

**Vysoká škola ekonomická v Praze**

**Fakulta informatiky a statistiky**

**Katedra informačních technologií**

Studijní program : Aplikovaná informatika

Obor: Informační systémy a technologie

# **Aplikace Best Practices agilních metodik do PRINCE 2**

**DIPLOMOVÁ PRÁCE**

Student : Bc. Petr Šíma

Vedoucí : Ing. Drahomír Chocholatý, MBA

Oponent : Ing. Petr Kínl

**2016**

## **Prohlášení:**

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a že jsem uvedl všechny použité prameny a literaturu, ze které jsem čerpal.

V Praze dne 23. dubna 2016

.....

Petr Šíma

## **Poděkování**

Chtěl bych poděkovat vedoucímu této práce panu Ing. Drahomíru Chocholatému, MBA za vedení této práce a za cenné rady, připomínky a inspiraci.

Dále bych rád chtěl poděkovat všem kolegům, jejichž rady a zkušenosti mi pomohly při tvorbě části této práce.

Nakonec bych chtěl poděkovat rodině a přátelům za podporu v době mého studia.

## **Abstrakt**

Předmětem této práce je definice Best Practices agilních metodologií, jejich aplikace na nejčastěji se vyskytující problémy projektového managementu a jejich integrace do PRINCE2. Práce se bude zaměřovat především na jednotlivé aplikaci agilní praktiky, nikoli na celé metodologie.

Tato diplomová práce bude rozdělena na dvě hlavní části. V první části budou představeny základy metodologie PRINCE2, vybrané praktiky agilních metodologií a samotný pojem Best Practices. V druhé části práce budou definovány nejčastěji se vyskytující problémy projektového managementu. Poté budou jednotlivé Best Practices aplikovány na definované problémy a na různé aspekty PRINCE2.

Nakonec budou vyhodnoceny nejpoužitelnější Best Practices na základě výsledků výzkumu.

## **Klíčová slova**

PRINCE2, projektový management, agilní metodologie, Best Practices, problémy projektového managementu, agilní vývoj

## **Abstract**

This thesis is focused on Best Practices of agile methodologies, their application to the most common problems of project management and their integration into PRINCE2. The thesis will be focused mainly on the application of individual agile practices, not on the whole methodologies.

This thesis will be divided into two main parts. In the first part PRINCE2 methodology basics, chosen agile methodology practices and Best Practices itself will be introduced. In the second part the most common problems of project management will be defined. Then every Best Practice will be applied to the defined problems and various aspects of PRINCE2.

In the end the most usable Best Practices based on the results of research will be evaluated.

## **Keywords**

PRINCE2, project management, agile methodologies, Best Practices, problems of project management, agile development

# Obsah

|          |                                                                                                        |           |
|----------|--------------------------------------------------------------------------------------------------------|-----------|
| <b>1</b> | <b>Úvod.....</b>                                                                                       | <b>6</b>  |
| 1.1      | Cíle práce.....                                                                                        | 6         |
| 1.2      | Použité metody a postupy.....                                                                          | 6         |
| 1.3      | Rešerše.....                                                                                           | 7         |
| <b>2</b> | <b>Teoretické základy práce.....</b>                                                                   | <b>8</b>  |
| 2.1      | PRINCE2 .....                                                                                          | 8         |
| 2.1.1    | Základní charakteristiky metodologie .....                                                             | 8         |
| 2.1.2    | Principy PRINCE2 .....                                                                                 | 13        |
| 2.1.3    | Témata PRINCE2 .....                                                                                   | 19        |
| 2.1.4    | Procesy PRINCE2.....                                                                                   | 34        |
| 2.2      | Agilní metodologie.....                                                                                | 45        |
| 2.2.1    | Historie agilních metodologií.....                                                                     | 46        |
| 2.2.2    | Agilní manifest .....                                                                                  | 46        |
| 2.2.3    | Výběr Best Practices agilních metodologií.....                                                         | 52        |
| 2.2.4    | Popis vybraných Best Practices.....                                                                    | 53        |
| 2.3      | Best Practices .....                                                                                   | 61        |
| 2.3.1    | Definice Best Practices .....                                                                          | 61        |
| 2.3.2    | Řízení Best Practices.....                                                                             | 62        |
| 2.3.3    | Proč vyhledávat a implementovat Best Practices.....                                                    | 62        |
| 2.3.4    | Vztah Best Practices a znalostí.....                                                                   | 63        |
| <b>3</b> | <b>Integrace Best Practices do PRINCE2 a nejčastějších problémů<br/>projektového managementu .....</b> | <b>65</b> |
| 3.1      | Definice bodů .....                                                                                    | 65        |
| 3.2      | Definice nejčastěji se objevujících problémů.....                                                      | 65        |
| 3.3      | Best Practices a konkrétní problémy .....                                                              | 70        |
| 3.3.1    | Neporozumění zadání .....                                                                              | 70        |
| 3.3.2    | Konflikt v týmu .....                                                                                  | 71        |
| 3.3.3    | Zákazník není přístupný kompromisům nebo nepochopil své<br>povinnosti.....                             | 72        |
| 3.3.4    | Člen týmu není tolik výkonný (nechápe účel projektu).....                                              | 73        |
| 3.3.5    | Realizace příliš vysokého počtu projektů.....                                                          | 74        |
| 3.3.6    | Špatná komunikace v rámci týmu .....                                                                   | 74        |
| 3.3.7    | Špatná komunikace směrem k zákazníkovi.....                                                            | 75        |
| 3.3.8    | Vágně definovaný rozsah.....                                                                           | 76        |
| 3.3.9    | Příliš často měněný rozsah .....                                                                       | 76        |
| 3.3.10   | Nízká flexibilita .....                                                                                | 77        |
| 3.3.11   | Špatné odhady pracnosti .....                                                                          | 77        |

|        |                                                 |    |
|--------|-------------------------------------------------|----|
| 3.3.12 | Příliš detailní a starostlivé řízení.....       | 78 |
| 3.3.13 | Přílišné spoléhání na nástroje .....            | 78 |
| 3.3.14 | Nedefinované (či málo definované) metriky ..... | 79 |
| 3.3.15 | Nedostatečné kapacity .....                     | 79 |
| 3.3.16 | Nedostatek kvalifikovaných zaměstnanců.....     | 79 |
| 3.3.17 | Konflikt priorit .....                          | 80 |
| 3.3.18 | Nejasně definované priority .....               | 80 |
| 3.3.19 | Nespolehlivý člen týmu .....                    | 80 |
| 3.3.20 | Velké množství připomínek a chyb.....           | 80 |
| 3.4    | Principy PRINCE2 .....                          | 81 |
| 3.4.1  | Neustálé zdůvodňování opodstatněnosti (1) ..... | 81 |
| 3.4.2  | Učení se ze zkušenosti (2).....                 | 82 |
| 3.4.3  | Definované role a odpovědnosti (3) .....        | 82 |
| 3.4.4  | Řízení po etapách (4) .....                     | 83 |
| 3.4.5  | Řízení výjimkami (5).....                       | 83 |
| 3.4.6  | Zaměření se na produkty (6) .....               | 84 |
| 3.4.7  | Uzpůsobení řízení prostředí projektu (7).....   | 85 |
| 3.5    | Témata PRINCE2.....                             | 85 |
| 3.5.1  | Obchodní případ (1).....                        | 85 |
| 3.5.2  | Organizace (2) .....                            | 86 |
| 3.5.3  | Kvalita (3).....                                | 86 |
| 3.5.4  | Plány (4).....                                  | 87 |
| 3.5.5  | Riziko (5).....                                 | 87 |
| 3.5.6  | Změna (6).....                                  | 88 |
| 3.5.7  | Průběh (7) .....                                | 88 |
| 3.6    | Procesy PRINCE2 .....                           | 88 |
| 3.6.1  | Proces zahájení projektu (1).....               | 88 |
| 3.6.2  | Proces řízení projektu (2) .....                | 89 |
| 3.6.3  | Proces nastavení projektu (3) .....             | 89 |
| 3.6.4  | Proces kontroly etapy (4) .....                 | 89 |
| 3.6.5  | Proces řízení dodání produktu (5) .....         | 89 |
| 3.6.6  | Proces řízení přechodu mezi etapami (6).....    | 90 |
| 3.6.7  | Proces ukončení projektu (7).....               | 90 |
| 3.7    | Výsledné tabulky.....                           | 91 |
| 4      | Závěr.....                                      | 95 |
|        | Seznam literatury.....                          | 96 |
|        | Seznam obrázků a tabulek.....                   | 98 |
|        | Seznam obrázků.....                             | 98 |
|        | Seznam tabulek .....                            | 98 |

# 1 Úvod

V dnešní době, kdy téměř každá společnost, která se zabývá vývojem software (ať se již jedná o malé, střední nebo velké společnosti) je kladen čím dále větší důraz na efektivnější řízení projektů. Existuje celá řada metodologií a přístupů, které lze použít v celé řadě projektů. Bohužel mnoho projektových manažerů má špatné zkušenosti obzvláště s agilními metodologiemi (z důvodu špatného nasazení, nebo nepochopení agilních principů), a tudíž se brání jejich implementaci. Každá metodologie totiž nemusí být ta správná pro danou společnost, ať již z důvodů kultury nebo jiné zavedené metodologie. Tato práce se proto zaměřuje na Best Practices vybraných agilních metodologií, které mohou být implementovány jednotlivě a nejsou závislé na metodologii, ze které pochází.

## 1.1 Cíle práce

Hlavními cíli této práce je identifikace nejvhodnějších Best Practices pro integraci do PRINCE2. Dále budou identifikovány nejčastější problémy při řízení projektů a jejich možné řešení pomocí Best Practices z vybraných agilních metodologií. Nakonec bude určena míra použitelnosti jednotlivých Best Practices v rámci metodologie PRINCE2 na základě četnosti jejich využitelnosti.

## 1.2 Použité metody a postupy

Cílů této práce bude dosaženo následujícími způsoby. Nejdříve bude objasněn pojem Best Practices. Poté budou definovány veškeré praktiky vybraných agilních metodologií z vybrané studie přední společnosti zabývající se touto tematikou. Na základě zkušeností autora a jeho kolegů v oblasti projektového managementu budou vymezeny nejčastěji problémy, které se objevují v rámci řízení IT projektů. Na tyto problémy budou aplikovány jednotlivé praktiky, aby bylo zjištěno, zda je lze použít k jejich prevenci nebo odstranění. Následně budou tyto praktiky integrovány do každého aspektu metodologie PRINCE2. Na konci práce budou na základě zjištěných údajů vyhodnoceny nejpoužitelnější agilní praktiky v projektovém managementu.

## 1.3 Rešerše

Níže jsou uvedeny práce, které se zabývají stejným tématem, avšak používají jiné metody a zaměřují se spíše na celé metodologie než na jednotlivé praktiky. Dále se věnují spíše porovnávání agilních a tradičních přístupů.

- **Projektové řízení v konkrétní společnosti – agilní vs. tradiční přístupy** – autorka této diplomové práce se zaměřuje na porovnání agilních metodologií a tradičních přístupů k projektovému řešení. Práce nebere v potaz užití jednotlivých agilních praktik, ale porovnání agilních metodologií jako celku. Výsledkem této práce je zjištění, že agilní metodologie se lépe hodí pro řízení projektu. (HONSOVÁ;2013)
- **Moderní metodiky vedení projektů (pro implementace zakázkového SW)** – Práce se zaměřuje na problematiku projektového řízení v oblasti zakázkového vývoje software. Co se agilních metodologií týče, tak je úspěšně nahrazen pouze proces PRINCE2 řízení projektu agilní metodologií Scrumban. (ZATLOUKAL;2014)
- **Zavedení agilního přístupu projektového managementu ve vybrané firmě** – Práce je zaměřena na implementaci agilní metodologie Scrum a standard PMBoK. Práce se nezaměřuje na individuální praktiky, ale na celkovou metodologii a používá i odlišný standard projektového managementu než tato práce. Výsledkem je metodologie postavená na základě výše uvedených metodologií., použita pro řízení projektů v malé společnosti. (BARTOVIČOVÁ;2014)

## 2 Teoretické základy práce

V této kapitole jsou rozepsány veškeré teoretické základy, které jsou potřeba k pochopení základních oblastí projektového managementu (zejména metodiky PRINCE2 a agilních metodologií) a ze které bude vycházet vlastní výzkum v praktické části práce. Nejdříve budou popsány základní aspekty PRINCE2. V další podkapitole bude stručně rozebrána historie agilních metodologií a Agilní manifest. Následně budou rozepsány jednotlivé vybrané praktiky agilních metodologií, které budou vybrány na základě studie Versionone. Nakonec bude přiblížen samotný pojem Best Practices.

