conjunction' });
console.log(formatter.format(colors));
const w = new Intl.PluralRules('en-US', {type: "ordinal"});
console.log(w.select(1)); // "one"
const deCHLocale = new Intl.Locale('de-CH');
```

9 Animation

Most properties that accept a length, number, color, or the function calc() can be animated. Most properties that take a keyword or other discrete values, like url(), can't.

transition-property: name z.B. color | all | none;
transition-duration: 3s | 300ms;
transition-timing-function: linear | ease-in | ease-out | ease-in-out;
transition-delay: 0.3s | 300ms;
transition: property duration [timing-function delay] [...];
transform: rotate(-3deg) | rotateX(-3deg) | rotateY(-3deg) | translateX(3px) | translateY(-3px) | translate(x, y) | scaleX(1.3) | scaleY(0.7) | scale(x, y) | skewX(3deg) | skewY(-3deg) | skew(x, y) | none [...]
transform-origin: % px. left, right, top, bottom, center;

@property: Initialwert, Interpolation, Type Safety



```
@keyframes wiggleAnimation {
  { transform: skewX(9deg); }
  { transform: skewX(-9deg); }
  % { transform: skewX(0deg); } }
.wiggle { animation-name: wiggleAnimation;
  animation-duration: 1s; }
```

10 User Experience

Don't listen to users: Schauen was User machen, nicht was sie sagen, Kunden fragen, User nicht, **Umfragen**
Problem Space: User analysieren, beschreiben, haben Bedürfnisse und Probleme

Solution Space: Designer, Vision, Storyboard, Prototyp

Problem / Future Scenario: User, Problem Beschreibung, Kontext, Trigger, Schritte, Lösung / Fail

Gute UI Ausschilderung: Wo bin ich? (Titel, Breadcrumbs) Wo kann ich hin? Was ist passiert? **Kofferraum**

Concept Model: Benutzer, Ressource und Beziehungen

Site Map: Seitenhierarchie (Baum) wie in der Navigation

Card Sort: Cards mit allen Seiten erstellen, Cars gruppieren & Gruppen benennen (Open), Gruppennamen validieren (Closed), 1 Hierarchiestufe

Tree Testing: Site Map aufnehmen, Szenarios zur Erreichung von Zielen stellen, **Wireframe:** Screen-Skizze

Screen Flow: Wireframe-Abfolge, zeigt User-Szenario

Gute Test-Szenarien: plausible Ziele, Kontext, Trigger, Status, enge Bezug zwischen System und realer Welt, Nutzerkontrolle und Freiheit, Konsistenz & Konformität mit Standards, Fehler-Vorbeugung, besser Sichtbarkeit als sich-erinnern-müssen, Flexibilität und Nutzungseffizienz, Ästhetik und minimalistischer Aufbau, Nutzern helfen, Fehler zu bemerken, zu diagnostizieren und zu beheben, Hilfe und Dokumentation

Messen: Effektivität – User können Ziele erreichen, **Effizienz** – angemessener Aufwand zur Zielerreichung, **Zufriedenheit:** positive Einstellung gegenüber System

Usability Kriterien Nielsen: Sichtbarkeit des System-Status, enge Bezug zwischen System und realer Welt, Nutzerkontrolle und Freiheit, Konsistenz & Konformität mit Standards, Fehler-Vorbeugung, besser Sichtbarkeit als sich-erinnern-müssen, Flexibilität und Nutzungseffizienz, Ästhetik und minimalistischer Aufbau, Nutzern helfen, Fehler zu bemerken, zu diagnostizieren und zu beheben, Hilfe und Dokumentation

Messen: Effektivität – User können Ziele erreichen, **Effizienz** – angemessener Aufwand zur Zielerreichung, **Zufriedenheit:** positive Einstellung gegenüber System

Usability Kriterien Nielsen: Sichtbarkeit des System-Status, enge Bezug zwischen System und realer Welt, Nutzerkontrolle und Freiheit, Konsistenz & Konformität mit Standards, Fehler-Vorbeugung, besser Sichtbarkeit als sich-erinnern-müssen, Flexibilität und Nutzungseffizienz, Ästhetik und minimalistischer Aufbau, Nutzern helfen, Fehler zu bemerken, zu diagnostizieren und zu beheben, Hilfe und Dokumentation

