

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Porovnání SW nástrojů pro řízení projektu JIRA a Trello
Bakalářská práce

Autor: Dominik Byrtus
Studijní obor: Aplikovaná informatika

Vedoucí práce: doc. RNDr. Petr Tučník, Ph.D.

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 14.8.2023

vlastnoruční podpis

Dominik Byrtus

Poděkování:

Děkuji vedoucímu bakalářské práce doc. RNDr. Petru Tučníkovi, Ph.D. za metodické vedení práce, za cenné rady a za čas, který mi věnoval. Dále děkuji svým spolupracovníkům, kteří mi poskytli důležité informace k případovým studiím a v neposlední řadě děkuji své přítelkyni, která mě při psaní práce taktéž velmi podporovala.

Anotace

Tato bakalářská práce se zabývá projektovým řízením s využitím agilních a vodopádových metodik a porovnáním dvou nástrojů pro řízení projektů z hlediska jejich implementace a funkcionalit. Pro srovnání možností projektového řízení jsou využity nástroje JIRA a Trello (obojí od společnosti Atlassian). Práce na případových studiích přibližuje řešení implementace nástroje pro projektové řízení a poukazuje na hlavní aspekty determinující výkonnost a efektivitu práce v kontextu projektového řízení. Na vybraných případových studiích demonstrujících projekty různého rozsahu také provádí srovnání klíčových charakteristik vybraných platforem pro řízení projektů. Aplikace Trello přináší výhody zejména díky své jednoduchosti a intuitivnosti, avšak u rozsáhlejších projektů s potřebou rozšířených funkcionalit a manažerských reportů je pro řízení projektů vhodnější využít platformu Jira.

Klíčová slova: Projektový řízení, Jira, Trello, agilní metoda, vodopádová metoda, softwarové nástroje, Scrum

Annotation

Title: Comparison of SW Tools for Project Management

JIRA and Trello

This bachelor thesis deals with the project management using agile and waterfall methodologies and compares two project management tools in terms of their implementation and functionalities. To compare project management possibilities, JIRA and Trello products (both from Atlassian) are utilized. The aim of this thesis is to use case studies to present the implementation solutions of the project management tool and to highlight the main aspects determining the performance and efficiency of work in the context of project management. A comparison of the key characteristics of the selected project management platforms is made as well, using selected case studies, demonstrating projects of different scales. Trello offers advantages primarily due to its simplicity and intuitiveness, however, for more extensive projects requiring advanced features and managerial reports, Jira is a more suitable project management platform.

Keywords: Project management, Jira, Trello, agile method, waterfall method, software tools, Scrum

Obsah

1	Úvod.....	1
2	Teoretická část práce	3
2.1	Projektové řízení.....	3
2.2	Projekt.....	4
2.2.1	Řízení rizik projektu	6
2.3	Světové standardy projektového managementu.....	8
2.3.1	PRINCE2	8
2.3.2	PMBOK Guide	10
2.3.3	IMPA.....	10
2.4	Agilní a vodopádové přístupy v projektovém řízení.....	12
2.4.1	Vodopádové přístupy pro řízení projektů	12
2.4.2	Agilní přístupy pro řízení projektu.....	14
2.4.3	Metodika Scrum	16
2.4.4	Srovnání agilních a tradičních přístupů v projektovém řízení	20
3	Nástroje pro řízení projektů	22
3.1	Softwarové nástroje	23
3.2	Jira.....	24
3.2.1	Úkol	24
3.2.2	Projekt	25
3.2.3	Pracovní postup.....	26
3.2.4	JQL.....	28
3.3	Trello	29
3.3.1	Pracovní prostor	30
3.4	Porovnání nástrojů Jira a Trello.....	31
4	Praktická část	34
4.1	Případová studie A – softwarový vývoj za pomoci Scrum metodiky.....	34
4.1.1	Případová studie A – projekt agilního řízení vývoje pomocí metodiky Scrum	34
4.1.2	Použití v Trello – případová studie A.....	37
4.1.3	Shrnutí případové studie	38

4.2	Případová studie B – řízení IT projektů za pomoci vodopádové metodiky.....	39
4.2.1	Použití v Jira – případová studie B.....	40
4.2.2	Použití v Trello.....	41
4.2.3	Shrnutí případové studie.....	42
4.3	Případová studie C – Projekt pro řízení operačních rizik	43
4.3.1	Použití v Jira.....	44
4.3.2	Použití v Trello.....	45
4.3.3	Shrnutí případové studie C.....	46
5	<i>Shrnutí výsledků, závěr a doporučení.....</i>	47
6	<i>Seznam použité literatury.....</i>	50

Seznam obrázků

Obrázek 1: Generický životní cyklus projektu	5
Obrázek 2: Struktura PRINCE2	9
Obrázek 3: Oko kompetencí	11
Obrázek 4: Vodopádová metodika s milníky	13
Obrázek 5: Zastoupení přístupů v projektovém řízení	21
Obrázek 6: Ukázka pracovního postupu – Jira	26
Obrázek 7: Ukázka filtrování pomocí JQL– Jira.....	28
Obrázek 8: Pohled na nástěnku v Trello	30
Obrázek 9: Přehled karty.....	31
Obrázek 10: Příprava sprintu v backlogu.....	36
Obrázek 11: Scrum board.....	36
Obrázek 12: Zobrazení časové osy v Jira	41
Obrázek 13: Zobrazení časové osy v Trello	42

Seznam tabulek

Tabulka 1: Principiální hodnoty agilního manifestu	14
Tabulka 2: Porovnání agilní a tradičních přístupů.....	20
Tabulka 3: Porovnání nástrojů	32

1 Úvod

Vývoj software je jedním z nejdynamičtějších odvětví a v kontextu současného vysokého tempa změn na trhu díky rychlému vývoji technologií a globální konkurence jsou kladeny stále vyšší požadavky na technologická řešení zejména ohledně kvality, inovativnosti a rychlosti dodání. V rámci vývoje softwaru se tak klade důraz na nepřetržité přizpůsobování se požadavkům a očekáváním trhu, z čehož plyne nutnost reagovat na nové příležitosti, ale i hrozby. Jedním ze způsobů, jak pružně reagovat na tyto změny a požadavky je zefektivňovat projektové řízení pomocí určitých softwarových nástrojů, které to umožňují.

Projektové řízení tvoří nedílnou součást softwarového vývoje pro úspěšnou realizaci projektů. V rámci projektového řízení existují mimo jiné dva z neznámějších nástrojů pro zefektivňování projektového řízení – Jira a Trello. Oba tyto nástroje vyvinula společnost Atlassian, kdy velmi zjednodušeně je Jira rozsáhlejší a komplexnější nástroj, za nějž se pro vyšší počet uživatelů platí licenční poplatek, kdežto Trello je jednodušší nástroj s omezenými funkcionalitami, který v základní verzi zpoplatněn není. Těmito nástroji a jejich porovnáním se detailněji zabývá i tato práce.

Samotná práce se dělí na tři hlavní části. První částí je teoretická část zabývající se problematikou projektového řízení. Čtenář se tak seznámí se základními pojmy jako je projektové řízení a projekt. Práce také přibližuje světové standardy projektového managementu a v neposlední řadě analyzuje a srovnává agilní a vodopádové přístupy k projektovému řízení. Po teoretické části následuje představení, analýza a srovnání funkcionalit dvou konkrétních nástrojů pro řízení projektů Jira a Trello. V navazující praktické části jsou aplikovány poznatky z předchozích kapitol do praxe pomocí případových studií, na kterých se demonstruje využití těchto nástrojů a ukázka jejich funkcionalit při použití na konkrétních IT/SW projektech, které byly úspěšně realizovány a na kterých se autor přímo podílel.

Hlavním cílem je tak poskytnout ucelený pohled na problematiku projektového řízení a jeho nástrojů a poté přiblížit možnost jejich využití na konkrétních případech. Práce tedy poskytne čtenáři se zájmem o problematiku, vývojovému týmu či projektovým manažerům teoretický i praktický podklad, který lze využít např. při rozhodování o vhodnosti softwarového nástroje pro konkrétní projekt.

Autor při tvorbě bakalářské práce využívá studium domácích i zahraničních publikací zaměřených na projektové řízení a jeho nástroje. Mezi tyto publikace patří mimo jiné i oficiální technická dokumentace zveřejněná společností Atlassian. Vedle toho však autor uplatňuje své znalosti a zkušenosti nabyté v zaměstnání na pozici Atlassian konzultanta. Tyto znalosti zužitkovává zejména v praktické části, v které do případových studií zpracovává informace o proběhlých projektech. Výsledkem by také mělo být srovnání zmíněných nástrojů projektového řízení a vyvození závěru o vhodnosti jejich využití. Jednou z otázek, na které autor v práci hledá odpověď je, jak Trello jako jednodušší nástroj obstojí v porovnání s nástrojem Jira v použitelnosti při konkrétních projektech řešených v případových studiích.

2 Teoretická část práce

Pro účely praktické části je nutné vymezit či seznámit čtenáře se základními pojmy, které jsou použity v praktické části práce. Teoretická část se zabývá popisem projektového řízení jako celku, také definuje termíny jako projekt a jeho životní cyklus. Další část teoretické práce se soustředí na mezinárodní standardy v oblasti projektového řízení, zahrnující IPMA, PMBOK a PRINCE2. Poslední sekce teoretického obsahu seznamuje čtenáře s tradičním projektovým řízením a agilním přístupem k řízení projektů.

2.1 Projektové řízení

Doležal a kol. (2016) uvádí, že projektové řízení (Project management) je metoda pro navrhování a realizaci procesu změny, aby se dosáhlo cíle v daném čase a rozpočtu, aniž by vznikly nežádoucí vedlejší efekty. Jde o soubor norem, doporučení a nejlepších praxí pro řízení projektu. Je to všeobecný přístup k řešení problému, nikoli konkrétní směrnice nebo návod.

Projektové řízení – definice ISO 21500 (Zandhuis a Stellingwerf, 2013): *„Projektové řízení je aplikace metod, nástrojů, technik a kompetencí na projekt. Projektové řízení zahrnuje integraci různých fází životního cyklu projektu. Projektové řízení se provádí prostřednictvím procesů.“*

PRINCE2 uvádí následující definici: *„Projektové řízení je plánování, delegování, monitorování a řízení všech aspektů projektu, a motivace k jejich řešení, dosažení projektových cílů s očekávanými výkonnostními parametry času, nákladů, kvality, rozsahu, přínosů a rizik.“*

Doležal a kol. (2016) projektové řízení definuje jako *„Soubor pravidel, postupů, metod a nástrojů, které pomáhají projektovým týmům koordinovat společné úsilí tak, aby dodaly správné výsledky, ve správný čas, pro správného zákazníka, a to vše s omezenými zdroji.“*

První z uvedených formulací se zaměřuje zejména na fáze projektového cyklu a procesy. Druhá z definic se vztahuje zejména na aktivity, které jsou s projektovým řízením spojeny jako je plánování, delegování a monitorování. Poslední definice naopak projektové řízení vnímá jako soubor pravidel pro koordinaci týmů. Výše uvedené rozdílné definice odrážejí odlišné pohledy a přístupy na proces projektového

řízení. Každý z citovaných rámců uvádí vlastní zaměření a priority, což vede k odlišným formulacím.

2.2 Projekt

Pro účinné řízení projektů je klíčové pochopit základní podstatu toho, co pojem projekt představuje. Projekty jsou součástí lidské historie od nejranějších dob civilizace a mohou spočívat v jednoduchých úkolech jako je lov až po složité jako stavba pyramidy nebo vyslání lidí do vesmíru. Termín "projekt" se také v každodenním jazyce používá poměrně běžně. Všechny typy projektů, ať už jde o lov, stavbu pyramid nebo opravu kohoutků, mají určité společné charakteristiky, které je činí projekty (Watt, 2014).

Na rozdíl od běžných nebo trvalých obchodních aktivit je dle Watta (2014) projekt specifický a disponuje určitými charakteristikami. Zároveň trvá pouze dočasně a je časově ohraničen. Významná část projektu se tak soustředí na to, aby byl projekt dokončen v předem určeném čase. To s sebou nese i nutnost tvorby časových plánů úkolů spojených s projektem. Projekty mohou trvat několik minut, ale i několik let. V předem určeném časovém rozmezí jsou tak tvořeny nové produkty či služby, které doposud vytvořeny nebyly. Výsledek takového projektu je považován za jedinečný, což znamená, že doposud vůbec neexistoval, nebo byl prováděn odlišně. Projekt je také třeba odlišit od pojmu operace. Ty jsou na rozdíl od projektu trvalé a opakují se. Operace zajišťují bezproblémový chod organizace. V případě projektu je účelem dosažení cíle a poté je projekt ukončen, kdežto operace trvá bez přerušení a neustále se opakuje za účelem zmíněného udržení chodu dané organizace. Projekt je tak oproti operaci jednorázový a jedinečný a je dokončen v tu chvíli, kdy je splněn jeho účel a cíl. Projekt však za určitých okolností může skončit i předčasně, a to v případě, že se během trvání projektu zjistí, že cíle projektu nelze naplnit.

Metodika PRINCE2 uvádí, že *“projekt je dočasná organizace, která je vytvořena za účelem poskytování jednoho nebo více specializovaných produktů, podle dohodnutého Obchodního Případu”*.

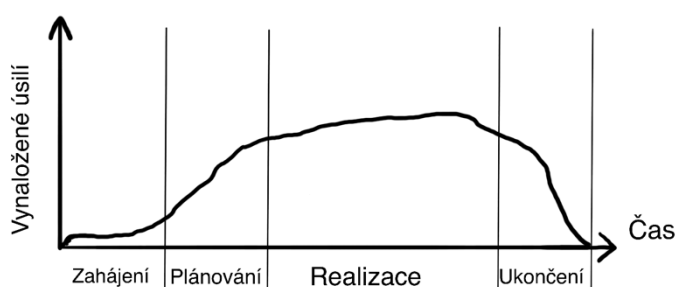
Naopak norma ISO 21500 (Zandhuis a Stellingwerf, 2013) projekt definuje jako *“unikátní sestavu procesů, která se skládá z koordinovaných a řízených činností s počátkem a koncem, které jsou prováděny za účelem dosažení stanoveného cíle”*.

Dosažení cíle projektu vyžaduje výstupy, které splňují specifické požadavky, včetně více omezení, jako jsou čas, náklady a zdroje.”

Podobně jako u projektového řízení existuje mnoho různých definic slova projekt, jelikož se používá v různých oblastech a každá oblast vyžaduje odlišný přístup. Jak uvádí Doležal a kol. (2016), projekt však lze vždy interpretovat jako jasně vymezenou úpravu, při níž dochází ke kontrolované transformaci ze vstupního stavu do zamýšleného cílového stavu.

Pro určení, zda se jedná o projekt, je důležité zvážit několik klíčových bodů neboli charakteristik projektu. Nejdříve je nutné určit, zda se jedná o jednorázový projekt nebo trvalou operaci. Dále musí být provedena identifikace jednotlivce nebo skupiny zainteresované na výsledku projektu. Také musí být rozpoznány specifické vlastnosti, které jej činí projektem. Jedná se o jeho jedinečnost a dočasnou a časově ohraničenou povahu (Watt, 2014).

Projekt má kvůli své časové ohraničenosti počáteční a konečný bod, kromě toho však prochází i řadou souvisejících a relativně předvídatelných fází, které jsou pro něho charakteristické. Wells (2019) ve své publikaci udává že, souhrn veškerých těchto fází se označuje jako životní cyklus projektu. V rámci tohoto životního cyklu se projekt nachází ve fázi zahájení, plánování, realizace a ukončení, což lze vyčíst z grafu na obrázku níže.



Obrázek 1: Generický životní cyklus projektu

Zdroj: Vlastní tvorba dle Wells, 2019

Dále ve své publikaci Wells (2019) uvádí, že přechod z jedné fáze do druhé bývá často provázen procesem schvalování, který napomáhá zajistit plynulý průběh. Toto schvalování se obvykle řídí smlouvou nebo dohodou. Tato dohoda obvykle uvádí projekt v platnost a uděluje garantovi projektu pravomoci, určuje podrobný plán

projektu, ze kterého vyplývá další postup, obsahuje akceptaci zákazníkem hlavních dílčích výstupů projektu a požadovanou dokumentaci nutnou pro dokončení projektu.

