



netidee

PROJEKTE

own.tiles

Endbericht | Call 18 | Projekt ID 6928

Lizenz CC BY-SA

Inhalt

1	Einleitung	3
2	Projektbeschreibung	3
3	Verlauf der Arbeitspakete	4
3.1	Arbeitspaket 1 – Detailplanung und Formales am Projektstart	4
3.2	Arbeitspaket 2 – Modellieren der Softwarearchitektur	4
3.3	Arbeitspaket 3 - Framework	5
3.4	Arbeitspaket 4 - Open Widget Library.....	6
3.5	Arbeitspaket 5 - Figma-to-Widget Guide	8
3.6	Arbeitspaket 6 - Dokumentation und Formales am Projektende	8
4	Umsetzung Förderauflagen	9
5	Liste Projektendergebnisse	9
6	Verwertung der Projektergebnisse in der Praxis	10
7	Öffentlichkeitsarbeit/ Vernetzung.....	10
8	Eigene Projektwebsite	11
9	Geplante Aktivitäten nach netidee-Projektende.....	11
10	Anregungen für Weiterentwicklungen durch Dritte	12

1 Einleitung

own.page hat sich auf die Entwicklung einer mobilen App spezialisiert, mit der Nutzer:innen ihre eigene Mini-Website direkt vom Smartphone aus schnell erstellen können. Ähnlich wie auf einem Homescreen können sie verschiedene Elemente wie Text, Bilder oder funktionale Widgets in einem Raster platzieren. Im Laufe der Entwicklung haben wir jedoch festgestellt, dass die Implementierung solcher Widgets für viele Entwickler:innen immer noch zu viel Zeit in Anspruch nimmt und es auch keine eigenständige, Open-Source-Lösung gibt, die es Endnutzer:innen ermöglicht, Widgets einfach zu modifizieren und in ihre Websites zu integrieren. Deshalb haben wir das Projekt own.tiles bei Netidee eingereicht, um dieses Problem zu lösen.

2 Projektbeschreibung

Own.tiles ist ein Framework, das die Entwicklung und Integration modularer Website-Komponenten (Widgets) effizient und zugänglich macht. Entwickler:innen können damit Widgets erstellen, die eigenständige Funktionen bieten oder Inhalte externer Dienste wie Calendly oder Shopify anzeigen. Nutzer:innen können diese Widgets ohne Programmierkenntnisse in ihre Websites leicht einbinden, unabhängig von der unterliegenden Plattform.

Projektziele und Zielgruppe

Das Hauptziel des Projekts war die Entwicklung eines Frameworks, das die Erstellung von Widgets für Entwickler:innen erheblich vereinfacht, dabei aber ausreichend Flexibilität bietet. Besonderen Wert haben wir auf Barrierefreiheit, Responsiveness und Konfigurierbarkeit gelegt – Aspekte, die essenziell für die breite Nutzbarkeit und Anpassbarkeit der Widgets sind.

Neben der technischen Grundlage war es unser Ziel, eine anwenderfreundliche Widget-Bibliothek zu schaffen, die insbesondere für Freelancer, Content Creator oder kleine Unternehmen nützlich ist. Diese Zielgruppe kann durch own.tiles schnell und einfach funktionale Widgets in bestehende Webseiten integrieren, ohne auf teure individuelle Entwicklungen angewiesen zu sein.

Projektergebnis

Im Rahmen des Projekts haben wir folgende Kernbestandteile erfolgreich umgesetzt:

- **Das own.tiles-Framework**, das eine einheitliche Schnittstelle für die Widget-Entwicklung bietet und Standardfunktionen wie Konfigurierbarkeit, Responsiveness und Barrierefreiheit unterstützt.

- **Die Open-Widget-Bibliothek**, in der Nutzer:innen eine Auswahl an sofort einsetzbaren, anpassbaren Widgets finden. Aktuell stehen zehn eigene Widgets zur Verfügung, mit weiteren in Planung.
- **Den Figma-to-Widget-Guide**, der Entwickler:innen hilft, Widgets aus Design-Dateien effizient in funktionale Module umzusetzen.
- **Umfassende Dokumentation** für Entwickler:innen und Endnutzer:innen, die den gesamten Prozess – von der Widget-Erstellung bis zur Einbindung auf Webseiten – verständlich beschreibt.

Mit diesen Ergebnissen haben wir alle gesetzten Projektziele erreicht und eine solide Grundlage für die Weiterentwicklung und Verbreitung von own.tiles geschaffen.

