

# **Inovace workflow pro řízení projektů**

**Diplomová práce**

**Vedoucí práce:**

**doc. Ing. Ivana Rábová, Ph.D.**

**Bc. Veronika Prokopová**

**Brno 2016**



Na tomto místě bych ráda poděkovala vedoucí doc. Ing. Ivaně Rábové, Ph. D. za odborné vedení, ochotu a poskytnutí cenných rad a připomínek při zpracování této diplomové práce. Také děkuji své rodině a přátelům, kteří mě v průběhu celého studia a při tvorbě práce podporovali.



## Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Inovace workflow pro řízení projektů** vypracovala samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací*.

Jsem si vědoma, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 20. května 2016

---



## **Abstract**

Prokopová, V. The innovation of project management workflow. Diploma thesis. Brno: Mendel University, 2016.

This thesis deals with analysis of project management process in middle-sized company. The process is modeled using of BPMN 2.0. Based on these models are proposed innovations. The part of innovations is the implementation of agile methods Scrum and implementation of web application for the project team support. According to the analysis and the requirements specification are created Use case diagrams. The structure and behavior of the application is modeled by IFML diagrams using WebRatio tool. The application is implemented using language PHP and Nette Framework, using MySQL database server.

## **Keywords**

Project management, agile methods, Scrum, BPMN, WebRatio, IFML, PHP, Nette Framework, MySQL.

## **Abstrakt**

Prokopová, V. Inovace workflow pro řízení projektů. Diplomová práce. Brno: Mendelova univerzita v Brně, 2016.

Diplomová práce se zabývá problematikou projektového řízení ve středně velké firmě. Na základě analýzy procesu a jeho modelování pomocí notace BPMN 2.0, jsou navrženy inovační opatření. Součástí inovací je nasazení agilní metodiky Scrum a tvorba webové aplikace pro podporu projektového řízení. Dle specifikace požadavků jsou vytvořeny Use case diagramy. Chování a struktura aplikace je zachycena v IFML modelech vytvořených v Case nástroji WebRatio. Samotná aplikace je následně realizována pomocí jazyka PHP a Nette Frameworku za použití databázového serveru MySQL.

## **Klíčová slova**

Projektový management, agilní metodiky, Scrum, BPMN, WebRatio, IFML, PHP, Nette Framework, MySQL.





# Obsah

<b>1</b>	<b>Úvod</b>	<b>11</b>
<b>2</b>	<b>Cíl práce</b>	<b>12</b>
<b>3</b>	<b>Metodika</b>	<b>13</b>
<b>4</b>	<b>Teoretická východiska práce</b>	<b>14</b>
4.1	Projektové řízení .....	14
4.1.1	Projekt .....	14
4.1.2	Projektový manažer .....	16
4.2	Oblasti projektového řízení .....	16
4.3	Procesní řízení.....	18
4.3.1	Workflow.....	19
4.4	Projektové fáze a životní cyklus projektu.....	19
4.5	Metodiky řízení a vývoje projektů.....	21
4.6	Agilní metodiky.....	22
4.6.1	SCRUM Development Process.....	24
4.7	Nasazení metodiky Scrum .....	27
<b>5</b>	<b>Použité technologie a nástroje pro řešení</b>	<b>28</b>
5.1	BPMN .....	28
5.2	IFML.....	29
5.3	Case nástroj WebRatio .....	32
5.4	PHP.....	32
5.5	Apache .....	32
5.6	MySQL.....	32
5.7	Nette Framework .....	32
<b>6</b>	<b>Vlastní řešení</b>	<b>34</b>
6.1	Charakteristika společnosti.....	34
6.1.1	Struktura organizace .....	34
6.2	Analýza procesu řízení projektů .....	35

---

6.2.1	Subproces Specifikace požadavků .....	35
6.2.2	Subproces Uzavření smlouvy.....	36
6.2.3	Subproces Realizace.....	37
6.2.4	Subproces Předání zákazníkovi.....	38
6.2.5	Subproces Poimplementační podpora.....	39
6.3	Návrh inovačních opatření .....	40
6.3.1	Návrh inovace subprocesu Realizace .....	43
6.4	Současný stav IS/ICT.....	43
6.5	Sběr a analýza požadavků .....	44
6.6	Identifikace uživatelů a rolí.....	44
6.7	Diagramy případů užití .....	45
6.7.1	Popis případů užití.....	48
6.7.2	Scénáře případů užití.....	50
6.8	Datová struktura.....	51
6.9	Modely IFML.....	53
6.9.1	Nepřihlášený uživatel .....	54
6.9.2	Vývojář.....	54
6.9.3	Product Owner .....	56
6.9.4	Scrum Master .....	58
6.10	Implementace .....	60
6.11	Testování a nasazení aplikace .....	63
6.12	Návrhy na vylepšení.....	63
<b>7</b>	<b>Diskuze</b>	<b>64</b>
<b>8</b>	<b>Závěr</b>	<b>66</b>
<b>9</b>	<b>Literatura</b>	<b>67</b>
<b>10</b>	<b>Seznam obrázků</b>	<b>70</b>
<b>11</b>	<b>Seznam tabulek</b>	<b>72</b>
<b>A</b>	<b>Procesní diagramy</b>	<b>74</b>
<b>B</b>	<b>Ukázky aplikace</b>	<b>80</b>

# 1 Úvod

Projektové řízení se stalo v dnešní době nedílnou součástí téměř všech organizací. Vstupem České republiky do Evropské unie vzrostl počet firem na trhu a s tím roste i nutnost silné konkurenceschopnosti. Právě projektový management je cestou k posílení konkurenceschopnosti na trhu. Na trhu dochází ke stále se rozvíjejícímu podnikatelskému prostředí a rostoucím požadavkům zákazníků. Ve společnosti také dochází k neustálému vývoji a inovacím v oblasti technologií a nových produktů či služeb. Na tyto změny je potřeba rychle a flexibilně reagovat. Firmy se potřebují mezi ostatními odlišit a úspěšné projektové řízení jim může pomoci realizovat úspěšné projekty, obstát a vyniknout mezi ostatními společnostmi a konkurencí.

Význam projektového řízení lze spatřit i v minulosti. V minulém století bylo projektové řízení primárně využíváno ve vojenském, počítačovém a stavebním průmyslu. V roce 1910 byl americkým strojním inženýrem Henry Ganttem vyvinut známý Ganttův diagram. Původně se tyto diagramy kreslily ručně a sloužily pro zobrazení informací, plánování a kontrolu práce v oblasti průmyslu, infrastruktury a především v rámci vojenských projektů. Stejně tak hojně využívané síťové grafy byly poprvé použity v roce 1958, a to v rámci projektu vývoje ponorkových raket realizovaným námořnictvem Spojených států amerických.

V dnešní době projektové řízení nachází aplikaci v rozsáhlejší oblasti. Projektové řízení se používá ve velkých, středních, ale i v malých organizacích. Uplatnění nachází především při zavádění nových výrobních procesů, rozšiřování prodejního sortimentu, realizaci vývoje nových technologií a postupů, ale i provádění organizačních změn a realizaci marketingového plánu.

V oblasti informačních a komunikačních technologií je projektové řízení téměř nutností. Využívá se především při vývoji a zavádění nových systémů, modernizaci stávajících, implementace datové infrastruktury apod. Při vývoji projektů je potřeba aktivně přistupovat k měnícímu se podnikovému prostředí. Z toho vyplývá, že je důležité i flexibilně reagovat na změny, které ovlivňují projektové prostředí a tím i celý vývoj a průběh realizace projektu. Pro řízení projektů existuje několik metod. Každý projekt je však jedinečný, a proto je potřeba ke každému přistupovat individuálně a zvážit vhodnou metodu z hlediska charakteru daného projektu.

## 2 Cíl práce

Cílem diplomové práce je návrh inovace řízení projektů ve zvolené firmě zabývající se vývojem softwarových produktů. Budou navrženy inovační kroky procesu zajišťujícího řízení projektů. K dosažení tohoto cíle bude provedena analýza současného stavu a následně vytvořen model procesu řízení projektů, a to pomocí notace BPMN (Business Process Model and Notation) a příslušných diagramů.

Součástí inovace bude také zavedení agilní metodiky Scrum. S ohledem na tuto metodiku budou navrženy změny workflow řízení projektů. Navrhovaná inovační opatření budou promítnuta v novém procesním modelu.

Dílním cílem práce je návrh prototypu webové aplikace, která bude sloužit jako podpora pro členy vývojového týmu a jejímž účelem bude zajištění hlavních oblastí workflow řízení projektů při dodržení metodiky Scrum. Návrh aplikace bude proveden prostřednictvím příslušných diagramů notace IFML (The Interaction Flow Modeling Language).

### 3 Metodika

Diplomová práce se zabývá inovacemi v projektovém řízení. Práce je rozdělena do několika kapitol. V první části budou podrobně rozebrána teoretická východiska důležitá k naplnění cílů diplomové práce, která budou následně použita v části praktické. Jedná se především o vysvětlení problematiky projektového řízení. Budou blíže definovány příslušné pojmy, oblasti projektového řízení, význam procesního řízení a životní cyklus projektu. Součástí kapitoly bude také nastínění principů agilních metodik. S ohledem na cíl práce bude tato část detailněji věnována agilní metodice Scrum a jejímu nasazení.

Před samotnou realizací praktické části je důležité zvolit vhodné nástroje a technologie, které budou pro řešení použity. Všechny tyto prostředky budou blíže popsány v samostatné kapitole.

V úvodu praktické části práce bude na základě analýzy současného stavu provedeno modelování procesu řízení projektů. K znázornění procesu bude využita notace BPMN (Business Process Modeling), pomocí které budou vytvořeny příslušné procesní diagramy. Součástí inovace bude také zavedení agilní metodiky Scrum. Všechna navržená inovační opatření budou promítnuta v novém procesním modelu s dodržáním principů metodiky Scrum.

V další části bude navržen prototyp webové aplikace, která bude sloužit jako podpora pro členy projektového týmu a jejímž cílem bude zajištění hlavní oblasti workflow řízení projektů, a to části realizační. V souladu s požadavky budou vytvořeny Use Case diagramy včetně scénářů případů užití znázorňující funkčnost aplikace. Use Case diagramy budou vytvořeny pomocí nástroje Enterprise Architect.

Návrh aplikace bude proveden prostřednictvím notace IFML (The Interaction Flow Modeling Language) pomocí nástroje WebRatio. Následně bude navržená aplikace implementována pomocí Nette Frameworku. V závěru práce bude vytvořená aplikace testována v podnikovém prostředí a na základě připomínek budou diskutována možná vylepšení aplikace.

## 4 Teoretická východiska práce

V této části práce bude blíže specifikována problematika projektového řízení a s ní související klíčové pojmy. Úvodní část této kapitoly je určena základním definicím projektového řízení, jeho součástí a hlavním principům. Další část se soustřeďuje na hlavní znalostní oblasti týkající se projektového řízení. V návaznosti na tuto část je uveden i rozbor agilních metodik pro řízení projektů.

### 4.1 Projektové řízení

Projektové řízení neboli projektový management je souhrn činností, jako je plánování, organizování, řízení a kontrola zdrojů společnosti. Úkolem projektového řízení je aplikace znalostí, dovedností, nástrojů a technik při realizaci projektových aktivit za účelem dosažení požadavků projektu. (Project Management Institute, 2008)

Projektové řízení je představováno působením a integrací základních pěti elementů, kterými jsou:

- projektová komunikace – důležitý prostředek efektivního dorozumívání účastníků projektu.
- týmová spolupráce – nutnost kooperace všech účastníků projektu k dosažení sdílených cílů.
- životní cyklus projektu – logická posloupnost jednotlivých fází projektu, včetně definovaných stavů a podmínek pro přechod z jedné fáze do druhé.
- nástroje řízení projektů aplikovaných v průběhu životního cyklu projektu.
- organizační závazek – odpovědnost projektového manažera za splnění cíle projektu s ohledem na organizační kulturu společnosti a poskytnuté zdroje vyhrazené pro realizaci projektu. (Mooz, Forsberg, Cotterman, 2003)

#### 4.1.1 Projekt

Nejdůležitějším prvkem projektového řízení je projekt. Projekt je definován jako jedinečný sled aktivit a úkolů, jejichž výkonem jsou projektové zdroje transformovány na výstupy. Hlavním požadavkem na projekt je specifikace jeho cíle, který má být po jeho úspěšné realizaci splněn. Dalším kritériem projektu je vymezení data začátku a ukončení projektu. V neposlední řadě je nutností projektu stanovení rámce pro čerpání zdrojů, které jsou nutné pro jeho realizaci. (Kerzner, 2013)

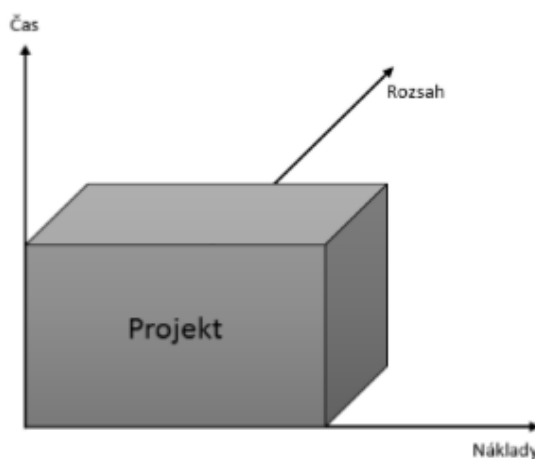
Project Management Institute (2008) definuje projekt jako dočasné úsilí vynaložené na vytvoření unikátního produktu, služby nebo určitého výsledku. Dočasnost v tomto pojetí znamená ohraničenost projektu, tzn. specifikaci data začátku a ukončení projektu.

Dočasnost a unikátnost jsou důležitými pojmy při definici projektu. Právě tyto dvě vlastnosti totiž činí projekt jedinečný a do jisté míry neopakovatelný. Výstup

projektu je produkt, který je pomocí realizace projektu vytvořen. Výsledkem projektu může být fyzický objekt, služba nebo výsledek, který je nadále použit jako vstup pro jiné procesy. Produkt projektu je stejně jako celý projekt unikátní a v případě opakování projektů, jsou jejich výsledky podobné, nikoliv však totožné. Veškeré náležitosti lze shrnout jako vlastnosti projektu, kterými jsou:

- datum zahájení a ukončení,
- existence projektu pro určitou dobu, odpovídající době trvání projektu,
- definovaný cíl a rozsah projektu,
- specifické potřeby a cíle, jejich naplnění je účelem projektu,
- dočasná existence projektového týmu,
- specifické vlastnosti a rozsah aplikovaných zdrojů,
- existence jedinečného projektového okolí zahrnující vlivy působící na projekt,
- organizační struktura projektu
- dostupnost zdrojů, které jsou projektu přiděleny. (Svozilová, 2011)

Každý projekt je omezen časem, předpokládanými náklady a plánovaným rozsahem projektu. Rozsahem je definována práce, jakou je v rámci realizace projektu nutné vykonat. Čas vyjadřuje období, během něhož se projekt realizuje. Náklady jsou představovány rozpočtem projektu. Tyto tři základní charakteristiky vymezují prostor, v němž se realizuje výstup projektu. Kathy Schwalbe (2011) definuje tyto tři omezení jako projektový trojimperativ, který je uveden na následující obrázku.



Obr. 1 Projektový trojimperativ  
Zdroj: Drahoslav Dvořák, 2008

S projektovým trojimperativem je úzce spjat úspěch projektu. Právě dodržení tohoto trojimperativu a dosažení všech tří cílů vede k úspěšnému projektovému ří-

zení. Takové řízení lze pak definovat jako dosažení plánovaného cíle a rozsahu projektu, a to při dodržení časového limitu, předpokládaných nákladů a s akceptací zákazníka projektu. (Svozilová, 2011)

#### 4.1.2 Projektový manažer

Významnou osobou v projektovém řízení je projektový manažer, který je odpovědný za plnění cílů projektu. Mezi hlavní úkoly a odpovědnosti projektového manažera patří:

- zajištění tvorby produktu projektu,
- řízení zdrojů projektu – času, pracovní síly, finančních a hmotných prostředků, informačních technologií apod.,
- plánování a kontrola postupu projektu – z hlediska efektivního využití zdrojů a optimalizace výkonu,
- snížení projektových rizik a optimální řešení problémových situací,
- řízení vztahů mezi projektem a jeho okolím, včetně vztahů se zákazníkem,
- zastoupení zájmů zákazníka - zároveň v souladu se zájmy společnosti. (Svozilová, 2011)

## 4.2 Oblasti projektového řízení

PMBOOK (2008) definuje a popisuje jednotlivé znalostní oblasti projektového řízení, které obsahují určité aktivity vedoucí ke specifickým projektovým cílům. Každá oblast pak využívá konkrétní nástroje a techniky, které pomáhají projektovým manažerům a jejich týmům při realizaci projektu. Toto rozdělení je blíže specifikováno v následující části.

### Integrované řízení projektu

Integrované řízení projektu spočívá v koordinaci všech oblastí napříč celým životním cyklem projektu a zajišťuje, že se všechny jeho části včas propojí a dojde k úspěšné realizaci. Součástí integrovaného řízení je zpracování a schválení zadávací listiny projektu, která představuje formální zahájení práce na projektu. (Doležal, Máchal, Lacko, 2012)

### Řízení rozsahu projektu

Řízení rozsahu projektu je klíčovou činností při řízení projektu. Rozsah projektu je souhrn všech činností a prací, které je potřeba vykonat pro dosažení cíle projektu. Řízení rozsahu pak zahrnuje veškeré procesy, jejichž úkolem je definice a kontrola rozsahu projektu. Součástí řízení rozsahu projektu je definice a následné zdokumentování požadovaných vlastností a funkcí projektu. Dalším krokem je detailní formulace rozsahu projektu a vytvoření hierarchické struktury prací. Tato struktura



ra dále poskytuje základy pro plánování a řízení harmonogramů, nákladů a zdrojů projektu. (Gordon, 2014)

Během celého životního cyklu projektu pak dochází k neustálému ověřování rozsahu projektu. V průběhu může dojít ke změnám jak v požadavcích zákazníka, tak i v použitých postupech či technologiích. Tyto faktory mohou způsobit změny v rozsahu projektu, které je potřeba správně zpracovat a zajistit jejich integraci do projektu. (Schwalbe, 2011)

### **Řízení času projektu**

Řízení času projektu je nezbytné ke včasnému dokončení projektu. Základem pro monitorování postupu projektu je vytvořený harmonogram, který obsahuje posloupnost předem definovaných aktivit, odhadovaný čas a požadované zdroje potřebné pro jejich vykonání. (Schwalbe, 2011)

Pro tvorbu harmonogramu se využívá několik technik a nástrojů, mezi které patří nejběžněji používaný Ganttův diagram a analýza kritické cesty. Ganttův diagram znázorňuje jednotlivé aktivity projektu, včetně jejich data zahájení a ukončení. (Janišová 2013). Metoda kritické cesty slouží k odhadu celkové doby, za kterou je možné projekt dokončit. Kritická cesta zahrnuje aktivity, které nemají žádnou časovou rezervu a tudíž je to nejkratší možná doba trvání projektu. (Dvořák, 2007)

### **Řízení nákladů projektu**

Úkolem řízení nákladů projektu je zajistit, aby projektový tým dokončil projekt v rámci schváleného rozpočtu. V první fázi tvorby rozpočtu projektu probíhá odhad nákladů projektu. Tvorba rozpočtu může proběhnout pomocí odhadů na základě podobnosti s předešlými projekty nebo pomocí matematických a statistických metod. V průběhu realizace projektu může být rozpočet upraven, ale pouze v souladu s ujednanými podmínkami a na základě předem podepsané smlouvy. Odpovědnost za sestavení cenové nabídky projektu má projektový manažer, který následně daný návrh předkládá ke schválení managementu společnosti. (Svozilová, 2011)

### **Řízení kvality projektu**

Řízení kvality projektu probíhá během celého životního cyklu. Kerzner (2013) definuje řízení kvality jako soubor plánovaných a systematických činností k zajištění realizace projektu splňujícího požadované standardy kvality. Norma ISO 9000 (2016) uvádí kvalitu jako souhrn vlastností a charakteristik, které jsou schopny uspokojit požadavky zákazníka.

### **Řízení lidských zdrojů v projektu**

Lidské zdroje hrají významnou roli při realizaci projektu. Právě lidé určují úspěch či neúspěch projektu. Součástí řízení lidských zdrojů je vytvoření plánu a organi-

začního schématu, který obsahuje a specifikuje veškeré pracovníky v projektu včetně jejich přidělených rolí. (Schwalbe, 2011)

### **Řízení komunikace v projektu**

S řízením případných konfliktů úzce souvisí řízení komunikace v projektovém týmu. Schwalbe (2011) uvádí, že největší hrozbou při neúspěchu projektu je právě selhání komunikace. Při iniciaci projektu je důležité identifikovat všechny zainteresované osoby a formulovat strategii řízení jejich komunikace.

Významnou součástí realizace projektu jsou také projektová jednání či schůze, jejichž obsahem může být řízení předmětu projektu, koordinace dílčích úkolů či projektová kontrola. Projektová jednání jsou základním prostředkem projektové komunikace. Tyto jednání se však také mohou stát zdrojem konfliktu. A to především v případě jejich špatné organizaci, přípravě a plánování. Schůzky by proto měly být předmětově zaměřeny a určeny pouze zainteresovaným pracovníkům týmu. (Svozilová, 2011)

### **Řízení rizik projektu**

Řízení rizik je důležitá složka projektového managementu, neboť rizika jsou nedílnou součástí realizace každého projektu. Součástí řízení rizik je jejich identifikace, kvalitativní a kvantitativní analýza, monitorování a v neposlední řadě jejich kontrola. Cílem identifikace rizik, je určení rizik, která mohou ovlivnit průběh realizace projektu. Po stanovení rizik následuje jejich kvalitativní analýza, jejíž úkolem je seřazení rizik dle závažnosti a pravděpodobnosti jejich výskytu a dopadu. Na kvalitativní analýzu rizik navazuje analýza kvantitativní, která zahrnuje vyčíslení odhadů dopadu jednotlivých rizik. (Svozilová, 2011)

### **Řízení dodávek projektu**

Řízení dodávek projektu zahrnuje všechny procesy potřebné k zajištění doplňkových produktů a služeb nezbytných k realizaci projektu. Toto obstarávání bývá řešeno buď formou přímého nákupu od dodavatelů, nebo prostřednictvím outsourcingu. Formou outsourcingu bývá nejčastěji zajišťován vývoj aplikací. Součástí jsou i poskytované služby související s vývojem softwaru jako je například klientská podpora, zajištění bezpečnosti dat, jejich zálohování či obnova. (Schwalbe, 2011)

## **4.3 Procesní řízení**

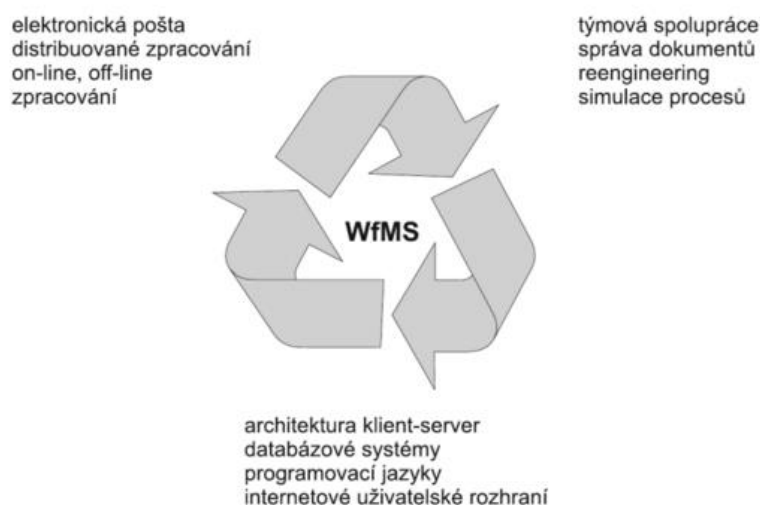
S projektovým řízením také úzce souvisí pojem procesní řízení. Tyto pojmy je však nutné rozlišovat. Projekt je sled činností a aktivit vedoucí ke splnění určitého cíle. Oproti tomu je proces definován jako posloupnost činností určených k vykonání stanoveného sledu operací za účelem vykonání dané práce. Vycházíme-li z těchto definic, pak lze říci, že průběh realizace projektu je určitým druhem procesu, který

má specifikovanou dobu vykonávání a přiděleny zdroje. Obecně lze tedy konstatovat, že realizace projektu má charakter procesu. Každý proces je charakterizován detailním popisem jeho průběhu, transformačních pravidel, jeho vlastností a vztahů mezi jednotlivými prvky procesu.

### 4.3.1 Workflow

Termín workflow lze zjednodušeně definovat jako tok a automatizované řízení informací v podnikovém procesu. Systémy, které tuto automatizaci zajišťují, se nazývají workflow management systémy (WfMS) a jsou provozovány prostřednictvím softwarových systémů. Jejich práce spočívá v organizaci a řízení posloupnosti jednotlivých pracovních činností, odpovídajících zdrojů a komunikaci s účastníky workflow. Systém řízení propojuje různorodé principy, metodiky a technologie, a podporuje tak mnoho funkcí a možností. Toto propojení je znázorněno na obrázku č. 2. Mezi tyto funkce patří mimo jiné řízení průběhu procesu, jeho definici, monitorování a vyhodnocení jeho průběhu. (Carda, Kunstová, 2003)

Zavedení workflow systému poskytuje několik výhod, mezi které se řadí zvýšení efektivity a kvality práce, snížení nákladů, zjednodušení podnikových procesů či zlepšení organizace pracovních činností a postupů.

