

Jihočeská univerzita v Českých Budějovicích

Přírodovědecká fakulta



Automatické sledování času pracovníků v oblasti profesionálních služeb

Diplomová práce

Bc. Vladimír Kročák

Vedoucí práce: Ing. Martin Čížek, MBA

České Budějovice 2019

Bibliografické údaje

Bc. Vladimír Kročák, 2019: Automatické sledování času pracovníků v oblasti profesionálních služeb. [Automatic employee time tracking in professional services. Mgr. Thesis, in Czech.] - 53p., Faculty of Science, University of South Bohemia, České Budějovice, Czech Republic.

Anotace

Tato diplomová práce se zabývá problematikou vykazování a sledováním času pracovníků v IT službách. Analyzuje (v době psaní této diplomové práce) dostupné nástroje používané pro tyto účely. Popisuje stav v konkrétní firmě (Orchitech Solutions, s.r.o.). Na závěr diplomové práce je z těchto nástrojů sestaven návrh řešení pro danou firmu.

Klíčová slova

Sledování času, vykazování, návrh

Anotation

This diploma thesis deals with the problematic of time tracking of employees in IT services. It analysis (at the time of writing this thesis) available tools used for these purposes. Describes actual example firm (Orchitech Solutions, s.r.o.). A solution for this company is designed from these tools at the end of this thesis.

Key words

Time tracking, time logging, design

Prohlášení

Prohlašuji, že svoji diplomovou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své diplomové práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích, dne 10. prosince 2019

Podpis:

Poděkování

Tímto bych chtěl poděkovat vedoucímu práce Ing. Martinu Čížkovi, MBA a garantovi diplomové práce Mgr. Jiřímu Pechovi Ph.D. za průběžné připomínky a možnost sepsat mou diplomovou práci.

Zároveň bych tímto chtěl poděkovat své rodině a přátelům, kteří mě podporovali v jejím dokončení „*a nenechali mě to vzdát*“.

Slovník pojmů

Tento slovník shrnuje výrazy použité v této diplomové práci. Vychází z názvosloví používaného v agilních vývojových metodikách (specificky Scrum [29]), projektových nástrojů (např. Jira [28]) a výrazy běžně využívané v Orchitech Solutions, s.r.o. Tento slovník pojmů zcela nenahrazuje znalost názvosloví využívaného v IT firmách jako je Orchitech Solutions, s.r.o.

Tento slovník shrnuje výrazy použité v diplomové práci. Vychází z názvosloví používaného v agilních vývojových metodikách (specificky Scrum [29]), projektových nástrojích (např. Jira [28]) a výrazy běžně využívané v Orchitech Solutions, s.r.o. Tento slovník pojmů zcela nenahrazuje znalost názvosloví využívaného v IT firmách, jako je Orchitech Solutions, s.r.o.

AFK „*Away from keyboard*“. Zkratka (slang) označující, že uživatel se nenachází u PC a nemůže komunikovat.

bug „*Chyba*“ v softwaru. Zároveň se tento název přenáší na issue zabývající se popisem a nápravou této chyby.

code review „*Kontrola zdrojového kódu*“. Činnost, kdy vývojář kontroluje kód napsaný jiným vývojářem. Cílem je omezit případné chyby a udržet konzistenci v kódu aplikace.

CRUD „*Create, Update, Delete*“. Zkratka označující základní běžné operace nad daty (vytváření, upravování, mazání).

deprecated Daná funkcionality není nadále podporována. V budoucích verzích aplikace bude odstraněna.

event „*Událost*“. V této práci je typicky míněna základní datová entita v ActivityWatch obsahující data, délku a časovou známku měřené události.

feature request Issue s novým požadavkem. Typicky se jedná o přidání funkcionality, nebo úpravu stávající funkcionality vyvíjené aplikace.

home office „Práce z domova“. Obvykle označuje, že zaměstnanec pracuje mimo kancelář.

IDE „*Integrated development environment*“. V rámci této diplomové práce autor většinou odkazuje na jako aplikace pro psaní zdrojového kódu (např. IntelliJ IDEA, NetBeans nebo Eclipse).

IoT „*Internet of things*“. V textu označuje typicky fyzické zařízení sloužící pro sběr dat nebo plnící jinou funkci. Tato zařízení jsou vybavena síťovou konektivitou, což umožňuje tato zařízení zpracovat nebo z nich získávat data.

issue „*An issue is simply a unit of work within Jira that will be traced through a workflow, from creation to completion. It can range from representing a single unit of work, like a simple task or bug, to a larger parent work item to be tracked, like a story or an epic*“. [28]

Velmi obecný název pro ucelenou a popsitelnou jednotku zadání. Podle typu mohou být dále specifikovány (bug, story/feature request). Tento název se obecně využívá v nástrojích, jako je Jira a Redmine.

JS JavaScript (zkratka).

MH „*Man Hour*“ („*Člověkohodina*“). Zkratka znamenající jednu pracovní hodinu jednoho zaměstnance.

proof of concept Minimální implementace, jejímž cílem je typicky zjistit, zda je vybrané řešení možné uskutečnit.

query „*Dotaz*“. V rámci této diplomové práce je většinou míněna ActivityWatch query [15]. Pojmenování významu vychází z jazyků, jako je SQL ap.

vykázanost pracovníků V rámci této práce je míněn poměr mezi pracovní dobou zaměstnance a časem, který reálně vykázal na jednotlivých pracovních činnostech.

watcher V této práci primárně označuje SW modul ActivityWatch, který v pravidelných intervalech kontroluje aktivitu uživatele (např. otevřené záložky aj.) a tyto

informace loguje do napojeného ActivityWatch serveru. Může označovat i jiný SW s obdobnou funkcionalitou.

UI „*User Interface*“. Zkratka označující uživatelské rozhraní aplikace.

UX „*User Experience*“. Zkratka souhrnně popisující chování aplikace (interakce s uživatelem).

Obsah

| | |
|---|-------------|
| Slovník pojmů | V |
| Obsah | VIII |
| 1 Úvod | 1 |
| 2 Cíle práce | 3 |
| 3 Přehled dostupných nástrojů | 4 |
| 3.1 Druhy nástrojů | 4 |
| 3.2 Sledované parametry | 6 |
| 3.3 Konkrétní ukázky | 9 |
| 4 Případová studie Orchitech Solutions, s.r.o. | 16 |
| 4.1 Vykazování v Orchitech Solutions, s.r.o. | 16 |
| 4.2 Nejčastější prohřešky | 18 |
| 4.3 Používané nástroje | 19 |
| 4.4 Způsob vykazování | 19 |
| 4.5 Požadavky | 19 |
| 4.6 Závěr | 20 |
| 5 Návrh vlastního řešení | 21 |
| 5.1 Vybraná kritéria | 21 |
| 5.2 Vybrané prvky | 22 |
| 5.3 Sestavení řešení | 22 |
| 6 Implementace řešení | 26 |
| 6.1 Redmine | 26 |
| 6.2 TTRebel | 26 |
| 6.3 IoT | 30 |
| 6.4 ActivityWatch | 30 |

| | | |
|----------|--------------------------------------|-----------|
| 6.5 | ActivityWatch Query | 33 |
| 6.6 | Sestavení finálního řešení | 37 |
| 7 | Závěr | 38 |
| | Literatura | 39 |
| | Seznam obrázků | 42 |
| | Seznam tabulek | 43 |
| | Seznam ukázek zdrojového kódu | 44 |

1. Úvod

Autor v této práci využívá názvosloví, které vychází z názvosloví běžně používaného při vývoji softwaru pomocí agilních metodik (specificky Scrum viz [29]), projektových nástrojů (např. Jira [28]) a výrazy běžně využívané v Orchitech Solutions, s.r.o. Krátké shrnutí těchto výrazů a jejich významu může být nalezeno v přílohách ve slovníku pojmů na str. č. V této diplomové práci. Tento slovník pojmů nenahrazuje znalost názvosloví běžně využívaného v IT firmách, jako je Orchitech Solutions, s.r.o. V rámci této práce budou popsány aktuálně dostupné nástroje a způsob vykazování a evidence času běžně využívaného v IT firmách. Následně z těchto nástrojů bude sestaven prototyp pro vybranou firmu.

Toto téma bylo sestaveno na žádost Ing. Martina Čížka, MBA. Praktickým cílem této práce je vylepšit vykázanost pracovníků společnosti Orchitech Solutions, s.r.o. Vykázaností je ve smyslu této práce míněn poměr mezi pracovní dobou zaměstnance a časem, který reálně vykázal na jednotlivých pracovních činnostech.

Jako subjekt pro praktickou analýzu, návrh a implementaci vlastních změn je využita společnost Orchitech Solutions, s.r.o.

Samotná textová část diplomové práce je rozdělena do následujících kapitol.

V analytické nebo rešeršní části (viz kapitola 3) si popíšeme aktuálně používané a dostupné nástroje pro evidenci času pracovníků. Tato kapitola obsahuje jak rešeršní nástrojů pro evidenci času v IT firmách, tak i rešeršní nástrojů pro sledování času jednotlivých zaměstnanců.

V kapitole 4 je popsán aktuální stav a způsob vykazování času pracovníků ve společnosti Orchitech Solutions, s.r.o. Zároveň v této kapitole budou stanoveny požadavky pro praktickou část práce.

V kapitole 5 jsou vybrány na základě předchozí analýzy konkrétní nástroje pro vlastní implementaci řešení. Jsou zde sepsány důvody pro výběr konkrétních technologií.

V praktické části (popsané v kapitole 6) jsou rozepsány kroky a prvky, které musely být implementovány pro vytvoření prototypu.

V závěru této práce jsou shrnuty výsledky diplomové práce. Je zde popsáno, jak

se podařilo implementovat cíle této diplomové práce. Zároveň jsou v poslední kapitole popsány důsledky pro společnost Orchitech Solutions, s.r.o. a návrhy pro další rozvoj prototypu, ať v rámci komunity, nebo i čistě v rámci společnosti Orchitech Solutions, s.r.o.

2. Cíle práce

Student vytvoří řešení pro automatické sledování času aktivit pracovníků v oblasti profesionálních služeb, sběr dat a návaznou analytiku, která umožní snadné zaúčtování času na projekty, projektové úkoly, zákazníky apod.

Dílčí cíle

- Student provede rešerši alternativních řešení a popíše jejich slabiny. Dále navrhne architekturu řešení a vytvoří návrh konkrétní implementace. Implementace by měla využívat existující open-source komponenty.
- Student vytvoří prototyp systému sběru dat na základě případové studie zadané školitelem. Předpokládá se sběr aktivity z operačních systémů Windows a Linux a vybraných aplikací, detekce stavu „*away from keyboard*“ a sběr hodnot z vybraného IoT senzoru – pohyb nebo identifikace přítomnosti dané osoby v místnosti.
- Do systému bude možné importovat data z existující mobilní aplikace pro sledování aktivit a start-stop systému pro zaznamenávání času.
- Student navrhne analytiku, která v dané případové studii umožní automaticky zaúčtovat většinu času na předdefinované úlohy v projektovém systému.
- Student své řešení popíše a zhodnotí.

3. Přehled dostupných nástrojů

V této kapitole si představíme některé již existující nástroje pro time-tracking. Cílem kapitoly je představit různé typy nástrojů a jejich alternativy. Představíme si zástupce jak softwarových, tak i některých fyzických zařízení. Některé z těchto nástrojů budou následně použity v praktické části této diplomové práce (viz 6).

