

**Univerzita Hradec Králové**  
**Fakulta informatiky a managementu**  
**Katedra managementu**

**Projektové řízení zaměřené na komunikaci vybraného  
podniku**  
Diplomová práce

Autor: Bc. Filip Kmínek  
Studijní obor: IM2-K

Vedoucí práce: Ing. Václav Zubr, Ph.D.

Hradec Králové

Duben 2022

Prohlášení:

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

V Hradci Králové dne 29.4.2022

Filip Kmínek

#### Poděkování:

Děkuji vedoucímu diplomové práce Ing. Václavu Zubrovi, Ph.D. za metodické vedení práce, praktické rady a ochotu podílet se o své zkušenosti.

Také bych rád poděkoval firmě Czech Brain Data za možnost zpracovávat diplomovou práci, a především vývojovému týmu Onling za ochotu a trpělivost při zavádění změn v projektovém řízení.

## **Anotace**

Cílem této práce je aplikování agilních metod projektového řízení do projektu Online kasino ve firmě Czech Brain Data. Pro dosažení vyšší efektivity organizace týmu bude využito dostupných možností v rámci nástroj Azure DevOps. Výsledkem práce by měl být mj. návrh na zlepšení předávání informací mezi jednotlivými zaměstnanci firmy.

V úvodní části diplomové práce je čtenář seznámen s vývojem projektového řízení v projektu Online kasino. Projektové řízení se v průběhu času realizace tohoto projektu vyvíjelo, celý vývoj včetně všech pozitivních i negativních dopadů na tým jsou zaznamenány v praktické části.

Projektové řízení se podařilo optimalizovat tak, aby vyhovovalo potřebám týmu, vedením firmy byl projekt vyhodnocen jako úspěšný. Jedním z kladně hodnocených oblastí projektu Online kasino bylo hodnoceno samotné projektové řízení týmu a na základě této zpětné vazby vznikly návrhy a doporučení pro změnu a optimalizaci projektového řízení u ostatních projektů firmy Czech Brain Data.

Klíčová slova: projektové řízení, agile, Azure DevOps, schůzky, komunikace

## **Annotation**

### **Title: Project Management Focused on Communication of Chosen Company**

The aim of this work is to apply agile project management methods to the Online Casino project in the company Czech Brain Data. The available options within the Azure DevOps tool will be used to achieve greater efficiency of the team organization. The result of the work should be, among other things, a proposal to improve the transfer of information between individual employees of the company.

In the introductory part of the diploma thesis, the reader is acquainted with the development of project management in the Online Casino project. Project management has evolved over time, the entire development, including all positive and negative impacts on the team, is described in the practical part.

Project management was optimized to meet the needs of the team, the company's leaders evaluated the project as successful. One of the positively evaluated areas of the Online Casino project was the project management of the team, and based on this feedback, proposals and recommendations were made for changing and optimizing project management for other Czech Brain Data projects.

Keywords: project management, agile, Azure DevOps, meetings, communication

## Obsah

1	Úvod.....	1
2	Cíl práce a metodika zpracování .....	2
3	Teoretická část.....	4
3.1	Tradiční vs. agilní projektové řízení.....	4
3.2	Agilní řízení v softwarové doméně .....	5
3.3	Standardní přístup projektového řízení a aktuální trendy v řízení .....	6
3.3.1	Model vodopádu (Waterfall Model).....	7
3.4	Přístupy Scrum a Kanban .....	10
3.4.1	Metodika Scrumu.....	10
3.4.2	Metodika Kanbanu.....	16
3.5	Možnosti využití Azure DevOps pro projektové řízení .....	20
3.5.1	Další prvky Azure DevOps a jejich využití .....	22
3.5.2	Souhrn použití Azure DevOps .....	23
3.6	Alternativní nástroje pro projektové řízení .....	24
3.6.1	Jira.....	24
3.6.2	Trello.....	27
3.6.3	You Track .....	29
3.7	Komunikační nástroje.....	30
3.7.1	Google Suite .....	30
3.7.2	Nástroj Slack.....	31
3.7.3	Microsoft Teams .....	32
3.8	Další přístupy projektového řízení využívané IT leadery v praxi .....	33
4	Vývoj projektu Online kasino .....	35
4.1	Představení firmy Czech Brain Data .....	35
4.1.1	Struktura firmy .....	36

4.2	Projekt Online kasino .....	38
4.3	Projektové řízení ve firmě Czech Brain Data.....	38
4.4	Zahájení projektu a složení týmu .....	41
4.5	Aplikování Backlog a Sprint v Azure DevOps.....	43
4.6	Změna komunikačních nástrojů .....	53
5	Dopad projektu Online kasino na firmu Czech Brain Data .....	55
5.1	Organizační struktura .....	56
5.1.1	Business Owner.....	58
5.1.2	Product Owner.....	58
5.1.3	Tým.....	59
5.2	Pravidelné schůzky zaměstnanců .....	60
5.3	Využití Azure DevOps .....	63
5.3.1	Předpoklady k úspěšné aplikaci.....	65
5.3.2	Využívané elementy .....	66
5.3.3	Doporučené nástroje pro řízení týmu.....	67
5.3.4	Zodpovědnosti a povinnosti členů týmu.....	69
6	Shrnutí výsledků.....	71
7	Závěry a doporučení .....	74
8	Seznam použité literatury.....	75
9	Seznam obrázků .....	80

# 1 Úvod

Zdá se, že neexistuje žádná standardní nebo univerzální definice slova komunikace v byznysu nebo pro každodenní použití. Berry [7] definuje komunikaci jako přenos a výměnu informací a také porozumění od jedné osoby k druhé. Vyžaduje použití symbolů, které mají mezi dotýčnými jednotlivci stejný význam.

Bryde a kol. [8, s. 22] souhlasí a uvádí, že: „*Komunikace je v podstatě interaktivní proces odesílání a přijímání informací.*“ Tento akt přenosu informací z jednoho místa na druhé umožňuje lidem porozumět zamýšlenému významu druhé strany. Z výše uvedených definic je komunikace obousměrný proces, ve kterém účastníci současně vyjednávají o významu (kódují - dekodují) výměnou informací, zpráv, nápadů a pocitů.

Komunikace je identifikována jako jedna z 10 oblastí znalostí pro efektivní realizaci projektu. Tento prvek realizace projektu vyžaduje, aby informace byly poskytovány správným lidem, ve správném formátu a včas.

Komunikace je nepostradatelným nástrojem pro efektivní realizaci projektu, protože informace jsou vyžadovány k doplnění různých úkolů probíhajících ve stejnou dobu.

V návaznosti na výše uvedené jsou komunikační dovednosti jednou z kritických kompetencí, které projektový manažer vyžaduje, aby byl úspěšný [17]. Projektoví manažeři se stejnou měrou podílejí na koordinaci úkolů prováděných v různých strukturách pracovního členění, proto je zde vyžadována komunikace, která může zahrnovat mimo jiné schůzky, zasílání zpráv ostatním členům, sestavování průběžných zpráv a grafů vysvětlujících stav projektu atd.

Práce se bude zabývat projektovým řízením zaměřeným na komunikaci a jejím cílem je analyzovat vývoj projektového řízení ve vybrané IT firmě.



## 2 Cíl práce a metodika zpracování

Jedním z cílů je seznámit čtenáře s novým projektem Online kasino, který firma Czech Brain Data označila jako klíčový pro budoucí směřování firmy.

Hlavním úkolem diplomové práce je aplikování agilních metod projektového řízení do projektu Online kasino, využití dostupných možností v rámci nástroje Azure DevOps a nastavení efektivní komunikace v rámci vývojového týmu.

V případě úspěšné aplikace agilních metod na dílčí projekt bude vytvořen seznam doporučení a návrhů na aplikování ověřených postupů do dalších projektů v rámci firmy Czech Brain Data. V případě neúspěchu bude vyhodnocen koncový stav, shrnuty aplikované přístupy včetně výsledků a následně budou navržena možná opatření a doporučení k dosažení lepších výsledků.

Součástí zpracování závěrečné práce je také nalezení odpovědí na celkem 3 položené výzkumné otázky:

1. Je vhodné kombinovat různé agilní metodiky v řízení jednoho projektu?
2. Existuje doporučená struktura pro evidenci úkolů?
3. Má zavádění agilních metod kladný vliv na produktivitu týmu?

Autor diplomové práce je pracovníkem vybrané firmy a členem zkoumaného projektového týmu. Z tohoto důvodu bude využita metoda pozorování pro získání potřebných informací k vytvoření seznamu kladných a záporných prvků využívaných v projektovém řízení u vybraného projektu. Prvotní fáze řízení projektu bude probíhat dle stávajících pravidel a s využitím nástrojů, na které jsou zaměstnanci firmy zvyklí. Na základě pozorování a mapování stávající situace budou navržena optimalizační doporučení, která mají za cíl zvýšit efektivitu projektového týmu a dodávat zákazníkovi kvalitní software dle jeho požadavků. Jelikož si je vedení firmy vědomé náročnosti změny, budou zaváděny postupně v menším rozsahu.

Každá fáze bude kolektivně vyhodnocena, budou prezentovány prvky s pozitivním vlivem na projekt a budou identifikovány případné další negativní vlivy na projekt. Toto zhodnocení bude probíhat formou rozhovorů nejprve na úrovni týmu, aby byla získána zpětná vazba od všech kooperujících členů a následně bude

probíhat pravidelná revizní schůzka s vedením firmy. Každá z plánovaných změn bude důkladně připravena, u komplikovanějších změn bude využito ilustračních diagramů či schémat pro snazší pochopení ze strany členů týmu.

Teoretická část se bude primárně zabývat popisem výhod a nevýhod vodopádového a agilního přístupu projektového řízení, dále budou specifikovány vlastnosti agilních metodik Scrum a Kanban. Závěrem teoretické části bude mapování dostupných nástrojů pro podporu projektového řízení. Na základě získaných informací v teoretické části a se znalostmi získaných z pozorování a rozhorů s členy projektového týmu budou doporučeny optimalizační změny. Doporučované přístupy a techniky z teoretické části budou vyhodnocovány individuálně s ohledem na specifika a požadavky vybraného projektu.

Vedení firmy stanovilo dobu 18 měsíců jako vhodnou dobu pro aplikování změn. Po uplynutí této doby zhodnotí vedení firmy dosažených výsledků projektu i s ohledem na změny v projektovém řízení. Jestliže bude dosaženo očekávaného výsledku, změny budou zavedeny i do dalších projektů vybrané firmy.

### 3 Teoretická část

Tato část práce se zaměřuje na zpracování literární rešerše ve vztahu k tématu, dále na porovnání tradičních přístupů a agilních technik, jaké jsou aktuální trendy v řízení, definuje přístupy Scrum a Kanban, popíše alternativní nástroje projektového řízení, komunikační nástroje, možnosti využití Azure DevOps, příklady z IT (informační technologie) praxe a další.

#### 3.1 Tradiční vs. agilní projektové řízení

Úspěch projektového řízení je obtížné definovat, protože lidé mají různé názory na to, co je úspěšné provedení projektu. Křivánek rozlišuje mezi úspěchem projektového řízení a projektovým úspěchem. Úspěch projektového řízení je, když je operace dokončena v rámci stanoveného rozpočtu, času a specifikací. Na druhé straně úspěch projektu znamená, že cíle, pro které byl projekt zahájen, by měly být splněny nebo dosaženy [24].

Kromě toho, podle Barndlunda [6] existuje 10 oblastí znalostí, které umožňují projektovému manažerovi úspěšně provádět proces projektového řízení. Jsou to:

- řízení akcionářů projektu;
- řízení projektového mixování;
- řízení období projektu;
- řízení omezení projektu;
- řízení rozpočtu projektu;
- řízení hodnoty projektu;
- řízení lidského kapitálu;
- řízení projektové komunikace;
- řízení hrozeb projektu;
- supervize projektových zakázek.

Tyto oblasti znalostí jsou považovány za klíčové pro úspěšné řízení projektů a vzájemně se doplňují [6].

Barnlund [6, s. 37] uvádí, že: „*konkurence je globální, příležitosti jsou dynamické a obchodní procesy jsou vysoce složité.*“ V tak složitém a rychle se měnícím prostředí, podle tradičního přístupu k projektovému řízení, se projektoví manažeři snažili plánovat a předvídat všechny možné detaily a potřeby ještě před realizací projektu, ale v dnešním rychle se měnícím prostředí je projektové řízení jako disciplína změny pod vlivem nedostatků tradičního přístupu k projektovému řízení [6].

Aby byly organizace konkurenceschopné, jsou nuceny rozpoznat změny a být flexibilnější, když se s nimi setkají. V této souvislosti se rozšíření agilního projektového řízení (APM), nad rámec vývoje softwaru, stává žádoucí reakcí na rychle se měnící a náročné obchodní prostředí. Agilita je dnes velmi aktuální téma, nicméně nejčastěji se s ní lze setkat v literatuře související s vývojem softwaru. Agilní metodika vývoje softwaru je založena na postupném přístupu, který se opakuje. Zaměřuje se na flexibilitu, přijatelnost změn, neustálý pokrok a silnou interakci.

Podle Doležala a kol. [13] se objevuje zásadní poznání, že agilní projektové řízení má co nabídnout v rámci širšího projektového prostředí.

### **3.2 Agilní řízení v softwarové doméně**

Agilitu organizace lze definovat jako schopnost rychle reagovat na změny v dynamickém podnikatelském prostředí. Přestože se agilita objevila jako koncept pro vývoj softwaru a IT projekty, dnes představuje jednu ze základních konkurenčních výhod současných organizací. V zásadě je agilita optimální rovnováhou mezi potřebami stability a adekvátní úrovní flexibility [38].

Počátek APM (Agile Project Management) je spojen s Agile Manifesto, dokumentem publikovaným v roce 2001, který kombinuje agilní hodnoty a principy pro vývoj softwaru. APM je podporován vývojem souboru postupů, nástrojů a technik používaných k poskytování hodnoty zákazníkům, s přihlédnutím k nepředvídatelnosti projektu a měnícím se požadavkům zákazníka [38].

Na druhou stranu tradiční projektové řízení zahrnuje detailní a komplexní plánování a kontrolu s významem definování požadavků klienta na začátku projektu, bez následných změn v průběhu projektu [28]. Tradiční projektové řízení

předpokládá, že okolnosti, které projekt ovlivňují, jsou předvídatelné, že požadavky jsou jasné a dobře srozumitelné, ale projekty jen zřídka sledují sekvenční tok během implementace a klienti obvykle nejsou schopni definovat všechny požadavky na začátku projektu.

APM je více odkázán na přizpůsobení procesu a metodiky projektového řízení řešeného problému s důrazem na dodání částí projektu, případně produktu a rychlé úpravy dle potřeby. Jiné typy projektů však sdílí mnoho výzev, které APM vyřešilo při vývoji softwaru. Zatímco APM obecně používají týmy vývoje softwaru, počet projektových manažerů a obchodních expertů, kteří doporučují implementaci APM pro řízení projektů a týmů mimo softwarový průmysl, roste.

APM poskytuje organizacím schopnost vyrovnat se s dynamickým vývojem a změnami v podnikatelském prostředí, a proto je důležité prozkoumat použitelnost APM i mimo vývoj softwaru a IT projekty [28].

### **3.3 Standardní přístup projektového řízení a aktuální trendy v řízení**

Během konce 50. a začátku 60. let 20. století se projektový management objevil jako disciplína, a to díky americkému ministerstvu obrany, které chtělo jedinečné kontaktní místo pro všechny projekty, někoho, kdo by byl dostupný ve všech fázích projektu. Evoluce projektového řízení se v dnešních organizacích stala nezbytností [12].

Projektový management byl vytvořen jako disciplína pro systematizaci a manipulaci se zdroji v rámci projektu, aby byla celá práce dokončena mezi stanoveným časem a rozpočtem. Za posledních padesát let mnoho společností zahrnuje projektové řízení do svých procesů, a proto se projektové řízení rychle rozvíjí [12].

U velkých IT oddělení je projektový manažer zodpovědný za manipulaci se zdroji, rozpočtem a včasností projektu. Těmito projekty mohou být vývoj softwaru, vytváření sítí, instalace hardwaru a další. Tradiční projekty začaly tím, že se přesně definovalo, co má projekt přinést, po odsouhlasení a podpisu se zákazníkem projektový tým začal vyvíjet produkt nebo systém s těmito specifikacemi a později poskytnul hotový produkt klientovi [14].

Tento tradiční přístup je založen na chronologickém procesu navrhování, jakémsi vodopádu, který prochází fázemi projektu. Navíc k tomuto procesu je jakákoliv změna během procesu velmi nákladná.

### 3.3.1 Model vodopádu (Waterfall Model)

Metodologie vodopádu – také známá jako model vodopádu – je sekvenční vývojový proces, který jako vodopád prochází všemi fázemi projektu (analýza, návrh, vývoj a testování), přičemž každá fáze zcela končí před další ze začínajících fází.

Říká se, že metodika Waterfall se řídí příslovím „*dvakrát měř, jednou řež*“. Úspěch metody Waterfall závisí na množství a kvalitě práce odvedené na počátku, kdy je vše zdokumentováno včetně uživatelského rozhraní, uživatelských požadavků, všech variant a výsledků funkcí. Vzhledem k tomu, že většina průzkumů se provádí předem, odhady času potřebného pro každý požadavek jsou přesnější, což může poskytnout předvídatelnější datum vydání. Pokud se u projektu Waterfall změní parametry, je těžší změnit kurz než s agilní metodologií [28].

Metodologie Waterfall se řídí chronologickým procesem a funguje na základě pevných dat, požadavků a výsledků. Díky této metodě nemusí jednotlivé realizační týmy neustále komunikovat, a pokud nejsou vyžadovány konkrétní integrace, jsou obvykle samostatné. Členové týmu mají také tendenci pracovat nezávisle a neočekává se od nich, že budou poskytovat zprávy o stavu tak často jako u agilního přístupu. Obvykle jedna fáze nezačíná, dokud není dokončena předchozí [28].

Pokud bude použit jako příklad projekt vývoje softwaru, proces Waterfall obvykle zahrnuje fáze, které vypadají takto:

1. **Požadavky** - metodologie Waterfall závisí na přesvědčení, že všechny požadavky projektu lze shromáždit a pochopit předem. Projektový manažer dělá vše pro to, aby podrobně porozuměl požadavkům investora projektu. Písemné požadavky, obvykle obsažené v jediném dokumentu, se používají k popisu každé fáze projektu, včetně nákladů, předpokladů, rizik, závislostí, metrik úspěchu a časových plánů dokončení [28].
2. **Design** - zde vývojáři softwaru navrhují technické řešení problémů stanovených požadavky na produkt, včetně scénářů, rozložení a datových

modelů. Nejprve se vytvoří vyšší nebo logický návrh, který popisuje účel a rozsah projektu, obecný dopravní tok každé komponenty a integrační body. Jakmile je toto dokončeno, je transformováno do fyzického návrhu pomocí specifických hardwarových a softwarových technologií [25].

3. **Implementace** - po dokončení návrhu začíná technická realizace. Toto může být nejkratší fáze procesu vodopádu, protože pečlivý soupis požadavků a design již byly provedeny. V této fázi programátoři kódují aplikace na základě požadavků a specifikací projektu, přičemž také probíhá určité testování a implementace. Pokud jsou během této fáze nutné významné změny, může to znamenat návrat do fáze návrhu [25].
4. **Ověřování nebo testování** - než může být produkt uvolněn zákazníkům, je třeba provést testování, aby se zajistilo, že produkt neobsahuje chyby a že byly splněny všechny požadavky, což zajišťuje dobrou uživatelskou zkušenost se softwarem. Testovací tým se zaměří na návrhové dokumenty, osoby a scénáře uživatelských případů dodané produktovým manažerem, aby vytvořil své testovací případy [25].
5. **Nasazení a údržba** - jakmile je software nasazen na trh nebo uvolněn zákazníkům, začíná fáze údržby. Jakmile budou nalezeny závady a budou přicházet požadavky na změny od uživatelů, bude přidělen tým, který se postará o aktualizace a vydá nové verze softwaru [25].

#### Výhody metodiky Waterfall:

- Metodologie Waterfall je přímočará, dobře definovaná metodika projektového řízení s osvědčenými výsledky. Vzhledem k tomu, že požadavky jsou od začátku jasně stanoveny, každý přispěvatel ví, co se musí kdy udělat, a může si efektivně naplánovat čas na dobu trvání projektu.
- Vývojáři mohou zachytit chyby návrhu během fází analýzy a návrhu, což jim pomůže vyhnout se psaní chybného kódu během fáze implementace.
- Celkové náklady na projekt lze přesně odhadnout, stejně jako časový plán, po definování požadavků.

- Díky strukturovanému přístupu je snazší měřit pokrok podle jasně definovaných milníků.
- Zákazníci ne vždy přidávají do projektu nové požadavky a nedochází tak ke zdržování výroby [21].

#### Nevýhody metodiky Waterfall:

- Metodika Waterfall trvá na předběžném plánování projektu a je později méně flexibilní nebo agilní. Změny, které přijdou dále v procesu, mohou být časově náročné a nákladné.
- S tímto chronologickým přístupem může realizace projektů trvat déle než s iterativním, jako je agilní metoda.
- Klienti často na počátku úplně nevědí, co chtějí, a otevírají dveře žádostem o změny a nové funkce později v procesu, když je obtížnější vyhovět.
- Klienti nejsou zapojeni do fází návrhu a realizace.
- Když je jedna fáze procesu zpožděna, všechny ostatní fáze jsou zpožděny.

Model Waterfall je přijat projektovými manažery, kteří čelí projektům, které [21]:

- a) mají jasně definované požadavky;
- b) nabízí od začátku jasný obraz o tom, jak se věci budou vyvíjet;
- c) klienti, u kterých se zdá nepravděpodobné, že by změnili rozsah projektu, jakmile bude probíhat.