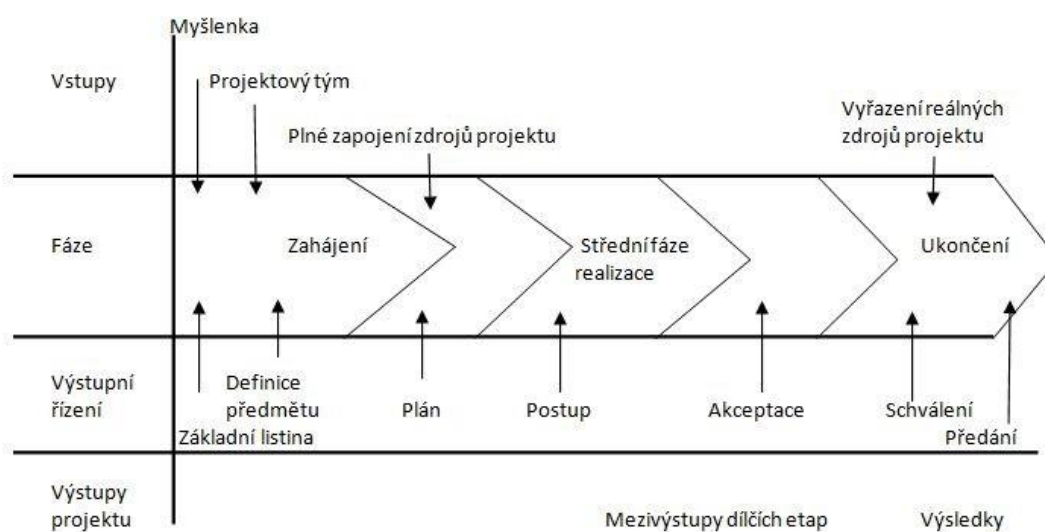


Obr. 2 Workflow systém  
Zdroj: Carda, Kunstová, 2003

## 4.4 Projektové fáze a životní cyklus projektu

Jak už bylo řečeno výše, projekt je sled činností a aktivit vedoucí ke splnění určitého cíle. Během jeho tvorby se projekt neustále vyvíjí a nachází se tak v různých fázích. Tento tok se nazývá životním cyklem projektu. Životní cyklus specifikuje dílčí fáze projektu, definuje výstupy a zainteresované strany jednotlivých fází. Každý projekt je jinak zaměřený a tudíž má i rozdílné fáze, které se liší v závislosti na

typu a předmětu projektu a oboru, v němž se projekt vyvíjí. PMBOOK (2008) definuje několik obecných fází životního cyklu. Těmi jsou – zahájení projektu, organizace a příprava, realizace projektu a jeho dokončení. Přejít mezi jednotlivými fázemi je uskutečněno pouze při dosažení a následného schválení definovaného výstupu či stavu předchozí fáze. Každý projekt během svého životního cyklu prochází specifickým vývojem, kde dochází ke změně řady jeho charakteristik. V první řadě dochází k čerpání předem přidělených zdrojů, dále dochází k úpravě míry nasazení a vlivu zainteresovaných skupin. Pro usnadnění kontroly dílčích procesů jednotlivých projektových fází, jsou tyto aktivity rozděleny do logické posloupnosti, která je znázorněna na následujícím obrázku č. 3.



Obr. 3 Typické rozložení fází životního cyklu projektu  
Zdroj: Alena Svozilová, 2011

Součástí projektů je realizace nových produktů a každý produkt má také svůj životní cyklus, kterým postupně prochází. V oblasti vývoje informačních systémů a aplikací, se rozlišuje několik typů životních cyklů softwarového produktu. Mezi první typ patří prediktivní životní cyklus, který jasně rozděljuje rozsah projektu a předem definuje jeho náklady a harmonogram. V tomto typu životního cyklu je zahrnuto několik dílčích modelů, mezi které patří například vodopádový model, spirálový model nebo přírůstkový model. Opakem prediktivního životního cyklu je adaptivní vývoj, jehož životní cyklus předpokládá adaptivní přístup. Tento přístup spočívá v postupném vytváření komponent a funkčnosti na základě požadavků zákazníka. V poslední době se stal moderní agilní softwarový vývoj, kterému je podrobněji věnovaná následující kapitola. (Schwalbe, 2011)

## 4.5 Metodiky řízení a vývoje projektů

V oblasti řízení projektů je využíváno několik druhů metodik a nástrojů. Metodika je obecně definována jako pracovní postup využívající soubor pravidel a nástrojů. Metodika vychází z modelu životního cyklu projektu a doporučuje nástroje, postupy a zavádí role a jejich povinnosti. Právě zvolená metodika je klíčovým faktorem úspěchu projektového managementu. Jednotlivé metodiky projektového řízení se liší použitými postupy a nástroji. Jedno mají ale společné, a to, že musí pokrývat následující oblasti projektového řízení:

- celý životní cyklus projektu,
- analýza rizik,
- definice organizační struktury, jednotlivých rolí a vztahů mezi nimi,
- určení odpovědnosti členů týmu projektu z hlediska jejich rolí. (Oškrdal, 2014)

V oblasti IS/ICT dochází k neustálému vývoji a zdokonalování softwarových produktů a služeb. Tradiční, tzv. rigorózní metodiky využívané pro vývoj softwaru se tak staly nedostačující, a proto vznikly agilní metodiky, které umožňují rychle vyvíjet software a dynamicky reagovat na měnící se požadavky. Hlavní odlišností agilních metodik od těch tradičních je tedy v přístupu k požadavkům. V případě tradičních metodik dochází k přesné specifikaci požadavků na počátku projektu a tyto požadavky jsou pak v průběhu realizace projektu neměnné, což je hlavní nevýhoda těchto metodik. Naopak agilní metodiky přistupují k požadované funkcionalitě dynamicky, což umožňuje flexibilně reagovat na změny a požadavky zákazníka.

Agilní vývoj předpokládá změnu požadavků v průběhu realizace a klade důraz na adaptaci na změny. Oproti tomu rigorózní metodiky jsou postaveny na definici veškerých požadavků na počátku vývoje a snaží se eliminovat změny v průběhu realizace projektu. Další oblastí, ve které se metodiky odlišují, je zaměření na dílčí procesy a jejich kvalitu. Použití rigorózních metodik vyžaduje podrobnou definici všech činností a předpokládá, že kvalitní procesy vedou ke kvalitnímu výsledku projektu. Naopak agilní metodiky se zaměřují na činnosti, které vytvářejí nějakou hodnotu a ty činnosti, které ji nevytvářejí, se snaží co nejvíce eliminovat. V projektech řízených agilními metodikami se projektový tým zaměřuje na hodnotu pro zákazníka a zákazník se tak stává řídicím subjektem celého projektu. To je zásadní rozdíl oproti rigoróznímu vývoji, kde je účast zákazníka pouze v počátečních a koncových fázích projektu. (Buchalceková, 2005)

S ohledem na obsah této práce budou v následující části blíže rozebrány agilní metodiky, konkrétně vybraná metodika Scrum Development Process (dále jen Scrum).

## 4.6 Agilní metodiky

Vývoj agilních metodik spadá do 90. let minulého století. Právě v tomto období došlo k výraznému technologickému rozvoji v oblasti IT. Informační technologie se tak staly dostupnější a více využívanější. Jak už bylo řečeno výše, rostoucí vývoj a změny technologií zapříčinil vznik nových metodik, které umožňují rychle vyvinout a realizovat řešení, které je adaptováno měnícím se požadavkům. Hlavní myšlenkou agilních metodik je co nejrychleji vyvinout daný produkt, předložit ho zákazníkovi a následně ho na základě jeho podnětů upravit. Většinou tedy dochází k dodání nekompletního produktu z hlediska jeho funkcionality. Agilní vývoj však umožňuje snadnější úpravy produktu během jeho používání. (Buchalcevdová, 2005)

### Současný stav užívání agilních metodik v ČR

Průzkumem využívání agilních metodik ve světě se zabývá americká společnost VersionOne, která se zabývá propagací a užíváním agilních nástrojů. Společnost každoročně vydává zprávu *State of Agile*, která na základě provedeného průzkumu uvádí, kolik z dotazovaných firem využívá agilní metodiky, jaké jsou nejčastěji používané a jaká je průměrná doba jejich využívání. Z výsledků provedeného průzkumu v roce 2015 vyšlo najevo, že nejvíce používanou metodikou je Scrum a následně extrémní programování. (VersionOne, 2015)

V posledních letech se odráží rostoucí trend používání agilních metodik i v České republice. V roce 2011 se v Praze uskutečnila první mezinárodní konference na podporu agilních metodik *Agile Prague*. Tato konference pak vedla ke vzniku sdružení Agilní Asociace, která je komunitou příznivců agilních přístupů a jejímž posláním je zvýšení povědomí o agilních metodách řízení. V současné době se konference Agile Prague koná každý rok a je zaměřená především na vývojáře, testery a manažery. Sdružení svými aktivitami podporuje společnosti, které chtějí zavést agilní přístupy a vytváří tak komunitu, jejichž členové si mezi sebou sdílí informace a zkušenosti. (Agilní Asociace, 2016)

V roce 2013 byl proveden rozsáhlý průzkum užívání agilních metodik v ČR. Za vznikem této aktivity stálo sdružení Agilní Asociace společně s českou společností Etnetera, která se zabývá vývojem webových portálů a aplikací. Průzkumu se zúčastnilo 171 respondentů. Z výsledků průzkumu vyšlo, že v používaných agilních přístupech jasně převažuje metodika Scrum, kterou z dotázaných využívá téměř 90 %. Na dalších místech je pak metodika Kanban, Extrémní programování a následně Scrumban (kombinace metodiky Scrum a Kanban). Další zajímavou analýzou je důvod zavedení agilních přístupů. Jako nejčastější důvod pro zavedení agilního přístupu respondenti uvádí zvýšení produktivity, kvality produktu, zjednodušení vývojového procesu, zlepšení organizace a komunikace v týmu a v neposlední řadě snížení rizik. Z hlediska porovnání s klasickými přístupy vyšlo najevo, že víc jak polovina dotázaných spatřuje výhody agilních přístupů v rychlosti dokončení, výsledné kvalitě produktu a ve spokojenosti zákazníka. (Pulkert, 2013)

## Principy agilních metodik

Pojem *být agilní* bývá definován jako rychlý, dynamický, přizpůsobivý, interaktivní a rychle reagující na změny. Základním pilířem agilních metodik je manifest agilního vývoje softwaru, který definuje čtyři důležité podstaty agilních metodik. Tyto vlastnosti jsou shrnuty v následujících bodech:

- Jednotlivci a interakce před procesy a nástroji – spolupracující týmy mají lepší výsledky než skupiny individuálních osob, procesy a nástroje nejsou pro jejich úspěch klíčové, ale pouze jim pomáhají k dosažení výsledků.
- Fungující software před vyčerpávající dokumentací – dokumentace k produktu je důležitá, ale neměla by převažovat nad samotnou funkčností produktu. Dokumentace má sloužit pouze jako podpůrný prostředek pro oblasti, které pro uživatele mohou být obtížnější a méně intuitivní.
- Spolupráce se zákazníkem před vyjednáváním o smlouvě – při vývoji produktu je důležité se zákazníkem komunikovat.
- Reagování na změny před dodržováním plánu – je potřeba se přizpůsobit a rychle reagovat na změny požadavků zákazníka. (Beck, 2001)

## Důvody pro zavedení agilní metodiky

S technologickým pokrokem a neustálými inovacemi v oblasti IS/ICT přibývá i požadavků na řízení projektů. Ze strany zákazníků roste tlak na rychlý a efektivní vývoj projektu. Při zadávání projektu častokrát zákazník nemá jasnou představu, co vlastně požaduje a očekává od výsledného produktu. Jeho požadavky se vyvíjí v průběhu životního cyklu projektu a je nutné na ně aktivně reagovat a přizpůsobit se jim. Vývoj probíhá inkrementálně a po určitých úsecích je zákazníkovi představena aktuální verze výsledného produktu, na jejímž základě může zákazník upravit své požadavky. V agilním vývoji je kladek důraz na komunikaci a spolupráci se zákazníkem. Právě zákazník je klíčovým prvkem procesu, který se podílí na realizaci projektu, a proto je vhodné, aby byl po celou dobu k dispozici. (Šochová 2014)

Jedním z hlavních důvodů při přechodu na jednu z agilních metodik je požadovaná výsledná kvalita produktu, která je zajištěna testováním v průběhu celého životního cyklu projektu. Časté testování, tak napomáhá k dřívějšímu odhalení chyb a rychlejší opravě. Jak už bylo zmíněno, zákazník patří mezi hlavní články vývojového procesu a agilní přístup mu poskytuje jasný pohled na celý průběh vývoje, což mu umožňuje řídit a optimalizovat jeho požadavky. (Waters, 2012)

Agilní vývoj přináší výhodu v podobě redukce rizik, která se týká harmonogramů nebo rozpočtů. Vývoj projektu je ovlivněn a určen na základě zpětné vazby zákazníka na poslední verzi, která mu byla poskytnuta. U tradičních metodik je zákazníkovi představena na konci vývoje finální verze, následně probíhá testování a v případě chyb probíhá jejich oprava. V takové situaci je nejen překročen harmonogram, ale dochází i k růstu nákladů. Při použití metodiky Scrum je vývoj rozdělen do několika sprintů a během každého z nich dochází k vyčíslení nákladů, což

umožňuje lépe a přesněji odhadnout konečné náklady na vývoj projektu. (Sutherland, 2007)

#### 4.6.1 SCRUM Development Process

Metodika Scrum je v dnešní době jednou z nejpoblárnějších a nejmladších agilních metodik a vychází z inkrementálního způsobu vývoje. Autoři Ken Schwaber a Mike Beedle ji poprvé prezentovali v roce 1995. Její název Scrum pochází z názvu ragbyové strategie jak dostat míč do hry. Metodika Scrum nedefinuje přesné technologie, postupy či nástroje, které by měly být použity, ale je zaměřena na řízení vývojového týmu, komunikaci a spolupráci. Stejně jako ostatní agilní metodiky, i Scrum vychází z Manifestu agilního vývoje. (Scrum Guide, 2014)

Pro pochopení základních principů a podstaty metodiky Scrum jsou v následující části popsány role, artefakty a události odehrávající se během realizace projektu. Všechny používané názvy jsou z důvodu nejednoznačného překladu a zavedené zvyklosti užívání v anglickém jazyce.

##### Role

Scrum definuje ve vývojovém týmu tři klíčové role, které mají specifické pravomoci a povinnosti.

- *Scrum Master* – jeho úkolem je dohled nad každým sprintem a celým vývojovým týmem. Dále má na starost dodržování obecných principů metodiky Scrum a jeho efektivitu. Náplní jeho práce je mimo jiné také motivace vývojového týmu a řešení případných problémů. Scrum master je součástí vývojového týmu a měl by s ním být v úzkém kontaktu.
- *Product Owner* – vlastník produktu, který zastupuje zájmy všech zainteresovaných osob a odpovídá za celý průběh a strukturu projektu. Jeho úkolem je sledovat zájmy zákazníka a dohlížet nad dodržováním jeho požadavků. Product Owner je zodpovědný za veškeré požadavky a vytvoření jejich seznamu tzv. Product Backlog. Jeho náplní práce je také seřazení požadavků dle jejich priority a zařazení do další iterace. Product Owner je také odpovědný za výběr členů vývojového týmu do každého sprintu.
- *Vývojový tým* – skupina vývojářů odpovídající za vývoj veškeré funkcionality projektu. Scrum tým je samořízený, organizovaný a zodpovědný za plnění všech úkolů v daném sprintu. Samořízený a samoorganizovaný v tomto pojetí znamená, že všichni členové vývojového týmu si sami vybírají a přiřazují úkoly, které jsou schopni v daném časovém úseku splnit. Jejich úkolem je také odhad pracnosti a času jednotlivých úkolů.

##### Artefakty

- *Product Backlog* – seznam veškerých úkolů, které je potřeba provést ke splnění požadované funkcionality. Každá funkcionality je popsána z pohledu zákaz-



níka, nikoliv z pohledu vývojáře či testera. Součástí položek v Product Backlog je také jejich náročnost, která je odhadnuta vývojovým týmem. Za tvorbu tohoto seznamu zodpovídá Product Owner a ten jednotlivé úkoly také řadí dle priority. Ze seznamu Product Backlog je následně vybrána množina úkolů, která se přesouvá do skupiny tzv. Sprint Backlog.

- *Sprint Backlog* - vybraná množina úkolů ze seřazeného Product Backlogu, která se stává náplní aktuálního sprintu. Tuto skupinu úkolů vybírá vývojový tým na začátku každého sprintu.
- *User Story* – funkcionalita, která je jednoznačně popsána z pohledu uživatele a má přesně definovaná akceptační kritéria, podle kterých se pozná, že je daná User Story dokončena. Jednotlivá akceptační kritéria jsou předem definována týmem a vlastníkem produktu při plánování Sprintu. Správně definovaná User Story obsahuje odpovědi na otázky kdo, co a proč. Jednotlivé User Story jsou pak obsaženy v Product Backlog.
- *Ohodnocení* – jednotlivé User Stories jsou na základě jejich náročnosti, a to včetně testování, revize, tvorby dokumentace, ohodnocena. Ohodnocení jsou vůči sobě relativní a lze tak porovnávat, který úkol je náročnější. Při hodnocení vždy vlastník produktu danou User Story popíše a každý člen týmu daný úkol ohodnotí. Způsobů jak provádět hodnocení je několik, ale vždy na počátku hodnocení je důležité si stanovit hodnotící škálu a podle té se následně řídit. Pro hodnocení se například používá řada čísel od 1 do 10 nebo velikosti oblečení XS, S, M, L, XL apod. Hodně užívanou metodou je tzv. Planning Poker, kde má každý člen týmu k dispozici hrací karty, pomocí nichž přiřazuje bodové ohodnocení. Každý zvolí příslušnou kartu a ve stejný okamžik se všechny otočí číslem nahoru. Následně je vedena diskuze s těmi členy týmu, kteří zvolili největší a naopak nejmenší hodnocení a musí svá rozhodnutí náležitě zdůvodnit. Poté se provádí opakované hlasování. Při tvorbě hodnocení se v žádném případě nedoporučuje provádět průměry. Je nutné, aby se na hodnocení shodl celý tým.
- *Epic* – nadřazená User Story, někdy také nazývaná Super User Story, spojuje několik User Story dohromady. Při tvorbě Product Backlog je Super User Story postupně rozložena na jednotlivé dílčí funkcionality a jim náležící User Story.
- *Burndown graf* – graf zobrazující stav celého projektu, umožňující predikci průběhu projektu.

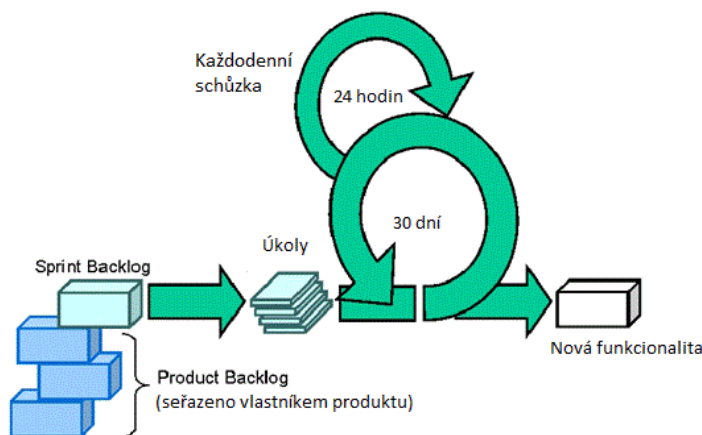
## Události

- *Sprint* - předem definované období, během kterého vývojový tým plní přidělené úkoly. Délka sprintu se liší na základě typu projektu. Obvykle sprint trvá jeden týden až měsíc. Celý vývoj projektu se pak skládá přibližně ze tří až osmi sprintů.
- *Sprint Planning* - před zahájením každého sprintu probíhá plánovací schůzka, kde jsou vybrány požadavky a úkoly, které budou součástí daného sprintu. Na

začátku schůzky vlastník produktu představí všechny User Stories a tým z nich vybírá ty, které zařadí do seznamu Sprint Backlog.

- *Daily Scrum* – této krátké schůzky konané na počátku dne, se účastní celý tým a její náplní je stručné zhodnocení práce předešlého dne a plánu na dnešní den. Každý člen vývojového týmu zodpoví otázky: jaké úkoly dokončil od poslední schůzky, jaké úkoly má v plánu na dnešní den a případně jaké má překážky při plnění svých úkolů. Schůzka by neměla překračovat délku 15 minut. Výhodou těchto krátkých a častých schůzek je neustálá informovanost o průběhu realizace. Účelem této schůzky je synchronizovat práci všech členů týmu a lépe zorganizovat jejich přidělenou práci.
- *Sprint Review* - po skončení sprintu se koná další schůzka tzv. Sprint Review, jejíž náplní je zhodnocení předešlého sprintu a požadavků, které se implementovaly a které se naopak splnit nepodařilo. U této schůzky jsou zúčastněny veškeré zainteresované osoby projektu, včetně zákazníka a je zde prezentována všechna funkčnost a požadavky, kterých bylo během sprintu dosaženo. Jednotlivé úkoly prezentují přímo členové týmu a představují se pouze dokončené úkoly, které splnily akceptační kritéria.
- *Sprint Retrospective* – nástroj pro získání zpětné vazby. Během této schůzky každý člen týmu vyjádří spokojenost či naopak nedostatky spojené s předešlým sprintem. Výsledkem retrospektivy jsou návrhy na zlepšení a změny ovlivňující další Sprint. (Schwaber, 2004)

Následující obrázek znázorňuje podstatu metodiky Scrum a propojení jednotlivých prvků.



Obr. 4 Schéma průběhu  
Zdroj: Přeloženo z Murphy, 2004

## 4.7 Nasazení metodiky Scrum

Zavedení agilních metodik je náročný a dlouhodobý proces. Jejich použití s sebou nese několik kroků, které je potřeba splnit. Hlavním předpokladem je detailní seznámení vývojového týmu s podnikovými cíli a prostředím. Následující část detailněji popisuje celkový proces metodiky Scrum.

Prvním krokem při zavedení metodiky Scrum je zvolení role Product Owner. Tato osoba je zodpovědná za práci na projektu, stanovení důležitosti a následně pořadí úkolů, které bude vývojový tým implementovat. Pozici Product Owner vykonává osoba, která detailně zná veškeré požadavky na realizaci projektu a působí jako hlavní komunikátor mezi zákazníkem a vývojovým týmem. Dalším důležitým krokem je zvolení Scrum Mastera, který je zodpovědný za vedení a podporu vývojového týmu.

V tuto chvíli jsou zvoleny veškeré role a jejich povinnosti v rámci realizace projektu a na řadě je vytvoření seznamu všech požadavků, tzv. Product Backlog. Tento seznam obsahuje jak funkční, tak i nefunkční požadavky. Po sestavení seznamu požadavků jsou jednotlivé úkoly seřazeny podle jejich priority.

Celková velikost všech úkolů je určena jejich ohodnocením. Toto hodnocení je důležité pro určování priorit a velikosti jednotlivých činností. Hodnocení se provádí pomocí přidělování bodů, ne však podle spotřeby času. Ve většině případů se používá bodový systém, například Fibonacciho posloupnost. Při využití Fibonacciho posloupnosti se přidělují jednotlivé body úkolům v závislosti na porovnání s ostatními úkoly. Na tomto hodnocení se podílí celý vývojový tým.

Po přiřazení bodů jednotlivým úkolům následuje schůzka, jejíž náplní je naplánování následujícího sprintu. Této schůzky se musí účastnit celý tým, včetně testerů, analytiků, vývojářů a vlastníka produktu. Hlavním krokem je určení délky sprintu a jeho cíle. Doporučovaná délka sprintu je od jednoho týdne až po celý měsíc.

Pro splnění cíle daného sprintu jsou vybrány požadavky ze seznamu Product Backlog, které jsou poté detailně popsány v podobě User Story a vývojový tým přidělí všem úkolům potřebný čas pro jejich implementaci. V průběhu každého Sprintu pracují členové vývojového týmu na přidělených úkolech. Výhodou však je týmová samo organizace, která umožňuje členům týmu vykonávat jejich rozhodnutí a přidělovat si úkoly. Každý den v průběhu celého sprintu se vývojový tým sejde a diskutuje o provedené práci. Úkolem všech členů týmu je představit, jaké úkoly se včera podařilo dokončit, jaké jsou v plánu dnes a případně diskutovat problémy spojené s jejich vývojem. Cílem tohoto krátkého setkání je rychle předat informace ostatním členům týmu. Po dokončení úkolů v daném Sprintu nastává čas na jeho zhodnocení, tzv. Sprint Review. Smyslem této schůzky je představit funkčnost, která byla v předchozím sprintu zhotovena. Následuje další schůzka Sprint Retrospective, jejímž účelem je získání zpětné vazby na dosud odvedenou práci. Klíčovým prvkem je diskuze celého týmu nad předešlým sprintem a jeho zhodnocení.