3 Verlauf der Arbeitspakete

3.1 Arbeitspaket 1 – Detailplanung und Formales am Projektstart

Dieses Arbeitspaket umfasste die Vertragsprüfung, das Unterzeichnen sowie das Versenden der notwendigen Dokumente an Netidee. Außerdem wurde die Projektplanung auf Basis der Excel-Vorlage erstellt. Der Inhalt der Projektwebsite wurde überprüft und ein erster Blogbeitrag verfasst. Des Weiteren wurde das Förderabrufformular für die erste Förderrate an Netidee übermittelt.

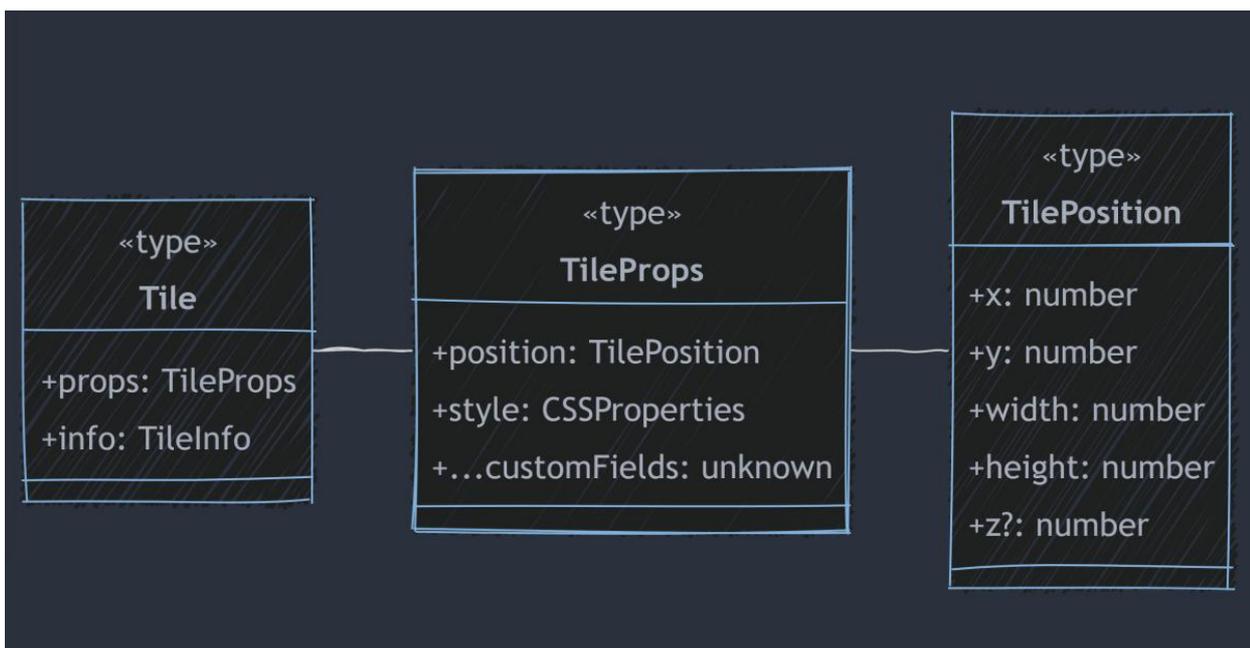
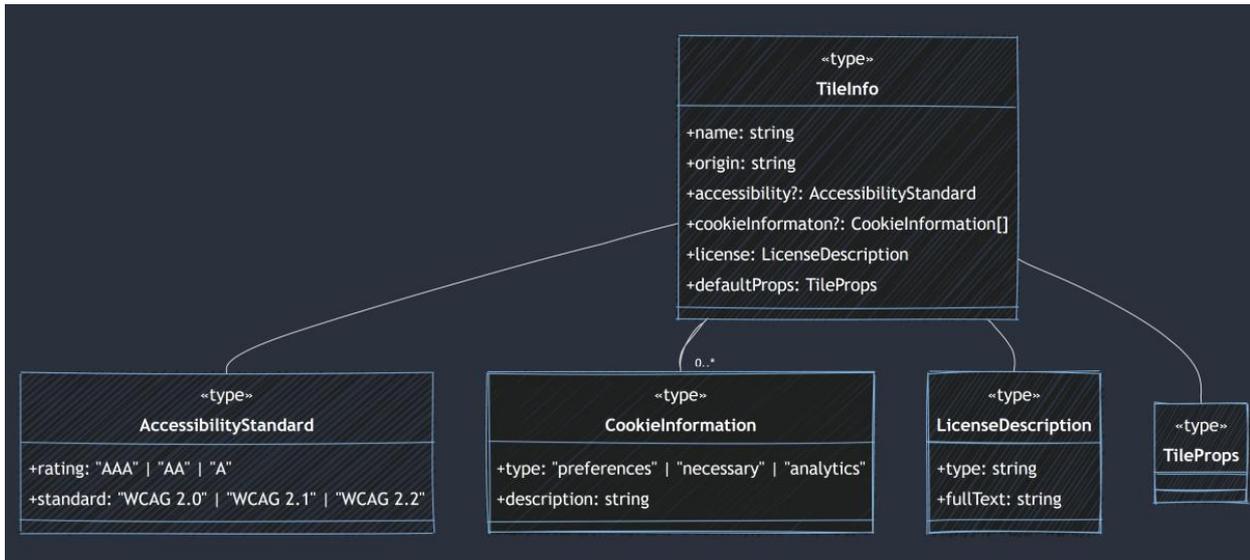
Innerhalb des Teams erfolgte eine detaillierte Planung, die die Roadmap sowie konkrete Arbeitspakete und deren Aufgabenstellungen umfasste. Es gab keine nennenswerten Probleme oder Schwierigkeiten bei diesem Arbeitspaket. Lediglich der Start verzögerte sich, da zeitgleich die Gründung unserer Firma stattfand. Dies wurde jedoch mit Netidee abgesprochen, und der neue Projektbeginn wurde genehmigt.

