Richtlinie zum „Software-Sprint"

Tuttle – schmerzlose Finanzplanung für Solo-Selbständige

Schlussbericht

Zuwendungsempfänger:
Dr. Christian Staudt

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS21S45 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Was war Deine Motivation? Welches Problem wolltest Du mit Deinem Projekt lösen? Wie war die geplante Vorgehensweise zur Problemlösung (auch Angabe der wichtigsten Meilensteine)?

Mit Tuttle entwickeln wir ein Businessplanungstool, das genau auf die Anforderungen von Solo-Selbständigen zugeschnitten ist. Wir automatisieren viele der üblichen Berechnungen und Arbeitsschritte - von der Zeitbudgetplanung bis zur Rechnungsstellung - und verschaffen Freelancer*innen mehr Zeit für die Arbeit, die sie lieben. Die Idee zu diesem Projekt entstand aus dem Erfahrungsaustausch unter Freelancer*innen.

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

Wer ist die Zielgruppe für Deine Lösung? Wie profitiert sie von den Ergebnissen Deines Projekts? Welche Bezüge gibt es zu den Themenfeldern und Zielen des Software Sprints?

Die Arbeitswelt ist im Wandel, der Trend geht zum Freelancing: Softwareentwickler*innen, Designer*innen oder Journalist*innen schätzen die Freiheiten und Gestaltungsmöglichkeiten der Solo-Selbständigkeit. Immer mehr Profis wählen sie für sich als die "artgerechte" Form des Arbeitens. Sie erlaubt es ihnen, sich zu spezialisieren und mit vielen Projekten und Kund*innen Erfahrung zu sammeln. Freelancer*innen spielen dadurch eine besondere Rolle - für kreative Ideen, unabhängige Beratung, innovativen Wissenstransfer und eine vielfältige Medienlandschaft.

Unter den 32 Millionen Selbstständigen (14 % aller Beschäftigten) in der Europäischen Union (EU) zielen wir auf das große Segment der Freelancer. Die Tendenz steigend, z.B. in den Bereichen Beratung & Coaching, Softwareentwicklung,Medien, Journalismus, Kunst & Design. In dieser Gruppe befinden sich auch ca. 100.000 freiberufliche Journalistinnen, die mehr als ein Viertel aller Journalistinnen in der Region ausmachen.

Beim Freelancing gibt es viele Nebentätigkeiten: Vermarktung, Kundenkommunikation, Rechtliches, und vor allem Zeit- und Finanzplanung - wohl nicht das, was am Freelancen am meisten reizt. Vernachlässigte Planung birgt aber das Risiko von Insolvenz, Verschuldung, prekärer Selbständigkeit, oder Altersarmut. Was aber, wenn Software die Finanzplanung beim Freelancing fast so einfach wie ein Angestelltenverhältnis machen würde? Was, wenn intelligente Datenauswertung dabei hilft, gute Entscheidungen als "Arbeitszeitunternehmer*in" zu treffen? Unser Tool minimiert Risiken und macht den finanziellen "Papierkram" einfach. Freelancing wird so effizienter, weniger riskant, und damit auch einsteigerfreundlicher.

Unser Ziel ist es Freelancern eine kostenfreie Open-Source-Lösung zur Verfügung zu stellen, die auch ohne ein Geschäftsmodell auskommt - insbesondere ohne ein Geschäftsmodell, das auf der Verwertung von personenbezogenen Daten beruht.

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?

Produkt des Projektes ist eine GUI-Applikation für die gängigen Desktop-Betriebssysteme. Die Verarbeitung von Geschäftsdaten erfolgt dabei bewusst lokal auf dem Endgerät - ohne zentrale Datenerfassung.

Für die Kernfunktionalität setzen wir auf Open-Source-Tools aus dem Python-Ökosystem. Die Kernfunktionen sind in Kürze:

1. **Kunden- und Vertragsmanagement:** Unsere Applikation wird zum zentralen Ort zur Ablage und Organisation von Kundendaten und vertraglichen Vereinbarungen (Laufzeiten, Stundensätze...).

2. **Zeiterfassung**: Wir ermöglichen den Import von Zeiterfassungsdaten über verschiedene Schnittstellen, wie beispielsweise den Kalender in der Cloud.

3. **Rechnungswesen**: Basierend auf Zeiterfassung und Vertragsdaten generiert die Applikation auf Knopfdruck Rechnungen.

4. **Zeitmanagement**: Als Freelancer*in möchte ich vorausschauend meine Kapazitäten für verschiedene Projekte und Kund*innen planen, beispielsweise über meinen Kalender. Auf Basis dieser Planung kann ich zukünftige Umsätze schätzen und die Einhaltung vertraglich vereinbarter Kontingente überwachen.