Messen: Effektivität – User können Ziele erreichen, **Effizienz** – angemessener Aufwand zur Zielerreichung, **Zufriedenheit:** positive Einstellung gegenüber System

Usability Kriterien Nielsen: Sichtbarkeit des System-Status, enge Bezug zwischen System und realer Welt, Nutzerkontrolle und Freiheit, Konsistenz & Konformität mit Standards, Fehler-Vorbeugung, besser Sichtbarkeit als sich-erinnern-müssen, Flexibilität und Nutzungseffizienz, Ästhetik und minimalistischer Aufbau, Nutzern helfen, Fehler zu bemerken, zu diagnostizieren und zu beheben, Hilfe und Dokumentation

Messen: Effektivität – User können Ziele erreichen, **Effizienz** – angemessener Aufwand zur Zielerreichung, **Zufriedenheit:** positive Einstellung gegenüber System

Usability Kriterien Nielsen: Sichtbarkeit des System-Status, enge Bezug zwischen System und realer Welt, Nutzerkontrolle und Freiheit, Konsistenz & Konformität mit Standards, Fehler-Vorbeugung, besser Sichtbarkeit als sich-erinnern-müssen, Flexibilität und Nutzungseffizienz, Ästhetik und minimalistischer Aufbau, Nutzern helfen, Fehler zu bemerken, zu diagnostizieren und zu beheben, Hilfe und Dokumentation

Messen: Effektivität – User können Ziele erreichen, **Effizienz** – angemessener Aufwand zur Zielerreichung, **Zufriedenheit:** positive Einstellung gegenüber System

Usability Kriterien Nielsen: Sichtbarkeit des System-Status, enge Bezug zwischen System und realer Welt, Nutzerkontrolle und Freiheit, Konsistenz & Konformität mit Standards, Fehler-Vorbeugung, besser Sichtbarkeit als sich-erinnern-müssen, Flexibilität und Nutzungseffizienz, Ästhetik und minimalistischer Aufbau, Nutzern helfen, Fehler zu bemerken, zu diagnostizieren und zu beheben, Hilfe und Dokumentation

Messen: Effektivität – User können Ziele erreichen, **Effizienz** – angemessener Aufwand zur Zielerreichung, **Zufriedenheit:** positive Einstellung gegenüber System

Usability Kriterien Nielsen: Sichtbarkeit des System-Status, enge Bezug zwischen System und realer Welt, Nutzerkontrolle und Freiheit, Konsistenz & Konformität mit Standards, Fehler-Vorbeugung, besser Sichtbarkeit als sich-erinnern-müssen, Flexibilität und Nutzungseffizienz, Ästhetik und minimalistischer Aufbau, Nutzern helfen, Fehler zu bemerken, zu diagnostizieren und zu beheben, Hilfe und Dokumentation

Messen: Effektivität – User können Ziele erreichen, **Effizienz** – angemessener Aufwand zur Zielerreichung, **Zufriedenheit:** positive Einstellung gegenüber System

Usability Kriterien Nielsen: Sichtbarkeit des System-Status, enge Bezug zwischen System und realer Welt, Nutzerkontrolle und Freiheit, Konsistenz & Konformität mit Standards, Fehler-Vorbeugung, besser Sichtbarkeit als sich-erinnern-müssen, Flexibilität und Nutzungseffizienz, Ästhetik und minimalistischer Aufbau, Nutzern helfen, Fehler zu bemerken, zu diagnostizieren und zu beheben, Hilfe und Dokumentation

Messen: Effektivität – User können Ziele erreichen, **Effizienz** – angemessener Aufwand zur Zielerreichung, **Zufriedenheit:** positive Einstellung gegenüber System

Usability Kriterien Nielsen: Sichtbarkeit des System-Status, enge Bezug zwischen System und realer Welt, Nutzerkontrolle und Freiheit, Konsistenz & Konformität mit Standards, Fehler-Vorbeugung, besser Sichtbarkeit als sich-erinnern-müssen, Flexibilität und Nutzungseffizienz, Ästhetik und minimalistischer Aufbau, Nutzern helfen, Fehler zu bemerken, zu diagnostizieren und zu beheben, Hilfe und Dokumentation