3.2 Arbeitspaket 2 – Modellieren der Softwarearchitektur

Dieses Arbeitspaket war von entscheidender Bedeutung für die Grundlegung unserer Softwarearchitektur. Einer der ersten Schritte, in den wir viel Zeit investiert haben, war die Durchführung einer umfassenden Recherche sowie die Erstellung einer Anforderungsanalyse. Im Anschluss daran planten wir die Softwarearchitektur des Projekts und testeten eine erste Prototypversion des Frameworks gemeinsam mit Endnutzer:innen über unsere own.page App.

Dieses Paket bildet die Basis für die Arbeitspakete 3 und 4, die sich mit dem Framework und der Open Widget Library befassen. Das angestrebte Ergebnis, ein Komponentendiagramm zur Modellierung der Daten und Beziehungen der Widgets sowie der dazugehörige Code für die Typendefinition der Domäne, wurde erreicht und es gab dabei keine großen Abweichungen oder

Überraschungen. Hier ist ein Teil des erstellten Diagramms, welches unsere Frameworkarchitektur darstellt:



Weitere Details zur Softwarearchitektur sind in diesem Blogpost zu finden: [own.tiles Softwarearchitektur](#).

3.3 Arbeitspaket 3 - Framework

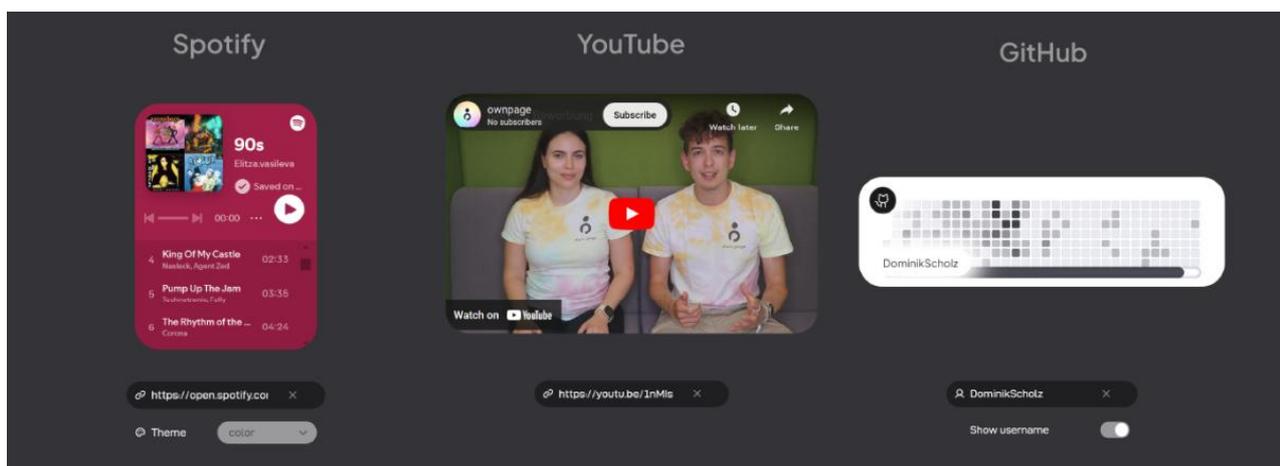
Nachdem die Anforderungsanalyse und die Modellierung der Softwarearchitektur abgeschlossen waren, begann die eigentliche Implementierung des Frameworks. Dieses Arbeitspaket war das umfangreichste, da unser Ziel darin bestand, das System für Entwickler:innen so einfach wie möglich zu gestalten und gleichzeitig die von Widget-Entwickler:innen bereitgestellten