Pokud projektový manažer upřednostňuje jasně definované procesy, kde jsou náklady, design a časové požadavky známy předem, pak je metoda Waterfall tou správnou cestou, pokud samotný projekt těmto omezením napomáhá.

#### **Hlavní rozdíly mezi modelem vodopádu a agilním řízením projektů**

Zatímco model Waterfall spoléhá na důkladné předběžné plánování, ve kterém musí být každá fáze plně dokončena před zahájením další, agilní řízení je flexibilnější, iterativní proces, který řeší úkoly plánování, návrhu, implementace a testování v kratších, opakujících se cyklech.



S Waterfall modelem zákazník obvykle nehraje praktickou roli, jakmile jsou jeho požadavky shromážděny. Díky agilnímu přístupu má zákazník větší vstup do vývoje.

V modelu Waterfall lze změny snadno přizpůsobit v raných fázích vývoje, například když projektový manažer stále dotváří specifikační dokumenty s vývojovým týmem a klienty. Později v procesu Waterfall, jakmile již začalo kódování, může být provedení změn návrhu obtížné a nákladné oproti flexibilitě v rámci agilního přístupu [21].

### **3.4 Přístupy Scrum a Kanban**

Jak již bylo uvedeno, agilní projektové řízení je strukturovaný a iterativní přístup k řízení projektů a vývoji produktů. Agilní projektové řízení si uvědomuje volatilitu vývoje produktů a poskytuje metodologii pro samoorganizující se týmy, aby reagovaly na změny, aniž by docházelo k nedostatkům a nepochopení v komunikaci. Následující podkapitoly podrobně definují přístupy agilního řízení, a to Scrum a Kanban [18].

#### **3.4.1 Metodika Scrumu**

V roce 1986 Hirotaka Takeuchi a Ikujiro Nonaka napsali článek The New Product Development Game, který se objevil v The Harvard Business Review [27]. Článek představil myšlenky, které měly velký vliv na to, jak byl Scrum navržen v rané fázi. Zavedli pravidlo hry pro vývoj nových produktů s důrazem na rychlost a flexibilitu. V roce 1993 vytvořili Jeff Sutherland, John Scumniotales a Jeff McKenna původní kontext Scrumu přijetím, implementací a dokumentací modelu pro vývoj softwaru ve společnosti Easel Corporation.

V roce 1995 začal Ken Schwaber formalizovat pravidla Scrumu a své poznatky zkomponoval do knihy „Agile Software Development with Scrum“ v roce 2001 [27].

Jak již bylo zmíněno, Scrum je počítán jako člen agilního modelu, který byl upraven tak, aby byl flexibilnější a adaptabilnější metodou, kterou lze aplikovat na různé obchodní situace. Scrum je jednoduchý proces ve vývoji softwaru, který se zaměřuje na kvalitu týmu, navržené techniky a metody údržby. Navíc se Scrum zaměřuje spíše

na lidi než na proces vývoje. Tato metoda klade důraz na komunikaci, spolupráci a rychlou výměnu informací mezi členy týmu. Díky své schopnosti zvýšit míru úspěšnosti ve vývoji softwaru je Scrum jedním z nejpoužívanějších procesů v agilním vývoji softwaru [27].

### **3.4.1.1 Proces Scrumu**

Hlavním cílem Scrumu je vyvinout a rychle dodat software klientovi s co nejmenšími chybami nebo defekty. V praxi jsou projekty normálně rozděleny na malé kousky, zatímco Scrum týmy se vyznačují vlastním řízením. Ve Scrumu projekty vždy postupují ve sprintech, sérii iterací, kdy každý sprint trvá obvykle méně než jeden měsíc. Na začátku každého sprintu se členové týmu zavazují dodat určitý počet funkcí, které jsou uvedeny v seznamu Product Backlog, což je seznam požadavků [26].

Na konci každého sprintu budou samostatně dokončené funkce integrovány do vyvíjejícího se produktu nebo systému. Kontrola sprintu se provádí na konci každého sprintu, když tým předvede nové funkce systému vlastníkovi produktu a zainteresovaným stranám, které poskytují zpětnou vazbu, která by mohla ovlivnit další sprint. To umožňuje upravit nebo přeorientovat směřování projektu na základě již dokončené práce [26].

### **3.4.1.2 Fáze Scrumu**

Podle Scrumova procesu existují tři hlavní fáze: fáze před hrou, fáze vývoje, fáze po hře. Scrum, zkratka „Scrummage“, je název odvozený ze hry Rugby, což dává smysl, proč se fáze ve Scrumu nazývají „hry“. Scrum se používá k popisu fáze, kdy je hra restartována, během které se několik hráčů rugby z každého týmu umístilo do formace, aby získali míč, který byl umístěn uprostřed mezi oba týmy. Rugby's „scrum“ je dokonalým příkladem pro Scrum, proces vývoje, který klade důraz na komunikaci tváří v tvář a úzkou spolupráci mezi členy týmu. Kromě toho Scrum také odráží povahu toho, čeho je prostřednictvím metody dosaženo, pokud jde o členy skupiny, kteří postupně usilují o lepší efektivitu práce. Podrobnosti o fázích Scrumu jsou popsány v následující části [27].

## 1. Před hrou

Tato fáze se týká dvou aspektů: plánování a architektury na vysoké úrovni, které jsou reprezentovány dvěma symboly černého obdélníku. Tato první fáze ve sportu se týká času; trvání této doby se může u jednotlivých sportů lišit a vyskytuje se před zápasem, kdy trenéři a hráči plánují strategie pro nadcházející zápas. Proces plánování definuje nové vydání na základě aktuálně známých nevyřízených položek spolu s odhadem jeho nákladů a harmonogramu (priority a odhady úsilí) [31].

Tento proces se skládá jak z konceptualizace, tak z analýzy, pokud se vyvíjí nový systém. Konceptualizace není vždy potřeba, pokud projekt vyžaduje pouze úpravu nebo vylepšení stávajícího systému. Seznam nevyřízených produktů nebo seznam položek nevyřízených položek je pravidelně aktualizován během procesu plánování, aby byl do vývojové fáze předložen nejaktuálnější seznam položek nevyřízených položek. Položky v backlogu obsahují všechny požadavky uživatelů a scénář systému.

Proces architektury pracuje v souladu s procesem plánování, aby zvažil všechny přiřazené položky nevyřízených položek a identifikoval změny nezbytné pro implementaci položek nevyřízených položek. Provádí také doménovou analýzu rozsahu, který je nutný k sestavení, aktualizaci nebo vylepšení doménových modelů, aby odrážely požadavky a kontext systému. Kromě toho také identifikuje problémy, které mohou nastat v důsledku měnících se požadavků. Výstup z tohoto procesu bude standardem kódování, který se předkládá do vývojové fáze [31].

## 2. Hra

Ve fázi vývoje musí být seznam produktového backlogu z fáze před hrou rozdělen na seznam nových funkcí systému nazývaný seznam nevyřízených položek Sprint. Seznam nevyřízených sprintů specifikuje vlastníka každé úlohy, a kdy bude provedena před přechodem na období Sprintu.

Sprint je určité časové období, kdy dochází k vývojovým aktivitám. Existuje několik iterativních vývojových sprintů nebo cyklů, které se používají k vývoji systému, na kterém se pracuje. Nový systém nebo funkce, které se vyvinuly z fáze sprintu, budou postupně přidávány (v pořadí dokončení) ke konečnému produktu, který má být vydán nebo připraven pro fázi „Ukončení hry“, kde bude tento produkt nebo funkce testována, integrována atd., dokud nebude připravena k vydání.

Sprint obvykle vyžaduje v průměru jeden až čtyři týdny na základě složitosti vydaného produktu, posouzení rizik a počtu a závažnosti nesplněných požadavků. Zatímco rychlost každého sprintu je řízena přidělenou délkou sprintu, riziko je průběžně vyhodnocováno během období sprintu [32].

Každý sprint zahrnuje jeden nebo více Scrum týmů provádějících všechny vývojové aktivity. V rámci přípravy na vývoj bude produktový backlog, který obsahuje seznam požadavků odvozených z aspektu plánování ve „Fázi před hrou“, rozdělen do různých systémových funkcí nazývaných Sprint Backlog s jednou nebo více funkcemi přiřazenými každému sprintu.

Během období Sprintu pořádá Scrum Master denní schůzky, které poskytují místo a příležitost týmům prezentovat práci, kontrolovat pokrok, nastolovat a řešit problémy a přidávat nové nevyřízené položky (v případě potřeby). Během poslední fáze je vytvořen prototyp nebo demo systém jako konečná verze produktu, která se před vydáním produktu předloží zákazníkovi ke konečnému schválení. Kontrola po závěrečném sprintu pokrývá funkční systémy, které zahrnují přiřazené položky nevyřízeného záznamu a zahrnují změny. Na kontrole by se měl podílet celý vývojový tým a tým produktového managementu, zatímco ostatní účastníci, jako jsou zákazníci, prodejní tým a marketingový tým, se mohou také zúčastnit [32].

### **3. Po hře**

Tato fáze ukončuje vývojový proces. „Fáze po hře“ ve sportu označuje čas těsně po zápase, kdy trenéři a hráči probírají, co se ve hře stalo, a diskutují o tom, jaká vylepšení lze udělat a jaká ponaučení si lze ze hry vzít. Uzavření nastane, když jsou splněny všechny požadavky a produkt se připravuje k uvolnění. Činnosti této fáze zahrnují integraci systému, testování systému, finální dokumentaci, školení a přípravu marketingových materiálů a finalizuje se proces uvolnění nebo odeslání produktu [32].

#### **3.4.1.3 Role Scrumu**

Scrum definuje celkem tři role: vlastník produktu (Product Owner), vedoucí Scrumu (Scrum Master) a scrum tým (Scrum Team) [34].

## 1. Vlastník produktu

Product Owner vždy úzce spolupracuje s klienty a dokáže zastupovat to, co klienti hledají. Vlastník produktu nejlépe rozumí vizi projektu a může pomoci zajistit, aby tým přinášel hodnotu klientovi podnikání. Vlastník produktu generuje uživatelské příběhy, určuje priority a přidává je do produktového backlogu.

Podobně jako u požadavku je uživatelský příběh přehledem o tom, co uživatel dělá nebo potřebuje k plnění svých povinností. Vlastník produktu je odpovědný a přístupný vývojovému týmu, aby formuloval požadavky zákazníků a objasnil kritéria přijetí, aby bylo zajištěno, že všechny požadavky a smlouva budou splněny při uvedení produktu na trh [34].

## 2. Vedoucí Scrumu

Scrum Master působí jako kouč, který přebírá odpovědnost za přeměnu uživatelských příběhů na detailní úkoly, upřednostňuje uživatelské příběhy a zajišťuje, že projekt probíhá podle plánu pořádáním každodenních schůzek Scrum.

Kromě toho Scrum Master slouží také jako facilitátor, který odstraňuje veškeré překážky, které brání týmu, a pomáhá týmu dokončit práci v požadované kvalitě a včas [34].

## 3. Scrum tým

Scrum tým je samoorganizovaná skupina s vysokým stupněm odpovědnosti a pravomocí. Tým by měl mít všechny potřebné role, které se běžně skládají ze softwarových inženýrů, architektů, analytiků a testerů [34]. Navíc v návaznosti na crossfunkční koncept; není třeba rozdělovat povinnosti a pozice v týmu. Členové Scrumu se mohou podílet na jakémkoli úkolu a spolupracovat na řešení požadavků zákazníků a vzájemně si pomáhat předcházet a řešit problémy, které se mohou neočekávaně objevit během pracovních procesů [34].

### 3.4.1.4 Implementace scrumu

Implementace Scrumu se zaměřuje na krátký vývojový cyklus založený na iterativních postupech [34]. Prvky implementace jsou následující:

#### 1. Charakteristika Scrumu

Scrum tým může být v rámci projektu rozdělen do několika malých týmů. Každý tým má méně než 6 členů a každý člen týmu spolupracuje s ostatními členy týmu

v rámci bezprostředního týmu i členů v jiném týmu. Vzhledem k napjatému harmonogramu projektu jsou nutné časté kontrolní schůzky k posouzení pokroku.

Implementace Scrumu také umožňuje flexibilnější harmonogram a harmonogram dodání. Vzhledem k tomu, že harmonogram úkolů může být vyžadován dříve nebo později, počáteční plán a výstup jsou diktovány vývojovým prostředím [40].

## **2. Odhad Scrumu**

Scrum lze odhadnout pomocí posouzení funkčních požadavků jako standardního nástroje pro odhad. Tento odhad je vyžadován pouze pro počáteční část projektu, protože náklady a harmonogram jsou definovány dynamicky v reakci na environmentální faktory rozvoje. Projektový přístup Scrum bere v úvahu rychlost i zrychlení z hlediska funkčních požadavků a lze jej rozdělit do tří fází. Na začátku je rychlost a zrychlení nízké, protože se buduje základní infrastruktura. Během vývoje se zrychlení zvyšuje, protože základní funkčnost je vložena do objektů. Na konci je zrychlení nízké, i když rychlost zůstává udržitelně vysoká [40].

## **3. Kontrola scrumu:**

Aby se zabránilo tomu, že se složitý projekt vymkne kontrole, je nutná kontrola managementu. Řízení kontroly v metodologii Scrum pomáhá zajistit, že všechny funkční požadavky z Product Backlog jsou v aktuální verzi produktu adekvátně řešeny. Budou zde nějaké další nevyřízené položky, které pocházejí z požadavku zákazníka na vylepšení chyb a defektů. Změny dopadu na aktuální komponenty produktu se musí změnit, aby bylo možné v novém vydání implementovat položku nevyřízené položky. Identifikovaná rizika, která budou bránit úspěchu projektu, a jejich řešení jsou navíc průběžně vyhodnocována v reakci na projektový záměr. Řešení problémů a rizik často vedou ke změnám požadavků, které mohou být nutné sdělit klientovi [40].

## **4. Doručování Scrumu:**

Jak je zmíněno ve Scrum Estimation, dodávaný produkt je flexibilní v závislosti na kontaktu se zákazníkem, dovednostech vývojáře a trhu. Produkt lze předložit klientovi ke zvážení a přidat jej do další verze kdykoli během projektu. Kromě toho je dodání produktu také diktováno různými proměnnými včetně času, nákladů, konkurence a funkčnosti [40].

### 3.4.2 Metodika Kanbanu

Metoda Kanban je přístup k postupným a evolučním změnám procesů a systémů pro organizace [16]. Využívá nedokončenou výrobu k odhalení problémů, stimulaci spolupráce a neustálému zlepšování systému.

Podle Hammarberga [16] lze metodu Kanban implementovat jako evoluční změnu existujícího systému v dané organizaci. Metoda Kanban je založena na třech základních principech:

1. Začít tím, co se dělá nyní: Neexistuje žádný univerzální předepsaný soubor pravidel, který by fungoval pro každou společnost, tým nebo projekt. Tým nebo osoba, která chce tento přístup implementovat, by se měla neustále snažit implementovat změny do svého stávajícího systému.
2. Souhlas s prováděním postupných, vývojových změn: Organizace nebo tým musí souhlasit s postupným zlepšováním způsobu své práce. Vzhledem k odporu organizací ke změnám se přírůstkové změny ukážou jako efektivnější. To podporuje malé změny, které mají větší šanci na udržení týmem.
3. Respekt současného procesu, role, odpovědnosti: Strach ze změny může být způsoben procesy nebo prvky organizace, které jsou obecně přijímány a které by stálo za to zachovat. Souhlasem s respektováním aktuálních rolí a pracovních názvů je možné eliminovat počáteční obavy. To pomáhá získat větší podporu pro nový systém. Prezentace metody Kanban jako alternativy k rozsáhlým změnám v titulech, rolích nebo odpovědnostech může jednotlivcům pomoci lépe si uvědomit její výhody [16].

#### 3.4.2.1 Pět základních vlastností Kanbanu

Podle Carrola a Morrise existuje pět základních principů, které jsou přítomny v každé úspěšné implementaci metody Kanban: [9]

1. vizualizace pracovního postupu;
2. omezení nedokončené práce;
3. řízení toku;
4. výslovnost zásad procesu;

5. společné zlepšování (pomocí modelů a vědeckých metod).

### 1. Vizualizace pracovního postupu

Hlavním způsobem vizualizace pracovního postupu je použití nástroje zvaného Kanban board. Může to být například tabule s nálepkami nebo elektronický kartový nástěnný systém. Board může být rozdělen do pěti stupňů, z nichž každý je schopen přijímat různé součásti projektu. Většina kanbanů obsahuje své vlastní řazení sloupců, ale obvykle by každá verze měla mít svou vlastní verzi sloupce „Úkoly“, „Rozpracovanost“ a „Hotovo“. Když je požadavek přidán do sekce „Rozpracované“, musí zůstat na nástěnce, dokud nedosáhne sloupce „Hotovo“ [9].

Čísla ukazují limity pro každý jednotlivý proces v pracovním postupu. Kanban board je nejdůležitějším prvkem přispívajícím k transparentnosti systému a jeho schopnosti snižovat plýtvání. Dělá celý proces transparentním pro tým, odhaluje problematická místa, fronty a hlavně veškerý „odpad“ ve smyslu nežádoucích stavů. Navíc z pohledu týmu přispívá ke kulturní změně, povzbuzuje spolupráci a diskusi, což je podpořeno ranými případovými studiemi. Výsledkem je neustálé vylepšování systému poskytující týmu pocit kontroly a odpovědnosti za celý projekt [9].

### 2. Omezení nedokončené práce

Kanban předepisuje označení sloupce, který se používá pro úkoly, na kterých se právě pracuje. Počet úkolů musí být omezen na základě aktuálního výkonu týmu, doby cyklu, počtu lidí zapojených do projektu, struktury týmu a dalších úvah. Nastavení optimálního počtu úkolů, které mohou být rozpracovány, je zásadní pro efektivitu systému a mělo by být nastaveno pro každý Kanban systém zvlášť [30]. Nastavení příliš vysokých limitů přetíží tým prací, která vytváří plýtvání a nečinné úkoly. Na druhou stranu, pokud jsou limity nastaveny příliš nízko, lidé mohou zahálet, což vede ke špatné produktivitě. Robert [33] navrhuje nejprve nastavit WIP (work in progress) tak, že se vezme počet členů v daném týmu a jeden se od něj odečte. Důvodem je zvýšení spolupráce a komunikace v rámci týmu [33].

Omezený WIP umožňuje týmu omezit práci, kterou má, na jeho aktuální kapacitu. Metoda omezení WIP je založena na teorii omezení, která zvyšuje schopnost týmů reagovat na potenciální problémy a pomáhá jim dokončit více položek za kratší dobu a méně defektů [33].



### 3. Řízení toku

Je rozdíl mezi tím, co uživatelé Kanban nazývají dobou cyklu. Jak uvádí Myslín [30] doba cyklu je čas měřený, kdy skutečná práce začíná na požadavku a končí, když je titul připraven k dodání. Na druhou stranu čas na dodání začíná okamžikem podání žádosti a končí doručením. Vést čas je z pohledu zákazníků důležitější, zatímco doba cyklu je způsob, jak vývojáři měří kapacitu procesu [30].

Hammaberg [16] navrhuje měřit dobu cyklu od okamžiku, kdy je položka umístěna do WIP a skončí ve sloupci „hotovo“, spíše než ve sloupci „nevyřízené položky“. Jako důvod pro to uvádí, že tým může často zamíchat požadavky ve sloupci „nevyřízené“, což ztěžuje přesné měření doby cyklu.

Měření doby cyklu umožňuje vývojářům měřit propustnost systému a poskytuje jim údaje o jeho výkonu. Umožňuje také efektivní alokaci zdrojů v týmu. Pokud například tým během určité doby nedokončí stanovený počet úkolů, může to znamenat, že limity WIP jsou nastaveny příliš vysoko nebo že je počet sloupců nesprávně uspořádán. Cykly by neměly být příliš dlouhé ani příliš krátké, aby umožnily dostávat pravidelnou zpětnou vazbu od klienta a poskytly týmu dostatek času na vytvoření vysoce kvalitního komponentu pro klientskou demonstraci. Umožňuje také vývojářům předpovídat dokončení každého úkolu, a tak efektivněji řídit očekávání klienta [16].

### 4. Výslovnost zásad procesu

Hlavním využitím kanbanové desky je řídit komunikaci v týmu a explicitně formulovat procesní zásady. Říká se, že je to klíčový faktor umožňující evoluční, kolaborativní změnu v systému Kanban. Dokud není proces pochopen a není pro jeho uživatele zjevný, je často obtížné nebo nemožné diskutovat o jeho zlepšení. Podle Kniberga a Skarina [23] bez toho bývá jakákoli diskuse o problémech emocionální, neoficiální a subjektivní. Zjednodušení procesů umožňuje vést racionální, empirickou a objektivní diskusi o relevantních otázkách.

Z toho důvodu musí celá organizace identifikovat příležitosti ke zlepšení systému a zapojit se do jeho implementace a řízení. Tím se učení stane hlavním cílem organizace [23].