Tento celý proces se opakuje několikrát během celého Scrumu. Obvykle trvá přibližně tři až čtyři Sprints, než se tým synchronizuje a proces vývoje se optimalizuje. (Waters, 2007)

## 5 Použité technologie a nástroje pro řešení

### 5.1 BPMN

Business Process Modeling Notation je grafická notace, která slouží k modelování podnikových procesů pomocí diagramů. Tato notace byla vyvinuta institutem BPMI (Business Process Management Institute), který se v roce 2004 sloučil s organizací OMG (Object Management Group). V současnosti existuje verze BPMN 2.0. Diagramy podnikových procesů jsou složeny z několika prvků, které se sdružují do čtyř základních skupin, které jsou blíže rozepsány v následující části. (Object Management Group, 2016)

#### Tokové objekty

- Událost – událost je znázorněna kruhem a reprezentuje situaci, která má přímý vliv na průběh procesu. Události se dělí na počáteční, průběžné a konečné.
- Aktivita – představuje činnost, ke které v průběhu procesu dochází. Je zobrazena jako obdélník se zaoblenými rohy. Aktivitu lze dále dekomponovat na subprocesy.
- Brána – brána se zobrazuje kosočtvercem a umožňuje větvení či naopak sloučení toků uvnitř procesu. (Dumas, 2013)

#### Spojovací objekty

- Sekvenční tok – znázorňuje posloupnost provádění daných činností v procesu.
- Tok zpráv – využívá se pro znázornění komunikace mezi objekty procesu.
- Asociace – asociční tok se používá pro připojení anotací či datových objektů a skladů. (Dumas, 2013)

#### Organizační prvky

- Bazén – shromažďuje zdrojové prvky procesu. Obecně se využívá pro znázornění celé společnosti. Bazén je dále rozdělen na několik plaveckých drah.
- Plavecká dráha – zdroj uvnitř bazénu. Typicky vyjadřuje nějaké oddělení uvnitř společnosti, interní role či užívané softwarové systémy. (Dumas, 2013)

#### Artefakty

- Datové objekty – datový objekt představuje shromážděná data využívaná jako potřebný vstup či vygenerovaný výstup aktivity.
- Datový sklad – místo, které zahrnuje datové objekty. (Dumas, 2013)

## 5.2 IFML

Jazyk IFML (Interaction Flow Modeling Language) byl vytvořen pro vyjádření chování a interakce mezi uživatelem a softwarovou aplikací. Vznik IFML byl inspirován dlouholetou zkušeností s modelovacím jazykem WebML (Web Modeling Language) a nástrojem WebRatio. Jazyk IFML byl přijat jako standard organizací OMG (Object Management Group) v roce 2013. O rok později byla organizací vydána první verze IFML 1.0 a zveřejněna byla během února roku 2015. IFML zobecňuje standard WebML a postupně se tak stává jeho náhradou. (IFML, 2016)

Jazyk IFML je určen pro vyjádření obsahu, interakce a kontrole chování front-end aplikací. IFML však nepokrývá modelování vzhledu a stylu aplikace. Proto jazyk není vhodný pro vývoj aplikací s vysoce interaktivním uživatelským rozhraním, multidimenzionální grafikou či vývoj videoher. Je zaměřen především na podnikově orientované aplikace. Tyto aplikace spadají do následujících oblastí:

- tradiční HTML webové aplikace,
- moderní webové aplikace, využívající standardy HTML5, Ajax, Flash apod.,
- mobilní aplikace,
- klient-server aplikace,
- desktopové aplikace.

Notace jazyka IFML se příliš nemění od notace standardu WebML. K odlišnostem došlo zejména v definici pojmu událostí, uživatelských událostí a jejich spouštění. (Brambilla, Butti, 2013)

Předpokladem pro modelování a vývoj aplikace je sběr a analýza požadavků. Tento krok zahrnuje shromáždění veškerých informací o očekávané funkčnosti aplikace. Požadavek je definován jako specifikace, co by mělo být implementováno. Rozlišují se dva typy požadavků, a to funkční a nefunkční. Funkční požadavky zahrnují očekávané chování systému a nefunkční pak specifikují vlastnosti nebo omezující podmínky systému. (Arlow, 2007)

Na základě definovaných požadavků je potřeba vytvořit pomocí notace IFML pět modelů:

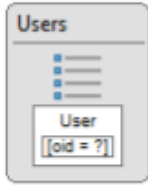

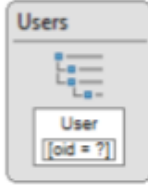
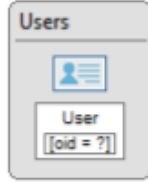

- Doménový model – neboli strukturální model, reprezentující datovou strukturu a organizaci dat. Wazlawick (2013) zmiňuje, že pro modelování datové struktury může být také použit libovolný prostředek pro tvorbu datových modelů, entitně relační diagram nebo diagram tříd notace UML.
- Odvozený model – tento model vzniká odvozením dat z doménového modelu. Data obsažena v odvozeném modelu lze chápat jako derivaci atributů a asociací dat z doménového modelu.
- Navigační model – tento model popisuje navigaci a vzájemné propojení jednotlivých stránek. Pomocí navigačního modelu je zobrazen průchod uživatele daným systémem.


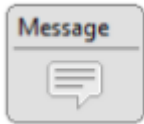

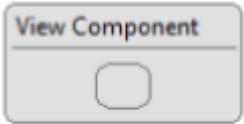
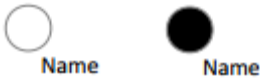


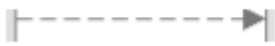
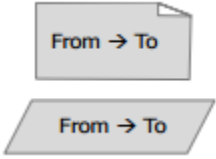

- Kompoziční model – definuje strukturu stránky a logické rozložení jednotlivých prvků na stránce.
- Prezentační model – tento model zachycuje zobrazení všech komponent modelované domény a zobrazuje vzhled a pozici jednotlivých prvků uživatelského rozhraní. (Wazlawick, 2013)

### Základní prvky IFML

Mezi základní prvky IFML patří kontejnery a komponenty. Vytvořené diagramy se skládají z kontejnerů, které reprezentují webové stránky a formuláře. Kontejnery je možné dále dekomponovat a modelovat tak vnořené stránky či formuláře. Mezi základní prvky IFML patří:

Tab. 1 Přehled elementů IFML

Název	Popis	Značení
List	Prvek zobrazující seznam instancí objektu.	
Checkable List	Seznam umožňující vícenásobný výběr.	
Hierarchy	Prvek zobrazující hierarchicky uspořádané instance objektu.	
Details	Prvek sloužící k zobrazení detailních informací o instanci.	
Form	Základní prvek představující formulář pro zadávání dat.	

Calendar	Prvek umožňující výběr data.	
Message	Prvek umožňující zasílání zpráv.	
View Container	Prvek rozhraní obsahující prvky, které zobrazují obsah a představují interakci s dalšími View Containers.	
View Component	Prvek rozhraní, který zobrazuje obsah nebo přijímá vstup.	
Event	Uživatelská nebo systémová událost, která má vliv na stav aplikace.	
Action	Část obchodní logiky spuštěna nějakou událostí.	
Navigation Flow	Představuje vstupně-výstupní vztah. Zdroj relace má výstup, který je spojen se vstupem připojeného objektu.	
Data Flow	Znázorňuje tok dat mezi View Components nebo jako důsledek uživatelské akce.	
Parameter Binding	Vymezuje spojení vstupního parametru zdroje s výstupním parametrem cíle.	
Scroller	Prvek sloužící pro pohyb mezi instancemi objektu.	

Zdroj: IFML, 2016

### 5.3 Case nástroj WebRatio

Case nástroj WebRatio vznikl v roce 2001 na italské univerzitě Politecnico di Milano. Tento nástroj podporuje notaci The Interaction Flow Modeling Language (IF-ML), která slouží jako vizuální modelovací prostředek webových aplikací. WebRatio také podporuje standard Business Process Model and Notation (BPMN), který se používá k zobrazení podnikových procesů. WebRatio na základě vytvořených modelů následně umožňuje generování kódu, který lze použít pro tvorbu webových a mobilních aplikací, které jsou pak kompatibilní s HTML5, CSS3 a Java. Tato vygenerovaná aplikace je založena na modelu Model-View-Controller. (WebRatio, 2016)

### 5.4 PHP

PHP je skriptovací jazyk, který slouží k tvorbě dynamických webových aplikací. PHP pracuje na straně webového serveru, kde je aplikace uložena a komunikuje s databázemi, jako je MySQL, PostgreSQL, Oracle a další. Jazyk PHP je velmi rozšířený prostředek pro tvorbu webových aplikací a v současné době je k němu dostupná rozsáhlá část literatury a dokumentace. PHP kód je možné vkládat do HTML stránek, což při tvorbě webu umožňuje kombinaci více technologií (Procházka, 2012).

### 5.5 Apache

Apache slouží jako webový server, jehož hlavní úlohou je vykonání požadavků, které přijímá prostřednictvím webového prohlížeče. Po zpracování požadavků následně zobrazuje výsledky. (Naramore, 2006)

### 5.6 MySQL

MySQL je relační databázový server, který je nejčastěji využíván v kombinaci se skripty napsanými v jazyce PHP. Systém MySQL vznikl ve Švédsku v roce 1996 pod firmou MySQL AB. MySQL není jediný na trhu dostupný databázový systém. Mezi další známé systémy patří např. Microsoft SQL Server, Oracle, Firebird. Mezi klíčové vlastnosti databázového systému MySQL se řadí multiplatformní využití, velký důraz na rychlost, ukládání dotazů do paměti, bezpečnost, open source licence, aktivní komunita uživatelů (Gilmore, 2007).

### 5.7 Nette Framework

Nette Framework je open source PHP Framework, poskytuje vlastní šablonovací systém Latte, podporu HTML5 a klade důraz na bezpečnost aplikace. Tento Framework umožňuje oddělení aplikační logiky a samotného zobrazení uživateli, a to



pomocí návrhového vzoru Model-View-Controller. Funkci Controlleru v Nette zastává Presenter a struktura aplikace je rozdělena do tří částí:

- Model – model je základem celé aplikace a obsahuje veškerou funkční logiku.
- View – pohled je vrstva aplikace, která má na starost zobrazení výsledků uživateli, které získává z modelu. K tomu využívá svůj šablonovací systém Latte.
- Presenter – úkolem této vrstvy je zpracování požadavků od uživatele a následné vygenerování odpovědi.

Toto oddělení jednak zpřehledňuje kód, ale také usnadňuje vývoj aplikace a její testování. (Nette Foundation, 2016)

## 6 Vlastní řešení

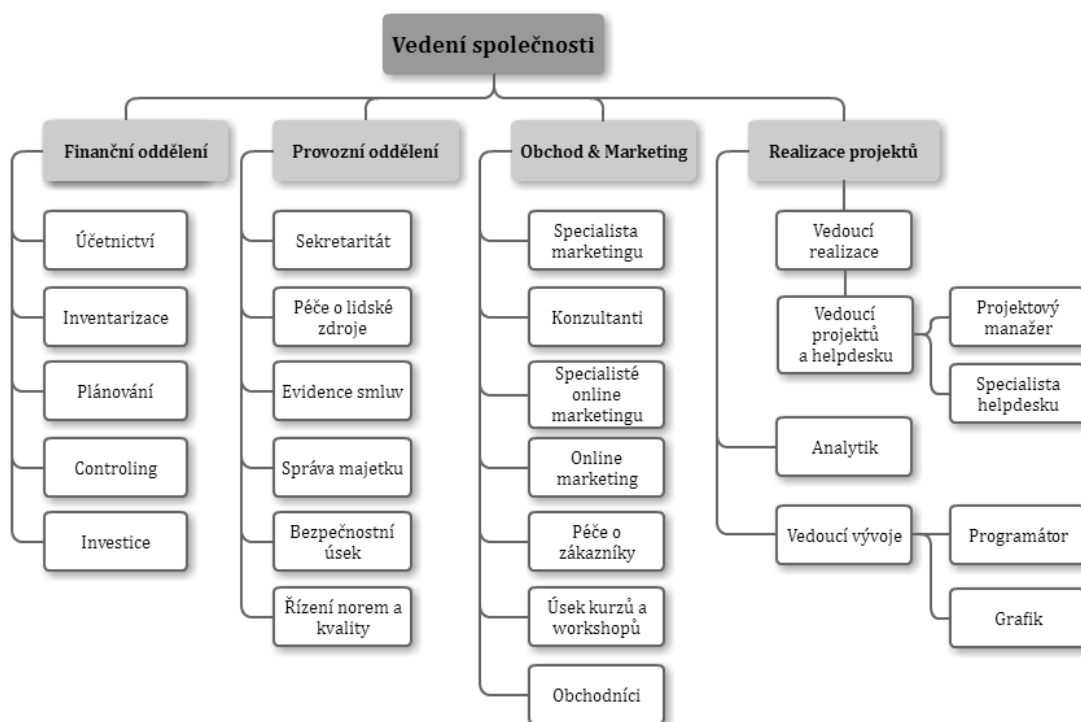
### 6.1 Charakteristika společnosti

Cíl práce je směřován do společnosti, která vznikla v roce 2001 a působí v Brně a v Praze. Zabývá se produkcí e-shopových řešení, ale i dalšími službami podporující obchodování na internetu. V současné době dodala na trh již několik stovek e-shopů, jiných e-commerce řešení a vydává několik populárních magazínů. Od roku 2009 se firma také zaměřila na školení a kurzy pro začínající podnikatele v oblasti internetového obchodování a SEO optimalizace. V následujících letech firma získala certifikát kvality ISO 9001, který stanovuje požadavky na systém řízení kvality. Významnou podporu úspěchu internetového obchodování zajišťuje také oddělení marketingu se službami v oblasti online marketingu, SEO, poradenství, online reklamy a dalšími aktivitami. Po předání spuštěného e-shopu klientovi, společnost rovněž poskytuje technickou podporu a rozšiřování funkcionality e-shopu.

#### 6.1.1 Struktura organizace

Společnost má v současné době přibližně 40 zaměstnanců a tak se řadí mezi středně velké společnosti. Na obr. 5 je zobrazena organizační struktura celé společnosti. Ve vedení je ředitel, pod kterého spadají další úseky. Společnost je organizačně rozdělena do čtyř hlavních částí, a to finanční oddělení, provozní oddělení, marketing a realizace. Z toho každé oddělení má svého ředitele. Mezi dílčí činnosti finančního oddělení patří účetnictví, inventarizace majetku, plánování, controlling a investice. Součástí provozního oddělení je sekretariát, péče o lidské zdroje, evidence smluv, správa majetku, bezpečnostní úsek, řízení kvality a norem. Dalším úsekem společnosti je obchod a marketing, který má na starost především online marketing a péči o zákazníky. V tomto úseku pracuje několik specialistů marketingu a jejich konzultanti. Mezi další činnosti oddělení marketingu patří organizace kurzů a školení pro internetové obchodování a činnosti s tím spojené.

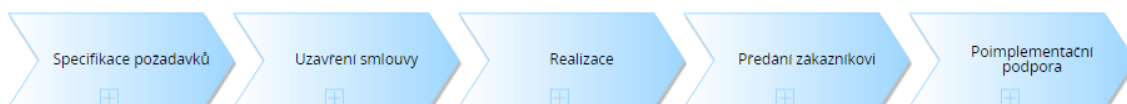
Pro účely této práce je klíčové oddělení realizace. V čele realizace stojí technický ředitel, který má na starost celé oddělení. Významnou roli zde hraje vedoucí projektů a helpdesku, mezi jehož hlavní odpovědnost patří řízení a dohled nad průběhem realizace projektů. Mezi jeho přímé podřízené patří projektoví manažeři a specialisté helpdesku, kteří zajišťují technickou podporu zákazníků. Součástí oddělení realizace je také analytik, který má na starost analýzy technické proveditelnosti projektů. Důležitou součástí realizace je vývojové oddělení, které zajišťuje samotnou implementaci projektů. V čele vývojového oddělení stojí vedoucí vývoje, který má na starost vývojový tým.



Obr. 5 Struktura organizace

## 6.2 Analýza procesu řízení projektů

Celý proces projektového řízení je hierarchicky rozdělen do pěti dílčích subprocesů. Struktura procesu je znázorněna na obr. 6. Jednotlivé subprocesy jsou blíže popsány a zobrazeny v následující podkapitole.

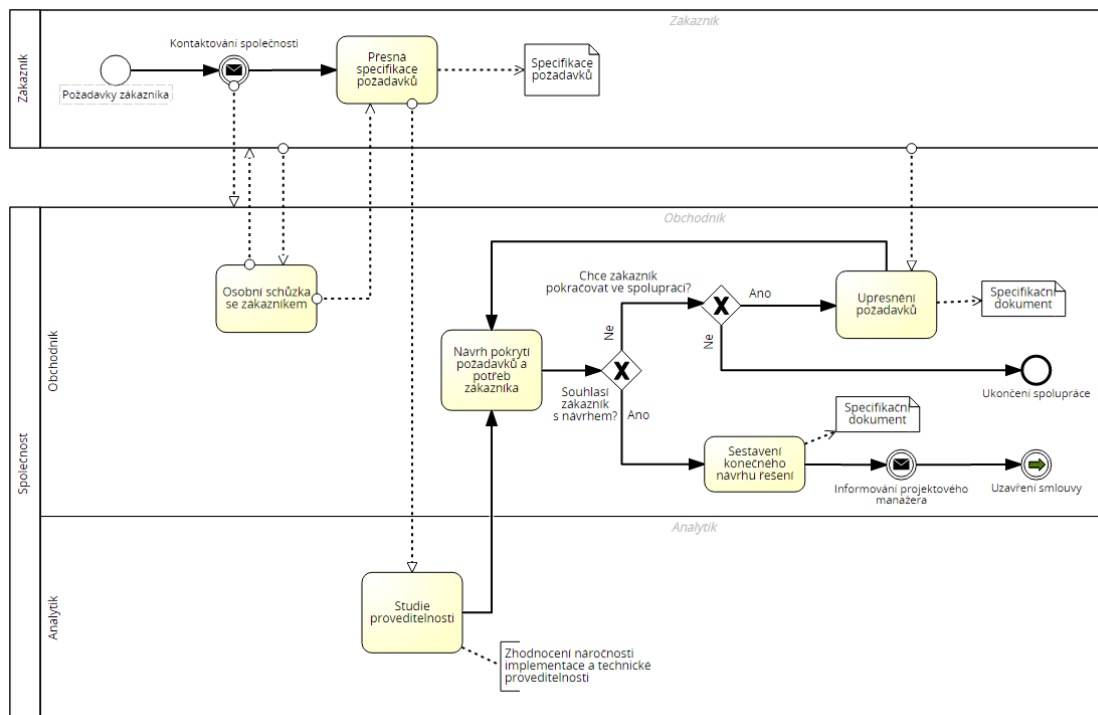


Obr. 6 Struktura procesu projektového řízení

### 6.2.1 Subproces Specifikace požadavků

Na následujícím obr. 7 je zobrazen subproces, jehož cílem je specifikace požadavků. Proces začíná oslovením potenciálního zákazníka. Ve většině případů dochází ke kontaktování společnosti ze strany zákazníka, ale výjimkou není ani oslovení potenciálního zákazníka ze strany společnosti. Na základě osobní schůzky obchodníka s klientem dochází k představení produktu a následně k přesné specifikaci požadavků na výsledný produkt. Po bližší specifikaci požadavků provádí systémový analytik studii proveditelnosti a zhodnocení náročnosti implementace. V souladu s touto studií je následně vytvořen návrh pokrytí požadavků a potřeb

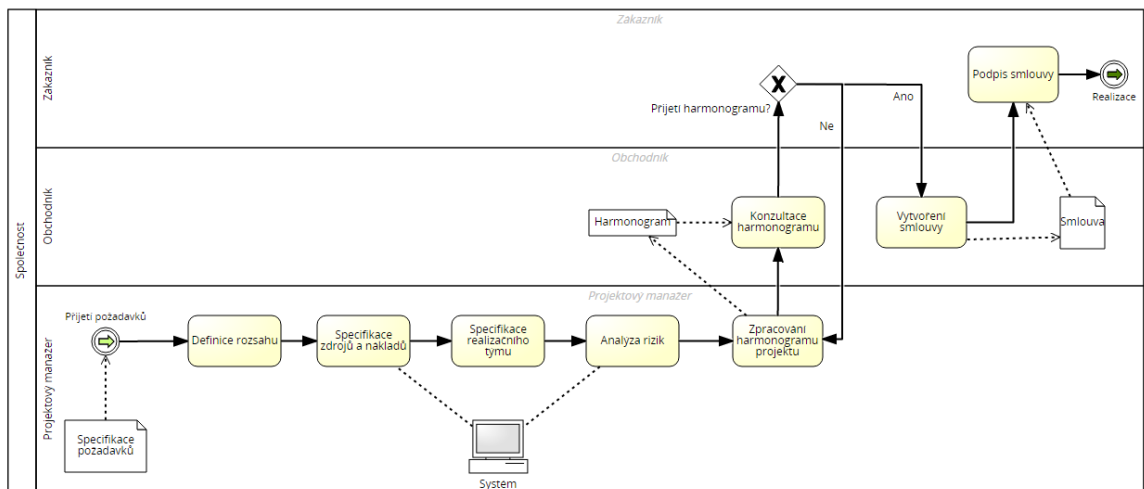
zákazníka. Pokud zákazník s návrhem souhlasí, tak je vytvořen konečný návrh řešení. V případě, že má zákazník k návrhu nějaké připomínky, dochází k upřesnění požadavků a nalezení vhodného řešení splňujícího jeho požadavky. Výsledkem této fáze je specifikační dokument, který obsahuje podrobnou definici veškerých požadavků a je dále předán jako podklad k uzavření smlouvy.



Obr. 7 Subproces Specifikace požadavků

## 6.2.2 Subproces Uzavření smlouvy

Cílem tohoto procesu (obr. 8) je uzavření smlouvy a předání návrhu projektu k realizaci. Na základě detailní specifikace požadavků od zákazníka a návrhu možné realizace řešení je projektovým manažerem vytvořena definice rozsahu. Následně jsou specifikovány potřebné zdroje a náklady a je provedena analýza rizik, která mohou v průběhu zpracování projektu nastat. V neposlední řadě projektový manažer určí realizační tým, kterému jsou v další etapě přiřazeny jednotlivé úkoly. Důležitým výstupem tohoto procesu je harmonogram projektu. Za tvorbu harmonogramu je odpovědný projektový manažer a spolu s obchodníkem a potenciálním zákazníkem ho důkladně konzultuje. Vytvořený harmonogram obsahuje bližší informace o termínech zahájení a předpokládaného dokončení projektu. V případě, že je zákazníkem harmonogram akceptován, dochází k vytvoření a podpisu smlouvy. Pokud zákazník s předloženým harmonogramem není spokojen, dochází k dalším úpravám. Výsledkem tohoto procesu je tedy uzavřená smlouva mezi zákazníkem a společností a veškeré dokumenty specifikující projekt, které dále slouží jako podklady pro samotnou realizaci projektu a budoucí spolupráci.

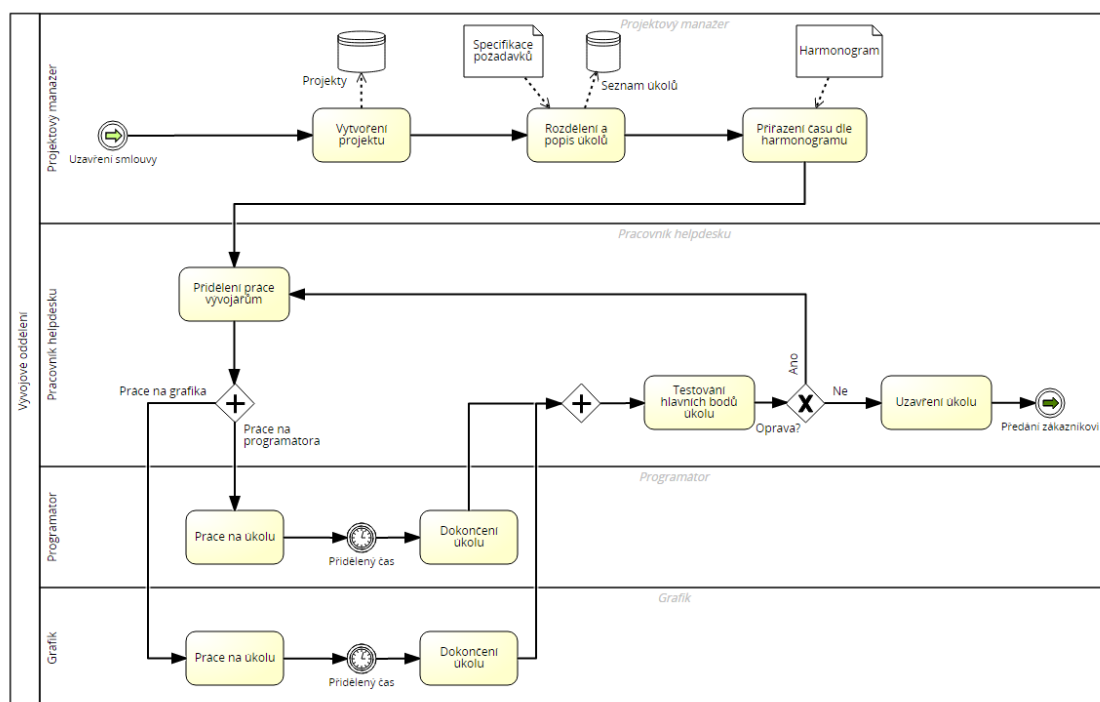


Obr. 8 Subproces Uzavření smlouvy

### 6.2.3 Subproces Realizace

Na obr. 9 je zobrazen průběh realizace a práce na projektu. Po úspěšném uzavření smlouvy a zahájení spolupráce se zákazníkem, je projektovým manažerem zadán nový projekt do systému. Na základě dříve vytvořené specifikace požadavků dochází k rozložení projektu na jednotlivé úkoly a je tak vytvořen seznam úkolů, které je třeba k dokončení projektu splnit. Projektový manažer následně podle harmonogramu přiřadí jednotlivým úkolům čas, za který je potřeba je splnit. Poté jsou úkoly přiřazeny vývojovému týmu. Po dokončení úkolu pracovník helpdesku kontroluje funkčnost a v případě správnosti daný úkol uzavře a předá na kontrolu zákazníkovi. V opačném případě, kdy je potřeba práci opravit, je úkol znovu poslán vývojovému týmu.