Přestože výše zmíněné charakteristiky vypadají velmi jednoduše, je třeba mít na paměti, že každý projekt disponuje svou jedinečností, a proto se projekty od sebe vždy značně liší. Generický životní cyklus projektu na obrázku 1 nelze aplikovat na jakýkoliv projekt.

Každý projekt má svůj specifický životní cyklus v závislosti na charakteristikách projektu a odvětví, ve kterém se projekt realizuje. Rozdíl v životních cyklech projektů často souvisí s mírou nejistoty existující na začátku projektu. Na základě této nejistoty je pak zvolena metodika řízení projektu. S extrémní nejistotou jsou spojeny metodiky vodopádové a agilní (Wells, 2019), kterým se dále detailněji bude věnovat tato práce.

Svozilová (2016) uvádí, že z definic projektu vychází tři hlavní charakteristiky, a tedy *tři základny projektového managementu*. Tyto základny jsou hlavními principy projektového managementu. Principy spočívají ve vytvoření rámce, který umožňuje na základě předem stanovených cílů realizaci výstupu nebo výsledku projektu. Za tyto principy lze označit:

- *Čas* představující horní hranici pro plánování jednotlivých aktivit projektu.
- *Disponibilita zdrojů* pro užívání a čerpání v rámci projektu.
- *Náklady* vyjadřující finanční využití zdrojů v určitém časovém období.

Jednou z podmínek úspěšného dokončení projektu je udržování výše zmíněného systému v rovnováze. Pro udržení rovnováhy je vypracován plán projektu, na základě kterého se veškerá tvorba na projektu koordinuje. V souladu s plánem je také nutno pomocí kontrolních mechanismů monitorovat jeho dodržování na základě předem stanovených cílů a limitů (Svozilová, 2016).

2.2.1 Řízení rizik projektu

Rizika jsou spojena s nejistotou. Riziko představuje nejistou událost nebo stav, který může mít pozitivní či negativní vliv na cíle projektu. Nepříznivá rizika se označují jako hrozby a prospěšná rizika jako příležitosti. Veškeré projekty se pojí s riziky, protože jsou unikátní a nesou s sebou určitý stupeň nejistoty (Project Management Institute, 2021).

Analýza rizik může představovat celý proces nebo pouze jeho část. V případě, že se hovoří pouze o části procesu, jedná se samotnou analýzu rizik spojených s projektem. Také však lze hovořit o tom, že termín zahrnuje celý proces řízení rizik projektu, který obsahuje více kroků, mimo jiné i analýzu rizik. Riziko v moderním projektovém řízení zahrnuje nejisté situace s negativním dopadem, ale také situace s potenciálně pozitivními dopady. Přípravným krokem je vytvoření plánu řízení rizik (*risk management plan*) jako součást celkového plánu projektu. Tento plán určuje strategie a pravidla, díky kterým se lze efektivně na rizika připravit a minimalizovat jejich dopady (Doležal a kol., 2016).

Doležal a kol. (2016) se také odkazuje na normu ČSN 31000, podle které mezi procesy pro řízení rizik patří:

- Stanovení kontextu
- Identifikace rizik
- Analýza rizik
- Hodnocení rizik
- Ošetření rizik
- Monitoring a přezkoumání
- Komunikace a konzultace

Po analýze rizika se musí určit, jak s ním naložit. Pokud je riziko spojeno s hrozbou, musí se určit strategie vypořádání se s rizikem a hrozbou. Project Management Institute (2021) hovoří o těchto strategiích:

- Vyhnutí se – snažit se zabránit hrozbě nebo chránit projekt před ní.
- Eskalace – situace se předá vyšší instanci za předpokladu, že je mimo pravomoci.
- Přenesení – riziko lze přesunout na třetí stranu.
- Zmírnění – pomocí preventivních opatření se sníží riziko hrozby.
- Akceptace – uznání existence hrozby za předpokladu připravení se na ní nebo pasivní reakce.

2.3 Světové standardy projektového managementu

Standardy v oblasti projektového řízení reprezentují kompilaci znalostí a praktických zkušeností získaných projektovými manažery během průběhu času. Oproti jiným odvětvím, kde existují striktní normy a jednoznačné procedury je v oblasti projektového řízení nezbytné přistupovat k jednotlivým projektům s ohledem na jejich jedinečné charakteristiky. Vzhledem k nekonečným variacím projektů není možné vytvořit univerzální standard, který by pokryl celé spektrum situací. V projektovém řízení se tak standard dle Doležala a kol. (2016) spíše proměňuje v doporučení než ve striktně závaznou normu. V současnosti existuje několik celosvětově uznávaných přístupů pro řízení projektu, mezi ně se např. řadí:

- Projects In Controlled Environments (*PRINCE2*)
- Project Management Body of Knowledge (*PMBOK Guide*)
- IMPA od organizace International Project Management Association
- ISO 21500

2.3.1 PRINCE2

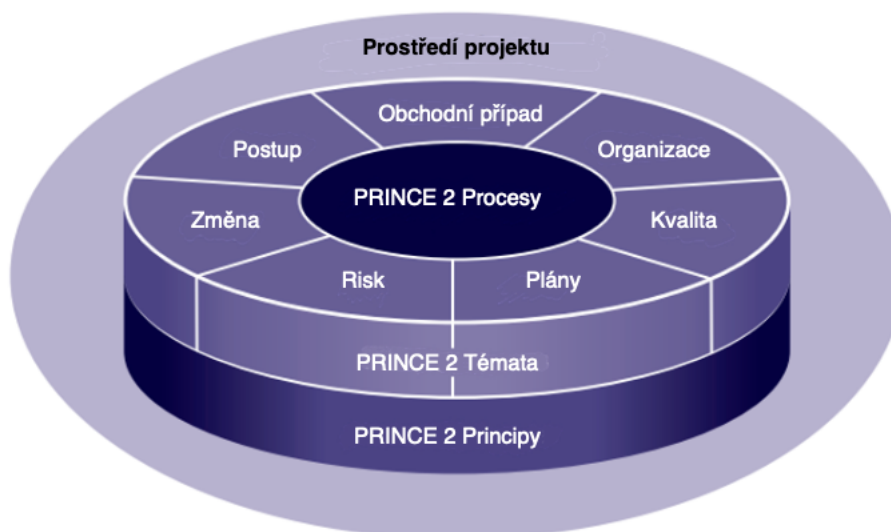
Zkratka PRINCE2 vznikla z anglického „*Projects IN Controlled Environments*“, tedy ve volném překladu projekty v řízeném prostředí. PRINCE2 je univerzální projektová metodika, kterou lze využít na jakýkoli projekt bez ohledu na jeho velikost, typ organizace, lokalitu nebo kulturu. Základy pro tuto metodiku tvoří ověřené principy. PRINCE2 by měla poskytovat správné informace ve správný čas správným lidem, aby mohli dělat správná rozhodnutí a mít projekt pod kontrolou. Tato metodika se dnes stále více využívá nejen při řízení projektů ve veřejném sektoru, ale také ji doporučuje Evropská komise pro projekty podporované z prostředků EU jako jednu z metodik projektového řízení (Máchal a kol., 2015).

PRINCE2 představuje návod a metodiku práce na projektech, nejedná se tak o standard jako je IPMA nebo PMI. Odlišnost metodiky PRINCE2 tak spočívá nejen ve způsobu zpracování, ale také ve využití odlišné terminologie, která byla zvolena s ohledem na prostřední státní správy, ve kterém vznikla. Máchal a kol. (2015) tak pro přehlednost využívá zažitou terminologie komerčního prostředí, a nikoliv původní terminologii PRINCE2.

Metodika PRINCE2 disponuje následujícími hlavními integrovanými prvky neboli elementy (Axelos, 2017):

- Principy – PRINCE2 je metodika řízení projektů, která se opírá o osvědčené principy. Používání těchto principů je to, co určuje, zda je projekt řízen pomocí PRINCE2. Tyto principy pomáhají efektivně využít PRINCE2. PRINCE2 je metodika, která klade důraz na přizpůsobení se projektu, nepoužívá se tedy příliš přísně. Měla by zejména napomáhat úspěchu projektu.
- Témata – PRINCE2 témata zahrnují aspekty řízení projektu, které je nutné monitorovat a řešit průběžně. V projektu se využije všech sedm témat, avšak za podmínky přizpůsobení se a specifickému přístupu k jednotlivým projektům.
- Procesy – Procesy popisují postup od počátečních aktivit před zahájením projektu až po fáze životního cyklu projektu a uzavření projektu. Každý proces obsahuje doporučené seznamy aktivit, výstupy a související odpovědnosti.
- Prostředí projektu – PRINCE2 je metodika pro řízení projektů, která může být používána bez ohledu na rozdíly ve velikosti, složitosti a dalších charakteristik projektu. Organizace často hledají jednotný přístup k řízení projektů a adaptují metodiku PRINCE2 pro vytvoření vlastního metodického přístupu k řízení projektů. Tato metodika je následně zahrnuta do pracovního rámce organizace.

Výše popsané elementy jsou pro přehlednost zasazeny do následujícího obrázku, který poukazuje na to, jak spolu jednotlivé elementy souvisí.



Obrázek 2: Struktura PRINCE2

Zdroj: Vlastní tvorba dle Axelos, 2017

2.3.2 PMBOK Guide

Project Management Institute (2021) používá označení PMBOK jako zkratku pro publikaci „*A Guide to the Project Management Body of Knowledge*“. Jedná se o komplexní soubor osvědčených postupů, terminologie, nástrojů a technik používaných v oblasti řízení projektů. PMBOK Guide definuje základní koncepty a procesy související s projektovým managementem a poskytuje strukturovaný přístup pro správu projektů různého rozsahu a složitost.

Jedná se o procesní přístup k oblasti řízení projektů. Definuje pět hlavních skupin procesů, deset oblastí znalostí a jednotlivé procesy a jejich vztahy. Všechny tyto procesy a kroky jasně definují své komponenty, výstupy a nástroje (Doležal a kol., 2016).

Dle informací Visual Paradigm (2023) se mezi hlavních pět procesních skupin řadí:

1. Zahájení – procesy zahrnující kroky, činnosti a dovednosti nezbytné k účinné definici počátku projektu.
2. Plánování – tyto procesy určují kroky pro vymezení rozsahu projektu, strategické plánování práce, tvorbu prioritních seznamů a plánování týmu.
3. Provádění – procesy pro efektivní řízení týmu, současně koordinují i časová očekávání a dosahování cílů.
4. Monitoring a controlling – procesy zahrnující správu změnových příkazů, rozpočtové úvahy a neočekávané situace, které mohou ovlivnit schopnost týmu naplnit očekávání projektu.
5. Uzavření – procesy pro úspěšné uzavření projektu, což znamená splnění stanovených časových a rozpočtových limitů.

Manažeři projektů, kteří jsou obeznámeni se standardy PMBOK, mají možnost upravit proces řízení projektů tak, aby co nejlépe odpovídal specifickým požadavkům jejich společnosti.

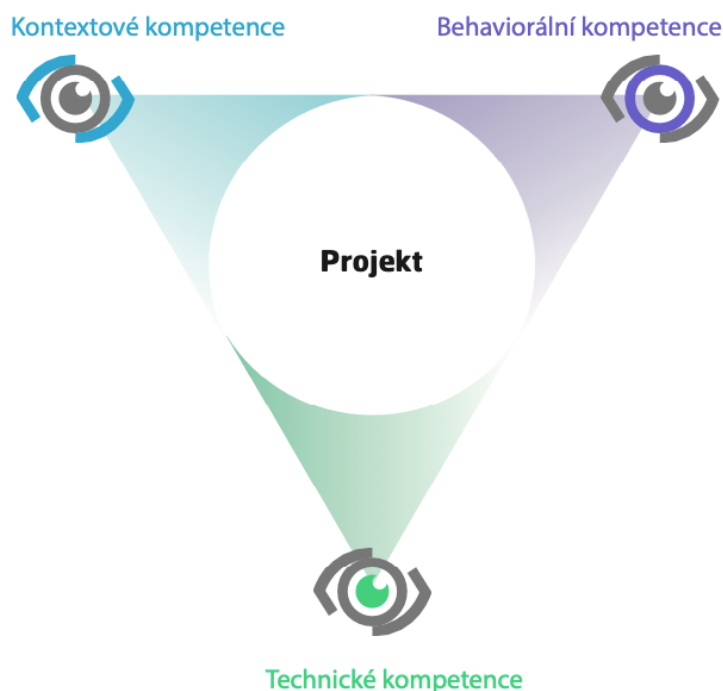
2.3.3 IMPA

International Project Management Association vydává standard ICB (IPMA Competence Baseline), který představuje mezinárodní rámec kompetencí v oblasti řízení projektů. Tento soubor standardů detailně definuje dovednosti, schopnosti

a znalosti, které projektoví manažeři a další odborníci v této oblasti potřebují k efektivnímu plánování, provádění a monitorování projektů (Máchal a kol., 2017).

Podle Doležala a kol. (2016) se standard zaměřuje více na dovednosti a kompetence než na přesnou aplikaci definovaných procesů. Jak dále uvádí, standard ICB nepřikazuje určitý chod procesů, poukazuje však jen na specifické kroky a jejich správné využití v konkrétním případě. Na místě je i určitá kreativita a vlastní názor.

Problematika projektového řízení se člení do tří kategorií. Pro úspěšné řízení projektů je nezbytné, aby projektový manažer disponoval specifickými dovednostmi, které jsou rozděleny právě do těchto kategorií. Kontextové kompetence se věnují širším souvislostem projektů, behaviorální kompetence se zaměřují na interpersonální a sociální dovednosti a technické kompetence se týkají konkrétních technických aspektů projektového řízení. Rámec kompetencí je také označován jako OKO kompetencí IPMA (Máchal a kol., 2017). Důvod pro toto označení znázorňuje následující obrázek.



Obrázek 3: Oko kompetencí

Zdroj: Mezinárodní standard projektového řízení IPMA ICB v. 4, Máchal a kol., 2017

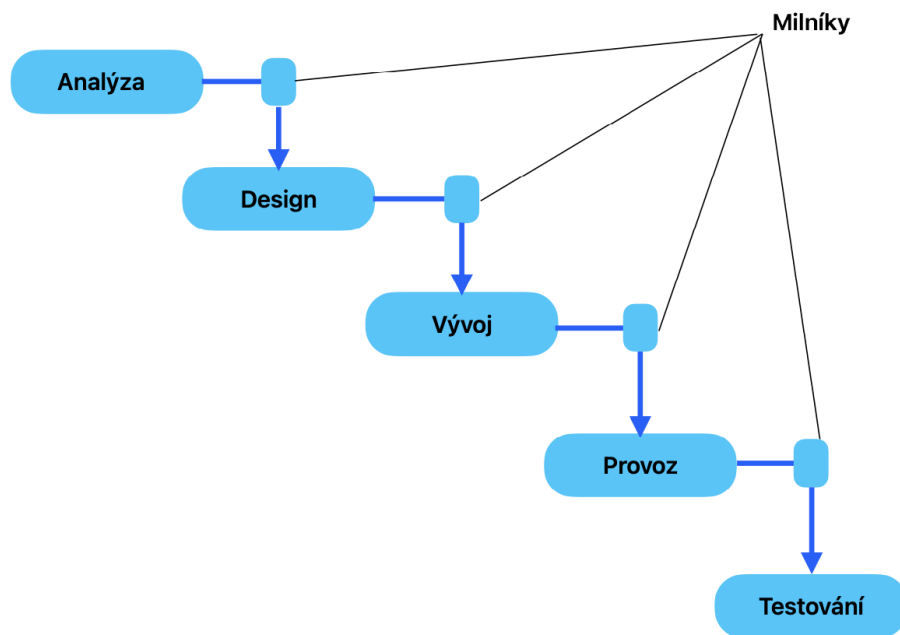
2.4 Agilní a vodopádové přístupy v projektovém řízení

Podle Hoory (2022) jsou agilní a vodopádové metodiky dvě často uváděné metodiky využívané pro řízení projektů. Obě metodiky jsou často využívány při vývoji softwaru, nicméně každá z nich je optimální pro různé druhy projektů.