### 2.1 PRINCE2

PRINCE2 (**P**ROjects **I**n **C**ontrolled **E**nvironment – v překladu „projekty v kontrolovaném prostředí“) je jednou z nejrozšířenějších projektových metodologií ve světě. Zastřešuje řízení, kontrolování a organizování projektu. Samotnou metodologii spravuje Cabinet Office (dříve OGC -Office of Government Commerce).

(MURRAY;2009)

#### 2.1.1 Základní charakteristiky metodologie

PRINCE2 je metodologií, která byla vytvořena britskou vládou v roce 1996 a postupně se stala nejvyužívanější projektovou metodologií na celém světě. Např. ke konci roku 2007 bylo zaznamenáno více jak 250 000 projektových manažerů, kteří vlastnili certifikaci PRINCE2 (Practitioner). Samotná metodologie je spravována Cabinet office (dříve OGC - Office of Government Commerce).

(MURRAY;2009)

#### Certifikace

Aby projektový manažer (potažmo společnost, která se uchází o zakázku při výběrovém řízení, kde je jedna z podmínek certifikovaný projektový manažer) prokázal schopnost řídit projekty pomocí metodologie PRINCE2, je možnost získat příslušnou certifikaci. V závislosti na úrovni projektového manažera se nabízejí 3 typy certifikace. Pro získání nejvyšší certifikace je nutné získat i certifikace nižší úrovně. Konkrétně se jedná o:

- **PRINCE2 Foundation** – základní certifikace, která znamená, že její držitel rozumí principům a terminologii PRINCE2 a může vystupovat jako informovaný člen pro-

jektového týmu. Tato základní certifikace nemá žádné předepsané předpoklady pro plnění zkoušky.

- **PRINCE2 Practitioner** – pokročilá certifikace, jejímž cílem je naučit uživatele za pomoci PRINCE2 řídit a spravovat projekt v daném prostředí. Po získání certifikace dokáže projektový manažer aplikovat PRINCE2 do projektového prostředí a uzpůsobit jej potřebám specifického projektu (či problému). Plně chápe vztahy mezi jednotlivými procesy, tématy a principy. Je zapotřebí mít předchozí certifikaci a doložit praxi v oboru.
- **PRINCE2 Profesional** – nejvyšší certifikace PRINCE2. Držitel dokonale ovládá metodologii a její aplikaci na specifické projekty. Zkouška na rozdíl od předchozích je ryze praktická. Je zapotřebí předchozí certifikace a doložit praxi.

(MURRAY;2009)

## Projekt

Projekt je definován jako dočasná organizace, která je vytvořena na určité časové období za účelem dodání jednoho či více produktů na základě definovaného obchodního případu.

Práce na projektu je v porovnání s klasickou rutinní činností charakterizována následujícími znaky:

- **Změna** – projekt se zavádí zpravidla kvůli zavedení nějaké změny, ať se jedná o změnu či implementaci procesu (či softwaru podporující onen proces) interně, nebo u zákazníka.
- **Dočasnost** - projekty jsou již ze své podstaty časově omezeny. Jakmile je požadovaná změna implementována, projekt se ukončuje a projektový tým se rozpouští (z pohledu aktuálního projektu) a začíná normální rutinní provoz (používá se anglický výraz „*business as usual*“) ve své nové podobě. Každý projekt musí mít nadefinovaný počátek a konec.
- **Funkčnost napříč celou organizací** – na začátku startu projektu se tvoří různé týmy, které budou pracovat na dílčích částech projektu. Je nezbytné sestavit funkční týmy podle specializací (např. testerský tým, programátorský tým atp.), protože pro úspěšné dodání produktů je zapotřebí mnoha lidí s různými schopnostmi. Je proto nutné sestavit tým mimo klasickou organizační strukturu, která je ve společnosti implementována. V podstatě tak vzniká naprosto nová organizace, která na konci projektu opět zanikne (za předpokladu, že ty samé týmy nepracují na jiných projektech). Toto může občas vést k problémům uvnitř nově vzniklé organizace, nebo směrem k zákazníkovi. Každý má totiž jinou motivaci k projektu (pokud jde např. o interní projekt zavedení nového účetního systému, bude mít největší motivaci tým složený právě z účetního oddělení).

- **Jedinečnost** – každý projekt je jedinečný. Nikdy se nestane, že by dva projekty byly naprosto identické. Každý z nich bude vždy v něčem unikátní. I kdyby se nasažoval nový účetní systém do podobného podniku se stejnými zaměstnanci a stejnou kulturou, ani tak se nemůže stát, že by byly oba identické. Projekt potkají jiné problémy a příležitosti (např. výpadky elektřiny, nemoc členů týmu, implementace jiného navazujícího systému, jiné požadavky na změnu atd.).
- **Nejistota** – každý projekt provází různá míra nejistoty, která je vyšší za rutinního provozu. Projektový manažer musí být připraven na veškerá rizika a musí na ně umět zareagovat. Také musí být schopen rozpoznat příležitosti a umět je využít k zefektivnění projektu.

(MURRAY;2009), (ESKEROD;2013)

### Projektový manažer

Jednou z ústředních postav celé dočasné organizace, která vzniká za účelem dodání produktu, je projektový manažer. Je to role, která určuje, co se bude dělat, dohlíží na plánování a alokaci zdrojů. Pokud by cílem projektu bylo postavit dům, projektový manažer je tím, kdo určí, kolik je potřeba čet dělníků, jaká je posloupnost prací (nelze stavět střechu, dokud nejsou hotové obvodové zdi, dělat dlažbu před rozvody atp.). I kdyby projektový manažer mohl teoreticky celý dům postavit sám, musí umět delegovat práci na specialisty. Umět delegovat je důležitou vlastností ve všech oblastech managementu, ale v projektovém managementu je tato vlastnost o to důležitější, že jsou zde specialisté napříč celou organizací. Projektový manažer také kontroluje, zda jde vše podle plánu a přizpůsobuje projekt aktuálnímu stavu. V poslední řadě musí umět předvídat rizika, mít připravený plán a nápravné akce, pokud se riziko stane aktuálním problémem. To samé platí i o příležitostech. Projektový manažer se především snaží řídit 6 oblastí:

- **Náklady** – projekt by měl být naplánován tak, aby si ho společnost mohla dovolit, nebo pokud se jedná o dodávku produktu zákazníkovi, aby byl rentabilní. Je důležité mít na paměti, že i dobře sestavený rozpočet skoro určitě nebude po celou dobu projektu stabilní a bude nutné ho změnit. Jako příklad lze uvést např. nepředpokládaný problém při integraci nového softwaru s klíčovým systémem. Odhadne se zvýšená pracnost, čili se zvýší náklady projektu. Náklady jde samozřejmě i snižovat, např. pokud se jako jiný projekt použije systém pro lepší integraci softwaru (např. jako cloudové řešení Integration as a Service), čímž se sníží pracnost a tím i náklady. Příležitost ke snížení nákladů může být i nasazení již hotové funkcionality z jiného projektu, jejíž pracnost byla řešena v rámci již dokončeného projektu a jen se provedou úpravy, aby řešení vyhovovalo novému zákazníkovi.
- **Časový plán** – projekt je časově omezen, čili je nutné mít správně zpracovaný časový plán. Tento plán definuje časové milníky (akceptace, fakturace atd.), začátek a konec projektu. Časový plán se v průběhu projektu mění také na základě událostí,

kteří provázejí projekt. Snížení časového plánu či jeho prodloužení se děje z velice podobných důvodů jako u nákladů.

- **Kvalita** – Dokončení projektu v stanoveném termínu a za stanovenou cenu je sice velice dobrá vizitka projektového manažera, ale je také důležité dbát na kvalitu. Pokud budou splněny veškeré termíny, ale dodaný produkt bude mít velice nízkou kvalitu, je více než zřejmé, že nedojde k akceptaci díla. Proto je nutné dohlížet na kvalitu dodávaného produktu a před jeho dodáním pravidelně testovat jeho funkčnost.
- **Rozsah** – Rozsah je soupis všeho, co musí obsahovat finální produkt. Pokud by takový soupis neexistoval, nebylo by možné zjistit, zda daná funkcionalita je součástí dodávaného produktu, nebo zda není částí rozsahu projektu. Bez něho by si mohl zákazník myslet, že např. funkcionalita rozšířeného logování v aplikaci je v ceně dohodnutého projektu. Díky odsouhlasenému rozsahu je možné dohledat, zda daný požadavek je součástí projektu a má se realizovat, nebo zda jeho součástí není a zákazník si za něj musí připlatit (nebo vyměnit za jiný požadavek). U větších projektů je velice důležité, aby byl rozsah plně definován a všechny zúčastněné strany chápaly, co je a co není jeho součástí. Projektový manažer by se měl snažit dodržet rozsah tak, jak byl definován a nedělat práci navíc.
- **Riziko** – všechny projekty mají větší či menší rizika, která mohou ovlivnit celkový produkt (ať už v kvalitě, nebo změnou časového plánu a nákladů). Před započatím projektu je velice důležité zhodnotit situaci a položit si otázku, jak velké riziko je projektový manažer ochoten přijmout. Je nutné si říci, zda přidaná hodnota projektu stojí za dané riziko. Pokud riziko přesahuje hodnotu výstupu z projektu, není moudré projekt realizovat, nebo je nezbytné se pokusit riziko snížit na únosnou úroveň. Proto pokud je možné riziko snížit, měl by se o to projektový manažer pokusit.
- **Přínosy** – při realizaci projektu je nejdůležitější znát jeho přínos. Projekt by neměl být realizován, pokud není jeho přínos jasný, nebo pokud není dostatečný. Velice často se stává, že otázku na odpověď „Proč to děláme?“ nezná sponzor projektu odpověď, nebo pokud odpověď zná na začátku projektu, nekontroluje během projektu, zda je daný přínos pořád aktuální a zda má projektu pořád smysl. Např. pokud je cílem projektu nový přehledný e-shop, avšak 95% všech zákazníků podle průzkumů nakupuje v kamenných prodejnách, nemá smysl tento projekt realizovat, jelikož jeho přínos je oproti vloženým nákladům příliš nízký.

(MURRAY;2009)

**Struktura PRINCE2**

Strukturu metodologie PRINCE2 tvoří 4 základní elementy, které jsou integrovány do jednoho celku. Jedná se o tyto elementy:

- 7 principů
- 7 témat
- 7 procesů
- Přizpůsobení metodiky projektovému prostředí.

(MURRAY;2009)

### 2.1.2 Principy PRINCE2

Cílem metodologie PRINCE2 je dodávat metodu projektového řízení, kterou lze aplikovat nehledě na druh projektu, velikost, organizaci nebo kulturu. Toto je možné díky skutečnosti, že PRINCE2 je založen na principech. Principy jsou charakterizovány takto:

- jsou universální – lze je aplikovat v jakémkoliv druhu projektu,
- byly prověřeny mnohaletou praxí,
- napomáhají projektovým manažerům, kteří používají tuto metodologii, dodává důvěru a schopnost ovlivňovat a měnit jak bude projekt řízen.

Principy, na kterých je PRINCE2 postaven mají svůj původ v získaných poznatcích (tzv. Lesson Learned – souboru poznatků, které jsou sepsány při vyhodnocování projektu, aby se mohly osvědčené postupy použít v dalších podobných projektech, nebo aby bylo možné se podobným problémům vyvarovat), které byly převzaty z úspěšných i neúspěšných projektů. Pokud by se projekt neřídil podle těchto principů, nebylo by možné ho považovat za projekt řízený metodologií PRINCE2, jelikož právě tyto principy jsou tím základem, jenž definuje projekt řízený podle PRINCE2.

PRINCE2 obsahuje těchto 7 principů:

- Neustálé zdůvodňování opodstatněnosti.
- Učení se ze zkušeností.
- Definované role a odpovědnosti.
- Řízení po etapách.
- Řízení výjimkami.
- Zaměření se na produkty.
- Uzpůsobení řízení prostředí projektu.

(MURRAY;2009)

#### Neustálé zdůvodňování opodstatněnosti

Pokud se má realizovat projekt, je nutné před jeho započatím zjistit, zda se projekt vyplatí při současných nákladech a riziku. Proto pro projekt vedený metodologií PRINCE2 jsou následující podmínky:

- Existuje obhajitelný důvod, proč projekt začít.
- Opodstatněnost projektu by měla zůstat platná po celou dobu běhu projektu.
- Opodstatněnost projektu je zdokumentována a schválena.

Opodstatněnost je zdokumentována v obchodním případě. Jelikož je projekt neoddělitelně spojen s obchodním případem, ovlivňuje tak rozhodovací proces skrze celý projekt tak, aby zajistil, že projekt zůstane zaměřen na obchodní cíl a jeho přínosy. Pokud by společnost, která realizuje projekt, neměla řádně (nebo vůbec nevedla) vedený obchodní případ, mohlo by se lehce stát, že projekt bude realizován i v případě, kdy nemá žádné (nebo jen velmi nepatrné) reálné přínosy a může se odklonit i od strategie společnosti. Dodržovat sjednocení projektů se strategií společnosti je velice důležité, a to z toho důvodu, že by pak portfolio projektů obsahovalo nekonzistentní projekty a projekty s duplicitními cíli.