Vývojové oddělení se skládá z projektového manažera, pracovníka helpdesku, programátorů a grafiků. Vývojový tým neobsahuje žádného testera, tudíž činnost testování spadá na pracovníka helpdesku, který komunikuje se zákazníkem a řeší případné reklamace a připomínky.

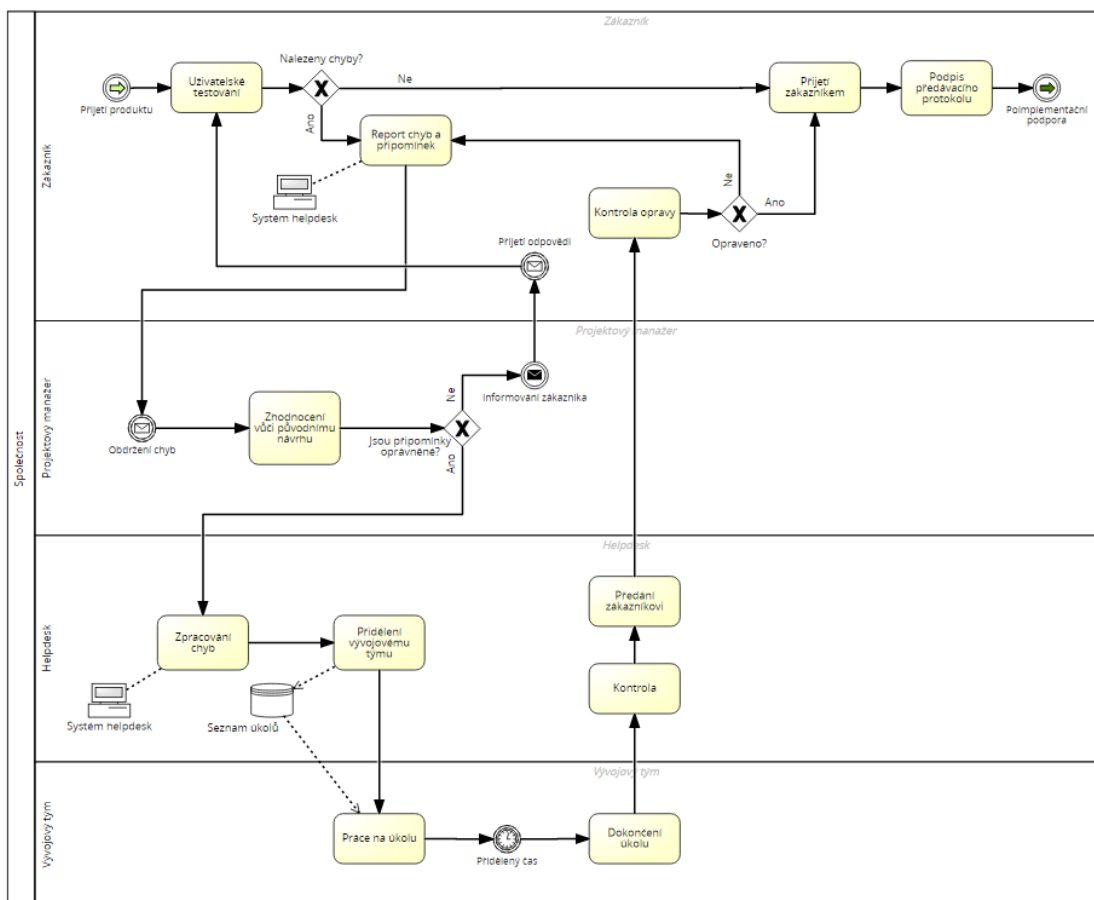


Obr. 9 Subproces Realizace

### 6.2.4 Subproces Předání zákazníkovi

Po dokončení úkolů následuje uživatelské testování ze strany klienta. Po předání produktu zákazník komunikuje s projektovým manažerem, který jeho případné připomínky zhodnocuje, zda jsou relevantní vůči původnímu schválenému návrhu řešení. V situaci, kdy jeho náměty k funkčnosti nejsou oprávněné, je zákazník informován a pokud z jeho strany nejsou další náměty k opravě, je projekt označen jako uzavřený a je podepsán předávací protokol. Pokud však zákazník objeví nějaké nesrovnalosti, které je potřeba opravit, jsou tyto chyby reportovány prostřednictvím systému helpdesk a následně předány vývojovému týmu k opravě.

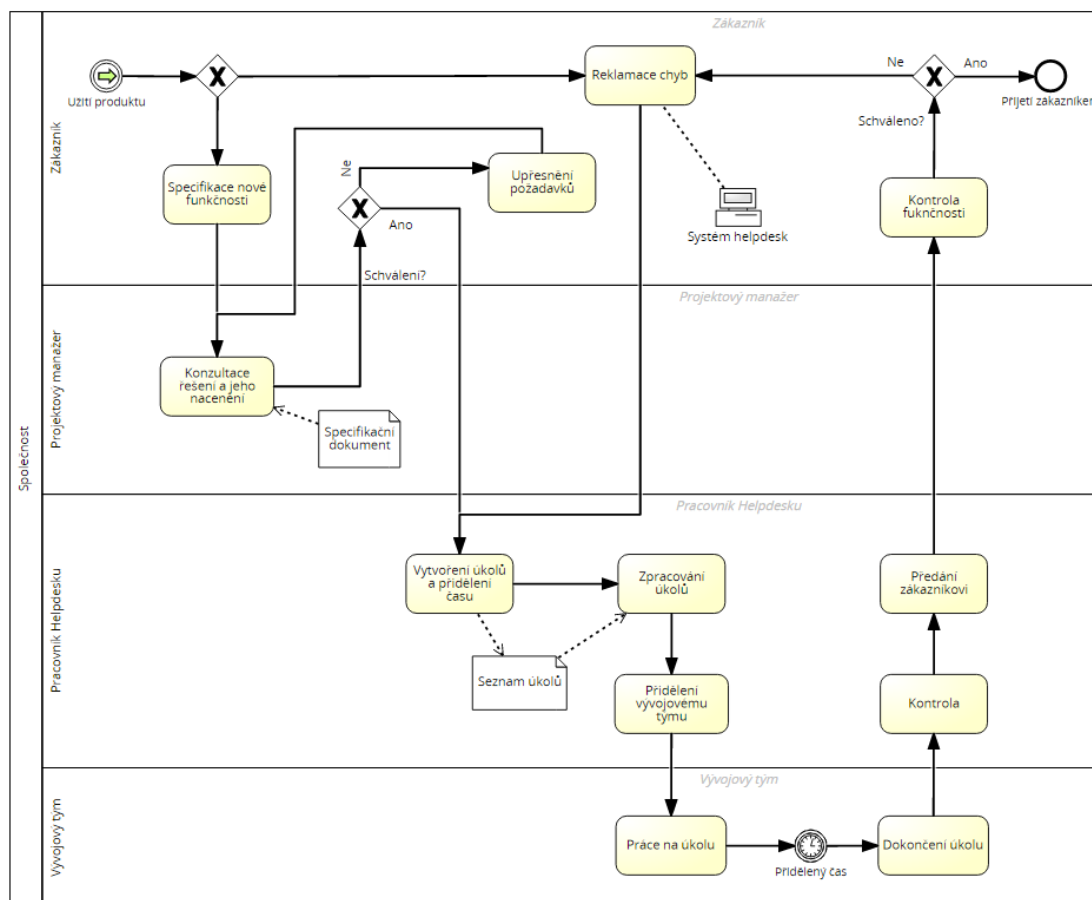
Cílem tohoto procesu (obr. 10) je tedy prvotní testování ze strany zákazníka a zjištění chyb. Výsledkem této fáze je předání hotového projektu. Po předání následuje technická podpora, která je poskytována všem zákazníkům, kterým byl implementován e-shop. Tato podpora je placená formou paušálního poplatku a spočívá v opravě nalezených chyb v průběhu používání e-shopu a případně doplnění nové funkčnosti do stávající implementace formou objednávek.



Obr. 10 Subproces Předání zákazníkovi

### 6.2.5 Subproces Poimplementační podpora

Po předání a schválení dokončeného projektu následuje tzv. poimplementační podpora. Součástí této fáze je reportování chyb ze strany zákazníka a vytváření objednávek na doplnění nových funkcionalit do stávajícího řešení. V případě požadavku na opravu chyb se zákazník prostřednictvím systému spojí s pracovníkem helpdesku, se kterým následně dané připomínky konzultuje. Pokud se jedná o požadavek na implementaci nové funkčnosti, tak proběhne konzultace nových požadavků s projektovým manažerem. Ten pak vytvoří cenovou nabídku na danou úpravu a pošle zpět zákazníkovi k jeho schválení. Pokud zákazník s návrhem souhlasí, jsou dané požadavky předány na vývojový tým a následně zpracovány. Po dokončení veškerých úprav, je výsledná funkcionalita předána zákazníkovi ke kontrole a schválení.



Obr. 11 Subproces Poimplementační podpora

### 6.3 Návrh inovačních opatření

Zavedení metodiky Scrum s sebou nese nejen změny v procesu řízení projektů, ale také nezbytné organizační změny. Společnost má v současné době 10 programátorů, z toho 8 zaměřených na programování samotného jádra a 2, kteří mají na starost uživatelské rozhraní a grafiku. Jak už bylo řečeno výše, ve vývojovém týmu není žádný tester a jeho práci zastává pracovník helpdesku, který po dokončení úkolu testuje základní funkčnost, která je náplní úkolu. Po dokončení celého projektu je výsledek předán zákazníkovi, který následně provádí testování a reportuje případné chyby a připomínky. Toto testování finální verze s sebou nese i několik nevýhod. Ve většině případů dochází nejen k překročení harmonogramu, ale také i ke zvýšení nákladů. To se také stalo i hlavním důvodem pro požadavek přechodu na metodiku Scrum. Ta totiž zajišťuje testování v průběhu celého životního cyklu projektu. Časté testování tak pomůže dříve odhalit případné chyby a rychleji je opravit.



## Organizační změny

Společnost zaměstnává 2 specialisty helpdesku, kteří komunikují se zákazníky, zajišťují technickou podporu a požadavky klientů dále předávají na vývojový tým. Pracovníci helpdesku přiřazují programátorům úkoly podle jejich vytíženosti. Obvykle se tedy stává, že se programátoři přesouvají z jednoho projektu na druhý. To z praktického hlediska může způsobit chaos nejen v organizaci práce, ale také méně efektivní a přehledný kód. I když je pevně daná syntaxe programovacího jazyka, každý programátor má jiné zvyklosti a používané standardy. Při hledání chyb v kódu a opravách stávající funkčnosti je žádoucí mít dobře strukturovaný a komentovaný kód. To může velmi usnadnit budoucí práci dalších programátorů. Proto je nutné, aby způsob programování měl jistou zavedenou konvenci, kterou budou všichni členové týmu dodržovat. Používání stanovených pravidel zajistí tzv. Code Review, jehož náplní bude kontrola kódu z hlediska způsobu zápisu podle stanovené konvence kódování. Protože každý projekt je unikátní a má dané zadání, je důležité být detailně seznámen s požadovanou funkčností. Ovšem častý přechod z jednoho projektu na druhý může zapříčinit nedostatečnou znalost dané domény.

Pro efektivní a optimální využívání agilní metodiky Scrum by bylo vhodné rozdělit vývojový tým na 2 týmy, z čehož každý bude mít na starost přidělené projekty. V každém týmu budou 4 programátoři a 1 grafik. Toto rozdělení s sebou mimo jiné přináší výhodu v možném soustředění programátorů na přiřazené projekty a detailnější znalost funkčnosti. Zpravidla tedy programátoři budou dostávat úkoly, se kterými už mají nějaké zkušenosti a danou problematiku dobře znají.

## Stanovení rolí

Pro zavedení agilní metodiky Scrum je potřeba určit i dané role a definovat události, které jsou pro Scrum nezbytné. Veškeré kroky byly důkladně konzultovány s vývojovým oddělením společnosti. V první řadě je důležité zvolit roli tzv. Scrum Master a Product Owner. Dle dokumentace se v některých případech doporučuje, aby vlastník produktu byl zákazník. Zákazník se pak dále účastní daných schůzek a je do realizace přímo zapojen. V případě implementace e-shopů, kde se realizuje více projektů současně, není možné, aby zákazník byl vlastníkem produktu. Po konzultaci s vývojovým oddělením bylo dohodnuto, že roli vlastníka produktu zastane současný pracovník helpdesku. Toto zvolení je především z toho důvodu, že je se zákazníkem během realizace nejvíce v kontaktu a konzultuje s ním dané požadavky a možnosti řešení. Vlastník produktu dále zastupuje zájmy všech zúčastněných stran. Jeho hlavní odpovědností je kontrola nad průběhem celého projektu. Při zahájení projektu vytváří seznam požadavků, které je potřeba splnit.

Další rolí, kterou bylo potřeba zvolit je tzv. Scrum master. Tuto roli zastane současný projektový manažer. Jeho úkolem je dohled nad každým sprintem a celým vývojovým týmem. Role pracovníků v rámci procesu realizace z hlediska metodiky Scrum jsou definovány. Co se týče zavedených názvů pracovních pozic, zůstávají dosud používané názvy (projektový manažer, pracovník helpdesku apod.).

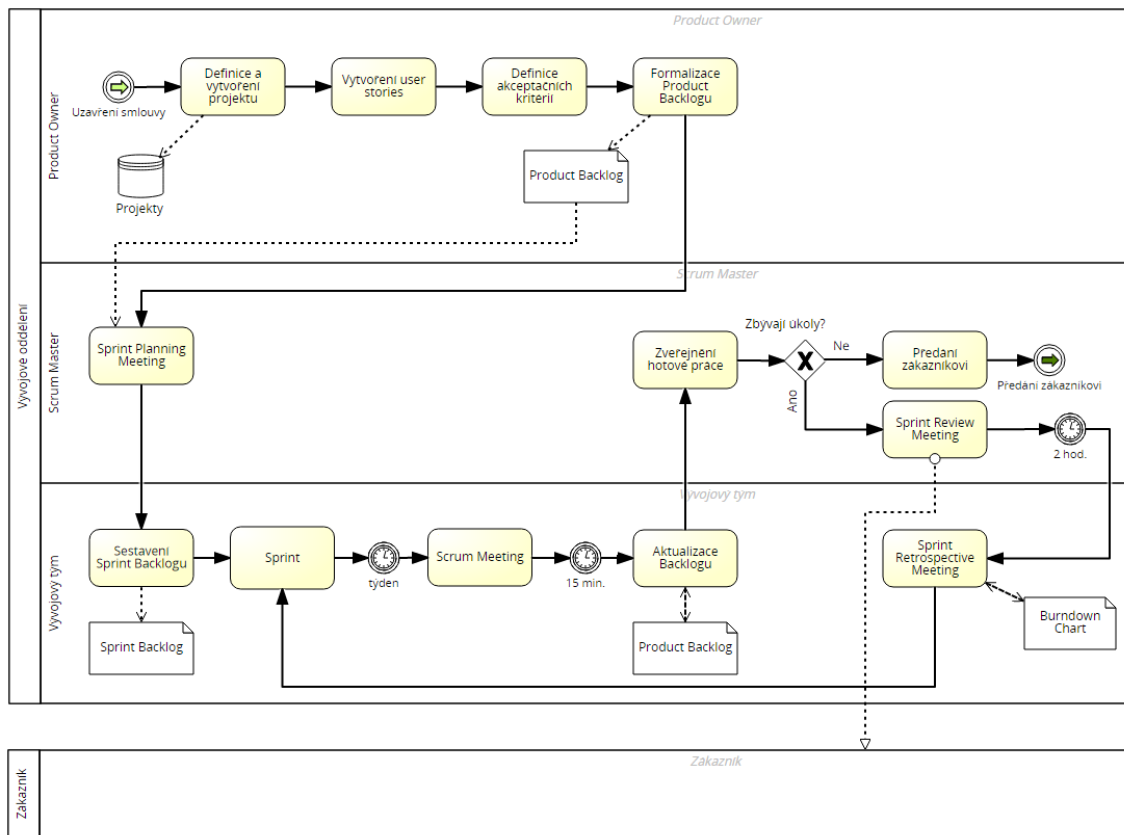
## Nové události

Z pohledu standardních událostí metodiky Scrum, je důležité zvolit délku jednotlivých sprintů, ze kterých se celý průběh vývoje skládá. Vzhledem k různorodosti a množství projektů byl zvolen týdenní sprint. Je to především i z důvodu implementační podpory, jejíž součástí jsou reklamace, které je nutné přednostně opravit a mohou tak zasáhnout do předem stanoveného Sprint Backlogu. Tato situace pak narušuje zavedený systém, kdy jsou požadavky dané v seznamu úkolů. Může tedy dojít k situaci, kdy se některé úkoly v souvislosti s reklamací odsunou. Těmto stavům je důležité se vyvarovat, a to zavedením určení priority úkolů.

Dále je potřeba zahájit každodenní schůzky, které je vhodné dodržovat ve stejný čas. Cílem těchto schůzek je rychlé zhodnocení práce předešlého dne a seznámení s náplní dnešního dne. Vzhledem k flexibilní pracovní době pracovníků, se budou tyto schůzky konat každý den v 10 hodin ráno. Dalším důležitým milníkem je schůzka zvaná Sprint Review, která se koná vždy po skončení jednoho sprintu. Náplní této schůzky je zhodnocení předešlého sprintu. Zde je důležité zmínit veškeré požadavky, které se implementovaly a které se splnit nepodařilo. Během této schůzky je zákazníkovi představena veškerá dokončená funkcionality. To přináší mnoho výhod, především ve včasném objevení případných chyb či nesrovnalostí a možnosti jejich rychlé nápravy. Vývojový tým se také účastní schůzky Sprint Retrospective, která slouží jako nástroj pro získání zpětné vazby daného sprintu. Jsou zde zmíněny nedostatky či připomínky k předešlému sprintu.

Veškeré navržené inovační kroky se týkají a přímo ovlivňují především proces realizace projektů a jsou zobrazeny v následující podkapitole a příslušném procesním diagramu (obr. 12).

### 6.3.1 Návrh inovace subprocessu Realizace



Obr. 12 Subproces Realizace

## 6.4 Současný stav IS/ICT

V současné době je ve společnosti využívána interní aplikace, která slouží jako podpůrný nástroj pro aktivity projektového řízení. Aplikace umožňuje správu projektů, úkolů a uživatelů. Pomocí konektoru je aplikace napojena na ekonomický informační systém, který následně na základě vykázaného času u jednotlivých úkolů vytváří navazující fakturaci. Webová aplikace je také napojena na helpdeskový systém, na základě kterého jsou následně vytvářeny jednotlivé úkoly. Do tohoto systému mají přístup jak zákazníci, tak i členové vývojového týmu.

Hlavním požadavkem společnosti je zavedení agilní metodiky Scrum. Tato metodika nejvíce ovlivňuje část realizace projektů. Interní webová aplikace však nepodporuje principy vývoje projektů s využitím metodiky Scrum. Z toho důvodu vznikl požadavek na vytvoření nové webové aplikace, která bude v souladu s novými požadavky a bude obsahovat funkčnost umožňující provádět jednotlivé aktivity metodiky Scrum. Pro agilní řízení projektů již existuje několik nástrojů, ty však není možné použít z důvodu žádosti využívání stávající interní aplikace. Jediným řešením se tedy jeví vývoj aplikace formou přídatného modulu, který bude

napojen na stávající aplikaci. Vytvořená aplikace bude sloužit jako podpora vývojového týmu a bude tak součástí procesu realizace projektů. V následující kapitole jsou blíže popsány funkční a nefunkční požadavky na tuto aplikaci.

## 6.5 Sběr a analýza požadavků

### Funkční požadavky

Tato část shrnuje požadovanou aplikační funkčnost z pohledu budoucích uživatelů. Analýza funkčních požadavků probíhala mezi členy vývojového týmu, následně byly pak jednotlivé požadavky shrnuty do formální podoby.

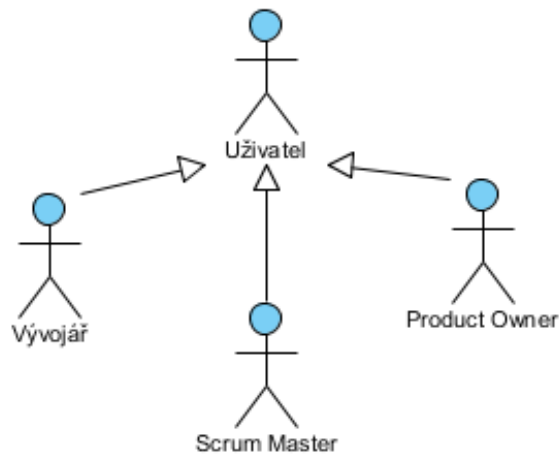
- Realizace aplikace formou přídatného modulu do stávající aplikace.
- Rozlišení uživatelů a jejich práv.
- Přihlašování do aplikace pomocí uživatelského jména a hesla.
- Evidence uživatelů.
- Správa a evidence úkolů.
- Přiřazování úkolů členům týmu.
- Podpora aktivit využívající metodiky Scrum.
- Zaznamenání času k úkolům.
- Přehledy vykázaného času a jednotlivých úkolů
- Export přehledů do formátů pdf a csv.

### Nefunkční požadavky

- Implementace formou webové aplikace s využitím objektově orientovaného jazyka PHP, který je použit i ve stávající aplikaci.
- Využití databázového systému MySQL.
- Zabezpečení aplikace proti neoprávněnému přístupu k interním datům.
- Podpora platformy Windows.
- Přehlednost aplikace a intuitivní uživatelské rozhraní.

## 6.6 Identifikace uživatelů a rolí

Navrhovaná aplikace bude sloužit především pro podporu realizace. To znamená, že ji budou využívat členové vývojového týmu, vlastník produktu (Product Owner) a vedoucí projektu (Scrum Master). Na obr. 13 je zobrazen diagram uživatelů, kteří budou aplikaci používat.