Informationen zu automatisieren. Dabei wurden zentrale Aspekte wie Barrierefreiheit, Cookies und Lizenzen für jede Webseite berücksichtigt.

Die Haupttätigkeiten dieses Arbeitspakets umfassten die Erstellung eines GitHub-Repositories für das Framework, die Implementierung des Widget-Baukastensystems, die Dokumentation des Codes sowie die Veröffentlichung des Frameworks. Besonders zeitintensiv war die Automatisierung der Typendefinitionen, wodurch dieses Arbeitspaket etwas länger dauerte als ursprünglich geplant. Dennoch blieb der zeitliche Aufwand im erwartbaren Rahmen, und es gab keine größeren Abweichungen oder Probleme.

Seit dem Zwischenbericht wurde dieses Arbeitspaket vollständig abgeschlossen. In den letzten Monaten haben wir alle fehlenden Funktionalitäten final implementiert. Dazu gehörten insbesondere zusätzliche Typendefinitionen und erweiterte Properties für jedes Widget. Neben den bereits berücksichtigten Aspekten wie Barrierefreiheit, Accessibility und Cookies haben wir weitere Attribute wie Author und Quelle ergänzt. Zudem wurde das System so erweitert, dass Entwickler:innen und Endnutzer:innen die Größe und Dimensionen von Widgets individuell anpassen können. Diese Informationen werden nun direkt im iFrame oder Link gespeichert, um die Integration der Widgets zu erleichtern.

Ein bedeutender Meilenstein war zudem die Implementierung von sechs weiteren Widgets, sodass wir unser Ziel von mindestens zehn fertigen Widgets erreicht haben. Zwei dieser neuen Widgets wurden vollständig frei programmiert, um die Flexibilität und Leistungsfähigkeit des Frameworks zu testen. Die Ergebnisse haben unsere Erwartungen übertroffen und bestätigen die Benutzerfreundlichkeit und Effizienz unseres Systems.



Mehr über das own.tiles-Framework kann man in diesem Blogpost finden: [Link](#).