Nedílnou součástí této kapitoly bude seznam nejdůležitějších požadavků na jednotlivé nástroje. Nebude se jednat o striktně definované parametry, ale spíše o obecná kritéria stanovená především s ohledem na praktickou část práce. Bude uveden jejich krátký popis a zdůvodnění jejich výběru.

Nástroje budou rozřazeny do několika kategorií. Ve většině případů ale jednotlivé programy zajišťují více funkcí, a rozdělení není vždy možné. Nástroje budou tedy rozděleny především rámcově dle nejobvyklejšího použití.

Na závěr kapitoly budou uvedeny konkrétní případy těchto nástrojů. Bude se jednat především o známější zástupce daných nástrojů a jejich nejznámější alternativy. Výběr těchto nástrojů bude zaměřen především na nástroje používané v odvětví profesionálních služeb a vývoje SW.

3.1 Druhy nástrojů

Následující rozdělení nástrojů je pouze orientační. V mnohých případech je nelze zcela jednoznačně zařadit do jediné kategorie. To je způsobeno především tím, že tyto nástroje typicky nenabízejí více než jedinou službu nebo funkcionalitu. Jedná se tedy spíše o orientační rozdělení dle nejčastějšího použití.

3.1.1 Analytické nástroje

Tyto nástroje slouží spíše pro analýzu a zobrazení statistických dat z různých zařízení a nejsou specifické pro vykazování odpracovaného času. Slouží pro ukládání, zpracování a zobrazení dat z IoT zařízení a různých zdrojů logů.

V praktické části práce mohou být tyto nástroje využity pro zpracování anebo zobrazení některých dat.

3.1.2 Ukládání dat

Oproti dalším nástrojům se jedná o čistě atomické nástroje, které řeší pouze ukládání dat. Tyto nástroje mají typicky velmi dobře definované rozhraní.

V podkapitole 3.3.2 budou opětovně uvedeny spíše technické nástroje. Budou zmíněny příklady některých databází a služeb, které lze využít pro uložení.

V praktické části práce mohou být využity při vývoji, buď pro ukládání vlastních dat, nebo jako součást jiné aplikace.

3.1.3 Projekt management

Tento typ nástrojů se využívá především k záznamu a vývoji softwaru nebo agilním vedení projektů. Jeho primárním účelem je udržovat seznam jednotlivých issue (bugy, feature requesty atd.) a jejich stav.

Time tracking zde slouží především k záznamu času, který byl třeba pro implementaci dané issue.

Může se jednat o pouhé bug trackery, které time tracking nenabízejí (např. Github Issues nebo Biebucket issues). V takových případech můžeme nalézt nástroje, které chování těchto bug trackerů rozšiřují, nebo existuje integraci s jiným nástrojem.

V rámci praktické části bude takovýto nástroj sloužit především jako cílový systém, do kterého budou data zaznamenávána. Data z těchto nástrojů jsou použita pro generování účetních výpisů nebo přímé vyúčtování klientovi.

3.1.4 Tlačítka

Jedná se o aplikace spoléhající především na vstup uživatele. Uživatel typicky ručně přepíná/zadáva v aplikaci svou stávající aktivitu. Při dokončení nebo změně aktivity musí v aplikaci opětovně ručně činnost přerušit.

Dle rozsahu informací sbíraných od uživatele se může jednat o nejdetailnější zdroj dat. Spoléhá ale pouze na vstup uživatele. Při zpětném dovykazování aktivity se nemusí jednat o zcela pohodlné řešení.

3.1.5 Trackers

V této kategorii můžeme nalézt především automatické trackery, které zaznamenávají činnost uživatele. Může se jednat o jednoduché utility. Mohou například kontrolovat aktivní otevřená okna v operačním systému nebo záložky ve webovém prohlížeči. Může se také jednat o specializovanější nástroje, které zaznamenávají další činnosti.

Tyto nástroje typicky produkují větší množství dat, které je nutné dále zpracovávat.

3.1.6 Přepínací kostky

V této sekci uvedu některá fyzická zařízení použitelná pro time-tracking. Uvedeme si zde pár zástupců IoT zařízení. Jedná se o principiálně jednoduchá IoT zařízení sloužící přímo pro sledování aktivity uživatele.

Jedná se o principiálně jednoduchá IoT zařízení obsahující pohybová čidla kontrolující náklon anebo pohyb zařízení. Při otočení této „kostky“ následně SW aplikace přepne režim a spustí asociovaný tracker.

Tímto způsobem lze indikovat změny činností. Takto lze indikovat pouze typ činnosti (např: vývoj, meeting, code-review, pauza atd.). Tento způsob vykazování je ovšem poměrně nedostatečný, v případě, že uživatel musí vykazovat mnohem podrobnější informace o své činnosti, nebo rozkládá svůj čas mezi různé projekty anebo issue.

Jedná se o celkem zajímavé řešení. Při kombinaci s daty z dalších trackovacích systémů (viz podkapitola 3.1.5) mohou doplnit chybějící kontext. Díky tomu by mělo být snadnější zpětně určit prováděné činnosti v daný moment.

3.2 Sledované parametry

3.2.1 Import a export dat

U všech nástrojů bude kontrolováno, zda a jakým způsobem umožňují import a export dat. Jedná se o kritickou vlastnost, která umožňuje data dále zpracovat. Data mohou být následně migrována či agregována v dalších nástrojích.

Export dat bude kritický především u nástrojů pro získání dat. Tyto nástroje budou použity především pro jednoduché zaznamenávání časových intervalů. U těchto nástrojů

budou další funkce, například analýza dat spíše jako dodatečná vlastnost.

Import dat umožňuje především migraci k novým nástrojům a agregování nebo kombinování různých vstupních nástrojů. Tato vlastnost bude kritická především pro systémy, které slouží pro analýzu činnosti a následně vlastní vykazování činnosti.

Při exportu dat bude brán ohled především na rozsah dat dostupných pro export a zda je možné exportovat samotná zdrojová data. Některé nástroje umožňují pouze souhrnné statistiky. V některých případech se může jednat o technický limit, nebo v případě komerčních produktů o limit nastavený platebním plánem.

Posledními požadavky na import a export jsou podporované formáty dat a způsob importu a exportu dat. V tomto směru se budeme zaměřovat především na strojové zpracování dat. S tímto přímo souvisí následující dvě podkapitoly 3.2.2 a 3.2.3.

3.2.2 API

Dostupné API je prvek, který především umožňuje automatizovat některé činnosti. Možnost využít API bude představovat kritickou výhodu u nástrojů, které nemají, nebo nebudou mít již předpřipravené integrace s jinými nástroji.

Ve většině případů se budeme zaměřovat především na možnosti importu a exportu dat. To umožní vytvořit vlastní integrace různých nástrojů a případnou snazší migraci.

Dostupné a otevřené API umožňuje aplikaci rozšířit a přidat novou vlastní integraci. V takovém případě bude ale brán ohled i na již existující integrace s jinými nástroji.

3.2.3 Množství integrací

Množstvím integrací je míněn počet integrací s dalšími nástroji, ať už od stejného výrobce, nebo i s nástroji třetích stran. Každá další integrace zvyšuje znovu použitelnost daného nástroje a usnadňuje vytvoření vlastního řešení.

Množství integrací může korelovat s používaností takového nástroje. To je platné především u komerčních nástrojů, kdy firma vyvíjející tento nástroj usnadňuje zapojení daného nástroje do již existujícího ekosystému.

V některých případech to bude představovat hlavní výhodu daného nástroje. Z důvodu technické náročnosti bude vždy preferováno již vytvořené napojení před vytvářením

vlastní integrace.

3.2.4 Open-source

Otevřenost řešení může být zavádějícím parametrem. Můžeme předpokládat, že open-source nástroje budou splňovat některá výše uvedená kritéria, jako je například dostupné API a vhodný způsob importu a exportu dat.

Tyto projekty ovšem nabízejí největší možnosti si daný nástroj upravit dle vlastních potřeb. Toho může být využito v praktické části k sestavení vlastního řešení na míru.

Open-source projekty ovšem mohou záviset výhradně na práci komunity, a mohou tedy trpět při nedostatku aktivních přispěvatelů. V takovém případě je nutné zvážit, zda se do projektu aktivně zapojit a pokusit se jej rozšiřovat či dát přednost zavedené komerční alternativě.

Některé projekty nutně nemusí splňovat všechny požadavky pro open-source, ačkoliv jejich tvůrci kód uveřejnili a udržují jej otevřeně (např. Toggle). V takových případech je běžné, že tvůrci poskytují placenou verzi služby, ze které financují chod své firmy.

V následující podkapitole 3.3 budou vždy uvedeny open-source alternativy ke komerčním produktům.

3.2.5 Aktivnost projektu

Aktivnost projektu je relativně subjektivní záležitost. Mnohé již zavedené a odzkoušené nástroje nemusí projevovat vysokou aktivitu a stejně se může jednat o spolehlivé řešení.

V případě využít podobného nástroje je ale vhodné se spoléhat na aktivní a aktuální nástroj. Použít zastaralé nástroje, nebo nástroje, jejichž vývojový cyklus nese známky úpadku, se může znatelně projevit ve výsledném produktu.

V případě použití open-source projektů nejsou často k dispozici profesionální audity, a musíme tedy spoléhat na veřejně dostupné metriky. Mezi tyto metriky může patřit počet vývojářů, aktivita komunity, rychlost vydávání nových verzí atd.

Tyto údaje mohou pomoci při výběru takového nástroje [1].

V případě komerčního produktu bychom se měli zaměřit na audity, hodnocení nejenom konkrétního nástroje, ale i vlastní firmy, která jej vyvíjí.

3.2.6 Cena a financování

Tento parametr bude sledován především u komerčních produktů. Při sestavování praktického řešení se může jednat o faktor, který může ovlivnit finální výběr nástrojů.

Cena jednotlivých produktů se může výrazně lišit a často se můžeme setkávat s různými platebními režimy a platebními plány pro různě velké společnosti. Mezi nejčastějšími platebními režimy jsou nabízeny subskripce dle počtu účtů nebo je omezena dostupnost některých služeb (např: kvalitnější analytika, support atd.).

V některých případech je služba nabízena pro jednotlivce zdarma, nebo je nabízeno vyzkoušení si dané služby. Ve většině takových případů je tato služba omezená jiným způsobem.

3.3 Konkrétní ukázky

3.3.1 Analytické nástroje

Graphana

Nástroj pro generování dashboard a grafů. Velmi oblíbený nástroj pro generování grafů a vizualizaci dat z různých zdrojů.

Kibana

Slouží ke zpracování logů a generování dashboard na základě Elasticsearch.

3.3.2 Ukládání dat

Ukládání dat pro potřeby této práce není zcela podstatná technologie použitá pro ukládání dat. Většina aplikací ukládá data do běžných databází (SQL, NOSQL atd.) nebo formou indexu (např. Apache Lucen nebo Elasticsearch).

