

Yappi - Developer Happiness

IP5 Project

Windisch, August 2025

Studenten: Xeno Isenegger, Gideon Monterosa

Fachbetreuer: Norbert Seyff, Nitish Patkar

Abstract

Inhaltsverzeichnis

1	Einleitung	6
1.1	Hintergrund und Motivation	6
1.2	Ziele und Vision	7
1.3	Fragestellungen	8
2	Hintergrund	10
3	State of the Art	11
3.1	Definition von Entwicklerzufriedenheit	11
3.2	Stand der Forschung und verwandte Arbeiten	12
3.3	Bestehende Lösungen und Wettbewerbsanalyse	14
4	Methoden	17
4.1	Projektmethodik	17
4.2	Prototypen	17
4.3	Proof of Concepts	17
5	Konzeptentwurf	18
5.1	Zugriffskontrolle über API Keys	18
5.2	Companion Apps	18
5.2.1	Integration in die Entwicklungsumgebung	18
5.2.2	Integration von Kalenderdaten	18
5.2.3	Integration von Gesundheitsdaten	21
5.3	Yappi Coach	22
5.4	Konzeptevaluation	22
6	Implementierung	23
6.1	Zugriffskontrolle über API Keys	23
6.2	Companion Apps	23
6.2.1	IntelliJ IDEA Companion	23
6.2.2	Calendar Companion	23

6.2.3	Health Companion	23
6.3	Deployment	23
7	Evaluation	24
7.1	Beantwortung der Fragestellung	24
8	Diskussion	25

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einleitung

1.1 Hintergrund und Motivation

TODO: evt Ausgangslage

TODO: aus Project Agreement: zu überarbeiten

Die Zufriedenheit von Entwicklerinnen und Entwicklern wird, wenn überhaupt, meist nur anhand der Menge ihrer geleisteten Arbeit gemessen. Dabei entstehen zwangsläufig Defizite, und ein halbjährliches Mitarbeitergespräch erweist sich oft als wenig wirksame Massnahme zur Problemlösung.

Dieses Projekt baut auf einer bestehenden Arbeit auf, in der eine Plattform zur Erfassung der Entwicklerzufriedenheit entwickelt wurde. Die Webapplikation Yappi ermöglicht es Entwicklerinnen und Entwicklern, ihre Zufriedenheit mit ihrer Arbeit und ihrer aktuellen Situation fortlaufend zu bewerten. Yappi erfasst emotionale Faktoren wie Happiness sowie weitere Zufriedenheitsindikatoren. Zusätzlich können spezifische Aufgaben und Arbeitstypen individuell bewertet werden. Die erhobenen Daten werden anonym auf Teamebene analysiert, um ein fundiertes Verständnis für die Stimmung innerhalb der Teams zu gewinnen.

Entwicklerinnen und Entwickler haben die Möglichkeit, ihre Zufriedenheit für verschiedene Teams zu erfassen, wodurch gezielte Analysen ermöglicht werden. Unternehmen erhalten dadurch wertvolle Einblicke, um das Arbeitsumfeld gezielt zu verbessern.

Unser Projekt baut auf Yappi auf und zielt darauf ab, die Erfassung der Zufriedenheit weiter zu optimieren. Es wird untersucht, wie die Daten noch präziser erfasst und ausgewertet werden können, um langfristige Verbesserungen zu unterstützen. Diese Arbeit dient als Grundlage für ein weiterfüh-

rendes Forschungsprojekt, das sich vertieft mit der Entwicklerzufriedenheit auseinandersetzt und zusätzliche Erkenntnisse gewinnen soll.

1.2 Ziele und Vision

TODO: Text aus dem Project Agreement noch zu überarbeiten

Yappi wird zu einer umfassenden Plattform weiterentwickelt, die nicht nur die Zufriedenheit misst, sondern sich nahtlos in den Arbeitsalltag integriert und wertvolle Handlungsempfehlungen liefert. Dazu werden folgende Kernaspekte umgesetzt:

Produktivitätsfaktoren identifizieren

Durch eine tiefere Analyse von Zufriedenheitsindikatoren sollen zentrale Faktoren ermittelt werden, die sich positiv oder negativ auf die Produktivität und das Wohlbefinden von Entwicklerinnen und Entwicklern auswirken. Diese Erkenntnisse werden genutzt, um Vorschläge zu Verbesserungsmaßnahmen abzuleiten.

Integration in den Arbeitsprozess

Yappi soll sich direkt in bestehende Arbeitsabläufe einfügen, um die Erfassung der Zufriedenheit möglichst intuitiv und effizient zu gestalten. Dies kann durch verschiedene Schnittstellen und Erweiterungen erfolgen, die eine nahtlose Interaktion ermöglichen.