## 5. Společné zlepšování (pomocí modelů a vědeckých metod)

Když týmy sdílejí porozumění teoriím o práci, pracovním postupu, procesu a rizicích, je pravděpodobnější, že přispějí prostřednictvím sdíleného porozumění a akcí, které vyplynuly z konsensu. Metoda Kanban navrhuje použití vědeckých metod k implementaci neustálých a evolučních změn. Běžně používané metody jsou [23]:

- **Teorie omezení (The Theory of Constraints - TOC):** Tato teorie říká, že procesy, organizace mohou být zranitelné, protože nejslabší osoba může nepříznivě ovlivnit výsledek. Tento jev lze ilustrovat následujícím idiomem: „Řetěz není silnější než jeho nejslabší článek“. Analytický přístup TOC se snaží identifikovat omezení systému, která brání systému dosáhnout více ze svého cíle. Tato metodika je úzce spojena s výrobním systémem společnosti Toyota. Jeho předpokladem je, že organizace mohou být měřeny a řízeny variacemi tří měřítek: propustností, provozních nákladů a zásob. Kromě toho mohou být omezení kategorizována jako vnitřní nebo vnější vůči systému. Externí omezení pocházejí většinou z vnějšího trhu, který může od systému vyžadovat buď příliš mnoho, nebo příliš málo propustnosti. V obou případech může systém vytvářet plýtvání buď nadprodukcí položek, nebo nevyužíváním plného potenciálu trhu k zisku. Vnitřní omezení mohou být: vybavení, lidé a zásady. Zařízení může být zastaralé, což uživatelům brání v efektivní práci. Lidé mohou také omezit systém tím, že budou mít mentální modely, které se samy stanou omezením [23].
- **Systém hlubokých znalostí:** Tuto teorii navrhl William E. Deming, který také obhájí transformaci stylu řízení organizace. Říká, že tato transformace vyžaduje pohled zvenčí kvůli zapojení lidí a připoutanosti k procesu. Tato filozofie obhájí změny, které může jednotlivec provést, aby změnil svou organizaci. Jsou to [23]:
  - a) Ocenění systému: člověk by se měl snažit porozumět procesům zahrnujícím dodavatele, výrobce a zákazníky zboží a služeb.
  - b) Znalost odchylek: rozsah příčin zahrnujících odchylky v kvalitě, použití statistického vzorkování při měření.
  - c) Teorie poznání: pojmy související s epistemologií vysvětlující znalosti, jejich původ, povahu a limity.

- d) Znalosti psychologie: pojmy zahrnující lidskou přirozenost, kognitivní procesy, motivaci a emoce. Podle Deminga člověk nemusí být odborníkem na nic z výše uvedeného, aby je mohl úspěšně aplikovat ve své organizaci. Také tvrdí, že pracovní síla je zodpovědná pouze za 15 % chyb, přičemž systém a management za 85 % [23].
- **Model štíhlé ekonomiky:** Model zahrnuje odstranění plýtvání, identifikaci a optimalizaci toků a hodnotových toků, implementaci odolnosti systému vůči změnám a úsilí. Všechny tyto modely se tak či onak doplňují a usilují o zlepšení organizace tím, že se o ní budou učit a neustále drobně vylepšovat proces zvaný Kaizen (tj. japonsky změna k lepšímu) [23].

### **3.5 Možnosti využití Azure DevOps pro projektové řízení**

Azure DevOps je sada souvisejících nástrojů, která softwarovým týmům umožňuje sledovat práci, spravovat kód, spouštět sestavení, nasazovat aplikace a řídit testy. Jedná se o centralizovanou, kompletní a hladce integrovanou sadu nástrojů, kterou lze použít téměř pro jakýkoli softwarový projekt. V minulosti byl známý pod jinými jmény, jako je Visual Studio Online (VSO) nebo Visual Studio Team Services (VSTS), a je evolucí Team Foundation Server (TFS) [2].

Funkce Azure DevOps lze klasifikovat podle členů týmu, procesů a nástrojů. Toto jsou hlavní funkce používané pro správu projektů Azure DevOps [2]:

#### **1. Lidé**

Lidé jsou srdcem každé organizace. Azure DevOps se hodí pro pracovní skupiny, týmy nebo tradiční přístupy správy. Lze jej propojit s Azure Active Directory (AD), synchronizovat s místními AD nebo dokonce sdílet a vyzývat „hosty“, aby pracovali v dané instanci. Všichni uživatelé podléhají řízení přístupu. Nakonec jsou uživatelé přidáni do týmů, kde mohou zobrazit práci, přístupový kód a spolupracovat s ostatními.

#### **2. Uživatelé**

Uživatele do Azure DevOps přidává správce. Po přidání si uživatel může nastavit svůj profil přidáním svého obrázku a dalších informací o sobě.

### 3. Azure AD/hosté

Pozvat hosty z jiných organizací, aby se připojili k dané práci a projektu, je vhodný způsob v rámci konzultace projektu nebo sdílení práce s externími zdroji.

### 4. Skupiny

Skupiny jsou primárním mechanismem řízení přístupu v Azure DevOps. Existuje několik vestavěných skupin, které uživatelům poskytují přístup k různým funkcím. Úrovně přístupu jsou podrobně popsány v dokumentaci, ale je poměrně jednoduché nastavit skupinu a přidat uživatele nebo dokonce další skupiny. Stránka nastavení organizace obsahuje správu skupin pro organizaci.

Stránka nastavení projektu má také místo pro nastavení oprávnění pro skupinu pro samotný projekt. Pokud existuje více projektů, lze ke každému nastavit různé úrovně přístupu. Tímto způsobem může mít jeden uživatel nebo skupina podle potřeby různé úrovně přístupu k mnoha projektům [2].

### 5. Týmy

Uživatelé mohou být zařazeni do jednoho nebo více týmů. Takto často funguje praxe a týmy omezují rozsah toho, co uživatel může vidět, když má vybraný konkrétní tým.

Tým může být zaměřen na portfolio, projekt, oblast nebo iteraci. Testovací tým může vidět pouze chyby, příběhy a úkoly na nástěnce Azure.

### 6. Procesy

Pracovní procesy vývoje se mezi organizacemi velmi liší. Azure DevOps má několik obecných šablon pro nástěnky Azure. Ty jsou přizpůsobitelné, ale s přizpůsobením přichází další složitost. Aby byl udržen hladký průběh procesu, lze si nastavit nevyřízené položky a pole pracovních položek podle potřeb organizace nebo projektu.

V souvislosti s procesy existují čtyři vestavěné typy pro nástěnky, od jednoduchých po složité. Jsou to typy základní, agilní, Scrum a CMMI (tzv. Capability Maturity Model Integration, tj. Model integrace vyspělosti schopností). Typy procesů se vztahují na projekt, ale týmy mohou mít své vlastní nástěnky. Například vývojářské týmy mohou potřebovat pouze vidět svou práci, zatímco manažer bude potřebovat vidět a monitorovat práci všech svých podřízených [2].

## 7. **Nevyřízené položky:**

Nevyřízené položky (backlogy) představují prostor, kde jsou spravovány pracovní položky. Jsou pro plánování produktů a funkcí a také pro organizaci práce. Backlogy lze rozdělit do úrovní podle potřeb. Typy problémů mapovaných na jednotlivé nevyřízené položky závisí na zvoleném typu procesu. Tři hlavní typy nevyřízených položek jsou následující [5]:

- a) **Nevyřízené požadavky** - podrobnější nevyřízené položky s problémy, položkami nevyřízených produktů nebo příběhy uživatelů.
- b) **Nevyřízené iterace** - nevyřízené položky na úrovni úkolu pro každou iteraci s časovým rozsahem.
- c) **Ostatní nevyřízené položky** - jako jsou chyby a překážky – zobrazení závisí na nastavení týmu.

### 3.5.1 **Další prvky Azure DevOps a jejich využití**

Nástroj Azure DevOps je komplexní nástroj, který usnadňuje projektovým manažerům práci s organizací projektu. Správce projektu má možnost zvolit různé nastavení pro každý projekt. V rámci projektu využívají projektoví manažeři mj. nástěnky pro transparentnost zadaných úkolů, iterace pro udržení tempa vývoje a dalších funkcionalit, které jsou popsány níže.

#### 1. **Nástěnky**

Slouží k organizaci a řízení práce. Pracovní předměty jsou karty na nástěnce, které jsou rozděleny do sloupců. Sloupce představují stav pracovních položek v nich obsažených. Po dokončení práce se přesouvají karty zleva doprava [4].

#### 2. **Iterace**

Iterace se používají pro plánování. Přestože Azure DevOps dokáže předpovídat práci automaticky, je obtížné zachytit každou pracovní položku předem smysluplným a přesným způsobem. To potenciálně činí předpovědi nepřesné.

Scrum se tento problém pokouší vyřešit pomocí cyklu plánování sprintu. Na druhou stranu, kanbanový přístup využívá kontinuální tah bez jakýchkoli specifických plánovacích cyklů. Ačkoli vydání nemusí spadat do iterace, může být užitečné používat je pouze jako plánovací cykly.

### 3. Oblasti

Pomocí oblastí lze práci logicky rozdělit, přiřadit k funkčním oblastem nebo dokonce vytvořit oblasti pro různé obory podnikání, jako jsou finance, řízení rizik, lidské zdroje atd. V rámci většího projektu jsou tyto oblasti výhodné, aby byla práce a nástěnky lépe zvládnutelné [4].

### 4. Nástroje

V Azure DevOps je mnoho nástrojů. Některé jsou pro vytváření a nasazení kódu. Další nástroje jsou pro správu zdrojového kódu. Následující nástroje jsou užitečné pro řízení projektů:

- **Oznámení:** Členové týmu mohou být upozorněni na téměř jakoukoli akci v Azure DevOps nebo se z nich odhlásit. Hlavními důvody pro upozornění jsou pracovní zadání a úspěch či neúspěch sestavení. Zúčastněné strany často chtějí být informovány, když dojde k nasazení nové verze. To je možné, i když zúčastněné strany nemají plný přístup k projektu.
- **Řídicí panely:** Řídicí panely se vytvářejí pro různé účely. Jeden styl řídicího panelu pro vývojáře může obsahovat odkazy na jejich pracovní položky a na kód. Řídicí panel pro testovacího inženýra může mít testovací pokrytí a widgety výsledků. Dále je možné využívat řídicí panel pro projektového manažera, který zobrazuje celkový průběh projektu a iterace.
- **Analýza:** Azure DevOps má několik typů integrovaných analytických sestav. Obsahuje regulační diagramy, kumulativní vývojové diagramy a vypalovací diagramy, které pomáhají porozumět vzorcům pracovních postupů. Grafy lze přidávat na řídicí panely nebo je jednotlivě prohlížet ze související funkce Azure Boards. Pomocí těchto datových zobrazení je možné získat široký pohled na portfolio projektů nebo filtrovat data přicházející do grafů a tabulek a zúžit tak konkrétní aspekty.

#### 3.5.2 Souhrn použití Azure DevOps

Zjednodušený proces pro Azure DevOps je následující [5]:

- založení jednoho nebo více projektů;
- vytvoření týmů;
- přidání uživatelů do týmů.

Uživatelé jsou zařazeni do skupin, které jim dávají oprávnění provádět určité věci, jako je aktualizace kódu, schvalování sestavení, změna procesů nasazení nebo vytváření a spouštění testů. Projekt používá šablonu procesu, která definuje, jak jsou pracovní položky organizovány. Mohou být umístěny do oblastí v jedné dimenzi a iterací v jiné. Oblasti slouží pro rozdělení aplikace a logické rozdělení práce. Iterace jsou pro účely plánování. Používají se k definování, kdy se na něčem bude pracovat.

Obecně platí, že jakmile je pracovní položka dokončena, vývojář odevzdá svůj kód. To může spustit sestavení, což jednoduše znamená, že nejnovější kód bude zkompileován nebo jinak připraven k nasazení. Může být nasazen manuálně nebo automaticky do jednoho nebo více prostředí. Testy lze rovněž spouštět manuálně nebo automaticky. Vývojáři mohou provádět manuální testy a plně je konfigurovat [5].

### **3.6 Alternativní nástroje pro projektové řízení**

Bez ohledu na zvolený přístup a strategii řízení pro implementaci projektu vždy existuje několik alternativních způsobů, jak projekt implementovat se stejnou základnou zdrojů a splňující stejné potřeby. Takové způsoby se nazývají „alternativy projektu“.

Projektová alternativa je další kombinací nákladů, harmonogramů, zdrojů a rizik projektu, která umožňuje dosáhnout stejných výsledků ve srovnání se základní linií projektu. Je to jeden nebo více způsobů za použití dalších nástrojů, jak vytvořit projekt a uspokojit obchodní potřeby při použití stejné zdrojové základny, a přitom pracovat v novém projektovém prostředí a čelit novým pracovním podmínkám. Příklady mohou být Jira, Trello, YouTrack a další.

#### **3.6.1 Jira**

Jira software je součástí rodiny produktů navržených tak, aby pomáhaly týmům všech typů řídit práci. Původně byl Jira software navržen jako nástroj pro sledování chyb a problémů. Dnes se však Jira vyvinul ve výkonný nástroj pro správu práce v rámci všech druhů případů použití, od správy požadavků a testovacích případů až po agilní vývoj softwaru [20].

Stále větší počet týmů se dnes vyvíjí více iterativně a Jira Software je centrem pro fáze kódování, spolupráce a vydání. Pro správu testů se Jira integruje s řadou doplňků, takže testování QA (Quality Assurance – tj. zajištění kvality) hladce zapadne do cyklu vývoje softwaru. Týmy mohou testovat efektivně a opakovaně. Týmy QA používají software Jira v rámci přizpůsobené obrazovky, pole a pracovních postupů ke správě manuálních a automatických testů.

Pro týmy, preferující agilní metodiku, poskytuje společnost Jira Software Scrum a Kanban nástěnky. Nástěnky jsou centra správy úloh, kde jsou úkoly mapovány do přizpůsobitelných pracovních postupů. Nástěnky poskytují transparentnost napříč týmovou prací a přehled o stavu každé pracovní položky. Funkce sledování času a zprávy o výkonu v reálném čase (grafy, zprávy sprintu, grafy rychlosti) umožňují týmům bedlivě sledovat jejich produktivitu v průběhu času [20].

Jira Software podporuje jakoukoli agilní metodiku vývoje softwaru. Jira Software lze nakonfigurovat tak, aby vyhovoval jakémukoli typu projektu. Týmy mohou začít se šablonou projektu nebo si vytvořit vlastní pracovní postup. Projekty Jira, známé také jako úkoly, sledují každou část práce, která musí projít kroky pracovního postupu až do dokončení. Přizpůsobitelná oprávnění umožňují správcům určit, kdo může vidět a provádět jaké akce. Když jsou všechny informace o projektu na místě, lze generovat zprávy ke sledování pokroku, produktivity a zajistit, aby nedošlo k žádnému pochybení.

Jira Software poskytuje nástroje pro plánování a plán, takže týmy mohou řídit zainteresované strany, rozpočty a požadavky v rámci funkcí od prvního dne. Jira se integruje s řadou nástrojů CI/CD, aby byla zajištěna transparentnost během životního cyklu vývoje softwaru. Když je připraven k nasazení, informace o stavu živého produkčního kódu se objeví ve vydání Jira. Integrované nástroje pro označování funkcí umožňují týmům zavádět nové funkce postupně a bezpečně.

V souvislosti s nástroji CI/CD (Kontinuální integrace - CI a kontinuální doručování - CD) je vhodné zdůraznit, že představují kulturu, sadu provozních principů a soubor postupů, které týmům vývoje aplikací umožňují doručovat změny kódu častěji a spolehlivěji. Implementace je také známá jako linie CI/CD [19].



CI/CD je jedním z nejlepších postupů pro implementaci týmů DevOps. Je to také osvědčený postup agilní metodologie, protože umožňuje týmům vývoje softwaru zaměřit se na splnění obchodních požadavků, kvality kódu a zabezpečení, protože kroky nasazení jsou automatizované.

V Jira Software mohou týmy vytvořit plán, který je spojen s každým projektem. Projektová mapa umožňuje týmům načrtnout dlouhodobý pohled na jejich práci a také sledovat a sdílet pokrok v rámci této mapy [19].

#### Výhody:

- Jira představuje způsob, jak zmapovat nápady a sdílet je danému týmu. Říká se tomu projektová (road) mapa a pomáhá sdílet velké plány a přidělovat jednotlivé úkoly.
- Tato aplikace má všechny funkce, které jsou třeba k udržení agilního vývoje. Pro jednotlivé příběhy lze generovat sestavy, jako jsou vyhodnocovací grafy, a pro správu týmu je možné sledovat pracovní vytížení.
- Integrace. Aplikace umožňuje snadnou integraci s dalším softwarem třetích stran. Například s Hipchatem a Slackem pro upozornění a komunikaci. K dispozici je více než 3000 dalších s různými potřebami a úkoly.
- Jira umožňuje uživatelům přizpůsobit své prostředí vytvářením a přizpůsobením různých prvků, jako jsou tabulky, formuláře, časové osy, sestavy, pole a další.
- Tento software se používá pro řadu různých úkolů různými typy uživatelů. Jsou to vývojáři, manažeři, projektoví manažeři nebo inženýři a další [19].

#### Nevýhody:

- Jira není nejlepší způsob, jak s ní pracovat na cestách. Uživatelé uvádějí, že ačkoli je webová zobrazení kvalitní, preferují práci pomocí aplikace.
- Integrace s jinými systémy je poměrně komplikovaná. Migrace projektu z Microsoft Team Foundation Server na Jira je velmi časově náročný proces.

- Přehledy nelze stáhnout. Grafické zprávy Jira nelze stáhnout jako obrázek. Použití jiných metod mění rozlišení obrázků, což má za následek nekvalitní obrázky.
- Jira má omezenou velikost nahrávání souborů, aby byl produkt rychlý. Nahrávání fotografie, videa nebo dokumentu o velikosti větší než 10 MB tedy není povoleno.
- Konfigurace některých částí Jira softwaru je nepřehledná a nové uspořádání je nepřehledné, informační architektura občas nedává smysl, zejména kombinovaná upozornění [19].

### 3.6.2 Trello

Trello je bezplatná aplikace pro produktivitu a správu úkolů, která byla vytvořena Fog Creek Software v roce 2011. Služba byla prodána společnosti Atlassian v roce 2017 a od října 2019 má více než 50 milionů uživatelů. Služba je dostupná na operačních systémech Windows, macOS, iOS, Android a také v prohlížeči.

Trello se odlišuje od tradičních aplikací v zobrazení úkolů. Tradiční aplikace umožňují odškrtnout dokončený úkol, kdežto Trello nabízí využití sloupců ke sledování průběhu činností v různých fázích. Každý úkol tak prochází jednotlivými fázemi, než dojde k označení jako dokončený [39].

Trello zdarma umožňuje uživatelům vytvářet libovolný počet nástěnek a sloupců. Kromě sledování pokroku lze sloupce nástěnky Trello použít také k zanechání poznámek pro ostatní uživatele, zdůraznění důležitých oznámení a k poskytování odkazů na online zdroje.

Mnoho organizací používá Trello ke správě velkých i menších skupinových projektů díky snadno srozumitelnému vizuálnímu rozhraní aplikace zobrazující různé fáze, ve kterých se úkoly nacházejí, a kdo na nich pracuje. Trello mohou také používat jednotlivci pro správu osobních a profesionálních plánů, organizování událostí a pro sledování pokroku. Trello je také oblíbeným organizačním nástrojem u studentů. Trello je celosvětově využíváno nejvíce v těchto oblastech [39]:

- sledování pokroku ve studiu jazyka;
- plánování svatby nebo narozeninové oslavy;

- přidělování úkolů zaměstnancům;
- vizualizace finančních cílů;
- sledování nákupů kryptoměn;
- organizování úkolů při rekonstrukcích domů;
- příprava na dovolenou nebo výlet;
- sdílení seznamu bucket s přáteli;
- vytvoření kampaně na sociálních sítích;
- sdílení sylabu se studenty;
- zobrazení a správa pracovních směn zaměstnanců a další.

Trello lze použít ke sledování základních úkolů pomocí jednoduchých sloupců „To-Do“, tj. úkoly ke splnění a „Completed“, tj. splněno nebo pro složitější úkoly a projekty s více fázemi.

Nástěnky Trello se mohou synchronizovat mezi různými verzemi aplikace Trello, takže lze kdykoli provádět změny svých úkolů ze smartphonu, tabletu nebo počítače.

Jak úkoly postupují, jednoduše je možné je přetáhnout do příslušných sloupců napravo, dokud není sloupec „Úkoly“ zcela prázdný.

Dalším běžným způsobem použití aplikací Trello pro správu osobních projektů je použití nástěnky jako týdenního kalendáře a vytvoření sloupce pro každý den v týdnu [39].

#### Výhody:

- Trello používá real-time. Členové mohou na projektech pracovat, aniž by došlo k narušení aktualizací. Jakmile je v systému provedena aktualizace, zobrazí se automaticky bez jakéhokoli zpoždění.
- Přidávání členů je snadné. Uživatel může přidat nového člena jednoduchým zadáním e-mailové adresy nového uživatele.
- Responzivní design. Ať už je Trello užíváno z přenosného počítače, smartphonu nebo tabletu, software se zobrazí bez problémů. To znamená, že pokud k němu uživatel přistupuje z jiného zařízení, než které obvykle používá, například z notebooku, bude moci Trello stále používat.

- Je vybaven systémem Kanban. Díky tomuto systému je nyní pro uživatele mnohem snazší vytvářet úkoly pro projekty a rozdělit je na ještě menší úkoly pomocí nástěnek.
- Notifikační systém je velice výhodný. To je jedna z nejsilnějších vlastností Trello, protože je vidět každý pohyb provedený v softwaru. O každé aktualizaci, provedeném komentáři a odstraněných úkolech se odešle e-mailové upozornění, zatímco uživatelům telefonu se odešle upozornění push.
- Uživatelsky přívětivý. Trello bylo navrženo tak, aby bylo jednoduché a snadno použitelné. Software tak může spravovat i jakýkoli začátečník [39].

#### Nevýhody:

- Trello spoléhá na online data. Jako online software se Trello spoléhá na dostupnost online dat, aby fungoval, což je také stejný problém, jaký má mnoho jemu podobných softwarů. Přístup k internetu je to, co jej pohání a umožňuje uživatelům dělat jejich úkoly. Takže když nejsou žádná online data, není možnost přístupu.
- Funkce komentářů Trello potřebuje vylepšení. Uživatelé jiného softwaru mohou psát komentáře a upravovat je. Trello umožňuje pouze komentáře, ale jejich úprava není možná. Pokud byl komentář již zveřejněn a uložen, poté jej není možné upravit. Možnou alternativou je vytvořit zcela nový komentář.
- Trello má omezené úložiště. Trello umožňuje přílohy, ale uživatelé bezplatné verze mají povoleno pouze 10 MB na jeden soubor, což je poměrně málo. Toto omezení je možné odstranit zakoupením placené verze aplikace Trello.
- Je vhodný pro menší projekty. Lidé, kteří řeší menší projekty a týmy, mohou Trello rozhodně velice výhodně využívat. Problém však může nastat, pokud tým roste a projekt se stává složitějším [39].