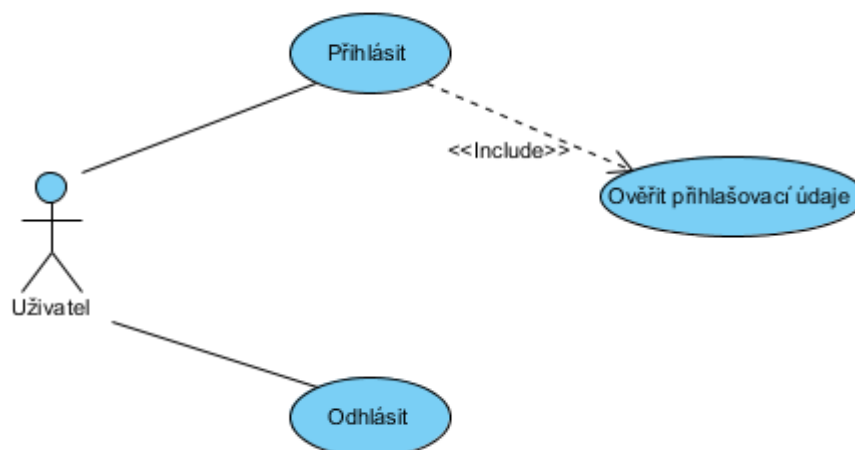


Obr. 13 Uživatelé

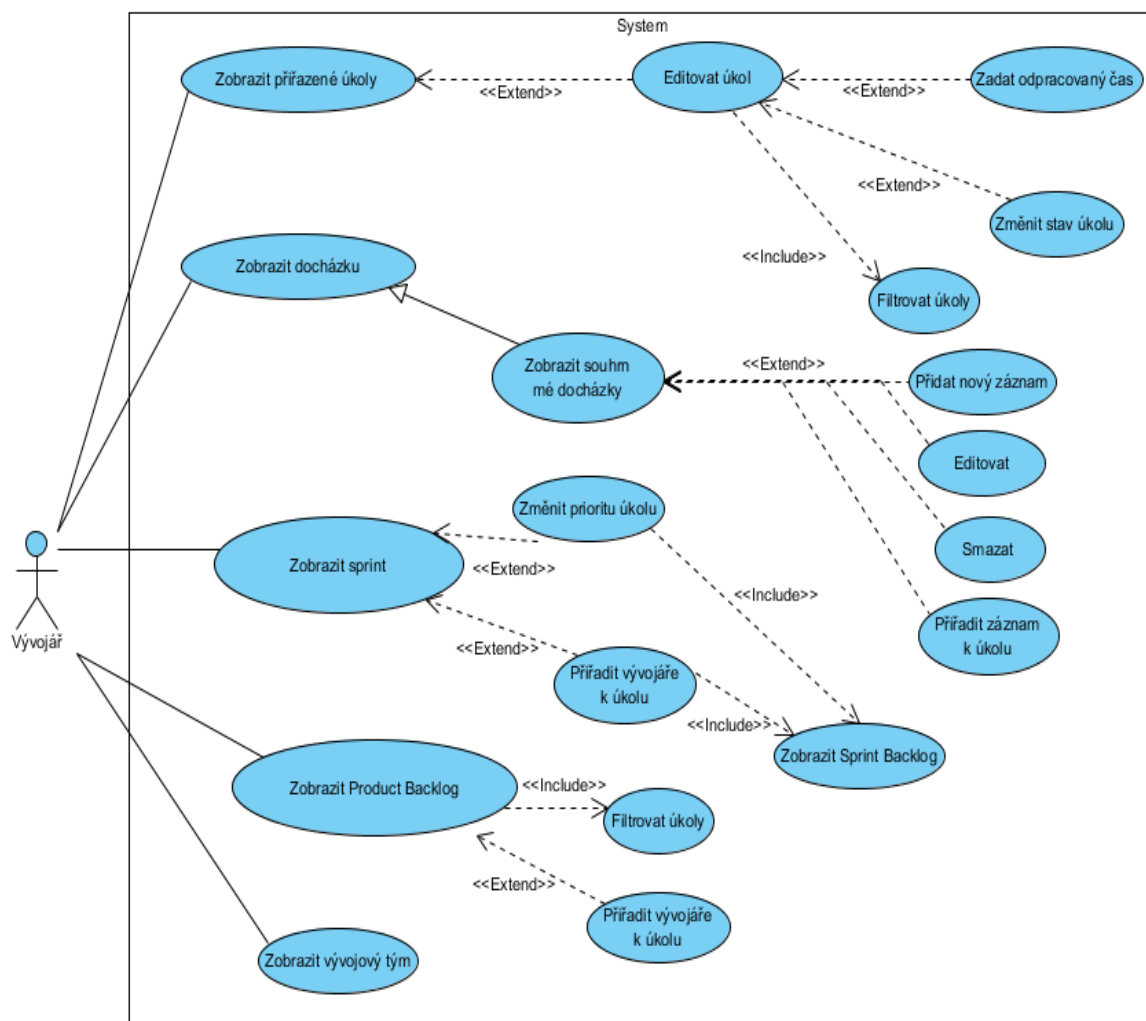
## 6.7 Diagramy případů užití

Všichni uživatelé systému mají společné aktivity přihlášení a odhlášení. Každý uživatel má své přihlašovací jméno a heslo. Pomocí těchto údajů se do systému přihlašují a na základě jejich ověření je jim umožněno se systémem pracovat.

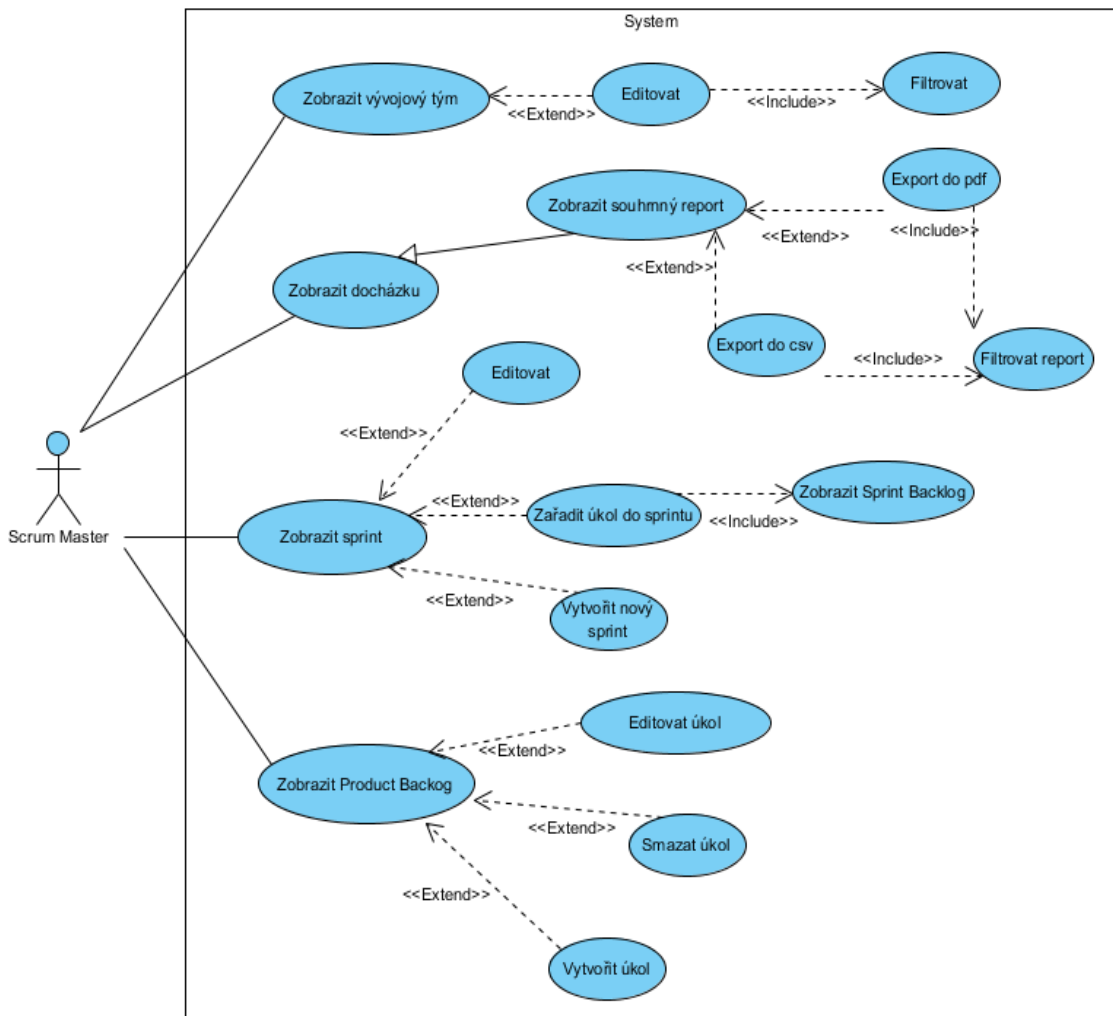
V následující části je pro každého aktéra uveden use case diagram, který znázorňuje funkcionalitu aplikace a jednotlivé činnosti, které smějí uživatelé prostřednictvím aplikace vykonávat.



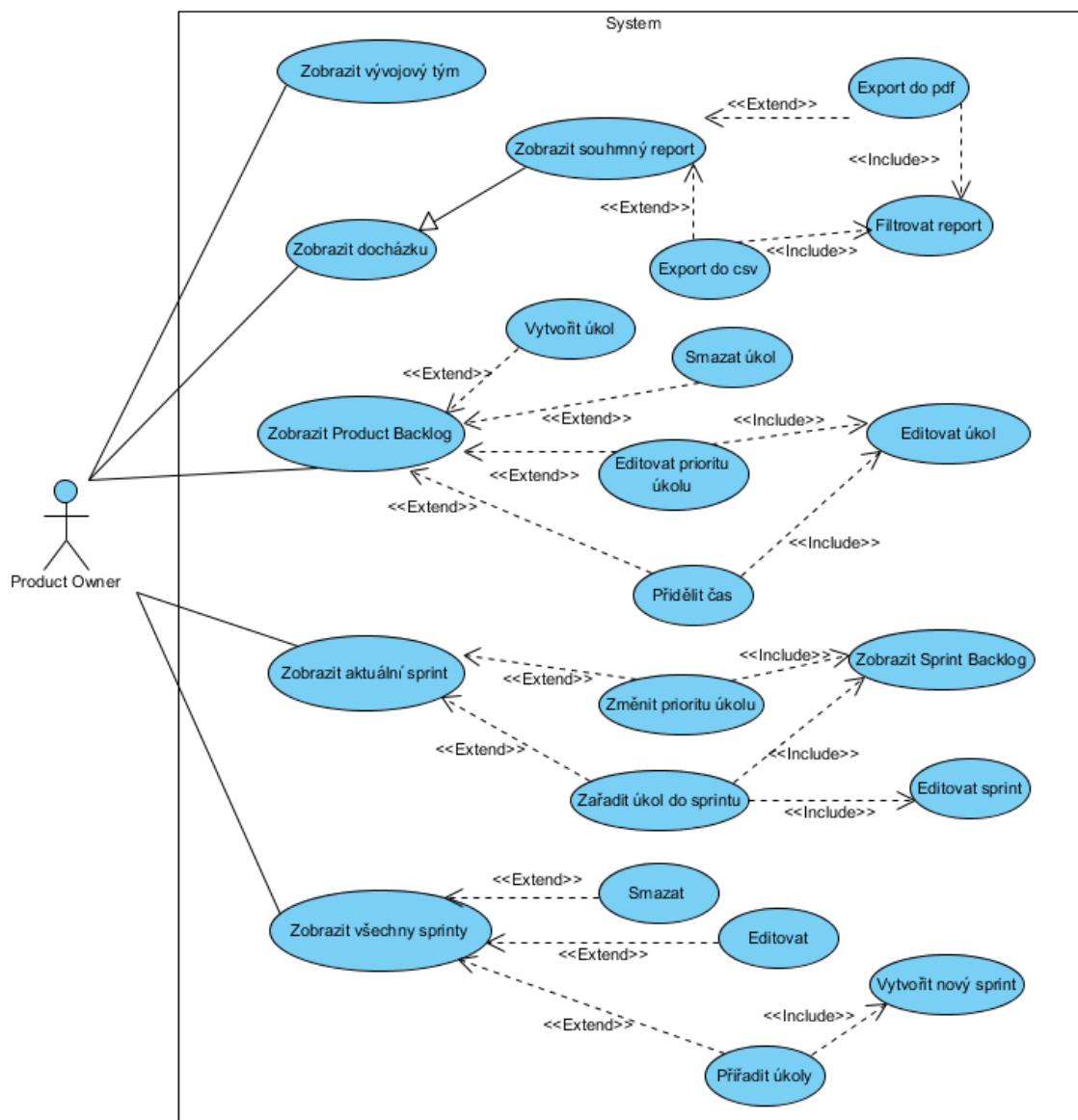
Obr. 14 Use case diagram Uživatel



Obr. 15 Use case diagram Vývojář



Obr. 16 Use case diagram Scrum Master



Obr. 17 Use case diagram Product Owner

### 6.7.1 Popis případů užití

#### Zobrazit Product Backlog

Uživatel má po přihlášení možnost zobrazit Product Backlog, který obsahuje všechny vytvořené úkoly. V tomto přehledu je možné filtrovat úkoly dle názvu, čísla úkolu, přiřazeného vývojáře, typu úkolu, stavu či projektu, do kterého je úkol zařazen. Vývojář si může přiřadit úkoly, na kterých bude pracovat. Product Owner i Scrum Master zde může dále vytvářet úkoly, editovat nebo mazat. Po vytvoření úkolu je možné úkolu přiřadit prioritu, dle které bude úkol následně zpracován.



## **Zobrazit souhrnný report docházky**

Přehled souhrnu docházky slouží pro uživatele Scrum Master a Product Owner. Souhrn lze filtrovat dle uživatele a následně exportovat do souboru ve formátech .pdf a .csv.

## **Zobrazit Sprint**

Každý uživatel si může v aplikaci zobrazit Sprint. Činnosti, které lze následně provádět jsou rozděleny podle práv na základě rolí uživatelů. Vývojář si v této sekci může zobrazit aktuální Sprint spolu s přiřazeným Sprint Backlogem. Zde má možnost upravit prioritu a editovat přiřazeného vývojáře. Ostatní uživatelé mohou v této části vidět nejen aktuální Sprint, ale i ostatní vytvořené Sprints. V přehledu mohou záznamy editovat, mazat či vytvářet nové. Po vytvoření nového Sprintu, je také možné přiřadit úkoly z Product Backlogu.

## **Zobrazit vývojový tým**

Aplikace také obsahuje sekci, ve které je možné si zobrazit členy vývojového týmu. Tento přehled zahrnuje základní údaje o jednotlivých členech týmu, jako je přihlašovací jméno, jméno, příjmení, e-mail, aktivita a datum posledního přihlášení. Product Owner může v tomto přehledu záznamy editovat.

## **Zobrazit přiřazené úkoly**

Tato část slouží pro vývojáře, kde po přihlášení vidí své úkoly, na kterých pracují. V přehledu přiřazených úkolů mohou vývojáři filtrovat úkoly dle jejich čísla, typu, stavu či projektu, do kterého je úkol přiřazen. Dále je možné úkoly editovat, konkrétně jejich stav a počet odpracovaných hodin.

## 6.7.2 Scénáře případů užití

Tab. 2 Scénář případu užití Přihlášení uživatele do systému

<b>Případ užití: Přihlášení uživatele do systému</b>		
<b>Krok</b>	<b>Role</b>	<b>Akce</b>
1	Uživatel	Spuštění aplikace v prohlížeči.
2	System	Zobrazení přihlašovacího formuláře.
3	Uživatel	Vyplnění přihlašovacího jména a hesla.
4	System	Ověření správnosti přihlašovacích údajů.
5	System	V případě správnosti údajů, je uživatel na základě jeho práv přesměrován na příslušnou stránku aplikace. V opačném případě je zobrazena hláška o špatně zadaných údajích.

Tab. 3 Scénář případu užití Editace priority úkolu

<b>Případ užití: Editace priority úkolu</b>		
<b>Krok</b>	<b>Role</b>	<b>Akce</b>
1	Uživatel	Volba „Aktuální Sprint“.
2	System	Zobrazení informací o Sprintu.
3	Uživatel	Zvolení detailu Sprintu – Sprint Backlog.
4	System	Zobrazení přiřazených úkolů k danému Sprintu.
5	Uživatel	Pomocí ikony přetažení úkolů do požadovaného pořadí.
6	System	Na základě zobrazeného pořadí systém uloží správné hodnoty priority do databáze.

Tab. 4 Scénář případu užití Zařazení úkolu do Sprintu

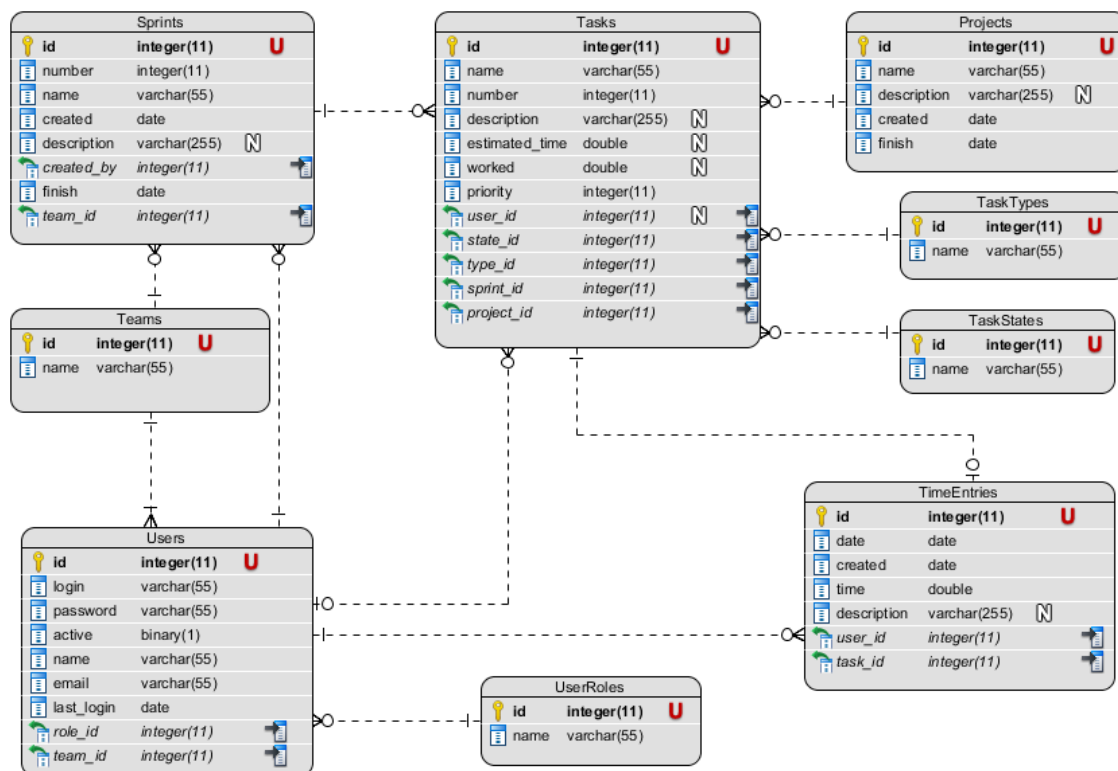
<b>Případ užití: Zařazení úkolu do Sprintu</b>		
<b>Krok</b>	<b>Role</b>	<b>Akce</b>
1	Uživatel	Volba „Editace“ na záložce Sprint.
2	Systém	Zobrazení příslušného formuláře pro editaci.
3	Uživatel	Pomocí checkboxů uživatel zatrhne úkoly, které chce do daného Sprintu přiřadit a poté klikne na „Uložit“.
4	Systém	Uloží do databáze do tabulky s úkoly ID daného Sprintu.

Tab. 5 Scénář případu užití Export do .pdf

<b>Případ užití: Export do .pdf</b>		
<b>Krok</b>	<b>Role</b>	<b>Akce</b>
1	Uživatel	Zvolení „Zobrazit souhrnný report“ na záložce „Docházka“.
2	Systém	Zobrazení přehledu záznamů docházky.
3	Uživatel	Volba „Export souhrnu do pdf“.
4	Systém	Vykreslení údajů docházky do šablony a následně stažení souboru ve formátu souhrn-dochazky.pdf.

## 6.8 Datová struktura

Jak už bylo zmíněno výše, aplikace bude napojena na stávající systém, který pracovníci využívají. Část dat, které bude aplikace využívat, jsou totožná s již zavedenými daty. Jedná se především o údaje o uživatelích, úkolech a projektech. Po úspěšném nasazení a důkladném testování aplikace bude databáze sjednocena a údaje potřebné pro nový modul budou získávány ze současné databáze. Samotné napojení na provozní server a následně databázi proběhne ve finální fázi po dokončení testování. Z tohoto důvodu je vývoj prototypu nového modulu založen na samostatné databázi, která obsahuje testovací data a slouží tak jen pro vývojové účely. Následující obrázek č. 18 znázorňuje datový model vývojové databáze.



Obr. 18 Entitně relační diagram

V datovém modelu se vyskytuje devět entit včetně svých atributů. Názvy entit a atributů jsou v anglickém jazyce, a to z důvodu zachování konvence a standardů využívané vývojovým týmem při kódování. Mezi nejdůležitější entity databáze patří:

- **Tasks** – tato entita obsahuje veškeré úkoly a příslušné informace. Jsou v ní evidovány odkazy na přiřazeného vývojáře, stav úkolu, jeho typ, zařazení do daného Sprintu a projektu. Prostřednictvím příslušných ID lze získat informace z tabulek *User*, *TimeEntries*, *TaskTypes*, *TaskStates*, *Sprints* a *Projects*. Vzhledem k situaci, že daný úkol náleží vždy právě jednomu projektu, ale zároveň každý projekt obsahuje mnoho úkolů, je vazba mezi tabulkami *Tasks* a *Projects*, N:1. Co se týká přiřazeného stavu a typu úkolu, zde je vazba také řešena typem N:1. To z důvodu, že každému úkolu náleží vždy právě jeden typ a stav. Naopak daného typu i stavu může nabývat více úkolů. Stejná vazba se týká i mezi entitami *Tasks* a *Sprints*. Každý úkol je součástí právě jednoho Sprintu, ale daný Sprint obsahuje více úkolů.
- **Users** – zde jsou uloženy informace o uživatelích využívající aplikaci. O každém uživateli je evidováno jeho přihlašovací jméno, heslo, celé jméno, e-mail, datum posledního přihlášení, ID role a týmu, jehož je součástí. Informace o rolích a týmech jsou následně uloženy v samostatných tabulkách, ke kterým se při-

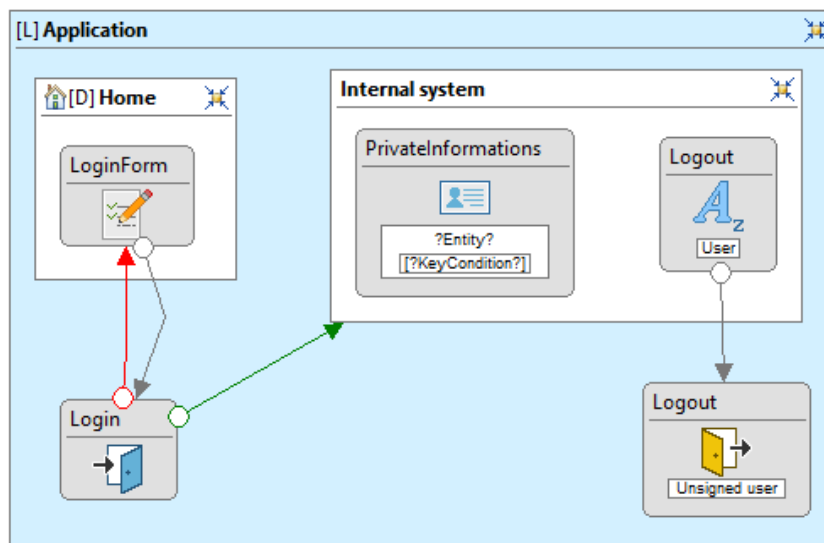
stupuje právě prostřednictvím ID. Vazba mezi uživatelem a jeho rolí je N:1, a to z toho důvodu, že každý uživatel zastává právě jednu roli a naopak jednotlivé role mohou být představovány několika uživateli. Každý uživatel je členem určitého týmu, který obsahuje více vývojářů.

- *TimeEntries* - vývojáři vykazují svůj čas prostřednictvím zadávání docházky. Tyto údaje jsou shromážděny v tabulce *TimeEntries*, kde každý záznam náleží právě jednomu úkolu.
- *Sprints* – zde jsou obsaženy veškeré údaje o vytvořených Sprintech. Každý Sprint obsahuje více úkolů, naopak úkoly jsou zahrnuty vždy v právě jednom Sprintu. Z toho důvodu je mezi těmito entitami vazba 1:N. U záznamu Sprintu je uložen název, číslo, informace o datu jeho vzniku, popis, datum ukončení a ID osoby, která jej vytvořila. U každého záznamu je také evidováno ID týmu. To vychází z faktu, že vývojáři jsou rozděleni do více týmu, z nichž každý má svůj Sprint.
- *Projects* – tato entita obsahuje základní údaje o projektu. Projekt je složen z více úkolů, které jsou obsaženy v tabulce *Tasks*. Vazba mezi entitami *Projects* a *Tasks* je tedy 1:N.

## 6.9 Modely IFML

Součástí návrhu webové aplikace jsou IFML modely, které znázorňují strukturu a chování aplikace. Modely jsou vytvořeny v CASE nástroji WebRatio s využitím pohledu *SiteView*. Pro lepší přehlednost je celkový IFML model rozdělen na dílčí modely podle přidělené role uživatele.