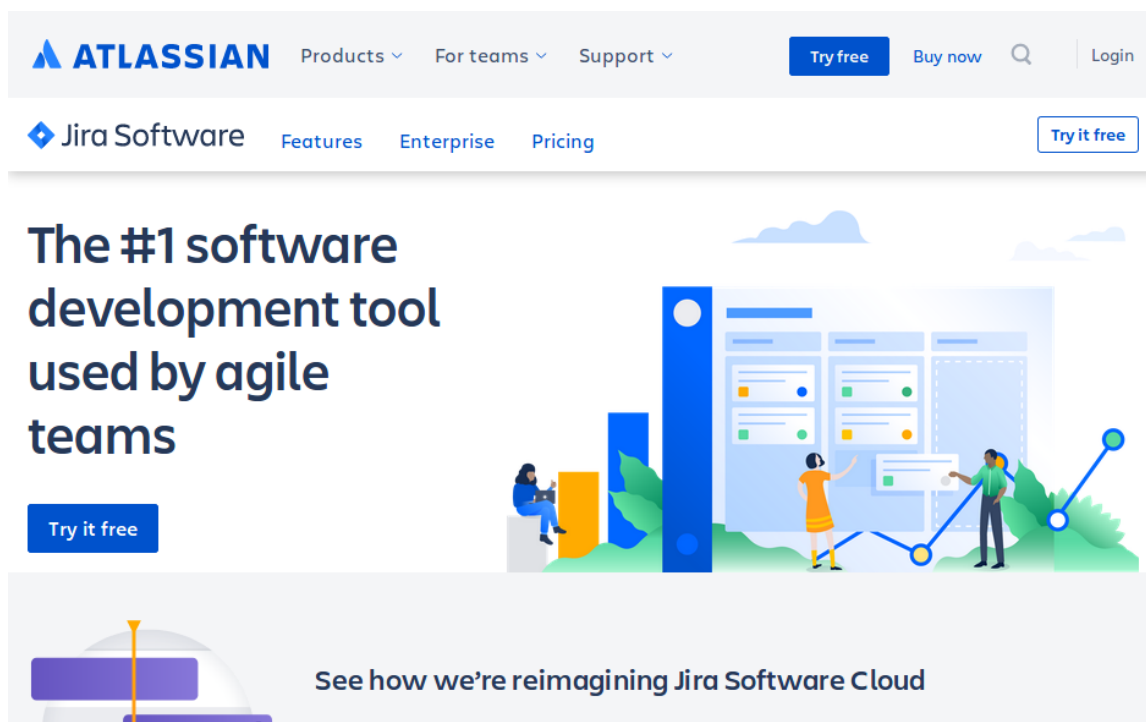
Pro uložení dat mohou být samozřejmě využity i databáze přímo určené k manipulaci s časovými řadami, jako je např. InfluxDB.

3.3.3 Project management

Jira

Komerční bug tracker od společnosti Atlassian [2].

Hlavní výhodou je vysoká integrace s dalšími nástroji od společnosti Atlassian (např. Bitbucket, Trello) a ohromné množství integrací s těmito nástroji. [3]

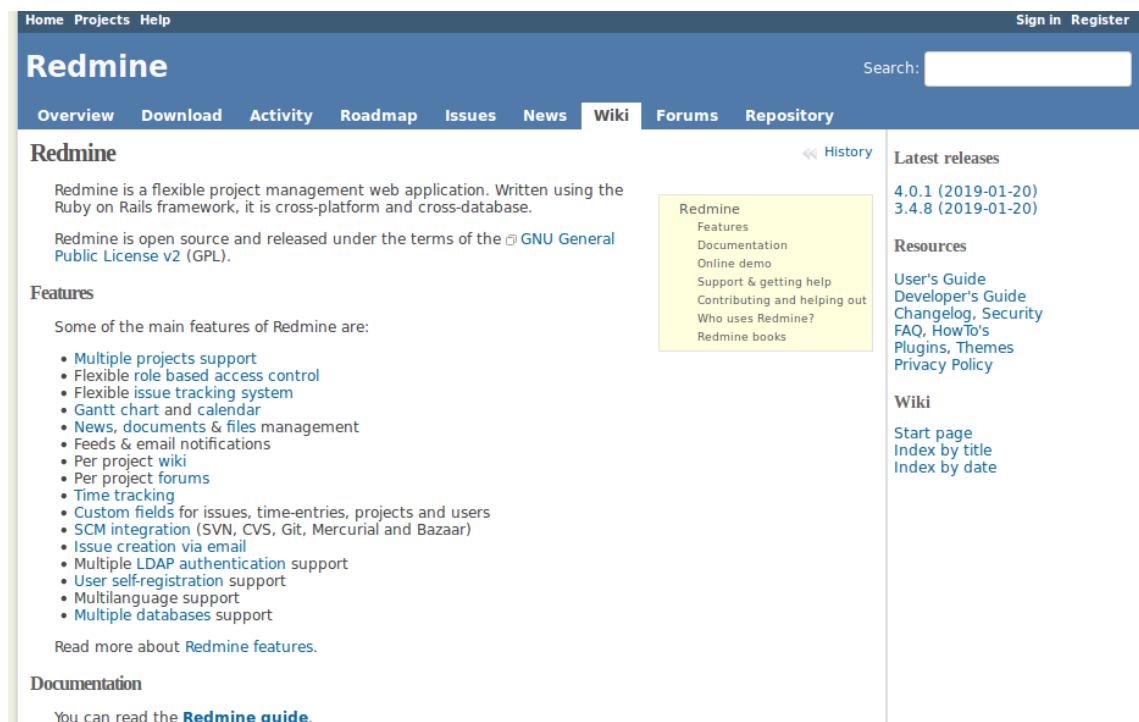


Obrázek 3.1: Jira

Redmine

Opensource alternativa vytvořena ve frameworku Rubi on Rails [4]. Oproti Jira chybí určité nástroje, ale díky své plné otevřenosti nabízí možnost jej rozšířit, nebo customizovat.

Tento projekt sám o sobě nenabízí žádnou komerční podporu, a je nutné jej spravovat individuálně. Alternativou mohou být společnosti třetích stran, které provoz Redminu poskytují jako službu, nebo jeho spinof OpenProject [4].



Obrázek 3.2: Redmine

3.3.4 Tlačítka

Toggl

Plně komerční produkt, který je nabízen zdarma pro osobní použití.

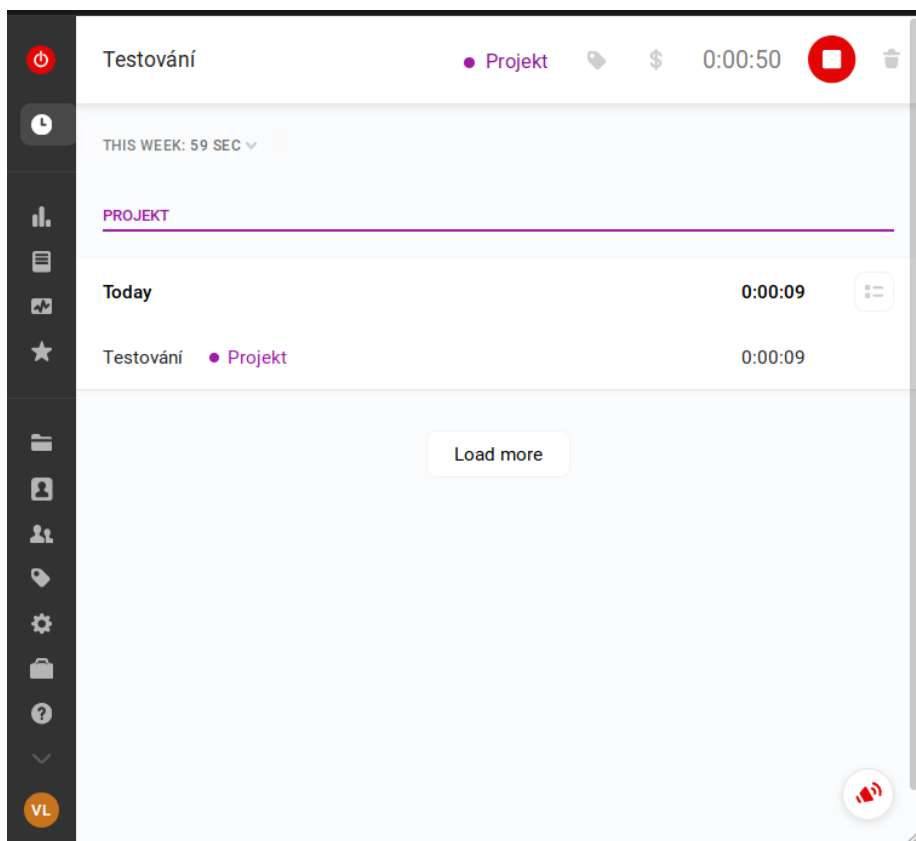
Nabízí webové rozhraní spolu s Android a IOS mobilními aplikacemi.

V případě mobilních aplikací je jejich kód uveřejněn na Githubu [6]. Kód není ale uveřejněn pod otevřenou licencí [7] a slouží tak pouze pro zvýšení průhlednosti SW a případnému zapojení komunity.

Toggl API je plně dokumentované (viz [8]).

Největší výhodou Toggl je množství integrací, které nabízí [9]. Typicky se ale jedná o nástroje, které usnadňují přidávat, přepínat trackování v Toggl. Například se jedná o přidání tlačítka do konkrétních webových stránek.

Export času z Toggl do aplikace, jako je Jira nebo Redmine, není typický. Toggl slouží spíše jako finální aplikace pro vykazování času.



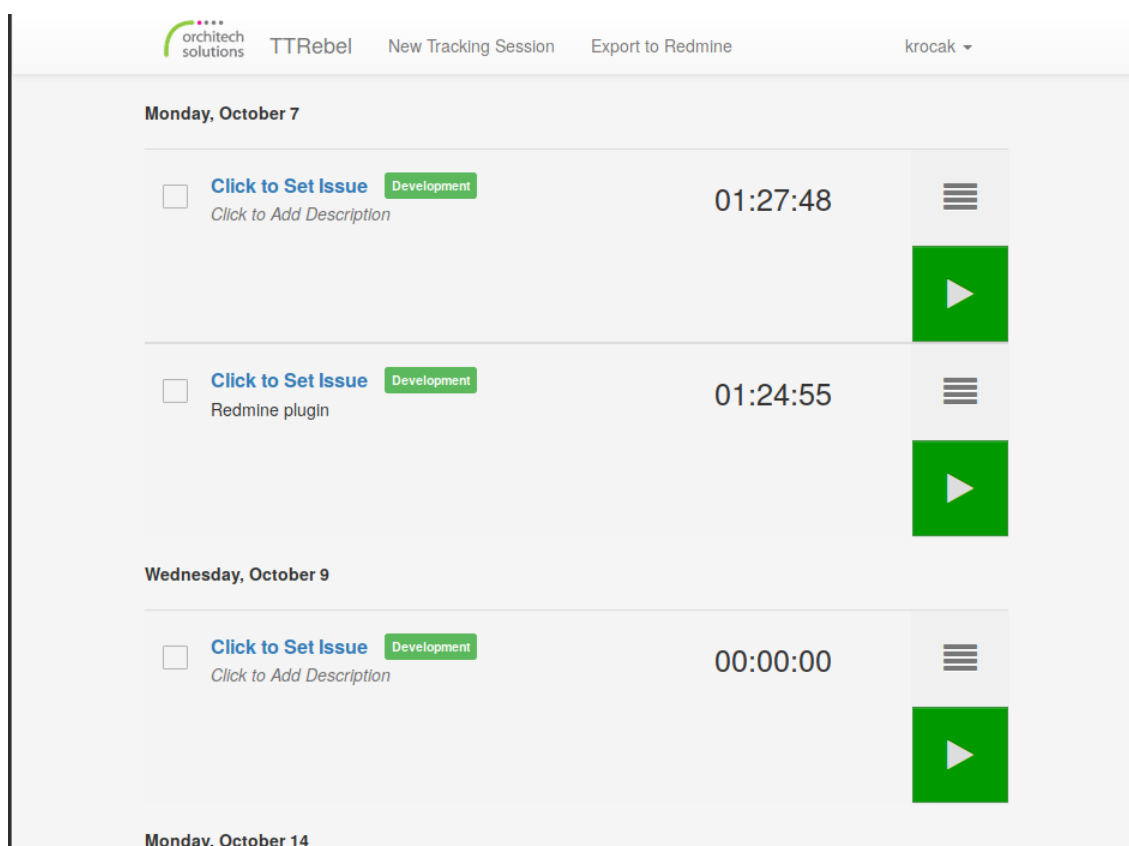
Obrázek 3.3: Toggl

TTRebel

TTRebel je privátní aplikace vyvinuta přímo Orchitech Solutions, s.r.o. Na rozdíl od Toggl slouží TTRebel přímo pro rozšíření trackovací funkcionality Redmine. Cílem tohoto projektu je nabídnout uživateli pohodlnější alternativu proti vykazování přímo u jednotlivých issue z Redminu.