Erweiterung um kontextbezogene Daten

Um ein umfassenderes Bild der Arbeitszufriedenheit zu erhalten, können weitere Einflussfaktoren berücksichtigt werden. Dazu gehören beispielsweise arbeitsbezogene Rahmenbedingungen oder individuelle Gesundheits- und Belastungsindikatoren. Diese Daten sollen helfen, ein besseres Verständnis für langfristige Trends und Zusammenhänge zu entwickeln.

Intelligente Analyse und Handlungsempfehlungen

Durch die Integration von AI schnittstellen können gezielte Analysen erstellt und individualisierte Empfehlungen abgeleitet werden. Dies kann sowohl auf individueller als auch auf Teamebene erfolgen, um nachhaltige Verbesserungen im Arbeitsumfeld zu fördern.

Fazit

Mit diesen Erweiterungen wird Yappi zu einem essenziellen Bestandteil des Entwickleralltags. Es bietet nicht nur eine präzisere Erfassung der Zufriedenheit, sondern liefert auch wertvolle Einblicke und Handlungsempfehlungen, um die Arbeitsbedingungen nachhaltig zu verbessern. Unternehmen erhalten fundierte Analysen und können gezielt Massnahmen ergreifen, um eine motivierte und produktive Entwicklergemeinschaft zu fördern.

1.3 Fragestellungen

TODO: Text aus dem Project Agreement noch zu überarbeiten

- A. Durch welche Technologien und Schnittstellen kann Yappi erweitert werden, um ein reibungsloses und einfaches Erfassen von Zufriedenheitsdaten zu ermöglichen?
 - a. Entwicklung von Entwickler-Tool-Plugins, die nahtlos in bestehenden Arbeitsumgebungen integriert werden können, um die Nutzung von Yappi angenehmer und effizienter zu gestalten. Diese Plugins sollen Entwicklern ermöglichen, direkt in ihrer bevorzugten Umgebung Feedback zu erfassen, ohne den Arbeitsfluss zu unterbrechen. Integration von Yappi in verschiedene Plattformen und Tools wie Webbrowser, IntelliJ, Microsoft Teams und Outlook.
- B. Wie können Gesundheitsdaten in die Auswertung der Entwicklerzufriedenheit einfließen?
 - a. Direkte Anbindung der Gesundheitsdaten-API, um relevante Gesundheitsmetriken wie Herzfrequenz, Schlafqualität oder Stresslevel automatisch in die Analyse der Entwicklerzufriedenheit zu integrieren. Dies ermöglicht eine genauere Einschätzung des Wohlbefindens und potenzieller Belastungsfaktoren.
- C. Wie kann Yappi Teams und Entwickler dabei unterstützen, aus den erfassten Zufriedenheitsdaten Handlungsempfehlungen abzuleiten, um die Zufriedenheit und Produktivität von Entwicklern zu erhöhen?
 - a. Entwicklung eines Yappi Coach, der anhand einer detaillierten Analyse der erfassten Daten gezielte Tipps zur Verbesserung der Arbeitsweise gibt. Beispielsweise könnte der Coach darauf hinweisen, dass Meetings nicht länger als 1,5 Stunden dauern sollten, da

längere Sitzungen die Zufriedenheit und Konzentration der Entwickler negativ beeinflussen können.

- b. Integration von KI-gestützten Diensten, die auf Basis der gesammelten Gesundheitsdaten sowie Zufriedenheits- und Produktivitätsmetriken individuelle Massnahmen vorschlagen. Diese KI-gestützten Empfehlungen können Teams dabei helfen, gezielt Optimierungen vorzunehmen, um die Arbeitsbedingungen und die Effizienz der Entwickler nachhaltig zu verbessern.

Kapitel 2

Hintergrund

TODO: unterkapitel für den Stand von Yappi vor dem Projekt

Kapitel 3

State of the Art

3.1 Definition von Entwicklerzufriedenheit

Entwicklerzufriedenheit wird in der Literatur als Balance zwischen positiven und negativen Erlebnissen bei der Arbeit definiert. Darunter versteht man eine Sequenz von Erfahrungen, bei der häufige positive Emotionen ein hohes Glücksgefühl erzeugen und häufige negative Erfahrungen das Gegenteil bewirken [1]. Auch Industriequellen fassen Entwicklerzufriedenheit als subjektives Wohlbefinden in Bezug auf Arbeitsinhalte und -umfeld auf, d.h. als Mass für Zufriedenheit, Freude oder innere Zufriedenheit bei der Arbeit [2]. Zufriedene Entwickler empfinden demnach mehr Arbeitsfreude und Inhaltlichkeit in ihrer Rolle, was eng mit der Arbeitsmotivation und dem Engagement bei der Arbeit verknüpft ist [3].