### 6.9.1 Nepřihlášený uživatel



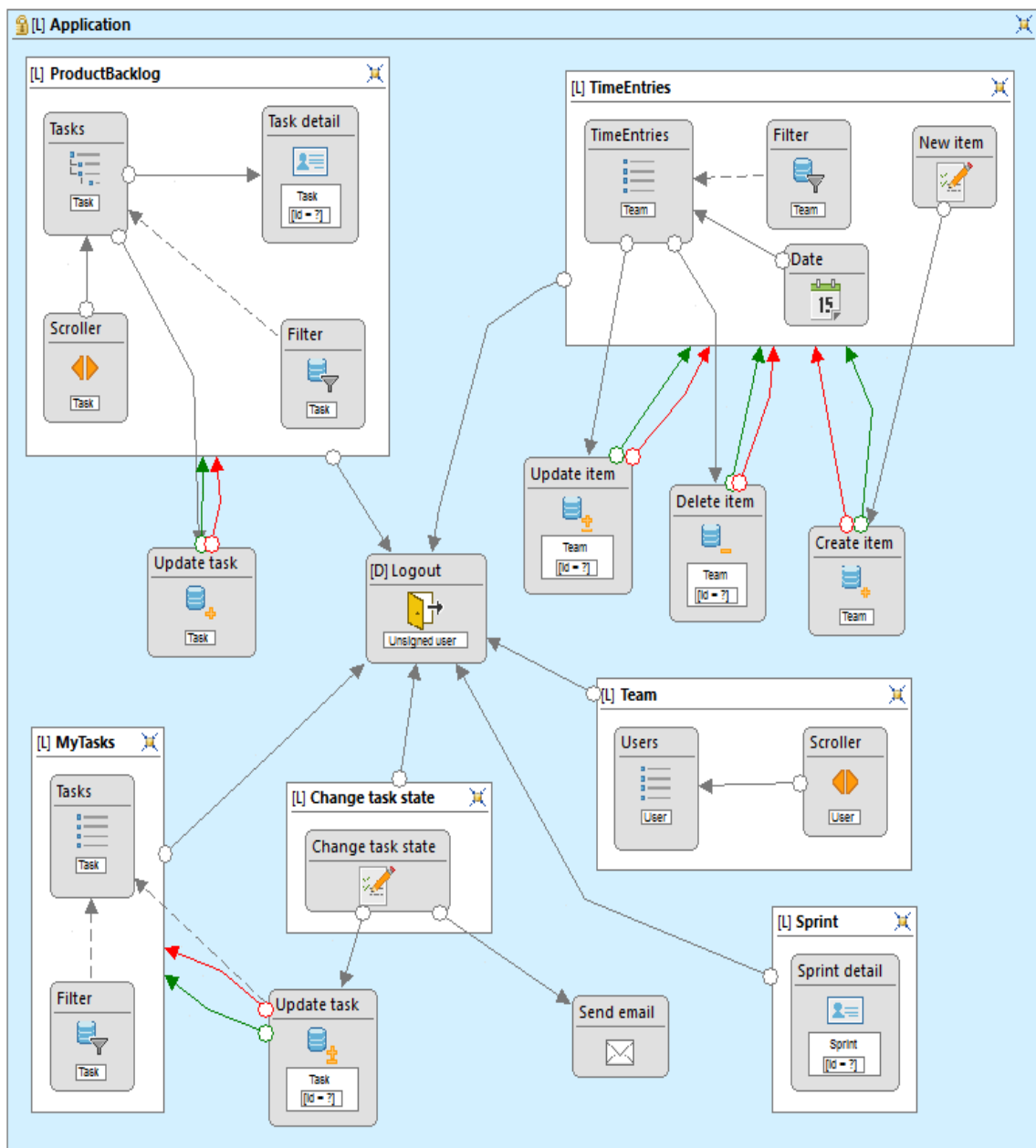
Obr. 19 IFML model pro nepřihlášeného uživatele

Po přístupu do aplikace se uživateli zobrazí přihlašovací formulář, kde musí vyplnit uživatelské jméno a heslo. Na základě ověření zadaných údajů je uživatel přesměrován na danou stránku odpovídající jeho přiřazené roli.

### 6.9.2 Vývojář

Uživatel v roli vývojáře vidí po přihlášení seznam úkolů (*Product Backlog*). V této části má možnost úkoly filtrovat a následně k danému úkolu přiřadit vývojáře. V menu aplikace má dále možnost zobrazit přiřazené úkoly, docházku, Sprint a vývojový tým. V sekci *Moje úkoly* jsou vývojáři zobrazeny veškeré úkoly, které má přiřazený. Zde může u úkolů editovat stav a počet odpracovaných hodin. Při editaci stavu úkolu má možnost vybrat z pěti stavů, a to - udělat, ve vývoji, revize kódu, test a hotovo. V přehledu úkolů může uživatel také filtrovat, a to podle čísla úkolu, typu, stavu či projektu, do kterého je úkol zařazen. Další důležitou sekcí je *Docházka*, která je rozdělena na dvě části. První je část *Souhrn*, kde vývojář vidí souhrn své docházky dle filtrovaného období. Druhou částí je *Moje docházka*, kde jsou zobrazeny všechny záznamy. Tyto záznamy lze jednak smazat, ale i editovat. Při editaci lze upravit úkol, ke kterému je daný záznam přiřazen, datum, počet odpracovaných hodin, stav úkolu, kterého se záznam týká a případně lze přidat poznámku. Samozřejmostí je také vytváření nových záznamů. Mezi přehledové části aplikace patří záložky *Vývojový tým* a *Sprint*. Pro vývojáře jsou tyto části pouze informativního charakteru. V přehledu *Vývojový tým* jsou uvedeni všichni členové daného týmu. Je zde uvedena informace o aktivitě uživatele, přihlašovací jméno, jméno, e-mail a datum posledního přihlášení. V části *Sprint* si může vývojář zobrazit aktuální

sprint a s ním související informace, jako je jeho číslo, název, popis, kdy byl vytvořen a datum jeho ukončení. Po otevření detailu daného Sprintu jsou zobrazeny přiřazené úkoly, tzv. *Sprint Backlog*, kde vývojář může měnit prioritu úkolu a přiřadit vývojáře.

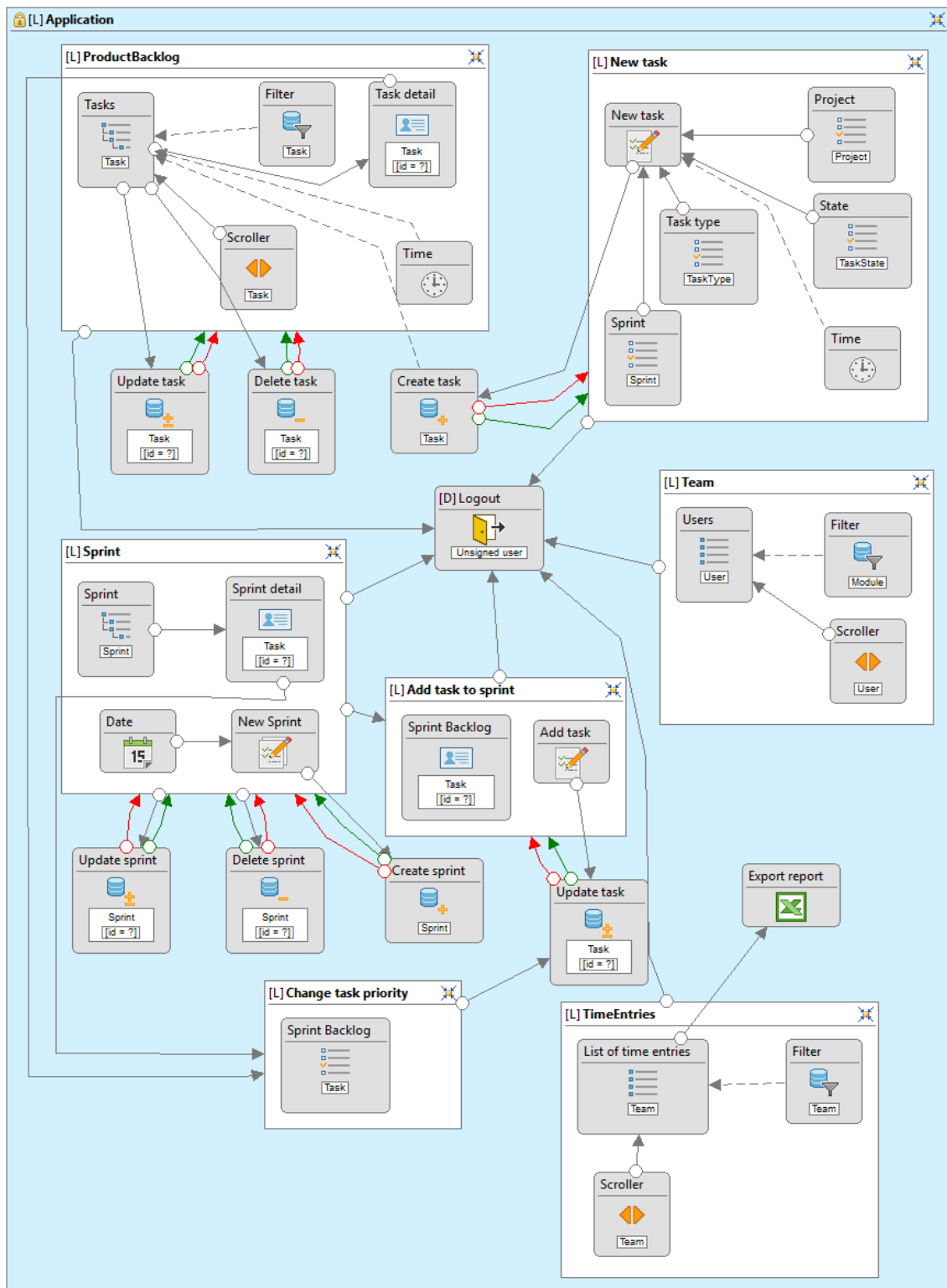


Obr. 20 IFML model pro vývojáře

### 6.9.3 Product Owner

Pro vlastníka produktu je klíčová část *Product Backlog*, která slouží pro tvorbu nových úkolů a editaci stávajících. Při vytvoření nového úkolu je zobrazen formulář, kde je potřebné vyplnit číslo úkolu, název, popis, čas, za který má být úkol dokončen, typ úkolu, stav, zařazení do Sprintu a projektu. V přehledu úkolů je také možné filtrovat záznamy dle několika kritérií. Po otevření detailu úkolu jsou uvedeny informace o daném Sprintu, do kterého je úkol zařazen. V části *Sprint* má Product Owner možnost zobrazit nejen aktuální Sprint, ale také všechny dosud vytvořené. Pro uživatele této role je také důležitá sekce *Docházka*, kde má uživatel možnost zobrazit souhrnný report, který lze následně exportovat do souborů o formátech .pdf a .csv. I v této části je samozřejmostí filtrace záznamů, a to podle uživatele.

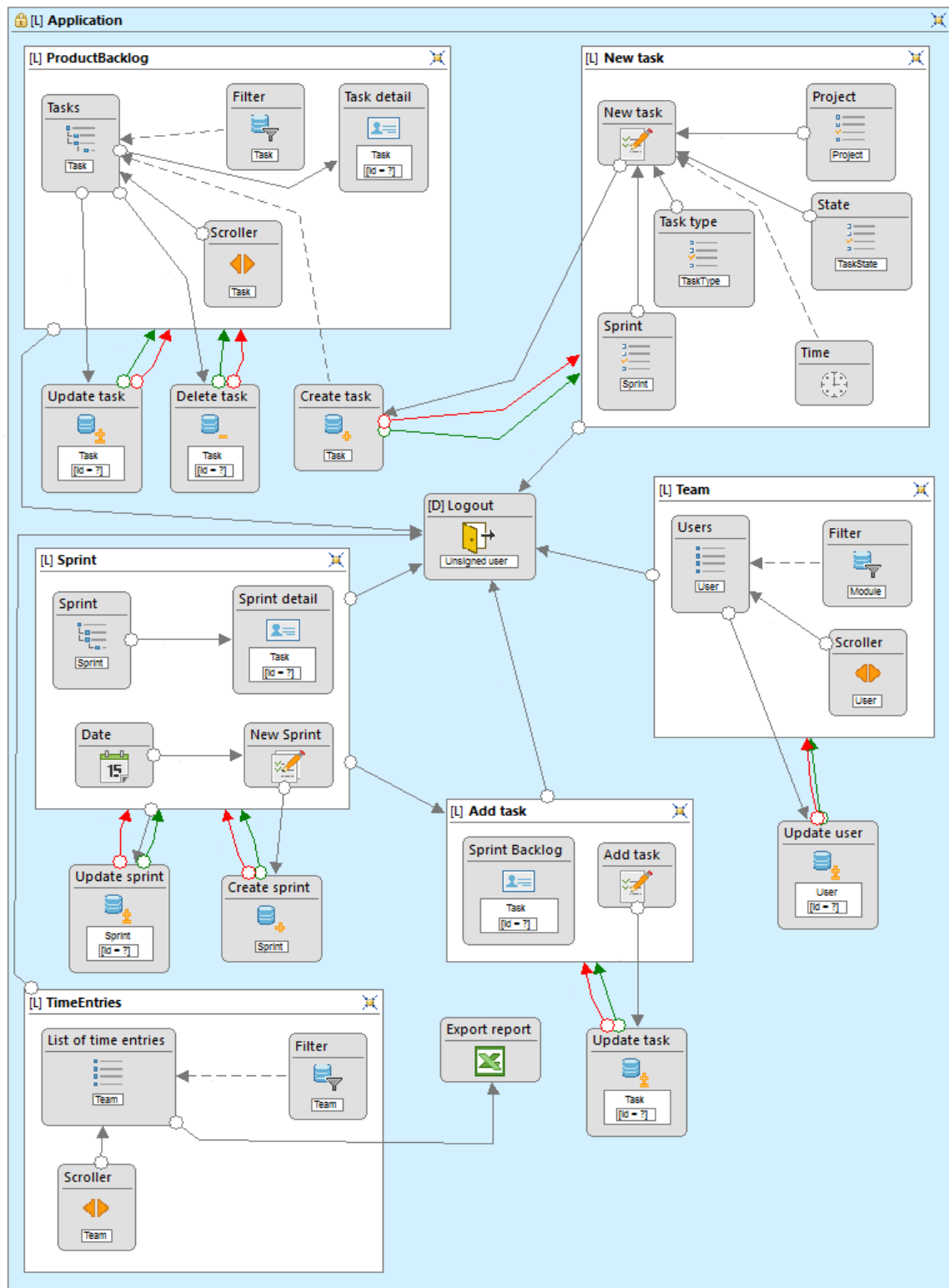




Obr. 21 IFML model pro uživatele Product Owner

#### 6.9.4 Scrum Master

IFML model pro uživatele v roli Scrum Master (obr. 22) je blízký předchozímu pro vlastníka produktu. Mezi hlavní rozdíly patří možnost editace vývojového týmu a zobrazení sekce *Sprint*. V této části má Scrum Master právo zobrazit pouze aktuální Sprint. Právo editace a zobrazení seznamu *Sprint Backlog* zůstává stejné.



Obr. 22 IFML model pro uživatele Scrum Master

## 6.10 Implementace

Implementace aplikace je založena na předchozí analýze požadavků. Byly specifikovány technologie, které jsou při implementaci použity. Vývoj aplikace probíhal na lokálním webovém serveru XAMPP. Po jejím dokončení byla aplikace pro prezentační účely nahrána na webhosting. Z důvodu předchozí kladné zkušenosti byl vybrán PHP5 Freehosting. Daný webhosting podporuje PHP verzi 5.6 MySQL 5.6, poskytuje prostor pro nahrání skriptů o velikosti až 50 MB a pro přístup používá FTP a SSH. Pro splnění účelu je tedy vybraný webhosting dostačující. Vytvořená webová aplikace je dostupná na adrese [www.bp-is.php5.cz/www/sign/in](http://www.bp-is.php5.cz/www/sign/in). V současné době dostupná aplikace využívá testovací data a není napojena na provozní databázi.

Aplikace je vytvořena s využitím open source PHP frameworku Nette. Framework je založen na návrhovém vzoru MVC (Model-View-Controller), který umožňuje oddělení aplikační logiky a samotného vzhledu zobrazení koncovému uživateli. V NetteFrameworku funkci Controlleru zastává Presenter, který stejně jako Controller zpracovává požadavky od uživatele a následně generuje odpovědi. Aplikace kromě defaultních využívá tyto presentery:

- *ProductBacklogPresenter* – zajišťuje správu úkolů.
- *SignPresenter* – slouží k přihlášení do aplikace.
- *SprintPresenter* – zajišťuje funkce pro správu Sprintů.
- *TasksPresenter* – slouží pro zobrazení a správu přiřazených úkolů.
- *TeamPresenter* – slouží pro zobrazení a správu vývojového týmu.
- *TimeEntriesPresenter* – zahrnuje funkcionalitu pro zobrazení přehledu docházky, jejího vytvoření, editaci a exportu.

Co se týká prezentačního návrhu aplikace, požadavky na vzhled nebyly blíže specifikovány. Hlavním kritériem bylo vytvořit přehlednou aplikaci s intuitivním uživatelským rozhraním. Jednotlivé prvky jsou rozvrženy tak, aby bylo zajištěno uživateli snadné ovládání. Pro samotný vzhled byl vytvořen vlastní kaskádový styl. Použité barvy jsou v souladu s barvami ve stávající aplikaci, na kterou bude nově vytvořená aplikace napojena. Některé prvky aplikace však využívají styl externí. Jedná se například o rozšiřující komponenty DatePicker a DataGrid.

Pro řízení práv a přístupů do jednotlivých částí aplikace byl vytvořen model *AclManager*, který obsahuje seznam pravidel pro identifikaci rolí a práv k daným zdrojům aplikace. Nastavení práv vyžaduje definici role, zdroje a operace, která má být vykonána. Ukázka použití rolí a nastavení přístupu ke zdroji aplikace:

```
$this->acl->addRole('scrum master');
$this->acl->addResource('productbacklog');

$this->acl->allow('scrum master', 'productbacklog',
    array('default', 'edit', 'create'));
```

Jak už bylo výše řečeno, v aplikaci jsou tři typy uživatelů. A to vývojář, Scrum Master a Product Owner. Pro vývojové účely a přístup do aplikace byli založeni následující uživatelé:

Tab. 6 Testovací přihlašovací údaje pro uživatele

Role	Login	Heslo
Vývojář	v.prokopova	devtest
Scrum Master	j.novak	scrumtest
Product Owner	p.herout	ownertest

Aplikace obsahuje jednoduchou úvodní stránku s přihlašovacím formulářem. Po přihlášení je uživatel přesměrován na první položku navigace – *Product Backlog* (obr. 23). Aplikace obsahuje horizontální navigaci. V její pravé části jsou informace o přihlášeném uživateli a volba odhlášení. V levé části jsou pak jednotlivé položky:

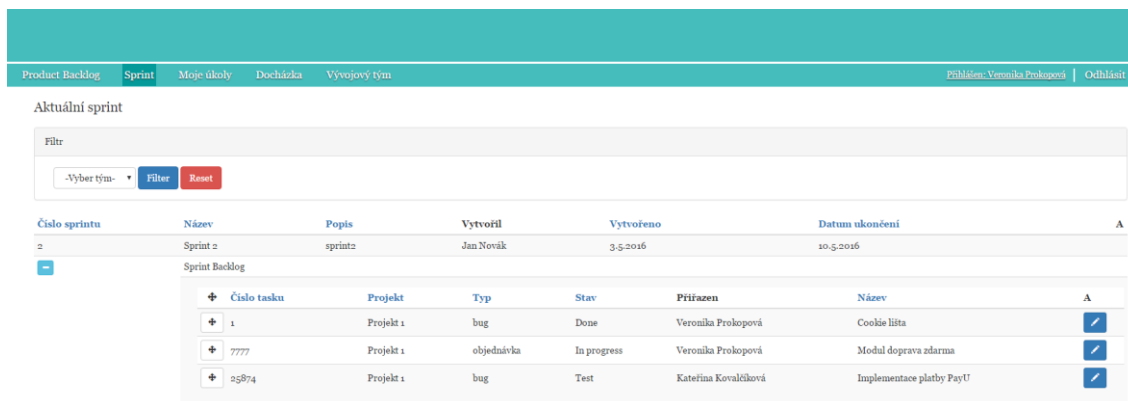
- Product Backlog,
- Sprint,
- Moje úkoly,
- Docházka,
- Vývojový tým.

Číslo úkolu	Projekt	Typ	Stav	Přiřazen	Název	Vytvořeno	Čas (h)	Odpracováno (h)	A
1	Projekt 1	bug	Test	Veronika Prokopová	Cookie lísta	22.4.2016	1.00	2.50	
+ Detail sprintu									
777	Projekt 1	objednávka	In progress	Veronika Prokopová	Modul doprava zdarma	28.4.2016	3.50	0.00	
+ Detail sprintu									
88364	Projekt 1	bug	To do	Jan Novák	Nefunkční konektor money5	22.4.2016	2.00	1.00	
+ Detail sprintu									
100	Projekt 1	bug	In progress	Veronika Prokopová	Nepřenesené objednávky	22.4.2016	1.00	4.50	
+ Detail sprintu									
89745	Projekt 1	bug	In progress	Veronika Prokopová	Nasazení modulu hromadná správa dat	22.4.2016	2.00	1.00	
+ Detail sprintu									
333	Projekt 1	objednávka	To do	Petr Klusáček	Mapa stránek	22.4.2016	2.00	0.00	
+ Detail sprintu									
25874	Projekt 1	bug	Test	Kateřina Kovalčíková	Implementace platby PayPal	22.4.2016	6.00	1.00	
+ Detail sprintu									

Obr. 23 Product Backlog

Na následujícím obrázku (obr. 24) je zobrazen aktuální sprint. Po kliknutí na detail sprintu, je zobrazen *Sprint Backlog*, který obsahuje všechny přiřazené úkoly do

daného sprintu. Pomocí funkce drag & drop je možné přesouvat úkoly a tím měnit jejich prioritu.



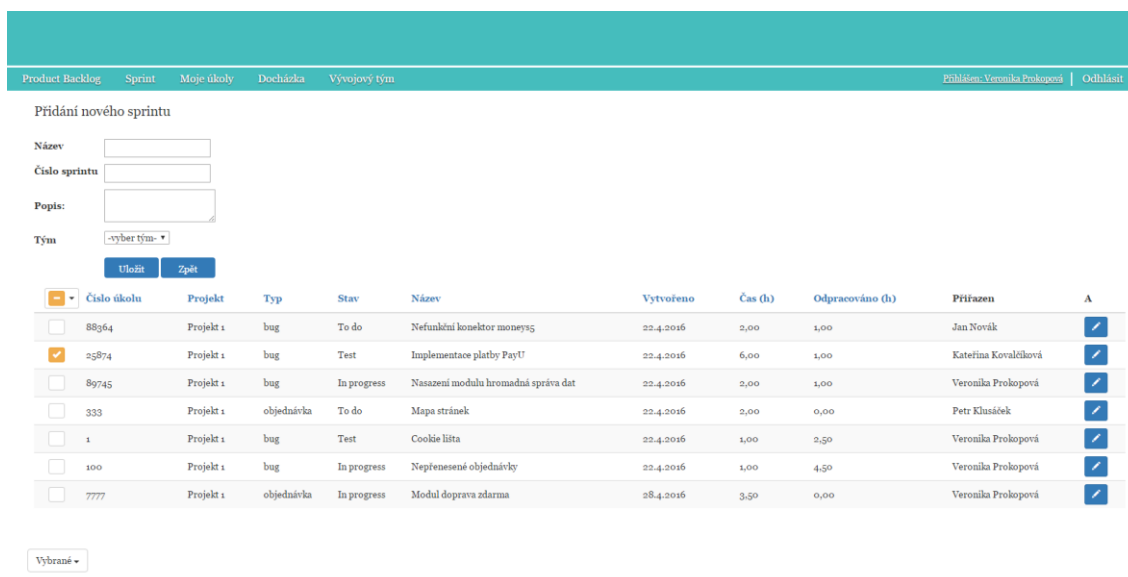
Číslo sprintu	Název	Popis	Vytvořil	Vytvořeno	Datum ukončení	A
2	Sprint 2	sprint2	Jan Novák	3.5.2016	10.5.2016	

Číslo úkolu	Projekt	Typ	Stav	Přifazen	Název	A
1	Projekt 1	bug	Done	Veronika Prokopová	Cookie lišta	<input checked="" type="checkbox"/>
7777	Projekt 1	objednávka	In progress	Veronika Prokopová	Modul doprava zdarma	<input checked="" type="checkbox"/>
25874	Projekt 1	bug	Test	Kateřina Kovalčíková	Implementace platby PayU	<input checked="" type="checkbox"/>

Obr. 24 Aktuální sprint

Obr. 25 zachycuje formulář pro vytvoření nového sprintu. Po vyplnění údajů a přidělení týmu je možné pomocí checkboxů u jednotlivých úkolů přidat vybrané úkoly do daného sprintu.



Přidání nového sprintu

Název:

Číslo sprintu:

Popis:

Tým:

Číslo úkolu	Projekt	Typ	Stav	Název	Vytvořeno	Čas (h)	Odpracováno (h)	Přifazen	A
<input type="checkbox"/> 88364	Projekt 1	bug	To do	Nefunkční konektor money5	22.4.2016	2,00	1,00	Jan Novák	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> 25874	Projekt 1	bug	Test	Implementace platby PayU	22.4.2016	6,00	1,00	Kateřina Kovalčíková	<input checked="" type="checkbox"/>
<input type="checkbox"/> 89745	Projekt 1	bug	In progress	Nasazení modulu hromadná správa dat	22.4.2016	2,00	1,00	Veronika Prokopová	<input checked="" type="checkbox"/>
<input type="checkbox"/> 333	Projekt 1	objednávka	To do	Mapa stránek	22.4.2016	2,00	0,00	Petr Klusáček	<input checked="" type="checkbox"/>
<input type="checkbox"/> 1	Projekt 1	bug	Test	Cookie lišta	22.4.2016	1,00	2,50	Veronika Prokopová	<input checked="" type="checkbox"/>
<input type="checkbox"/> 100	Projekt 1	bug	In progress	Nepřenesené objednávky	22.4.2016	1,00	4,50	Veronika Prokopová	<input checked="" type="checkbox"/>
<input type="checkbox"/> 7777	Projekt 1	objednávka	In progress	Modul doprava zdarma	28.4.2016	3,50	0,00	Veronika Prokopová	<input checked="" type="checkbox"/>

Vybráně:

Obr. 25 Formulář pro nový sprint

Ve všech sekcích aplikace lze zobrazená data filtrovat podle několika kritérií, která jsou přizpůsobena zobrazovaným datům, což umožňuje snadnější orientaci v datech. V případě vztahů mezi více tabulkami propojenými prostřednictvím klíčů, je zobrazena roletka pro výběr dané hodnoty. Pro filtraci data je možné vybrat konkrétní časový úsek, například týden nebo měsíc. Lze zadat ale i vlastní definované rozmezí. Tento druh filtru má vhodné využití především v části *Docházka*, kde je možné díky němu zobrazovat přehled docházky za specifické období.