TTRebel si z Redminu dotahuje názvy a čísla jednotlivých issue a projektů, které uživateli poskytuje v kontextových nabídkách. Uživatel si takto sestaví trackovací session, ke které TTRebel začne zaznamenávat jednotlivé časové intervaly, ve kterých uživatel na daném tasku pracoval.

Uživatel poté ručně spustí export dat do Redmine. TTRebel zaznamenané časové intervaly převede na čistý čas a exportuje jej do napojeného Redminu.



Obrázek 3.4: TTRebel

Orchitech Solutions, s.r.o. tento nástroj aktivně vyvíjí.

3.3.5 Trackers

ActivityWatch

Komplexní modulární tracker. Základním blokem je ActivityWatch `aw-server`, který slouží jako backend pro jednotlivé trackery. Základní ideou je, že každý uživatel bude mít k dispozici svůj `aw-server`, který bude moci využít. Veškerá trackovací data bude mít tedy pod kontrolou uživatel [10].

Ke dni 10. prosince 2019 byly úspěšně implementovány trackery pro kontrolu *X session* (`aw-watcher-afk` a `aw-watcher-window`), pluginy pro prohlížeče (Firefox a Chrome) a pluginy pro nejrozšířenější IDE (JetBrains, NetBeans, Eclipse) [11].

RescueTime

RescueTime sbírá komplexní statistiky aktivity uživatele. Cílem RescueTime je uživateli pomoci rozlišit a pomoci mu oddělit anebo zmenšit podíl prokrastinace

K druhům zaznamenaných aktivit uživatel přiřadí, zda se jednalo o rozptýlení (např: brouzdání po Facebooku), nebo se jednalo o produktivní činnost a RescueTime mu následně zobrazí procentuální výsledek (pracovní efektivnost) [16].

WakaTime

Na rozdíl od předchozích trackerů je WakaTime zaměřen čistě na sledování a počítání metrik v IDE. Měří aktivitu a to, na kterých projektech uživatel pracuje. Data jsou zaměřena na rozlišení aktivity při vývoji.

Veškerá data jsou logována na centrálním privátním serveru WakaTime [17].

3.3.6 Přepínací kostky a senzory

Timeular

Komerční produkt, osmihran sloužící pro sledování činnosti (viz podkapitola 3.1.6)) [18].

Timeular nabízí kromě vlastní aplikace pro napojení na vlastní fyzické zařízení i již připravené integrace s již existujícími a rozšířenými nástroji (např. Toggl, Jira, Google

| | App & Window Title | AFK | Browser Extensions | Editor Plugins | Extensible |
|---------------|--------------------|-----|--------------------|----------------|-----------------------|
| ActivityWatch | Ano | Ano | Ano | Ano | Ano |
| Selfspy | Ano | Ano | Ne | Ne | Ne |
| ulogme | Ano | Ano | Ne | Ne | Ne |
| RescueTime | Ano | Ano | Ano | Ne | Ne |
| WakaTime | Ne | Ano | Ano | Ano | Pouze textové editory |

Tabulka 3.1: Porovnání funkcí trackrů

Calendar). Zároveň nabízí možnost vytvářet vlastní integrace a exportovat data pomocí vlastního API [19].

Jedná se ovšem o plně komerční produkt. Export dat, přístup k API a všechny integrace jsou dostupné pouze pro platící klienty s platným měsíčním předplatným [20].

TimeFlip

Jedná se o komerční alternativu k Timeular. Na rozdíl od Timeular nabízí větší množství druhů tvarů vlastního zařízení [21].

Bussiness model je zaměřen pouze na prodej vlastních senzorů a nevyžaduje měsíční předplatné [22].

Nevýhodou TimeFlip jsou chybějící integrace. Ke dni 10. prosince 2019 nebyly vytvořeny žádné specifické integrace s nástroji, jako je Toggl, Jira atd. Zároveň chybí i připravené exporty dat do specifických formátů [23].

Veškeré integrace s tímto nástrojem jsou tedy implementovány pomocí připraveného API [24].

Fyzické senzory

Výše uvedené produkty mohou být nahrazeny libovolným jiným senzorem.

4. Případová studie Orchitech Solutions, s.r.o.

V této kapitole si popíšeme aktuální stav a používané nástroje pro evidenci a vykazování času ve společnosti. Cílem kapitoly je tento popsany stav zhodnotit a navrhnout efektivní řešení.

Součástí této kapitoly bude zároveň popsána motivace pro evidování času ve společnosti Orchitech Solutions, s.r.o. a budou zde sepsány i hlavní požadavky na finální řešení.

V závěru kapitoly bude uveden návrh finálního řešení spolu se zdůvodněním pro vybrání nástrojů.

4.1 Vykazování v Orchitech Solutions, s.r.o.

V Orchitech Solutions, s.r.o. je čas vykazován z více důvodů. Mezi tyto důvody patří jak vyúčtování odvedené práce, tak se využívá i pro udržení efektivity vlastního vedení projektů.

4.1.1 Hlavní důvody

Účetnictví

Jedním z hlavních důvodů je vedení účetnictví. Vykazování odvedené práce se využívá jak k vyúčtování klientů, tak i k ohodnocení vlastních zaměstnanců.

Pro potřeby účetnictví lze potřebu evidence dále rozdělit. V prvním případě jsou časové výkazy využity pro vyučování odvedené práce klientovi. Vlastní způsob účtování se u jednotlivých klientů často liší. Záleží, v jakém režimu s klientem společnost spolupracuje a na konkrétních smluvních podmínkách.

Orchitech Solutions, s.r.o. má zkušenost s oběma případy, kdy byly vyžadovány přesné výkazy k jednotlivým issue, tak i benevolentnější zákazníky, kteří byli účtováni pouze za celou zakázku nebo za jiný režim.

V druhém případě jsou výkazy využívány k proplácení vlastních zaměstnanců a externích dodavatelů. Nejpodrobnější výkazy Orchitech Solutions, s.r.o. vyžaduje u externích pracovníků a pracovníků na zkrácený pracovní úvazek (většinou se jedná především o studenty).

V případě pracovníků s pevnou pracovní dobou, kteří v některých případech pracují na zadané issue celou svou pracovní dobou, Orchitech Solutions, s.r.o. experimentoval i s možností vykazovat pouze celou pracovní dobu.

Vedení projektů

V základu je předem známá celková kapacita všech pracovníků. Z pohledu projekt managementu slouží tedy striktní time management především jako feedback.

Do vývojového sprintu jsou vybrány issues o celkové náročnosti odpovídající kapacitě vývojového týmu. Typicky je náročnost těchto issue odhadována právě v MH. K odhadování náročnosti feature requestů slouží kromě odhadu pracovníků i data z předchozích obdobných issue.

Při překročení odhadované doby by měly detailnější výkazy poskytnout data k identifikaci problémů. Může se jednat o další neplánované činnosti, jejich neefektivitu, tak i o potenciálně nové problémy, které nebyly v původním odhadu zohledněny.

Ve všech případech přesnější a podrobnější výkazy tuto činnost ulehčují.

Time-management

Time tracking by měl také sloužit i vlastním uživatelům. Uživatelé mohou tato data využít k zlepšení organizace vlastního pracovního času a zvýšení vlastní efektivity.

Na této úrovni jim podrobný time tracking nabízí feedback k identifikaci vlastních činností ke zlepšení své efektivity.

4.1.2 Granularita vykazování

Výše uvedené důvody pro time tracking se liší především nároky na potřebnou podrobnost sbíraných dat.

Pro potřeby účetnictví typicky Orchitech Solutions, s.r.o. nepotřebuje zcela detailní rozpis všech činností, ale postačí alespoň jejich souhrn.

Pro vedení projektů a time-management jsou detailnější data výhodou. Zvýšená potřeba na množství a detailnost dat ovšem klade na uživatele zvýšené nároky.

4.2 Nejčastější prohřešky

4.2.1 Nedostatečná evidence

V případě, že si pracovník v průběhu pracovní doby neeviduje své činnosti, je téměř nemožné zpětně dopočítat, kterým činností se věnoval. Pouhý odhad v tomto směru není dostatečný a pracovníci mívají typicky tendenci strávený čas podhodnocovat.

Chybějící čas nemůže být vyúčtován ani klientovi, ani zaměstnanci.

4.2.2 Podhodnocování výkazů

Laxní evidence činnosti a nutnost evidovat odvedenou práci může svádět pracovníky k snížení evidovaného času.

Vedou je k tomu především situace, kdy přepínají mezi více činnostmi, nebo se nemohou věnovat jedné činnosti naplno. Alternativně se domnívají, že v činnosti nebyli efektivní nebo čas strávili jinou nesouvisející činností. Při absenci dat si typicky nejsou schopni uběhlý čas obhájit a raději zaevidují kratší dobu.

Z pohledu projekt managementu se jedná o zcela nevhodné chování a nevhodné řešení problému. Toto chování zkresluje sledování stavu projektu a brání jejich potenciálnímu řešení.

Podobné situace by měly být řešeny interně v rámci týmu i za cenu vykázaní vyšších hodnot.

4.2.3 Práce o víkendu

Nedostatečná evidence, nebo podhodnocené výkazy mohou budít v pracovnících pocit, že se práci nevěnují a měly by se ji snažit dohnat. Spolu s možností práce z domova toto lze. Pracovníci se tedy občas snaží prioritní úkoly dohánět i v rámci svého volného času.

Pracovníci tím pouze odsouvají původní problém a zkreslují stav projektu. Zároveň tím pracovníci zasahují do svého času vymezeného k osobním činnostem a odpočinku.

4.3 Používané nástroje

V Orchitech Solutions, s.r.o. je projekt management řízen pomocí Redmine, který zároveň slouží k evidenci času stráveného u jednotlivých issue.