TODO: evt braucht es hier eine citation oder fussnote für den originalen Begriff

TODO: Ich bin mir unsicher ob die citation in diesem Absatz sauber genug ist

Eng verwendet mit der Zufriedenheit ist der Begriff **Flow**. In Anlehnung an Csikszentmihalyis Konzept beschreibt Flow einen Zustand von völliger Vertiefung und hohen Fokus beim Programmieren. Flow tritt dann auf, wenn die Anforderungen einer Aufgabe im Gleichgewicht mit den Fähigkeiten des Entwicklers stehen, wodurch man in einen Zustand von intensiver Konzentration gelangt. Zufriedene Entwickler gelangen einfacher in einen anhaltenden Flow-Zustand. Unzufriedenheit hingegen unterbricht diesen Flow, was zu Frustration führt und Schwierigkeiten führt, nach Unterbrechungen wieder in eine Aufgabe zurückzufinden. Teilnehmer einer Untersuchung berichten, negative Erlebnisse reißen einen aus dem Flow Zustand und machen es schwer, die Arbeit wieder aufzunehmen [1].

TODO: maybe absatz über burnout

Motivation und Zufriedenheit hängen eng zusammen, sind aber konzeptionell unterscheidbar. Motivierte Entwickler zeigen hohes Engagement und Fokus auf ihre Aufgaben, während Zufriedenheit eher durch allgemeines Wohlbefinden und gute Laune charakterisiert ist. Faktoren wie Autonomie, Kompetenzerleben und Zugehörigkeitsgefühl steigern die intrinsische Motivation von Entwicklern, was sich positiv auf ihre Zufriedenheit auswirkt. Zufriedenheit ist zugleich das Ergebnis und die Voraussetzung von Motivation, zufriedene Entwickler weisen in der Regel eine höhere Antriebskraft auf, was wiederum ihre Arbeitszufriedenheit weiter stärkt [3].

Schliesslich spielt auch das Team- und Organisationsklima eine fundamentale Rolle. Eine offene, unterstützende Kultur steigert nachweislich die Zufriedenheit von Entwicklern. Der DORA Report misst die Leistungsfähigkeit von Softwareentwicklungsteams anhand von vier Schlüsselkennzahlen: Deployment Frequency, Lead Time for Changes, Change Failure Rate und Time to Restore Service. Der DORA Report von Google basiert auf umfangreichen wissenschaftlichen Studien und gilt als Branchenstandard. Der Report von 2024 betont, dass Teams mit stabilem, ermutigendem Umfeld bessere Ergebnisse erzielen [4]. Positive Emotionen und ein Zugehörigkeitsgefühl im Team fördern den Gruppenzusammenhalt, was wiederum die Teamleistung von Teammitgliedern verbessert. Umgekehrt können toxische Kulturen oder ständig wechselnde Prioritäten die Zufriedenheit und Motivation untergraben, was sich negativ auf die Leistung auswirkt [1].

Somit unterstreichen sowohl akademische als auch industrielle Befunde: Entwicklerzufriedenheit entsteht in einem komplexen Zusammenspiel aus individuellen Faktoren (Flow, Motivation, ...) und Umfeldfaktoren (Team- und Organisationsklima, Arbeitskultur, ...).

3.2 Stand der Forschung und verwandte Arbeiten

QUESTION: Definitionen für Erfolgsparameter einfügen?

In den letzten Jahren haben zahlreiche Studien den Zusammenhang zwischen der Entwicklerzufriedenheit und Erfolgsparametern wie Produktivität, Codequalität und Mitarbeiterbindung untersucht. Bei einer grossangelegten Studie wurden 317 Softwareentwickler befragt und dabei 42 Konsequenzen von Unzufriedenheit sowie 32 Konsequenzen von Zufriedenheit beim Programmieren identifiziert. Die Ergebnisse zeigen, dass Entwicklerzufriedenheit messbare Auswirkungen auf den Entwicklungsprozess, die erzeugten Software-Artefakte und das Wohlbefinden der Person hat. So führt Unzufriedenheit zu einer

Reihe negativer Effekte: verzögerte Prozessabläufe, nachlässige Arbeitsweise und häufige unterbrechungen des Flows wurden als typische Folgen von negativer Stimmung genannt. Unzufriedene Entwickler berichten von langen langen Verzögerungen oder Qualitätsproblemen, weil Frustration sie aus dem Konzept brachte. Zufriedenheit hingegen wirkt sich positiv aus: Zufriedene Entwickler zeigen bessere Problemlösungsfähigkeiten, höhere Konzentration, berichten von einem anhaltenden Flow Zustand und lernen schneller. Die Ergebnisse legen nahe, dass Zufriedenheit die Codequalität begünstigt. Zufriedene Entwickler treffen sorgfältigere langfristige Entscheidungen. Ein Teilnehmer beschrieb, er dokumentiert seinen Code gründlicher und achten stärker auf Wartbarkeit, wenn er Zufrieden ist [5].