3.4 Arbeitspaket 4 - Open Widget Library

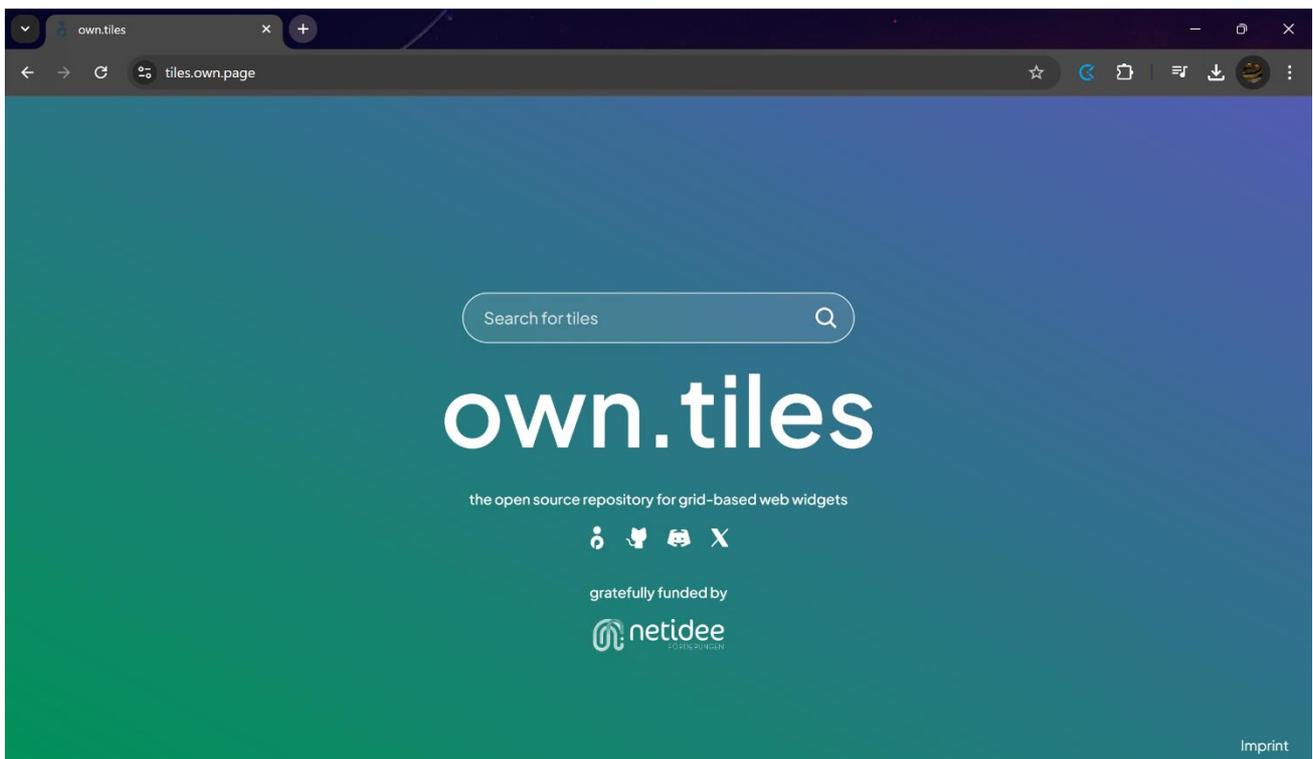
Die Entwicklung der Open Widget Library wurde auch erfolgreich abgeschlossen. Ziel dieses Arbeitspakets war es, eine zentrale Plattform bereitzustellen, auf der Nutzer:innen Widgets

durchsuchen, individuell anpassen und einfach in ihre Webseiten integrieren können. Dazu gehörten die Erstellung eines GitHub-Repositories für die Library-Website, die Implementierung eines dynamischen Widget-Konfigurators, der Aufbau einer Widget-Datenbank sowie die Erstellung einer verständlichen Dokumentation für Endnutzer:innen.

Seit dem letzten Bericht haben wir zahlreiche Verbesserungen und Erweiterungen umgesetzt. Ein besonderer Fokus lag auf der Optimierung der Barrierefreiheit gemäß den WCAG-Richtlinien. Die Landing Page der Open Widget Library wurde angepasst, indem Kontraste verbessert und essenzielle Elemente mit passenden Labels versehen wurden. Zusätzlich haben wir die Widgets selbst transparenter gestaltet: Jedes Widget enthält nun detaillierte Informationen zu Autor, Quelle, Lizenz, Cookies und Barrierefreiheitsstatus, sodass Nutzer:innen direkt nachvollziehen können, welche Anforderungen es erfüllt.

Ein weiterer wesentlicher Meilenstein war die Erweiterung der Konfigurationsmöglichkeiten für Widgets. Nutzer:innen können nun Höhe und Breite in verschiedenen Einheiten definieren, wobei diese Einstellungen direkt im iFrame gespeichert werden. Dadurch wird sichergestellt, dass die gewählte Größe exakt eingebettet wird. Zudem wurde eine Suchfunktion implementiert, mit der sich alle verfügbaren Widgets schnell finden lassen. Neben diesen strukturellen Verbesserungen haben wir die Widget-Bibliothek weiter ausgebaut und sechs neue Widgets integriert.