Vodopádová metodika se řadí mezi tradiční přístupy. Označení tradiční má za úkol odlišení od moderních agilních přístupů. Název tradiční tak poukazuje na jejich dřívější existenci a také využití konvenčních postupů ve vývoji software. Uplatňují se v projektech s větším rozsahem, větší stabilitou prostředí a tam, kde je nutné dodržovat funkcionalitu, rozpočty a termíny. Na druhé straně agilní metodika znamená odlišný přístup k procesům. Vše je podřízeno kvalitě výstupu a efektivitě procesů, eliminuje se zbytečná byrokracie (Myslín, 2016). Na rozdíl od tradičních metodik jsou však vhodné v prostředí, kde panuje více nejistoty, méně organizace a stability. Mezi agilní metodiky se řadí Scrum, Lean a Kanban.

2.4.1 Vodopádové přístupy pro řízení projektů

Myšlenka vodopádové metodiky spočívá dle Jøsang a kol. (2015) v tom, že jednotlivé fáze musí být úplně dokončeny před tím, než dojde k přesunu na následující fázi, což je pomocí „vodopádu“ symbolizováno na obrázku 4. Na začátku tak musí být stanoveny a detailně popsány veškeré požadavky. Mělo by se předcházet návratům k předchozí fázi, jelikož to může způsobit značný nárůst režijních nákladů. Nicméně v praxi se toto často děje, je nutné se k předchozí fázi vrátit a přepracovat některé požadavky projektu, a proto projekty založené na vodopádové metodice často utrpěly velké finanční a časové ztráty.



Obrázek 4: Vodopádová metodika s milníky

Zdroj: Vlastní zpracování dle Rubin, 2012

Myslín (2016) také uvádí, že vodopádová metodika ukazuje na tři hlavní charakteristické znaky metodiky:

1. Lineární průběh – po ukončení jedné fáze následuje druhá.
2. Jednoznačnost – lze přesně určit v jaké fázi se projekt nachází.
3. Úplné zadání – přesun do následující fáze následuje až dokončení a zhotovení výstupů fáze předchozí.

Jedním z cenných prvků u vodopádových projektů je pečlivě vymezená sada milníků. Tyto milníky slouží jako kontrolní body v průběhu životního cyklu projektu, na které se manažeři mohou orientovat a které často souvisí s rozhodnutími ohledně dalších fází. Přestože mohou být tyto milníky užitečné pro účely řízení, poskytují pouze omezený pohled na skutečný stav dodávky hodnoty pro zákazníka (Rubin, 2012).

2.4.2 Agilní přístupy pro řízení projektu

Dle Doležala a kol. (2016) při řízení IT projektů můžou nastat následující jevy:

- V projektu bude velké množství změnových požadavků a implementací, které budou ovlivňovat náročnost a objem práce na projektu.
- Projekt je inovační nebo neexistuje zavedené know-how pro jeho přesné plánování.
- Zadavatel (zákazník) není schopen poskytnout plnou specifikaci požadavku a zadání, tudíž není možné odhadnout náklady na pracnost a dobu trvání.

Na některé z výše uvedených jevů je nutné reagovat, a proto pokud se projekt potýká s těmito jevy, je vhodné použít agilní řízení projektu.

Vývoj technologií a dnešní ekonomické prostředí vyžaduje využití agilních metodik. Tradiční přístupy nejsou dostatečně efektivní a flexibilní, jelikož je třeba rychlý vývoj a pružná adaptace na měnící se požadavky. Agilní metodiky vznikající od druhé poloviny 90. let staví na rychlém vývoji a zpětné vazbě zákazníka (Buchalceová, 2005).

Agilní filozofie řízení projektů byla oficiálně zavedena roku 2001, kdy skupina odborníků zabývajících se softwarovým inženýrstvím a vývojem softwaru publikovala „*Manifest agilního softwarového vývoje*“. Tento manifest identifikuje čtyři hlavní hodnoty, které preferuje v kontrastu s tradičními přístupy. Hodnoty v levé části tabulky 1 jsou preferovány před hodnotami v pravé části (Beck a kol., 2001).

Tabulka 1: Principiální hodnoty agilního manifestu

jednotlivci a interakce	procesy a nástroje
fungující software	vyčerpávající dokumentace
spolupráce se zákazníkem	vyjednávání o smlouvě
reagování na změny	dodržováním plánu

Zdroj: Agilní manifest, Beck a kol., 2001

Současné agilní přístupy navazují na Agilní manifest a dodržují několik základních principů při vývoji softwaru (Beck a kol., 2001):

- Časné a průběžné dodávky hodnotného softwaru.
- Vítané jsou změny v požadavcích i v průběhu vývoje za účelem zvýšení konkurenceschopnosti.
- Dodávka fungujícího software s preferencí kratších period.
- Spolupráce lidí z byznysu na projektu po celou dobu vývoje.
- Motivace lidí zapojených do projektu.
- Důraz na osobní komunikaci.
- Za hlavní měřítko pokroku se považuje fungující produkt.
- Agilní procesy podporují udržitelný rozvoj.
- Pozornost upřená na technickou výjimečnost a design.
- Jednoduchost je klíčová – maximalizace nevykonané práce.
- Nejlepší návrhy vycházejí ze samo-organizujících se týmů.
- Tým pravidelně pracuje na své efektivitě.

Je třeba poznamenat, že výše uvedené termíny nejsou v pojmech agilního vývoje „deklasovány“, pouze na ně není kladen takový důraz. V současném světě se využívá mnoha agilních metodik například: Extreme Programming (XP), Kanban, Crystal, Feature Driven Development (FDD), Dynamic Systems Development Method (DSDM), Adaptive Software Development (ASD), Lean Development (LD). V teoretické části práce, je věnována část metodice Scrum, jelikož se dle průzkumu Komuse a kol. (2020) jedná o jeden z nejpoužívanějších agilních přístupů vůbec.

2.4.3 Metodika Scrum

Scrum je agilní rámec pro inovativní řízení projektů, jenž napomáhá strukturovat a spravovat týmu svou práci skrze soubor hodnot, principů a praktik. Důraz je kladen na neustálé zlepšování přístupu týmu (Atlassian, 2023n). Veškerá práce a vývoj se dělí na menší segmenty, jež pak probíhají v menších časových úsecích, a to v řádech dnů až týdnů. Ty se opakují do splnění všech požadavků a zadání. Takový postup umožňuje lépe řídit rizika a pomáhá k lepšímu odhadu pracnosti.

Scrum využívá bezmála 84 % uživatelů agilních metodik, což jej činí nejpoužívanějším nástrojem agilních metodik. Mezi další hojně využívané patří Kanban, DevOps, Lean a Design Thinking (Komus a kol., 2020).

První zmínky o Scrumu se objevují v roce 1986, kdy byly popsány nové možnosti řízení týmů. Scrum se dokáže vyrovnat s měnícími se požadavky a vylepšit produktivitu procesu vývoje softwaru. Převážně se hodí pro malé týmy o několika členech (Adámek, 2017). V následujícím textu budou stručně popsány nejdůležitější aktivity Scrumu – sprint, plánovací schůzka, stand-up, sprint review a retrospektiva sprintu.

Sprint je jedním z hlavních prvků Scrumu. Je to iterace s přesnou dobou trvání, obvykle v řádech týdnů, ale někdy i dnů. Délka sprintu by neměla překračovat 1 měsíc. Sprints mají pevně stanovený začátek i konec a vzájemně na sebe navazují. Výsledkem takové iterace je hotový produkt nebo jeho část, kterou lze u klienta uvést do ostrého provozu (Doležal a kol., 2016).

Plánovací schůzka (Sprint planning) slouží k tomu, aby si celý Scrum tým stanovil cíle a rozsah pro nadcházející Sprintu. V backlogu jsou dále vybrány nejvhodnější úkoly pro přínos celého projektu a je také třeba dodržet pravidlo, že zadání sprintu nelze měnit po jeho dobu trvání (Doležal a kol., 2016).

Stand-up představuje každodenní poradou projektového týmu. Cílem této schůzky je si stanovit plán na následující den, identifikovat překážku v plnění úkolů a zkontrolovat práci, jež byla vykonána od poslední stand-up schůzky (Adámek, 2017). Během schůzky každý člen vývojového týmu odpovídá na tři otázky týkající se jejich práce (Schwaber and Sutherland, 2020):

- Co má splněno od poslední schůzky?
- Na čem bude pracovat?

- Vyskytují se nějaké překážky v jeho práci?

Sprint review se koná na samotném konci sprintu. Porada se zaměřuje na právě dokončené funkcionality dosažené prací vývojového týmu. Vývojový tým pak prezentuje svoji odvedenou práci za uplynulé období.

Retrospektiva sprintu dle Adámka (2017) slouží ke zpětné kontrole práce v rámci Scrum týmu. Pomáhá lépe naplánovat kroky, které pomohou zlepšit činnost v nadcházejícím sprintu.

Scrum role

Celý Scrum tým se skládá ze tří základních rolí. Tyto role jsou Product Owner, vývojový tým a Scrum (Schwaber and Sutherland, 2020).

Vlastník produktu (Product owner) zodpovídá za návratnost investice a za určení priorit úkolů a také vlastní produktový backlog. Mezi další úkoly vlastníka produktu patří poskytnutí specifikace produktu pro vývojový tým na základě komunikace, schvalování či zamítnutí finálního produktu či možnost být členem vývojového týmu a dodávat výstupy (Doležal a kol., 2016).

Projektový (vývojový) tým dle Schwabera a Sutherlanda (2020) představuje multifukční tým, uvnitř kterého jsou zastoupeny veškeré potřebné role pro minimalizaci prostojů mezi plněním úkolů. Na konci každého sprintu tým dodává hotovou část produktu, proto se jako celkové uskupení snaží uspokojit závazky a splnit cíle stanovené ve sprintu.

Scrum master je zcela specifická role, neřadí se mezi klasický management a ani vedoucí týmu, jelikož nemá rozhodovací pravomoci. Zodpovídá za průběh Scrum procesů, dohlíží na dodržení lhůty a udržuje Scrum nástroje. Vedle toho podporuje tým a měl by odstraňovat překážky a nepotřebné elementy vyvstávající z porad i z dotazů zákazníka, aby se tým mohl soustředit na potřebnou práci (Doležal a kol., 2016).

Scrum nástroje

Produktový backlog je prioritizovaný seznam veškerých možných návrhů nebo úprav (user stories) nutných k finalizaci produktu, zároveň se jedná o jedinou možnost, jak lze provést požadavek na změnu tohoto produktu. Dle Schwabera a Sutherlanda (2020) je produktový backlog „seznam všech vlastností, funkcí, požadavků, rozšíření a oprav chyb, které představují všechny změny, které budou provedeny v produktu v příštích vydáních. Položky produktového backlogu mají popis, prioritu a odhad.“

Sprint backlog je jakousi podmnožinou user story z product backlogu a definuje, na čem projektový tým bude pracovat v určitém sprintu. O sprint backlog se stará samotný vývojový tým. Jednoduše lze říci, že sprint backlog reflektuje aktuální stav sprintu (Šochová a Kunc, 2014).

Jakékoliv položky backlogu se definují jako tzv. user story. Taková user story popisuje funkcionalitu, kterou chceme přidat, nebo změnu již zavedených. User story specifikuje:

- kdo je příjemce hodnoty (*uživatel*)
- jak má produkt fungovat (*funkcionalita*)
- jaké očekávání je tím plněno (*přidaná hodnota*)

Daný popis user story musí být pochopitelný jak pro zadavatele, tak i pro vývojový tým, a proto musí splňovat následující charakteristiky (Wake, 2012):

- *Nezávislost* – uživatelské příběhy by měly být nezávislé jedna na druhé, tím se eliminuje možnost komplikace s plánováním, určením priorit a celkovým odhadem. V praxi je nezávislost většinou vyloučená konkrétním technologickým rozhraním vývoje, kdy poté může vznikat závislost v rámci technologického postupu nebo vzájemných rozhraní.
- *Srozumitelnost* – vysvětlení by mělo být výstižné a jasné tak, aby tým dokázal zcela pochopit požadavky zákazníka, proto by user story neměla obsahovat až příliš mnoho složitostí.
- *Přínos* – není přípustné, aby projektový tým pracoval na úkolech, které nejsou přínosné pro zákazníka a zbytečně implementoval do systému části, které nebudou mít pro koncové uživatele využití.

- Odhadnutelnost – tým by měl být schopný odhadnout pracovníci user story a dokázat určit správnou prioritu a plánování.
- Malý rozsah – user story by měla být dostatečně malá na to, aby si tým dokázal představit množství práce a dala by se bez problému zvládnout během jedné iterace.
- Otestovatelnost – musí se zcela určit sada kritérií, na základě, kterých lze objektivně potvrdit, že řešení je funkční a správné, jinak nelze nikdy definovat, zda je user story hotová.

Jednotlivé user story jsou tvořeny menšími úkoly a dílčími úkoly. Tyto menší úkoly není ideální zapisovat znovu do backlogu vzhledem k jejich podrobnosti a specifikaci úkolu, a také jelikož jsou již zastoupeny v backlogu svými mateřskými user story. Naopak je lepší mít tyto úkoly na očích, k čemuž slouží Scrum board.

Na jednotlivé story jsou navázané úkoly, které jsou zobrazené na Scrum boardu. Členové týmu s nimi mohou v rámci tabule libovolně manipulovat a tím úkoly přesouvat mezi jednotlivými stavy na Scrum board.

Burndown chart report znázorňuje, jak velkou část práce už projektový tým splnil v rámci jedné iterace a kolik zbývá do konce sprintu. Tyto reporty jsou nápomocné k dalšímu plánování sprintů, lze z nich snadno vyčíst, jak si tým vedl, zda potřebuje snížit pracovní vytížení, aby dosáhl svých závazků, nebo zda tým odvádí velmi dobrou práci a je schopný do další iterace zpracovat více požadavků (Rehkopf, 2022).

2.4.4 Srovnání agilních a tradičních přístupů v projektovém řízení

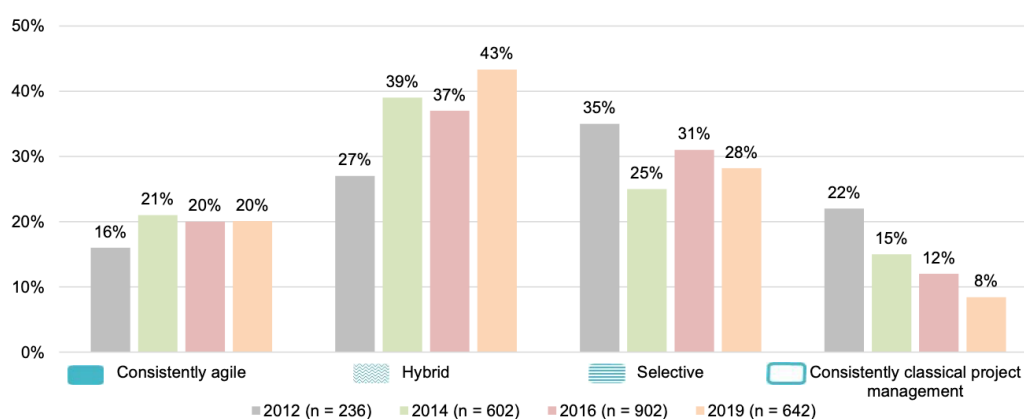
Agilní a tradiční metodiky představují dva různé pohledy na vývoj softwaru. Agilní projektové týmy jsou na rozdíl od těch tradičních tvořeny menším počtem lidí, avšak s důrazem na jejich kvalifikovanost a individualitu. Porovnání aspektů klasického a agilního přístupu řízení projektů je zaměřeno na klasický vodopádový model a agilní metodiku Scrum. Základní rozdílné aspekty obou přístupů podrobně zobrazuje tabulka níže.

