

1. Sichtbarkeit des Systemzustands
 - a. System informiert Nutzer darüber, was gerade passiert
 - b. Rückmeldungen erfolgen schnell und sind eindeutig
2. Übereinstimmung zwischen System und Realwelt
 - a. System verwendet bekannte Konzepte (z.B. „Papierkorb“ für Dateien)
 - b. System verwendet bekanntes Vokabular
 - c. Technische Begriffe und Fehlercodes bleiben verborgen
3. Benutzerkontrolle und Freiheit
 - a. Nutzer haben die Möglichkeit Dialoge zu verlassen und Fehler zu korrigieren
 - b. Fälschlicherweise aufgerufene Dialoge können abgebrochen werden
 - c. Aktionen können rückgängig gemacht und wiederholt werden
4. Konsistenz und Standards
 - a. Gleiche Begriffe und Aktionen haben in unterschiedlichen Kontexten eine einheitliche Bedeutung
 - b. Plattform-Konventionen werden eingehalten
5. Fehlerprävention
 - a. Typischen Fehlern (z.B. bei der Eingabe von Werten) wird vorgebeugt (z.B. durch Information über zulässiges Format; sofortige Validierung)
 - b. Nutzer bestätigt Durchführung kritischer Aktionen
6. Erkennen statt Erinnern
 - a. Gedächtnisleistung minimieren; Objekte und Optionen sind sichtbar
 - b. Informationen, die an anderer Stelle wieder gebraucht werden, bleiben sichtbar
7. Flexibilität und Effizienz
 - a. Erfahrene Nutzer können mit Shortcuts schneller ans Ziel kommen
 - b. Häufig durchgeführte Aktionen können angepasst oder verkürzt werden
8. Ästhetik und minimalistisches Design
 - a. Dialoge enthalten keine irrelevanten oder selten gebrauchten Informationen
 - b. Wichtige Informationen ‚verstecken‘ sich nicht zwischen unwichtigen
9. Hilf Nutzern beim Erkennen, Diagnostizieren u. Rückgängigmachen von Fehlern
 - a. Fehlermeldungen sind in einfacher Sprache (ohne Fehlercodes) gehalten, beschreiben das Problem präzise und bieten Lösungsvorschläge an
10. Hilfe und Dokumentation
 - a. Wenn nötig, werden Hilfe und Dokumentation angeboten
 - b. Sie sind leicht durchsuchbar, fokussieren sich auf die Aufgabe der Nutzer und listen konkret die zu unternehmenden Schritte auf