### **3.6.3 You Track**

YouTrack je produktem společnosti JetBrains a kombinuje komerční nástroj pro sledování chyb založený na prohlížeči, systém sledování problémů a schopnosti projektového řízení do jednoho komplexního balíčku. Systém nabízí funkci vyhledávání problémů na základě dotazů s automatickým dokončováním a

umožňuje uživatelům manipulovat s problémy v jednotlivých modulech, přizpůsobovat moduly problému a vytvářet vlastní pracovní postupy [3].

Mezi aktualizace YouTrack patří widget pro sledování času v souvislosti s řídicími panely, řazení problémů podle počtu komentářů, možnost jedním kliknutím zobrazit nevyřešené problémy v zobrazení „Seznam problémů“ a vylepšení jak znalostní báze, tak integrace LDAP (Lightweight Directory Access Protocol).

YouTrack umožňuje sledovat a spravovat problémy od okamžiku, kdy jsou nahlášeny, až do okamžiku, kdy jsou opraveny. Pomocí vestavěného editoru obrázků lze snadno vytvářet problémy, přidávat štítky, sledovače, upravovat vlastní pole, nastavovat prioritu a upravovat přidané přílohy. Při vytváření problému není nutné se starat o zavření karty – YouTrack uchovává koncepty.

Agilní nástěnky v YouTrack jsou navrženy tak, aby pomáhaly týmům sledovat širokou škálu procesů agilního řízení projektů. Je tak možné vytvářet desky Scrum, Kanban nebo Scrumban [3].

### **3.7 Komunikační nástroje**

K úspěchu projektu přispívá mnoho věcí, ale jednou z nejdůležitějších je dobrá komunikace s klienty. Každý projekt má svůj soubor jednotlivců a okolností, které určují, které formy komunikace budou fungovat nejlépe, ale pro všechny projekty je cíl stejný: pomoci všem v týmu činit správná rozhodnutí. Rozhodnutí vyžadují otevřené komunikační kanály, aby informace mohly proudit tam a zpět podle potřeby.

Možné příklady těchto komunikačních nástrojů v souvislosti s projektovým řízením jsou uvedeny v následujících kapitolách.

#### **3.7.1 Google Suite**

Google Workspace (dříve známý jako Google Apps a později Google Suite) je sbírka cloud computingu, nástrojů pro produktivitu a spolupráci, softwaru a produktů vyvinutých a nabízených společností Google. Poprvé byl spuštěn v roce 2006 jako Google Apps a v roce 2016 byl přejmenován na Google Suite (G Suite).

Google Workspace se skládá z Gmailu, Kontaktů, Kalendáře, Meet a Chatu pro komunikaci a dalších [15].

V závislosti na edici může Google Workspace obsahovat také digitální interaktivní tabuli Jamboard a možnost zakoupení doplňků, jako je telefonní služba Voice. Edice pro vzdělávání přidává výukovou platformu Google Classroom a dnes nese název Workspace for Education.

Zatímco většina těchto služeb je jednotlivě zdarma dostupná pro spotřebitele, kteří používají své bezplatné účty Google (Gmail), Google Suite přidává podnikové funkce, jako jsou vlastní e-mailové adresy v doméně (např. @vašespolečnost.com), což je možnost neomezeného úložiště, další nástroje pro správu a pokročilá nastavení, stejně jako nepřetržitá telefonická a e-mailová podpora [15].

V souvislosti s projektovým řízením lze vytvářet dynamické plány projektů a Ganttovy diagramy, aby bylo možné spravovat své projekty, úkoly a termíny. Členové týmu na celém světě mohou aktualizovat své postupy přímo v tabulce, která je vždy aktuální.

Pokud je třeba sdílet informace se svým projektovým týmem, je vhodné vytvořit seznam e-mailů „Skupin“. Po přidání týmu do skupiny je možné zasílat aktualizace a zdroje všem najednou pomocí jedné e-mailové adresy atd.

Dále je výhodné vytvářet sdílené kalendáře, aby byly k dispozici všechny aktuální informace: projektové kalendáře pro sledování milníků, termínů a připomenutí; týmové kalendáře pro schůzky, školení a dovolené atd. [15].

### **3.7.2 Nástroj Slack**

Slack je aplikace zasílání zpráv, která spojuje uživatele prostřednictvím informací, které potřebují. Představuje tak propojování uživatelů, kteří pracují jako jeden tým. Slack pomáhá pracovat propojenějším a flexibilnějším způsobem [35].

Slack usnadňuje přístup ke kolegům a podporuje tak asynchronní práci. Když je práce organizována v kanálech, bez ohledu na polohu, časové pásmo nebo funkci, může uživatel stále přistupovat k informacím, které potřebuje, ve svém prioritním čase, aniž by musel koordinovat plány.

Ve Slacku má každý v organizaci přístup ke stejným sdíleným a vyhledávaným informacím. Když týmy spolupracují ve vytyčených informačních kanálech, lze informace sdílet se všemi najednou, což pomáhá udržet týmy v souladu a rychleji se rozhodovat [35].

### 3.7.3 Microsoft Teams

Microsoft Teams je výkonná platforma pro řízení projektů. Jednoduše nabízí vytvoření týmů, využití chatů, vedení videokonferencí a další. Marketingový tým BrightWork kombinuje Microsoft Teams s webem projektu SharePoint pro komunikaci a spolupráci.

Microsoft Teams, součást sady Microsoft 365, která byla spuštěna na začátku roku 2017. Aplikace nabízí: [29]

- chat a zprávy v reálném čase;
- videokonference a schůzky až pro 300 osob;
- živé akce až pro 20 000 lidí;
- cloudové volání;
- ukládání dokumentů;
- společné úpravy v reálném čase;
- integrace s aplikacemi a konektory třetích stran;
- vyhledávání;
- veřejné nebo soukromé kanály;
- správa úkolů;
- velké možnosti zabezpečení a ochrany soukromí.

Ekonomické přínosy Microsoftu Teams mohou být: [29]

- Teams snižuje počet a trvání schůzek a služebních cest.
- Organizace mohou pomocí Teams snadno nahradit řadu hardwarových a softwarových komunikačních řešení, jako jsou videokonference a místní telefony.

- Díky snadnému přístupu k informacím na jednom místě a menšímu plýtvání časem přepínáním mezi aplikacemi ušetří Teams pracovníkům až čtyři hodiny týdně.
- Teams umožňuje vzdálenou práci s různými možnostmi spolupráce a sdíleným úložištěm dokumentů. To zase urychluje rozhodování a zvyšuje produktivitu.

### **3.8 Další přístupy projektového řízení využívané IT leadery v praxi**

Mezi některé další přístupy v souvislosti s projektovým řízením v IT praxi jsou [10]:

#### **1. Extrémní programování**

Extrémní programování (XP) je jednou z agilních metodologií vytvořených pro projekty vývoje softwaru, která obhájí časté „vydání“ v krátkých vývojových cyklech, jejichž cílem je zlepšit týmovou produktivitu a kvalitu softwaru. Jádrem XP se skládá z pěti hodnot, čtyř vývojových procesů: kódování, testování, naslouchání a navrhování – a více postupů vývoje softwaru, které lze provádět buď společně, nebo samostatně. Je vhodné pro malé týmy vyvíjející IT projekty s dynamicky se měnícími požadavky.

#### **2. Adaptivní projektový rámec**

Adaptivní projektový rámec (APF) vzešel z myšlenky, že většinu IT projektů nelze řídit pomocí tradičních metod projektového řízení. APF je iterativní, na klienta zaměřené a adaptivní plánování projektu navržené tak, aby pomohlo projektovým manažerům reagovat na neočekávané změny [10].

#### **3. Extrémní projektové řízení**

Metoda extrémního projektového managementu (XPM) umožňuje měnit projektový plán, rozpočty a konečný výsledek v jakékoli fázi projektu tak, aby vyhovoval měnícím se potřebám. Jinými slovy, nabízí přístup řízený lidmi, nikoli procesem řízení, což znamená, že v XPM lidé přizpůsobují modely tak, aby vyhovovaly projektu, ne jinak. Je nejvhodnější pro složité projekty s nejistými a nepředvídatelnými faktory.



#### **4. Kritický řetězec:**

Kritický řetězec představuje sled činností s nejkratším časovým předstihem, který zohledňuje omezení zdrojů (lidé, vybavení) a přesouvá část implicitních rezerv do tzv. nárazových činností. Například vyrovnávací paměť projektu nastavená mezi poslední aktivitou v řetězci a termínem projektu zajišťuje, že zpoždění nebude mít vliv na termín dodání projektu. Mezi další vyrovnávací paměti patří vyrovnávací paměti pro napájení a zdroje. Tato technika klade důraz na stanovení priorit, analýzu závislostí a optimalizaci časových nákladů. Je nejvhodnější pro složité projekty s omezenými zdroji [10].

#### **5. Metodika řetězce událostí**

Tato metodika představuje přístup k modelování nejistot, který je založen na identifikaci vztahů mezi událostmi projektu a zaměřuje se na události, jež mohou ovlivnit harmonogram projektu. Je považován za efektivnější přístup k plánování projektů než metody kritické cesty a kritického řetězce. Slouží ke správě očekávaných i neočekávaných událostí, které ovlivňují plánování projektu [10].

## 4 Vývoj projektu Online kasino

Tato část se zabývá identifikací problémů projektového řízení v projektu Online kasino, který byl zahájen v dubnu 2020. Tento projekt realizuje firma Czech Brain Data, ve které je autor, v době psaní této diplomové práce, zaměstnaný na pozici business analytik.

V úvodní části je stručně popsána firma Czech Brain Data. Hlavním obsahem této kapitoly je popis vývoje sdělovacích/komunikačních technik projektového řízení v průběhu realizace projektu Online kasino. Projektové řízení je rozděleno do několika životních etap. Každá etapa obsahuje analýzu, kritické zhodnocení a případné návrhy, které vycházejí z aktuálních standardů a metod projektového řízení pro vývoj softwaru.

Cílem kapitoly je vytvoření doporučení pro nastavení komunikačních a sdělovacích nástrojů tak, aby řízení projektu bylo efektivní, srozumitelné a přehledné pro všechny členy týmu, kteří pracují na projektu Online kasino.

### 4.1 Představení firmy Czech Brain Data

Firma Czech Brain Data byla založena v polovině roku 2011 současným majitelem jako softwarová firma specializující se na hazardní průmysl. Počet zaměstnanců se za 10 let své existence několikanásobně zvýšil na stávajících 25 kmenových zaměstnanců a 7 externistů. Jelikož firma do vypuknutí pandemie COVID-19, tzn. března 2020, spolupracovala se střední odbornou školou v místě sídla firmy, většinu zaměstnanců tvoří absolventi této školy, a i z tohoto důvodu je silné zastoupení zaměstnanců v kategorii do 30 let.

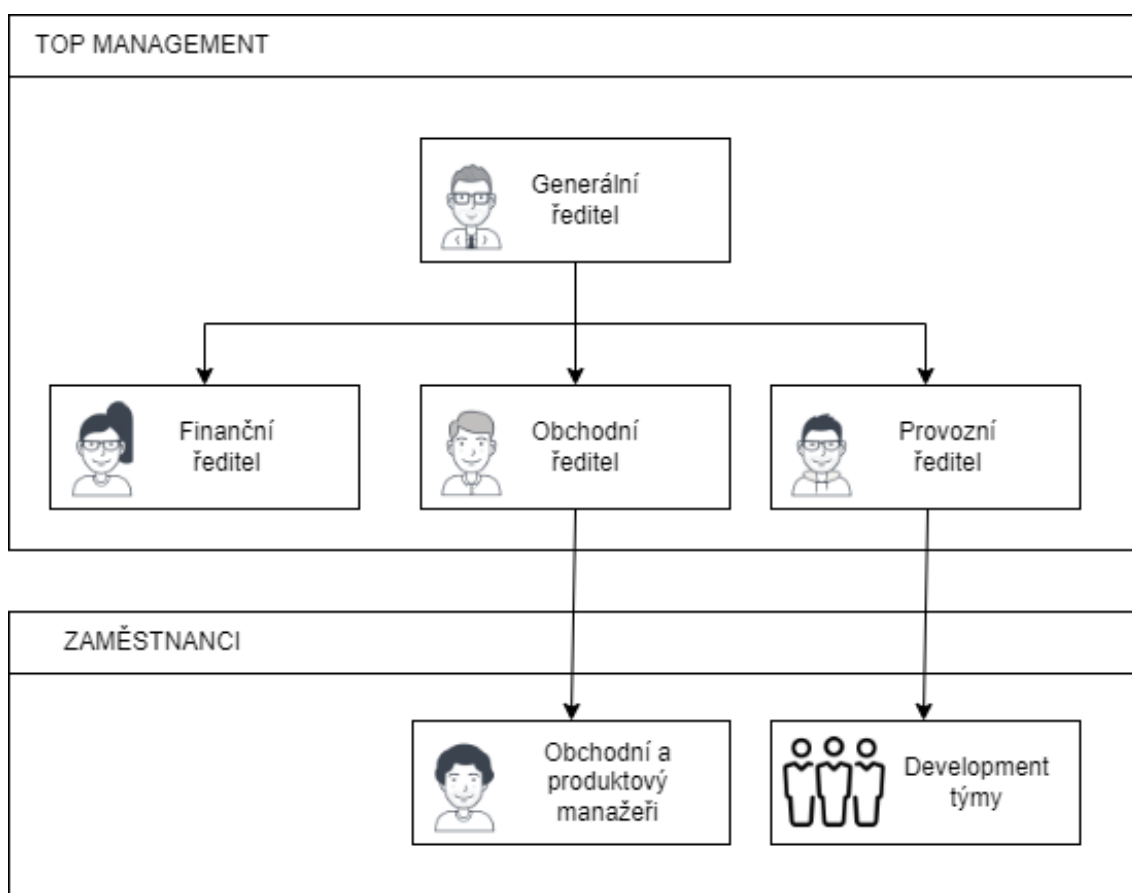
Firma si zakládá, v rámci interních vztahů, na tvorbě přátelského kolektivu založeného na neformálním mezilidském vztahu a přívětivém pracovním prostředí. Svou filosofii uplatňuje firma také u externích vztahů, a to především pravidelným pořádáním odborných přednášek pro své zákazníky a obchodní partnery, ale také pravidelnou účastí na zahraničních konferencích.

Jak bylo uvedeno již v úvodu, firma se specializuje na hazardní průmysl, konkrétně na neherní IT řešení pro herní průmysl. Firma má ve svém portfoliu několik produktů umožňujících provoz heren a kasin, a to včetně možnosti online

gamblingu. Jelikož firma aktivně monitoruje a reaguje na podmínky regulátora, všechna její řešení jsou plně v souladu s legislativou pro daný trh.

#### 4.1.1 Struktura firmy

Z organizačního hlediska se společnost snaží být co nejvíce srozumitelná. Hlavní pozorovatelné stupně jsou top management a zaměstnanci. Za svého hlavního nadřízeného pracovníka považuje každý zaměstnanec jednoho člena z top managementu, přičemž se všemi ostatními členy firmy je obvykle v kolegiálním vztahu.



Obrázek 1: Firemní struktura

Zdroj: vlastní zpracování

Obrázek 1 ilustruje organizační strukturu firmy Czech Brain Data, která byla platná od 1.1.2021.

- Chief Executive Officer – zakladatel a majitel firmy. Monitoruje celkový chod firmy.

- Chief Financial Officer – finanční ředitelka, do její kompetence spadá především ekonomická správa firmy.
- Chief Sales Officer – obchodní ředitel, monitoruje a zprostředkovává obchodní příležitosti.
  - Area Manager – přímý pracovník obchodu.
  - Product Manager – stará se o svěřený produkt, vyhledává produktové příležitosti na trhu, účastní se jednání v podobě technického odborníka.
- Chief Operation Officer – provozní ředitel. Stará se o denní chod firmy, účastní se významných jednání, v projektech mívá rozhodující slovo.
  - Každý Development tým má svěřený svůj produkt/projekt, na kterém pracuje. Tým se zpravidla skládá z několika členů:
    - projektový manažer/Product Owner;
    - analytik;
    - backend vývojář;
    - frontend vývojář;
    - tester.
  - V rámci struktury firmy existují dále oddělení, kde se pohybuje vždy jeden vedoucí pracovník:
    - oddělení vývojářů;
    - oddělení testerů;
    - oddělení projektových manažerů.
  - V praxi je považována jako vedoucí osoba, která je zodpovědná za svěřený produkt/projekt. Vedoucí pracovník oddělení je považován za mentora a řešitele odborných problémů daného oddělení.
  - V každém projektu není nutné mít zastoupení všech zmíněných členů uvedených výše. Dle povahy a typu projektu bývá analytik nahrazován projektovým/produktovým manažerem, určité projekty vyžadují pouze backend nebo naopak pouze frontend vývojáře.
  - Obecně je možné označit 3 povinné členy:
    - Zadavatel (v podobě projekt/produkt manažera nebo analytika).

- Řešitel (backend/frontend vývojář ve spolupráci s testerem).

## **4.2 Projekt Online kasino**

Dopad restrikcí a vládních omezení způsobených pandemií COVID-19 ovlivnilo chod firmy a změnilo představy a vize top managementu o budoucích projektech. Tato nově vzniklá a nikým neočekávaná situace paradoxně urychlila rozhodnutí zahájit zcela nový projekt. Tento projekt byl v době rozhodnutí (duben 2020) pro mnohé zaměstnance nepředstavitelný (ať už svým rozsahem či cílovým trhem), avšak pro firmu představoval určitou naději.

## **4.3 Projektové řízení ve firmě Czech Brain Data**

Řízení projektů celé firmy vedla, do zahájení projektu Online kasino, pouze jedna projektová manažerka. V portfoliu firmy převládají tzv. krabicová řešení, a proto je nutné zdůraznit, že stav a povaha realizovaných projektů a produktů tuto variantu umožňovala.

Většina projektů je řízena vodopádovým přístupem, kdy každý požadavek prochází procesem dílčích aktivit:

1. Přijetí požadavku – projektová manažerka je informována obchodním oddělením o novém požadavku. Tento požadavek projektová manažerka zařadí do projektového plánu (v prvotní fázi za účelem vytvoření pre-sale analýzy).
2. Pre-sale analýza – prostá analýza, zpracovaná analytikem, určená jako podkladový materiál pro vytvoření cenové nabídky. Analýza zachycuje důležité milníky, odhaluje potenciální rizikové oblasti, a především definuje předpokládaný rozsah a náročnost prací.
3. Cenová nabídka – na základě pre-sale analýzy je zpracována cenová nabídka, kterou obdrží zákazník. Projektová manažerka poskytuje obchodníkovi firmy informace o předpokládané náročnosti a rozsahu požadavku. Dále projektová manažerka předává informaci o termínu realizace, čímž je definován orientační termín nasazení u zákazníka.

4. Funkční analýza – jestliže zákazník odsouhlasí cenovou nabídku, analytik zpracuje komplexní funkční zadání. Funkční analýza je výchozím zadáním pro vývojáře a testery.
5. Zařazení do verze – vývojář dle funkční analýzy implementuje změny, které následně tester otestuje.
6. Nasazení u zákazníka – po finálním uvolnění verze ze strany testerů, kde podmínkou je bezchybné chování stávajících i nových funkcionalit, naplánuje projektová manažerka nasazení nové verze u zákazníka. Samotné nasazení provádějí systémoví administrátoři.

Případná změna chování po nasazení (vyžadovaná zákazníkem) musí projít celým procesem popsaným výše. Časový rozsah jedné verze je přibližně 2-3 měsíce.