Projekt musí zohledňovat mnoho faktorů (zevnitř i zvenčí), které mohou markantně ovlivňovat jeho průběh. Tyto faktory mohou ovlivnit a i znatelně změnit opodstatněnost projektu. Může se jednat např. o legislativní změny, kdy se musí výsledný produkt změnit, nebo se musí úplně vypustit jeden z dílčích produktů. Proto je nutné si neustále pokládat otázku, zda je projekt stále výhodný a zda jeho opodstatněnost pořád platí. Obchodní případ tak musí neustále být jedním z ústředních dokumentů projektu. Lze ho i změnit, pokud lze projekt považovat za opodstatněný při stávajících podmínkách. Pokud však pokračování na projektu není rentabilní a není možné prokázat jeho opodstatněnost, pak je nutné projekt ukončit. Za těchto podmínek je nutné brát tento krok jako pozitivní pro společnost, jelikož pokud by se na projektu pokračovalo, skončil by dozajista neúspěchem, neměl by žádné či pouze zanedbatelné přínosy a je velice pravděpodobné, že jeho finanční výsledek by byl záporný.

(MURRAY;2009)

### Učení se ze zkušenosti

Týmy, které fungují podle PRINCE2 se učí ze zkušeností ostatních týmů a projektů. Své vlastní zkušenosti posbírané během projektu zaznamenávají během celého projektu a z těch se pak učí při realizaci podobných projektů a při podobných problémech. Práce na projektu je efektivnější, pokud tým narazí na potenciální problém a dokáže ho vyřešit právě díky přeneseným zkušenostem. Další výhoda vychází z faktu, že projekt je dočasná organizace vytvořená za účelem splnění daného cíle, se kterým si klasický liniový management neví rady. Pokud je projektový tým z větší části tvořen lidmi, kteří nikdy na projektu či na projektu tohoto typu nepracovali, může jim jejich úkol ulehčit právě použití zkušeností převzatých z obdobného projektu, díky čemuž se lépe aklimatizují v novém prostředí. PRINCE2 definuje metodu pro učení se ze zkušeností takto:

- **Při spouštění projektu** – Jako první krok by se měly zanalyzovat minulé projekty a zejména projekty stejného typu. Po dokončené analýze se stanoví, zda je možné

aplikovat získané poznatky ze zkoumaného projektu. Tento krok nabývá na důležitosti s přítomností méně zkušených lidí (pokud jsou lidé na projektu poprvé přiřazení na projekt, je tento krok nezbytný).

- **V průběhu projektu** – Poučení se z projektu by mělo probíhat napříč celým projektem po celou dobu jeho běhu. Poučení by mělo být součástí každého reportu, jelikož tyto poznatky v rámci toho samého projektu lze ještě použít. Hlavním cílem je odhalit rizika, ale především příležitosti na zefektivnění projektu ještě během jeho běhu.
- **Při uzavírání projektu** – Po ukončení projektu by se získané poznatky měly v rámci poučení z projektu zapsat a předat dále, aby mohly být použity v dalším projektu, nebo aby posloužily jako inspirace pro méně zkušené týmy. V případě, že tyto poznatky neevokují žádné zlepšení či změnu, jsou brány jako pouze jako identifikované poznatky (Lessons Identified).

Povinností každého, kdo je součástí projektového týmu, je snažit se identifikovat poznatky, které mohou vést ke zlepšení, a nečekat až je identifikuje někdo jiný.

(MURRAY;2009)

### Definované role a odpovědnosti

Každý projekt zahrnuje mnoho rozličných lidí, každý má specifické schopnosti a znalosti. I přes skutečnost, že je projekt skvěle naplánován a důsledně kontrolován, nemůže být úspěšný, pokud nejsou v projektu zahrnuti ti správní lidé, nebo pokud jsou zahrnuti špatní lidé, nebo ti správní lidé nevědí, co je od nich očekáváno a jaké jsou jejich povinnosti. To se může stát velice snadno, jelikož lidské zdroje jsou získávány napříč organizací. Projekt mnohdy zahrnuje více organizací a v každé organizaci mohou být definované různé role a odpovědnosti. Projekt se také musí vypořádat se zdroji, které není možné plně využívat (např. z důvodu částečného úvazku, nebo protože je zdroj využit i v jiném projektu). Z výše uvedených důvodů jen nutné, aby měl projekt pevně stanovenou a schválenou strukturu projektového týmu. V této struktuře musí být definované jednotlivé role a odpovědnosti používané v rámci projektu. Pevně stanovené role a odpovědnosti pomáhají efektivnější komunikaci a nekonzistenci vzniklé kvůli neznalosti kompetencí.

PRINCE2 určuje základní zainteresované role:

- Sponzory zodpovídající za obchodní stránku projektu, za cíle projektu a zajišťující, že investice poskytují za vložené peníze odpovídající hodnotu.
- Uživatelé, kteří po dokončení projektu budou výsledný produkt užívat.

- Dodavatele, kteří poskytují zdroje a expertní znalosti potřebné pro úspěšné dosažení cílů projektu (dodavatel může být jak externí, čili dodává produkt pro zákazníka, nebo interní, což bývá zpravidla jiné oddělení jako např. IT oddělení).

Všechny zájmy výše uvedených zainteresovaných rolí je nutné efektivně promítnout v projektovém týmu. Pokud nebude výsledný projekt použitelný pro jednu z výše uvedených skupin, nebo nebudou zohledňovány zájmy některé ze zainteresovaných rolí, je velice pravděpodobné, že se tato skutečnost promítne do přínosů daného projektu. Pokud z pohledů sponzorů nebude projekt přinášet požadované přínosy a ty budou převýšeny náklady, projekt skončí neúspěchem. Obdobně skončí projekt, pokud výsledný produkt bude pro uživatele nepoužitelný, nebo nebude existovat způsob jak elegantně dodat produkt od dodavatele. Závěrem lze u tohoto principu říct, že poskytuje odpověď na jednu ze základních otázek každého zainteresovaného jedince, a to „Co se ode mne očekává?“.

(MURRAY;2009)

### **Řízení po etapách**

Řízení po etapách poskytuje managementu jisté kontrolní body (po konci každé etapy se reportuje současný stav Projektovému výboru) napříč projektem. Na konci každé etapy je nutné shrnout současný stav projektu. Dále by se měl prověřit obchodní případ a projektový plán, aby se zajistilo, že je projekt nadále opodstatněný, a učiní se rozhodnutí, zda v projektu nadále pokračovat.

Rozdělení projektu na etapy poskytuje možnost managementu kontrolovat průběh jednotlivých projektů a zasahovat do jejich chodů podle aktuálních priorit a rizik. Platí, že kratší etapy umožňují efektivnější kontrolu a delší etapy zase tolik nezatěžují management.

Plánování by mělo být prováděno na takové detailní úrovni, aby mohlo být efektivně řízeno a co nejpřesněji předpovídáno. Plánování by nemělo přesáhnout tzv. plánovací horizont. Lze to uvést na příkladu. Pokud se naplňuje práce jednotlivých členů týmů na rok dopředu, je více než jisté, že po týdnu práce na projektu bude tento odhad naprosto mimo realitu. Naopak plánování práce na týdenní bázi bude znatelně efektivnější.

Tyto problémy překonává PRINCE2 následujícími přístupy:

- Rozdělením projektu na rozličný počet etap.
- Prováděním plánování na vysoké úrovni, jehož výsledkem je projektový plán a detailní plánování každé etapy, jehož výsledkem jsou plány jednotlivých etap.
- Plánováním, kontrolováním a monitoringem jednotlivých etap.

PRINCE2 požaduje, aby existovaly minimálně dvě etapy a to etapa nastavení projektu (nebo také iniciační fáze) a jedna či více dalších fází.

(MURRAY;2009)

## Řízení výjimkami

Tento princip se zaměřuje především na definici tolerancí pro každý projektový cíl a úroveň delegace, kterou lze použít. PRINCE2 definuje rozličné odpovědnosti pro řízení projektu na každé jeho úrovni:

- Delegováním odpovědnosti z jedné úrovně managementu na další definováním tolerancí pro každou z 6 hlavních oblastí, které sleduje PRINCE2.
  - o Náklady – definuje se horní a případně i spodní hranice, o kolik může být překročen plánovaný rozpočet. Např. že rozpočet projektu může být překročen maximálně o 15%.
  - o Časový plán – definuje se horní a případně i spodní hranice, o kolik může být překročen časový plán. Např., že může být časový plán dané etapy přesažen maximálně o 2 měsíce.
  - o Kvalita – definuje se horní a případně i spodní hranice, o kolik se může lišit cílová kvalita. Např. může jít o maximální počet softwarových chyb v dodané verzi produktu, nebo že každá hřídel, která je součástí celkového dodávaného stroje, má toleranci od -1 milimetru do +2 milimetrů.
  - o Rozsah – definuje varianty plánovaného produktu (a dílčích produktů), které jsou přípustné. Jedná se hlavně o definici požadovaných dílčích produktů (Must have) a dalších, bez kterých se výsledný produkt obejde (desirable). Např. se může jednat o soupis hlavních funkcionalit softwaru, bez kterých se zákazník neobejde a minimálního počtu implementovaných funkcionalit, které sice nejsou nezbytné pro chod softwaru, ale zákazník by je rád měl (může jít o detailní logování, nebo změnu uživatelského rozhraní podle každého uživatele).
  - o Riziko – definuje dva druhy limitů. Prvním limitem je celkový počet nákladů pro odstranění všech vyskytlých rizik. Např. může tento limit být nastaven na 8% celkového projektového rozpočtu. Druhým limitem je limit pro každé jednotlivé riziko. Zde může jít např. o riziko markantního zásahu do funkcionalit navazujících systémů.
  - o Přínosy – definuje horní a případně i spodní hranici každého projektového cíle. Např. může jít o snížení nákladů na vedení skladu o 30 - 40%.
- Nastavení kontrol, které jsou při překročení nadefinovaných tolerancí (nebo při předpovědi, že budou v následující etapě překročeny) ihned reportovány vyšší úrovni managementu, který se musí rozhodnout, jak bude dále postupovat. Pokud se tedy překročí rozpočet o nastavený limit 20%, vyšší úroveň managementu rozhodne, zda je projekt i nadále rentabilní (a případně posune i onu toleranci), nebo zda projekt ukončit.

- Nastavení dohledového mechanismu, aby si každá úroveň managementu mohla být jista, že nastavené kontroly fungují a jsou efektivní.

Implementace řízení výjimkami poskytuje velice efektivní využití času managementu, jelikož snižuje jeho časové vytížení, aniž by utrpěla míra jejich kontroly nad projektem a to zajištěním, že správná rozhodnutí jsou přijímána na správných úrovních managementu.

(MURRAY;2009)

### **Zaměření se na produkty**

Metodologie PRINCE2 je především zaměřena na dodání produktu, obzvláště na jeho požadovanou kvalitu. Úspěšný projekt je vždy zaměřen na výstupy a nikoli na aktivity. Produktově zaměřený projekt je takový projekt, který upřednostňuje schválení a definici produktu před aktivitami k jeho vytvoření. Skupina dohodnutých produktů vytváří rozsah projektu, který slouží jako výchozí dokument k plánování a kontrole. Cílem projektu je splnění očekávání zainteresovaných stran v souladu s opodstatněním projektu. Aby toto bylo možné, je nutné, aby všichni, kterých se projekt dotýká, chápali, které produkty jsou potřeba a jaká je jejich očekávaná kvalita a jak bude kontrolováno, zda bylo dosaženo požadované úrovně kvality.

PRINCE2 využívá popis produktu k objasnění všech jeho důležitých aspektů. U každého produktu jsou popsány jeho účel, složení, odvození, formát a jakostní kritéria. Tento popis je realizován v plánovací fázi a je důležité ho sestavit, co nejdříve to bude možné. Popis produktu je výchozí při sestavování odhadů, potřebných zdrojů, závislostí a plánu aktivit.

Bez zaměření na produkt je projekt vystaven nebezpečí, že dojde k realizaci mnoha rizik jako neakceptování produktu, např. z důvodu, že nebylo dostatečně (nebo nebylo vůbec) definováno, jaké parametry a v jaké kvalitě musí produkt splňovat, aby došlo k jeho akceptaci. Dále, pokud není přesně definováno, co se musí udělat, velmi pravděpodobně dojde k odchýlení od plánů a zvýšení nákladů, jelikož bude docházet k velkému počtu změnových požadavků a realizaci požadavků, které neměly být součástí projektu.

