Richtlinie zum „Software-Sprint“

Fullscreen: Ein kollaboratives Whiteboard,
das auch offline funktioniert

Schlussbericht

Zuwendungsempfänger:
Ahrend Wagner GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS22S08 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Für die virtuelle Zusammenarbeit benötigen wir immer mehr Anwendungen, die Teams und Organisationen in der Kommunikation und Dokumentation unterstützen. Dazu zählen auch Whiteboards, die virtuelle Meetings durch Visualisierungen vereinfachen.

Die Cloud-Infrastruktur hat gängige Standards für virtuelle Zusammenarbeit neu definiert: Nahtlose Synchronisierung ermöglicht Menschen im digitalen Raum die unmittelbare Interaktion. Dieser neue Standard kommt jedoch zum Preis von Leistung und Datenschutz, da Cloud-Infrastruktur heute meist nur als temporäres, kostenpflichtiges Abonnement der AWS-Infrastruktur erhältlich ist, und durch Ladezeiten an Performance verliert. Zudem kann man die gängigen Whiteboards wie Miro oder Mural überhaupt nicht offline bearbeiten.

Fullscreen ist eine “Local First” Whiteboard-Anwendung für Teams und Organisationen, die hier eine effiziente und sichere Open-Source-Alternative zu den Datensilos von Cloud-Infrastrukturen liefert. Die Anwendung synchronisiert Visualisierungen wie z.B. Haftnotizen und Diagramme für die virtuelle Zusammenarbeit, funktioniert aber gleichzeitig auch offline.

Der Kern der technischen Umsetzung von Fullscreen ist die Local-First-Architektur, ein neues Paradigma der Anwendungsentwicklung. Hierbei werden die Daten auf dem eigenen Gerät gespeichert und bearbeitet und zugleich über das Internet synchronisiert, um Zusammenarbeit zu ermöglichen. Dieses Modell vereint die Vorteile der Beständigkeit und direkten Verfügbarkeit klassischer Desktop-Anwendungen mit den flexiblen Kollaborationsmöglichkeiten der Cloud.

Da beim gleichzeitigen Bearbeiten von Daten auf mehreren Geräten immer Konflikte auftreten können, setzt Fullscreen CRDTs (Conflict Free Replicated Datatypes) ein. CRDTs sind eine neue Art von Datenstruktur, die das dezentrale Zusammenführen von

Änderungen ermöglicht, ohne dabei einen zentralen Server zur Konfliktauflösung zu benötigen.

Durch diese Technologien wandelt sich die Bedeutung der Cloud: Sie steht nicht mehr im Mittelpunkt der Datenverarbeitung, sondern wird austauschbar und dient nur noch der Weiterleitung von Informationen.

Geplant waren folgende Meilensteine:

M1: Architektur-Update und UX-Entwicklung

Die Architektur ist auf Automerge und tldraw umgestellt und Ende-zu-Ende verschlüsselt. UX für Konfliktlösung und Synchronisierung wird neu entwickelt.

M2: Facilitation-Funktionen

Haftnotizen, Cursor, Moderation, Timer und Agenda-Funktionen sind implementiert.

M3: Teams und Zusammenarbeit

Teams, Zugang und Freigabe sind implementiert.

M4: Performance-Optimierung

Gute Testergebnisse für langsame Verbindungen und Offline-Bearbeitung.

Beitrag des Projektes zu den Zielen der Förderinitative „Software-Sprint“

Unsere Hauptzielgruppe besteht aus Teams und Organisationen, die aufgrund von Sicherheits- oder Datenrichtlinien keine Cloud-basierten Lösungen am Arbeitsplatz nutzen können oder wollen und Whiteboarding-Funktionen benötigen (Haftnotizen, Diagramme, usw.). Sie profitiert von einem Produkt, das unabhängig von der Cloud-Infrastruktur funktioniert und gleichzeitig eine intuitive Bedienung anbietet.

Einerseits ist damit eine neue und innovative Software-Architektur implementiert wurden—CRDTs gibt es zwar schon seit ein bis zwei Jahren, aber bisher wurden sie kaum in Anwendungen erprobt—und andererseits gibt es nun eine Open-Source-Lösung für kollaborative Whiteboards, was zum Themenschwerpunkt der 11. Runde, „Arbeit“, einen Beitrag leistet.

Ausführliche Darstellung der Ergebnisse

Welche konkreten Ergebnisse hast Du erzielt? Konnten alle Meilensteine erreicht werden? Welche zusätzlichen Erkenntnisse hast Du aus der Projektarbeit gewonnen, auch im Hinblick auf die Begleitung durch die Open Knowledge Foundation?

- Fullscreen gibt es sowohl als Web App (alle gängigen Browser) als auch als Native App (macOS) in der Alpha-Version.

