WORKSHOP: UI-KIT FÜR STUD.IP

Ron Lucke – elan e.V.

Warum brauchen wir ein UI-Kit?

- Aktuelle UI-Landschaft in Stud.IP ist uneinheitlich
- Unterschiedliche Komponenten und Architektur ohne klare Struktur
- Herausforderungen bei Wartung und Weiterentwicklung
- Anhäufung technologischer Schulden

Anforderungen an das UI-Kit

- Konsistenz in der Benutzeroberfläche
- Wiederverwendbare Komponenten für verschiedene Anwendungen
- Barrierefreiheit als fester Bestandteil
- Erweiterbarkeit, offen für neue Entwicklungen

Design-Prinzipien für das UI-Kit

- Modular & flexibel → Einfach erweiterbar
- Klare Guidelines → Einheitliches Design für alle
- Dokumentation → Auflistung von Elementen und Definition von Einsatzzwecken
- Storybook → Arbeitsumfeld für Designschaffende und Entwickelnde

elan

Technische Überlegungen

- Vue Komponenten als Grundlage für Look and Feel
- HTML Struktur und CSS-Klassen der Vue Komponente als Fallback für PHP-Templates
- Design Tokens: einheitliche semantische Variablen
- Utility-Klassen: wiederverwendbare CSS-Klassen für konsistentes Styling
- Alias für das Einbinden aus der Component Library anbieten → @component-library

Was umfasst das UI-Kit?

- Atomare Elemente (z.B. Inputs, Buttons etc.)
- Kompositionen (z.B. Multipersonsearch)
- Seiten Vorlagen

Elemente werden als Vue-Komponente erstellt und bringen zusätzlich zum Design auch noch ein eindeutiges Verhalten mit. Als Fallback für alten Code dienen Vorgaben von HTML-Struktur und CSS-Klassen.

Wie kann das UI-Kit etabliert werden?

- Aufbau einer Grundstruktur → Storybook, Components, Definition eines Code-Standards, Dokumentation, Erstellung erster Elemente und Verbreitung dieser
- Erstellung und Verbreitung aller Elemente und Kompositionen
- Erstellung von Seiten-Vorlagen und Umsetzung dieser Designs

Wie verändert das UI-Kit das aktuelle System?

Für Entwickelnde:

- klare Vorgaben und kein Unklarheiten wie etwas gestaltet sein soll
- schneller Aufbau einer UI
- kein langes suchen im Code was wofür verwendet werden soll
- bessere Wartbarkeit des Codes und Abbau von technologischen Schulden

Für Nutzende:

- Besserer Gesamteindruck, System wirkt durch die Konsistenz deutlich wertiger
- Niedrigere Lernkurve, Einheitlichkeit vereinfacht die Bedienung

Was haben wir davon?

- Bessere UX
- Vereinfachte Bedien-Logik
- Vereinfachtes Onboarding von Entwickelnden
- Bessere und aktuellere Dokumentation dank Storybook

elan

Diskussion & nächste Schritte

- Was denkt ihr über diese Anforderungen?
- Welche Aspekte sind besonders wichtig?
- Wie könnte ein Fahrplan zur Umsetzung aussehen?

elan


VIELEN DANK FÜR DIE
AUFMERKSAMKEIT
elan