Messen: Effektivität – User können Ziele erreichen, **Effizienz** – angemessener Aufwand zur Zielerreichung, **Zufriedenheit:** positive Einstellung gegenüber System

Usability Kriterien Nielsen: Sichtbarkeit des System-Status, enge Bezug zwischen System und realer Welt, Nutzerkontrolle und Freiheit, Konsistenz & Konformität mit Standards, Fehler-Vorbeugung, besser Sichtbarkeit als sich-erinnern-müssen, Flexibilität und Nutzungseffizienz, Ästhetik und minimalistischer Aufbau, Nutzern helfen, Fehler zu bemerken, zu diagnostizieren und zu beheben, Hilfe und Dokumentation

Messen: Effektivität – User können Ziele erreichen, **Effizienz** – angemessener Aufwand zur Zielerreichung, **Zufriedenheit:** positive Einstellung gegenüber System

Usability Kriterien Nielsen: Sichtbarkeit des System-Status, enge Bezug zwischen System und realer Welt, Nutzerkontrolle und Freiheit, Konsistenz & Konformität mit Standards, Fehler-Vorbeugung, besser Sichtbarkeit als sich-erinnern-müssen, Flexibilität und Nutzungseffizienz, Ästhetik und minimalistischer Aufbau, Nutzern helfen, Fehler zu bemerken, zu diagnostizieren und zu beheben, Hilfe und Dokumentation

Messen: Effektivität – User können Ziele erreichen, **Effizienz** – angemessener Aufwand zur Zielerreichung, **Zufriedenheit:** positive Einstellung gegenüber System

Usability Kriterien Nielsen: Sichtbarkeit des System-Status, enge Bezug zwischen System und realer Welt, Nutzerkontrolle und Freiheit, Konsistenz & Konformität mit Standards, Fehler-Vorbeugung, besser Sichtbarkeit als sich-erinnern-müssen, Flexibilität und Nutzungseffizienz, Ästhetik und minimalistischer Aufbau, Nutzern helfen, Fehler zu bemerken, zu diagnostizieren und zu beheben, Hilfe und Dokumentation

Messen: Effektivität – User können Ziele erreichen, **Effizienz** – angemessener Aufwand zur Zielerreichung, **Zufriedenheit:** positive Einstellung gegenüber System

Usability Kriterien Nielsen: Sichtbarkeit des System-Status, enge Bezug zwischen System und realer Welt, Nutzerkontrolle und Freiheit, Konsistenz & Konformität mit Standards, Fehler-Vorbeugung, besser Sichtbarkeit als sich-erinnern-müssen, Flexibilität und Nutzungseffizienz, Ästhetik und minimalistischer Aufbau, Nutzern helfen, Fehler zu bemerken, zu diagnostizieren und zu beheben, Hilfe und Dokumentation

Messen: Effektivität – User können Ziele erreichen, **Effizienz** – angemessener Aufwand zur Zielerreichung, **Zufriedenheit:** positive Einstellung gegenüber System

Usability Kriterien Nielsen: Sichtbarkeit des System-Status, enge Bezug zwischen System und realer Welt, Nutzerkontrolle und Freiheit, Konsistenz & Konformität mit Standards, Fehler-Vorbeugung, besser Sichtbarkeit als sich-erinnern-müssen, Flexibilität und Nutzungseffizienz, Ästhetik und minimalistischer Aufbau, Nutzern helfen, Fehler zu bemerken, zu diagnostizieren und zu beheben, Hilfe und Dokumentation

Messen: Effektivität – User können Ziele erreichen, **Effizienz** – angemessener Aufwand zur Zielerreichung, **Zufriedenheit:** positive Einstellung gegenüber System

Usability Kriterien Nielsen: Sichtbarkeit des System-Status, enge Bezug zwischen System und realer Welt, Nutzerkontrolle und Freiheit, Konsistenz & Konformität mit Standards, Fehler-Vorbeugung, besser Sichtbarkeit als sich-erinnern-müssen, Flexibilität und Nutzungseffizienz, Ästhetik und minimalistischer Aufbau, Nutzern helfen, Fehler zu bemerken, zu diagnostizieren und zu beheben, Hilfe und Dokumentation