Na přehledech je také implementováno stránkování, které poskytuje rychlejší přesun mezi jednotlivými záznamy. Pro každé stránkování je nastavena hodnota počtu záznamů, od kterého se stránkování zobrazuje.

Každý přehled obsahuje řazení, u kterého je možné nastavit zobrazení pouze pro zvolené sloupce. Jinak řečeno, pro sloupce, pomocí kterých nechceme přehled řadit, lze tuto funkci vypnout. Tato funkčnost se týká i filtrace. Lze jasně definovat, které sloupce mají být součástí filtrace a které naopak ne.

Všechny přehledy záznamů obsahují funkčnost tzv. inline editace. Při dvojkliku na dané pole, je možné ho editovat a při stisknutí klávesy enter, se daná změna uloží. Přehledy obsahují i klasikou editaci prostřednictvím editační ikony u každého záznamu a následného zobrazení editačního formuláře. Inline editace však umožňuje provádět rychlejší úpravy. Při editaci jsou kontrolovány vstupní hodnoty a při jejich špatném zadání jsou zobrazeny informační hlášky. Pro zadání data do formuláře je využita komponenta DatePicker, která má uživatelsky přívětivé zobrazení a umožňuje výběr data prostřednictvím zobrazeného kalendáře.

## 6.11 Testování a nasazení aplikace

Souběžně s implementací probíhalo testování definovaných funkčních požadavků. Po dokončení prototypu aplikace proběhla ve vybrané firmě prezentace navrženého řešení. Zástupci vývojového týmu byly s daným řešením spokojeni a podle jejich tvrzení, aplikace odpovídá zadaným požadavkům. Poté mohlo začít samotné testování aplikace ve vývojovém prostředí. Testování probíhalo v několika fázích. Nejdříve bylo potřeba ověřit napojení na server a komunikace s databází. V další fázi se testování účastnili členové vývojového týmu, kteří se zaměřovali především na testování funkčnosti. Toto testování zahrnovalo ověření odkazů, tvorbu, editaci a mazání záznamů. Součástí testování byla také kontrola rozhraní aplikace, zda je uživatelsky přívětivá a intuitivní.

## 6.12 Návrhy na vylepšení

I přes kladné ohlasy během testování aplikace, se objevily návrhy na vylepšení. Tyto nové požadavky by bylo možné zahrnout do vývoje aplikace v další fázi. V budoucnu by aplikace mohla být rozšířena o následující funkcionality:

- automatické zasílání souhrnu docházky – vždy na konci měsíce zasílání přehledu na email.
- jazykové mutace – přidání možnosti zobrazit aplikaci v anglickém jazyce.
- tvorba dokumentace k úkolům – možnost ukládání příloh a doplňujících souborů, obrázků apod.

## 7 Diskuze

V oblasti projektového řízení v IT je využíváno několik druhů metodik a nástrojů. Obor informačních technologií prochází neustálým vývojem a spolu s ním také rostou požadavky na zdokonalování softwarových produktů a služeb. Pro kvalitu produktu je stěžejní právě metoda jeho řízení a vývoje. Tradiční, tzv. rigorózní metodiky využívané pro vývoj softwaru se stávají nedostačujícími, a proto vznikly agilní metodiky. Právě agilní metodiky umožňují rychlý vývoj a dynamickou reakci na měnící se požadavky zákazníků. Hlavní odlišností těchto dvou přístupů je způsob specifikace požadavků. V případě použití rigorózních metodik jsou požadavky definovány v počáteční fázi projektu, naopak agilní metodiky umožňují změnu požadavků v průběhu projektu. V agilním vývoji je kladen velký důraz na zákazníka. Zákazník je zapojen do vývoje během celého průběhu životního cyklu projektu. Důležitou součástí je neustálá interakce nejen uvnitř vývojového týmu, ale i se zákazníkem.

Agilní metodiky jsou využívány především ve Spojených státech. V posledních letech dochází k rostoucímu trendu těchto metodik i v České republice. Z uvedených výsledků průzkumu vyplývá, že u dotazovaných převažuje využívání agilní metodiky Scrum. Hlavním důvodem pro jejich zavedení je pak zvýšení produktivity, kvality produktu či zjednodušení vývojového procesu. Ačkoliv tyto metodiky vykazují velké množství výhod, jejich používání ve vývoji softwaru není příliš obvyklé. V porovnání s ostatními přístupy má využití agilních metodik při řízení projektů téměř 50 % zastoupení. S přihlédnutím k jejich vývoji a vzniku, tedy přibližně přelomu 20. a 21. století, je tento pokrok však výrazný. Převaha tradičních metodik při vývoji a řízení projektů může být způsobena nejen nedostatečnou znalostí agilních metodik, ale také nedůvěrou v nové, inovativní přístupy a změny, které mohou s jejich zavedením nastat.

Pro znázornění procesu řízení projektů byla využita notace BPMN. Procesní diagramy poskytují přehlednou a názornou představu o průběhu celého procesu projektového řízení. Hlavní výhodou je možnost dekompozice procesu, což umožňuje jeho detailní popis. Namodelované diagramy pak slouží nejen pro zobrazení současného stavu procesu, ale také i pro jeho následnou optimalizaci. Na základě analýzy procesních modelů je možné pozorovat nedostatky procesu a navrhnout jeho inovace.

Právě zavedení metodiky Scrum se stalo součástí této práce. V práci jsou blíže popsány základní principy a podstata této metodiky. Pro agilní vývoj a řízení projektů je dostupných několik nástrojů. Tyto aplikace nabízí mnoho funkcí a ve většině případů jsou k dispozici ve formě webových aplikací. V současné době společnost využívá vlastní interní aplikaci, která však nepodporuje principy agilního vývoje projektů a proto není pro navrhované změny vhodná. Hlavním požadavkem společnosti bylo využití stávající aplikace a z tohoto důvodu není možné použít již dostupná řešení.

Před samotnou implementací webové aplikace je vhodné danou aplikaci namodelovat a znázornit tak její strukturu a chování. K tomuto účelu slouží specifické



modelovací jazyky určené právě pro vývoj webových aplikací. V práci byl využit relativně nový jazyk IFML, který vznikl přibližně v roce 2014 v návaznosti na jazyk WebML. Hlavním rozšířením, které IFML oproti svému předchůdci nabízí, je možnost vývoje mobilních aplikací. Pro modelování aplikací pomocí notace IFML slouží nástroj WebRatio. Pomocí IFML modelů lze vyjádřit chování a interakci mezi uživatelem a aplikací. Metodika IFML však nepodporuje analýzu a specifikaci požadavků, která je pro modelování funkcionality nezbytná. Proto je vhodné použít Use case diagramy jazyka UML. Na vytvořené diagramy lze pak navázat při tvorbě IFML modelů. Nástroj WebRatio umožňuje na základě vytvořených modelů generování kódu. Nevýhodou je však omezení generování pouze v jazyce Java. S ohledem na vývoj notace IFML i nástroje WebRatio je možné v budoucnu očekávat přidání možnosti generování i do jiných programovacích jazyků.

Prototyp aplikace splňuje téměř všechny požadavky společnosti a přes drobné nedostatky je zcela funkční. V další fázi vývoje by měly být postupně doplněny nové funkcionality. Jedná se o automatické zasílání souhrnu docházky, kdy by se měl vždy na konci měsíce zasílat přehled docházky na uvedený email. Dalším rozšířením je přidání možnosti zobrazit aplikace v anglickém jazyce. Přínosnou funkcionalitou by také byla možnost ukládání doplňujících souborů k úkolům. Po úpravách následně proběhne opětovné testování a napojení na provozní databázi. V případě spokojenosti vývojového týmu lze v budoucnu doporučit rozšíření aplikace o tvorbu projektů, evidenci požadavků, cenových nabídek, harmonogramu projektů či tvorbu dokumentace. Toto rozšíření by také znamenalo úpravu propojení se stávajícími informačními systémy.

## 8 Závěr

Cílem této práce bylo navrhnout inovaci řízení projektů ve zvolené firmě, která se zabývá vývojem e-shopů a dalšími službami podporující obchodování na internetu. Součástí inovace bylo zavedení agilní metodiky Scrum a s ohledem na tuto metodu navrhnout změny workflow řízení projektů. Dílčím cílem práce bylo navrhnout prototyp webové aplikace, sloužící jako podpora vývojového týmu.

První část práce byla věnována teoretickým znalostem z oblasti projektového řízení. Byly blíže definovány základní pojmy, oblasti projektového řízení, význam procesního řízení a životní cyklus projektu. Tato část se také zabývala přístupy k řízení a vývoji softwarových projektů. Byly diskutovány tradiční a agilní metodiky vývoje, jejich výhody a nevýhody. Agilním metodikám byla věnována samostatná kapitola, kde byl blíže specifikován princip agilních metodik, současný stav užívání a důvody pro jejich zavedení. Část této kapitoly byla zaměřena na konkrétní agilní metodiku Scrum, kde byly blíže popsány základní principy této metodiky a postup při jejím zavedení do podnikového prostředí.

Před praktickou částí bylo nezbytné zvolit vhodné nástroje a technologie, použité při realizaci vlastního řešení. Všechny prostředky byly blíže popsány v samostatné kapitole.

Další část práce je zaměřena na analýzu informačních toků a procesu řízení projektů. V úvodu této části byla charakterizována společnost a její struktura. Následně byla provedena analýza procesu řízení projektů, na jejímž základě byly pomocí notace BPMN namodelovány procesní diagramy. Na základě současného stavu, byly navrženy a detailně popsány změny v procesu řízení projektů. Veškeré inovační kroky byly navrženy s ohledem na metodiku Scrum. Poté byl vytvořen nový procesní model části realizace projektů.

S ohledem na současný stav IS/ICT byl navržen prototyp webové aplikace zajišťující hlavní oblasti realizace projektů při dodržení metodiky Scrum. Návrhu aplikace předcházela analýza funkčních a nefunkčních požadavků. Funkční požadavky byly následně zobrazeny pomocí Use case diagramů. Struktura dat aplikace, včetně atributů a vazeb mezi jednotlivými entitami, byla znázorněna ve formě entitně relačního diagramu.

Součástí návrhu aplikace bylo vytvoření IFML modelů, které znázorňují strukturu a chování aplikace. Tyto modely byly vytvořeny pomocí nástroje WebRatio. Vytvořené modely se pak staly základem pro samotnou realizaci. Webová aplikace byla implementována pomocí jazyka PHP a Nette Frameworku. Výsledný prototyp aplikace byl prezentován a následně testován členy vývojového týmu ve společnosti. Na závěr práce byla diskutována možná vylepšení funkcionality. Veškeré připomínky a návrhy na rozšíření se stanou předmětem dalšího vývoje aplikace. Po úspěšných ohlasech ve společnosti lze konstatovat, že se podařilo splnit stanovený cíl práce a do budoucna lze v dané firmě předpokládat využití vytvořené aplikace.

## 9 Literatura

- ARLOW, J., NEUSTADT, I. *UML 2 a unifikovaný proces vývoje aplikací*. Praha: Computer Press, 2007. 568 s.
- BECK K., BEEDLE M. A KOL. *Manifest agilního vývoje software* [online]. 2001 [cit. 2016-02-20]. Dostupné z: <http://agilemanifesto.org/iso/cs/>
- BPMN Specification - Business Process Model and Notation* [online]. Object Management Group, Inc., 2016 [cit. 2016-04-10]. Dostupné z: <http://www.bpmn.org/>
- BRAMBILLA, M., BUTTI, S. *Fifteen Years of Industrial Model-Driven Development in Software Front-End: from WebML to WebRatio and IFML* [online]. Milán: Politecnico di Milano, 2013 [cit. 2016-04-09]. Dostupné z: <http://dbgroup.como.polimi.it/brambilla/sites/dbgroup.como.polimi.it.brambilla/files/referencematerials/WebML-IFML-WebRatio-Novatica-EN.pdf>.
- BUCHALCEVOVÁ ALENA. *Metodiky vývoje a údržby informačních systémů*. 1. vyd. Praha: Grada, 2005. 163 s. ISBN 80-247-1075-7.
- CARDA ANTONÍN, RENATA KUNSTOVÁ. *Workflow: nástroj manažera pro řízení podnikových procesů*. 2. rozš. a aktualiz. vyd. Praha: Grada, 2003, 155 s. ISBN 80-247-0666-0.
- ČSN EN ISO 9000:2016. *Technické normy* [online]. Praha, 2016 [cit. 2016-02-02]. Dostupné z: [http://www.iso-normy.cz/ISO\\_9000.html](http://www.iso-normy.cz/ISO_9000.html)
- Dokumentace [online]. 2016 [cit. 2016-03-11]. Dostupné z: <http://doc.nette.org/cs/getting-started>
- DOLEŽAL JAN, PAVEL MÁCHAL A BRANISLAV LACKO. *Projektový management podle IPMA*. 2., aktualiz. a dopl. vyd. Praha: Grada, 2012, 526 s. ISBN 978-80-247-4275-5.
- DUMAS, MARLON. *Fundamentals of business process management*. Heidelberg: Springer, c2013. ISBN 978-3-642-33142-8.
- DVOŘÁK DRAHOSLAV. *Řízení projektů: nejlepší praktiky s ukázkami v Microsoft Office*. Vyd. 1. Brno: Computer Press, 2008, 244 s. Business books. ISBN 978-80-251-1885-6.
- GILMORE JASON W. *Velká kniha PHP a MySQL 5: kompendium znalostí pro začátečníky i profesionály*. Vyd. 1. Brno: Zoner Press, 2007, 864 s. ISBN 80-868-15536.
- GORDON ANN. *What is a Work Breakdown Structure?* In: Bright Hub Project Management [online]. 2014 [cit. 2016-02-01]. Dostupné z: <http://www.brighthubpm.com/templates-forms/2645-what-is-a-work-breakdown-structure/>
- HAL MOOZ, KEVIN FORSBERG, HOWARD COTTERMAN. *Communicating Project Management*, Wiley & Sons, New Jersey, 2003. ISBN 0-471-26924-7.
- IFML: The Interaction Flow Modeling Language/The OMG standard for front-end design* [online]. Object Management Group, Inc., 2016 [cit. 2016-04-09]. Dostupné z: <http://www.ifml.org/>

- JANIŠOVÁ DANA, MIRKO KŘIVÁNEK. *Velká kniha o řízení firmy: [praktické postupy pro úspěšný rozvoj]*. 1. vyd. Praha: Grada, 2013, 394 s. ISBN 978-80-247-4337-0.
- KERZNER HAROLD. *Project Management, A Systems Approach to Planning, Scheduling, and Controlling*. Sixth Edition, Wiley & Sons, New York, 2013. ISBN 978-1-118-02227-6.
- Metoda kritické cesty – CPM (Critical Path Method)*. Management mania. [online]. 2015 [cit. 2016-02-01]. Dostupné z: <https://managementmania.com/cs/metoda-cpm>
- MURPHY CRAIG. *Adaptive Project Management Using Scrum*. In: *Methods & Tools* [online]. 2004 [cit. 2016-03-11]. Dostupné z: <http://www.methodsandtools.com/archive/archive.php?id=18>
- MVC aplikace & presentery*. Nette Foundation. Nette Framework: Dokumentace [online]. 2016 [cit. 2016-03-11]. Dostupné z: <http://doc.nette.org/cs/presenters>
- NARAMORE, E. A KOL. *Vytváříme webové aplikace v PHP5, MySQL a Apache*. 1. vyd. Brno: Computer Press, 2006. 813 s. ISBN 80-251-1073-7.
- OŠKRDAL VÁCLAV, DOUCEK PETR. *Praktické řízení ICT projektů*. Vydání první. Praha: Oeconomica, nakladatelství VŠE, 2014, 255 stran. ISBN 978-80-245-2073-5.
- PROCHÁZKA DAVID. *PHP 6: začínáme programovat*. 1. vyd. Praha: Grada, 2012, 183 s. Průvodce (Grada). ISBN 978-80-247-3899-4.
- PROJECT MANAGEMENT INSTITUTE. *A Guide to The Project Management Body of Knowledge*, IV. vydání, (PMBOOK Guide), 2008. ISBN 978-1933890517.
- PULKERT DALIBOR. *Průzkum agilního řízení v ČR 2013* [online]. Etnetera. [cit. 2016-02-22]. Dostupné z [http://www.etnetera.cz/public/1b/43/e5/52571\\_103079\\_agilni\\_dotaznik\\_report\\_2013\\_5.pdf](http://www.etnetera.cz/public/1b/43/e5/52571_103079_agilni_dotaznik_report_2013_5.pdf)
- Rapid mobile App and Web application development Platform*. In: WebRatio [online]. 2016 [cit. 2016-04-09]. Dostupné z: <http://www.webratio.com/site/content/en/home>
- Scrum Guide* [online]. Scrum.Org and ScrumInc, 2014 [cit. 2016-03-11]. Dostupné z: <http://www.scrumguides.org/scrum-guide.html>
- Seznámení s Nette Frameworkem*. Nette Foundation. Nette Framework:
- SCHWABER KEN. *Agile project management with scrum*. 2nd edition. Redmond, Wash.: Microsoft Press, 2004. ISBN 978-073-5696-938.
- SCHWALBE KATHY. *Řízení projektů v IT: kompletní průvodce*. Vyd. 1. Brno: Computer Press, 2011, 632 s. Business books. ISBN 978-80-2512882-4.
- SUTHERLAND, J., JAKOBSEN, C., JOHNSON, K.. *Scrum and CMMI Level 5: The Magic Potion for Code Warriors*. In *Proceedings of the Agile Conference (AGILE) 2007* (s. 272-278).
- SVOZILOVÁ ALENA. *Projektový management*. 2., aktualiz. a dopl. vyd. Praha: Grada, 2011, 380 s. ISBN 978-80-247-3611-2.

- ŠOCHOVÁ ZUZANA A EDUARD KUNCE. *Agilní metody řízení projektů*. 1. vyd. Brno: Computer Press, 2014, 175 s. ISBN 978-80-251-4194-6.
- VersionOne: 10th Annual State of Agile Survey* [online]. VersionOne Inc. [cit. 2016-02-22]. Dostupné z: <http://www.versionone.com/pdf/2015-state-of-agile-survey.pdf>
- WATERS KELLY. *All About Agile: Agile Management Made Easy!*. CreateSpace Independent Publishing Platform, 2012. ISBN 1469915510.
- WATERS KELLY. *How To Implement Scrum in 10 Easy Steps*. In: All About Agile [online]. 2007 [cit. 2016-03-11]. Dostupné z: <http://www.allaboutagile.com/how-to-implement-scrum-in-10-easy-steps/>
- WAZLAWICK, R. *Object-oriented analysis and design for information systems: modeling with UML, OCL, and IFML*. Waltham: Morgan Kaufmann Publisher, 2014, 376 p. ISBN 978-012-4186-736.

## 10 Seznam obrázků

Obr. 1	Projektový trojimperativ Zdroj: Drahoslav Dvořák, 2008	15
Obr. 2	Workflow systém Zdroj: Carda, Kunstová, 2003	19
Obr. 3	Typické rozložení fází životního cyklu projektu Zdroj: Alena Svozilová, 2011	20
Obr. 4	Schéma průběhu Zdroj: Přeloženo z Murphy, 2004	26
Obr. 5	Struktura organizace	35
Obr. 6	Struktura procesu projektového řízení	35
Obr. 7	Subproces Specifikace požadavků	36
Obr. 8	Subproces Uzavření smlouvy	37
Obr. 9	Subproces Realizace	38
Obr. 10	Subproces Předání zákazníkovi	39
Obr. 11	Subproces Poimplementační podpora	40
Obr. 12	Subproces Realizace	43
Obr. 13	Uživatelé	45
Obr. 14	Use case diagram Uživatel	45
Obr. 15	Use case diagram Vývojář	46
Obr. 16	Use case diagram Scrum Master	47
Obr. 17	Use case diagram Product Owner	48
Obr. 18	Entitně relační diagram	52
Obr. 19	IFML model pro nepřihlášeného uživatele	54
Obr. 20	IFML model pro vývojáře	55
Obr. 21	IFML model pro uživatele Product Owner	57
Obr. 22	IFML model pro uživatele Scrum Master	59

---

<b>Obr. 23</b>	<b>Product Backlog</b>	<b>61</b>
<b>Obr. 24</b>	<b>Aktuální sprint</b>	<b>62</b>
<b>Obr. 25</b>	<b>Formulář pro nový sprint</b>	<b>62</b>
<b>Obr. 26</b>	<b>Subproces Specifikace požadavků</b>	<b>74</b>
<b>Obr. 27</b>	<b>Subproces Uzavření smlouvy</b>	<b>75</b>
<b>Obr. 28</b>	<b>Subproces Realizace</b>	<b>76</b>
<b>Obr. 29</b>	<b>Subproces Předání zákazníkovi</b>	<b>77</b>
<b>Obr. 30</b>	<b>Subproces Poimplementační podpora</b>	<b>78</b>
<b>Obr. 31</b>	<b>Návrh změn v procesu Realizace</b>	<b>79</b>
<b>Obr. 32</b>	<b>Product Backlog</b>	<b>80</b>
<b>Obr. 33</b>	<b>Aktuální sprint</b>	<b>81</b>
<b>Obr. 34</b>	<b>Formulář pro nový sprint</b>	<b>82</b>
<b>Obr. 35</b>	<b>Přehled vývojového týmu</b>	<b>83</b>
<b>Obr. 36</b>	<b>Docházka</b>	<b>84</b>

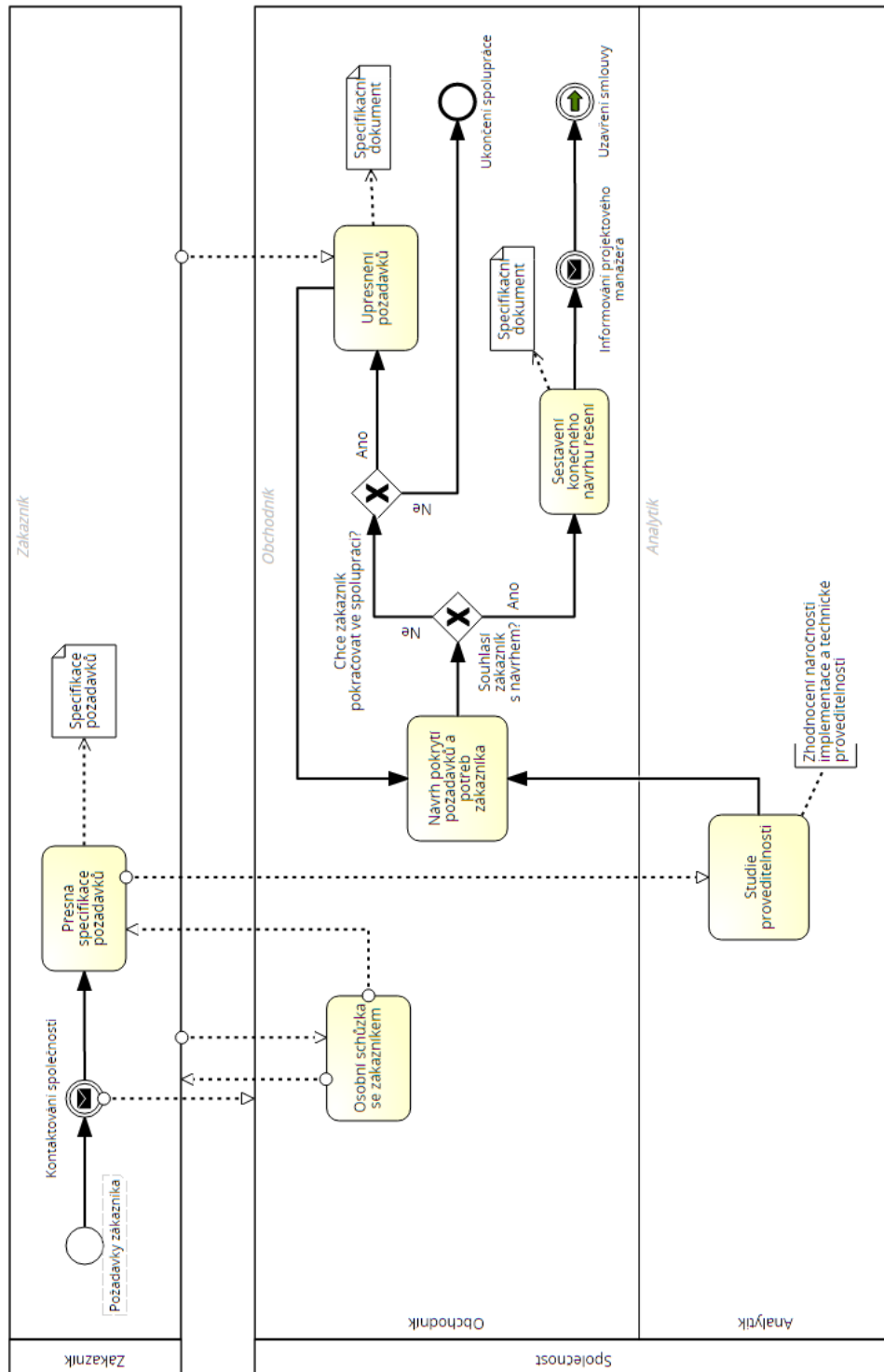
## 11 Seznam tabulek

<b>Tab. 1</b>	<b>Přehled elementů IFML</b>	<b>30</b>
<b>Tab. 2</b>	<b>Scénář případu užití Přihlášení uživatele do systému</b>	<b>50</b>
<b>Tab. 3</b>	<b>Scénář případu užití Editace priority úkolu</b>	<b>50</b>
<b>Tab. 4</b>	<b>Scénář případu užití Zařazení úkolu do Sprintu</b>	<b>51</b>
<b>Tab. 5</b>	<b>Scénář případu užití Export do .pdf</b>	<b>51</b>
<b>Tab. 6</b>	<b>Testovací přihlašovací údaje pro uživatele</b>	<b>61</b>