Redmine je využíván dlouhodobě a ve většině projektů. Postupně byl nastaven a upraven pro potřeby Orchitech Solutions, s.r.o. K zjednodušení samotného vykazování Orchitech Solutions, s.r.o. začal vyvíjet vlastní nástroj TTRebel z důvodu, že jiný time tracker tracker pro Orchitech Solutions, s.r.o. nevyhovoval. Použití TTRebelu není povinné a slouží pouze jako doplňkový nástroj k evidování času do Redmine.

Původní verzi TTRebelu používala pouze část zaměstnanců především z důvodu ne zcela pohodlného UX a UI. Zbytek zaměstnanců vykazuje čas přímo do Redmine.

Orchitech Solutions, s.r.o. má zájem TTRebel dále vyvíjet a upravovat, jak pro své potřeby, tak i pro potřeby komunity.

4.4 Způsob vykazování

Typický příklad vykazování je následující.

Pracovníkovi je přidělena issue. Pracovník si zaznamená, kdy na issue začal pracovat a po jejím dokončení nebo na konci pracovní doby zaznamená odpracovaný čas k issue do Redmine.

4.5 Požadavky

4.5.1 Míra monitorování

Účelem není a nemělo by být uživatele přímo monitorovat či kontrolovat. Účelem je uživateli pomoci s vykazováním a nabídnout mu možnost, jak ze sebraných dat doplnit chybějící výkazy a ulehčit mu tak jeho práci.

Veškerá sebraná data by měl mít uživatel pod kontrolou a zaměstnavatel by neměl mít právo tato data využít přímo proti zaměstnanci.

4.5.2 Jednoduchost

Vykazování není hlavní činností zaměstnanců. Využívání nástrojů by tedy nemělo příliš zaměstnance zatěžovat, nebo jim příliš komplikovat jejich běžné činnosti

4.5.3 Automatizace

Cílem je ulehčit práci všem uživatelům. Finální řešení by tedy mělo poskytovat co největší míru automatizace.

4.5.4 Multiplatformnost

Pracovníci v Orchitech Solutions, s.r.o. pracují na různých platformách (Linux, Windows, iOS). Navržené řešení by mělo fungovat pro všechny tyto platformy.

4.6 Závěr

Z důvodů preferencí stanovených Orchitech Solutions, s.r.o. bude navržené řešení vycházet z použití Redmine, TTRebel a ActivityWatch.

Redmine bude použit z důvodu množství modifikací, které Orchitech Solutions, s.r.o. od nástroje pro správu projektů požaduje. Orchitech Solutions, s.r.o. zároveň do budoucna předpokládá možné investice do dalších úprav a rozvoje Redmine.

TTRebel bude využit z obdobných důvodů. Orchitech Solutions, s.r.o. jej má v úmyslu dále rozvíjet. Zároveň, že je implementován primárně pro Orchitech, může být tato aplikace upravena na míru potřebám stávajících zaměstnanců společnosti Orchitech Solutions, s.r.o.

ActivityWatch byl vybrán pro svou snadnou rozšiřitelnost a aktivnost projektu i přes některé nedostatky, které tento projekt aktuálně má.

Zároveň všechny tyto tři nástroje splňují požadavek na multiplatformní řešení. Buď se jedná o webové aplikace, nebo poskytují verze pro všechny požadované platformy.

5. Návrh vlastního řešení

V této kapitole se budeme zabývat návrhem prototypu pro zlepšení vykazování a evidenci času pro společnost Orchitech Solutions, s.r.o. V první části této kapitoly budou stanovena konkrétní kritéria a požadavky vycházející na základě průzkumu a kritérií zadaných vedoucími pracovníky v Orchitech Solutions, s.r.o. (viz kapitola 4).

Na základě těchto kritérií budou vybrány nástroje, které budou použity pro samotnou implementaci (plný popis těchto nástrojů je uveden v kapitole 3).

V závěru kapitoly bude z těchto nástrojů sestaveno konkrétní řešení. Zároveň budou v závěru kapitoly určena kritéria pro splnění požadavků stanovených v kapitole 4.

5.1 Vybraná kritéria

Orchitech Solutions, s.r.o. jakožto SW. firma je ochotná dále vyvíjet a udržovat své vlastní řešení (řešení implementované v rámci praktické části této diplomové práce).

Zároveň preferuje pokračovat ve využívání některých svých nástrojů. Toto se odráží i na finálních kritériích.

Rozšiřitelnost Orchitech Solutions, s.r.o. má zájem si řešení upravovat. Zároveň požaduje možnost přidávat nové zdroje dat a možnost dále provázat používané nástroje se svými dalšími systémy.

API Souvisí přímo s rozšiřitelností. Jedná se o kritický prvek umožňující navazovat nové prvky. Požadavkem je tedy kvalitně popsání a dostupné API. Mimo jiné toto API by mělo být možné i testovat.

Aktivnost projektu Mnohé (nejenom) open-source projekty. Pro praktickou část budou preferovány mladší a viditelně aktivní projekty.

Cena Při sestavování bude preferována nižší cena v dlouhodobém úseku. Případně budou upřednostňovány open-source projekty. Toto vyvažuje ochota udržovat a vyvíjet vlastní řešení.

5.2 Vybrané prvky

Po dokončení případové studie byli vybrány primárně následující nástroje, které budou využity k sestavení finálního řešení.

Redmine Bude nadále sloužit jako hlavní backend pro vykazování.

TTRebel Hlavní prvek, který bude sloužit pro zjednodušení vykazování.

ActivityWatch Pro sběr statistických dat pro zaměstnance.

5.3 Sestavení řešení

5.3.1 Redmine

Redmine bude nadále využíván pro spravování projektů a jako cílový systém, do kterého ho budou zapisovány výsledné výkazy. Toto ovlivňuje především formát a informace, které je nutné určovat.

Číslo issue Identifikátor ticketu, ke kterému budou přidávány jednotlivé výkazy.

Aktivita Kategorie činnosti, kterou zaměstnanec prováděl (vývoj, meeting aj.). Jedná se v podstatě o fixní seznam předdefinovaných činností. Seznam aktivit je předem definován, a není v praxi příliš často měněn. Navržené řešení by mělo, ale počítat, že seznam aktivit může být změněn. Zároveň je nutné podotknout, že ne všechny aktivity dávají jsou využity všemi uživateli.

Popis činnosti Vlastní konkrétní popis činnosti.

Čas Čas strávený na dané činnosti.

Tyto čtyři atributy je nutné vyplnit vždy. Kromě těchto atributů lze do Redmine přidat další nepovinná pole. Tyto atributy jsou v Orchitech Solutions, s.r.o. typicky vyplňovány pouze ve výjimečných případech.

Chování Redminu samo o sobě v tomto směru nevyžaduje další úpravy. Redmine bude pouze případně rozšířen pomocí pluginů, pro snazší použití spolu s TTRebelem a ActivityWatchem.

5.3.2 TTRebel

TTRebel bude nadále upravován a rozšiřován o novou funkcionalitu. V průběhu diplomové práce bude rozšířen o nové prvky a bude zajištěna lepší integrace s vlastním Redmine.

Mezi prvky a feature requesty implementované v průběhu psaní této diplomové práce nebo implementované přímo v rámci této diplomové práce budou zařazeny následující prvky:

Tlačítko v Redmine Po vzoru Toggl bude do Redmine issue přidáno tlačítko. Toto tlačítko umožní uživateli automaticky zakládat a ovládat trackovací session v TTRebelu.

Rozšířit API API bude postupně vyvíjeno spolu s rozšiřující se funkcionalitou přidaného tlačítka. Zároveň bude muset být rozšířeno pro napojení s ActivityWatch.

Kromě výše uvedených prvků bude implementováno nové GUI.

5.3.3 ActivityWatch

ActivityWatch zde bude působit jako hlavní prvek, pro individuální zaměstnance. Bude sloužit jako agregátor všech dat z ostatních systémů. Zároveň poskytuje nástroje (`aw-watcher-window` a `aw-watcher-afk`), které mohou být použity bez nutnosti dalších úprav. Zpráva tohoto nástroje bude zcela individuální a bude pod plnou kontrolou individuálních zaměstnanců.

Bude implementováno napojení na TTRebel, jehož data budou doplněna do ActivityWatch pro dodání kontextu. Tím bude usnadněna uživatelům zpětná úprava dat v TTRebelu.

V prvním kroku bude implementováno pouze doplňování dat z TTRebel do ActivityWatch. Úprava ActivityWatch pro úpravu časových údajů nebude implementována v rámci této diplomové práce. To je především omezeno již existujícími feature requesty založenými přímo pro ActivityWatch (viz „*Implement tagging/annotation #95*“ [25]). Vytvoření zpětné úpravy a update dat z ActivityWatch do TTRebela může být imple-

mentováno až s návazností na implementaci těchto prvků (mimo rozsah této diplomové práce).

Doplňování dat z TTRebel do ActivityWatch má za cíl zajistit konzistenci dat a jednodušší vytváření prototypů ActivityWatch query. Query budou moct být testovány a spouštěny samostatně nad ActivityWatch. Pro praktické použití ovšem není doplňování dat zcela potřebné. Data z TTRebela budou použita především pro vymezení času.

Kromě nástrojů pro synchronizaci dat budou připraveny návrhy na ActivityWatch query.

Předběžný návrh query

ActivityWatch API kromě endpointů pro manipulaci s daty nabízí i endpoint `/0/query/`.

¹ Tento endpoint nabízí možnost pomocí omezeného skriptu data předzpracovat přímo na straně serveru. Data je mimo jiné možno filtrovat, agregovat nebo do určité míry naformátovat do potřebné výstupní podoby [15]. Toto umožňuje využít ActivityWatch přímo pro získání odvození informací bez nutnosti data zpracovávat zcela samostatně.

Navržené query by především měly představovat šablony, které by měly být dále rozšiřovány a individualizovány. Pro jejich úpravu může být navržen vhodný nástroj.

Pro potřeby prototypu musí navržená query doplnit chybějící informace. ²

Čas

Jedním z hlavních cílů ActivityWatch je poskytovat statistické údaje (kolik času uživatel strávil nad danou činností). Využití ActivityWatch k stanovení stráveného času nad danou činností vyžaduje projít všechny typy (události lze pro zjednodušení sloučit) událostí. Prohledávání dat musí být nějak časově vymezeno. Z těchto důvodů vytvoření query vygenerování stráveného času je samo o sobě velmi časově náročné. Query pro odhad stráveného času bude nahrazena TTRebelem.

¹Kromně `/0/query/` nabízí ActivityWatch plné REST API pro práci s ukládanými daty (procházení, ukládání, mazání dat, ap.).

²Typickým výsledkem query je buď seznam profiltrovaných eventů, jejich dat, nebo suma jejich trvání. Data budou muset tedy být dále zpracována.

Aktivita

Nejjednodušeji query pro konstrukci. Seznam možných vykazatelných aktivit je předem znám. Uložené eventy tedy lze jednoduše kategorizovat.³

Issue

Query pro identifikování konkrétní issue (nebo jejího id) by vyžadovala opětovně specifikovat query pro každou issue zvlášť. Takovéto query v rámci projektu mohou být prakticky totožné.

Mnohem jednodušším řešením je z ActivityWatch se snažit získat konkrétní issue id (typicky vždy začíná znakem #) nebo se pokusit odhadnout konkrétní projekt a na základě těchto informací filtrovat potenciální kandidáty.