(MURRAY;2009)

### **Uzpůsobení řízení prostředí projektu**

Aby projekt byl úspěšný, je důležité ho přizpůsobit každému projektu, jelikož každý projekt je unikátní, co se týče velikosti, prostředí, složitosti, rizik atd. Tento princip potvrzuje, že PRINCE2 lze považovat za universální metodologii, která lze použít na každém projektu nezávisle na jeho druh a velikost. Pokud by se projekt nepřizpůbil, tak je více než jisté, že projekt skončí neúspěchem, nebo se alespoň bude potýkat s velkými problémy. Řízení interního projektu, jehož výstupem má být vedlejší systém, nebude možné aplikovat na velký projekt pro státní správu, jehož výstupem bude klíčový systém pro fungování dané instituce a bude na něm spolupracovat několik různých dodavatelů. Nedodržování toho-

to principu může vést ke dvěma extrémům. Prvním extrémem je robotické dodržování metodologie, aniž by bylo vzato v potaz, zda je nutné řídit daný projekt přesně podle manuálu. Druhým extrémem je takzvaný „hrdinský“ styl řízení, kdy se metodologie téměř nedodržuje. Účelem uzpůsobení řízení prostředí projektu je tedy:

- Zajištění, aby metodologie řízení projektu správně propojena s projektovým prostředím, aby byly jednotlivé metody sjednoceny s procesy, které řídí nebo podporují projekt)
- Zajištění kontrol projektu tak, aby odpovídaly velikosti, složitosti a rizikům projektu. Není efektivní, pokud malý interní projekt na uzpůsobení vedlejšího podpůrného procesu byl kontrolován tak detailně a často jako projekt nasazení klíčového systému.

Uzpůsobení řízení prostředí projektu vyžaduje, aby projektový manažer a projektový výbor definovali, jak a jaké metody budou použity, jaké kontroly, jak často a jak detailně budou prováděny.

Aby bylo zajištěno, že všichni lidé, kteří jsou součástí projektu, chápou, jakým způsobem je PRINCE2 používán, mělo by být v dokumentaci o nastavení projektu uvedeno, jak je řízení uzpůsobeno aktuálnímu projektu.

(MURRAY;2009)

### 2.1.3 Témata PRINCE2

Témata uvedená v metodologii PRINCE2 popisují aspekty projektového managementu, kterým musí být věnována neustálá pozornost. Hlavní výhodou PRINCE2 je způsob, jakým jsou všechny témata integrována mezi sebou. Toho je dosaženo specifickým přístupem PRINCE2 k jednotlivým tématům, která jsou navržena tak, aby bylo možné je mezi sebou efektivně integrovat. PRINCE2 obsahuje celkem 7 témat:

- Obchodní případ – Na začátku každého projektu se nachází myšlenka na zefektivnění chodu organizace (ať už se jedná o implementaci nového softwaru, nebo o zefektivnění procesu atd.). Tato myšlenka je poté analyzována, zda má pro danou organizaci nějaký potenciální přínos. Právě toto téma rozvíjí základní myšlenku do potenciálního podnětu pro investici. Toto téma poskytuje odpověď na otázku „Proč?“.
- Organizace – Organizace, která projekt sponzoruje, potřebuje práci na projektu alokovat manažerům, kteří budou zodpovědní za její dokončení. Toto téma popisuje role a odpovědnosti a definuje projektovou organizační strukturu. Toto téma poskytuje odpověď na otázku „Kdo?“.

- Kvalita – Rozvádí počáteční myšlenku z hlediska kvality. Poskytuje všem zúčastněným na projektu informace, jaké kvalitativní atributy musí cílový produkt splňovat a jak projektový management zajistí jejich kontinuální plnění. Toto téma poskytuje odpověď na otázku „Co?“.
- Plány – Metodologie PRINCE2 řídí projekt na základě série schválených plánů. Toto téma doplňuje téma kvalita popisem potřebných kroků pro vytvoření plánu a technik, které mají být aplikovány. Plány jsou sjednoceny s potřebami jednotlivých rolí napříč různými úrovněmi organizace. Jsou nástrojem ke kontrole projektu. Toto téma odpovídá na otázky „Jak?“, „Kolik?“ a „Kdy?“.
- Riziko – Rizika jsou součástí každého projektu, který je typicky rizikovější než běžná rutinní činnost. Toto téma popisuje, jak si projektový management poradí s nejistotou v projektových plánech a s riziky z vnějšího projektového prostředí. Toto téma poskytuje odpověď na otázku „Co když?“.
- Změna – Popisuje, jak projektový management jedná s otevřenými body, které mají potenciální dopad na základní aspekty projektu (plány a produkty). Otevřeným bodem může být požadavek na změnu, nepředvídatelný problém atd. Toto téma odpovídá na otázku „Jaký je dopad?“.
- Průběh – Toto téma se zabývá stávající životaschopností projektu. Popisuje proces schvalování plánů, monitorování současného stavu a eskalační proces v případě, že projekt nejde podle plánu. Nakonec určuje, jak by měl projekt pokračovat. Toto téma odpovídá na otázky „Kde jsme teď?“, „Kam směřujeme?“ a „Měli bychom pokračovat?“.

Všechna výše uvedená témata musí být dodržena. Je samozřejmě možné (a žádoucí) uzpůsobit je projektovému prostředí, aby nejlépe vystihovala potřeby projektu, rozsah, rizikovitost a složitost projektu. Níže následuje podrobný popis každého z nich. U každého tématu je popsán jeho účel, definice a přístup PRINCE2 k tématu.

(MURRAY;2009)

## **Obchodní případ**

### **Účel obchodního případu**

Hlavním účelem obchodního případu je nastavení mechanismu, který vyhodnocuje, zda projekt je (nebo nadále zůstává) životaschopný, dosažitelný a zda stále může přinést požadované přínosy a je tedy nadále přínosné do něj investovat. Toto vychází z jednoho ze základních principů PRINCE2, a to z neustálého zdůvodňování opodstatněnosti. Bez tohoto opodstatnění nemůže žádný projekt začít, a pokud by se přestalo platit v průběhu projektu, je nutné projekt zastavit, nebo úplně zrušit. V metodologii PRINCE2 je právě toto opodstatnění zaneseno v obchodním případě, ve kterém je popsán důvod, proč je pro-

jekt veden za stávajícího rozpočtu, při identifikovaných rizikách a očekávaných přínosech. Při změnách v projektu je důležité mít při analýze dopadu na paměti, že veškeré techniky a metody projektu jsou jenom nástrojem k dosažení cíle. Proto by tyto analýzy měly být především realizovány s ohledem na obchodní případ (jak moc bude obchodní případ ovlivněn).

Hlavní uživatelé jsou zodpovědní za specifikaci jednotlivých přínosů, které má požadovaný produkt přinést. Sponzor projektu je zodpovědný za potvrzení, že přínosy, které byly uživateli specifikovány, jsou podloženy reálnou hodnotu, která se dá vyčíslit penězi, jsou v souladu s podnikovou strategií a jsou skutečně realizovatelné.

Obchodní případ je vytvořen na počátku projektu a je udržován po celou dobu běhu projektu (jedná se tedy o živý dokument). Tento dokument je kontrolován projektovým výborem na konci každé etapy.

(MURRAY;2009)

### Definice obchodního případu

Obchodní případ je optimální směsice informací, která se používá k určení, zda je projekt životaschopný, dosažitelný a zda je tedy moudré do něj investovat. Projektový výbor a jednotlivé zainteresované strany musí mít důvěru v obchodní případ po celou dobu běhu projektu (po celou dobu jeho životaschopnosti). Obchodní případ tak dodává odpověď na otázku, zda je projekt stále výhodný pro naše investice. Jelikož tato otázka je aktuální po celou dobu běhu projektu, je důležité se na ni neustále ptát zvláště v souvislosti s obchodním případem, který by dozajista neměl být použit pouze jako informace, zda projekt vůbec začít, ale zda je výhodné v něm i nadále pokračovat. Proto by měl být tento dokument upravován napříč projektem aktuálními informacemi o aktuálních nákladech, rizikách a přínosech (zda se objevila nová rizika či přínosy, zda je projekt stále výhodný i po změně celkových nákladů apod.).

PRINCE2 rozlišuje 3 pojmy, které se týkají přínosu projektu:

- **Výstupy** – Jedná se o jakýkoliv (dílčí) produkt, který je vytvořen kterýmkoliv specialistou pracujícím na projektu (ať se jedná již o hmatatelný produkt nebo nehmatatelný). Zde se může jednat např. o nový CRM systém.
- **Výsledek** – Výsledek projektu je výsledkem změn zapříčiněných užíváním výstupů projektu. Zde se může jednat o zrychlení přijímání objednávek od zákazníků prostřednictvím nově implementovaného CRM systému.
- **Přínos** – Jedná se o měřitelné vylepšení, které vychází z výstupu projektu. Zde se jedná např. o zvýšení tržeb o 15% díky zrychlenému přijímání a plnění objednávek a snížení nákladů o 10%, které je zapříčiněno nově implementovaným CRM systémem.

Během běhu projektu se většinou nejvíce zaměřuje na produkty (výstupy). Je to dáno skutečností, že výsledky a přínosy jsou měřitelné (a vůbec patrné) až po dokončení projektu.

Každý projekt je utvářen prostředím, ve kterém bude realizován. Právě prostředí a druh projektu určuje, jaké budou cíle projektu. Dosažení cílů projektu bude měřeno v závislosti na druhu projektu. PRINCE2 rozlišuje tyto druhy projektu:

- Projekt důležitý pro další budoucnost organizace.
- Neziskový projekt.
- Vývojový projekt.
- Dodavatelský projekt.
- Projekt, na kterém spolupracuje více organizací.

Značnou část projektů lze měřit pomocí návratnosti investice. Existují však i projekty, kde není možné tuto metodu použít, jako např. u neziskových projektů, které mohou být měřeny jinou metodou (např. se může jednat o projekty neziskových organizací, kde může být projekt vyhodnocen celkovým počtem vybraných peněz na charitativní akci, nebo počtem rodin zasažených povodněmi, kterým byla rozdána potravinová pomoc). Nehledě na zvolenou metodu měření výsledků je stále důležité si říct, zda projekt při plánovaných nákladech, rizikách a přínosech je výhodný.

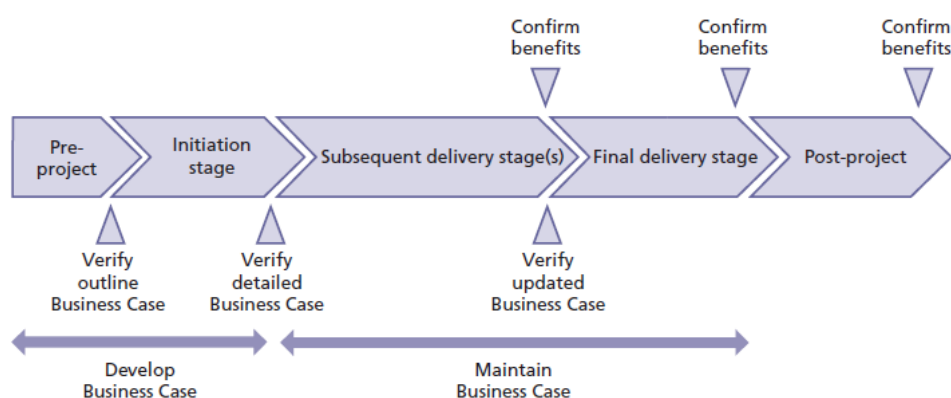
(MURRAY;2009)

## Přístup PRINCE2 k obchodnímu případu

V metodologii PRINCE2 je obchodní případ vytvořen na začátku projektu. Během běhu projektu je udržován a na konci každé etapy je formálně zkontrolován projektovým výborem a je potvrzena jeho stálá platnost. V souvislosti s obchodním případem lze tedy hovořit o 4 akcích, kterými prochází:

- **Vytvoření** – Za obchodní případ je odpovědný sponzor projektu (sepsání samotného dokumentu je většinou delegováno na business analytiku). Prvotní obchodní případ je odvozen z projektového mandátu (tj. informace vytvořená mimo projekt, která je základem zadání a vstupem pro zahájení projektu) a je vytvořen v procesu spuštění projektu a poté je schválen projektovým výborem. Detailní obchodní případ je pak odvozen ze základního obchodního případu, projektového plánu a registru rizik.
- **Kontrola a udržování** – Po konci každé etapy je nutné kontrolovat, zda je projekt stále životaschopný a rentabilní. Kontrola se provádí na konci každé etapy (viz obrázek 1).
- **Potvrzení** – Na základě výsledků kontroly se provede rozhodnutí, zda se na projektu bude pokračovat či nikoliv.

(MURRAY;2009)



Obrázek 1 Obchodní případ (MURRAY;2009)

## Organizace

### Účel organizace

Účelem tématu organizace je definovat projektovou organizační strukturu. PRINCE2 je založen na zákaznicko-dodavatelských vztazích. Tyto dvě skupiny spolu musí navzájem komunikovat a mít definované jednotlivé role a odpovědnosti. Pro efektivní řízení projektu je tedy nutné mít dobře definovaný projektový tým. Takový projektový tým by měl:

- Mít zástupce za business, uživatele a další zainteresované strany.
- Zajistit odpovídající správu definicí odpovědností pro řízení a dodávání produktu, včetně odpovědnosti na každé úrovni.
- Pravidelně revidovat projektové role, aby bylo zajištěno, že jsou i nadále efektivní.
- Nastavit efektivní strategii pro řízení komunikačního toku mezi týmem a zainteresovanými stranami.