Tabulka 2: Porovnání agilní a tradičních přístupů

Kategorie	Agilní přístupy	Tradiční přístupy
Produkt	Produkt se dodává po částech, jednotlivé dodané části jsou použitelné. Jeho finální podoba se mění na základě zpětné vazby zákazníka.	Produkt se začne využívat až v okamžiku předání zákazníkovi. Jeho podoba je předem daná.
Změny produktu	Jsou možné díky tomu, že produkt se dodává v iteracích.	Nejsou doporučovány. V případě jejich potřeby může být ohrožen termín nebo finanční plán.
Finanční plán	Není pevně stanoven, odvíjí dle rozsahu potřebné práce.	Projekt má předem stanové finance na základě plánování odhadu práce.
Zákazník v průběhu vývoje	Zákazník je zapojován do průběhu vývoje, sbírá se od něho zpětná vazba a má právo rozhodovat o změně nebo úpravě produktu.	Zákazník se zpravidla zapojuje na začátku, kde definuje produkt. Další zapojení zákazníka se očekává až při předání.
Vývojový tým	Malý tým, který zpracovává sadu úkolů v dané iteraci. Většinou s plochou strukturou vedení a vysokou mírou komunikace	Zpravidla velké týmy, kdy komunikace prochází spíše přes manažera. Ten je plně zodpovědný za chod projektu
Procesy vývoje	Během každé iterace vývoj opakuje proces od základních funkcí až po implementaci nejmenších detailů, postupy jsou flexibilní.	Vývoj se řeší v dlouhém cyklu, kde je dodáván fungující systém, postupy vývoje jsou předem určeny.

Zdroj: Vlastní tvorba

I přes rozdílné přístupy klasických a agilních metodik a protichůdné názory jejich tvůrců se často objevují tendence k jejich kombinaci. Taková kombinace přístupů je stále častější, což zobrazuje obrázek 5, jenž se zakládá na výsledcích průzkumu *Status Quo (Scaled) Agile* (Komus a kol., 2020). Největší zastoupení má hybridní model, který kombinuje agilní a klasické přístupy pro řízení projektů. Dále lze pozorovat že projekty se řídí i pomocí selektivního přístupu, tedy jen některé projekty v rámci organizace jsou řízeny agilně.



Obrázek 5: Zastoupení přístupů v projektovém řízení

Zdroj: Result Report: *Status Quo (Scaled) Agile*, Komus a kol. (2020)

3 Nástroje pro řízení projektů

Existuje mnoho různých nástrojů, které v rámci projektového řízení pomáhají plánovat, organizovat a řídit projekty. Autor se běžně setkává s následujícími:

- *Ganttovy diagramy* představují grafické zobrazení časového plánu projektu, které ukazuje, jak jednotlivé úkoly souvisí s časovým harmonogramem.
- *Tabulky úloh* pomáhají řadit úlohy projektu v chronologickém pořadí s informacemi o termínech, stavu, zodpovědné osobě a dokončení.
- *Mind mapy* vizualizují myšlenky a úkoly, což napomáhá organizovat a přehledně zobrazovat informace v rámci projektu.
- *Nástroje pro komunikaci*, do kterých patří e-maily, chatovací aplikace a videohovory. Tyto nástroje umožňují efektivní komunikaci mezi členy týmu a se zákazníky.
- *Nástroje pro správu rizik*, mezi něž patří matice rizik. Tyto matice pomáhají identifikovat a hodnotit potenciální rizika pro projekt a plánovat opatření k jejich řešení.
- *Nástroje pro sledování a hodnocení* jako jsou reporty a dashboardy na základě kterých lze sledovat a hodnotit výkon projektu a přijímat opatření k případnému zlepšení.

Některé z těchto nástrojů mohou být součástí softwarových řešení pro projektové řízení, které umožňují centralizovanou správu projektu a snadnou komunikaci mezi členy týmu.

3.1 Softwarové nástroje

Existuje mnoho softwarových nástrojů pro projektový management, které mohou být použity k řízení a plánování projektu, sledování progresu, komunikaci s týmem a řízení rizik. Autor dle své praxe mezi nejběžnější softwarové nástroje pro projektový management řadí:

- *Microsoft Project* je software umožňující týmům plánovat, sledovat a kontrolovat úkoly, rozpočet a harmonogram.
- *Trello* představuje webovou aplikaci pro týmovou spolupráci, která umožňuje plánovat, spravovat a sledovat úkoly a projekty pomocí Kanban nástěnek.
- *Jira* software se využívá pro správu projektů a řízení úloh. Týmy pomocí Jira mohou plánovat, monitorovat a řešit problémy.
- *Basecamp* je webová aplikace určena pro týmovou spolupráci, plánování, správu a sledování úkolů, komunikaci s týmem a sdílení souborů.
- *Slack* představuje komunikační nástroj pro týmy umožňující rychle komunikovat, sdílet soubory a spolupracovat na úkolech.
- *Monday* patří mezi cloudové platformy pro řízení projektů a úloh, pro plánování, sledování a správu úkolů a projektů.
- *Smartsheet* je rovněž cloudovou platformou pro řízení projektů a úloh, která umožňuje týmům plánovat, sledovat a spravovat úkoly a projekty pomocí tabulky.
- *Confluence* představuje nástroj pro týmovou spolupráci a projektové řízení, díky kterému lze v týmu sdílet informace, spolupracovat na dokumentech a plánovat úkoly.

V další kapitole se bakalářská práce zaměřuje na softwarové nástroje pro projektové řízení Jira a Trello od společnosti Atlassian, protože jedním z cílů je právě porovnání těchto dvou nástrojů pro řízení projektů od společnosti Atlassian.

Atlassian je australská společnost působící na trhu více než 20 let. Společnost se zabývá vývojem software, oblastí marketingu, managementu, správy IT služeb a rovněž působí v oblastech vyžadujících průběžný monitoring projektů (Atlassian, 2023a).

3.2 Jira

Jira je proprietární software pro správu projektů a sledování požadavků vyvinutý společností Atlassian v roce 2002 (Atlassian, 2019). Jira umožňuje plánovat, sledovat a řešit problémy v rámci projektů. Jak uvádí jeden ze zakladatelů společnosti v rozhovoru pro Business Insider (Weinberger, 2019), Jira je prvním produktem od Atlassian, zároveň však stále produktem nejvýznamnějším. Navíc se za dobu své existence stala standardem pro sledování pokroku týmů nebo průběhu odstraňování chyb. Jira je aktuálně k dispozici pouze ve dvou typech hostingu, a to cloud a data center/server (Atlassian, 2023b).

Od doby svého vzniku Jira prošla různými změnami a vylepšeními. V dnešní době Jira nereprezentuje pouze nástroj na správu projektů a sledování úkolů, ale umožňuje práci v několika oblastech od softwarového vývoje, projektového řízení a řady dalších podnikových procesů až po zákaznický servisní portál.

Jak prezentují webové stránky Atlassian (Atlassian, 2023c), Jira tvoří řadu produktů někdy označovaných jako Jira family, které se dělí na tři druhy produktů – Jira Work Management, Jira Software a Jira Service Management. Jednotlivé aplikace budou blíže analyzovány níže v této práci.

3.2.1 Úkol

Atlassian (2023d) uvádí, že úkol (issue) tvoří základní stavební kámen všech Jira aplikací a představuje činnost, kterou je třeba vykonat. Tato akce může představovat úkol, opravu chyb nebo jakýkoli jiný typ úkolu, se kterým se lze během práce na projektu setkat.

Všechny úkoly jsou definovány pomocí specifických polí, které shromažďují a zobrazují potřebné informace ke spolupráci nebo vyhotovení požadavku. Tyto sady polí lze libovolně mazat, upravovat nebo přidávat další, avšak některá pole jsou pevně stanovená systémem a nelze je měnit ani odebírat.

Každý typ úkolu disponuje definovanými specifickými poli a pracovním postupem. Odpovídající volba typu úkolu je zásadní pro správné řízení projektů a sledování výkonu týmu. Pro umožnění lepšího pochopení, jaké typy úloh jsou v projektu řešeny, musí být vybrány správné typy problémů pro daný projekt. Na základě toho lze lépe plánovat a prioritizovat práci a efektivněji řídit celkový postup projektu.

3.2.2 Projekt

Projekt v Jira si lze představit jako sbírku všech úkolů, které slouží k dosažení určitého cíle a ke sledování a reportování. Projekt je také jeden ze základních stavebních kamenů aplikace Jira, nelze totiž vytvořit samostatný požadavek, vždy musí být spojen s již existujícím projektem. Každý projekt v Jira má své specifické typy problémů, pracovní postup a různá schémata nastavení (Atlassian, 2023f). Z výše uvedeného vyplývá, že Jira projekt lze flexibilně konfigurovat a lze jej snadno přizpůsobit tak, aby odpovídal organizační struktuře, různým pracovním postupům nebo agilním metodikám.

Typy projektů

Při zakládání nového projektu v Jira připadá v úvahu jeden ze tří typů projektů. Každý z uvedených umožňuje veškeré základní operace v rámci práce s projektem nebo požadavkem, jedinečné funkce navržené pro specifické využití chystaného projektu jsou definovány pomocí typu projektu.

Software projekt se využívá převážně (ale ne nutně) pro vývoj software. Obsahuje specifické funkce pro vývoj software, mezi které patří správa verzí, speciální metriky a reporty nebo i integrace na vývojové systémy. Dále tento typ projektu umožňuje používat Scrum, Kanban nástěnky, nebo řídit projekt za pomoci sprintů. Díky tomu je určen pro týmy, které pracují agilně (Atlassian, 2023g).

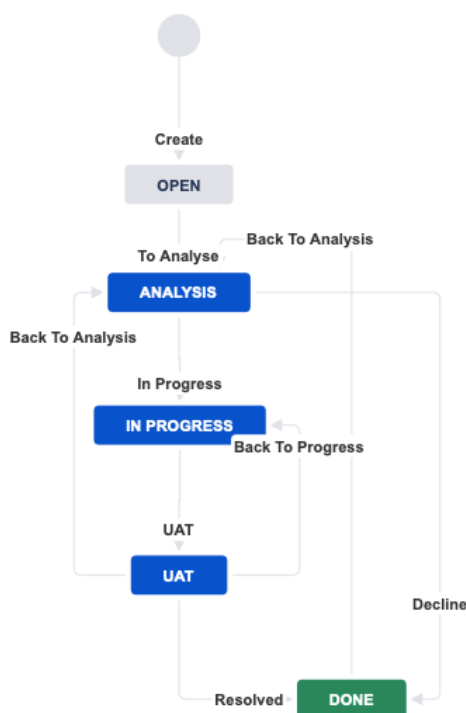
Business projekt je projekt, který může pomoci každému týmu řídit jeho práci. Atlassian (2023h) doporučuje využívání business projektů v týmech jako jsou marketing, HR, finance, právní oddělení, design aj. Pevně s ním pracují týmy, které využijí funkce pro řízení projektů a u kterých není na místě zatěžování se s nastavováním projektů a požadavků. Zároveň zde není požadován tak velký počet sledovacích metrik jako u softwarových projektů.

Servisní projekt – tento typ projektu se používá pro správu a řešení zákaznických požadavků a problémů. Tento typ je velmi populární mezi týmy IT, HR a helpdesků, které potřebují efektivně spravovat své služby a řešit problémy zákazníků. Mezi funkce projektu patří správa zákaznických požadavků, kompletní řešení pro správu incidentů, řešení problémů, změnové řízení a asset management. Hlavní bod představuje samoobslužný portál pro zákazníky, reportování a integrace s dalšími nástroji pro správu IT služeb (Atlassian, 2023ch).

3.2.3 Pracovní postup

Pracovní postup definuje postup řešení, tedy jakými stavy a akcemi úkol projde od jeho vytvoření až po jeho řešení či uzavření. Celý proces se nazývá pracovní postup (workflow).

Každý pracovní postup se skládá ze stavů, přechodů a rezoluce, kterými úkol během svého životního cyklu projde. Obvykle pracovní postup kopíruje pracovní procesy organizace (Atlassian, 2023i). Na obrázku níže je uveden příklad.



Obrázek 6: Ukázka pracovního postupu – Jira

Zdroj: Aplikace Jira Software – Unicorn Systems a.s, vlastní tvorba

Jak již bylo zmíněno v předchozím textu, celý pracovní postup se skládá ze dvou hlavních prvků:

Stavy (statuses) – stav představuje fázi, ve které se požadavek aktuálně nachází. Označení stavu je reprezentováno pomocí názvu a lze k němu přidat i popis. Jira umožňuje stavy seskupovat a rozlišovat pomocí barev. Jak Atlassian uvádí ve své dokumentaci (Atlassian, 2023j), existují 3 druhy kategorií pomocí barev, a to šedá (To Do), modrá (In Progress), zelená (Done).

Přechody (transitions) – reprezentují možnost přesunu mezi stavy v pracovním postupu za pomoci tlačítka na požadavku. Jelikož je přechod jednosměrný, umožňuje

přesun do nového stavu pouze jedním směrem. Pokud je potřeba se přesouvat mezi dvěma stavy tam a zpět, je nutno vytvořit dva přechody. Přechod také obsahuje množinu akcí, které se mají vykonat po přechodu do nového stavu, nebo různé druhy podmínek, které omezují, kdo a za jakých podmínek může přechod vykonat (Atlassian, 2023k).

Jira administrátor může vytvořit nový pracovní postup nebo upravit již existující. Také je pracovní postup zcela nastavitelný a umožňuje upravit tok práce a přizpůsobit projekt specifickým požadavkům.

3.2.4 JQL

Jira Query Language (JQL) je dotazovací jazyk speciálně vytvořen pro aplikaci Jira, Uživatelé za pomoci dotazů mohou efektivně vyhledávat a filtrovat požadavky v aplikaci (Atlassian, 2023l). Jednotlivé dotazy lze pak ukládat a znovu používat jako předvolené filtry. Ty lze použít například k vytvoření vlastních nástrojů pro sledování a zobrazování dat, pro automatizaci procesů nebo pro vytváření vlastních nástěnek a reportů. JQL obsahuje různá klíčová slova a operátory, tyto syntaxe jsou velice obdobné různým databázovým systémům a umožňují specifikovat vyhledávání, definovat podmínky pro výběr a řazení výsledků dotazu (Atlassian, 2023m). Možnost filtrování je vyobrazena na následujícím schématu.



The screenshot shows a Jira search interface. At the top, a green bar contains the JQL query: `issuetype not in ("CR / Servis", Servis) AND status not in (Dokončeno, Done) AND resolution = Unresolved AND assignee in (currentUser()) order by created DESC`. Below the query, it indicates "1-24 of 24" results. The main content is a table with columns for "T Summary", "Assignee", and "P Status".

T Summary	Assignee	P Status
<input type="checkbox"/> Načtení formuláře Request type trvá několik vteřin	Dominik Byrtus	NOVÝ POŽADAVE
<input type="checkbox"/> Nepřifazují se skupiny podle Group Assigner pluginu	Dominik Byrtus	NOVÝ POŽADAVE
<input type="checkbox"/> Problém se zobrazováním Permission Scheme	Dominik Byrtus	NOVÝ POŽADAVE
<input checked="" type="checkbox"/> Time in Status in Days	Dominik Byrtus	IN PROGRESS
<input type="checkbox"/> Test phase - Uni - on test environemnt	Dominik Byrtus	IN PROGRESS
<input type="checkbox"/> GCP-13 / CR - rozšíření funkcionalit	Dominik Byrtus	IN PROGRESS
<input type="checkbox"/> KP - Kapitalizace worklogs - Structure	Dominik Byrtus	NOVÝ POŽADAVE

Obrázek 7: Ukázka filtrování pomocí JQL- Jira
Zdroj: Aplikace Jira Software – Unicorn System a.s., vlastní tvorba

3.3 Trello

Trello je softwarová aplikace pro řízení projektů a organizace úkolů. Byla vydána v roce 2011. Po úspěšném startu, získávala rychle na počtech uživatelů, v roce 2017 Trello koupila společnost Atlassian (Trello, 2023a).