Popis

Generovat přesný popis pomocí dat z query není zcela možné. Popis musí buď doplnit uživatel, nebo bychom jej museli odvozovat od vybrané issue a aktivity.

5.3.4 Další prvky - prostor pro individualizaci

Další prvky budou připraveny nebo doporučeny dle individuálních potřeb zaměstnanců.

³ Od verze v0.8.1 nabízí ActivityWatch možnost jednotlivé eventy v query jednoduše kategorizovat pomocí funkce `categorize()`. Tato funkce ovšem přidělí každému eventu efektivně jedinou kategorii. Kategorie mohou být zanořeny, ale nemohou se překrývat (ActivityWatch přiřadí konkrétní kategorii a všechny jí nadřazené kategorie). Tato funkce by mohla do určité míry pokrýt potřeby pro konstrukci výše uvedených query, ovšem nenabízí tolik prostoru pro modifikace. z tohoto důvodu nebude použita v prototypu.

6. Implementace řešení

Tato kapitola popisuje vlastní implementaci prototypu na základě stanovených požadavků a předchozí analýzy procesů ve společnosti Orchitech Solutions, s.r.o. Jsou zde implementovány úpravy stávajících aplikací, popis nově implementovaných částí a finální popis sestaveného řešení (prototypu). Návrh prototypu je sestaven z Redmine, TTRebel a ActivityWatch.

6.1 Redmine

Veškeré úpravy byly do Redmine zaváděny pouze jako nové pluginy, které budou rozepsány v dalších podkapitolách (viz 6.2.4). V rámci této diplomové práce nebylo nutné zavádět žádné úpravy do samotného Redmine.

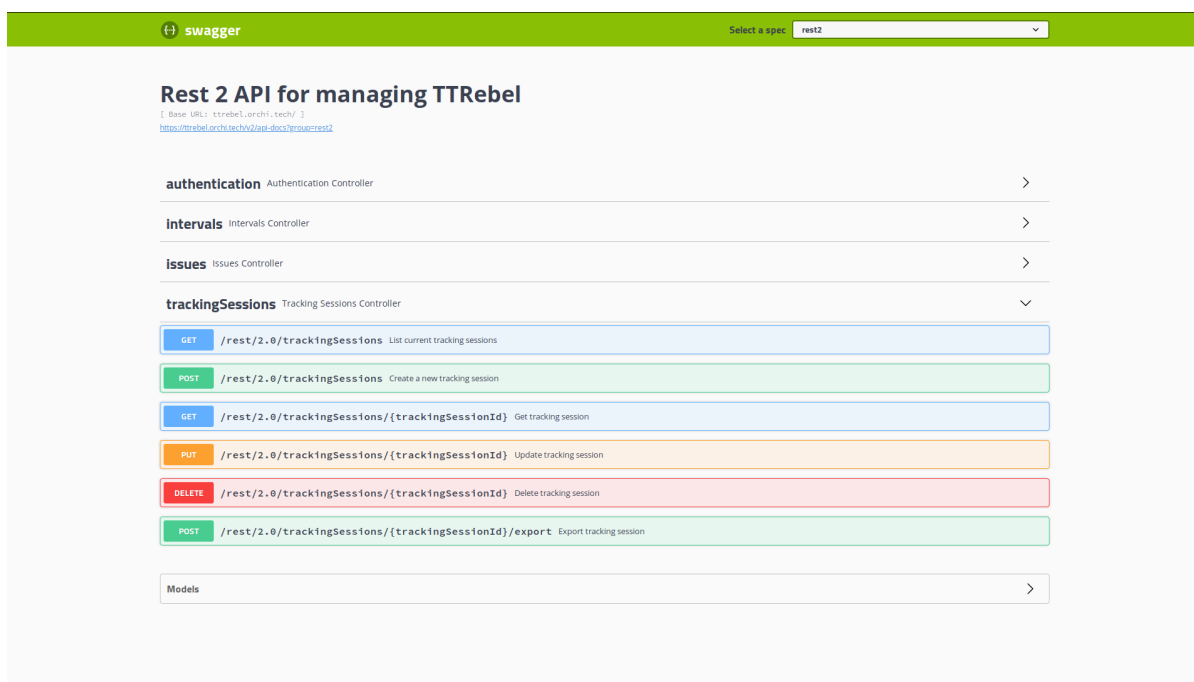
6.2 TTRebel

V průběhu roku 2019 byl TTRebel aktivně vyvíjen Orchitech Solutions, s.r.o. Autor této diplomové práce se na tomto vývoji aktivně podílel. ¹

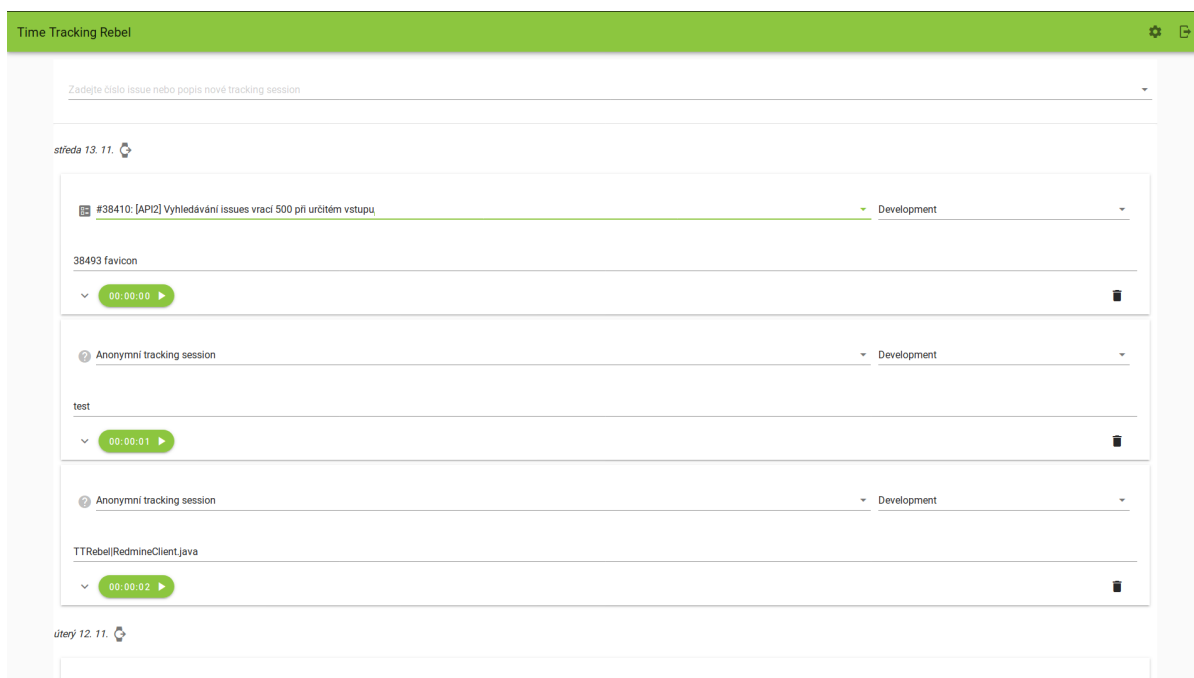
Průběžně bylo zavedeno nové API a GUI. Zároveň byly v TTRebel postupně zaváděny další úpravy. Původní funkcionality byla zatím zachována, ale je považována za deprecated.

6.2.1 API

Nové API bylo navrženo jako jednoduché restové rozhraní („Rest v2“). Toto API je postupně rozšiřováno o další nové funkce. Původní API bude odstraněno až s odstraněním původního GUI.



Obrázek 6.1: Vypsání TTRebel API.



Obrázek 6.2: Ukázka TTRebel GUI v 2.

6.2.2 GUI

Nové GUI je postupně implementováno ve frameworku Vue (Vuetify). Autor této diplomové práce se na vývoji GUI pouze podílel jen jako konzultant (code review, návrh, administrativa aj.).

6.2.3 Datový model

`TrackingSession` Představuje objekt, který je následně exportován jako záznam do napojeného Redmine. Dále jen `session`.

`TrackingInterval` Představuje jednotlivé časové intervaly, ve kterých uživatel pracoval na dané `session`.

Datový model TTRebelů samozřejmě obsahuje další entity, jako jsou například entity uživatele, systému (údaje o napojeném Redmine), atd. Pro potřeby této diplomové práce stačí takto zjednodušený model.

V průběhu roku 2019 byl tento model (a hlavně provázání s dalšími entitami) postupně zjednodušován.

6.2.4 TTRebel tlačítko v Redminu

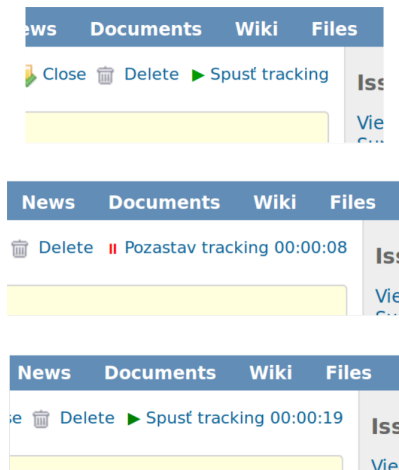
Jako jeden z prvních požadavků, které autor řešil, byla možnost založit tracking session přímo v Redminu, bez nutnosti ji pracně zakládat a definovat v TTRebelů.

V rámci vývoje tohoto pluginu se autor zároveň seznámil s vnitřním fungováním Redmine, jak vytvářet pluginy pro Redmine a s již aktuálně existujícími pluginy [26].

Jako inspirace posloužil obdobný plugin pro Toggle. Tento plugin posloužil pouze jako hrubý nástin možností, které by uživatelé mohli požadovat. Tyto požadavky byly následně zadávány a budou implementovány postupně do vytvořeného prototypu.

Prototyp tlačítka a jeho konfigurace byla vytvořena jako plugin do Redmine. Zároveň tak byla usnadněna jeho instalace.

¹TTRebel je implementován v frameworku Java Spring. Pojmenování entit a prvků tedy vychází z jmenné konvence běžně používané v jazyce Java.



Obrázek 6.3: Prvotní design TTRebel tlačítka v Redminu

Samotný prototyp byl implementován v JS a jQuery (běžně dostupná součást Redmine UI).

Tento prototyp zahrnuje pouze základní funkcionalitu, která umožňuje uživateli založit, spustit nebo přerušit trackování. Prototyp vyžaduje předchozí přihlášení do TTRebel, do kterého plugin přímo provádí veškeré změny pomocí CORS [27].

Největší slabinou tohoto prototypu byla neschopnost vyřešit některé nekonzistence nebo errorry. Například v případě, že uživatel trackování issue přerušil přímo v TTRebel, nebo v případě, že uživatel nebyl přihlášen v TTRebel.

Podměty na další funkcionalitu

- Přesměrovat nepřihlášeného uživatele do TTRebel, nebo zobrazit iframe přihlašovací pop-up.
- Detekovat změnu stavu session v TTRebel.
- Umožnit uživateli editovat popis činnosti z Redmine.
- Vyřešit existenci více tracking session pro jednu issue.

6.3 IoT

V podkapitole 3.3.6 byl uveden seznam IoT produktů určených pro time-tracking. Výběr vhodného zařízení z tohoto seznamu silně závisí na preferencích konkrétního uživatele. Orchitech Solutions, s.r.o. má pouze omezený počet zaměstnanců a nasazení jakéhokoliv zařízení by bylo vždy velmi individuální.