Do dubna 2020 bylo řízení firmy směrem k zaměstnancům organizováno následovně:

- Pravidelné osobní schůzky celé firmy vždy v pondělí dopoledne v sídle firmy.
- Součástí osobní schůzky bylo mj. představení projektového plánu na nadcházející týden včetně přidělení řešitelů.
- Projektový plán byl zpracován v podobě PDF formátu a uložen na sdílené úložiště v Repos Azure DevOps. Tzn. zaměstnanec, který se nemohl zúčastnit osobní schůzky, byl informován individuálně, případně svůj rozvrh prací a priorit našel v tomto souboru.
- Součástí PDF souboru bylo kromě názvu úkolu a řešitele také odkaz do Board Azure DevOps, kde byl pro daný úkol vytvořený Work Item (Task/Bug - dle povahy úkolu). Tento Work Item sloužil především jako zadání pro řešitele a prostor, kde je možné vykazovat a sledovat náklady na zpracování úkolu (především čas strávený vyřešením úkolu).
- U méně komplikovaných úkolů bylo zadání popsáno přímo v popisu Work Item, u komplexních požadavků bylo zadání popsáno v textovém dokumentu s příloženými wireframes a diagramy. Textový dokument byl přiložen jako příloha Work Item

Zodpovědná osoba	Priorita	Systém	Release/ Termín	Úkol	Zadání	Azure	Zákazník	Poznámka	
Martin	kontinuálně	cbSAS	kontinuálně	bugfix	dle aktuálních požadavků	dtto			
	kontinuálně	cbCAP	kontinuálně	bugfix	dle aktuálních požadavků	dtto			
	kontinuálně	cbJackpot	kontinuálně	bugfix	dle aktuálních požadavků	dtto			
	3	ET Server		Úprava ET serveru	Nutná oprava do stávajícího serveru, aby respektoval na ET3 "okénko" pro povinné údaje. Ve spolupráci s MartinemP.				čekáme na finální grafické podklady od EGT. Úpravy MC předány do testů
	čekáme na pokyn z obchodu	cbJackpot		Nová grafika - haléřové pole + historie padlých jackpotů		<a href="#">task 2632</a>			
	1	cbSAS		EDP	Periodicky proaktivně stahovat ze SAS čteček stavy IN/OUT/BET/WIN	<a href="#">feature 2652</a>			
		cbCAP		Topologie	Ve spolupráci s Davidem sjednocení topologie podle dohody				
		cbSAS		Nový vývoj	Refaktoring SAS serveru + převod komunikace s ET3 na TCP + zabezpečení				
	2	cbSAS		Rulety	Dokončit ostré spuštění rulet Synot	<a href="#">task 2198</a>			
Ondra D.	1	cbCAP/cbCardin		Server/Cluster	migrace	<a href="#">task 1611</a> <a href="#">task 1610</a>		čekáme na stanovení termínu zákazníkem	
	2	cbCAP/cbCardin	25. 2. od 8:00	Server	Migrace na virtuál	<a href="#">task 2350</a>		hotovo	
	3	cbCAP/cbRAS		Server	Příprava CAP serveru vč. GAPI/RAS				
	4	190		Server	Migrace na virtuál				
		cbRAS/GAPI		Údržba serverů	nasazení nových certifikátů			ve spolupráci Láďa, Hanka, OndraP	

Obrázek 2: Týdenní projektový plán v PDF

Zdroj: vlastní zpracování

#### **4.4 Zahájení projektu a složení týmu**

Nastavené projektové řízení firmy do určité míry ovlivnilo řízení projektu Online kasino při jeho zahájení. Projekt má za cíl zabezpečit firmu i v období, kdy není možnost provozovat hazardní hry formou landbase, tzn. projekt je určený pro online gambling. Pro realizaci projektu vznikl zcela dedikovaný vývojový tým, který se v době zahájení projektu skládal z pozic:

- Produkt Owner – z důvodu minimálních zkušeností s obdobným projektem z řad vlastních zaměstnanců byl angažován externí pracovník, který má bohaté zkušenosti s vývojem řešení pro online gambling. V týmu vystupuje jako zodpovědná osoba za výsledný produkt, definuje oblasti, které je nutné vyřešit, obecně koordinuje celý projekt.
- Projektová manažerka – stávající projektová manažerka firmy, zajišťuje lidské zdroje napříč projekty (s ohledem na plynulost projektu Online kasino).
- Analytik – úzce spolupracuje s Product Ownerem a zpracovává jeho vize a představy do podoby zadávací dokumentace určenou pro vývojáře a testery. Před sestavením týmu a zahájením realizace projektu připravil základní technickou specifikaci budoucího řešení, díky které bylo možné vhodně vybrat budoucí členy týmu a definovat všem členům práci pro první týdny vývoje.
- Grafička – po prvotní analýze trhu vytváří nefunkční model projektu, který slouží jako doplňující materiál k výstupu od analytika a definuje jasná pravidla pro vizualizaci produktu.
- Backend vývojář – v úvodní části jsou zapojeni do projektu 2 backend vývojáři. První pracuje na vývoji vlastního online řešení gamblingu, druhý se podílí na integraci se systémy třetích stran.
- Frontend vývojář – obdobně jako u backend vývoje, i zde jsou zapojeni 2 (frontend) vývojáři. Z pohledu vizualizace je potřeba připravit samotné online řešení pro koncové uživatele a také řešení pro správu obsahu a řízení denního provozu ze strany provozovatele.



- Tester – z počátku byl do projektu zařazen pouze vedoucí tester, a to z důvodu konzultace navrhovaného řešení. Jakmile byly vytvořeny první funkční celky, které bylo možné otestovat, do projektu byl přizván i zbytek týmu testerů.

Cíl pro první týdny až měsíce byl poměrně prostý – vytvořit takové řešení, které bude možné prezentovat odborné zkušebně, která v ideálním případě následně vystaví odborný posudek (certifikaci). Jelikož projekt Online kasino vznikl na základě iniciativy top managementu firmy, bylo stěžejní dosáhnout tohoto cíle v co nejkratším čase a s minimálními náklady. V případě komplikace s certifikací a možným vetováním produktu ze strany regulátora by vznikl nenávratný náklad na tento vývoj.

Projekt organizačně řídil Product Owner, a to především v podobně pravidelných schůzek, které se konaly 3x týdně:

1. Pondělí – Stand up – shrnutí předešlého týdne, vyhodnocení dokončených prací, představení priorit pro aktuální týden.
2. Středa – Status – průběžná kontrolní schůzka, kde tématem bývaly zpravidla nejasné otázky a případné další nesrovnalosti.
3. Pátek – UI/UX – schůzka primárně určená pro grafičku a frontend vývojáře, kde byly prezentovány již hotové vizualizační části a byla diskutována podoba rozpracovaných částí.

V této fázi projektu byly hlavní komunikační nástroje aplikace Slack (pro psanou komunikaci) a Google Meets (pro online schůzky). Aplikace Slack nabízí široké spektrum možností v podobě doplňků, přívětivého API rozhraní apod. Firma využívala free verzi, která je omezena pouze na 10 000 zpráv napříč všemi chaty, kanály apod. Nezřídka se stalo, že důležité informace zaznamenané v konverzaci byly nenávratně smazány.

Samotná organizace týmu probíhala na základě zápisu z jednání, tzn. z každé schůzky vytvořil Product Owner zápis do textového dokumentu, který byl umístěný na Google Drive (oproti ostatním dokumentům firmy, které byly umístěny v Repos Azure DevOps). Každá schůzka začínala revizí předchozího zápisu ze schůzky.

Pokud byla předešlá schůzka (zápis za schůzky) dlouhá, revize tohoto dokumentu v rámci aktuální schůzky byla této délce úměrná. Úkoly pro jednotlivé členy týmu nebyly nijak strukturované, byly rozděleny pouze selektivně dle prostředí – backend/frontend. Součástí dokumentu byly mj. všechny dohody a poznatky, které byly diskutovány v rámci schůzky. V úvodní fázi realizace projektu byl tento přístup dostačující, jelikož tým měl jasně nastavený cíl a úkoly byly jasné a srozumitelné.

#### **4.5 Aplikování Backlog a Sprint v Azure DevOps**

Díky získanému odbornému osvědčení mohla firma zahájit jednání s partnery o možnosti vzájemné integrace a plánované řešení následně prezentovat potencionálním zákazníkům. Obchodním zástupcům se podařilo velmi rychle získat několik zákazníků z českého a slovenského trhu, a tak již po necelých 6 měsících vývoje byly podepsány první obchodní smlouvy.

Podepsáním smluv došlo k zásadním změnám ve vývoji a v jeho řízení. Řešení bylo primárně vytvářeno a certifikováno pro český trh. V této době byly podepsány smlouvy pro český a slovenský trh, čímž se zvýšil tlak na kvalitní evidenci úkolů a určování priorit mj. pro jednotlivá prostředí. Organizace projektu se tak dostala do situace, kdy s přibývajícimi komplikacemi během vývoje podpořené nedostatečnou zadávací dokumentací (jako zadání byla často považována část textu v zápisu ze schůzky) se začaly schůzky prodlužovat, vývoj přestal mít potřebný spád a začalo docházet k nepřehledné situaci mezi již hotovými a nedokončenými úkoly. V tomto období nezdědka nastávaly situace, kdy vývojáři považovali chybu za opravenou, jelikož chybu hledali na jiném prostředí, než na kterém byla nalezena.

Zásadní nedostatky přístupu popsaného výše:

- Velké množství informací, které není nijak strukturovaně organizováno.
- Díky nestrukturovaným informacím dochází k prodlužování týdenních schůzek.
- Jelikož informace jsou strukturované a schůzky se prodlužují, zbývá méně času na samostatný vývoj.

- Nedostatečná zadávací dokumentace, která komplikuje především práci testerů (nejsou popsány alternativní scénáře, které vývojář v lepším případě vyřešil bez dalšího popisu, v horším případě nejsou realizovány vůbec).
- Hlášené chyby neobsahují všechny potřebné informace, především informaci o prostředí, na kterém byla chyba zjištěna.

Vedení projektu zastoupené Product Ownerem začalo nastalou situací okamžitě řešit. Jelikož v meziobdobí se stal Product Owner kmenovým zaměstnancem firmy a jeho hlavní role byla člen top managementu (provozní ředitel), preferoval přenechat řízení projektu Online kasino jinému členovi. Byla svolána diskuze mezi členy týmu, kde měl každý možnost sdělit svůj názor a přispět tak ke zlepšení organizačního řízení. Závěrem bylo nové složení týmu, na hlavních změnách a inovacích v projektovém řízení projektu se podíleli 2 stávající členové – analytik a vývojář (dle diskuze projevíli velký přehled v problematice podpořený praktickými zkušenostmi). Nová struktura týmu:

- Supervisor – bývalý Product Owner – role vznikla z důvodu nového Product Ownera, do správy supervisora nepatří denní chod vývojového týmu, avšak stále určuje priority jednotlivým úkolům (po vzájemné konzultaci s novým Product Ownerem, který má k dispozici přehled volných kapacit členů). V nejasných situacích má nejvyšší rozhodovací slovo.
- Product Owner – bývalý analytik – role analytika, tzn. tvorba zadávací dokumentace, byla rozšířena o správu backlogu a sprintů v DevOps, tzn. plánování, koordinace a zajišťování kapacit jednotlivých členů spravuje tato osoba. Product Owner vytváří týdenní plány, sleduje dodržování nastaveného plánu, zajišťuje komunikaci s externími partnery a vytváří dokumentaci potřebnou pro vývoj funkcionalit.
- Backend vývojář – na vývoji backendové části se podílejí pravidelně 3-4 vývojáři.
- Frontend vývojář – oba vývojáři jsou alokováni převážně na vizualizační část řešení.

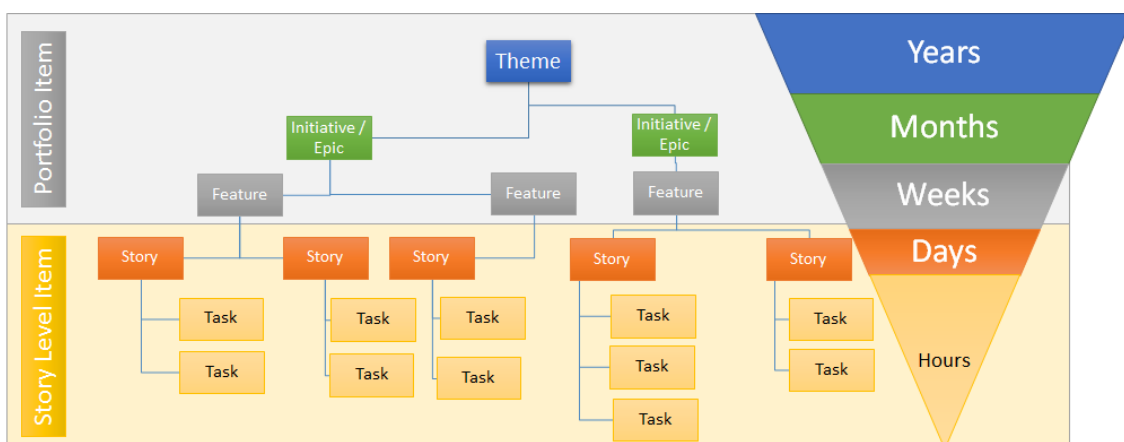
- Tester – z důvodu více prostředí a častých změn byl dosazen celý testovací tým. Dále se využívá služeb externích kontraktorů pro výkonnostní a penetrační testování.

Oproti původnímu týmu není součástí týmu grafička – noví zákazníci preferují vlastní grafická studia – a projektová manažerka. Ta se schůzek účastní i nadále, avšak nyní je to z důvodu informovanosti a možné diskuzi o využití dalších členů firmy. Již nezajišťuje kapacity členů týmu Online kasino.

Zásadní změnou bylo využívání nástrojů určených pro projektové řízení dostupných v již dříve používaném Azure DevOps. V první řadě byl vytvořen backlog, tzn. soupis všech požadavků, které je potřeba vytvořit k úspěšnému dokončení projektu. Tento soupis nebylo možné vytvořit ve 100 % rozsahu, a tak jsou některé požadavky do backlogu doplňovány později. Dále se začala více využívat agilní metodika řízení, ze které byla pro tento projekt vybrána většina metod z přístupu Scrum. Soupis požadavků v backlogu byl definován dle stromové struktury definované metodikou Scrum [41]:

- Epic – komplexní vlastnost produktu, rozsah vývoje je v řádu měsíců. Příklad v tomto projektu je „Implementace platebních metod“ [1].
  - Feature – konkrétní funkcionality pro danou vlastnost, rozsah vývoje je v řádu týdnů. Příklad v tomto projektu je „Integrace platební brány“.
    - User Story – popis funkcionality z pohledu konkrétního uživatele. Pohledů může být několik. Rozsah vývoje je v řádu dní. Příklad v tomto projektu je „Navržení grafického rozhraní pro platby kartou“ nebo „Implementace API platební brány“ do stávajícího řešení“ [36].
      - Task/Bug – jedná se o dílčí úkol, který musí být splněn, aby user story přinášelo očekávanou hodnotu. Rozsah vývoje je v řádu hodin. Příklad v tomto projektu je task „Vytvoření číselníků bank“ nebo „Vytvoření tlačítka pro vklad“. Typ bug využívají především testeři v případě,

kdy naleznou chybu nebo odlišnost oproti zadání celého user story [37].



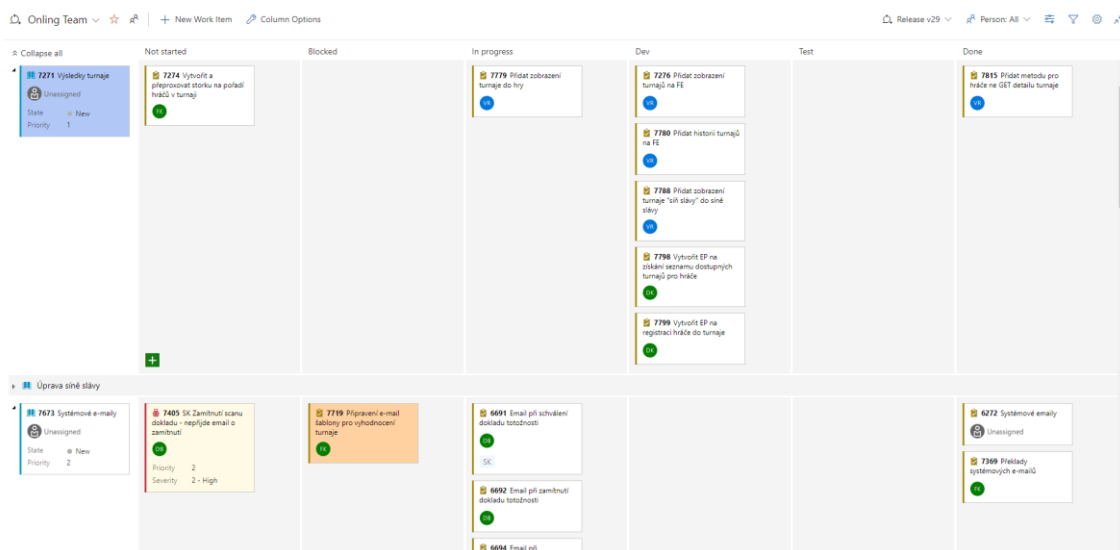
Obrázek 3: Stromová struktura backlogu

Zdroj: [1]

Backlogem byly plně nahrazeny zápisy ze schůzek. Na základě existujícího backlogu a po doplnění priorit k jednotlivým user story bylo možné začít plánovat sprinty. Sprint připravuje Product Owner ve spolupráci se supervisorem projektu. Nejprve supervisor definuje požadavky evidované v backlogu, které preferuje prioritně zpracovat. K vybraným požadavkům přiděluje Product Owner řešitele. Nezřídka se stává, že preferované požadavky mají společného řešitele, který pro plánovaný sprint nemá potřebnou volnou kapacitu. V takovém případě dochází k diskusi mezi Product Ownerem a supervisorem a na základě společné diskuze se optimalizují požadavky pro plánovaný sprint. Cílem přípravy sprintu je přidělení úkolů pro každého člena týmu. Rozsah úkolů pro každého člena by měl být splnitelný a v ideálním případě by měl pokrýt celou délku trvání sprintu. Odhad pracnosti a kapacit probíhá převážně empiricky na základě znalostí Product Ownera, který dle svého uvážení odhadne náročnost daného úkolu pro vybraného člena týmu. Bohužel tento přístup bývá v některých případech velmi nepřesný.

Sprint byl nastaven na délku jednoho týdne, tzn. práce se vždy plánovala na 1 pracovní týden. Na konci každého týdne předali vývojáři novou verzi testerům k otestování. Testeři začali testovat verzi až v rámci následujícího sprintu, tzn. s týdenním zpožděním. Aby nedocházelo k nepřehledné situaci, nalezené chyby a nedostatky byly evidovány ve sprintu aktuálním. Touto evidencí vznikaly situace,

kdy v rámci daného user story byly vývojové úkoly evidovány ve sprintu X a chyby nalezené testery pro dané user story byly evidovány ve sprintu Y.



**Obrázek 4: Týdenní plán v Azure DevOps**

**Zdroj: vlastní zpracování**

Pro tzv. sprintování byla použita šablona „Agile“, která je dostupná přímo v nástroji Azure DevOps. Tato šablona byla upravena a dále rozšířena o volně dostupné prvky. Jednou z úprav byly nově definované swimline (sloupce) sprintu a state (stavy) jednotlivých úkolů:

- Swimeline New, state New – výchozí stav každého úkolu. Po přiřazení úkolu do sprintu se úkol vždy nachází v prvním sloupci. V tomto stavu má již přiděleného řešitele.
- Swimeline Blocked, state Blocked – úkol, na kterém není možné pracovat, protože pro zpracování daného úkolu je zpravidla potřebné vyřešit jiný úkol, nebo čeká na doplňující informace, případně na akci třetí strany.
- Swimeline InProcess, state Active – jakmile řešitel rozpracuje úkol, přesouvá jej do stavu Active. Touto informací jsou ostatní členové týmu informováni, že se na úkolu již pracuje a je možné očekávat jeho brzké dokončení.
- Swimeline Dev, state Resolved – v okamžiku, kdy řešitel úkol zpracuje (zpravidla se jedná o vývojové úkoly), ve sprintu přesouvá úkol do stavu Resolved. Tento stav informuje, že řešení bylo nasazeno na Dev prostředí, kde je tato nová funkcionální dostupná pro potřeby ostatních řešitelů.

- Swimeline Test, state Testing – po dokončení všech úkolů k dané funkcionalitě se celá verze nasazuje na testovací prostředí, které je dostupné mj. testerům. Ve sprintu se změní řešitel na některého z testerů, aby testovací tým byl informován o nových funkcionalitách k otestování.
- Swimeline Done, state Closed – po úspěšném projití testovacího procesu tester vyjádří svůj souhlas se správným chováním funkcionality (úkolů) tak, že daný úkol převede do stavu Closed.
  - Během testování může dojít k alternativnímu scénáři – jestliže chování funkcionality (úkolů) není dle zadání, poté vrací tester úkol do stavu Active (sloupec InProcess), jako řešitel je nastavený původní řešitel úkolů. Po úspěšné opravě se pokračuje ve scénáři popsaném výše, tzn. Dev -> Test -> Done.

Pro každý sloupec (swimeline) byl definován právě jeden stav. Toto nastavení umožňuje pracovat s úkoly z různých obrazovek při zachování konzistence. Někteří členové preferují zobrazení úkolů v řádkovém zobrazení. Jestliže takový uživatel změní stav, členovi, který využívá zobrazení ve standardním „sprintovacím“ režimu, se úkol automaticky přesune do definovaného sloupce dle hodnoty stavu.