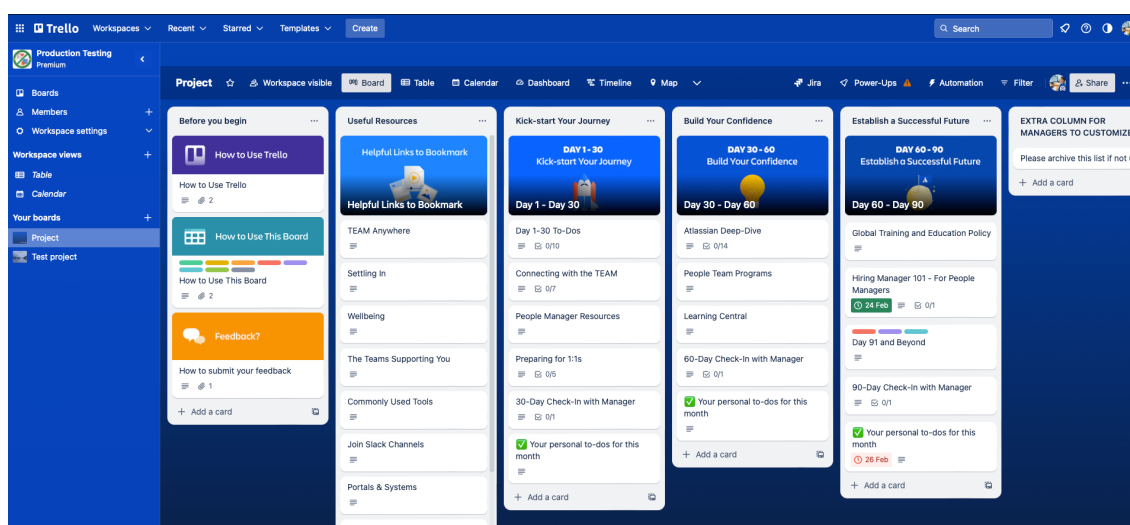
Trello si lze představit jako digitální tabuli, která používá paradigma, známé jako Kanban board. Koncept celé aplikace se pomyslně skládá ze tří klíčových základních komponent: nástěnky, sloupce a karty. Tyto komponenty umožňují koordinovat práci, zobrazovat její průběh a spolupracovat s týmem. Tento nástroj umožňuje uživatelům snadno vizualizovat úkoly, zobrazovat zodpovědnou osobu za daný úkol a priority jednotlivých úkolů aj. (Trello, 2023b).

Trello je velice oblíbený nástroj napříč všemi odděleními ve společnosti. Oblíbenost Trello tkví v jeho jednoduchosti a přehlednosti, protože umožňuje uživatelům snadno vizualizovat, co je potřeba vypracovat za úkoly, kdo je zodpovědný za každý úkol nebo jaké jsou priority jednotlivých úkolů, a to vše produktivním a organizovaným způsobem.

3.3.1 Pracovní prostor

Pracovní prostor (workspace) si lze představit jako speciální kontejner sdružující pracovní nástěnky, různé pohledy na nástěnky a kalendáře. Bez pracovního prostoru nelze vytvářet nástěnky a celkově pracovat v Trello (Trello, 2023c).

Nástěnka v aplikaci Trello obvykle reprezentuje projekt, pracovní postup nebo tým. Lze ji sdílet s ostatními uživateli, umožní sledování stavu a sledování práce členů týmu. Dále nástěnka nabízí různé pohledy, mezi něž patří tabulka karet, kalendář, kontrolní panel, časová osa a mapa (Trello, 2023d). Následující obrázek zachycuje pohled na nástěnku v Trello.

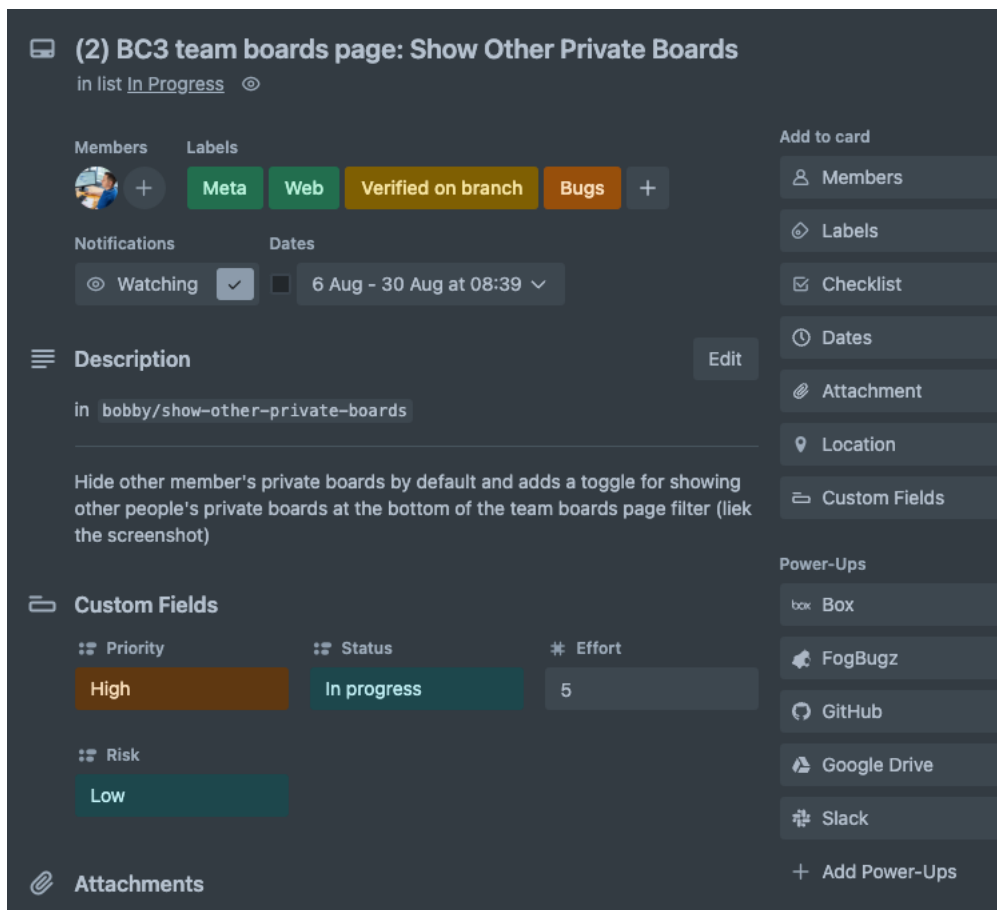


Obrázek 8: Pohled na nástěnku v Trello

Zdroj: Aplikace Trello – Autorova tvorba

Sloupec (list) slouží k vizuálnímu představení fází v projektu. Každý list nese svůj název podle toho, jaký krok v projektu označuje. Sloupec je nadále určen k uspořádání karet. Je možné zvolit si filtr zobrazení karet, karty lze pak na jednotlivé listy jednoduše přidávat nebo je přetahovat mezi různými listy. To může být využito k vytvoření projektového pracovního postupu. Díky tomu dochází k přesouvání karet postupnými kroky procesu od počátku až do konce. Neexistuje omezení pro počet sloupců na nástěnku a možnost uspořádání závisí pouze na uživateli (Trello, 2023d).

Karta je nejpodrobnějším prvkem na nástěnce. Karty zastupují jednotlivé úkoly a nápady. Kartu lze přizpůsobit podle možnosti a specifikací projektu tak, aby obsahovala užitečné informace. Počet karet na nástěnce není omezen (Trello, 2023d). Přehled karty znázorňuje následující schéma.



Obrázek 9: Přehled karty
Zdroj: Aplikace Trello – vlastní tvorba

3.4 Porovnání nástrojů Jira a Trello

Porovnání nástrojů pro správu projektů, jimiž jsou JIRA a Trello se provádí na základě zohlednění různých charakteristik těchto produktů. JIRA vyniká svým robustním systémem poskytujícím široké spektrum funkcí pro monitorování, správu a koordinaci projektů. Naopak Trello klade důraz na jednoduchý přístup k práci a vizuální prezentaci úkolů.

Autor na základě své zkušenosti s aplikacemi a teoretických poznatků z této práce sestavuje tabulku s porovnáním nástrojů Jira a Trello, přičemž autor klade důraz na podstatné aspekty daných nástrojů. Porovnávání zohledňuje nástroje v bodě, kdy uživatel aplikaci používá tzv. „out of the box“, tedy bez možných rozšíření a nástrojů, které se dají do aplikací stáhnout.

Tabulka 3: Porovnání nástrojů

Aspekt	Jira	Trello
Pracovní postupy	Možnost vytváření pracovních postupů dle zvoleného řízení.	Nelze vytvářet vlastní pracovní postupy.
Zaučovací proces uživatele.	Složitější prostředí a nastavení, uživateli trvá déle ovládnout nástroj.	Jednoduché prostředí, uživatel se rychle orientuje.
Agilní nástroje	Vlastní backlog, plánování sprintů, Scrum a Kanban nástěnka, speciální agilní reporty.	Pouze Kanban nástěnka.
Filtrování a vyhledávání	Použití dotazovacího jazyka JQL, vytváření vlastních filtrů na základě výsledků hledání.	Umožňuje pouze filtrovat na základě předdefinovaných hodnot.
Přehledové nástěnky a reporty	Velké množství nástrojů, přehledů, tabulek a reportů pro dané řešení.	Umožňuje jen základní zobrazení dle 4 atributů, absence reportů.
Integrace a vlastní řešení/nástroje	Jira nabízí širokou škálu integrací, rozšíření a také možnost zavedení vlastního řešení bez technického omezení.	Jen základní integrační nástroje, menší výběr rozšíření, možnost použití vlastního řešení v omezeném režimu.
Nastavení práv přístupu	Nabízí komplexní schémata pro zabezpečení a práva při práci na projektu, plně volitelné.	Umožňuje pouze 3 úrovně práv a nelze je měnit.
Cena/uživatelé	Jira je zdarma do 10 uživatelů, cena se poté odvíjí od počtu aktivních uživatelů.	Neomezené množství uživatelů v základní verzi, která je zadarmo.
Vhodnost pro projekty	Velké a komplexní projekty.	Jednoduché projekty, osobní nástěnka s úkoly.
Zaměření	Vhodné pro všechny velikosti týmu.	Vhodné pro menší týmy.

Zdroj: Vlastní tvorba

Jira je zaměřena na vývojové týmy, poskytuje široké možnosti pro úkolovou správu, sledování chyb a řízení projektů. Učit se s aplikací pracovat může být náročnější kvůli složitějšímu prostředí a nastavení, následně však tým může těžit z rozsáhlých agilních nástrojů pro využití agilních metodik jako Scrum a Kanban. Díky integračním nástrojům je ideální pro větší, složitější projekty s pokročilým sledováním. Naopak Trello patří mezi jednodušší nástroje s důrazem na vizuální úkolovou správu. Jeho intuitivní prostředí umožňuje rychlou orientaci uživatelů v základní verzi. Aplikace není zpoplatněna a je nezávislá na počtu uživatelů, což je ideální pro menší týmy či jednotlivce preferující jednoduchost a vizualizaci bez dodatečných provozních nákladů.

4 Praktická část

V praktické části práce se autor zaměřuje na aplikaci případové studie z vlastní praxe s vybranými softwarovými nástroji Jira a Trello. Oba nástroje jsou pro účely práce bez jakýchkoliv rozšíření zakoupených od třetích stran nebo vlastních softwarových přírůstků a úprav mimo těch nabízených v základu. Autor se na případových studiích snaží přiblížit použití těchto nástrojů v praxi na konkrétních IT/SW projektech a porovnat jejich nedostatky či rozdíly nebo naopak ukázat na pozitivní zjištění. Práce obsahuje tři případové studie. Případové studie jsou prováděny na třech v minulosti prováděných odlišných projektech ve společnosti Unicorn Systems a.s.

4.1 Případová studie A – softwarový vývoj za pomoci Scrum metodiky

Ve společnosti Unicorn Systems a.s., která vytváří a nabízí jedny z nejrozsáhlejších informačních systémů a IT řešení, působí tým zabývající se vývojem softwaru pro řízení energetických systémů (Unicorn 2023). Mezi klienty týmu patří společnosti provozující přenosové a distribuční sítě, obchodující s elektrickou energií a rovněž subjekty odpovědné za provoz podzemních zásobníků plynu.

Díky této rozmanitosti se vývojový tým setkává s různorodými projekty a požadavky, což vyžaduje agilní a flexibilní přístup k vývoji.

Tým se silně zaměřuje na potřeby zákazníka a pravidelně spolupracuje s interními i externími klienty. Tým klade důraz na získání zpětné vazby, jelikož mu pomáhá zajistit relevanci a užitečnost vyvíjeného systému.

4.1.1 Případová studie A – projekt agilního řízení vývoje pomocí metodiky Scrum

Pro použití řízení projektu za pomoci metodiky Scrum se vytvoří nový projekt v Jira. Při vytváření nového projektu je nutné zvolit typ projektu Software, tím se odemkne možnost využití agilních nástrojů. Tato zvolená konfigurace také automaticky vytvoří nové typy požadavků (uživatelský příběh, úkol, dílčí úkol, chyba), základní schémata a pracovní postupy. Takto vytvořený projekt v základní konfiguraci již umožňuje práci v rámci Scrum metodiky, pro správné použití případové studie však musí být rozšířeno projektové nastavení.

Projekt se rozšíří o nové typy požadavku – epic a změnový požadavek. Epic je druh požadavku, který představuje jeden velký uživatelský příběh, který může být dále rozdělen na menší uživatelské příběhy. Použití epicu je důležité, jelikož umožňuje lépe sledovat průběh práce na dodávaných funkcích, které utváří část vyvíjeného systému. To usnadní sledování stavu projektů a celkovou vizualizaci v přehledových tabulkách. Změnový požadavek se používá k návrhu/žádosti na úpravu systému, procesu nebo produktu. Taková žádost obsahuje požadavky na změnu funkcionality, přidání nových funkcí nebo jinou modifikaci, která přinese vylepšení řešení.

Dále jsou do projektu přidány nové pracovní postupy, jelikož základní přednastavené postupy nejsou dostačující pro správnou reprezentaci této studie a neodpovídají tomu, co se reálně na projektu používá.

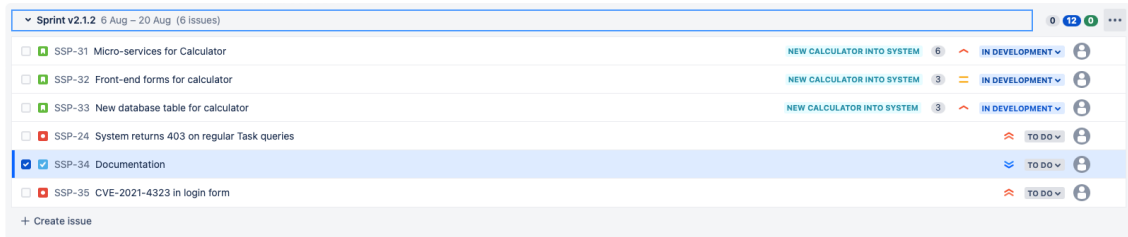
Pro většinu typů požadavků jako úkol, dílčí úkol a epic se používá pracovní postup, který není nikterak složitý a popisuje životní cyklus od založení požadavku až po jeho konec.

U změnového požadavku a uživatelského příběhu je pracovní postup rozsáhlejší z důvodu složitější práce na změně a nutnosti evidence stavu požadavku ve správném životním cyklu tak, aby práce s požadavkem byla přehledná.

Při tvorbě Scrum nástěnky se aplikace dotáže, jaký druh nástěnky má být použit. Na výběr je ze dvou možností – Scrum a Kanban. V projektu lze vytvořit více než jednu nástěnku. Všechny tyto nástěnky jsou konfigurovatelné a všechny spolu sdílí jediné – backlog. Tímto se docílí možnosti pro projektový tým vizualizovat aktuální sprint několika pohledy a také není vyloučená kombinace obou nástěnek – Scrum a Kanban.