Reálně pro potřeby vyžádaného prototypu mohou tato zařízení poskytovat pouze údaje o aktivitě (kategorii činnosti) uživatele nebo fyzickou lokaci uživatele (např. kancelář v zasedací místnost ap.). Aktivita uživatele může být určena z dalších údajů (viz 6.5). Základní údaje o lokalitě mohou být do určité míry také odvozeny, ale představují hlavní benefit použití jakéhokoliv senzoru nebo jiného zařízení.

Pro potřeby prototypu vyvíjeného v rámci této diplomové práce jsou veškeré informace z těchto zařízení nutné importovat do ActivityWatch. Data mohou být použita jak v podobně generovaného pomocí konkrétního zařízení, tak i v interpretované podobě (data jsou předem zpracována a do ActivityWatch jsou uloženy konkrétní výsledky).

6.4 ActivityWatch

V průběhu psaní této diplomové práce byl ActivityWatch nezávisle aktivně vyvíjen a došlo postupně k vydání verze až verze v0.8.3 (vydáno 13. listopadu 2019 viz [13]). Autor této diplomové práce se na tomto vývoji nepodílel.

Mezi nově přidanou funkcionalitu ActivityWatch patří rozšíření query o možnost jednotlivé eventy rozdělit do kategorií (slouží k vytvoření statistiky) a tagování eventů. Dále jsou postupně implementovány kroky k umožnění synchronizace ActivityWatch.

6.4.1 Import dat z TTRebel

V rámci vývoje prototypu byly připraveny dva moduly do ActivityWatch `aw-watcher-ttrebel` a `aw-importer-ttrebel`.² Import dat do ActivityWatch byl implementován jako jedno-

²ActivityWatch je v současné době implementován v jazyce Python. Pojmenování modulů a proměnných vychází z jmenné konvence jazyka Python a z konvencí ActivityWatch. PZN: Veškeré moduly ActivityWatch začínají prefixem `aw-`.

dušší, aby byla poskytnuta možnost pracovat nad všemi potřebnými daty čistě v rámci ActivityWatch.

`aw-watcher-ttrebel`, jako ostatní ActivityWatche watchery, periodicky kontroluje a ukládá TTRebel aktuální spouštěnou session. Tento modul byl implementován hlavně jako *proof of concept* a jako jednodušší varianta k `aw-importer-ttrebel`. Tento testovací modul neposkytuje žádnou zpětnou synchronizaci dat. Toto umožnilo autorovi se primárně blíže seznámit s vlastní implementací a chováním ActivityWatch.

Ačkoliv `aw-watcher-ttrebel` byl míněn hlavně jako *proof of concept* do určité míry poskytuje určitou formu logování změn v TTRebel (pouze aktivní session). Tato funkcionalita je spíše bonusem.

`aw-importer-ttrebel` byl vytvořen jako komplikovanější varianta `aw-watcher-ttrebel`. Tento modul automaticky synchronizuje všechny známé informace z TTRebel do ActivityWatch. Zároveň tyto informace dokáže do určité míry updatovat. ActivityWatch sám o sobě nenabízí endpoint pro update jednotlivých eventů. Toto může způsobovat určité inkonzistence v průběhu synchronizace (původní uložené eventy musí být vždy smazány). Tyto inkonzistence pro potřebu prototypu nepředstavují výrazný problém.

Data

Oba dva moduly sdílejí nástroje pro komunikaci s TTRebel a zpracování dat. Data mohou být jednoduše rozšířena o další atributy entit z TTRebel. Data ukládaná v ActivityWatch jsou tedy popsána v tabulce 6.1.

Data mohou být následně využita přímo v query pro fitování eventů (viz 6.1). Pouze eventy, které byli zaznamenány ve stejný čas jako `TrackingsSession „X“`, budou využity. Konkrétní sestavení query pro konkrétní účely je zmíněno v podkapitole 6.5.

| Název proměnné | Popis |
|-----------------------------------|---|
| <code>tracking_session_id</code> | Id TTRebel session. |
| <code>issue_id</code> | Id vybrané issue v Redmine. |
| <code>system_id</code> | Id napojeného Redmine v TTRebel. Pzn. TTRebel umožňuje napojit více různých instancí Redmine. Do budoucna se počítá i s napojením na jiný typ projektového nástroje (např. Jira). |
| <code>issue_name</code> | Název issue z Redmine. |
| <code>description</code> | Popis činnosti zaměstnance uvedený v TTRebel session. |
| <code>tracking_interval_id</code> | Id konkrétního TTRebel intervalu, ve kterém zaměstnanec pracoval na dané session. |
| <code>title</code> | Formátovaný název issue s id issue (pzn. ActivityWatch automaticky zobrazuje atribut <code>title</code> ve webovém GUI). |

Tabulka 6.1: Základní proměnné TTRebel session.

Ukázka 6.1: Query filtrující data pro specifickou TrackingSession.

```

ttrebel_events = query_bucket(find_bucket('aw-watcher-ttrebel_localhost'));
window_events = query_bucket(find_bucket('aw-watcher-window_localhost'));

filtered_window_events = filter_period_intersect(window_events,
    filter_keyvals(ttrebel_events, 'tracking_session_id', ['X']));

```

| Název modulu | Popis |
|-----------------------------------|--|
| <code>aw-watcher-afk</code> | Modul detekující sledující aktivitu myši a klávesnice. Jedná se o původní ActivityWatch modul [14]. |
| <code>aw-watcher-window</code> | Modul detekující aktivního okna a jeho titulku. Jedná se o původní ActivityWatch modul [14]. |
| <code>aw-inporter-ttrebel</code> | Vlastní modul, implementovaný autorem této diplomové práce, který synchronizuje data z TTRebel do ActivityWatch. |
| <code>aw-watcher-in-office</code> | Vlastní modul, implementovaný autorem této diplomové práce, který indikuje přítomnost zaměstnance v kanceláři. |

Tabulka 6.2: Použité ActivityWatch moduly.

Ukázka 6.2: Proměnné použité v query.

```
afk_events = query_bucket(find_bucket('aw-watcher-afk_localhost'));
window_events = query_bucket(find_bucket('aw-watcher-window_localhost'));
window_events_not_afk = filter_period_intersect(window_events,
    filter_keyvals(afk_events, 'status', ['not-afk']));

in_office = query_bucket(find_bucket('aw-in-office_localhost'));
window_events_not_afk_in_office =
    filter_period_intersect(window_events_not_afk, filter_keyvals(in_office,
        'status', ['in-office']));

window_events_in_office = filter_period_intersect(window_events_not_afk,
    filter_keyvals(in_office, 'status', ['in-office']));
```

6.5 ActivityWatch Query

Použité query využívají (filtrují) data z následujících modulů popsaných v tabulce 6.2.

Informace poskytované `aw-watcher-window` mohou být doplněny nebo nahrazeny informacemi z modulů sledující konkrétní aplikace jako je webový prohlížeč nebo IDE (např. `aw-watcher-web` a `aw-watcher-jetbrains`).

Data z těchto jsou přiřazena v query do proměnných (viz ukázka 6.2), které usnadní zápis dalších ukázek.³

Seznamy mohou být dále profiltrovány na základě dat z TTRabela, nebo mohou být časové intervaly specifikovány přímo jako součást query.⁴

Níže uvedené ukázky mohou být nadále upravovány dle preferencí jednotlivých uživatelů.

³Ukázky z ActivityWatch query by mohly být psány efektivněji za cenu jejich čitelnosti. Pro potřebu textové části této diplomové práce byla zvolena tato čitelnější verze.

⁴ query a timeperiods jsou součástí requestu odesílaného na ActivityWatch /0/query/.

Ukázka 6.3: Query filtrující eventy zařaditelné do předprodeje.

```
presale = filter_keyvals_regex(window_events_not_afk, 'title',  
    'Lucidchart|Redmine|Jira|Google\sDocuments|Google\\sDocumenty');
```

Ukázka 6.4: Query filtrující eventy zařaditelné do analýzy.

```
analysis_design = filter_keyvals_regex(window_events_not_afk_in_office,  
    'title', 'Lucidchart|Redmine|Jira|Google\sDocuments|Google\\sDocumenty');
```

6.5.1 Aktivita

Následující ukázky (6.3 - 6.8) rozřazují eventy dle jednotlivých aktivit. Regulární výrazy použité pro v tyto query se úmyslně překrývají. Některé nástroje, které zaměstnanci používají při více činnostech. Jediným řešením by bylo výraz dále konkretizovat (např. kromě použitého programu specifikovat jméno otevřeného souboru ap.).

Předprodej

V ukázce 6.3 jsou data pouze filtrována dle použité aplikace. Předpokládá se použití především komunikačního nástroje. Nepředpokládá se, že se uživatel nachází v kanceláři (může se nacházet na schůzce se zákazníkem).

Analýza & Design

V ukázce 6.4 se předpokládá použití nástrojů, používaných sestavováním návrhů. Předpokládá se, že zaměstnanec tráví čas především v kanceláři.

Konzultace

Pro určení konzultací (ukázka 6.5) jsou použity pouze komunikační nástroje. Data nejsou filtrována dle stavu AFK. Především u videokonferencí nebo telefonátů uživatel není typicky aktivní, a tudíž data z modulu `aw-watcher-afk` obsahují chybná data.

Ukázka 6.5: Query filtrující eventy zařaditelné jako konzultace s kolegy nebo zákazníkem.

```
consulation = filter_keyvals_regex(window_events, 'title', 'Meet|Slack');
```

Ukázka 6.6: Query filtrující eventy identifikovatelné jako vývoj.

```
development = filter_keyvals_regex(window_events_not_afk_in_office, 'title',  
    'Eclipse|Intellij\sIDEA|PhpStorm|Gitlab|Redmine|Jira|Stack\sOverflow');
```

Vývoj

V 6.6 jsou data filtrována dle IDE, ale i dle nejčastěji navštěvovaných stránek, jako je např. Gitlab nebo Stack Overflow. Předpokládá se, že zaměstnanec se nachází v kanceláři, ale tato informace se dá nahradit např. informací, zda uživatel měl spuštěnou VPN (míněno pro tzv. home office).

Management

Query definující manažerskou činnost (viz ukázka 6.7) opětovně předpokládají použití spíše komunikačních a administrativních nástrojů. Na rozdíl např. od předprodeje (viz ukázka 6.3) předpokládá se, že zaměstnanec se nachází v kanceláři a opětovně se stejně jako v případě konzultací (viz ukázka 6.5).

Code review

Code review lze opětovně určit pouze jako kombinaci vybraných stránek (viz ukázka 6.8).

Výsledek těchto query (6.3 - 6.8) následně převedeme na strávený čas a naformátujeme jakožto (viz 6.9). Výsledkem je ovšem asociativní pole (kategorie - strávený čas v sekundách). Z tohoto výsledku vybereme kategorii s nejvyšší hodnotou (časem). Tento krok musí být proveden mimo ActivityWatch (query v aktuální verzi nenabízí logické operace).