5. **Umsatzvorschau**: Basierend auf Zeiterfassung und Zeitplanung können wir den voraussichtlichen Umsatz berechnen und live anpassen sobald sich die Planung ändert. Damit hilft die Applikation, Entscheidungen bei der Kapazitätsplanung zu treffen.

6. **Vorschau auf Steuern und Sozialversicherungsbeiträge**: Diese umsatzabhängigen Zahlungsverpflichtungen schätzt das Tool basierend auf Zeiterfassung und vorausschauender Zeit- und Projektplanung.

7. **Effektives Einkommen**: Was ist mein effektives "Gehalt" aus meiner selbständigen Tätigkeit? Unsere App beantwortet nach guter Pflege der Daten auch diese Frage.

In Interviews mit Solo-Selbständigen aus verschiedensten Branchen haben wir dazu Anforderungen gesammelt und analysiert. Dies half uns dabei zu konsolidieren, was für unser Projekt als "in scope" gilt und was für ein Minimal Viable Product (MVP) priorisiert werden sollte. Die Interviews lieferten auch die Erkenntnis, dass schon relativ einfache Automatisierungen vielen Freelancer*innen relevant Zeit sparen können.

In der Förderphase haben wir uns auf das Prototyping fokussiert, d. h. Entwurf und Implementierung der Kernfunktionalität sowie des benötigten Technolgiestacks. Der Technologiestack besteht aus folgenden Open-Source-Komponenten:

- pandas (https://pandas.pydata.org): Auswertung tabellarischer Daten
- altair (https://altair-viz.github.io): Datenvisualisierung
- sqlmodel (https://sqlmodel.tiangolo.com) und sqlalchemy (https://www.sqlalchemy.org): Datenbankmodell
- .... sowie eine Vielzahl an weiteren Open-Source-Bibliotheken aus dem Python-Ökosystem

Es folgte die Evaluation eines geeigneten Technologiestacks, die Implementierung eines Datenmodells, sowie die Implementierung der für ein MVP benötigten Workflows und Automatisierungen. Diese Grundfunktionen stellen wir in Form einer Python-Library zusammen. Diesen Prototyp in Form von Workflows validieren wir im Test mit diversen Freelancer*innen, sowohl während der Förderphase also auch im Nachgang.

Während der Förderphase abgeschlossen wurde die Entwicklung der Kernfunktionalitäten 1-5. Die Funktionsgruppen 6 & 7, die auf den Funktionsgruppen 1-5 aufbauen, wurden begonnen und folgen in Kürze.

Während der Förderphase nicht fertiggestellt wurde die Integration der Funktionen in eine plattformübergreifende GUI-Applikation. Es wurden zunächst GUI-Plattformen evaluiert. Die Entscheidung fiel auf Electron (https://www.electronjs.org) in Kombination mit pyodide (https://pyodide.org). Wir werden das Tool im Anschluss an die Förderphase auf dieser Basis zu einer anwenderfreundlichen Desktop-Applikation weiterentwickeln. Nächste Schritte beinhalten also das Design und Integration der Funktionen in einer Cross-Platform GUI-Applikation.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Welcher Nutzen ergibt sich für die Zielgruppe aus den Ergebnissen Deines Projekts? Welche weitergehenden Effekte ergeben sich aus der Open-Source-Stellung der Ergebnisse? Gibt es Ideen für die Weiterentwicklung Deiner Lösung und Pläne zu deren Umsetzung?

Nutzerinterviews haben den Bedarf nach einem Open-Source Businessplanungslösung für Solo-Selbständige bestätigt. Wir werden das Tool im Anschluss an die Förderphase zu einer anwenderfreundlichen Desktop-Applikation für Nutzer*innen ohne Programmierkenntnisse weiterentwickeln. Nächste Schritte beinhalten also das Design und Integration der Funktionen in

einer Cross-Platform GUI-Applikation.

Ein weiteres Ziel für die Phase dem Release eines MVP ist, das Projekt für die kommenden Jahre finanziell nachhaltig aufzustellen, beispielsweise mit Hilfe von GitHub Sponsoring.

Hat die Arbeit in dem Projekt Dich in Deiner persönlichen, fachlichen Weiterentwicklung unterstützt?

Als Freelancer setze ich mittlerweile Tuttle erfolgreich für mein eigenes Zeit- und Rechungsmanagement und meine Auftragsplanung ein.