Práce se sprintem

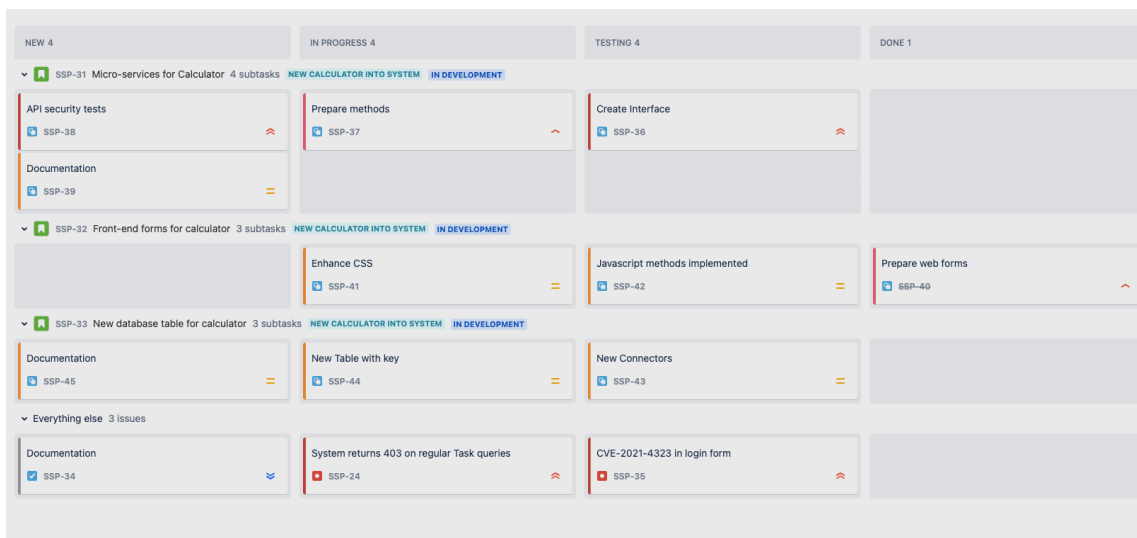
K tomu, aby na projektu mohl probíhat sprint, musí projekt obsahovat požadavky ve formě uživatelských příběhů, chyb k opravě systému a úkolů. Potom, co jsou vytvořeny, lze sestavit budoucí sprint. Sprint se plánuje v backlog projektu. Uvádí se název, doba trvání a popřípadě popis cíle. Hlavním obsahem sprintu jsou požadavky, o jejichž plnění v následujícím sprintu rozhodl projektový tým. Celá tato příprava probíhá během sprint planning schůzky, která patří do agilních ceremonií. Následující obrázek znázorňuje přípravu sprintu v backlog.



Obrázek 10: Příprava sprintu v backlogu

Zdroj: Aplikace Jira Software – vlastní tvorba

Jakmile je nadcházející sprint připraven, lze ho spustit. Na nástěnce se názvy sloupců mapují na názvy stavů dle pracovních postupů. K tomu dochází v nastavení nástěnky. Lze také změnit vizuální seřazení požadavků. Na následujícím obrázku jsou na nástěnce zobrazeny sdružené skupiny požadavků dle uživatelského příběhu.



Obrázek 11: Scrum board

Zdroj: Aplikace Jira Software – vlastní tvorba

V takto probíhajícím sprintu poté pracuje vývojový tým. Plní jednotlivé požadavky, které jsou reprezentovány na nástěnce. Přejít mezi sloupci je realizován buď přímo na nástěnce (uživatel pomocí drag & drop přetáhne požadavek do požadujícího sloupce) nebo lze měnit stav přímo v požadavku. Jira automaticky dle namapovaných sloupců podle přiřazených pracovních postupů požadavek na nástěnce přesune.

Důležitost reportů ve sprintu spočívá v poskytnutí strukturovaného a pravidelného způsobu sledování pokroku. Reporty také pomáhají udržovat sprint na správné cestě a podporují týmovou efektivitu. Jira nabízí širokou škálu reportů.

V případové studii se používá klasický sprint report, a to burndown chart spolu s přehledovou nástěnkou na sledování stavu požadavků v aktuálním sprintu.

Po uplynutí doby trvání sprintu nebo manuálního uzavření v backlogu se ukončí aktivita sprint. Po ukončení sprintu se Jira automaticky dotáže, co má provést s neukončenými požadavky. Na výběr jsou dvě možnosti – přesun do nového sprintu (pokud nový sprint nebyl vytvořen, tak Jira automaticky připraví nový sprint) nebo přesun do backlogu. Po ukončení se v záložce reportů aktualizují nové statistiky ohledně práce v ukončených sprintech. To pomůže v lepším plánování práce pro nadcházející sprinty. Týmy tak dokážou lépe plánovat svoji kapacitu a možnosti na plnění úkolů s každým ukončeným sprintem. Právě s těmito reporty se pojí schůzka ohledně retrospektivity uplynulého sprintu (sprint review), ve které se vyhodnocují získané reporty a statistiky.

4.1.2 Použití v Trello – případová studie A

Pro možnost využití případové studie v aplikaci Trello je třeba vytvořit pracovní prostor. Ten slouží k uchování a vytváření nástěnek. Každá nástěnka představuje projekt/úkol, na kterém tým v pracovním prostoru pracuje.

Při vytvoření nové nástěnky v pracovním prostoru nástěnka neobsahuje žádné sloupce, je tak potřebné tyto sloupce vytvořit. Nově vytvořené sloupce představují stavy karet (úkoly) na nástěnce, ale také nesmí chybět sloupec pro backlog sprintu. Dalším sloupcem je projektový backlog, v němž se uchovávají všechny karty, které nejsou součástí aktivního či dokončeného sprintu. Poslední vytvořený sloupec pak dále představuje ukončený sprint.

V aplikaci Trello jsou všechny možnosti nastavení karet a jejich úpravy na nástěnce generické, tudíž nelze vytvářet různé typy karet, tudíž jsou do projektu zavedeny šablony (templates). Šablony jsou předdefinované karty obsahující předem nastavené hodnoty pro usnadnění práce a umožní uživateli rychlejší orientaci na nástěnce nebo jednodušší vytvoření nových karet.

Takto připravené šablony jsou uloženy v novém sloupci s názvem Templates. Potom, co jsou všechny potřebné šablony vytvořeny, sloupec se zarchivuje, tím se docílí toho, že sloupec nezabírá místo během práce na nástěnce a uživatelům je stále umožněno vytvářet nové karty pomocí připravených šablon.

Práce se sprintem

K řádnému naplánování a průběhu sprintu musí být připravené karty s úkoly v backlogu. Uživatelé k založení karet mohou využívat šablony nebo si kartu založí ručně. Po naplnění backlogu úkoly začne sprint planning schůzka, kde se řeší úkoly na další (nebo první) iteraci sprintu. Určené karty s úkoly jsou tedy přesunuty do sloupce Sprint Backlog. Ve stejném sloupci je vytvořena nová karta, která obsahuje informace o aktuálním sprintu jako je název, doba trvání, případně popis cíle.

Trello umožňuje pracovat s vlastní přehledovou nástěnkou, ačkoliv výběr možnosti zobrazení dat je značně omezený. Výběr se omezuje pouze na sledování karet dle sloupce, uživatele, datumu splatnosti a zobrazení pomocí štítků.

Nejvhodnější zobrazení přehledu se skládá z obsazení karet dle řešitelů úkolů a přehledu o tom, kolik karet s úkoly se nachází v aktuálním stavu.

Poté, co je sprint ukončen (manuálně nebo datem ukončení trvání sprintu) jsou všechny karty, jež se nacházejí ve sloupci Done, přesunuty do sloupce s názvem ukončeného sprintu. Stejně tak je přesunuta karta s informací o sprintu ze sloupce Sprint Backlog do sloupce s názvem ukončeného sprintu. Všechny ostatní neukončené karty se přesunou do sloupce Backlog nebo do Sprint Backlog.

Trello nenabízí žádné vyhodnocovací nástroje, proto se retrospektiva uplynulého sprintu provádí manuálně pouze s informacemi, které lze získat z přehledu nebo ze sloupce, v němž jsou uzavřené karty.

4.1.3 Shrnutí případové studie

Tato případová studie se zabývala srovnáním nástrojů Jira a Trello v kontextu agilního řízení projektu za pomoci metodiky Scrum. Během analýzy bylo identifikováno několik klíčových zjištění.

Práce s produktovým backlogem byla kritickou fází procesu. V Jira byl backlog zobrazen na samostatné stránce umožňující ucelený přehled o aktuálních úkolech, nových sprintech a probíhajících sprintech. Naopak v Trello byl backlog pouze sloupcem úkolů, což mohlo vést k nepřehlednosti a omezenému filtrování. Rankování úkolů bylo v obou nástrojích podobné, jednalo se o posouvání úkolů s vyšší prioritou na začátek sloupce.

Plánování úkolů bylo proveditelné v obou aplikacích, ale Trello mělo obtíže s přílišným množstvím karet na nástěnce, což snižovalo přehlednost. Naopak Jira

oddělovala backlog od nástěnky, čímž eliminovala tento problém. Vytváření sprintů bylo srovnatelné, přičemž Jira nabízela více možností nastavení a různé pohledy.

Během sprintu byla práce v nástrojích podobná, ale Jira měla výhodu v přesném mapování sloupců na stavy pracovního postupu. V Trello se musely manuálně měnit stavy úkolů, což může být zdlouhavé. Ukončení sprintu bylo efektivnější v Jira, která automaticky přesunula nevyřešené úkoly do nového sprintu a ukládala užitečné údaje do historie sprintů. Trello toto nemělo a uživatelé museli úkoly ručně přesouvat.

Největší nedostatek Trello byl v reportovacích nástrojích. Chyběla možnost vytvořit rozsáhlé reporty jako v Jira, což omezovalo možnosti sledování a hodnocení projektu. Trello mělo sice přehledovou nástěnku, ale byla nedostatečná pro agilní řízení.

Celkově lze říci, že Jira je specializovaným nástrojem pro agilní řízení Scrum projektů s bohatými nástroji pro práci a reportování. Naopak Trello je spíše vizuálním nástrojem vhodným pro menší týmy a projekty. Pro komplexní projekty s vyšším objemem úkolů a potřebou podrobných reportů je Jira lepší volbou. Vhodnost Trello spočívá ve využití pro začínající týmy v agilním prostředí díky jednoduchosti a vizualizaci.

4.2 Případová studie B – řízení IT projektů za pomoci vodopádové metodiky

Další případová studie se zabývá migracemi do cloudu (dodávání IT řešení v podobě migrace systému), které provádí projektový tým. Každá migrace se rozděluje do několika fází. První fáze, od které se vše odvíjí, spočívá ve vyjednávání se zákazníkem před tím, než je uzavřen obchod. Dochází zde ke komunikaci požadavků klientem ohledně náročnosti a obsahu migrace. Na to pak navazují další aktivity zahrnující analýzu celkového projektu pro zákazníka spolu s technickým plánem řešení. Dále projektový tým provádí testovací migraci, na kterou navazuje poslední fáze, při níž probíhá produkční (ostrá) migrace, post-migrační podpora a akceptace zhotovené migrace klientem. Případová studie pohlíží na proces v minulosti proběhlé migrace požadované klientem.

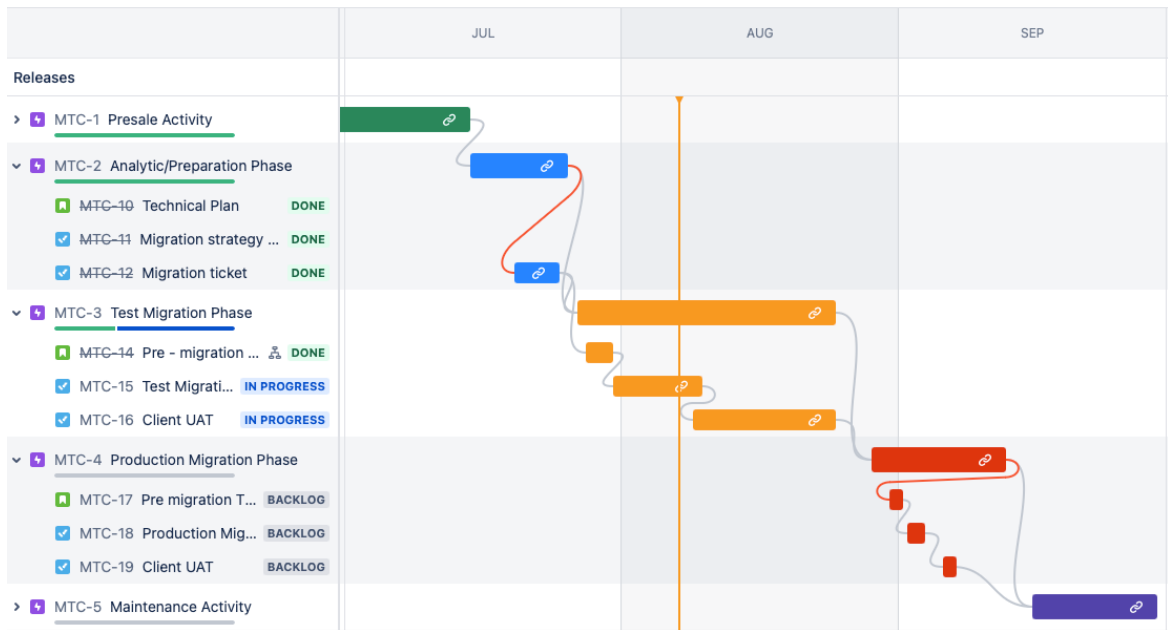
4.2.1 Použití v Jira – případová studie B

Celý průběh migrace se vždy pojí s konkrétním projektem. Tento projekt v aplikaci Jira nese název podle názvu společnosti klienta. Pokud už takový projekt existuje, znamená to, že projektový tým již s klientem v minulosti spolupracoval, a lze tak pokračovat v již vytvořeném projektu. V aplikaci Jira se vybere druh projektu software proto, aby mohl tým využívat Ganttovy diagramy pro grafický přehled fází na projektu.

V projektu se pracuje s několika druhy požadavků. Epic je druh požadavku, jenž reprezentuje určitou fázi projektu. Každý epic pod sebou pak sdružuje další požadavky, které jsou spjaté s danou fází projektu. Dále se na projektu využívá typ požadavku úkol, dílčí úkol a uživatelský příběh. Pracovní postupy nejsou na tomto projektu komplikované, a to z důvodu přímocárnosti úkolů plněných v rámci průběhů celé migrace.

Je vytvořeno několik fází, jimiž projekt prochází od jeho začátku až po jeho ukončení. Tyto fáze jsou předem určené u každého projektu, který se zabývá migrací. Každá fáze je reprezentována požadavkem typu epic. Ten má na sobě připojené úkoly spjaté s danou fází. Každá fáze má odhadovaný čas k realizaci a s klientem smlouvanou dobu trvání. Doba trvání určuje, jak dlouho fáze potrvá a projektový manažer na základě této doby může lépe plánovat a řídit kapacity svého týmu podle typu fáze. Např. v analytické fázi se více zapojuje role architekta řešení. Naopak v migrační fázi se více zapojuje role technického konzultanta, který migraci provádí. Každá jednotlivá fáze nebo úkol musí mít správně vyplněné údaje o začátku nebo plánovaném konci, což hraje důležitou roli v budoucích plánováních a grafických přehledech.

Jira software umožňuje použití časových os (timeline). Jedná se o grafické znázornění časové osy usnadňující ilustraci průběhu, délky trvání a vzájemných souvislostí úkolů v projektu. Jednoduše lze vykreslit dobu, kterou úkoly potřebují k dokončení, a predikovat, kdy budou jednotlivé události uskutečněny (Atlassian, 2023e). Časovou osu vyobrazuje následující schéma.



Obrázek 12: Zobrazení časové osy v Jira

Zdroj: Aplikace Jira Software – vlastní tvorba

Na obrázku jsou vyobrazeny jednotlivé fáze projektu. Fáze jsou propojeny pomocí linků (propojení požadavků za pomoci názvu). Tím se na mapě vytvoří grafická prezentace spojníc poukazujících na to, jak spolu souvisí jednotlivé fáze nebo jak na sebe navazují jednotlivé úkoly fáze.