Neben akademischen Studien liefern auch Industireumfragen und Community-Studien wertvolle Einblicke. Der jährliche Stack Overflow Developer Survey etwa spiegelt wieder, dass weiterhin Verbesserungsbedarf besteht. laut der Umfrage 2024 bezeichnen sich nur rund 20% der Entwickler als wirklich zufrieden in ihrem Job, während etwa 80% unzufrieden oder "gelassen"(complacent) sind. Diese Zahl unterstreicht, dass ein grosser Teil der Entwicklergemeinschaft nicht glücklich im Arbeitsumfeld ist. Als Hauptgründe werden oft Faktoren wie schlechte Work-Life-Balance, zu viele Meetings oder fehlende Anerkennung genannt [6]. Eine Untersuchung von Zenhub kam zu ähnlichen Ergebnissen: Zwar gaben die meisten befragten Entwickler an, überwiegend zufrieden zu sein, doch lediglich 31% fühlten sich äusserst zufrieden in ihrer aktuellen Arbeitssituation [2].

Darüber hinaus tragen Anerkennung und Sinnhaftigkeit der Arbeit entscheidend zur Bindung bei. Wenn Entwicklerinnen und Entwickler stolz auf die Qualität ihrer Projekte sind und regelmässig Wertschätzung für ihre Arbeit erhalten, steigt sowohl ihre Zufriedenheit als auch ihre Loyalität gegenüber dem Unternehmen [1], [5]. Die empirischen Befunde deuten somit klar darauf hin, dass Entwicklerzufriedenheit kein rein „weiches“ Thema ist, sondern messbare Auswirkungen auf Produktivität, Codequalität und Personalbindung hat. Zufriedene Entwickler arbeiten effizienter, treffen sorgfältigere Entscheidungen und bleiben ihrem Team länger erhalten, während Unzufriedenheit mit erhöhten Kosten für Projekte und Rekrutierung verbunden ist.

TODO: evt. Methoden und Instrumente zur Messung der Entwicklerzufriedenhe

TODO: evt. Technologische und methodische Ansätze in verwandten Arbeiten

TODO: Forschungslücken

3.3 Bestehende Lösungen und Wettbewerbsanalyse

Der Markt bietet bereits verschiedene Tools und Plattformen, die Teilaspekte der Entwicklerzufriedenheit adressieren. Im Folgenden werden einige repräsentative bestehende Lösungen vorgestellt und hinsichtlich ihres Fokus und ihrer Lücken bewertet:

Officevibe: Ein Software-as-a-Service-Tool, das wöchentliche Pulse-Umfragen an Mitarbeitende verschickt. Ziel ist es, Engagement und Stimmung im Unternehmen kontinuierlich zu messen. Officevibe bietet anonyme wöchentliche Kurzbefragungen per E-Mail oder Chat an, deren Ergebnisse in übersichtlichen Dashboards für Teamleiter aufbereitet werden. Entwicklerteams erhalten dadurch Stimmungs-Trendkurven und allgemeines Mitarbeiterfeedback. Allerdings ist Officevibe eher generisch auf Mitarbeiterengagement ausgerichtet und liefert keine speziell auf Entwickler zugeschnittenen Kontextdaten, z.B. werden keine technischen Metriken aus der Softwareentwicklung einbezogen [7].

TeamMood: Ein schlankes Stimmungsbarometer für Teams, das täglich eine einfache Mood-Abfrage durchführt. TeamMood sendet jedem Teammitglied jeden Tag einen kurzen Prompt (per E-Mail, Slack, microsoft teams etc.), in dem die Person mit einem Klick ihre aktuelle Stimmung angibt. Die Antworten werden als anonymer Team-Stimmungsverlauf visualisiert, was es erlaubt, Trends über die Zeit zu erkennen. Die Hürde zur Teilnahme ist sehr niedrig (niedrigschwelliges Feedback). Jedoch erfasst TeamMood keine technischen Prozessmetriken (wie Code-Commits, Buildzeiten o.ä.) und bietet keine automatisierten Empfehlungen. Das bedeutet, dass Entwickler und Manager die Stimmungsverläufe selbst interpretieren und Massnahmen ableiten müssen, ohne direkte Handlungsempfehlungen durch das Tool [8].

Happimeter: Hervorgegangen aus einem Forschungsprojekt (u.a. TU Wien und MIT) setzt Happimeter auf Wearable Sensoren, um einen persönlichen Happiness-Score zu bestimmen. Entwickler tragen z.B. eine Smartwatch mit der Happimeter-App, die kontinuierlich physiologische Daten wie Herzfrequenz, Bewegung oder Schlaf erfasst. Ein Machine-Learning-Modell sagt daraus die aktuelle Stimmung bzw. den Stresslevel der Person voraus. Auf diese Weise sollen objektive Gesundheitsmetriken mit dem subjektiven Wohlbefinden verknüpft werden. Der Ansatz liefert interessante biometrische Einblicke (z.B. Stressspitzen

während der Arbeit), jedoch fehlt der direkte Bezug zur eigentlichen Entwicklungsarbeit. Happimeter weiss nichts über Tasks, Code oder Arbeitskontext, sodass die sensorbasierten Glücks-Werte ohne diesen Kontext schwer zu interpretieren sind [9].