Als Softwareentwickler mit Fokus Data Science konnte ich sowohl auf meinen Kenntnissen aufbauen als auch mein Skillset deutlich erweitern, da die Entwicklung von Anwendersoftware für den Desktop eine neue Herausforderung darstellte. Darüber hinaus ergaben sich neue Perspektiven auf Open Source-Entwicklung als Geschäftszweig und Karrierezweig.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Gab es Arbeiten bzw. Lösungsansätze, die nicht weiter verfolgt wurden? Was waren die Hintergründe, und wie bist Du alternativ vorgegangen?

Eine Python-Anwendung als Desktop-Applikation umzusetzen bleibt weiterhin eine technische Herausforderung. Bei der Auswahl geeigneter GUI-Frameworks kamen verschiedene Lösungen in Frage, die Wahl war nicht offensichtlich und konnte erst nach experimentellem Programmieren getroffen werden.

Ein Kandidat für die Lösung war beispielsweise die BeeWare Plattform (https://beeware.org), die zwar die Erstellung von cross-platform-Applikationen vereinfachte, allerdings nicht die die Gestaltungsmöglichkeiten bei der Benutzeroberfläche bot, die unsere Zielgruppe gewohnt ist.

Abgesehen von üblichen Entscheidungen gegen verschiedene Alternativen bei den Komponenten erwies sich der angepeilte Technologiestack als geeignet und der geplante Lösungsweg als umsetzbar.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Wo können sich Interessenten detailliert über Deine Projektergebnisse informieren (z.B. Webseite, GHitHub, Veröffentlichungen)?

Wir hosten das Projekt auf GitHub unter https://github.com/tuttle-dev. Entwicklung und Projektmanagement finden hier vollständig transparent statt. Aus dem GitHub Repository generieren wir auch die die Projektwebseite, die unter https://tuttle-dev.github.io/tuttle/ abzurufen ist.

Die Kernfunktionen haben wir in einer Python-Library gebündelt (https://github.com/tuttle-dev/tuttle), die in Kürze im Python Package Index (https://pypi.org) veröffentlicht wird.

Eine Demo in Form von Jupyter Notebooks wurde bereits mit einigen ausgewählten Tester*innen validiert und ist nun ebenfalls auf GitHub zugänglich.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Gab es im Projektverlauf Ereignisse, die eine Anpassung der Planung erforderlich machten – z.B. Mehr- oder Minderaufwand bei der Bearbeitung von Teilaufgaben?

In der Anfangsphase des Projektes, in der es um das Sammeln und Strukturieren von Anforderungen ging, fielen zunächst weniger Arbeitsstunden als geplant an. Dieses Budget konnte später in die Entwicklungsphase verschoben werden. Anders als vorab geplant haben wir die Fertigstellung einer GUI-Applikation aus der Prototyping-Phase ausgeklammert.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Gab es Entwicklungen anderer Personen oder Institutionen, die Einfluss auf Deine Arbeiten und die Zielsetzung hatten? Wenn ja, worin bestand dieser und wie bist Du damit umgegangen?

Die Förderung durch den Prototype Fund ermöglicht es Softwareentwickler*innen, sich auf das Open-Source-Modell und freie Software zu fokussieren. Durch die Förderung wurde es für uns zur realen Möglichkeit, von Grund auf eine moderne Desktop-Applikation in Python zu entwickeln und dabei komplett auf Open-Source-Technologien aufzubauen.

Einerseits konnten wir auf ein reichhaltiges Open-Source-Ökosystem zurückgreifen - viele freie Module boten bereits benötigte Funktionalität, sodass wir in einigen Fällen unsere Funktionen aus bereits existierenden Modulen "zusammenzustecken" konnten. Andererseits stellten wir fest, dass einige dieser Module eher unregelmäßig maintained werden und oft mit wenig Kapazität auskommen müssen. Grundlegende digitale Infrastruktur entsteht oft in der Freizeit einiger weniger Entwickler*innen.

Unser Projekt wurde also zum Stakeholder anderer Open-Source-Projekte und folgerichtig auch zu Unterstützer*in und Contributor. Eine sehr motivierende Erfahrung war dabei die Effizienz und Geschwindigkeit der Zusammenarbeit auf GitHub: Bei einem Python-Modul, das man Anfang der Woche entdeckt, kann man mithilfe der Maintainer zum Ende der Woche bereits Contributor sein und einen Pull Request für ein fehlendes Feature beitragen.

Wir danken dem Bundesministerium für Bildung und Forschung für die Förderung sowie der Projektbetreuung durch die Open Knowledge Foundation und das Deutsche Luft- und Raumfahrtzentrum für die Unterstützung.