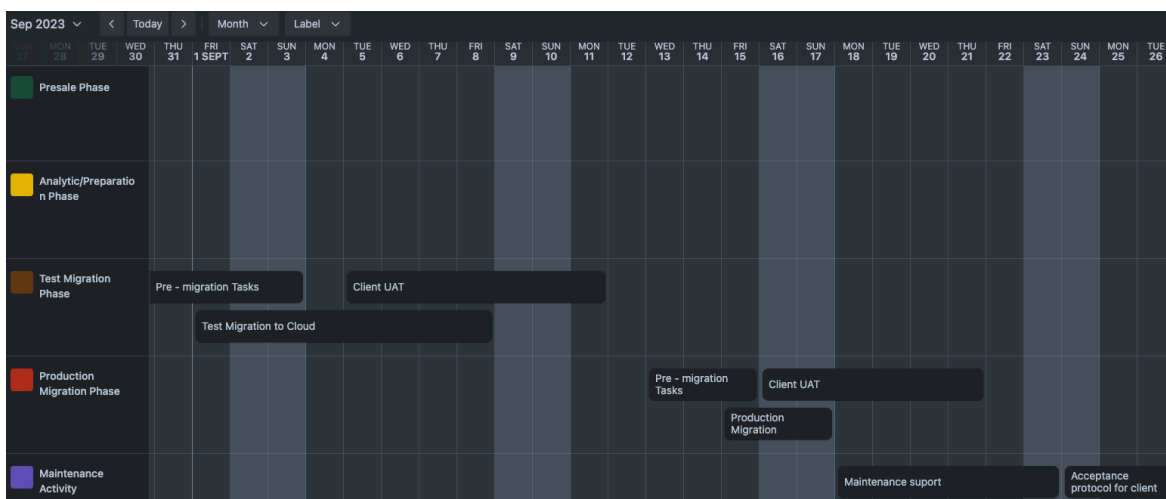
Pro lepší orientaci v řízení projektu využívá projektový manažer dva náhledy. Prvním náhledem je kalendář, ve kterém jsou zaznamenány daty splnění úkolů, kdy lze vidět lhůtu plnění jednotlivých úkolů (zda se blíží ke konci), a mohly tak být podniknuty kroky k jejich dokončení, nebo prodloužení termínu. Dalším náhledem je celková odpracovaná doba na fázích projektu. Tento náhled umožní, aby mohl projektový manažer nahlížet na vykázanou dobu. V předstihu tak může řešit např. navýšení pracovní síly nebo analyzovat, proč dříve stanovená pracovní síla není dodržena.

4.2.2 Použití v Trello

V pracovním prostoru se vytvoří nová nástěnka, jež obsahuje název klienta a stanovený úkol. V tomto případě se jedná o migraci. Jednotlivé fáze projektu jsou v Trello reprezentovány štítkem na kartě. Nově vytvořená nástěnka zatím neobsahuje žádné sloupce, ty musí být následně vytvořeny. Tyto nově vytvořené sloupce reprezentují stav karet (úkolů) na nástěnce. Konkrétně se jedná o sloupce Backlog, New, In Progress a Done. Backlog je určen pro uchování karet, které nejsou zatím aktivně řešené během fáze, ostatní sloupce pak slouží k reprezentaci stavů.

Na nástěnce se podobně jako v předchozí případové studii opět využije možnost šablony s předem definovanými hodnotami, které jsou určeny k usnadnění práce a umožňují uživatelům rychlejší orientaci na nástěnce. Také usnadňují proces vytváření nových karet a automatické rozdělení do jednotlivých fází pomocí štítků. Šablony jsou opět uloženy v nově vytvořeném sloupci Templates. Tento sloupec se archivuje stejně jako u předchozí případové studie.

Při startu migrace projektový manažer vytvoří předem dané úkoly, jež jsou spojené právě s migrační aktivitou. Nové karty se vytvoří pomocí šablon a tím dojde k automatickému rozlišení, do jaké fáze úkol spadá. Projektový tým pak musí aktualizovat časové údaje, a to začátek a plánovaný konec v kartách tak, aby odpovídaly naplánovaným aktivitám s klientem. To vše se poté promítne na časové ose projektu. V Trello lze vytvářet časové osy karet. Výběr možností osy však není rozsáhlý, pro vizualizaci případové studie tak poslouží zobrazení podle štítků.



Obrázek 13: Zobrazení časové osy v Trello

Zdroj: Aplikace Jira Software – vlastní tvorba

Jak bylo avizováno v předchozím textu, štítky reprezentují jednotlivé fáze projektu, a tudíž se využívá pohled na časovou osu dle štítků. Na časové ose lze pak sledovat, jak jsou časově naplánované jednotlivé úkoly v rámci celé migrace.

4.2.3 Shrnutí případové studie

V rámci případové studie zaměřené na proces migrace do cloudu a jeho řízení pomocí nástrojů Jira a Trello bylo získáno několik klíčových poznatků. Prvotní fáze projektu

spočívá ve vyjednávání se zákazníkem, kde dochází ke komunikaci ohledně náročnosti a obsahu migrace, což tvoří stavební kámen pro celý projekt.

Ve fázi vytváření byl již evidentní rozdíl v přístupu mezi oběma nástroji. Jira umožňuje vytvoření komplexního plánu pomocí typu požadavku epic, ten reprezentoval jednotlivé fáze. Úkoly zařazené do uvedených fází se pak propojovaly s požadavkem epic, což zajišťovalo strukturovaný vzhled jednotlivých fází. Naopak Trello neumí jednotlivé úkoly strukturovat. Nejbližší možností, alespoň na vizuální úrovni, bylo označení pomocí barevných štítků. Každý štítek tak označoval jednotlivou fázi.

Další rozdíl spočíval ve využití pruhového diagramu časové osy pro grafické znázornění jednotlivých činností v rámci plánu. Jira umožňuje použití vlastního diagramu označované časové osy (*timeline*). Tento diagram vizuálně poukazuje na to, kdy jaká fáze končí, jaké úkoly spolu souvisí a jaké jsou milníky jednotlivých fází. Trello, má obdobný nástroj nesoucí stejný název, avšak disponující pouze triviálními daty. Z diagramu není jednoznačné, jak spolu fáze souvisí, pro řízení či orientaci v harmonogramu však postačuje.

Dalším nedostatkem Trello jsou omezené možnosti kontrolních nástěnek. Manažer projektu na základě přehledu nedokáže jednoduše říci, kolik práce se vykonalo na jednotlivé fázi/úkolů v porovnání se schváleným plánem.

Závěrem lze konstatovat, že nástroje Jira i Trello mohou úspěšně podporovat řízení projektu migrace do cloudu, a to navzdory menším rozdílům mezi nimi v rámci této specifické případové studie. Oba nástroje nabízejí funkcionality pro správu projektu a sledování postupu, což jsou klíčové aspekty při řízení tohoto typu projektu.

4.3 Případová studie C – Projekt pro řízení operačních rizik

Případová studie se zabývá aplikací procesů pro projekt IT řízení operačních rizik. Popisuje celý proces řízení správy operačních rizik od samotného vzniku až po jeho úplné zpracování. Tímto způsobem významně přispívá k efektivnímu zvládnutí interních procesů následujícího charakteru.

Zadání operačního rizika – proces začíná zjištěním nového operačního rizika, jehož detaily jsou zaznamenány a identifikovány v rámci systému.

Zpracování návrhu operačního rizika – detaily o nově identifikovaném riziku jsou pečlivě analyzovány a zpracovány. Tento krok zahrnuje podrobný popis rizika a jeho potenciálních dopadů.

Získání odborného názoru – pro zajištění komplexního zhodnocení je získán odborný názor relevantních osob, které přispívají k hloubkovému porozumění riziku a jeho souvislostem.

Zjištění finančního dopadu – kvantifikace možných finančních dopadů, které by mohlo riziko přinést, patří mezi důležité prvky v procesu rozhodování a alokace zdrojů.

Definice nápravného opatření – pokud je to nezbytné, jsou navržena vhodná nápravná opatření, která by měla minimalizovat nebo eliminovat riziko.

Řešení nápravného opatření – implementace navržených nápravných opatření je pečlivě sledována a dokumentována, aby se zajistilo účinné a včasné provádění.

Monitoring stavu operačního rizika – po zavedení nápravných opatření se provádí pravidelný monitoring stavu rizika, aby se zajistilo, že zavedená opatření fungují a riziko je efektivně řízeno.

Tento projekt reprezentuje klíčový nástroj pro transparentní a systematické řízení operačních rizik.

4.3.1 Použití v Jira

V Jira je vytvořen nový projekt určený pro řízení rizik. V tomto případě se jedná o projekt typu Business. Pro potřeby řízení se využívá několik typů požadavků, které jsou vytvořeny na míru tohoto projektu. Mezi nové požadavky se řadí operační riziko, odborný názor, finanční ocenění, nápravná akce a monitoring.

Pro celý proces řízení operačních rizik je důležité vytvořit kvalitní pracovní postup v Jira. K požadavku na operační riziko je vytvořen nový pracovní postup. Postup umožňuje sledovat stav životního cyklu požadavku. Základní pracovní stavy obsahují stavy reflektující události operačních rizik jako „navrhuto“, „připraveno“ „k vyhodnocení“, „akceptováno“ (riziko se akceptuje), „vyřešeno“ (tento stav značí, že riziko není relevantní) a „mitigace“. Dalším druhem požadavku, pro který se vyrobí nový pracovní postup je nápravná akce. Tento typ disponuje obdobnými stavy jako u požadavku operační riziko, pouze se u stavu „akceptováno“ (mitigováno na přijatelnou úroveň) a vyřešeno (plně mitigováno) liší významy. U ostatních typů požadavku lze využít výchozí pracovní postup.

Celý proces začíná v momentě založení nového druhu požadavku Jira – operační riziko. Zde přichází do procesu další uživatel, správce operačního rizika, ten má za úkol vytvářet nové požadavky, řídit jejich stav a vytvářet propojení pomocí aplikačních linků. Správce vytváří nové úkoly proto, aby získal stanovisko v podobě odborného názoru na riziko. Poté se obrací s novým úkolem na finančního analytika pro vypracování finančního dopadu rizika.

Poté co správce získá veškeré podklady, dochází k hodnotícímu procesu požadavku. Zde se rozhodne, zda je riziko třeba nadále řešit. Dojde tedy k jednomu ze tří stavů – riziko buď není relevantní, nebo je akceptováno či mitigováno. V případě mitigace se zakládá nový požadavek, jenž se propojí pomocí aplikačního linku. Pokud nebyla zvolena mitigace, požadavek se uzavírá.

Pokud je vytvořen nový požadavek typu nápravná akce, řeší různé projektové role návrh nápravného opatření. Jakmile se návrh akceptuje, řeší se implementace daného návrhu, případně eskalace termínu manažerem pomocí nového požadavku. Výsledné řešení spočívá v úplné nebo částečné mitigaci rizika.

Celý proces, od vytvoření až po ukončení je reprezentován v Jira za pomoci stavů. Jelikož stavy požadavků se hojně využívají v přehledových nástěnkách, tyto stavy musí být udržovány v aktuálním stavu. V opačném případě hrozí, že dojde ke zneefektivnění pracovního postupu a možnému chaosu ohledně stavu požadavků.

Přehledové nástěnky slouží manažerovi ke sledování potřebných data, zároveň však i jednotlivé další role využívají tuto nástěnku pro zobrazení svých aktuálně řešených požadavků. V rámci reportingu se používá jednoduchý přehled aktivních požadavků rozdělených dle typů a aktuálních stavů, dále přehledová nástěnka obsahuje možnost zobrazení aktuálně řešených požadavků přihlášeného uživatele.

4.3.2 Použití v Trello

Prvním krokem se v Trello vytvoří nová nástěnka. Ta bude sloužit jako centrální místo pro správu všech aktivit souvisejících s řízením operačních rizik. Nově vytvořené sloupce reprezentují různé fáze procesu tak, aby bylo vizuálně jasné, kde se nachází karta v každém jednotlivém případě rizika. Sloupce obsahují názvy „navrhuto“, „připraveno“, „k vyhodnocení“, „akceptováno“, „vyřešeno“, „mitigace“, „v řešení“ a „v čekání“.

Pro každé operační riziko se vytvoří samostatná karta. Název karty by měl jasně identifikovat riziko, například "možný výpadek systému". V popisu karty může být uvedeno více informací o riziku, včetně jeho dopadu a priority. Karta obsahuje seznam kontrolních bodů, který obsahuje kroky, jež je třeba provést v rámci řízení tohoto rizika. Tento seznam může zahrnovat položky jako „vytvoření nápravného plánu“, „vypracování finančního dopadu“ a další.

Během průběhu procesu se sleduje každá karta a její checklisty. Jakmile je daný krok proveden označuje se jako hotový. Jakmile je celá karta hotová, může být přesunuta na další seznam. Karty jsou přesouvány podle fáze procesu.

V Trello jsou využité různé vizuální prvky jako barevné značky nebo prioritní štítky, aby mohla být rychle identifikována důležitá rizika nebo aktuální stav.

4.3.3 Shrnutí případové studie C

Proces řízení operačních rizik je klíčovým prvkem pro udržení stability a efektivity v organizaci. S využitím moderních nástrojů pro projektový management, mezi něž patří Jira a Trello, lze tento proces efektivně strukturovat, monitorovat a spravovat. Oba nástroje mají své vlastní výhody a vhodné scénáře použití pro projekt této případové studie.

Oba zmiňované nástroje jsou vhodné pro použití v projektu této studie, Jira proces více strukturalizuje do nových úkolů. Jednotlivé úkoly jsou pak propojené společným aplikačním linkem. Mezi výhody takové struktury patří možnost úkoly rozprostřít a přiřazovat více řešitelům zároveň. Zpracovávané informace (stanovisko, finanční dopad apod.) jsou pak udržovány/řešeny v příslušných úkolech, a nevzniká tak jeden velký úkol, ve kterém jsou řešeny všechny procesy operačního rizika. Naopak Trello propojení a vytváření takové struktury neumožňuje, proto všechny úkoly a procesy jsou realizovány pomocí jedné karty, což je z hlediska přehledů na nástěnce přívětivé. Řešení procesů v rámci operačních rizik se ukládá do jedné karty, kdy s množstvím obsahu může docházet k nepřehlednosti.

Mezi další problémy Trello opět patří nedostatek nástrojů k reportingu stavu úkolu. V tomto případě je na tom opět lépe Jira.

Při výběru mezi Jira a Trello lze říci, že Trello může aplikaci Jira plně zastoupit jako nástroj pro řízení projektu operačních rizik. To vše ale závisí na potřebách a komplexitě dané organizace.

5 Shrnutí výsledků, závěr a doporučení

Tato bakalářská práce komplexně rozvíjí problematiku projektového řízení. Dochází v ní k porovnání agilních a tradičních přístupů k řízení projektu a práce se také věnuje softwarovým nástrojům pro projektové řízení. Práce si kládla za cíl zejména poskytnout ucelený pohled na zmíněnou problematiku a aplikovat problematiku vybraných nástrojů projektového řízení (Jira a Trello) do praxe pomocí případových studií. Autor chce také odpovědět na otázku, jak Trello jakožto jednodušší nástroj, který však není zpoplatněn, obstojí při použití na vybraných projektech v konkurenci s Jira.

Obsah práce se dělí na tři hlavní části, první teoretická část přibližuje problematiku projektového řízení, definuje projekt a také se zaměřuje na řízení rizik projektu. Následně autor seznamuje čtenáře se standardy projektového managementu (PRINCE2, PMBOK Guide a IMPA) a agilními a vodopádovými neboli tradičními přístupy v projektovém řízení a provádí jejich srovnání. Další část práce přibližuje softwarové nástroje pro projektové řízení Jira a Trello, jež jsou klíčovými elementy této práce. Následuje porovnání těchto dvou nástrojů na teoretické rovině, na což plynule navazuje praktická část, jež tyto nástroje analyzuje a srovnává jejich nejdůležitější charakteristiky ve třech případových studiích zabývajících se třemi odlišnými druhy projektu v IT a SW oblasti.

Nejdůležitější praktická část poskytuje náhled na řešení třech různých projektů – projekt pro agilní řízení vývoje pomocí metodiky Scrum, řízení migrace do cloudu pomocí vodopádové metodiky a projekt řízení operačních rizik. Všechny tři projekty jsou využívají pro účely této práce řešení v platformě Jira i Trello. Autor nejdříve popisuje skutečnosti spojené s řešením případové studie a poté vždy kriticky zhodnocuje, jakými výhodami či nevýhodami použití Jira a Trello v konkrétním případě disponuje, zda je aplikace pro řízení daného projektu vhodná a zda by Trello jakožto jednodušší nástroj pro řízení onoho projektu postačoval.