Code Climate Velocity: Code Climate Velocity: Eine Datenplattform, die Entwicklungsmetriken aus Git-Repositories analysiert, um die Team-Performance zu bewerten. Velocity konzentriert sich voll auf quantitative Software-Engineering-Kennzahlen. Es misst etwa die Durchlaufzeit von Pull Requests, die Commit-Frequenz, die Review-Geschwindigkeit und diverse weitere Metriken der Entwicklungspipeline. Auch Industrie-Standards wie die vier DORA-Metriken (Deployment Frequency, Lead Time for Changes, Change Failure Rate, Mean Time to Restore Service) sind integriert. Dadurch erhalten Führungskräfte einen detaillierten Blick auf Code-Qualität, Liefergeschwindigkeit und Prozess-Effizienz. Allerdings fehlen subjektive Zufriedenheitsdaten vollständig, die menschliche Stimmungslage der Entwickler wird nicht erfasst. Etwaige Einflüsse von Motivation oder Frustration auf die gemessenen Leistungsindikatoren bleiben somit unsichtbar [10].

GitHub Insights: Als integrierter Teil von GitHub bietet Insights grundlegende Analysen zur Repository-Aktivität. In jedem GitHub-Repository steht ein Insights-Dashboard zur Verfügung, das Statistiken zu Commits, Pull-Request-Aktivitäten, Issue-Verläufen und Release-Frequenzen visualisiert. Teams können so ihre Entwicklungsaktivität und Geschwindigkeit verfolgen (“Wie viele PRs werden pro Woche gemerged? Wie oft wird deployed?” etc.). Diese Metriken sind wertvolle Aktivitätsstatistiken, berücksichtigen jedoch keine emotionalen Faktoren. GitHub Insights liefert also Kennzahlen zur Produktivität, blendet aber das Stimmungsbild der Entwickler aus. Etwa ob eine Phase hoher Commit-Rate auf Überstunden und Stress zurückzuführen ist, bleibt unklar [11].

Microsoft Viva Insights: Viva Insights ist Teil von Microsoft 365 und zielt darauf ab, durch Analyse von Arbeitsmustern die Produktivität und das Wohlbefinden von Mitarbeitenden zu verbessern. Die Plattform wertet vor allem Kalender- und Kommunikationsdaten aus (z.B. E-Mail- und Teams-Nutzung, Meeting-Häufigkeit). Entwickler erhalten z.B. Hinweise, wenn Meeting-Überlast droht, oder Vorschläge, regelmässige Fokuszeiten für ungestörtes Arbeiten einzuplanen. Führungskräfte sehen aggregierte Team-Insights, etwa ob viele Überstunden anfallen oder wenig Konzentrationsphasen vorhanden sind. Zwar gibt Viva nützliche Empfehlungen (z.B. “Schützen Sie wöchentlich 4 Stunden

Fokuszeit” oder “Vermeiden Sie Meetings über 1 Stunde”). Allerdings werden Wohlbefinden und Zufriedenheit nur indirekt aus den Verhaltensdaten abgeleitet, eine direkte Erfassung der Gefühlslage oder ein Bezug zu konkreten Entwickler-Tätigkeiten (Code, Tickets etc.) fehlt vollständig. Somit bleibt die emotionale Dimension in Viva Insights eher implizit und generalisiert [12].

Zusammenfassend zeigt sich: Es existieren Engagement-Tools, die Stimmungen einfangen (Officevibe, TeamMood), Engineering-Analytics-Tools, die Leistungsdaten auswerten (Code Climate, GitHub Insights), und Gesundheits-Tracker für Entwickler (Happimeter). Doch keine Lösung integriert alle drei Dimensionen, also Emotion, technischen Kontext und physisches Wohlbefinden in einer einheitlichen Plattform mit konkreten Handlungsvorschlägen. Genau hier setzt Yappi an. Yappi positioniert sich als Brückentechnologie, die die Lücken zwischen den bestehenden Ansätzen schliesst. Das beinhaltet die folgenden Kernfunktionen:

- das Erfassen von Zufriedenheitsdaten in den Arbeitsprozess integriert,
- Daten zu Commits und Kalenderereignissen automatisch als Kontext mit einbezieht,
- zusätzlich Gesundheitsdaten sammelt,
- per KI Coach handlungsorientierte Empfehlungen liefert.