Messen: Effektivität – User können Ziele erreichen, **Effizienz** – angemessener Aufwand zur Zielerreichung, **Zufriedenheit:** positive Einstellung gegenüber System

Usability Kriterien Nielsen: Sichtbarkeit des System-Status, enge Bezug zwischen System und realer Welt, Nutzerkontrolle und Freiheit, Konsistenz & Konformität mit Standards, Fehler-Vorbeugung, besser Sichtbarkeit als sich-erinnern-müssen, Flexibilität und Nutzungseffizienz, Ästhetik und minimalistischer Aufbau, Nutzern helfen, Fehler zu bemerken, zu diagnostizieren und zu beheben, Hilfe und Dokumentation

Messen: Effektivität – User können Ziele erreichen, **Effizienz** – angemessener Aufwand zur Zielerreichung, **Zufriedenheit:** positive Einstellung gegenüber System

Usability Kriterien Nielsen: Sichtbarkeit des System-Status, enge Bezug zwischen System und realer Welt, Nutzerkontrolle und Freiheit, Konsistenz & Konformität mit Standards, Fehler-Vorbeugung, besser Sichtbarkeit als sich-erinnern-müssen, Flexibilität und Nutzungseffizienz, Ästhetik und minimalistischer Aufbau, Nutzern helfen, Fehler zu bemerken, zu diagnostizieren und zu beheben, Hilfe und Dokumentation

Messen: Effektivität – User können Ziele erreichen, **Effizienz** – angemessener Aufwand zur Zielerreichung, **Zufriedenheit:** positive Einstellung gegenüber System

Usability Kriterien Nielsen: Sichtbarkeit des System-Status, enge Bezug zwischen System und realer Welt, Nutzerkontrolle und Freiheit, Konsistenz & Konformität mit Standards, Fehler-Vorbeugung, besser Sichtbarkeit als sich-erinnern-müssen, Flexibilität und Nutzungseffizienz, Ästhetik und minimalistischer Aufbau, Nutzern helfen, Fehler zu bemerken, zu diagnostizieren und zu beheben, Hilfe und Dokumentation

Messen: Effektivität – User können Ziele erreichen, **Effizienz** – angemessener Aufwand zur Zielerreichung, **Zufriedenheit:** positive Einstellung gegenüber System

Usability Kriterien Nielsen: Sichtbarkeit des System-Status, enge Bezug zwischen System und realer Welt, Nutzerkontrolle und Freiheit, Konsistenz & Konformität mit Standards, Fehler-Vorbeugung, besser Sichtbarkeit als sich-erinnern-müssen, Flexibilität und Nutzungseffizienz, Ästhetik und minimalistischer Aufbau, Nutzern helfen, Fehler zu bemerken, zu diagnostizieren und zu beheben, Hilfe und Dokumentation

Messen: Effektivität – User können Ziele erreichen, **Effizienz** – angemessener Aufwand zur Zielerreichung, **Zufriedenheit:** positive Einstellung gegenüber System

Usability Kriterien Nielsen: Sichtbarkeit des System-Status, enge Bezug zwischen System und realer Welt, Nutzerkontrolle und Freiheit, Konsistenz & Konformität mit Standards, Fehler-Vorbeugung, besser Sichtbarkeit als sich-erinnern-müssen, Flexibilität und Nutzungseffizienz, Ästhetik und minimalistischer Aufbau, Nutzern helfen, Fehler zu bemerken, zu diagnostizieren und zu beheben, Hilfe und Dokumentation

Messen: Effektivität – User können Ziele erreichen, **Effizienz** – angemessener Aufwand zur Zielerreichung, **Zufriedenheit:** positive Einstellung gegenüber System

Usability Kriterien Nielsen: Sichtbarkeit des System-Status, enge Bezug zwischen System und realer Welt, Nutzerkontrolle und Freiheit, Konsistenz & Konformität mit Standards, Fehler-Vorbeugung, besser Sichtbarkeit als sich-erinnern-müssen, Flexibilität und Nutzungseffizienz, Ästhetik und minimalistischer Aufbau, Nutzern helfen, Fehler zu bemerken, zu diagnostizieren und zu beheben, Hilfe und Dokumentation