(MURRAY;2009)

### Definice organizace

Téma organizace definuje několik pojmů v souvislosti s řízením projektu.

- **Projekt** – Dočasná organizace vytvořená za účelem dodání jednoho či více produktů na základě dohodnutého obchodního případu.
- **Program** – Projekt může být samostatný, nebo řízen jako součást programu příbuzných projektů. Program je stejně jako projekt dočasná flexibilní organizace, která je zřízena za účelem řízení a dohledem nad skupinou propojených projektů. Zpravidla má delší trvání než projekt.
- **Organizace (Společnost)** – Projekty jsou realizovány v prostředí organizace, která může být tradiční ve smyslu rozdělení zaměstnanců podle typu práce až po netypické struktury.
- **Role** – PRINCE2 definuje různé role, které mohou být sdíleny nebo kombinovány podle potřeb projektu. Vždy však musí být definovány jejich odpovědnosti

PRINCE2 dále v rámci tohoto tématu definuje tři základní zájmové skupiny:

- **Business** – Projekt by měl poskytovat hodnotu za zainventované peníze. Každý produkt, který je výsledkem projektu by měl mít přidanou hodnotu pro business.

- **Uživatel** – PRINCE2 vnímá rozdíl mezi potřebami businessu a potřebami uživatelů, kteří budou výsledný produkt používat pro vytváření přínosů, které jsou hlavním cílem projektu.
- **Dodavatel** – Tvůrce produktů, který vlastní zdroje s určitými schopnostmi.

(MURRAY;2009)

### Přístup PRINCE2 k organizaci

PRINCE2 používá různé úrovně řízení. Jedna úroveň řízení je potřeba na řízení projektu jako celku a další úroveň je nutná na řešení každodenních potřeb projektu. Celkově PRINCE2 rozlišuje 4 úrovně řízení, z nichž tři reprezentují projektový management a čtvrtá se nachází mimo projekt.



Obrázek 2 Čtyři úrovně řízení v PRINCE2(MURRAY;2009)

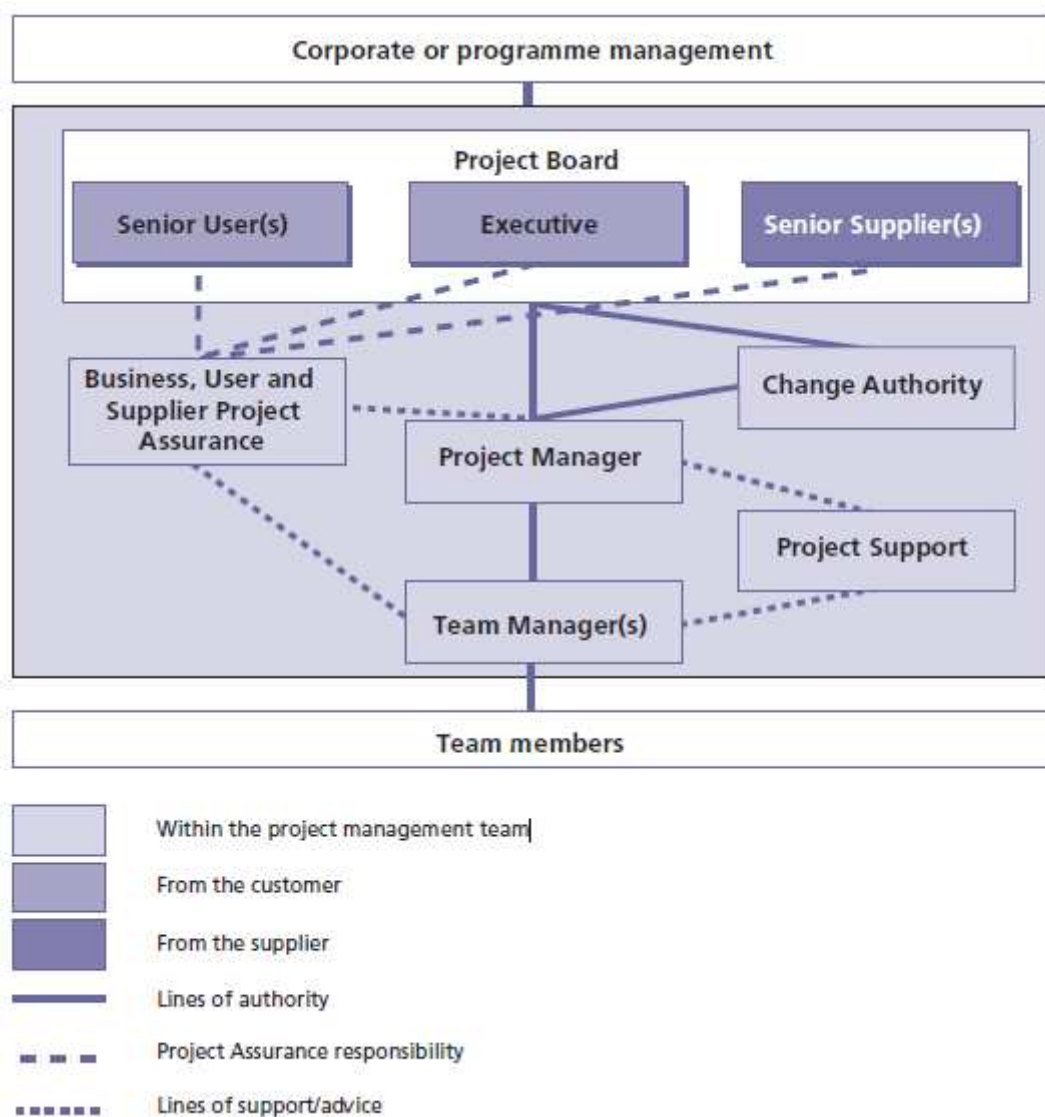
- **Vedení organizace nebo programový management** - Tato úroveň se nachází mimo projektový management. Nastavuje tolerance pro projekt a zavádí projekty. Tyto informace jsou zaneseny v projektovém mandátu.
- **Řízení** – Projektový výbor je odpovědný za řízení projektu v nastavených tolerancích. Na této úrovni se kontrolují a schvalují veškeré plány, rozpočty a obchodní případy.
- **Řízení na denní bázi** – Projektový manažer řídí projekt na denní bázi v rámci omezení nastavených projektovým výborem. Hlavní odpovědností projektového

manažera je dodávání produktu v souladu s naplánovaným rozsahem, náklady, kvalitou, časem a přínosy.

- **Dodávání** – Jednotliví členové týmu jsou zodpovědní za dodávání produktu v adekvátní kvalitě, nákladech a čase. V závislosti na složitosti projektu lze některé kompetence týkající se plánování jednotlivých produktů delegovat na týmové vedoucí.

Na obrázku 3 lze vidět, různé role v projektovém týmu definované v PRINCE2.

(MURRAY;2009)



Obrázek 3 Role v projektovém týmu a jejich závislosti(MURRAY;2009)

**Kvalita**

**Účel kvality**

Téma kvalita definuje prostředky, jakými bude vytvořen a posléze kontrolován produkt, zda vyhovuje svému účelu. Poskytuje všeobecné porozumění toho, co projekt vytvoří (rozsah) a kritéria oproti kterým bude projekt produkty kontrolovat (kvalita). Toto téma má také za úkol kontinuální zlepšování po celou dobu běhu projektu.

(MURRAY;2009)

### Definice Kvality

PRINCE2 definuje tyto pojmy v souvislosti s tímto tématem:

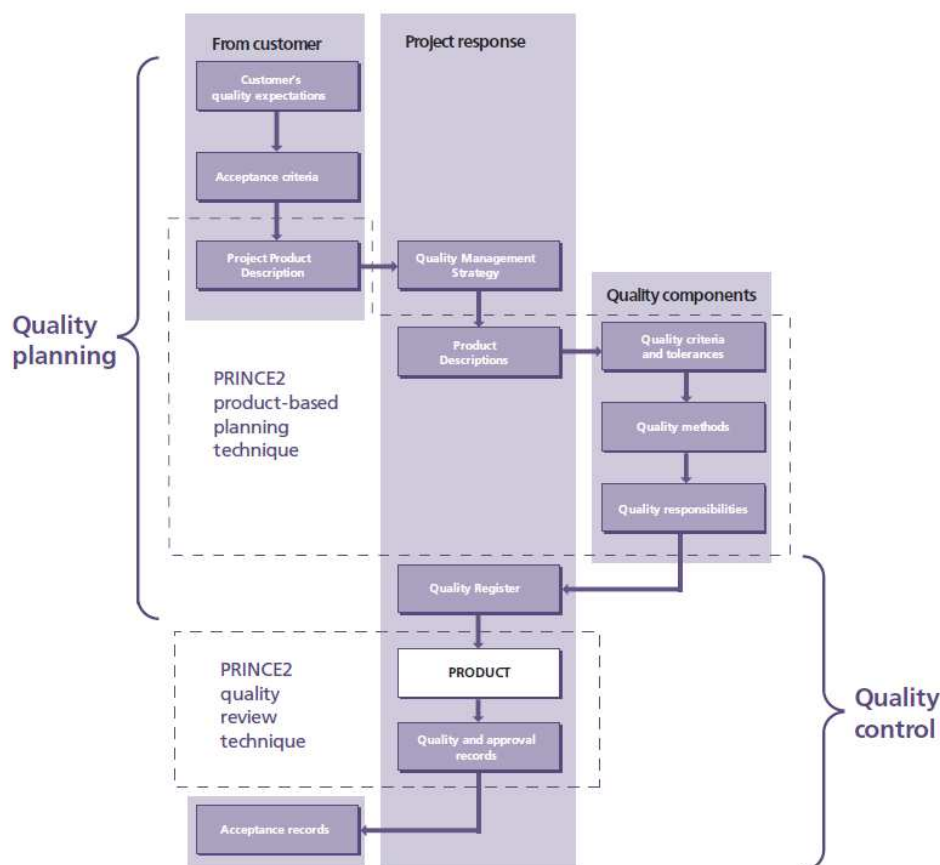
- **Kvalita** – Kvalita je obecně definována jako skupina charakteristik produktu, procesu, služby atp., které splňují určitá požadovaná kritéria nebo specifikace.
- **Rozsah** – Rozsah je celkový součet všech produktů. Je definován produktovým rozpadem (tj. hierarchie všech produktů, která má být dodána během daného plánu).
- **Management kvality a systémy řízení kvality** – Management kvality je definován jako koordinované aktivity určené k řízení a kontrole organizace. Systém řízení kvality je celkový soubor standardů, procedur a odpovědností určených pro organizaci.
- **Plánování kvality** – Zde se jedná o definici všech produktů a jejich kvalitativních kritérií a metod. Také určuje odpovědnosti za kvalitativní aspekty projektu.
- **Kontrola kvality** – Provádí se, aby bylo zajištěno, že produkty splňují kvalitativní požadavky.

(MURRAY;2009)

### Přístup PRINCE2 ke kvalitě

Nejdříve je nutné identifikovat veškeré požadované produkty. Poté následuje jejich definice v popisu produktu. Tento dokument obsahuje popis účelu produktu, složení, odvození a kvalitativních kritérií. Nakonec se implementují kvalitativní metody napříč projektem.

(MURRAY;2009)



Obrázek 4 Kvalita v PRINCE2(MURRAY;2009)

## Plány

### Účel Plánu

Úspěšný projektový management se spoléhá na plánování, jelikož bez plánu není možné kontrolovat. Plánování poskytuje všem zúčastněným osobám informace o tom co je potřeba, jak toho bude dosaženo (jakými specialisty, vybavením a zdroji) a zda jsou dosažitelné. Pomocí plánů se dá také nejlépe měřit postup projektu.