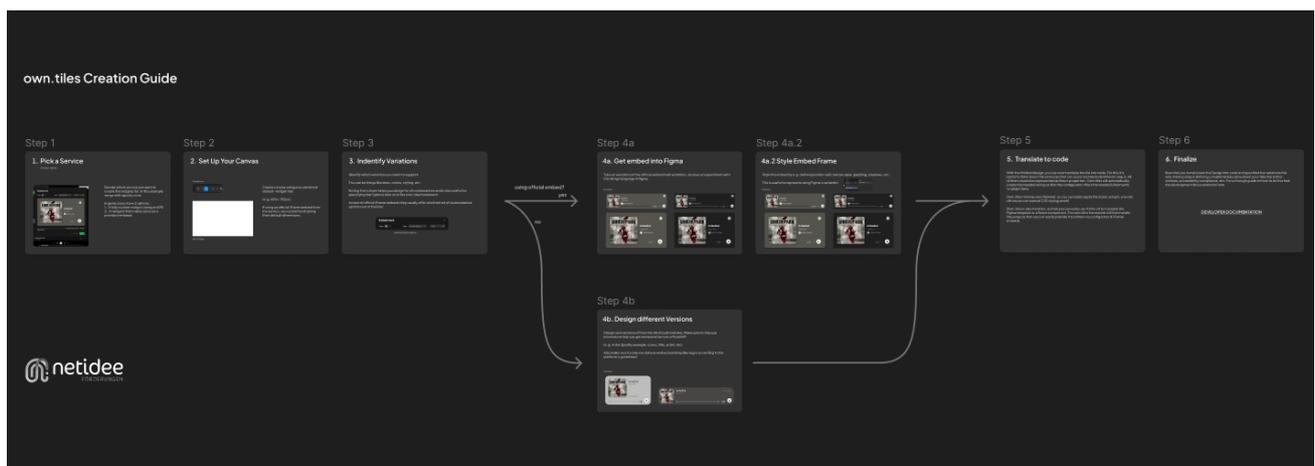
Obwohl dieses Arbeitspaket etwas mehr Zeit in Anspruch genommen hat als ursprünglich geplant, konnten wir die Verzögerung im nächsten Arbeitspaket ausgleichen. Es gab keine größeren Hürden oder Probleme, und insgesamt sind wir mit dem Ergebnis sehr zufrieden. Tatsächlich haben wir mehr umgesetzt als erwartet, wodurch die Open Widget Library nun noch leistungsfähiger und benutzerfreundlicher ist.



3.5 Arbeitspaket 5 - Figma-to-Widget Guide

Arbeitspaket 5 hatte das Ziel, Entwickler:innen einen klaren, visuellen Leitfaden zur Erstellung von own.tiles zu bieten. Dies beinhaltete das Design von Figma-Templates, die dynamisches Skalieren in unterschiedlichen Größen ermöglichen, um sicherzustellen, dass die Widgets auf verschiedenen Geräten und Bildschirmgrößen gut aussehen und funktional bleiben. Darüber hinaus wurde ein detaillierter grafischer Guide erstellt, der Schritt für Schritt erklärt, wie Figma-Designs effektiv in own.tiles umgesetzt werden können, sodass Entwickler:innen nicht nur den technischen, sondern auch den designtechnischen Anforderungen gerecht werden. Dieser Leitfaden soll Entwickler:innen helfen, ansprechende und funktionale Widgets zu erstellen, die das Nutzererlebnis verbessern.

Das Arbeitspaket wurde schneller als ursprünglich erwartet abgeschlossen, jedoch nahmen die vorherigen Arbeitspakete mehr Zeit in Anspruch. Es gab keine größeren Hürden oder unerwarteten Herausforderungen bei der Umsetzung. Die Figma-Templates und der Guide sind nun öffentlich zugänglich und können von der Entwickler:innen-Community genutzt werden, um eigene Widgets zu erstellen und zu integrieren. Der Guide ist unter folgendem Link verfügbar: [Figma-Design-Guide](#).



3.6 Arbeitspaket 6 - Dokumentation und Formales am Projektende

Dieses Arbeitspaket umfasst die abschließende Dokumentation und formale Abwicklung des Projekts. Dazu gehört die Erstellung des Projektendberichts mit einer detaillierten Zusammenfassung und Endabrechnung, welche zusammen mit den gescannten Originalbelegen an Netidee übermittelt werden. Außerdem wird das Förderratenabrufformular für die dritte Förderrate ausgefüllt und eingereicht. Die Enddokumentation, einschließlich der Anwender:innen- und Entwickler:innen-Dokumentation, wird auf der Projektwebsite veröffentlicht. Abschließend wird ein Blogbeitrag zum Projektabschluss verfasst, der auf der Homepage von Netidee veröffentlicht wird.

Es gab keine größeren Hürden oder Probleme mit diesem Arbeitspaket und alles wurde in dem eingeplanten Zeitfrist erledigt.

4 Umsetzung Förderauflagen

Es sind keine speziellen Förderauflagen vereinbart.