Kapitel 4

Methoden

TODO: Kanban erwähnen

- 4.1 Projektmethodik
- 4.2 Prototypen
- 4.3 Proof of Concepts

TODO: arc42 erwähnen

Kapitel 5

Konzeptentwurf

5.1 Zugriffskontrolle über API Keys

TODO: Kapitel evt umbenennen; Aussage soll sein die das Yappi zu Plattform wird

5.2 Companion Apps

5.2.1 Integration in die Entwicklungsumgebung

5.2.2 Integration von Kalenderdaten

TODO XENO:

Ein vollständiges erfassen der Entwicklerzufriedenheit erfordert die Berücksichtigung aller Arbeitsaspekte, insbesondere auch Besprechungen (Meetings). Empirische untersuchungen zeigen, dass Entwickler durchschnittlich zwischen 10,9 Stunden und 16,5 Stunden pro Woche in Besprechungen verbringen. Ein erheblicher Teil dieser Zeit wird insbesondere bei grossen oder ungünstig terminierten Besprechungen als wenig produktiv bewertet. Eine hohe Meetingfrequenz kann den Arbeitstag fragmentieren und den Flow unterbrechen [13], [14].

Ziel dieses Konzeptentwurf ist die Entwicklung einer Kalendererweiterung für Yappi, welche durch das Integrieren des Persönlichen Kalender in die bestehenden Webanwendung Yappi und einer Companion-App welche durch eine Browser-Erweiterung einsicht in die Meetings der Entwickler ermöglicht. Die Erweiterung soll nach Ende eines Meetings automatisch, den Entwickler nach Feedback abfragen und die Ergebnisse davon in Yappi für

weitere Auswertungen bereitstellen.

Messung der Meetingqualität

TODO XENO: rework mit quelle

Zur Bewertung werden sieben Kriterien definiert. Diese sind in Anlehnung an Best Practices und gängige Meeting-Umfrageinstrumente ausgewählt:

1. Zielklarheit und Relevanz

Prüft, ob die Meetingziele klar formuliert und relevant waren. Eine präzise Agenda und eindeutige Zieldefinition steigern die Produktivität.

2. Inhaltliche Tiefe und Verständlichkeit

Bewertet, ob Themen angemessen tief und zugleich verständlich behandelt wurden. Unnötige Detailfülle ist zu vermeiden.

3. Zeitmanagement

Umfasst Pünktlichkeit beim Start und Einhaltung der geplanten Dauer. Studien empfehlen, Meetingzeiten bewusst zu verkürzen (z. B. 25 statt 30 Minuten), um Effizienz zu erhöhen.

4. Moderation und Beteiligung

Erfasst die Qualität der Moderation und die aktive Einbindung der Teilnehmenden. Eine hohe Beteiligungsquote gilt als Indikator für Interaktivität.

5. Ergebnisorientierung

Misst, ob konkrete Entscheidungen oder Aufgaben resultierten. Die Anzahl abgeschlossener Aktionspunkte kann als Kennzahl dienen.

6. Allgemeine Zufriedenheit

Ermittelt das subjektive Gesamturteil, beispielsweise auf einer Skala von 1 bis 10.

7. Meetingdauer

Vergleicht geplante mit tatsächlicher Dauer und bewertet die Angemessenheit.

Diese Kriterien ermöglichen ein umfassendes Bild der Meetingqualität und liefern Ansatzpunkte für Verbesserungen.

Kalenderintegration auf Basis offener Standards

Zur automatischen Erkennung relevanter Meetings wird eine Anbindung an das Kalendersysteme des Entwickler vorgesehen. Wichtig dabei ist es einen Offenen Standard zu verwenden um eine möglichst hohe Kompatibilität mit Kalendersystemen zu schaffen. Grundlage dazu bildet das weit verbreitete iCalendar-Format (ICS), das von nahezu allen gängigen Plattformen unterstützt wird. Das Konzept sieht vor, dass der Nutzer in Yappi auf seinem Profil den zugriff auf den Kalender hinterlegen kann. Regelmässig werden die auf einen vom Nutzer im Kalender erstellten Kalendereinträge in die Datenbank synchronisiert. Auf dieser Basis kann die Anwendung automatisch identifizieren, wann ein Meeting stattfindet. Dadurch lassen sich Feedback-Anfragen unmittelbar nach dem Ende einer Besprechung auslösen, ohne dass der Nutzer manuell eingreifen muss.

Die Synchronisation des Kalender erfolgt ausschliesslich lesend, um Eingriffe in persönliche Kalender zu vermeiden. Alle übermittelten Kalenderinformationen werden vertraulich behandelt und ausschliesslich für den definierten Zweck genutzt. Informationen zu Kalendereinträge sind nur für den Nutzer persönlich einsehbar.

Durch diese Anbindung entfällt das manuelle Erfassung von Kalendereinträgen. Meetings werden automatisch erkannt, Änderungen übernommen und die Nutzer benachrichtigt. So entsteht ein nahtloser integrierter feedback flow um die Zufriedenheitsdaten von Meetings für die Entwickler zu erheben.