(MURRAY;2009)

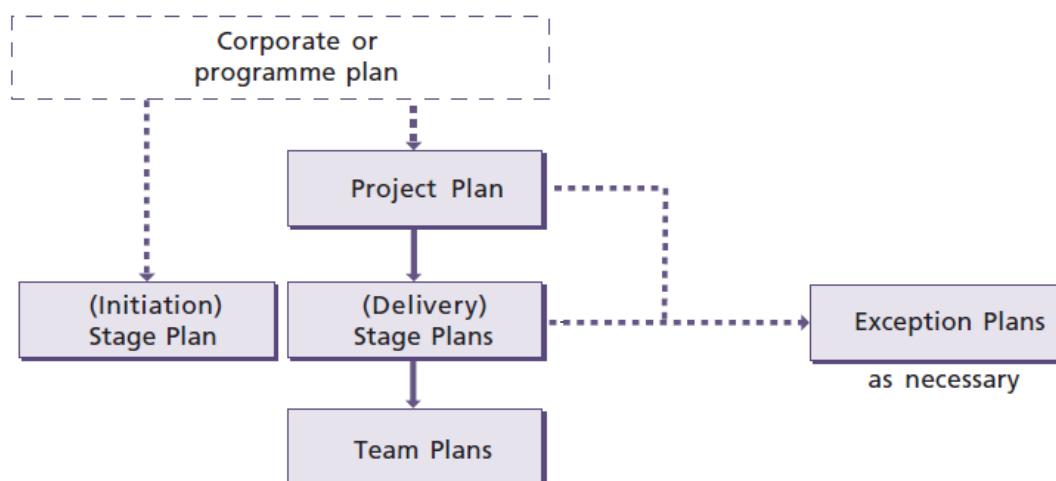
### Definice Plánu

Plán v PRINCE2 není jenom časová osa. Je to dokument, který popisuje jak, kdy a kým bude specifický cíl (nebo cíle) dosažen. Proto musí plán obsahovat dostatečné informace, aby bylo možné potvrdit dosažitelnost vytyčených cílů. Plánování je tedy činnost pro vytvoření a udržování plánu. Jeho výstupem jsou dokumenty a diagramy. Plánování je nezbytné pro úspěšné řízení projektu nezávisle na jeho velikosti, složitosti nebo prostředí. Bez plánů by nebylo možné předvídat události projektu ve smyslu změn rozsahu, rizik, času, nákladů a přínosů. Je také nutné mít více úrovní plánů. Jeden plán bude sloužit k plá-

nování projektu z celkového pohledu a jiný bude sloužit k plánování práce v horizontu několika týdnů. PRINCE2 definuje 3 druhy plánů:

- **Projektový plán** – Obsahuje informaci o tom, jak a kdy budou časové, rozsahové, nákladové, a kvalitativní cíle splněny, a to definicí hlavních produktů, aktivit a zdrojů potřebných pro projekt.
- **Etapový plán** – Plán potřebný pro každou etapu. Etapový plán je velice podobný projektovému plánu, ale etapový plán je detailnější a zaměřený na každodenní řízení projektu. Plán pro nadcházející etapu je vytvářen vždy na konci etapy předchozí.
- **Týmový plán** – Týmový plán je vytvářen týmovým vedoucím k naplánování jednoho či více balíků práce. Týmové plány nejsou povinné a jejich množství závisí na složitosti a potřebách projektu.
- **Plán realizace výjimky** – Plán, který slouží jako záložní plán. Je připravený pro odpovídající úroveň managementu, aby nastínil potřebné akce, aby se plán dokázal vyrovnat s událostí, která přesahuje nastavené tolerance. Tento plán nahrazuje projektový plán nebo plán etapy. Pro nahrazení požadovaného plánu je potřeba souhlas projektového výboru. Plán realizace výjimky je plánován na takové úrovni detailu, v jakém je plán, který má být nahrazen.

(MURRAY;2009)



Obrázek 5 Plány v PRINCE2(MURRAY;2009)

### Přístup PRINCE2 k Plánu

Nejdříve se musí identifikovat požadované produkty. Teprve po identifikaci produktu či produktů lze identifikovat i aktivity, závislosti a zdroje potřebné pro úspěšné dodání požadovaného produktu či produktů. Plánování v metodologii PRINCE je produktově založeno.

(MURRAY;2009)

## Riziko

### Účel Rizika

Účelem tématu riziko je identifikace a kontrola nejistoty, jehož výsledkem je zvýšení pravděpodobnosti, že projekt skončí úspěchem. Riziku jako takovému se v projektu nedá vyhnout. Dá se však předvídat a připravit se na to, že jeho naplnění může výrazně změnit chod projektu. Proto je důležité se na potencionální rizika připravit a eliminovat je, nebo alespoň zmírnit jeho potencionální dopady. Řízení rizika by mělo být systematické. Jde o aktivní identifikaci, posouzení a kontrolu rizik, která mohou ovlivnit cíle projektu. Účelem je nastavit nákladově efektivní systém řízení rizik, a podpořit lepší proces přijímání rozhodnutí díky dobrému pochopení jednotlivých rizik (jejich příčiny, pravděpodobnost výskytu, dopady, načasování a výběr adekvátních opatření).

(MURRAY;2009)

### Definice Rizika

Riziko je definováno jako nejistá událost nebo souběh událostí, které pokud se projeví, mohou mít neblahý dopad na dosažení cílů projektu. Riziko lze rozdělit na dvě základní entity a to na hrozbu a příležitost. Hrozba je taková nejistá událost, která může mít negativní dopad na cíle projektu. Příležitost je taková nejistá událost, která může mít pozitivní dopad na cíle projektu. Řízení rizik je systematická aplikace procedur identifikace a posouzení rizik a implementace reakcí na rizika. Aby bylo řízení rizik efektivní, musí být rizika:

- **Identifikována** – Zde se jedná o analyzování potencionálních rizik a zvážení jejich dopadů na projektové cíle a jejich popsání tak, aby byly obecně pochopeny.
- **Posouzena** – Každé riziko musí být ohodnoceno z hlediska pravděpodobnosti jeho výskytu a dopadu. Součástí je také pochopení celkového rizika, jež projekt ovlivňuje.
- **Kontrolována** – Obsahuje odpovídající reakce na každé riziko, přiřazení vlastníka každému riziku a poté monitoring, kontrolování a provádění těchto reakcí.

(MURRAY;2009)

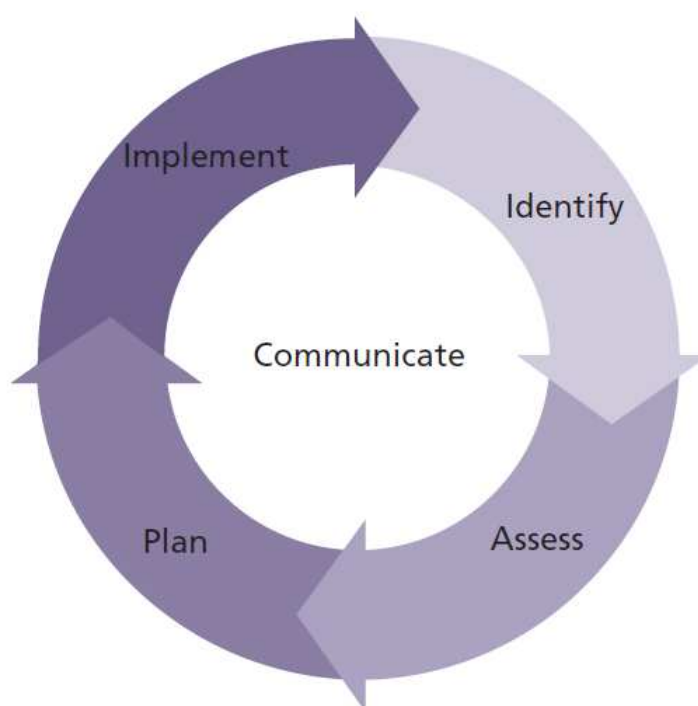
### Přístup PRINCE2 k Riziku

Řízení rizik v PRINCE2 je založen na těchto principech:

- Pochopení kontextu projektu.
- Zahrnutí všech zainteresovaných stran.
- Volba jasného cíle projektu.
- Vytvoření řízení rizik v projektu.

- Pravidelné reportování rizik
- Definice jasných rolí a odpovědnosti.
- Vytvoření kultury podporující řízení rizik.
- Monitorování pro včasné odhalení rizik.
- Vytvoření retrospektivních cyklů a hledání kontinuálního zlepšení.

(MURRAY;2009)



Obrázek 6 Řízení Rizik v PRINCE2 (MURRAY;2009)

## Změna

### Účel Změny

Změna je v každém projektu nevyhnutelná. Každý projekt potřebuje systematický přístup k identifikaci, posouzení a kontrole otevřených bodů, které mohou vyústit ve změnu. Změny mohou vyvstat od členů týmů, z požadavků zainteresovaných stran nebo z okolních vlivů. PRINCE2 nabízí systematický přístup, díky kterému se projekt může vypořádat s otevřenými body, které mohou ovlivnit projektové cíle. Kontrola otevřených bodů a změn je kontinuální činnost, která je prováděna po celou dobu běhu projektu. Pokud by projekt neměl zavedenou (nebo měl zavedenou neefektivní) kontrolou otevřených bodů a změn, mohl by se rychle vymknout kontrole, nebo by byl vůči zainteresovaným stranám absolutně ignorantní. Účelem tohoto tématu není zabránit změnám, nýbrž aby každá změna byla před implementací schválena relevantní autoritou. (MURRAY;2009)

## **Definice Změny**

Kontrola otevřených bodů a změn zajišťuje, že veškeré otevřené body a změny, které mohou projekt a jeho schválenou baselinu ovlivnit, jsou identifikovány, posouzeny a poté schváleny, zamítnuty nebo odloženy.

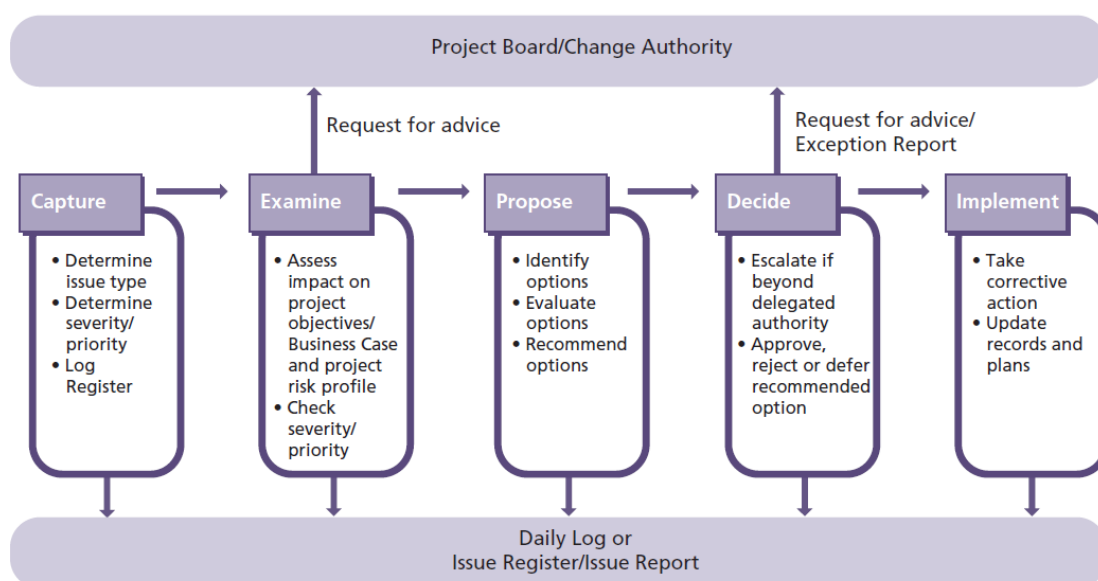
PRINCE2 definuje v rámci tohoto tématu konfigurační management, což je technická a administrativní činnost, která se zabývá vytvářením, údržbou a kontrolovanou změnou konfigurace během celé doby běhu projektu. V rámci konfiguračního managementu definujeme konfigurační položku. Konfigurační položka může být část produktu, produkt nebo několik produktů, které tvoří předávací balíček. Předávací balíček je celistvá a konzistentní množina produktů, které jsou řízeny, testovány a nasazeny jako jediná entita, která bude posléze předána uživatelům.

PRINCE2 využívá pojem otevřený body, pod kterým si lze představit jakoukoliv relevantní událost, která se stala, nebyla plánována a vyžaduje pozornost managementu. Může se jednat o otázku, znepokojení, požadavek na změnu, doporučení atp.