5 Liste Projektergebnisse

Kurzbeschreibung der erreichten Projektergebnisse jeweils mit Open Source Lizenz und Webadresse (netidee Vorgaben beachten!)

1	Projektzwischenbericht	CC BY-SA 4.0	https://www.netidee.at/owntiles
2	Projektendbericht	CC BY-SA 4.0	https://www.netidee.at/owntiles
3	Entwickler_innen-DOKUMENTATION	CC BY-SA 4.0	https://www.netidee.at/owntiles
4	Anwender_innen-DOKUMENTATION	CC BY-SA 4.0	https://www.netidee.at/owntiles
5	Veröffentlichungsfähiger Einseiter / Zusammenfassung	CC BY-SA 4.0	https://www.netidee.at/owntiles
6	Dokumentation Externkommunikation zur Erreichung Sichtbarkeit /Nachhaltigkeit (als Teil des Endberichtes)	CC BY-SA 4.0	https://www.netidee.at/owntiles
7	Software Modul für andere Entwickler_innen	MIT	https://github.com/own-page/own.tiles https://www.netidee.at/owntiles
8	Software Server für Anwender_innen	MIT	https://tiles.own.page/ https://github.com/own-page/own.tiles https://www.netidee.at/owntiles

9	<i>Daten-Projektergebnisse als Open Data</i>	CC BY-SA 4.0	https://www.figma.com/design/9LznTVJuZZOIEInqmbMihSj/own.tiles-Creation-Guide https://www.netidee.at/owntiles
10	<i>Dokumentation und Developer Guide</i>	CC BY-SA 4.0	https://github.com/own-page/own.tiles https://www.figma.com/design/9LznTVJuZZOIEInqmbMihSj/own.tiles-Creation-Guide https://www.netidee.at/owntiles

6 Verwertung der Projektergebnisse in der Praxis

Die Projektergebnisse von own.tiles werden bereits aktiv in der Praxis genutzt, insbesondere durch die Integration in own.page. Jede mit own.page erstellte Webseite erhält automatisch Zugriff auf own.tiles, sodass Nutzer:innen interaktive Widgets direkt und ohne zusätzlichen Einbettungsaufwand verwenden können. Die Nutzung in own.page hat bereits bewiesen, wie ausgereift und stabil own.tiles ist und welchen hohen Nutzen es sowohl für uns als Entwickler:innen als auch für unsere Nutzer:innen bietet.

Darüber hinaus wurde own.tiles bereits in Plattformen wie Notion getestet und eingesetzt, wo sich Widgets nahtlos als eingebettete Inhalte integrieren lassen. Dies ermöglicht eine einfache und schnelle Nutzung interaktiver Elemente in einem weit verbreiteten Tool. Durch diese praktischen Anwendungen trägt own.tiles aktiv zur Erweiterung bestehender digitaler Ökosysteme bei.

Zudem stehen wir im Austausch mit Entwickler:innen aus der Community und arbeiten daran, own.tiles in Zukunft auf weiteren Webseiten zu integrieren. Unser Ziel ist es, die Vielseitigkeit und den Einsatzbereich unserer Widgets kontinuierlich zu erweitern.

7 Öffentlichkeitsarbeit/ Vernetzung

Die Sichtbarkeit und Vernetzung von own.tiles wurden bereits durch verschiedene Maßnahmen vorangetrieben. Wir haben own.tiles auf mehreren Veranstaltungen wie dem, Netidee Spring Talk Call 18, Connect Day und Vienna Up vorgestellt, um es einer breiteren Öffentlichkeit näherzubringen und wertvolles Feedback zu sammeln. Nutzer:innen-Interviews halfen uns, die Bedürfnisse unserer Zielgruppe besser zu verstehen und die Entwicklung gezielt darauf auszurichten. Die eigene Website von own.tiles wurde veröffentlicht und enthält die Open Widget

Library, wodurch Entwickler:innen und Interessierte direkten Zugang zu den verfügbaren Widgets erhalten.

Zudem ist own.tiles bereits in unsere Plattform own.page integriert, wodurch alle own.page-Nutzer:innen die Möglichkeit haben, Widgets nahtlos zu nutzen und so zur Verbreitung der Plattform beizutragen. Die Beta-Phase von own.page hat uns weiteres wertvolles Feedback geliefert, insbesondere hinsichtlich gewünschter Widget-Funktionalitäten. Darüber hinaus haben wir in den letzten Monaten begonnen, own.tiles aktiv auf X (ehemals Twitter) zu präsentieren, uns dort mit der Build-in-Public- und Web-Dev-Community zu vernetzen und unsere Reichweite organisch zu vergrößern. Erste Entwickler:innen haben sich bereits mit dem Projekt beschäftigt, es auf GitHub mit Sternen versehen und sogar eine erste Abspaltung (Fork) erstellt.