Před zahájením každé studie byla v Jira i v Trello provedena počáteční konfigurace a nastavení umožňující efektivní práci pro různý typ projektu. Nastavení nového projektu v Jira je náročnější a obsahuje mnoho možností konfigurace, zatímco Trello nabízelo intuitivnější a jednodušší rozhraní, což ale souvisí s menším výběrem funkcionalit oproti Jira.

Výsledkem případových studií je popis a zhodnocení toho, jak oba nástroje fungují na jednotlivých typech projektů. Aplikace Jira byla již v úvodu práce spojena s očekáváním komplexnějšího, sofistikovanějšího a rozmanitějšího nástroje v porovnání s Trello, což se v případových studiích potvrdilo. Jedna z výzkumných otázek se však zabývala tím, jak Trello na konkrétních případových studiích v konkurenci s aplikací Jira obstojí.

Práce s Trello byla nejproblémovější převážně v případové studii A, v projektu agilního řízení vývoje pomocí metodiky Scrum. Při využití jedné nástěnky pro řízení sprintů docházelo k jejímu zahlcení úkoly. Na jedné nástěnce se totiž vyskytuje backlog s úkoly a zároveň samotné sloupce pro řízení stavu ve sprintu. Naopak Jira pro tyto účely disponuje záložkami, tudíž nedochází k zahlcení nástěnky. Trello rovněž na rozdíl od Jira nedisponuje agilními reporty, což znemožňuje sledování a vyhodnocování sprintu. V tomto případě by autor určitě zvolil nástroj Jira.

V případě případové studie B při migraci do cloudu je Trello pro základní řízení dostačující, avšak Jira pochopitelně disponuje daleko rozsáhlejší možností vizuálního zpracování. Trello obdobně jako v případové studii A neobstálo zejména v oblasti manažerských reportů pro účely monitoringu, které jsou zde velmi omezené.

V projektu zabývajícím se IT řízením operačních rizik, tedy v případové studii C, si Trello vedle již zmíněných omezených možností reportů neumí poradit se strukturalizací úkolů, navzdory tomu si však v tomto případě aplikace vedla velmi obstojně. Autor by dokonce na řízení projektu tohoto typu zvolil aplikaci Trello před aplikací Jira, jelikož v tomto případě lze profitovat z jednoduchosti aplikace Trello.

Obecně lze říci, že platforma Trello v případových studiích obstála, v případové studii C si dle názoru autora vedla dokonce lépe. Pro různé typy projektů musí být sledovány odlišné charakteristiky a na základě toho se lze rozhodnout, jaký nástroj použít. V každém případě by volba nástroje měla záviset na velikosti týmu, rozsahu projektu a preferovaném pracovním stylu. Jira je vhodná pro rozsáhlejší projekty s komplexním řízením a reportováním, zatímco Trello je ideální pro menší týmy a projekty, kde lze profitovat z jednoduššího a intuitivnějšího prostředí a počáteční vyšší přehlednosti. Hlavní výhoda Trello tkví v tom, že je zdarma, což je pro společnosti velkou motivací pro výběr této aplikace pro řízení projektů. Trello však neuspěje při požadovaném rozšíření funkcionalit zejména u větších týmů a projektů, kde začne být vyšší množství úkolů zahlcující.

Lze konstatovat, že oba nástroje mohou úspěšně podporovat řízení projektu a volba by měla být provedena s ohledem na potřeby projektu a týmu. Nicméně každá společnost má své specifické požadavky a zavedené postupy, proto musí být k řízení přistupováno velmi individuálně. Zároveň je tak pochopitelné, že společnosti velmi často přistupují k hybridnímu modelu řízení (kombinace přístupů).

Tato práce splnila cíle vytyčené v úvodu práce. Práce přináší zajímavé poznatky z oblasti projektového řízení, je přínosem pro autora, ale i pro praxi projektového řízení. Poskytuje ucelený pohled na problematiku, kterou zasazuje do praxe a zároveň odpovídá na výzkumnou otázku. Práce může být podkladem pro rozhodování o volbě vhodného nástroje řízení, lze ji použít jako pomocnou dokumentaci pro technické konzultanty. Autor díky nově nabytým poznatkům při tvorbě této práce získal nabídku podílet se na budoucím projektu spočívajícím v implementaci aplikace Trello jako nástroje pro projektové řízení v konkrétní společnosti.

Na závěr lze dodat, že poznatky z práce mohou být využity pro návazné zkoumání a analýzy, případné srovnání s dalšími aplikacemi. Pro další zkoumání lze např. zapojit integrace, nástroje třetích stran nebo použít vlastní software v rámci Jira v kontextu projektového řízení.

6 Seznam použité literatury

- [1] ADÁMEK, P. (2017). *Projektový management: Distanční studijní opora*. Karviná: Slezská Univerzita v Opavě: Obchodně podnikatelská fakulta v Karviné. ISBN 978-80-7510-272-0.
- [2] Atlassian (2019). Atlassian Shareholder Letter Q2 FY19 [online]. 2019 [cit. 2023-07-03]. Dostupné z: https://s28.q4cdn.com/541786762/files/doc_financials/2019/q2/TEAM-Q2-2019-shareholder-letter_FINAL.pdf
- [3] Atlassian (2023a). Sydney: Atlassian, 2023 [online]. [cit. 2023-02-28]. Dostupné z: <https://www.atlassian.com/company>
- [4] Atlassian (2023b). Hosting options for Jira Software. Jira Software guides [online]. Sydney: Atlassian [cit. 2023-03-28]. Dostupné z: <https://www.atlassian.com/software/jira/guides/more/jira-hosting>
- [5] Atlassian (2023c). What is the Jira family of applications?. Jira Work Management support [online]. Sydney: Atlassian [cit. 2023-04-02]. Dostupné z: <https://support.atlassian.com/jira-work-management/docs/what-is-the-jira-family-of-applications/>
- [6] Atlassian (2023d). What is an issue?. Jira Software support [online]. Sydney: Atlassian [cit. 2023-02-28]. Dostupné z: <https://support.atlassian.com/jira-software-cloud/docs/what-is-an-issue/>
- [7] Atlassian (2023e). What is scrum and how to get started. A guide to scrum: what it is, how it works, and how to start. [online]. Sydney: Atlassian [cit. 2023-08-12]. Dostupné z: <https://www.atlassian.com/agile/scrum>
- [8] Atlassian (2023f). Defining a project. Administering Jira applications Support [online]. Sydney: Atlassian [cit. 2023-04-04]. Dostupné z: <https://confluence.atlassian.com/adminjiraserver/defining-a-project-938847066.html>
- [9] Atlassian (2023g). Jira Software projects overview. Jira Software guides [online]. Sydney: Atlassian [cit. 2023-04-04]. Dostupné z: <https://www.atlassian.com/software/jira/guides/projects/overview - what-is-a-jira-project>
- [10] Atlassian (2023h). What is a business project? Atlassian Community [online]. Sydney: Atlassian [cit. 2023-04-04]. Dostupné z: <https://community.atlassian.com/t5/Jira-Work-Management-Articles/What-is-a-business-project/ba-p/1693132>
- [11] Atlassian (2023ch). What is a service project Jira Service Management support [online]. Sydney: Atlassian [cit. 2023-04-04]. Dostupné

- z: <https://support.atlassian.com/jira-service-management-cloud/docs/what-is-a-service-project/>
- [12] Atlassian (2023i). Jira Software workflows overview Jira Software guides [online]. Sydney: Atlassian [cit. 2023-04-04]. Dostupné z: <https://www.atlassian.com/software/jira/guides/workflows/overview-what-is-a-jira-workflow>
- [13] Atlassian (2023j). Defining status field values. Administering Jira applications Support [online]. Sydney: Atlassian [cit. 2023-04-04]. Dostupné z: [https://confluence.atlassian.com/adminjiraserver084/defining-status-field-values-979404594.html - Definingstatusfieldvalues-Defininganewstatus](https://confluence.atlassian.com/adminjiraserver084/defining-status-field-values-979404594.html-Definingstatusfieldvalues-Defininganewstatus)
- [14] Atlassian (2023k). Work with issue workflows. Atlassian support [online]. Sydney: Atlassian [cit. 2023-04-04]. Dostupné z: <https://support.atlassian.com/jira-cloud-administration/docs/work-with-issue-workflows/>
- [15] Atlassian (2023l). JQL: Get started with advanced search in Jira. Software guides [online]. Sydney: Atlassian [cit. 2023-04-05]. Dostupné z: <https://www.atlassian.com/software/jira/guides/jql/overview-what-is-jql>
- [16] Atlassian (2023m). JQL: the most flexible way to search Jira. Product & News [online]. Sydney: Atlassian [cit. 2023-04-05]. Dostupné z: <https://support.atlassian.com/jira-service-management-cloud/docs/what-is-a-service-project/>
- [17] Atlassian (2023n). What is the timeline view and how do I use it?. [online]. Sydney: Atlassian [cit. 2023-08-12]. Dostupné z: <https://support.atlassian.com/jira-work-management/docs/what-is-the-timeline-view-and-how-do-i-use-it/>
- [18] Axelos (2017). Managing successful projects with PRINCE2. Sixth edition. [cit. 2023-08-12] [London]: TSO. ISBN 978-0-11-331533-8.
- [19] BECK, K. a kol. (2001). Manifesto for Agile Software Development [online]. Scrum Alliance. [cit. 2022-08-24]. Dostupné z: <https://agilemanifesto.org/iso/cs/manifesto.html>
- [20] BUCHALCEVOVÁ, A (2005). Metodiky vývoje a údržby informačních systémů. 1. vyd. Praha : Grada. 163 s. Management v informační společnosti. [cit. 2023-08-12] ISBN 80-247-1075-7.
- [21] DOLEŽAL, J. (2016). Projektový management: komplexně, prakticky a podle světových standardů. Praha: Grada Publishing, Expert (Grada). ISBN 978-80-247-5620-2.
- [22] HOORY, L. (2022). Agile Vs. Waterfall: Which Project Management Methodology Is Best For You? [online]. New York: Forbes Media. [cit. 2023-08-12]. Dostupné z: <https://www.forbes.com/advisor/business/agile-vs-waterfall-methodology/>

- [23] JØSANG, A. a kol. (2015). Cybersecurity Through Secure Software Development. In: Bishop, M., Miloslavskaya, N., Theocharidou, M. (eds) Information Security Education Across the Curriculum. WISE. IFIP Advances in Information and Communication Technology, vol 453. Springer, Cham. ISBN 978-3-319-18500-2 Dostupné z: https://doi.org/10.1007/978-3-319-18500-2_5
- [24] KOMUS, A. a kol. Result Report: Status Quo (Scaled) Agile 2019/2020 (2020). [online]. Koblenz University of Applied Sciences. [cit. 2023- 01-12]. Dostupné z: <https://www.process-and-project.net/studien/studienunterseiten/status-quo-scaled-agile-2020-en/>
- [25] MÁCHAL a kol. (2015). Světové standardy projektového řízení: pro malé a střední firmy: IPMA, PMI, PRINCE2. Praha: Grada. Manažer. ISBN 978-80-247-5321-8.
- [26] MÁCHAL, P. a kol. (2017). Mezinárodní standard projektového řízení IPMA ICB v. 4 2017. [Praha]: IPMA Czech Republic, 2017. Publikace (IPMA). ISBN 978-80-7326-286-0.
- [27] PROJECT MANAGEMENT INSTITUTE (2021). A Guide to the Project Management Body of Knowledge (PMBOK Guide) and the Standard for Project Management. 7th edition. Pennsylvania: PMI. 250 s. ISBN 978-1-62825-664-2.
- [28] RUBIN, K. (2012). Essential Scrum: a practical guide to the most popular agile process. Upper Saddle River, NJ: Addison-Wesley. ISBN 9780137043293.
- [29] REHKOPF, M. (2022). Learn how to use burndown charts in Jira Software. Atlassian [online]. [cit. 2022-11-25]. Dostupné z: <https://www.atlassian.com/agile/tutorials/burndown-charts>
- [30] SCHWABER K. a SUTHERLAND J. (2020). The Scrum Guide. The Definitive Guide to Scrum: The Rules of the Game. [online]. [cit. 2023-08-12]. Dostupné z: <https://billlewisstraining.com/wp-content/uploads/2017/02/PMP-Agile-Study-Materials.pdf>
- [31] SVOZILOVÁ, A. (2016) Projektový management: systémový přístup k řízení projektů. 3., aktualizované a rozšířené vydání. Praha: Grada Publishing. Expert (Grada). ISBN 978-80-271-0075-0.
- [32] ŠOCHOVÁ, Z. a KUNCE E. (2014). Agilní metody řízení projektů. Brno: Computer Press. ISBN 978-80-251-4194-6.
- [33] Trello (2023a). A brief history of Trello. About Trello [online]. Trello – Atlassian [cit. 2023-05-07]. Dostupné z: <https://trello.com/about>
- [34] Trello (2023b). Trello Fundamentals. Atlassian University [online]. Trello – Atlassian [cit. 2023-05-07]. Dostupné z: <https://university.atlassian.com/student/path/1043254-trello-fundamentals-how-to-use-trello>

- [35] Trello (2023c). What are Trello Workspaces? Trello support [online]. Trello – Atlassian [cit. 2023-05-07]. Dostupné z: <https://support.atlassian.com/trello/docs/what-are-trello-workspaces/>
- [36] Trello (2023d). Learn Trello board basics. Getting started with Trello [online]. Trello – Atlassian [cit. 2023-05-07]. Dostupné z: <https://trello.com/cs/guide/trello-101>
- [37] Unicorn a.s. (2023) O společnosti Unicorn. [online]. [cit. 2023-08-11]. Dostupné z: <https://unicorn.com/cs/company-profile>
- [38] WAKE, B. (2012). Invest Model For User Stories [online]. [cit. 2022-12-28]. Dostupné z: <http://wiki.c2.com/?InvestModelForUserStories>
- [39] WATT, A. (2014). Project Management - 2nd Edition [cit. 2023-01-28]. ISBN 978-1-77420-013-1. Dostupné z: <https://opentextbc.ca/projectmanagement/>
- [40] WEINBERGER, M. (2019). The co-CEOs of \$26 billion Atlassian changed the way programmers work together. Now, they explain their plan to do it for everybody else too. Insider [online]. New York City: Business Insider [cit. 2023-02-28]. Dostupné z: <https://www.businessinsider.com/atlassian-mike-cannon-brookes-scott-farquhar-interview-2019-4>
- [41] WELLS, K. a kol. (2019) Project Management Essentials: Second Edition. New York: Business Expert Press. ISBN 978-1-94897-640-4.
- [42] ZANDHUIS A., Stellingwerf R. (2013). ISO 21500 Guidance on Project Management: A Pocket Guide. Zaltbommel: Van Haren Publishing. ISBN 978-90-8753-809-5.



Zadání bakalářské práce

Autor: Dominik Byrtus

Studium: I2000261

Studijní program: B1802 Aplikovaná informatika

Studijní obor: Aplikovaná informatika

Název bakalářské práce: Porovnání SW nástrojů pro řízení projektu JIRA a Trello

Název bakalářské práce AJ: Comparison of SW Tools for Project Management JIRA and Trello

Cíl, metody, literatura, předpoklady:

Tato bakalářská práce se zabývá porovnáním dvou nástrojů pro řízení projektů z hlediska jejich implementace a funkcionalit. Pro srovnání možností projektového řízení jsou využity produkty JIRA (spol. Atlassian) a Trello. Cílem této práce je pomocí případových studií přiblížit řešení implementace nástroje pro projektové řízení a poukázat na hlavní aspekty determinující výkonnost a efektivitu práce v kontextu projektového řízení. Na vybraných případových studiích, demonstrujících projekty různého rozsahu, bude také provedeno srovnání klíčových charakteristik vybraných platforem pro řízení projektů.

Osnova práce:

1. Úvod
2. Cíle práce
3. Teoretická část (principy projektového řízení, metodiky, představení platform JIRA a Trello)
4. Praktická část (vybrané aspekty implementace, porovnání platform a jejich funkcionalit)
5. Srhnutí
6. Závěr
7. Použitá literatura

Zadávací pracoviště: Katedra informačních technologií,
Fakulta informatiky a managementu

Vedoucí práce: doc. RNDr. Petr Tučnák, Ph.D.

Oponent: Ing. Tereza Otčenášková, BA, Ph.D.

Datum zadání závěrečné práce: 15.10.2021