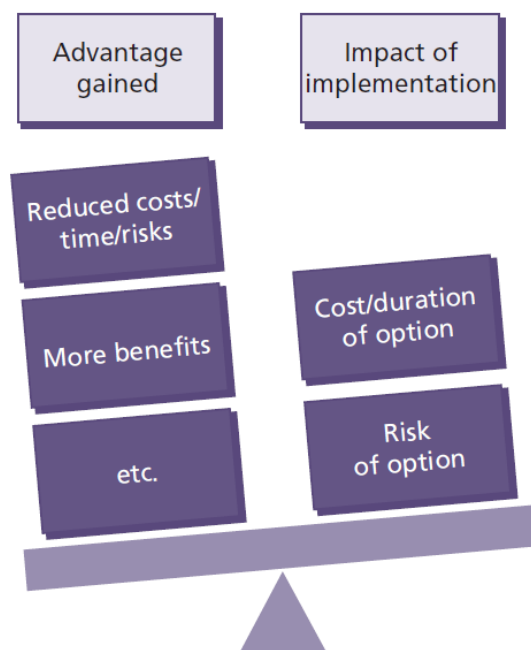
(MURRAY;2009)

## **Přístup PRINCE2 ke změnám**

Nejdříve se nastaví systém kontrol otevřených bodů, poté se nadefinuje řízení změn a konfigurace. Níže na obrázku 7 je vyobrazena procedura kontroly otevřených bodů a změn a analýza možností. (MURRAY;2009)



Obrázek 7 Procedura kontroly otevřených bodů a změn(MURRAY;2009)



Obrázek 8 Analýza možností(MURRAY;2009)

## Průběh

### Účel Průběhu

Hlavním účelem tématu průběh je nastavit mechanismus monitorování a porovnávání aktuálně dosažených úspěchů a jejich porovnání s plánem. Z těchto informací se dá pak předvídat budoucí vývoj a slouží také jako podklad pro rozhodnutí, zda je projekt i nadále životaschopný. (MURRAY;2009)

## **Definice Průběhu**

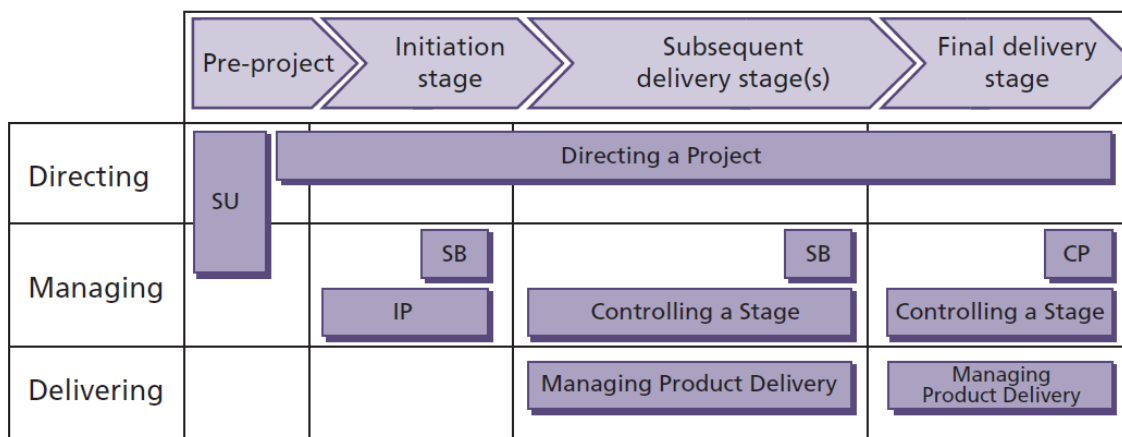
Průběh je měřítko úspěchu plnění cílů projektu. Průběh může být monitorován na úrovni balíčků práce, etap a projektu. Kontroly průběhu jsou přítomny na všech úrovních projektového managementu, kde se monitoruje průběh. Získané informace se porovnají s plánem, který se poté reviduje. Poté jsou identifikovány aktuální problémy a rizika. Zahájí se opravná opatření a autorizuje se další práce. Toto téma úzce souvisí s principy řízení po etapách (průběh je měřitelný v každé etapě), řízení výjimkami (jsou nastaveny tolerance, které určují, jaký je průběh projektu). Průběh má jisté očekávané hodnoty, kterých má být dosaženo v určitých nastavených tolerancích. (MURRAY;2009)

## **Přístup PRINCE2 k průběhu**

Kontrolou průběhu oproti plánu získáme informace, které následně využijeme pro úpravu balíků práce nebo i plánu. Kontrola postupu se provádí nejdříve delegováním odpovědnosti z jedné úrovně managementu na nižší. Poté se rozdělí projekt na etapy. Následuje kontinuální reportování na základě času a událostí. (MURRAY;2009)

### **2.1.4 Procesy PRINCE2**

PRINCE2 je projektovou metodologií založenou na procesech. PRINCE2 definuje proces jako set činností, které jsou navrženy tak, aby splnily určitý cíl. Každý proces má určité vstupy, které zpracovává a přeměňuje je na definované výstupy. PRINCE2 má celkově 7 procesů. Tyto procesy probíhají v rámci 4 druhů etap a ve 3 úrovních řízení (strategické řízení, operativní řízení a dodávání). (MURRAY;2009)



Obrázek 9 Procesy a Etapy v PRINCE2(MURRAY;2009)

### Před projektem

Projekt začíná nápadem, nebo jiným podnětem ke změně (nápadem či jiným podnětem může být vidina získání konkurenční výhody, nebo i vnější vlivy, jako např. změna legislativy). V této etapě se rozhoduje, zda je projekt životaschopný a rentabilní. Tyto základní informace se nacházejí v dokumentu zvaném projektový mandát. (MURRAY;2009)

### Nastavení projektu

V předchozí etapě padlo rozhodnutí, že projekt se může realizovat. V dalším kroku je tedy nutné tento projekt detailně naplánovat. V souvislosti s plánováním je potřeba také vyřešit finanční stránku projektu a nastavit kontroly, aby projekt probíhal přesně podle požadavků a očekávání sponzorů a uživatelů. V této fázi je také dále rozvíjen obchodní případ a dokumentace nastavení projektu. Na základě těchto dokumentů je projekt autorizován projektovým výborem. (MURRAY;2009)

### Následující dodávající etapy

Projektový výbor deleguje řízení projektu na denní bázi na projektového manažera v každé dodávající etapě. Během této etapy dochází k plánování práce na celou dobu trvání etapy, kontroluje se kontinuální dodávání produktu, a zda produkt splňuje definované kvalitativní požadavky. Projektový manažer musí zajistit, že projekt pokračuje podle schválených plánů a pravidelně reportuje o průběhu. Týmový vedoucí přidělují ostatním členům týmu pracovní balíčky a reportují vše projektovému manažerovi. Na konci každé dodávající etapy žádá projektový manažer o povolení začít další etapu. Reportuje, jak dopadla etapa a předává informace, zda je projekt stále životaschopný. Případně upraví obchodní případ. (MURRAY;2009)

### **Poslední dodávající etapa**

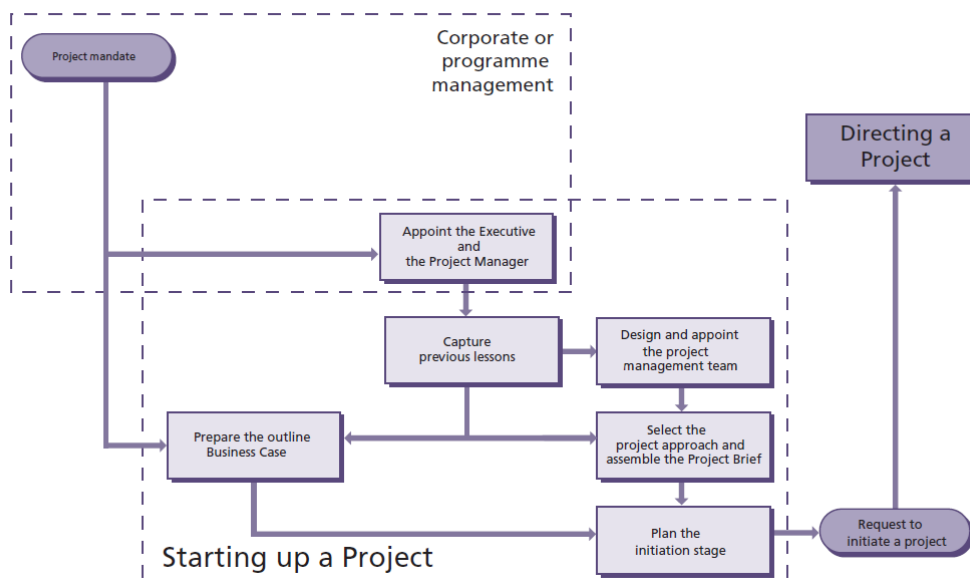
Jakmile projektový manažer získá schválení všech produktů, je nutné projekt ukončit. Uživatelé finálního produktu musí být schopni využívat produkty vytvořené během projektu ke svému účelu. Uzavírá se projektová dokumentace a provede se revize, jak si projekt vedl a jak si vedl oproti původním odhadům a plánům. Provede se analýza přínosů, obzvláště těch, které je možné měřit až po konci projektu. Zaevidují se všechny zbývající zkušenosti z projektu pro další příbuzné projekty. (MURRAY;2009)

### **Proces zahájení projektu**

Hlavním účelem procesu zahájení projektu je zajistit všechny předpoklady pro zahájení projektu. Výstupem toho procesu je mimo jiné odpověď na otázku, zda je projekt životaschopný a má smysl do něj investovat čas a zdroje. Během tohoto procesu se zabrání, aby neperspektivní (málo výhodné, špatně definované, příliš riskantní atp.) projekty nebyly zahájeny a naopak, aby všechny perspektivní projekty mohly být započaty. Tento proces zajišťuje aby:

- Projekt byl opodstatněný z obchodního hlediska (to je zdokumentováno v obchodním případě).
- Byla opatřena veškerá povolení pro nastavení projektu.
- Byly dostupné veškeré potřebné informace k sestavení a potvrzení rozsahu projektu (to je zavedeno v chartě projektu).
- Byly vyhodnoceny veškeré možnosti, jak může být projekt veden, a je stanoven projektový přístup.
- Byli jmenováni jedinci, kteří budou vykonávat veškeré činnosti potřebné pro nastavení projektu a byly definovány všechny řídicí role.
- Byly definovány veškeré práce, které budou potřeba pro nastavení projektu.
- Nebyl promarněn čas na nastavování projektu, který není životaschopný z pohledu rozsahu, času, přínosů atp.

(MURRAY;2009)



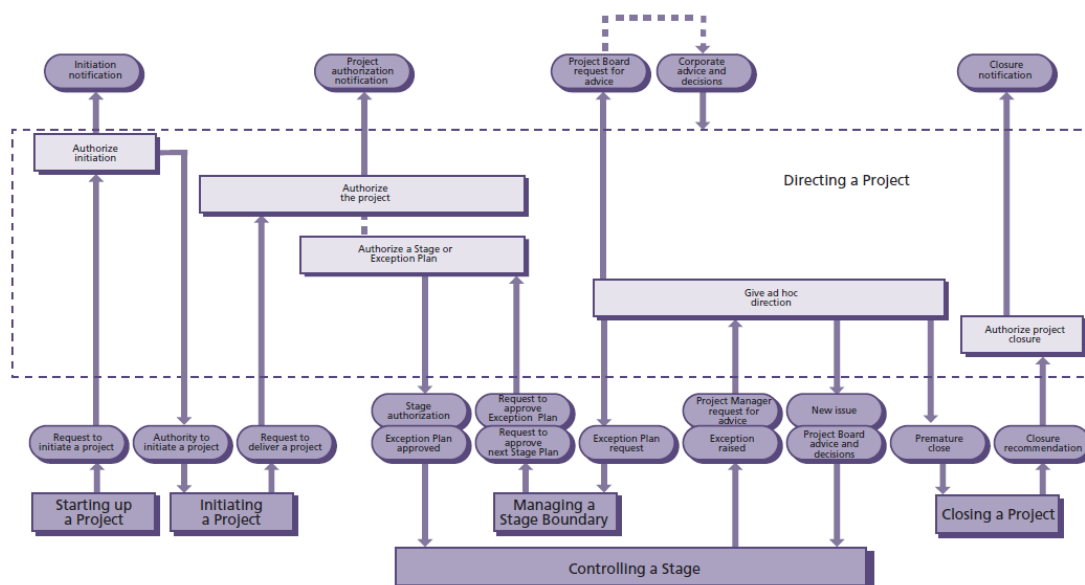
Obrázek 10 Proces zahájení projektu(MURRAY;2009)

### Proces řízení projektu

Účelem procesu je popis činností projektového výboru, který je odpovědný za úspěch projektu prostřednictvím přijímání klíčových rozhodnutí, kontrolou a delegováním řízení na denní bázi na projektového manažera. Proces dbá na to aby:

- Existovalo oprávnění zahájit projekt.
- Existovalo oprávnění dodávat produkty projektu.
- Směřování, řízení a kontrolování byly poskytovány po celou dobu běhu projektu, a aby projekt zůstal životaschopný.
- Existuje oprávnění projekt ukončit.
- Byly realizovány a revidovány plány pro přínosy projektu.