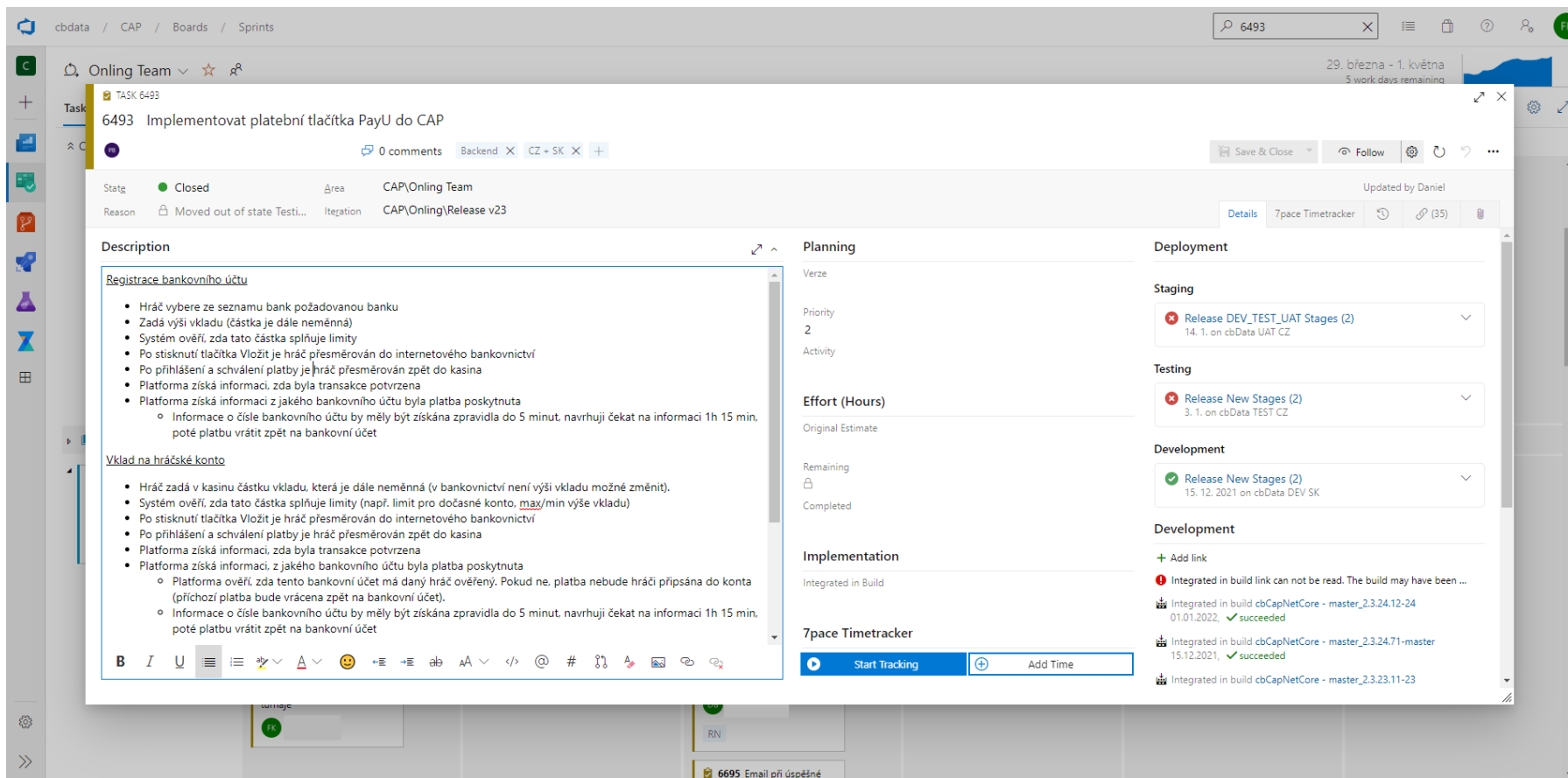
V této fázi projektu vývojářský tým intenzivně pracoval na vývoji, který měl za cíl pokrýt co nejvíce funkcionalit v krátkém čase. K tomuto cíli bylo nastavené verzování celého řešení – vytvářeli se verze s malým rozsahem funkcionality. Takto nastavený intenzivní vývoj byl vhodný do doby, kdy byla potřeba prezentovat tzv. release, neboli ucelenou verzi řešení určenou pro koncového uživatele. Jelikož vývoj se soustředil pouze na dílčí části projektu a testeři testovali pouze tyto nové změny, neexistovala 100 % otestovaná a funkční verze. Pro účely zákaznické certifikace (např. certifikace systému 3. stran, které jsou podmínkou pro provozování) bylo nutné zastavit standardní (intenzivní) vývoj a soustředit se pouze na dopracování stávající verze do plně funkčního stavu.

Využívaná šablona Agile byla rozšířena o další prvky, tentokrát konkretizující konkrétní úkoly:

- Zavedení tagů – z důvodu řešení pro různé trhy respektující různé legislativní požadavky byla vytvořena různá, na sobě nezávislá řešení. Každé řešení má své specifické požadavky, členěné nejenom legislativou, ale také požadavky zákazníka. Proto bylo nutné konkrétní úkoly v rámci sprintu více rozlišovat. Jako vhodné řešení byly zvoleny tagy, které umožňují použít několik tagů v rámci jednoho úkolu. Tímto řešením mohly vzniknout úkoly s tagy „Legislativa CZ“, „Zákazník Y“. Tagy se využívají pouze pro prostředí s odlišnou implementací, tzn. úkoly, které jsou dostupné pro všechny prostředí, nejsou odlišeny speciálním tagem.
- Rozšíření informačních polí o verzi – jak bylo popsáno již výše, intenzivní vývoj nerespektoval občasné potřeby pro zákaznický release. S blížící se nutností release verze byly úkoly, které byly součástí jednoho sprintu, označovány jako „Verze X“ a „Verze Y“. Tímto rozlišením se docílilo informace, ze které bylo zřejmé, který úkol je nutné zpracovat do stávající verze a které úkoly je možné, v rámci sprintu, odložit na později.
- Rozšíření informačních polí o prioritu – v rámci jednoho sprintu jsou i úkoly, které vyžadují nejdříve zpracování jiného úkolu (typicky se jedná o backendovou podporu pro frontendovou funkcionalitu). Proto má každý úkol přidělenou prioritu (nízká, střední, vysoká). Každý řešitel nejprve zpracuje úkoly s vysokou prioritou, aby případní navazující řešitelné mohli pokračovat v dopracování funkcionality.

Tato rozšíření byla nastavena tak, aby byla zobrazena uživateli i v režimu „sprintování“, tedy úvodní obrazovce v Azure DevOps – Sprint. Standardně jsou tyto informace zobrazeny v detailu každého úkolu.

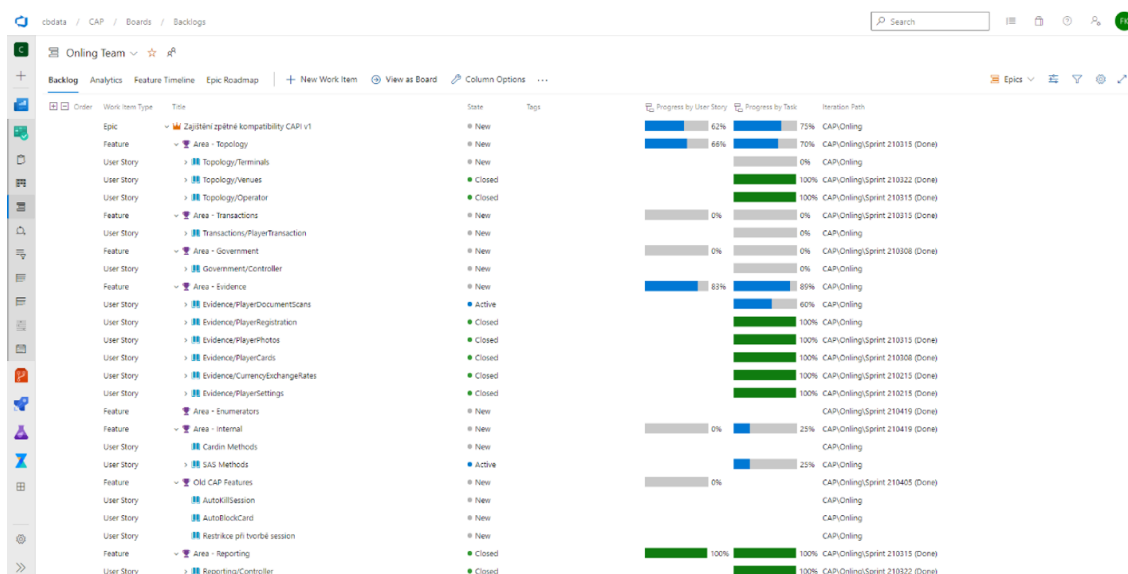




Obrázek 5: Detail úkolu v Azure DevOps

Zdroj: vlastní zpracování

Zavedení backlogu v rámci Azure DevOps mělo velký přínos také pro obecný přehled plnění úkolů v rámci projektu. Nástroj nabízí několik analytických modulů, díky kterým je možné sledovat vývoj zpracování jednotlivých úkolů. Jestliže se na začátku správně definoval rozsah celé Feature, je poté možné sledovat podíl zpracovaných a nezpracovaných User Story, případně Task. Tento modul velmi napomáhá zodpovědným osobám za projekt k rychlému a přesnému přehledu nad již dokončenými nebo nedokončenými úkoly.



**Obrázek 6: Přehled vyřešených úkolů v Azure DevOps**

**Zdroj: vlastní zpracování**

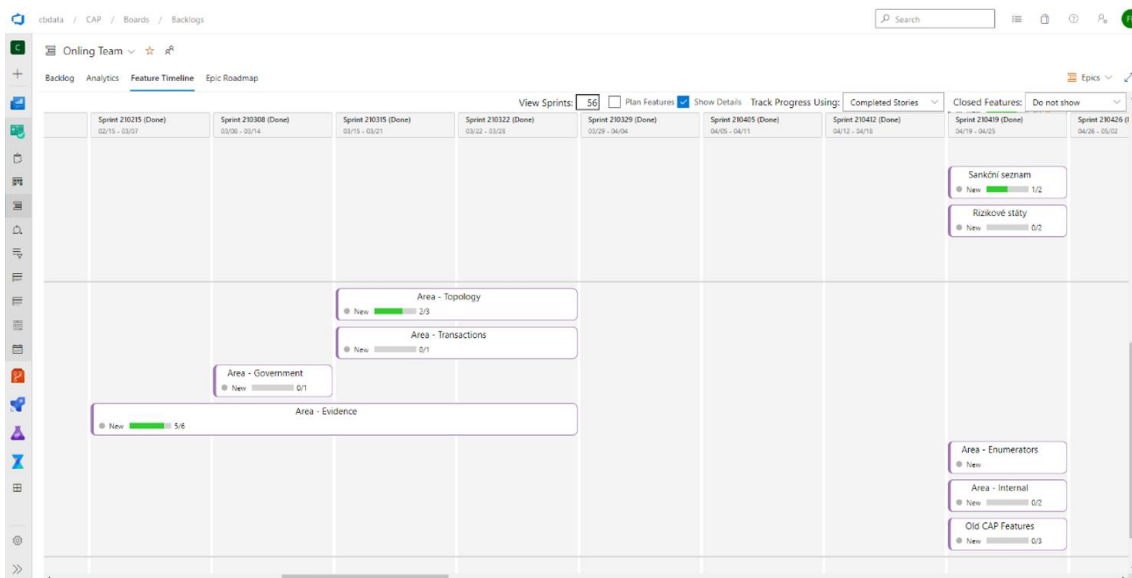
Díky zavedeným změnám a převedení správy úkolů do Azure DevOps bylo možné zavést pouze 2 schůzky týdně (oproti původním 3 schůzkám):

1. pondělí, 9:30 – Stand up – představení úkolů pro nadcházející sprint;
2. čtvrtek, 9:30 – Status – revize dokončených úkolů, případná dospecifikace zpracovaných úkolů.

Cílem je mít týmovou schůzku (této schůzky se účastní všichni členové týmu) v maximálním rozsahu 30 minut. Případné nejasnosti či komplikace jsou řešeny ihned kdykoliv během sprintu, tzn. během týdne probíhá několik „minischůzek“, kde jsou zúčastněni pouze dotčení členové, např. Product Owner a testeři nebo backend a frontend vývojáři.

Další z modulů, který byl aplikován do projektového řízení projektu, bylo zpracování roadmapy. V rámci backlogu umožňuje Azure DevOps vytvářet z Feature

dlouhodobou roadmapu. Tímto nástrojem je možné vytvořit přehledný obsah jednotlivých sprintů na několik týdnů až měsíců dopředu. Modul neumožňuje zohlednit události mimo sprint (např. milník – v tomto projektu certifikace), a proto byl využíván jen pro přibližné rozvržení kapacit členů v budoucích sprintech.



**Obrázek 7: Roadmapa v Azure DevOps**

**Zdroj: vlastní zpracování**

Zavedení Azure DevOps v rozsahu popsaném výše mělo nesporně zásadní dopad na řízení a organizaci projektu. Všem členům byl představen cíl, a tak nastavená pravidla byla brzy respektována. Oblastí, která byla především pro Product Ownera se supervizorem složitá a těžko udržitelná, bylo plánování rozsahu úkolů pro sprint. Vzhledem k nepřesně nastaveným sprintům a časté změně priorit (např. upřednostňování vývoje pro jedno konkrétní zákaznické prostředí z důvodu ah-hoc certifikace) nastávala situace, kdy část sprintu bylo nutné, z důvodu nedokončení práce v rámci plánovaného sprintu, přesunout do sprintu následujícího. Tímto přístupem začal být průběh sprintu zkreslený, jelikož úspěšnost dokončení všech zadaných úkolů se v dlouhodobém horizontu pohybovala mezi 50 – 70 %. Tento fakt měl dopad mj. do roadmapy, která předpokládala splnění naplánovaného sprintu. Díky pravidelnému přesouvání úkolů do následujícího sprintu se roadmapa stala po pár měsících neudržitelnou a nyní se již roadmapa v původně zamýšlené formě nevyužívá.

Přístup intenzivního vývoje bez pravidelných release byl vhodný do okamžiku produkčního spuštění projektu Online kasino u prvního zákazníka. Od tohoto

okamžiku bylo důležité mít přehled o obsahu verze, což plánování týdenních sprintů neumožňovalo. První vážnou komplikací bylo vytvoření release notes (popis funkčního obsahu verze). Jelikož verze nebyly vydávány pravidelně a nebyly vytvářené žádné release notes pro tento nový projekt, nebylo možné jednoduše vytvořit release notes. Sprint se často skládal z několika verzí a poté bylo obtížné dohledat, co která verze obsahuje za funkcionality.

Dlouhodobě evidovaným nedostatkem organizace projektu byla téměř žádná evidence změn oproti zadávací dokumentaci. V praxi docházelo příležitostně k situacím, kdy vznikla zadávací dokumentace k funkcionalitě, ale během vývoje došlo k odchylce a v některých případech i k zásadním změnám oproti zadávací dokumentaci. Tyto změny však nebyly dodatečně zapracovány do zadávací dokumentace. V okamžiku, kdy se dostala funkcionalita do fáze testování, přiložená dokumentace nebyla již aktuální a ze strany testerů vznikalo několik otázek a chybně zadaných „bugs“ na základě neaktuální dokumentace. V této problematice přetrvávala neshoda ohledně zodpovědné osoby, která by měla tyto změny zaznamenávat.

Shrnutí nedostatků tohoto přístupu:

- neudržitelnost dlouhodobé roadmapy;
- neexistence přehledu změn oproti předešlé verzi;
- neexistence uživatelské dokumentace.

#### **4.6 Změna komunikačních nástrojů**

Vedení firmy se rozhodlo dále nevyužívat k hlavní interní komunikaci aplikace Slack v bezplatné verzi a Google Suite primárně využívaný pouze k online schůzkám. Veškerá komunikace a cloudové řešení pro ukládání interních dokumentů se převedlo pod služby nabízené společností Microsoft, konkrétně se pro komunikaci používá Microsoft Teams, jako úložiště nejen projektové/produktové dokumentace je využíván Microsoft Sharepoint. Pro osobní dokumenty je možné využívat cloudové úložiště Microsoft One Drive. Tímto krokem se velmi zjednodušil přehled v potřebných dokumentech, jelikož každý projekt má v MS Sharepoint své

dedikované úložiště. Veškeré dokumenty související s daným projektem by měly být uloženy na tomto místě.

Zaměstnanci ocenili také plnohodnotný komunikační kanál, který nabízí plnohodnotnou historii. MS Teams se dále rozšířily o externí moduly typu Polly (ankety využívané projektovými týmy) a přímými přístupy do projektového úložiště.

## 5 Dopad projektu Online kasino na firmu Czech Brain Data

Po přibližně dvou letech od zahájení příprav a v době, kdy firma spravovala několik vlastních produkčních platforem online kasina, vyhodnotilo vedení firmy projekt Online kasino jako úspěšný a jednou ze silných stránek projektu bylo označeno řízení a organizace týmu. Pozitivně bylo dále vyhodnoceno:

- využití dostupných možností Azure DevOps;
- efektivita vývoje vycházející z bezproblémové reakce na změny v zadání;
- transparentnost zákaznických požadavků;
- přehlednost stavu vývoje v kterékoliv fázi;
- dlouhodobá evidence nedořešených úkolů;
- spolupráce celého vývojového týmu.

Pozitivní zpětná vazba přesvědčila vedení firmy rozšířit využívané metodiky a praktiky aplikované v projektu Online kasino i na další projekty v rámci firmy Czech Brain Data. Vedení firmy identifikovalo firemní nedostatky:

1. dělení zaměstnanců na landbase a online;
2. špatná čitelnost struktury firmy pro obchodní zástupce;
3. nejasné kompetence a nepokryté projekty;
4. interní boj o priority a kapacity;
5. nejednotné vedení týmu komplikující sdílení vývojářů mezi týmy.

Na základě identifikovaných nedostatků a díky získaným zkušenostem s řízením projektu Online kasino byl přizván autor této práce, který je zároveň analytikem / Product Ownerem v projektu Online kasino, k poskytnutí metodických konzultací. Cílem bylo provést reorganizaci firmy a zavedení podpůrných praktik a metod v řízení projektů. Byly identifikovány 3 oblasti určené k revizi:

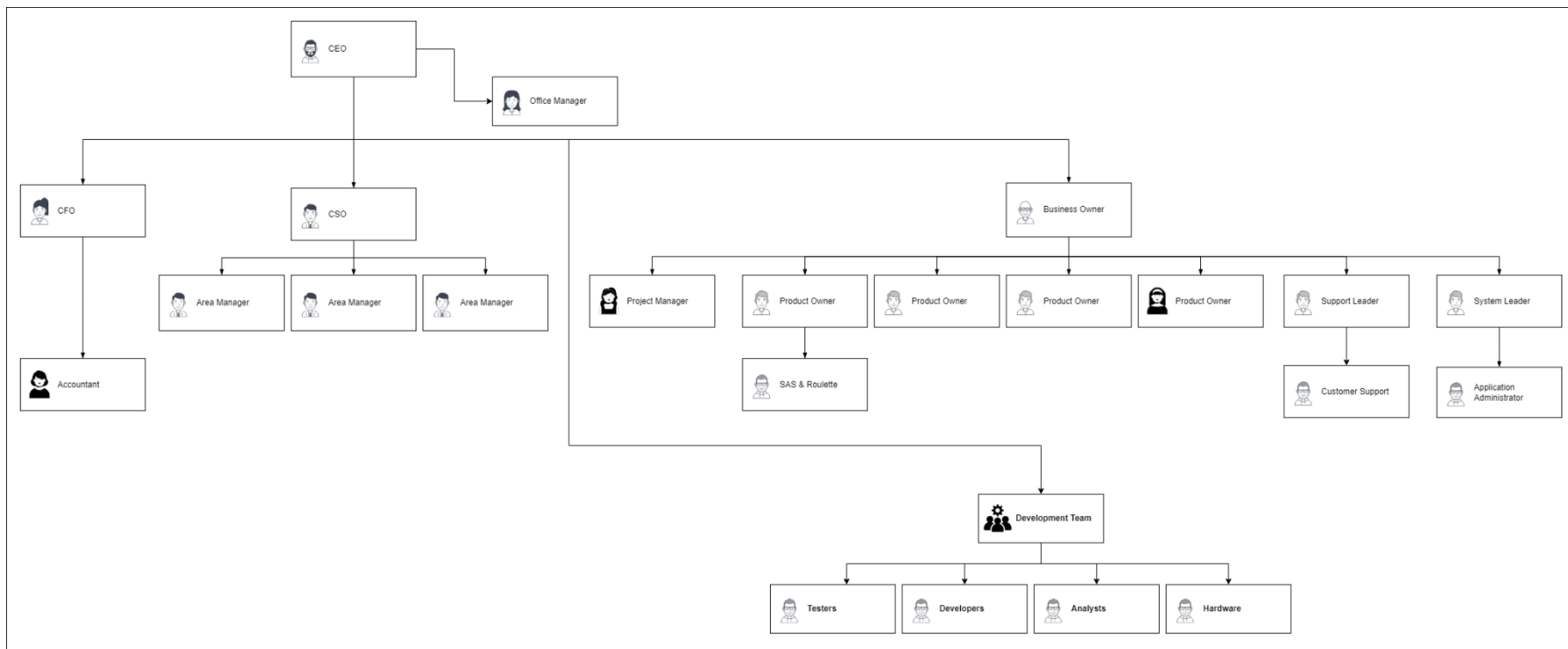
1. organizační struktura;
2. sjednocení společných schůzek;
3. vedení a správa vývojových týmů včetně aplikování Azure DevOps.

## **5.1 Organizační struktura**

V první fázi reorganizace byla definována nová organizační struktura firmy, která odstraní identifikované nedostatky.

Hlavní změnou je definování produktového managementu firmy. Jelikož firma pracuje na několika projektech tvořících jeden produkt (interní název produktu je iCore), byla vytvořena role Business Owner – osoba zodpovědná za úspěch celého produktu. Za každý dílčí projekt je zodpovědný Product Owner. Dílčí projekt nemá přímo dedikovaný vývojářský tým, vývojářským týmem je jeden a je dostupný pro každého Product Ownera. Každý z Product Ownerů může využít kohokoliv z vývojářského týmu po předchozí domluvě s ostatními Product Ownery.

Projektová manažerka je k dispozici Business Ownerovi a Product Ownerům a zajišťuje denní operativu napříč všemi projekty.



**Obrázek 8: Nová organizační struktura platná od 1.4.2022**

**Zdroj: vlastní zpracování**



### 5.1.1 Business Owner

- Zodpovědný za úspěch celého iCore.
- Definiuje vizi a směr působení iCore.
- Sestavuje roadmapu produktu.
- Má rozhodující slovo při případném sporu.
- Má nadhled nad všemi projekty a vidí produkt jako celek.

### 5.1.2 Product Owner

- Zodpovědný za dílčí projekt iCore.
- Jeho nápady/požadavky vytváří business value projektu, která přirozeně vytváří business value iCore.
- Společně s ostatními Product Ownery tvoří tým, který má 1 cíl – maximalizovat business value iCore.
  - Jiný Product Owner není konkurent, ale partner pro dosažení vyšší business value (je efektivnější realizovat jedno řešení napříč projekty, než vytvářet vlastní řešení pro každý projekt).
- Cílem společné diskuze Product Ownerů je mj. definice rozsahu společné funkcionality, identifikování dopadů do dílčích projektů a koordinace kapacit pro společnou realizaci.
- Je v úzké komunikaci/kooperaci s obchodními zástupci a byznysem/zákazníkem.
  - Pro vývojářský tým je zástupce byznysu – předává business know-how.
  - Pro byznys/zákazníka (a případně manažery) je kontaktní osobou pro vybranou oblast – z pohledu nových požadavků/nápadu, ale i denní operativy (mimo kompetence zákaznické podpory).
- Sbírá zákaznické požadavky na rozvoj. Reprezentuje všechny zákazníky.
- Přináší vlastní inovace na rozvoj.
- Nové požadavky a inovace spravuje v product backlogu v podobě Feature a User Story. Zajišťuje, aby v backlogu bylo vše, co zákazník chce a potřebuje.