Companion-App als Feedback-Trigger

Die Companion-App wird als Browsererweiterung umgesetzt, um eine direkte Interaktion mit den Entwicklern zum richtigen Zeitpunkt zu ermöglichen. Nach dem Ende eines im Kalender erfassten Meetings erhält der Nutzer eine Benachrichtigung, die zur Abgabe einer kurzen Bewertung auffordert. Diese zeitnahe Erhebung stellt sicher, dass Eindrücke und Wahrnehmungen noch frisch sind und die Datenqualität hoch bleibt. Die Interaktion erfolgt freiwillig, um Befragungsmüdigkeit zu vermeiden, und ist so gestaltet, dass die Beantwortung wenige Sekunden benötigt. Durch die Integration in den Browser wird kein zusätzlicher Systemwechsel nötig, wodurch die Hemmschwelle zur Teilnahme sinkt.

Datenverarbeitung und Auswertung

Die erhobenen Feedbackdaten werden ausschliesslich auf dem Yappi-Server verarbeitet. Vor der Auswertung erfolgt eine Anonymisierung, um Rückschlüsse auf einzelne Personen zu verhindern. Für jedes Meeting werden

aggregierte Kennzahlen, wie Durchschnittswerte der sieben Kriterien oder Verteilungen der Bewertungen, berechnet. Diese werden nur dann angezeigt, wenn eine vordefinierte Mindestanzahl an Rückmeldungen vorliegt. Neben der Auswertung einzelner Meetings können über längere Zeiträume Trends analysiert und Optimierungsmassnahmen evaluiert werden.

Integration in die Yappi Webanwendung

Die bestehenden Funktionen von Yappi werden um einen Bereich zur Meetinganalyse ergänzt. Entwickler sehen hier ihre eigenen Feedbackabgaben und offene Anfragen. Auf der Teamebene kann auf aggregierte, anonymisierte Auswertungen zu den Meetings ihres Teams zugegriffen werden. Die Konfiguration der Kalenderintegration und der Verbindung zur Companion-App wird im Nutzerprofil zentral verwaltet. Dadurch bleibt die Erweiterung vollständig in die bestehende Systemarchitektur eingebettet und benötigt keine separaten Plattformen.

5.2.3 Integration von Gesundheitsdaten

Zusätzlich zur Meetinganalyse kann Yappi um die Integration von Gesundheits- und Belastungsdaten erweitert werden, sofern der Nutzer dies explizit freigibt. Mögliche Datenquellen sind Wearables oder manuell erfasste Angaben zu Schlaf, Bewegung oder Arbeitsbelastung. Durch die Kombination von Meetingfeedback und Gesundheitsindikatoren lassen sich Korrelationen erkennen, beispielsweise zwischen hoher Meetingdichte und erhöhter Ermüdung. Diese Informationen können genutzt werden, um Arbeitsprozesse gesundheitsförderlich zu gestalten.

Die folgenden Gesundheitsmetriken sind für unsere Arbeit relevant, da sie einen interessanten Einfluss auf die Zufriedenheit von Softwareentwicklern haben.

TODO: Nochmals anschauen

1. Schlafdauer

Untersuchungen zeigen, dass längere Schlafdauer mit höherer Arbeitszufriedenheit einhergeht: Männer mit mehr Schlaf berichten signifikant grössere Zufriedenheit am Arbeitsplatz als solche mit verkürztem Schlafpensum.

2. Ruheherzfrequenz

Eine erhöhte Ruheherzfrequenz spiegelt häufig chronisch erhöhte Stresslevel wider. In einer Querschnittsstudie war hoher Job-Strain mit er-

höherer RHR assoziiert, und gleichzeitig berichten Beschäftigte in stark belastenden Jobs über signifikant geringere Zufriedenheit.

3. Stress (HRV)

Die Herzratenvariabilität (HRV) ist ein objektiver Marker für die autonome Balance: Niedrige HRV-Werte korrelieren konsistent mit höheren Stresslevels am Arbeitsplatz. Da hohe Stresslevels nachweislich die Zufriedenheit verringern, eignet sich HRV-Monitoring als indirektes Mass für potenzielle Unzufriedenheit.

4. Aktivitätsminuten und Schritte

Regelmässige moderate Bewegung, erfasst über Schritte und aktive Minuten, steht in direktem Zusammenhang mit höherer Jobzufriedenheit. Eine aktuelle Studie belegt, dass wöchentliche Freizeitaktivität signifikant positive Effekte auf die Zufriedenheit am Arbeitsplatz hat.

TODO: zu viel?

Diese Kennzahlen erlauben es, Zusammenhänge zwischen erholungsbezogenen Faktoren und der subjektiven Arbeitszufriedenheit zu erkennen und so gezielte Massnahmen zur Förderung des Wohlbefindens und der Leistungsfähigkeit abzuleiten.