Ukázka 6.7: Query filtrující eventy identifikovatelné jako management.

```
management = filter_keyvals_regex(window_events_in_office, 'title',  
    'Redmine|Jira|LibreOffice|Google\sDocuments|Google\sDocumenty|Slack|Meet');
```

Ukázka 6.8: Proměnné použité v query.

```
code_review = filter_keyvals_regex(window_events_not_afk_in_office, 'title',  
    'Redmine|Gitlab|Slack');
```

Ukázka 6.9: Naformátování výstupu query kategorizující aktivitu uživatele.

```
RETURN = {  
    'presale': sum_durations(flood(presale)),  
    'analysis_design': sum_durations(flood(analysis_design)),  
    'consulation': sum_durations(flood(consulation)),  
    'development': sum_durations(flood(development)),  
    'management': sum_durations(flood(management)),  
    'code_review': sum_durations(flood(code_review))  
};
```

6.5.2 Issue

Issue může být odvozena z prohlížených issue (viz 6.10). Z výsledků této query můžeme následně přechít id vhodných issue. Tyto hodnoty, které nabídneme uživateli z kontextové nabídky.

Query 6.10 může ovšem produkovat i id z jiných stránek. O něco vhodnějším řešením je uživateli nainstalovat do prohlížeče plugin, odpovědný za logování do ActivityWatch historie přímo. Toto umožní filtrovat data i dle url (viz 6.11).

Funkčnost tohoto řešení spoléhá především na to, že uživatel pravidelně navštěvuje webovou stránku s danou issue nebo merge requestem.

Ukázka 6.10: Query filtrující potenciální issue id.

```
window_events = query_bucket(find_bucket('aw-watcher-window_localhost'));  
window_events = filter_keyvals_regex(window_events, 'title', '#\d+.*Redmine');
```



```
RETURN = merge_events_by_keys(window_events, ['app', 'title']);
```

Ukázka 6.11: Query filtrující potenciální issue id.

```
browser_events = query_bucket(find_bucket('aw-watcher-web-firefox'));  
browser_events = filter_keyvals_regex(window_events, 'url',  
    'redmine.orchitech.cz|gitlab.orchitech.cz');  
browser_events = filter_keyvals_regex(window_events, 'title', '#\d+');  
  
RETURN = merge_events_by_keys(window_events, ['app', 'title']);
```

6.6 Sestavení finálního řešení

Jak již bylo zmíněno, finální řešení je sestaveno z Redmine, TTRebel a ActivityWatch. Další vývoj tedy bude směřován na provázání TTRebel a ActivityWatch. Do TTRebel bude zavedeno napojení na lokální ActivityWatch.

Bude vyžadováno vytvoření uživatelské nastavení, výsledky query použít pro doplňování nebo našeptávání inputu uživatele.

7. Závěr

Tato diplomová práce se zabývala problematikou sledování a automatizováním vykazování pracovníků v oblasti IT. Hlavním cílem této práce bylo sepsat aktuálně dostupné nástroje, způsob vykazování a evidenci času běžně využívané v IT firmách a na základě těchto údajů navrhnout a sestavit prototyp řešení pro vybranou firmu.

V rámci této diplomové práce byla provedena rešerše používaných nástrojů a alternativ. Z těchto nástrojů byl sestaven návrh implementace řešení pro společnost Orchitech Solutions, s.r.o. Zároveň byla navržena analytika, která umožňuje zaměstnanci do určité míry doplnit výkazy.

Tento systém byl navržen na základě případové studie zadané zadavatelem.

Byly provedeny veškeré požadované kroky k naplnění cílů této diplomové práce. Bohužel z důvodu dlouhého vývojového cyklu a návazných projektů nebyl navržený prototyp nasazen v době dokončení psaní této práce.

Plány na další rozvoj

V budoucnu by měly nástroje a návrhy vytvořené v této práci být integrovány přímo do aplikace TTRebel a využity ve společnosti Orchitech Solutions, s.r.o.

Literatura

- [1] WILSON, Scott. How to evaluate the sustainability of an open source project. *OSS Watch* [online]. Oxford: University of Oxford, 2019, 2013-12-11 [cit. 2019-12-10]. Dostupné z: <http://oss-watch.ac.uk/resources/evaluating-sustainability>
- [2] Jira: Issue & Project Tracking Software. *Atlassian* [online]. Sydney (Australia): Atlassian Corporation, 2019 [cit. 2019-12-10]. Dostupné z: <https://www.atlassian.com/software/jira>
- [3] Explore apps for Jira. *Atlassian Marketplace* [online]. Sydney (Australia): Atlassian Corporation, 2019 [cit. 2019-12-10]. Dostupné z: <https://marketplace.atlassian.com/addons/app/jira>
- [4] *Redmine: Overview* [online]. 2019 [cit. 2019-12-10]. Dostupné z: <https://www.redmine.org/projects/redmine/wiki>
- [5] *OpenProject: online management software* [online]. Berlín (Německo): OpenProject Foundation, 2019 [cit. 2019-12-10]. Dostupné z: <https://www.openproject.org/>
- [6] *Toggl: Github* [online]. Tallinn (Estonsko): Toggl OÜ, 2019 [cit. 2019-12-10]. Dostupné z: <https://github.com/toggl>
- [7] *toggldesktop: LICENSE* [online]. Tallinn (Estonsko): Toggl OÜ, 2019 [cit. 2019-12-10]. Dostupné z: <https://github.com/toggl/toggldesktop/blob/4ba785e93d681e3601c622f6916c5c10dc14602d/LICENSE>
- [8] *toggl_api_docs: toggl_api.md* [online]. Tallinn (Estonsko): Toggl OÜ, 2019 [cit. 2019-12-10]. Dostupné z: https://github.com/toggl/toggl_api_docs/blob/6061dce2a6d345fef3c3d9ce0cc09c47a3eb0efd/toggl_api.md
- [9] *Toggl: Integrations* [online]. Tallinn (Estonsko): Toggl OÜ, 2019 [cit. 2019-12-10]. Dostupné z: <https://toggl.com/integrations/>
- [10] *ActivityWatch* [online]. 2019 [cit. 2019-12-10]. Dostupné z: <https://activitywatch.net/>

- [11] *ActivityWatch: Watchers* [online]. 2018 [cit. 2019-12-10]. Dostupné z: <https://activitywatch.readthedocs.io/en/latest/watchers.html>
- [12] *ActivityWatch v0.8.1* [online]. 2019 [cit. 2019-12-10]. Dostupné z: <https://activitywatch.readthedocs.io/en/latest/changelog.html#v0-8-1>
- [13] *ActivityWatch v0.8.3* [online]. 2019 [cit. 2019-12-10]. Dostupné z: <https://activitywatch.readthedocs.io/en/latest/changelog.html#v0-8-3>
- [14] *ActivityWatch Watchers* [online]. 2019 [cit. 2019-12-10]. Dostupné z: <https://activitywatch.readthedocs.io/en/latest/watchers.html#watchers>
- [15] *ActivityWatch Querying Data* [online]. 2019 [cit. 2019-12-10]. Dostupné z: <https://activitywatch.readthedocs.io/en/latest/querying-data.html>
- [16] *RescueTime: Time management software for staying productive and happy in the modern workplace* [online]. United States: RescueTime, 2019 [cit. 2019-12-10]. Dostupné z: <https://www.rescuetime.com/>
- [17] *WakaTime: Open source plugins, goals, and automatic time tracking for programmers* [online]. San Francisco (USA): WakaTime, 2019 [cit. 2019-12-10]. Dostupné z: <https://wakatime.com/>
- [18] *Timeular: Track, understand and improve how you work* [online]. Graz (Rakousko): Timeular, 2019 [cit. 2019-12-10]. Dostupné z: <https://timeular.com/>
- [19] *Timeular: Export & Integrate* [online]. Graz (Rakousko): Timeular, 2019 [cit. 2019-12-10]. Dostupné z: <https://timeular.com/#1506004083222-91cc8c6e-5f898bb8-bce5>
- [20] *Tracker - the most effortless and accurate way to track your time* [online]. Graz (Rakousko): Timeular, 2019 [cit. 2019-12-10]. Dostupné z: <https://timeular.com/product/tracker/>
- [21] *TimeFlip* [online]. Time Solutions, 2019 [cit. 2019-12-10]. Dostupné z: <https://timeflip.io/>

- [22] *TimeFlip: Buy TimeFlip* [online]. Time Solutions, 2019 [cit. 2019-12-10]. Dostupné z: <https://timeflip.io/en#buytimeflip>
- [23] *TimeFlip: Integrations* [online]. Time Solutions, 2019 [cit. 2019-12-10]. Dostupné z: <https://timeflip.io/en/integration>
- [24] *TimeFlip: API Playground* [online]. Time Solutions, 2019 [cit. 2019-12-10]. Dostupné z: <https://app.timeflip.io/api>
- [25] *Implement tagging/annotation* [online]. 2019 [cit. 2019-12-10]. Dostupné z: <https://github.com/ActivityWatch/activitywatch/issues/95>
- [26] Developer guide. *Redmine* [online]. Redmine, 2019-01-25 [cit. 2019-12-10]. Dostupné z: https://redmine.org/projects/redmine/wiki/Developer_Guide/history
- [27] Cross-Origin Resource Sharing (CORS). *MDN web docs* [online]. Mozilla, 2019-03-18 [cit. 2019-12-10]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>
- [28] Your Go-To Jira Glossary. *Atlassian Community* [online]. Sydney (Australia): Atlassian, 2019, 2017-07-05 [cit. 2019-12-10]. Dostupné z: <https://community.atlassian.com/t5/Jira-articles/Your-Go-To-Jira-Glossary/ba-p/605232>
- [29] Scrum Glossary. *Scrum.org* [online]. 2019 [cit. 2019-12-10]. Dostupné z: <https://www.scrum.org/resources/scrum-glossary>

Seznam obrázků

| | | |
|-----|---|----|
| 3.1 | Jira | 10 |
| 3.2 | Redmine | 11 |
| 3.3 | Toggl | 12 |
| 3.4 | TTRebel | 13 |
| 6.1 | Vypsání TTRebel API. | 27 |
| 6.2 | Ukázka TTRebel GUI v 2. | 27 |
| 6.3 | Prvotní design TTRebel tlačítka v Redminu | 29 |

Seznam tabulek

| | | |
|-----|--|----|
| 3.1 | Porovnání funkcí trackrů | 15 |
| 6.1 | Základní proměnné TTRebel session. | 32 |
| 6.2 | Použité ActivityWatch moduly. | 32 |

Seznam ukázek zdrojového kódu

| | | |
|------|---|----|
| 6.1 | Query filtrující data pro specifickou <code>TrackingSession</code> | 32 |
| 6.2 | Proměnné použité v query. | 33 |
| 6.3 | Query filtrující eventy zařaditelné do předprodeje. | 34 |
| 6.4 | Query filtrující eventy zařaditelné do analýzy. | 34 |
| 6.5 | Query filtrující eventy zařaditelné jako konzultace s kolegy nebo zákazníkem. | 35 |
| 6.6 | Query filtrující eventy identifikovatelné jako vývoj. | 35 |
| 6.7 | Query filtrující eventy identifikovatelné jako management. | 35 |
| 6.8 | Proměnné použité v query. | 36 |
| 6.9 | Naformátování výstupu query kategorizující aktivitu uživatele. | 36 |
| 6.10 | Query filtrující potenciální issue id. | 36 |
| 6.11 | Query filtrující potenciální issue id. | 37 |