- Stará se o to, aby v backlogu bylo vše stěžejní, aby se méně důležité (nepodstatné) požadavky z backlogu odebraly.
- Na základě své úvahy a požadavků zákazníka prioritizuje backlog.
- S ostatními Product Ownery diskutuje obsah backlogu (důležité feature pro případný rozvoj celého iCore, alternativně aktuální feature pro následující iteraci).
- S týmem diskutuje obsah iterace (samotný obsah následující iterace definuje tým).
- V průběhu iterace je k dispozici týmu – dospecifikuje zadání, je-li potřeba.
- Je-li je od týmu k dispozici „hmatatelný“ výstup, provede revizi a dá týmu zpětnou vazbu (vhodný stav – Testing).
- Je-li zadaná feature komplexní a user story jsou zařazeny do více iterací, po ukončení dílčí iterace prezentuje výstup zákazníkovi.
- Feedback od zákazníka prezentuje týmu, který zohlední připomínky v dalším vývoji.

### 5.1.3 Tým

- Zodpovědný za dodání celého obsahu iterace v definovaném čase.
- Jedná se o tým, nikoliv o jednotlivce (programátoři, testeři, analytici jsou nahrazeny pojmem tým). Úspěch i neúspěch je kolektivní. Pokud se např. nepodaří otestovat feature během iterace, je to chyba týmu, nikoliv testerů. Pokud se podaří implementovat novou výkonnou komponentu do produktu, je to úspěch týmu, nikoliv vývojářů.
- Členové mezi sebou pravidelně komunikují a předchází tak tím nedorozumění a „hluchým místům“ (např. nepředání informace o připraveném řešení pro realizaci dalších kroků).
- Pravidelně se schází a diskutují o iteraci jako celku – mj. informují Product Ownera o průběhu iterace, snaží se odstranit překážky zjištěné při vývoji.
- Provádí pravidelné vyhodnocení průběhu a realizace iterace – představí se klady a zápory, přijímají se opatření pro další iterace.

- Cílem je předat zákazníkovi feature přinášející požadovanou business value, nikoliv 100% splněné prvotní zadání.

## **5.2 Pravidelné schůzky zaměstnanců**

Vedení firmy jako jednu z kritických oblastí označilo komunikaci mezi zodpovědnými pracovníky za dílčí projekty, obecně byla informovanost mezi zaměstnanci vyhodnocena jako nedostatečná. Na základě těchto informací byl definován pravidelně se opakující cyklus schůzek na odlišných úrovních komunikace.

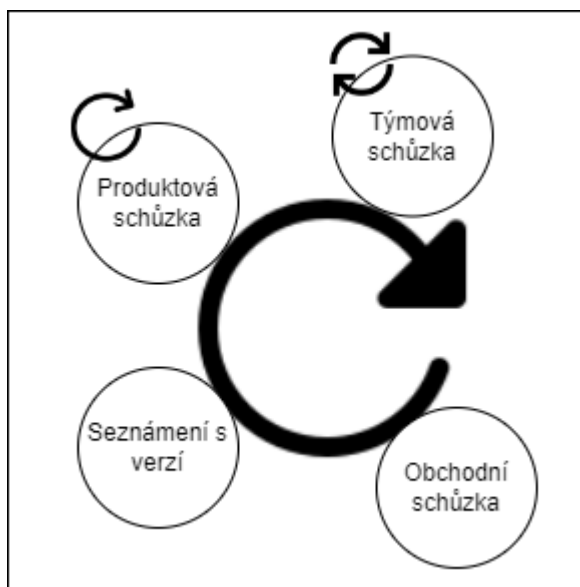
Cyklus zahajuje obchodní schůzka, kde probíhá především diskuze mezi obchodními zástupci a business ownerem, případně Product Ownery. Obchodní zástupci reportují aktuální stav u zákazníků a informují ostatní o možných příležitostech na trhu. Naopak business owner informuje obchodní zástupce o stavu vývoje a případných důležitých eventech (např. certifikace herních zařízení).

Na obchodní schůzku navazuje produktová schůzka, kde diskutují product ownery např. o plánu vývoje a potřebných kapacitách. Do diskuze se zapojuje support leader se zpětnou vazbou o řešených situacích zákaznické podpory, administrator leader informuje o stavu zákaznických prostředí a případných plánovaných odstávkách. Business owner je pouze příjemcem informací, případně moderátor schůzky. Hlavním cílem je definice priorit a kapacit na nadcházející týden a informování vedoucích pracovníků napříč projekty. Produktová schůzka probíhá 1x týdně.

Jsou-li známé priority a kapacity, je možné uskutečnit schůzku, která má za cíl seznámit vývojový tým s obsahem verze. V roli moderujícího a prezentujícího je Product Owner, který představuje týmu rozsah verze. Tým diskutuje s Product Ownerem o nejasnostech, výsledkem jsou požadavky k řešení, případně jsou identifikovány úkoly k dopracování a je naplánován harmonogram návazných úkolů.

Tým se pravidelně schází alespoň 2x týdně a vzájemně se členové informují o rozpracovaných a dokončených úkolech, čímž se snaží udržovat tempo vývoje. Jsou zde zmíněny případné překážky ve vývoji. Product Owner je informován o stavu vývoje a je k dispozici týmu v případném řešení komplikací.

Obrázek č. 9 znázorňuje opakující se cyklus pravidelných schůzek, kdy se některé schůzky mohou opakovat vícekrát v rámci jednoho hlavního cyklu.



Obrázek 9: Cyklus pravidelných schůzek

Zdroj: vlastní zpracování

#### Obchodní schůzka:

- účastníci: Business Owner, Product Owneři, Obchodní zástupci;
- četnost: 1x měsíčně;
- délka: 45 – 60 minut;
- zpětná vazba od obchodu o možných příležitostech;
- informování o zpětné vazbě z provozu / od zákazníků;
- informování o stavu vývoje, chystaných feature, plánovaných certifikacích, blížících se verzích.

#### Produktová schůzka:

- účastníci: Business Owner, Product Owneři, Support Leader, Administrator Leader;
- četnost: 1x týdně;
- délka: 30 minut;
- zhodnocení odvedené práce za projekt / tým;
- nové náměty a nápady na vývoj od Product Ownera;

- získání zpětné vazby od ostatních Product Ownerů, zda je nápad využitelný i pro ostatní;
- pokud je potřeba součinnost dalších týmů, domluvit přibližný harmonogram;
  - není cílem diskutovat detailně se všemi členy, cílem je zajistit dotčené osoby a s nimi následně diskutovat odděleně;
- plán práce na další týden za tým;
- blížící se eventy;
  - dokončení verze;
  - nasazení verze na UAT/PROD;
- plánované certifikace - např. nové online hry;
- informování o plánovaných dovolených napříč firmou Hazard;
- pro vizualizaci úkolů je možné využít DevOps Kanban, kde bude zobrazen cyklus dotčené feature;
- závěry ze schůzky zaznamenat do meeting minutes - slouží jako nástroj pro rychlou orientaci napříč projekty a případnou prezentaci vedení / obchodu.

#### Týmová schůzka:

- účastníci: Product Owner, vývojářský tým;
- četnost: alespoň 2x týdně;
- délka: 60 minut / týden;
- především týmová schůzka o stavu vývoje;
  - k diskuzi využít Azure DevOps - Sprint -> úkoly jsou zařazeny pod stavy, ve kterých se úkol nachází;
  - zhodnotit, co se podařilo zpracovat od poslední schůzky a na čem se bude pracovat nyní;
  - ověřit, zda neexistují žádné blokace ve vývoji;
    - pokud ano, zajistit pomoc - samotnou pomoc je možné řešit už mimo schůzku, není nutné řešit detaily (dle preferencí);
- zpětná vazba pro Product Ownera o stavu vývoje;
- Product Owner naslouchá týmu a nabízí pomoc v případných nejasnostech - např. dospecifikování zadání.

### Seznámení s verzí:

- účastníci: Product Owner, vývojářský tým;
- četnost: před zahájením nové iterace;
- délka: dle potřeby. Cílem je dokonalá informovanost celého týmu;
- Product Owner má připravený základní popis User Story, čím více informací je k dispozici, tím efektivnější je schůzka;
  - na přípravu je možné využít další členy týmu, např. analytika, testera,..;
- Product Owner prezentuje rozsah verze;
- rozsah je prezentovaný ve formě User story, tým vytváří úkoly, které vznikají na základě požadavků v User Story;
- je vhodné definovat priority, které User Story realizovat dříve (po diskuzi s týmem). Doporučení na priority:
  - 1 – nízké náklady na vývoj, vysoká business value pro zákazníka;
  - 2 – vysoké náklady na vývoj, vysoká business value pro zákazníka;
  - 3 – nízké náklady na vývoj, nízká business value pro zákazníka;
  - 4 – vysoké náklady na vývoj, nízká business value pro zákazníka;
- délka schůzky je dána rozsahem verze, čas není prioritní, prioritní je porozumění a informovanost všech členů týmu;
- každý člen by měl mít představu, co má dělat;
- vhodný prostor na vznešení otázek na Product Ownera, pokud není jasné zadání - dotazovat se může kdokoliv, otázky a odpovědi je nutné doplnit do zadání.

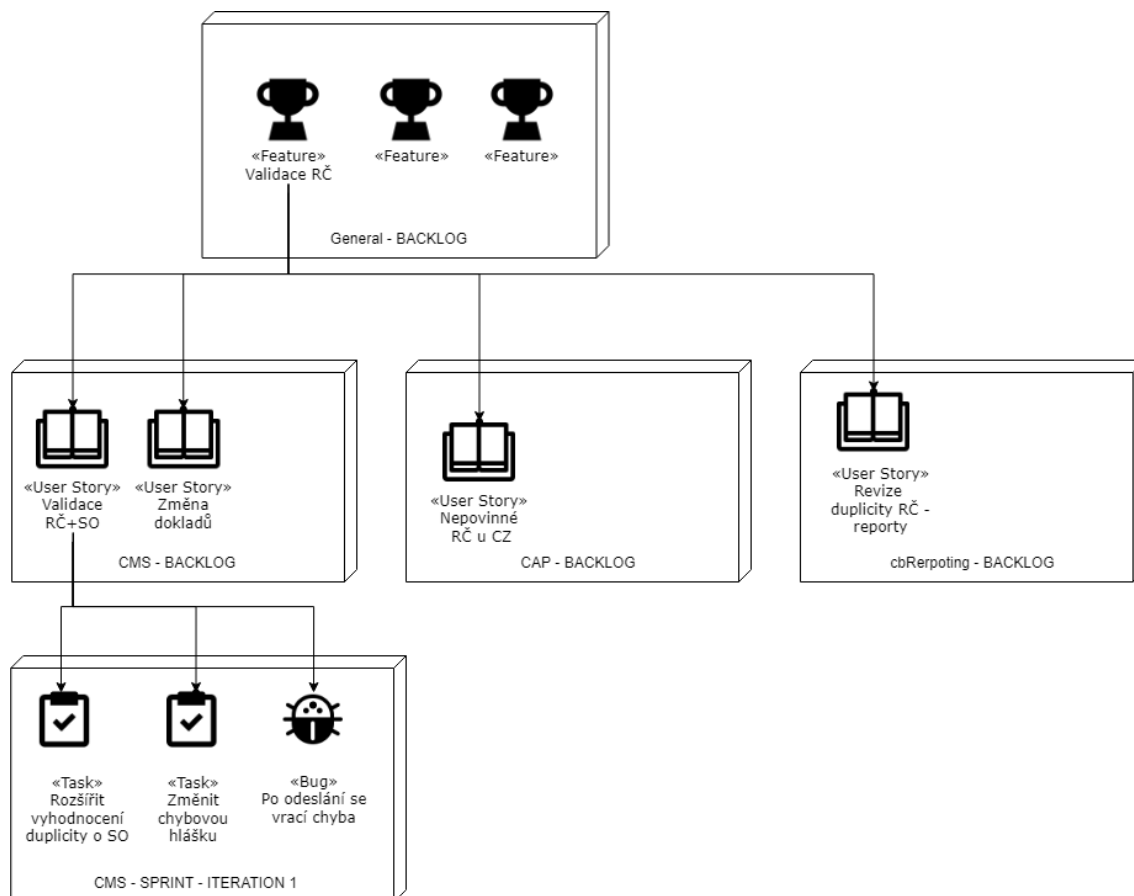
### **5.3 Využití Azure DevOps**

Nástroj Azure DevOps je využíván ve firmě především pro organizaci projektů, avšak většina projektů využívá jen minimum dostupných možností. Zároveň každý vedoucí projektu praktikuje odlišný přístup, a proto migrace členů vývojářského týmu bývá často komplikovaná z důvodu nepochopení. Cílem je jednotné řízení a maximální možné využití dostupných nástrojů v rámci Azure DevOps.

Aby byla maximalizace využití efektivnější, zavádí se jednotná struktura zadávání funkcionalit. Kořenem každé funkcionality je Feature – abstraktní popis požadavku zákazníka, který je vhodný pro komunikaci mezi Product Ownery, případně business ownerem a obchodními zástupci. Diskuze v rámci produktové schůzky probíhá nad požadavky typu Feature, tyto požadavky jsou zařazeny a evidovány v rámci společné „tabule“ (Board) produktového managementu. Každá funkcionalita se nově člení na Feature – User Story – Task/Bug. Díky evidování nové funkcionality v této stromové struktuře je možné sledovat snadno např. náklady na vývoj. Vývojový tým vykazuje strávený čas na vývoj u každého dílčí User Story / Task, avšak výsledný čas je možné zobrazit v kořenu – Feature (zde je zobrazena celková suma i dílčí hodnoty dle jednotlivých elementů). Další výhodou je jasné a srozumitelné zadání. V rámci jedné Feature je možné vytvořit několik User Story a každé story může být platné pro jiný projekt. Část nové funkcionality může zpracovat tým Onling, další část může zpracovat tým Reporting. Stav je možné sledovat opět z kořene Feature a není tak nutné hledat informace v dílčích projektech. Provázanost mezi jednotlivými elementy zaručuje transparentnost nad novou funkcionalitou a předchází tak vzniku možných duplicit ve vývoji.

K přehlednosti stavu vývoje je důležité definovat každou iteraci, je možné využít 1 iterace = 1 verze. Rozsah iterace by měl být neměnný, změna v průběhu iterace narušuje tempo vývoje a snižuje morální pohodu týmu.

Vizualizačním nástrojem pro organizaci vývoje je tabule – Board. Tabule může mít pro každý projekt odlišené stavy či sloupce – Swimeline. Stav, sloupce a další prvky je vhodné definovat dle povahy a potřeb každého týmu.



Obrázek 10: Struktura evidence požadavků na vývoj

Zdroj: vlastní zpracování

### 5.3.1 Předpoklady k úspěšné aplikaci

Aplikování výše popsaných změn není možné okamžitě v celém rozsahu. Týmy, případně celé firemní organizace musí mít k takové změně předpoklady, aby změna byla přirozená a přechod byl také přirozený. Nejsou-li splněné některé předpoklady, aplikace změn bude komplikovaná, ne-li nemožná. Hlavními předpoklady jsou:

1. Dostatečná znalost nové funkcionality – k identifikaci vztahu Feature – User Story.
2. Předem definovaný rozsah verze.
3. Evidence úkolů pro tým pouze v Azure DevOps (nevidovat úkoly v dalších nástrojích, aby pro tým byla organizace přehledná).



4. Nebát se hovořit o potřebách projektu a snažit se přizpůsobit Azure DevOps potřebám týmu. Není žádoucí, aby se tým přizpůsoboval možnostem Azure DevOps.

### 5.3.2 Využívané elementy

Byla definována nová vhodná struktura pro zadávání nových požadavků a dílčích úkolů do plánu vývoje. Azure DevOps nabízí předpřipravené struktury dle dostupných šablon, na základě specifických potřeb firmy Czech Brain Data byla definována individuální struktura.

Tato struktura se skládá z Feature, což je kořen nové funkcionality a je vhodná pro komunikaci mezi Product Ownery (je dostatečně obecná pro komunikaci mezi odborníky a businesssem, a zároveň vystihuje všechny potřeby na vývoj).

Každý Feature se skládá z několika User Story, které definují konkrétní požadavek na vývoj z pohledu uživatele. User Story se může dále skládat z úkolů (Task) a chyb (Bug), které jsou evidovány týmem dle jeho potřeb.

#### Feature:

- Kořen nové funkcionality.
- Př. Datové balíčky.
- Dělí se na User Story:
  - Přehled balíčků "Jako uživatel chci vidět všechny vytvořené balíčky, abych věděl, co se odeslalo na Ministerstvo Financí (MF)".
  - Stav balíčků "Jako uživatel chci mít stavy zpracování balíčků, abych ihned viděl zamítnuté balíčky z MF".
  - Historie balíčků "Jako uživatel chci možnost zobrazit starší balíčky, abych mohl provést kontrolu úspěšného doručení".

#### User Story:

- Jedna User Story by neměla tvořit více jak 40 % kapacit týmu.
- Je testovatelná.
- Obsahuje akceptační kritéria:

- Definuje je Product Owner a může to být:
  - Set pravidel, sdělení vývojářům, co je cílem a jiné varianty nejsou přípustné.
  - Např. „User Story musí obsahovat tyto 3 věci“.
- Každé User Story by mělo přinášet business value. User Story by mělo být definováno tak, aby bylo jasné, jakou business value přinese.
- Je psaná z pohledu koncového uživatele, neměla by obsahovat technické informace (to je už v kompetenci vývojáře).

### 5.3.3 Doporučené nástroje pro řízení týmu

Aby změna v evidenci elementů přinesla maximální efektivitu, je vhodné tuto evidenci podpořit dalšími podpůrnými nástroji, které jsou dostupné v Azure DevOps. Pokud by se nová struktura zavedla, avšak evidence zůstala i nadále v podobě MS Excel, či PDF souborech, přínos ze změny elementů by byl minimální a hrozilo by riziko v nepochopení, které by pokračovalo v odmítnutí změn u řady zaměstnanců.

Hlavními doporučenými prvky je využívání sprintů (iterací) a board (tabule). Sprint udává rozsah realizovaných funkcionalit za časové období, kdy rozsah by měl být neměnný. Každá User Story, která je zařazena do vývoje, by měla být zařazena do sprintu. Naopak, Feature může a nemusí být zařazena do sprintu, jelikož se může skládat z několika User Story napříč projekty a každý projekt má vlastní sprinty.

Pro zobrazení stavu Feature je vhodné použít board, což je zobrazovací nástroj stavu vývoje. Uživatel má při využívání board okamžitý přehled o stavu Feature a jeho dílčích User Story (např. informace, že se Feature skládá z 5 User Story, kdy do dokončení celé Feature zbývá dokončit poslední User Story). Board je možné využívat i na nižší úrovni, tzn. stav User Story. Jak již bylo zmíněno, každá User Story se může skládat z několika úkolů a případných chyb a pro Product Ownera i celý jeho tým je důležité mít přehled o tom, co je potřeba vyřešit nebo dořešit, aby byla User Story vyřešena.

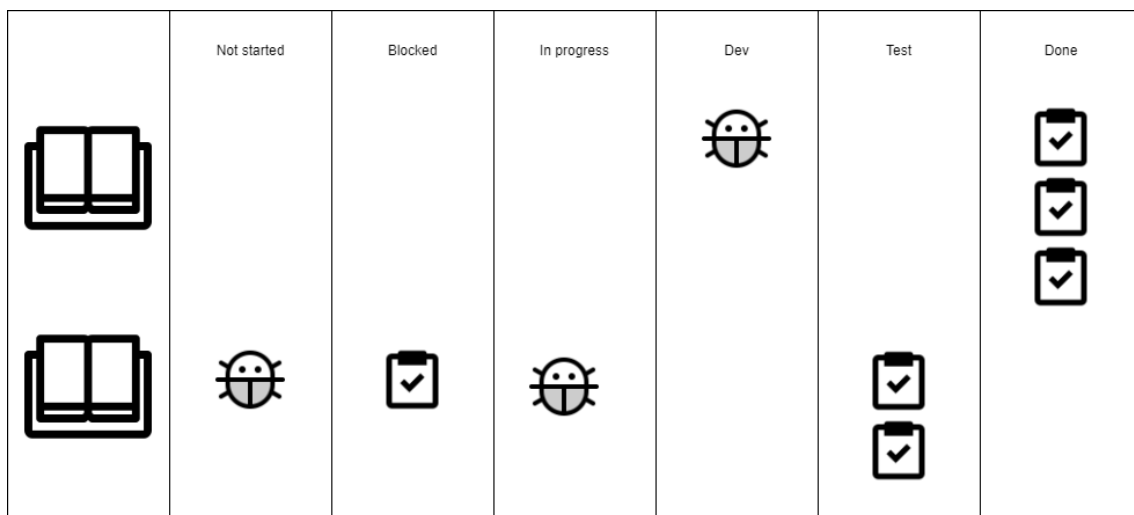
#### Sprint:

- Předem definovaná doba, která má pevně stanovaný konec.

- Doporučená délka je 2 nebo 4 týdny.
- Obsah sprintu je pevně daný.
- Nemělo by docházet ke změně obsahu v průběhu sprintu.
- Pokud se nestihne User Story dokončit do stavu Done, celá User Story se vrací do backlogu a při dalším plánování dojde k novému odhadu a novému zařazení do sprintu.
- Pokud se naruší tempo sprintu (některá User Story se nestihne), ohroží se tím release date. Je možné reagovat tím, že se ze sprintu odeberou některé plánované User Story, nebo se přidají kapacity, aby se dodrželo datum release.