5.3 Yappi Coach

5.4 Konzeptevaluation

TODO XENO: Fragebogen

Kapitel 6

Implementierung

6.1 Zugriffskontrolle über API Keys

TODO: Quelle für spring Security Architektur

<https://docs.spring.io/spring-security/reference/servlet/architecture.html>

6.2 Companion Apps

6.2.1 IntelliJ IDEA Companion

6.2.2 Calendar Companion

TODO XENO:

6.2.3 Health Companion

6.3 Deployment

TODO XENO:

TODO: nicht sicher ob es auch ein entsprechendes Kapitel im Konzeptdesign benötigt

TODO: UML Deployment Diagramm

Kapitel 7

Evaluation

7.1 Beantwortung der Fragestellung

TODO XENO:

Kapitel 8

Diskussion

Literatur

- [1] D. Graziotin und F. Fagerholm, “Happiness and the productivity of software engineers”, in *Rethinking Productivity in Software Engineering*, C. Sadowski und T. Zimmermann, Hrsg., Berkeley, CA: Apress, 2019, S. 109–124, ISBN: 9781484242209 9781484242216. DOI: 10.1007/978-1-4842-4221-6_10. besucht am 8. Aug. 2025. Adresse: https://link.springer.com/10.1007/978-1-4842-4221-6_10.
- [2] “2022 software developer happiness report - developer productivity data”, Zenhub, besucht am 8. Aug. 2025. Adresse: <https://www.zenhub.com/reports/software-developer-happiness>.
- [3] C. França, F. Q. B. da Silva und H. Sharp, “Motivation and Satisfaction of Software Engineers”, *IEEE Transactions on Software Engineering*, Jg. 46, Nr. 2, S. 118–140, Feb. 2020, ISSN: 1939-3520. DOI: 10.1109/TSE.2018.2842201. besucht am 8. Aug. 2025. Adresse: <https://ieeexplore.ieee.org/document/8370133>.
- [4] “Announcing the 2024 DORA report”, Google Cloud Blog, besucht am 8. Aug. 2025. Adresse: <https://cloud.google.com/blog/products/devops-sre/announcing-the-2024-dora-report>.
- [5] D. Graziotin, F. Fagerholm, X. Wang und P. Abrahamsson, *What happens when software developers are (un)happy*, 23. Apr. 2018. DOI: 10.48550/arXiv.1707.00432. arXiv: 1707.00432. besucht am 8. Aug. 2025. Adresse: <http://arxiv.org/abs/1707.00432>.
- [6] “Not just a vibe, the stack overflow developer survey is really here - stack overflow”, besucht am 9. Aug. 2025. Adresse: <https://stackoverflow.blog/2025/05/29/not-just-a-vibe-the-stack-overflow-developer-survey-is-really-here/>.
- [7] “With courier, officevibe enables open communication for teams in the workplace.”, courier, besucht am 9. Aug. 2025. Adresse: <https://www.courier.com/use-cases/workleap-officevibe>.

- [8] “Tech team health check: An overall guide”, besucht am 9. Aug. 2025. Adresse: <https://www.revelo.com/blog/tech-team-health-check-a-template-for-your-organization>.
- [9] P. Budner, J. Eirich und P. A. Gloor, “*Making you happy makes me happy*” – *Measuring Individual Mood with Smartwatches*, 14. Nov. 2017. DOI: 10.48550/arXiv.1711.06134. arXiv: 1711.06134. besucht am 9. Aug. 2025. Adresse: <http://arxiv.org/abs/1711.06134>.
- [10] “How software engineering intelligence platforms boost developer productivity”, InfoWorld, besucht am 9. Aug. 2025. Adresse: <https://www.infoworld.com/article/2335502/how-software-engineering-intelligence-platforms-boost-developer-productivity.html>.
- [11] “Git analytics: Challenges, tools & key metrics”, besucht am 9. Aug. 2025. Adresse: <https://axify.io/blog/git-analytics>.
- [12] zachminers. “Introduction to viva insights”, besucht am 9. Aug. 2025. Adresse: <https://learn.microsoft.com/en-us/viva/insights/introduction>.
- [13] V. Stray und N. B. Moe, “Understanding coordination in global software engineering: A mixed-methods study on the use of meetings and Slack”, *Journal of Systems and Software*, Jg. 170, S. 110717, Dez. 2020, ISSN: 0164-1212. DOI: 10.1016/j.jss.2020.110717. besucht am 8. Aug. 2025. Adresse: <https://www.sciencedirect.com/science/article/pii/S0164121220301564>.
- [14] A. Meyer, E. T. Barr, C. Bird und T. Zimmermann, “Today was a Good Day: The Daily Life of Software Developers”, eng, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Jg. 47, Nr. 5, S. 863–880, Mai 2021, ISSN: 0098-5589. DOI: 10.1109/TSE.2019.2904957. besucht am 8. Aug. 2025. Adresse: <https://www.zora.uzh.ch/id/eprint/170375/>.