In Zukunft planen wir, own.tiles gezielt auf relevanten Entwickler:innen-plattformen vorzustellen, sobald das Projekt vollständig abgeschlossen ist und aktiv von der Community genutzt wird. Ein möglicher Product Hunt Launch könnte ebenfalls dabei helfen, eine größere Zielgruppe zu erreichen und das Projekt innerhalb der Entwickler:innen-Community bekannter zu machen. Jetzt, da die Grundlagen des Frameworks und der Open Widget Library abgeschlossen sind, werden wir auch verstärkt auf X über die Entwicklung von own.tiles posten, aktiv mit der Community interagieren und erste Entwickler:innen dazu einladen, Widgets für die Plattform zu testen und beizutragen.

8 Eigene Projektwebsite

- Own.tiles GitHub Repository (Framework und Widget Bibliothek):
<https://github.com/own-page/own.tiles>
- Own.tiles Webseite (Open Widget Library für Anwender:innen):
<https://tiles.own.page/>
- Own.tiles Figma-to-Widget Guide:
<https://www.figma.com/design/9LznTVJuZZOIEInqmbMihSj/own.tiles-Creation-Guide>

9 Geplante Aktivitäten nach netidee-Projektende

Ja, wir planen, own.tiles kontinuierlich weiterzuentwickeln und auszubauen. Neben der aktiven Nutzung in unserer own.page-App wollen wir die Verbreitung der Bibliothek fördern und die Entwickler:innen-Community stärken. Dafür setzen wir auf verschiedene Maßnahmen:

- **Community-Aufbau:** Ausbau unserer Discord-Community und X-Präsenz, um Entwickler:innen zur Mitgestaltung neuer Widgets zu motivieren.

- **Erweiterung der Widget-Bibliothek:** Förderung neuer Open-Source-Widgets für verschiedene Anwendungsfälle, von Business-Tools bis hin zu kreativen Elementen.
- **Integration in weitere Plattformen:** Neben Websites auch gezielte Unterstützung für Plattformen wie Notion, Medium oder CMS-Systeme.
- **Optimierung der Usability:** Vereinfachung des Konfigurationsprozesses, um die Anpassung von Widgets noch intuitiver zu gestalten.

Durch diese Maßnahmen wollen wir die Nutzung und Weiterentwicklung von own.tiles langfristig fördern.

10 Anregungen für Weiterentwicklungen durch Dritte

Own.tiles bietet eine Open-Source-Plattform, die es ermöglicht, interaktive Widgets ohne tiefgehende Programmierkenntnisse in bestehende Webseiten oder Plattformen zu integrieren. Neben der Nutzung durch Endanwender:innen eröffnet dies auch zahlreiche Weiterentwicklungsmöglichkeiten für Entwickler:innen und Dritte.

Mögliche Weiterentwicklungen:

- **Spezialisierte Widgets für neue Anwendungsfälle:** Entwicklung neuer Widgets für spezifische Bedürfnisse, etwa für Barrierefreiheit, E-Commerce, Automatisierung oder interaktive Medien.
- **Erweiterung der Kompatibilität:** Anpassung von own.tiles für weitere Plattformen wie E-Learning-Systeme, Portfolio-Builder oder Community-Foren, um neue Nutzergruppen zu erreichen.
- **Optimierung für mobile Nutzung:** Verbesserung der Widgets für eine nahtlose mobile Erfahrung, um den steigenden Anteil mobiler Nutzer:innen besser zu bedienen.
- **Erweiterte Sicherheits- und Datenschutzfunktionen:** Einführung von Widgets mit strikteren Datenschutzoptionen, z. B. datenschutzfreundliche Analysetools oder Cookie-freie Alternativen.
- **Mehrsprachigkeit und Internationalisierung:** Bereitstellung von Widgets mit Unterstützung für mehrere Sprachen, um die Nutzung in internationalen Märkten zu erleichtern.
- **Verknüpfung mit KI-gestützten Diensten:** Integration von KI-gestützten Funktionen, etwa für automatisierte Textgenerierung, Chatbots oder intelligente Suchfunktionen.