#### Board:

- Podpůrný prvek pro vizualizaci stavu vývoje.
- Musí být zřejmé, kdo na čem pracuje.
- Musí být jasné priority.
- Musí být zřejmé, co je hotové.
- Swimline je možné konfigurovat dle potřeb týmu.



Obrázek 11: Vizualizace požadavků na vývoj

Zdroj: vlastní zpracování

### 5.3.4 Zodpovědnosti a povinnosti členů týmu

Pro složení projektového týmu ve firmě Czech Brain Data byly definovány 2 role – Product Owner a Development tým. Role Scrum Mastera prozatím nebyla zavedena, předávání zkušeností, rad a mentoring bude realizován formou společných prezentací.

Každý projekt má vlastního Product Ownera, který zodpovídá za úspěch projektu, spravuje a definuje požadavky na vývoj, kdykoliv je k dispozici týmu. Product Owner je stejným členem projektového týmu jako Development a není týmu nadřízený.

Development tým se skládá z několika vývojářů, testerů, dle potřeb projektu může být zapojen také analytik nebo grafik. Development tým zpracovává zadané User Story, je v úzké komunikaci s Product Ownerem a především vytváří hodnotu svěřeného projektu.

#### Product Owner:

- Spravuje backlog v podobě Feature a User Story (přidat nový nápad může kdokoli z týmu).
- User story definuje včetně funkčního scénáře (1 US = 1 hlavní scénář + alternativní scénáře).
- K definování je možné využít celý tým (např. analytik, tester, developer) – Product Owner prezentuje představu o Feature, tým pokládá otázky k objasnění. Cílem je definovat chování v rozpadu na dílčí User Story.
- Je vhodné týmu prezentovat cíle:
  - Cíl sprintu/iterace -> čeho chceme dosáhnout a co získáme dokončením tohoto sprintu.
  - Cíl releasu -> co nám přinese nasazení verze do produkce. Pomáhá při rozhodování, jakou User Story zařadit do sprintu, aby byl naplněn cíl.
  - Cíl/vize produktu – „chceme dělat precizní (bez chyb) online kasina na kvalitních technologiích a dle platné legislativy,...“.

### Tým:

- Poskytuje odhad Product Ownerovi na připravené User Story.
- Po konzultaci s Product Ownerem a zohlednění priorit definuje obsah iterace.
- User Story zařazené do vybrané iterace doplní o dílčí tasky.
- Odbavuje tasky v procesu: New – Blocked/InProgress – Resolved – Testing – Done.
- Větší/složitější User Story odbavuje prioritně na začátku iterace, méně komplexní požadavky řeší později – cílem je netestovat komplexní změny v poslední den iterace.
- Poskytuje zpětnou vazbu PO o stavu řešení – pozitivní/negativní výstupy, změny v chování, omezení řešení apod.

## 6 Shrnutí výsledků

Diplomová práce byla zaměřená na vývoj projektového řízení a aplikaci agilních metod na vybraném projektu IT společnosti. Projekt byl zahájen v dubnu 2020 a diplomová práce tak zachycuje přibližně 2 roky vývoje projektového řízení tohoto projektu. Zahájení projektu korespondovalo se zahájením studia agilních metod v řízení IT projektů, a proto byla úvodní fáze vývoje projektu Online kasino řízena tradičním vodopádovým přístupem. V úvodní fázi bylo zadání projektu všem zřejmé a připravené v dostatečném rozsahu, i z tohoto důvodu byl tradiční přístup projektového řízení v úvodní fázi dostatečný.

Zlom nastal v okamžiku, kdy platforma online kasina získala certifikaci. Zadání na vývoj se velmi rychle a často měnilo, mnohokrát nastala situace, kdy vývojáři „předběhli“ zadavatele. V této fázi začalo být zřejmé, že aplikovaný přístup dosahuje svých limitů a je nutné transformovat projektové řízení do podoby vyhovující aktuálnímu stavu. Do řízení projektu byly zapracovány přístupy a metody z metodik Scrum a Kanban, rozšířilo se využití nástroje Azure DevOps a to především o backlog a board. Vývoj se začal plánovat dle předem připravených iterací, kdy jedna iterace představuje jednu verzi. Suma těchto změn tvořila zásadní reorganizaci projektového řízení vybraného projektu, jelikož změny představovaly:

- Agilně řízený vývoj oproti tradičnímu vývoji – důsledkem byla rychlejší reakce na změnu.
- Veškeré požadavky včetně plánu vývoje evidované na jednom místě Azure DevOps oproti decentralizované evidenci (projektový plán v souboru PDF, zadání úkolu v Azure DevOps/Word/atd).

Prvotní změna projektového řízení se ukázala jako zásadní, avšak ne jako dlouhodobě vyhovující. Z tohoto důvodu byly metodiky a přístupy diskutovány s celým týmem, byly vyhodnoceny výsledky a vznášeny návrhy na další změny. Přibližně po roce od aplikování prvních změn se projektové řízení ustálilo a za další půlrok bylo vedením firmy vyhodnoceno jako úspěšné a efektivní.

Toto hodnocení vedení firmy mělo za následek vytvoření souhrnných změn a doporučení pro ostatní projekty v rámci firmy Czech Brain Data. Navrhované

změny pro ostatní projekty jsou platné od dubna 2022 a proto v době dokončení této diplomové práce nejsou známy výsledky ostatních týmů.

V úvodu práce byly stanoveny tři výzkumné otázky, které byly během zpracování zodpovězeny. Na základě aplikování teoretických a metodických postupů do praxe byly získány tyto výsledky:

### **Otázka 1: Je vhodné kombinovat různé agilní metodiky v řízení jednoho projektu?**

Na projektu Online kasino bylo potvrzeno, že je možné aplikovat více agilních technik v rámci jednoho projektu. Na tomto projektu byla využita kombinace technik Scrum a Kanban, tuto kombinaci označuje literatura jako Scrumban [22]. Metodika je snadnější na zavádění agility u týmů, které nemají s agilním vývojem zkušenosti. Díky Kanbanu je možné rychle aplikovat agilní přístup vývoje software do řízení projektu. Scrum obohacuje tým o pravidelné schůzky, jako jsou například daily scrum (týmová schůzka), sprint planning (seznámení s verzí), nebo sprint review (tato schůzka doposud nebyla zavedena v rámci firmy Czech Brain Data).

V rámci projektu Online kasino byl využit z Kanban technik například předem definovaný rozsah verze (iterace), tzn. délka iterace není definována časem (Scrum), avšak svým rozsahem prací. Scrum rozšířil tento projekt mj. o pravidelné týmové schůzky a častou komunikaci mezi členy týmu.

### **Otázka 2: Existuje doporučená struktura pro evidenci úkolů?**

Každá z metodik doporučuje konkrétní názvy, např. User Story vs. Task, avšak každý z podpůrných nástrojů nabízí vlastní šablony s vlastní strukturou. V rámci doporučených změn ve firmě Czech Brain Data byla vybrána již existující struktura dostupná v Azure DevOps skládající se z Epic - Feature - User Story - Task/Bug. V praxi se element Epic nevyužívá a element Feature slouží především pro komunikaci mezi Product Ownery, případně pro komunikaci se zákazníkem. Elementy, které zpravidla zpracovává Development tým, jsou User Story a případně Task nebo Bug.

### **Otázka 3: Má zavádění agilních metod kladný vliv na produktivitu týmu?**

Literatura v tomto ohledu zcela jasně popisuje možná úskalí a možné negativní dopady na produktivitu týmu [11]. Transformace z tradičního na agilní řízení ve firmě Czech Brain Data nebyla výjimkou. Úvodní prezentace, na které bylo představeno zavedení prvních prvků agility, vyvolala mezi členy plno otázek a nejasností, v některých případech byla metodika špatně pochopena, a tak bylo nutné realizaci procesů kontrolovat a upozorňovat na kritická místa. Mnoho komplikací způsobila také nedokonalá znalost všech negativních scénářů ze strany metodiků a iniciátorů změn. Z tohoto důvodu bylo nutné vymýšlet některá řešení a správné postupy ad-hoc, což vedlo k opětovnému nevhodnému nastavení. Zavádění všech agilních technik trvalo přibližně 1 rok, a především v první třetině byla produktivita týmu nižší než před zavedením změn.

Po úspěšném aplikování změn se produktivita týmu výrazně zvýšila i oproti stavu před zavedením agilních technik, tým začal být samoorganizující a mnoho technik využívají členové automaticky. Nespornou výhodou je bezproblémová reakce na změny zákazníka v zadání nové funkcionality v průběhu zahájeného vývoje, což ve výsledku znamená dodání funkcionality dle přesné představy zákazníka a s minimálními náklady na vývoj.



## 7 Závěry a doporučení

Cílem této práce bylo aplikování agilních metodik do projektového řízení vybraného projektu a v případě úspěchu aplikovat osvědčené metody i na další projekty vybrané firmy. Z tohoto pohledu se dá vyhodnotit závěrečná práce jako splněna. Avšak očekávání, především v prvotní fázi, byla výrazně odlišná.

Literatura nabízí nespočet „best practices“ agilního řízení projektu a již tento počet předvídá, že řízení IT projektů bývá vždy originální. Projekt Online kasino nebyl v tomto směru výjimkou. V začátcích bylo nutné změnit myšlení členů týmu – oprostít se od předem připravené a vysoce kvalitní zadávací dokumentace a preferovat vytvoření iniciální hmatatelné nové funkcionality, kterou bude možné prezentovat. Na základě prezentace získat zpětnou vazbu, kterou zapracovat dále do vývoje. Překonání této změny bylo náročné, avšak po zpětném vyhodnocení je možné jej označit za klíčové.

Další klíčovou oblastí v reorganizaci bylo nastavení širokého využití nástroje Azure DevOps. I zde bylo v prvotních fázích složité argumentovat členům týmu, že je vhodnější moduly a komponenty využívat jinak, než byli zvyklí. Nejvyšší náročnost byla kladena na Product Ownera v tvoření strukturovaných elementů (Feature – User Story) a v definování rozsahu plánované verze. Microsoft nabízí pro začínající týmy předem připravené šablony, které byly v průběhu využívání vyhodnoceny jako nedostatečné, a proto byly tyto šablony dále upravovány, což způsobilo další změny v projektovém řízení k nelibosti některých členů týmu.

Pro nastavení celého modulu v Azure DevOps určeného ke správě požadavků a projektovému řízení doporučuje autor této práce hlubší studium zavádění změn v podpůrných nástrojích, které jsou často realizovány formou modelových her. Tyto hry slouží k zachycení potřeb týmu a následně je snazší připravit vhodné prostředí projektového řízení.

## 8 Seznam použité literatury

- [1] AGILE DIGEST. *Story Hierarchy* [online]. 2022 [cit. 2022-04-24]. Dostupné z: <https://agiledigest.com/agile-digest-tutorial-2/user-story/story-hierarchy/>.
- [2] AKSHAYA, H L, Jagadish S NISARGA, J VIDYA a K VEENA. *A Basic Introduction to DevOps Tools* [online]. 2015 [cit. 2022-03-23]. DOI: 10.1.1.735.2925. ISSN 0975-9646.
- [3] ALLEN, Michael. *Journal of Information Technology. Project Planning and Best Practices* [online]. 2018. [cit. 2022-0-16]. Dostupné z: <http://web.a.ebscohost.com/ehost/pdfviewer/pdfviewer?sid=7a284b48-5199-40e6-9555-70849481f1cd%40sessionmgr4003&vid=0&hid=4114>.
- [4] ARVIND. *DevOps Life cycle: Everything You Need To Know About DevOps Life cycle Phases* [online]. 2019 [cit. 2022-03-22]. Dostupné z: <https://www.edureka.co/blog/devops-lifecycle/>.
- [5] ATLASSIAN. *DevOps Maturity Model* [online]. 2019 [cit. 2022-03-20]. Dostupné z: <https://www.atlassian.com/devops/maturity-model>.
- [6] BARNLUND, D. C. A transactional model of communication. In *Communication theory*. Routledge. 2017.
- [7] BERRY, G. R. Enhancing effectiveness on virtual teams: Understanding why traditional team skills are insufficient. *The Journal of Business Communication*, 48(2). 2011.
- [8] BRYDE, D. et al. The project benefits of building information modelling (BIM). *International journal of project management*, 31(7). 2013.

- [9] CARROLL, J. and MORRIS, D. Agile project management in easy steps. 2nd ed. *Leamington Spa: In Easy Steps (In easy steps)*. 2015.
- [10] CROWDER, Alain., FRIESS, Sam. *Agile Project Management and other tools: Managing for Success in Practice*. 2015. Springer International Publishing.
- [11] [CZONŁKOWSKI](https://www.boldare.com/blog/how-to-improve-productivity-in-agile-team/), Romuald. *How to improve productivity in agile scrum teams* [online]. 2021 [cit. 2022-04-25]. Dostupné z: <https://www.boldare.com/blog/how-to-improve-productivity-in-agile-team/>.
- [12] DOLEŽAL, Jan. *Projektový management: komplexně, prakticky a podle světových standardů*. Praha: Grada Publishing, 2016. Expert (Grada). ISBN 978-80-247-5620-2.
- [13] DOLEŽAL, Jan, Pavel MÁCHAL a Branislav LACKO. *Projektový management podle IPMA. 2., aktualiz. a dopl. vyd.* Praha: Grada, 2012. Expert (Grada). ISBN 978-80-247-4275-5.
- [14] FARASHAH, Ali Dehghanpour, Janice THOMAS a Tomas BLOMQUIST, 2019. *Exploring the value of project management certification in selection and recruiting*. International Journal of Project Management [online]. 37(1), 14-26 [cit. 2022-03-03]. ISSN 02637863. Dostupné z: doi:10.1016/j.ijproman.2018.09.005.
- [15] GOOGLE. *Migrating all G Suite domains from classic Hangouts to Hangouts Meet* [online] 2019. [cit. 2022-03-20]. Dostupné z: <https://gsuiteupdates.googleblog.com/2018/05/migrating-all-g-suite-domains-from.html>
- [16] HAMMARBERG, M. *Kanban in Action*. 2014. USA: Manning Publications. ISBN 9781617291050.

[17] HAYES, R. M. Measurement of information and communication: A set of definitions. In *Between communication and information*. Routledge. 2017.

[18] ISLAM, Gibrail a Tim STORER, 2020. *A case study of agile software development for safety-Critical systems projects*. Reliability Engineering & System Safety [online]. 200 [cit. 2022-03-12]. ISSN 09518320. Dostupné z: doi:10.1016/j.ress.2020.106954.

[19] JIRA. *Jira Java APIs* [online]. 2019 [cit. 2022-03-17]. Dostupné z: <https://developer.atlassian.com/server/jira/platform/java-apis/>.

[20] JIRA. *Software* [online]. 2019 [cit. 2022-03-22]. Dostupné z : <https://cs.atlassian.com/software/jira>.

[21] KERZNER, H. *Project management: a systems approach to planning, scheduling, and controlling*. 12. vydání. Wiley & Sons, 2017. ISBN 978-1-119- 16535.

[22] KISSFLOW PROJECT. *The Definitive Guide to Scrumban Methodology* [online]. 2021 [cit. 2022-04-25]. Dostupné z: <https://kissflow.com/project/agile/guide-to-scrumban-methodology/>

[23] KNIBERG, H., SKARIN, M. *Kanban and Scrum - Making the Most of Both*, C4 Media Inc, 2010. ISBN 978-0-557-13832-6.

[24] KŘIVÁNEK, Mirko. *Dynamické vedení a řízení projektů: systémovým myšlením k úspěšným projektům*. Praha: Grada, 2019. ISBN 978-80-271-0408-6.

[25] LACKO, B., 2017. Systémový a procesní přístup v metodě RIPRAN. *Acta Informatica Pragensia*, 6(1), 86-93. Dostupné z: doi: 10.18267/j.aip.102.

[26] MACHÁČKOVÁ, Eva, 2014. Osedlejte si každý projekt. *Computerworld*. Vol. 25, No. 19-20, s. 10-11.

[27] MACHÁČKOVÁ, Eva a Zdeněk MACHÁČEK, 2017. KANBAN: Inspirace z výroby pro efektivní řízení práce v rámci týmu. *IT Systems*. 2-3.

[28] MÁCHAL, Pavel, KOPEČKOVÁ, Martina a PRESOVÁ, Radmila. *Světové standardy projektového řízení: pro malé a střední firmy: IPMA, PMI, PRINCE2*. Praha: Grada, 2015. Manažer. ISBN 978-80-247-5321-8.

[29] MICROSOFT. *Microsoft 365 Business and Microsoft Teams* [online]. 2019 [cit. 2022-03-15]. Dostupné z: <https://www.microsoft.com/cs-cz/microsoft-365/business#pmg-cmp-desktop>.

[30] MYSLÍN, J. *Scrum a Kanban: průvodce agilním vývojem softwaru*. Brno: Computer Press, 2016. ISBN 978-80-251-4650-7.

[31] PROJECT MANAGEMENT INSTITUTE. *A Guide to the Project management body of Knowledge, sixth edition*. Newton Square: Project Management Institute, 2017. ISBN : 978-1-62825-184-5.

[32] RADUJKOVIĆ, Mladen a Mariela SJEKAVICA, 2017. *Project Management Success Factors*. *Procedia Engineering* [online]. 196, 607-615 [cit. 2022-03-13]. ISSN 18777058. Dostupné z: doi:10.1016/j.proeng.2017.08.048.

[33] ROBERT, M. *Agile software development, principles, patterns, and practices*. Harlow: Pearson, 2014. ISBN 9781292025940.

[34] ŘEHÁČEK, Petr, 2019. *P3M: řízení projektu, řízení programu, řízení portfolia*. I. vydání. Jesenice: Ekopress. ISBN 978-80-87865-49-1.

[35] SLACK. *Voice, video & screen sharing apps*. In: Slack [online]. 2020 [cit. 2022-03-18]. Dostupné z: <https://get.slack.help/hc/en-us/articles/208492868-Voice-video-screen-sharing>.

[36] SUN MARKETING, s.r.o. *User Story* [online]. 2017 [cit. 2022-04-24]. Dostupné z: <https://www.sun.cz/nastroje/navody-pro-klienty/user-story>

[37] ŠOCHOVÁ, Zuzana. *Kdo píše User Story? A kdo tasky?* [online]. 2011 [cit. 2022-04-24]. Dostupné z: <https://soch.cz/blog/management/agile/scrum-management/kdo-pise-user-story-a-kdo-tasky/>.

[38] ŠOCHOVÁ, Zuzana a KUNCE, Eduard. *Agilní metody řízení projektů*. 2. vydání. Brno: Computer Press, 2019. ISBN 978-80-251-4961-4.

[39] TRELLO. *Jak využít Trello* [online]. 2018 [cit. 2022-03-16]. Dostupné z: <https://trello.com>.

[40] WAUTELET, Yves, Samedi HENG, Soreangsey KIV a Manuel KOLP, 2017. *Userstory driven development of multi-agent systems: A process fragment for agile methods*. *Computer Languages, Systems & Structures* [online]. 50, 159-176 [cit. 2022-03-12]. ISSN 14778424. Dostupné z: doi:10.1016/j.cl.2017.06.007.

[41] YAMASHITA, S., MABEE, D. A KOL. *Define features and epics, organize your product and portfolio backlogs in Azure Boards*. 2022 [cit. 2022-04-24]. Dostupné z: <https://docs.microsoft.com/en-us/azure/devops/boards/backlogs/define-features-epics?view=azure-devops&tabs=agile-process>.

## 9 Seznam obrázků

Obrázek 1: Firemní struktura .....	36
Obrázek 2: Týdenní projektový plán v PDF .....	40
Obrázek 3: Stromová struktura backlogu .....	46
Obrázek 4: Týdenní plán v Azure DevOps.....	47
Obrázek 5: Detail úkolu v Azure DevOps.....	50
Obrázek 6: Přehled vyřešených úkolu v Azure DevOps.....	51
Obrázek 7: Roadmapa v Azure DevOps.....	52
Obrázek 8: Nová organizační struktura platná od 1.4.2022 .....	57
Obrázek 9: Cyklus pravidelných schůzek.....	61
Obrázek 10: Struktura evidence požadavků na vývoj .....	65
Obrázek 11: Vizualizace požadavků na vývoj.....	68

## Zadání diplomové práce

**Autor:** Bc. Filip Kmínek

**Studium:** I1900354

**Studijní program:** N0688A140001 Informační management

**Studijní obor:** Informační management

**Název diplomové práce:** **Projektové řízení zaměřené na komunikaci vybraného podniku**

**Název diplomové práce AJ:** Project management focused on communication of chosen company AJ:

### Cíl, metody, literatura, předpoklady:

#### Cíl práce:

Hlavním cílem diplomové práce je popis stávajícího stavu projektového řízení vybraného podniku a navržení optimalizačních metod a nástrojů pro zvýšení efektivnosti řízení projektů.

#### Osnova:

1. Úvod
2. Cíl práce
3. Metodika zpracování
4. Teoretická část
5. Praktická část
6. Závěr
7. Seznam použité literatury

1. ŠOCHOVÁ, Zuzana a KUNCE, Eduard. *Agilní metody řízení projektů*. 2. vydání. Brno: Computer Press, 2019. ISBN 978-80-251-4961-4.
2. KŘIVÁNEK, Mirko. *Dynamické vedení a řízení projektů: systémovým myšlením k úspěšným projektům*. Praha: Grada, 2019. ISBN 978-80-271-0408-6.
3. MÁCHAL, Pavel, KOPEČKOVÁ, Martina a PRESOVÁ, Radmila. *Světové standardy projektového řízení: pro malé a střední firmy: IPMA, PMI, PRINCE2*. Praha: Grada, 2015. Manažer. ISBN 978-80-247-5321-8.

**Garantující pracoviště:** Katedra managementu,  
Fakulta informatiky a managementu

**Vedoucí práce:** Ing. Václav Zubr, Ph.D.

**Datum zadání závěrečné práce:** 31.3.2020