- Die Local-First-Architektur konnte mit der CRDT *Y.js* umgesetzt werden und die wichtigsten Whiteboard-Funktionen deckte wie geplant die Open-Source-Library *tldraw* ab.
- Es wurden UX-Prozesse entwickelt und mit Probanden getestet, vor allem für Download und Synchronisation.
- Es wurde ein eigenständiges Dateiformat *.fullscreen* entwickelt.
- Whiteboard-Funktionen: Haftnotizen, Cursor, Board über URL teilen, Board als Datei speichern
- Ein eigenes Branding und eine einfache Website tragen zur Wiedererkennung des Produkts bei.

Damit haben wir M1 (bis auf E2E) vollkommen und M2 und M3 teilweise erreicht. M4 sind wir nicht angegangen, da wir mit der Weiterentwicklung warten müssen.

Eines unser ursprünglichen Ziele war, Behörden in ihrer Arbeit zu unterstützen. In unserer Nutzerbefragung haben wir erfahren, dass lokale Software-Installationen typischerweise einen langwierigen Prüfprozess durchlaufen müssen und daher dieser Einsatz sich zunächst schwierig gestalten lässt. Wir sind der Open Knowledge Foundation sehr dankbar, einen Kontakt mit dem Innovationslabor Baden-Württembergs hergestellt zu haben.

Die Vernetzung mit anderen Projekten sowie die Vorstellung beim Demo Day hat uns die akademische Zielgruppe nochmal näher gebracht. Besonders in den kreativen Disziplinen wird oft nach Whiteboard-Lösungen gesucht und es scheint eine Marktlücke für bezahlbare und offene Software zu geben.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Zunächst ist Fullscreen als Prototyp nutzbar. Wir stellen Software und Infrastruktur bereit, die für kleine Teams, die schnell eine Whiteboard aufsetzen möchten, die nötigsten Funktionen hat. Selbstverständlich können wir in der Alpha-Version keine Stabilität der Software garantieren und werden Nutzerfeedback weiter beobachten.

Als Prototyp hat Fullscreen aber vor allem eine Vorführfunktion. Während der Förderzeit haben wir Fullscreen in der Local-First-Community rund um *Ink & Switch* vorgestellt. Die Entwickler:innen von *Y.js* und *tldraw* kennen auch Fullscreen. Damit ist ein wesentlicher Beitrag für Anwendungen mit CRDTs als sog. „Proof of Concept“ für die erweiterte Community geschaffen. Die nächste Generation von Anwendungen mit CRDTs wird mit Fullscreen ein erstes, offenes Umsetzungsbeispiel für die Entwicklung und die User Experience haben. Das ist enorm.

Sofern der Prototyp weiterentwickelt wird (s. unten) wird auch ein entsprechendes Business Model entwickelt, um dieses Projekt weiter zu finanzieren. Dafür gibt es bereits erste Konzepte, die vielversprechend sind.

Für das Projektteam war die Auseinandersetzung mit CRDTs eine produktive Zeit. Die UX-Erkenntnisse werden in die offene Bibliothek *Decent Patterns* mit eingebunden, und gleichzeitig arbeitet ein Teammitglied an *p2panda*, einem neuartigen CRDT.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Ursprünglich war unsere Projektidee, *tldraw* als Ausgangspunkt zu nutzen und unsere eigene Version auszubauen (neue UI und zusätzliche Funktionen). Doch mitten in der Förderzeit hat *tldraw* VC-Funding bekommen und sich für die Entwicklung einer Core-Library entschlossen. Das ist an sich eine sehr gute Nachricht für Fullscreen—denn wir können auf die Funktionen von *tldraw* direkt aufbauen—doch (1) ist die Core-Bibliothek noch nicht fertig bzw. ausgereift, und (2) geschieht die Entwicklung hinter verschlossenen Türen. Wir haben mit der zwischenzeitlichen Umstellung auf *tldraw/core* wertvolle Zeit verloren und mussten schlussendlich doch die veraltete Version von *tldraw* nutzen. Wir würden abwarten, bis *tldraw/core* in einer stabilen Version veröffentlicht wird. Entsprechend würden wir Fullscreen weiterentwickeln.

Des Weiteren haben wir unterschiedliche UX getestet. Es wurde schnell klar, dass Dateien im Browser-Kontext nicht geeignet sind, obwohl Browser wie Chrome diese UX unterstützen würde. Das haben die ersten User Tests ergeben. Wir haben uns also bewusst für die Native App entschieden, damit das Nutzungsverhalten klar ist.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Website: https://fullscreen.space/
Github: https://github.com/interalia-studio/fullscreen/
UX-Bibliothek: https://decentpatterns.com/
p2panda: https://p2panda.org/
Ink & Switch: https://www.inkandswitch.com/

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Wie bereits erwähnt hat uns die *tldraw/core* Umstellung wertvolle Zeit gekostet. Wir mussten wichtige Funktionen aus M2 und M3 streichen und haben uns stattdessen auf das Backend fokussiert.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Das US-amerikanische Industrial Research Lab *Ink & Switch* hat uns strategisch und technisch inspiriert. Wir sind im regen Austausch mit der Local-First-Community und wollen auch zukünftig mitwirken und Local-First-Anwendungen—aufbauend auf CRDTs—entwickeln.