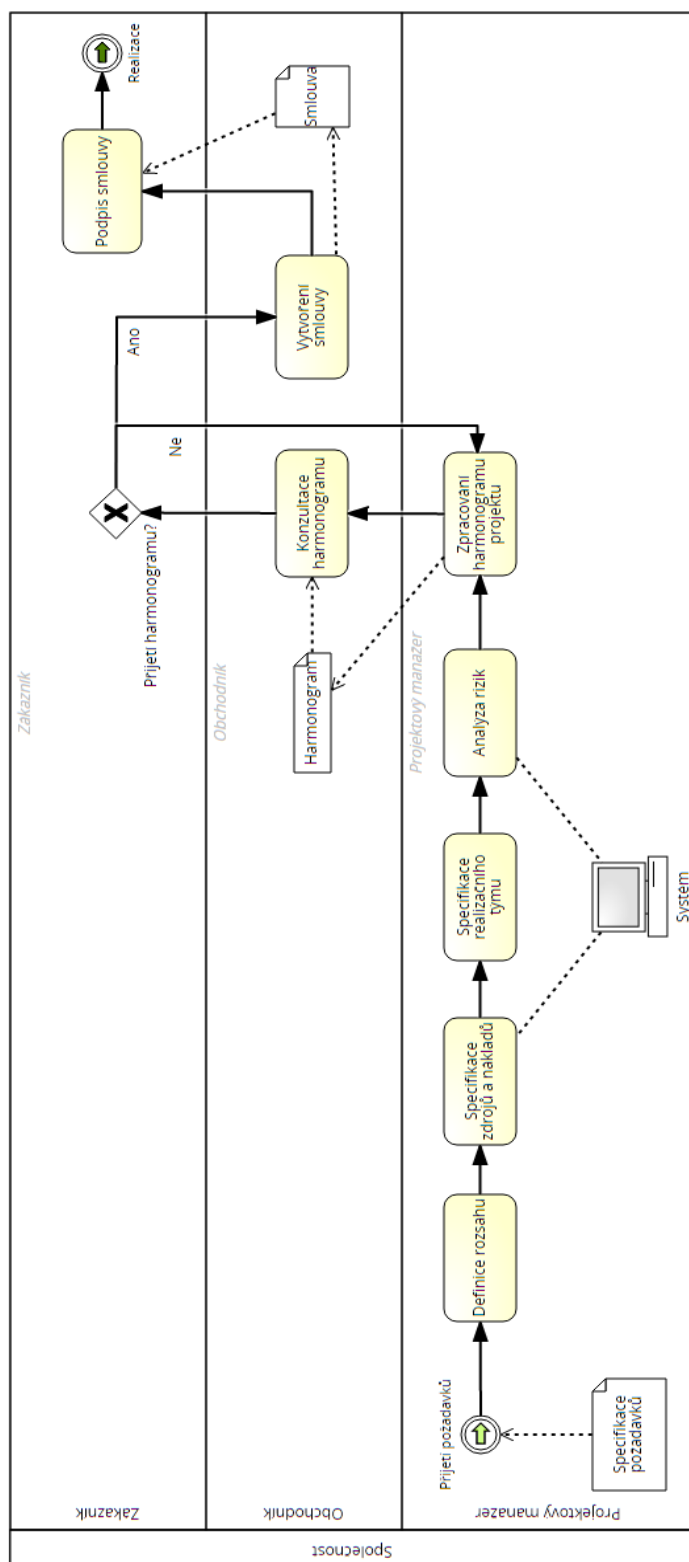


# **Přílohy**

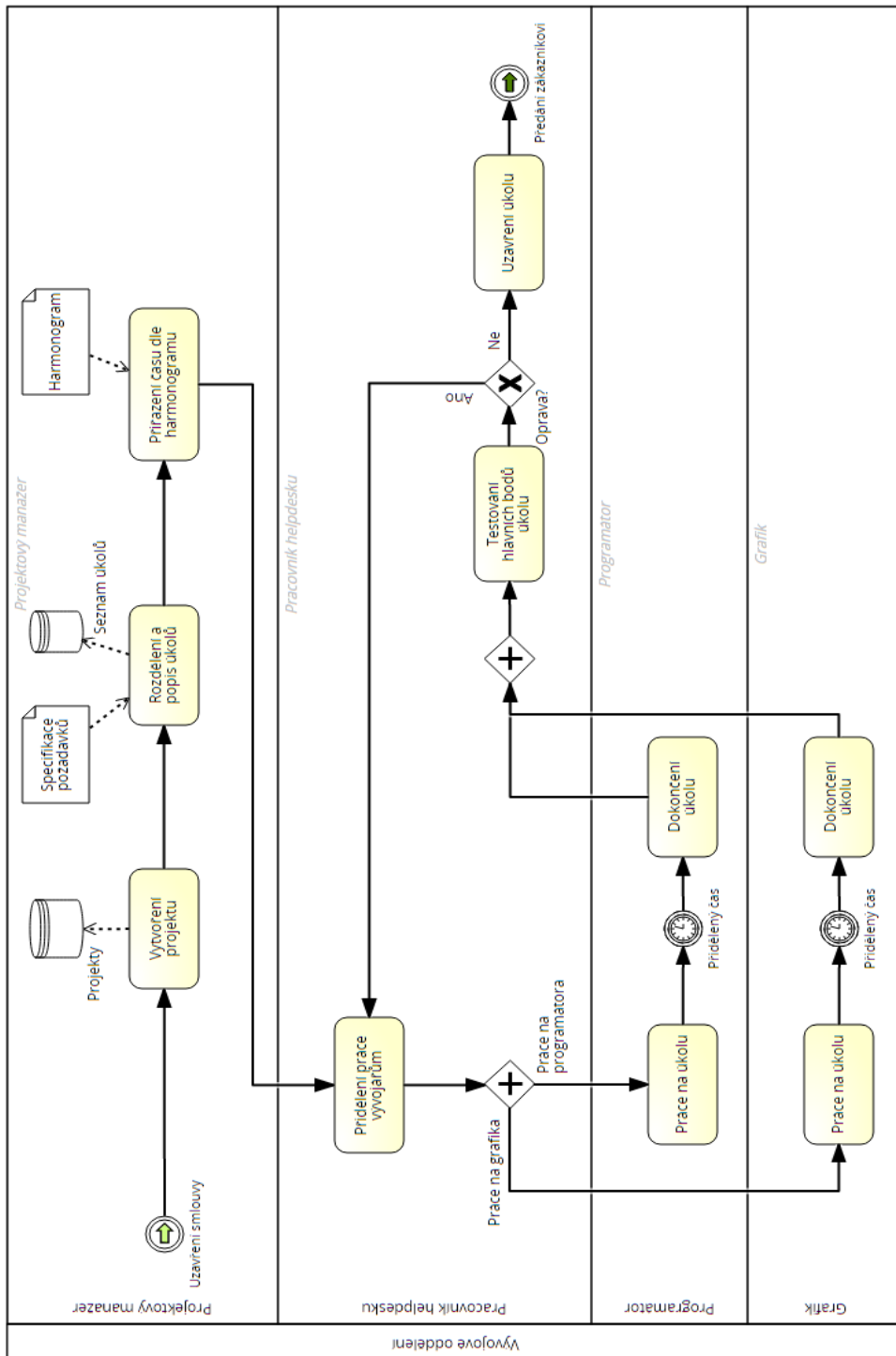
# A Procesní diagramy



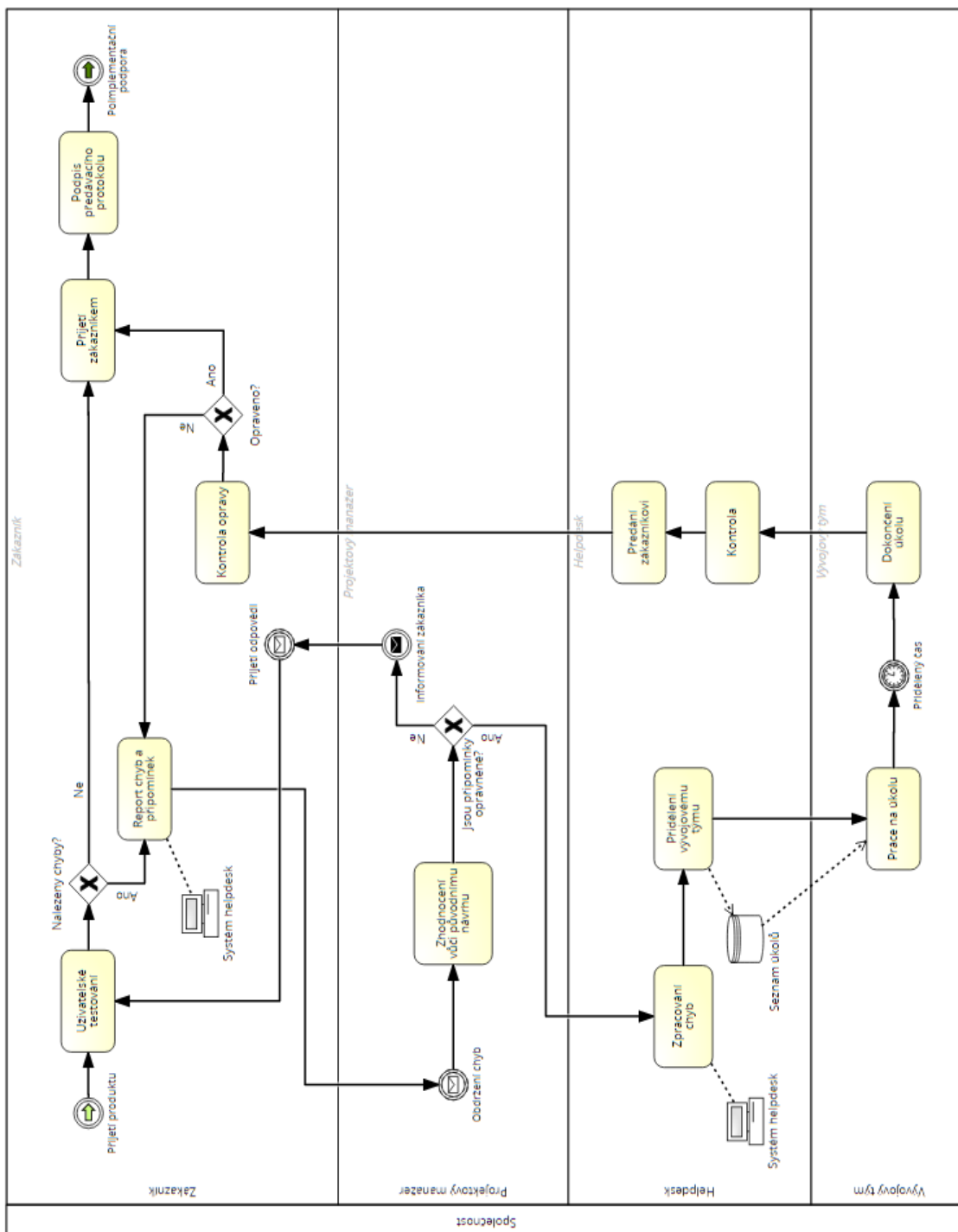
Obr. 26 Subproces Specifikace požadavků



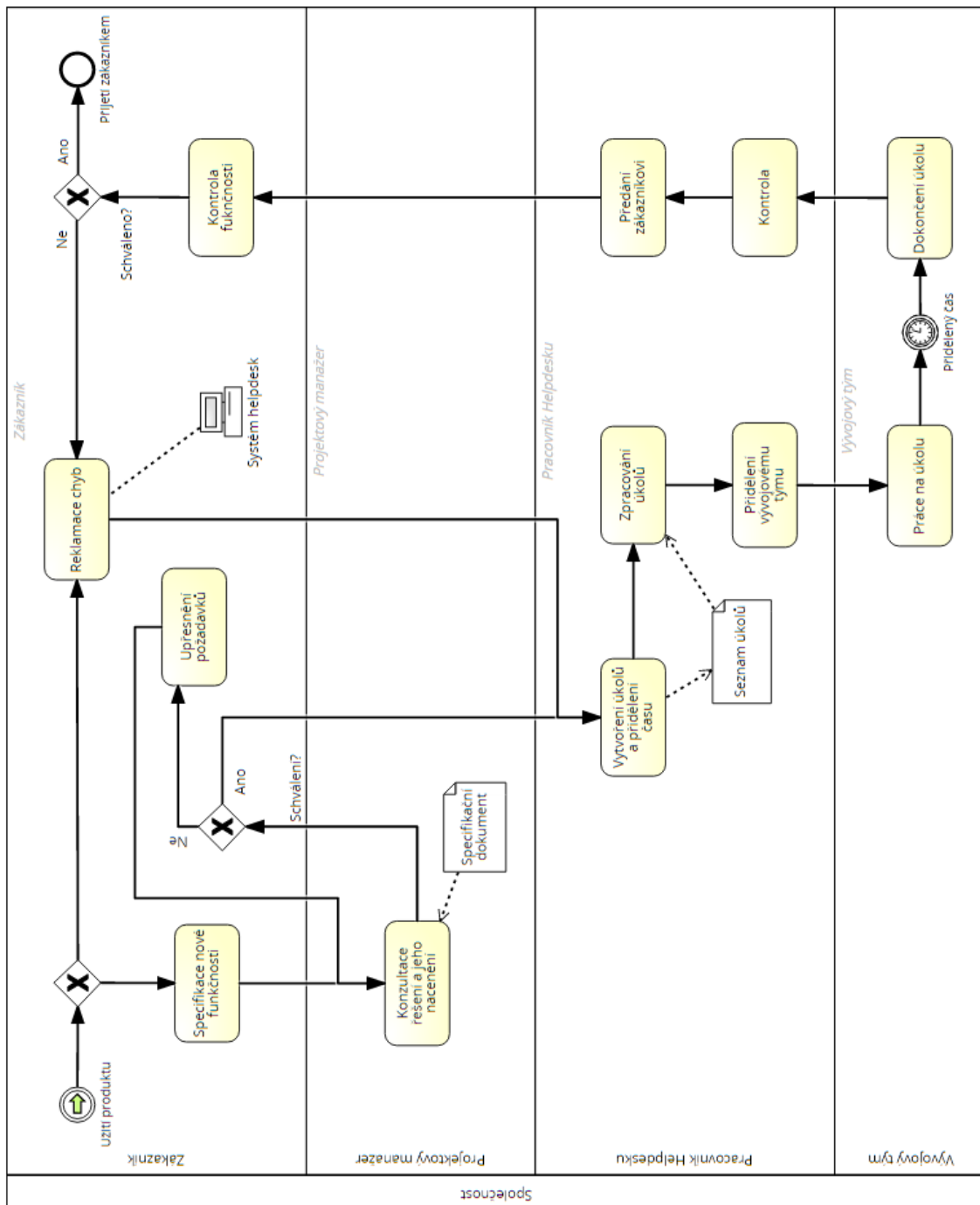
Obr. 27 Subproces Uzavření smlouvy



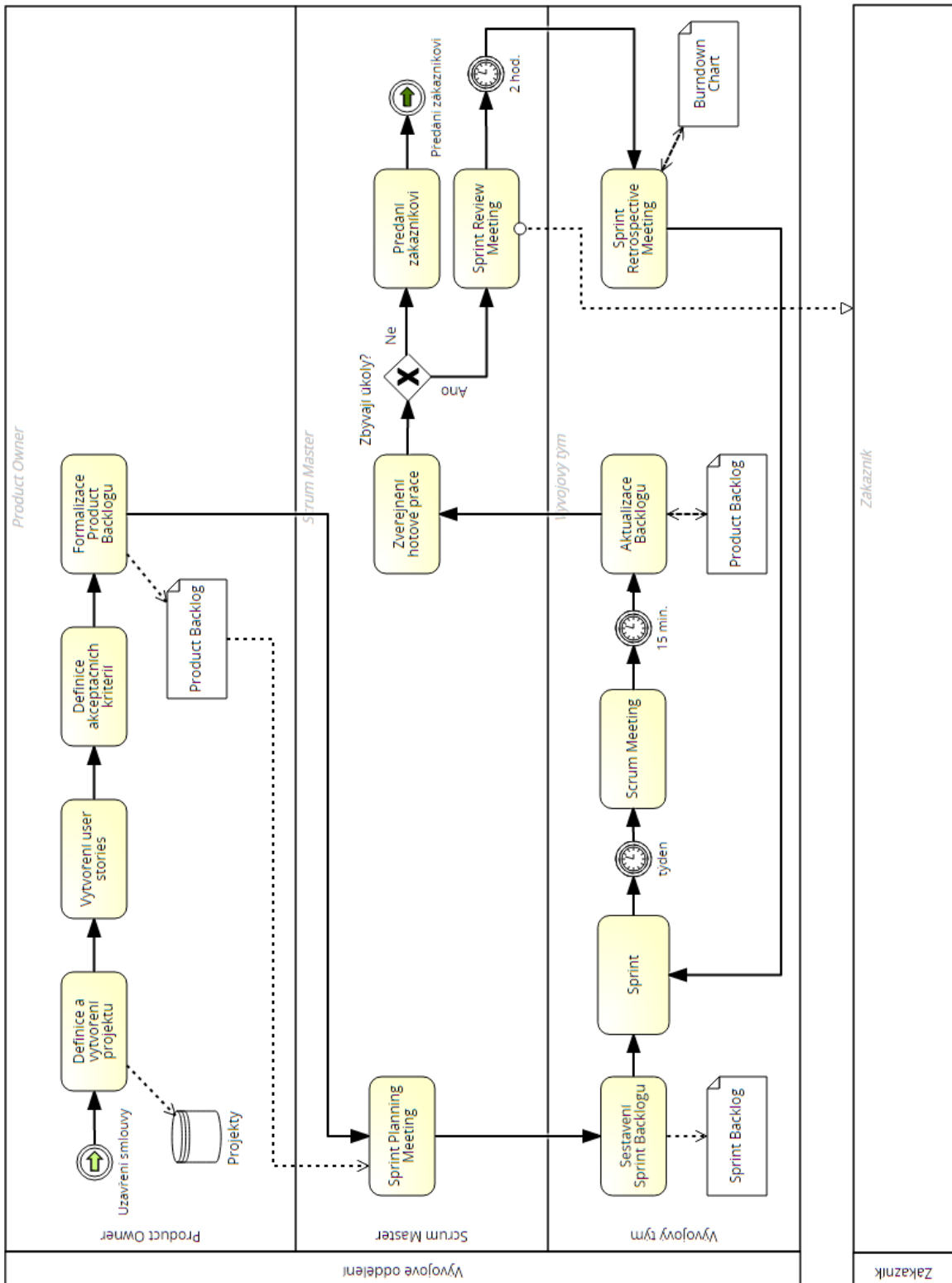
Obr. 28 Subproces Realizace



Obr. 29 Subproces Předání zákazníkovi



Obr. 30 Subproces Poimplementační podpora



Obr. 31 Návrh změn v procesu Realizace

## B Ukázky aplikace

Product Backlog
Sprint
Moje úkoly
Docházka
Vývojový tým
Přihlášená Veronika Prokopová
Odhlásit

Filter

+	Číslo úkolu	Projekt	Typ	Stav	Přirazen	Název	Vytvořeno	Čas (h)	Opracováno (h)	A
+	1	Projekt 1	bug	Test	Veronika Prokopová	Cookie líšta	22.4.2016	1.00	2.50	<a href="#">/</a>
+	Detail sprintu									
+	7777	Projekt 1	objednávka	In progress	Veronika Prokopová	Modul doprava zdarma	28.4.2016	3.50	0.00	<a href="#">/</a>
+	Detail sprintu									
+	88364	Projekt 1	bug	To do	Jan Novák	Nefunkční konektor money55	22.4.2016	2.00	1.00	<a href="#">/</a>
+	Detail sprintu									
+	100	Projekt 1	bug	In progress	Veronika Prokopová	Nejřemesně objednávky	22.4.2016	1.00	4.50	<a href="#">/</a>
+	Detail sprintu									
+	89745	Projekt 1	bug	In progress	Veronika Prokopová	Násazení modulu hromadná správa dat	22.4.2016	2.00	1.00	<a href="#">/</a>
+	Detail sprintu									
+	333	Projekt 1	objednávka	To do	Petr Klisáček	Mapa stránek	22.4.2016	2.00	0.00	<a href="#">/</a>
+	Detail sprintu									
+	25874	Projekt 1	bug	Test	Kateřina Kovalčíková	Implementace platby PayU	22.4.2016	6.00	1.00	<a href="#">/</a>
+	Detail sprintu									

Obr. 32 Product Backlog



Product Backlog | **Sprint** | Moje úkoly | Docházka | Vývojový tým | Přihlášen: Veronika Prokopová | Odejít

### Aktuální sprint

Filter

-Vyber tým- Filter Reset

Číslo sprintu	Název	Popis	Vytvořil	Vytvořeno	Datum ukončení	A
2	Sprint 2	sprint2	Jan Novák	3.5.2016	10.5.2016	

Sprint Backlog

Číslo tasku	Projekt	Typ	Stav	Přirazen	Název	A
25874	Projekt 1	bug	Test	Katerina Kovaliková	Implementace platby PayU	

Obr. 33 Aktuální sprint

Product Backlog
Sprint
Moje úkoly
Docházka
Vývojový tým
Přihášen: Veronika Prokopová
Odhlásit

### Přidání nového sprintu

Název

Číslo sprintu

Popis:

Tým  Uložit

<input type="checkbox"/>	Číslo úkolu	Projekt	Typ	Stav	Název	Vytvořeno	Čas (h)	Odpracováno (h)	Přirazen	A
<input type="checkbox"/>	88364	Projekt 1	bug	To do	Nefunkční konektor moneys5	22.4.2016	2,00	1,00	Jan Novák	<input type="button" value=""/>
<input checked="" type="checkbox"/>	25874	Projekt 1	bug	Test	Implementace platby PayU	22.4.2016	6,00	1,00	Kateřina Kovalčíková	<input type="button" value=""/>
<input type="checkbox"/>	89745	Projekt 1	bug	In progress	Nasazení modulu hromadná správa dat	22.4.2016	2,00	1,00	Veronika Prokopová	<input type="button" value=""/>
<input type="checkbox"/>	333	Projekt 1	objednávka	To do	Mapa stránek	22.4.2016	2,00	0,00	Petr Klusáček	<input type="button" value=""/>
<input type="checkbox"/>	1	Projekt 1	bug	Test	Cookie lista	22.4.2016	1,00	2,50	Veronika Prokopová	<input type="button" value=""/>
<input type="checkbox"/>	100	Projekt 1	bug	In progress	Nepřenesené objednávky	22.4.2016	1,00	4,50	Veronika Prokopová	<input type="button" value=""/>
<input type="checkbox"/>	7777	Projekt 1	objednávka	In progress	Modul doprava zdarma	28.4.2016	3,50	0,00	Veronika Prokopová	<input type="button" value=""/>

Výbráné ▾

Obr. 34 Formulář pro nový sprint

Product Backlog   Sprint   **Moje úkoly**   Docházka   **Vývojový tým**   Přihlášen: Veronika Prokopová | Odehlásit

Filter

-Vyber tým- ▾   Filter   Reset

A	Login	Jméno	E-mail	Poslední přihlášení	Tým
<input checked="" type="checkbox"/>	v.prokopova	Veronika Prokopová	v.prokopova@test.cz	8.5.2016 21:17:57	Tým1
<input checked="" type="checkbox"/>	k.kovalcikova	Kateřina Kovalčíková	k.kovalcikova@test.cz	1.4.2016 16:32:18	Tým1
<input checked="" type="checkbox"/>	j.novak	Jan Novák	jan.novak@test.cz	8.5.2016 17:17:13	Tým1
<input checked="" type="checkbox"/>	p.klusacek	Petr Klusáček	petr.klusacek@test.cz	8.5.2016 08:01:10	Tým2
<input type="checkbox"/>	p.herout	Pavel Herout	p.herout@test.cz	2.10.2015 06:52:46	Tým2
<input checked="" type="checkbox"/>	p.herout	Pavel Herout	p.herout@test.cz	9.5.2016 18:27:09	Tým2

Obr. 35   Přehled vývojového týmu

Ochlásit
Přihlášen: Veronika Prokopová

Produkt Backlog
Sprint
Moje úkoly
Docházka
Vývojový tým

Přidat nový záznam
Moje docházka

### Souhrn

Filtr
 

Datum ▾ Vymazat

Jméno	Datum	Opracováno
Veronika Prokopová	22.4.2016	2.00
Veronika Prokopová	28.4.2016	0.75

### Moje docházka

Úkol	Vytvořeno	Datum	Opracováno	Popis	A
1 [Coobie lista]	23.4.2016 23:26:20	22.4.2016	2.00	commit configu, úprava šablón	<span style="color: red; font-size: small;">✖</span> <span style="color: blue; font-size: small;">✎</span>
7777 [Modul doprava zdarma]	28.4.2016 12:43:16	28.4.2016	0.75	studování dokumentace, příprava modulu	<span style="color: red; font-size: small;">✖</span> <span style="color: blue; font-size: small;">✎</span>

Obr. 36 Docházka