

Mensch-Maschine-Kommunikation: UI Design Frameworks & Tools (Kurzversion)

Parameter	Kursinformationen
Veranstaltung:	Mensch-Maschine-Kommunikation
Semester	Sommersemester 2025
Hochschule:	Technische Universität Bergakademie Freiberg
Inhalte:	Frameworks & Tools für UI-Design
Link auf GitHub:	https://github.com/smn-hrtzsch/MMK/blob/main/tools_for_ui_design.md
Autoren	Björn Muchow, Simon Hörtzsch

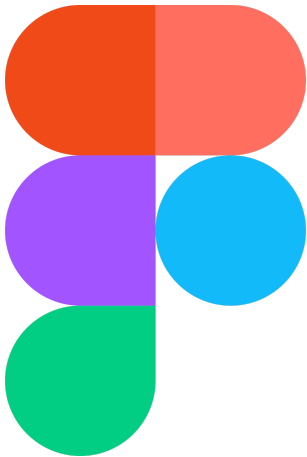
- Gruppenmitglieder: Björn Muchow, Simon Hörtzsch
- Studiengang: Robotik | Angewandte Informatik
- Betreuer: Prof. Dr.-Ing. Bernhard Jung, Florian Richter
- Datum: 25.04.2025

1. Einleitung: Was ist UI-Design?

User Interface (UI) Design: Gestaltung von Benutzeroberflächen für Software & Maschinen. **Ziel:** Einfache, effiziente & intuitive Interaktion. **Wichtigkeit:** Benutzerzufriedenheit, Effizienz, Markenidentität.

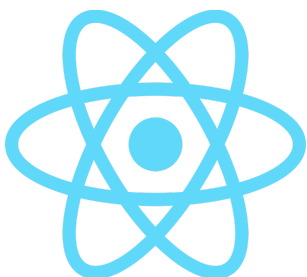


2. Design & Prototyping: Figma



- **Art:** Kollaboratives Design-Tool (Webbasiert)
- **Kosten:** Freemium (Kostenloser Plan + Bezahlpläne)
- **Plattform:** Web, Windows, macOS
- **Vorteile:** Echtzeit-Kollaboration, Plattformunabhängig, viele Plugins, gutes Prototyping.
- **Nachteile:** Internet nötig, kann bei großen Dateien langsam werden.
- **Alternativen:** Sketch (nur macOS, kostenpflichtig), Adobe XD (Entwicklung eingestellt).

3. Web UI Framework: React



- **Art:** JavaScript-Bibliothek (Meta/Facebook)
 - **Kosten:** Open Source (Frei)
 - **Plattform:** Web
 - **Nutzung:** JavaScript/JSX, TypeScript
 - **Vorteile:** Komponenten, große Community, Performance (Virtual DOM), Flexibel.
 - **Nachteile:** Benötigt oft Zusatz-Tools (Routing etc.), Lernkurve.
 - **Alternativen:** Angular (Umfassendes Framework, steilere Lernkurve), Vue.js (Einsteigerfreundlich, kleineres Ökosystem).
 - **Quelle:** [React Website](#)
-

4. Mobile UI Framework: Flutter



- **Art:** Cross-Platform UI-Toolkit (Google)
 - **Kosten:** Open Source (Frei)
 - **Plattform:** iOS, Android, Web, Desktop
 - **Nutzung:** Dart
 - **Vorteile:** Sehr gute Performance, schnelle Entwicklung (Hot Reload), konsistentes UI, umfangreiche Widgets.
 - **Nachteile:** Größere App-Größe, Dart weniger verbreitet.
 - **Alternativen:** Native Entwicklung (SwiftUI/Jetpack Compose - Beste Performance/Integration, aber doppelter Code), React Native (Nutzt native Widgets, JavaScript).
 - **Quelle:** [Flutter Website](#)
-

5. Desktop UI Framework: Electron



- **Art:** Framework für Desktop-Apps mit Web-Technologien (GitHub/Microsoft)
 - **Kosten:** Open Source (Frei)
 - **Plattform:** Windows, macOS, Linux
 - **Nutzung:** HTML, CSS, JavaScript
 - **Vorteile:** Nutzung bekannter Web-Tech, Cross-Platform, große Community (VS Code, Slack etc.).
 - **Nachteile:** Hoher Ressourcenverbrauch, große Installationsdateien.
 - **Alternativen:** Qt (Sehr performant, C++/QML, komplexere Lizenz).
 - **Quelle:** [Electron Website](#)
-

6. Fazit

- **Tool-Wahl:** Hängt stark vom Projekt, Team, Budget und Zielplattform ab.
 - **Trends:** Deklarative UI, Cross-Platform, Design-Systeme, Kollaboration.
 - **Empfehlungen (Kurz):**
 - **Design:** Figma (Kollaboration, Web) - **Web:** React/Vue (Flexibel, Community) - **Mobile:** Native (Performance) oder Flutter (Cross-Platform) - **Desktop:** Electron (Web-Tech) oder Qt (Performance)
-

Vielen Dank für Ihre Aufmerksamkeit!

Fragen?