(MURRAY;2009)



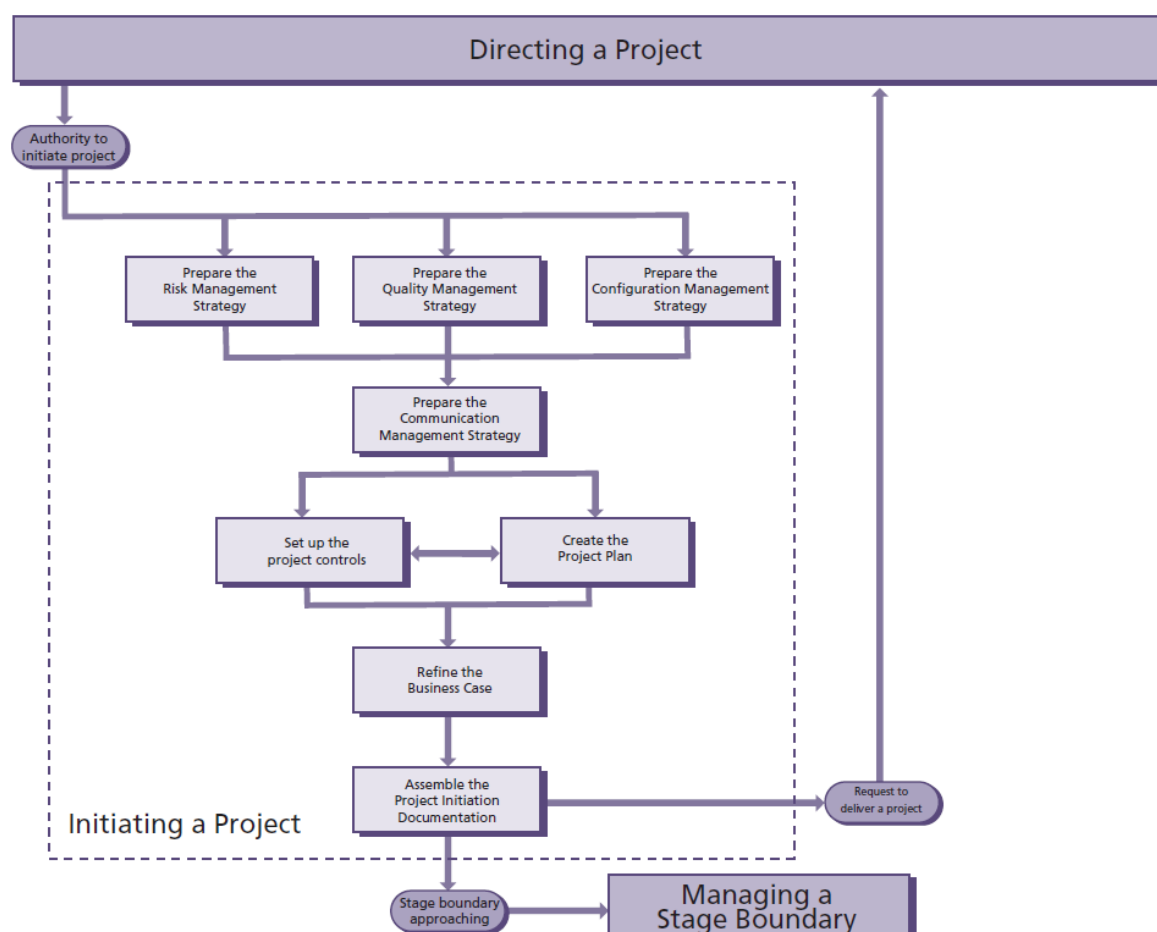
Obrázek 11 Proces řízení projektu(MURRAY;2009)

### Proces nastavení projektu

Účelem procesu nastavení projektu je vytvoření pevných základů pro projekt a pomoci organizaci pochopit projekt, jeho výstupy a přínosy, které má přinést. V rámci tohoto projektu je vytvořená také projektová strategie, jsou vypracovány detailní plány a je upravován obchodní případ. Proces se snaží zejména zajistit, aby organizace dobře chápala:

- Důvod realizace projektu, očekávané přínosy a s tím spjatá rizika.
- Rozsah projektu, čili co se musí udělat a jaké produkty musí být dodány.
- Kdy a jak budou produkty dodány a za jakou cenu.
- Kdo musí přijímat, nebo se jinak podílet na přijímání rozhodnutí, která ovlivňují projekt.
- Jak se dosáhne požadované kvality.
- Jak budou vytvořeny baseliny a jak budou kontrolovány.
- Jak budou identifikovány, posouzeny a kontrolovány otevřené body, rizika a změny.
- Jak bude monitorován a kontrolován průběh projektu.
- Kdo potřebuje jaké informace, v jakém formátu je potřebuje a kdy je potřebuje.
- Jak bude upravena metodologie řízení projektu, aby vyhovovala danému projektu (jeho velikosti, složitosti atp.).

(MURRAY;2009)



Obrázek 12 Proces nastavení projektu(MURRAY;2009)

### Proces kontroly etapy

Účel procesu kontroly etapy je zadání práce, která musí být odvedena, monitorovat tuto práci, vypořádat se s otevřenými body, reportovat průběh projektovému výboru a přijímat případné korektivní akce, aby bylo zajištěno, že projekt zůstane v nastavených tolerancích. Hlavním cílem tohoto procesu je zajistit, aby:

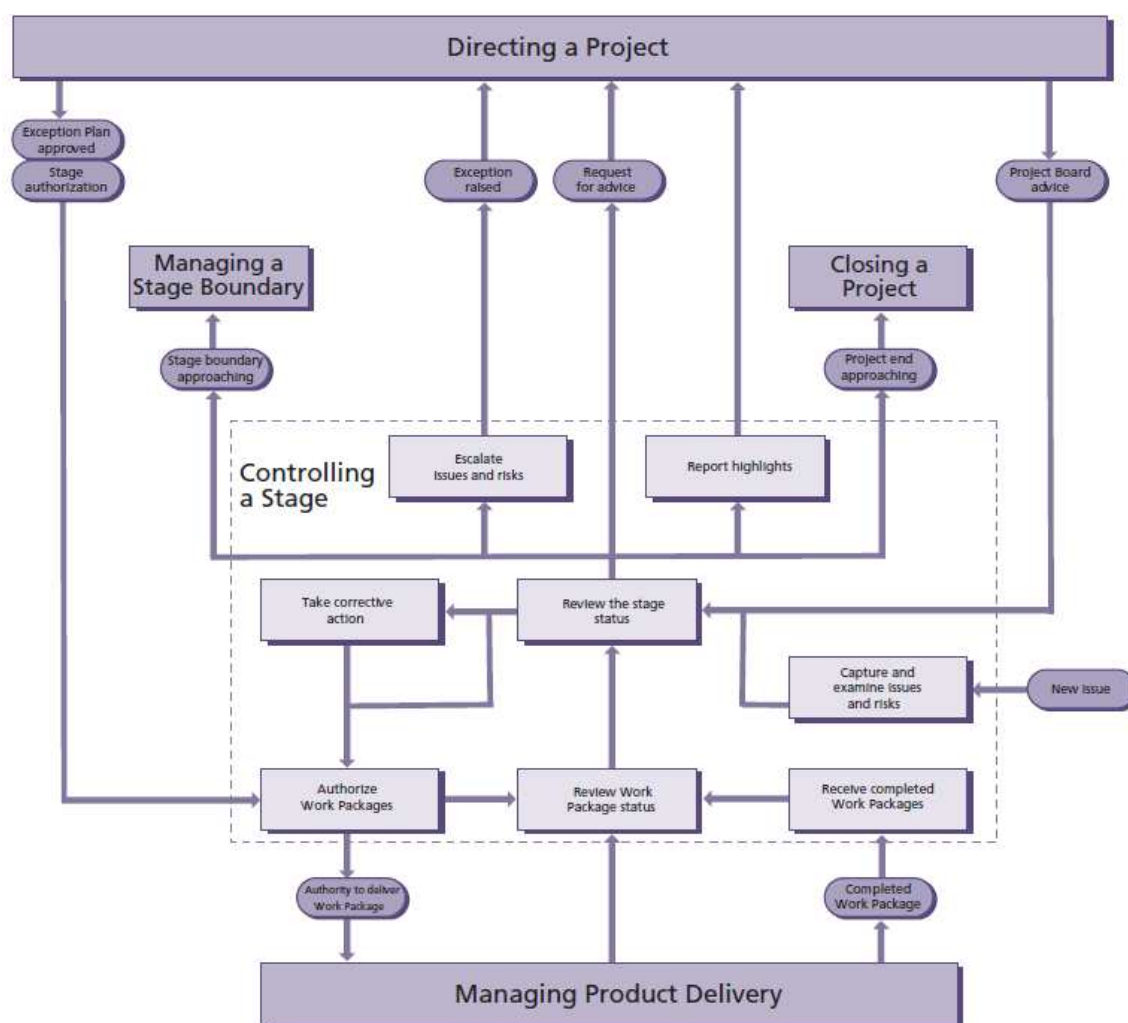
- Byla zajištěna pozornost na dodání produktů (či balíků práce) v dané etapě. Je nutné ohlídat, aby se odvedená práce a tím i výsledný produkt neodchyloval od dohodnutého balíku práce pro danou etapu (cílem je se vyhnout nekontrolovaným změnovým požadavkům).
- Rizika i otevřené body byly drženy pod kontrolou.
- Obchodní případ byl neustále revidován.
- Produkty realizované v dané etapě byly dodány v dohodnuté kvalitě, nákladech, pracnosti a v dohodnutém čase.

- Projektový tým byl zaměřen na dodávání produktu v nastavených tolerancích.

V rámci tohoto procesu jsou realizovány tyto činnosti:

- Balíky práce
  - o Povolení balíku práce.
  - o Revidování stavu balíku práce.
  - o Příjem dokončeného balíku práce.
- Monitoring a reportování
  - o Revize stavu etapy.
  - o Reportování důležitých událostí.
- Otevřené body
  - o Zachycení a prozkoumání otevřených bodů a rizik.
  - o Eskalace otevřených bodů a rizik.
  - o Přijmutí nápravného opatření.

(MURRAY;2009)



Obrázek 13 Proces kontroly etapy(MURRAY;2009)

### Proces řízení dodání produktu

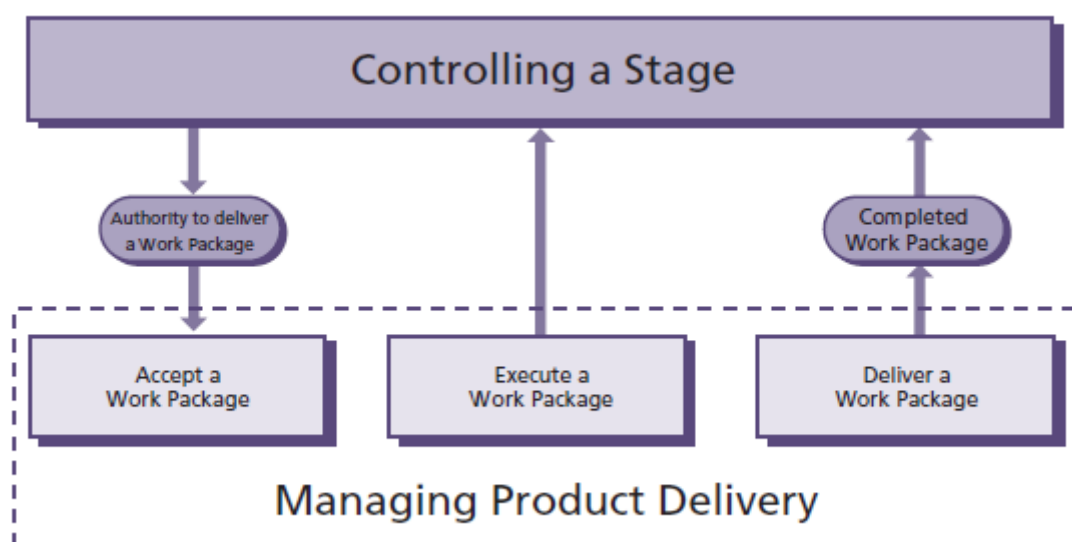
Účelem procesu řízení dodání produktu je kontrola spojení mezi projektovým manažerem a manažerem týmu. Toho se dosahuje nastavením formálních požadavků pro akceptaci, realizaci a dodání balíků práce. Hlavním zaměřením procesu je každodenní práce manažera na projektu. Cílem procesu je zajistit, aby:

- Práce přidělená týmům byla řádně sjednána a schválena.
- Manažeři týmu, členové týmu a dodavatelé porozuměli, co se má vytvořit a jaké jsou očekávané náklady, pracnost a čas.
- Plánované produkty byly dodány v požadované kvalitě.
- Byly projektovému manažerovi dodávány přesné informace o průběhu práce v sjednané frekvenci.

V rámci tohoto procesu jsou realizovány tyto činnosti:

- Přijetí balíku práce.
- Provedení balíků práce.
- Dodání balíku práce.

(MURRAY;2009)



Obrázek 14 Proces řízení dodání produktu(MURRAY;2009)

### Proces řízení přechodu mezi etapami

Účelem procesu řízení přechodu mezi etapami je umožnit aby projektový výbor byl zásoben dostatečným množstvím informací od projektového manažera tak, aby bylo možné provést revizi současné etapy, schválit plán následující etapy, revidovat upravený projektový plán a potvrdit opodstatněnost projektu při současných rizicích a přínosech. Tento proces se realizuje na konci každé dodávající etapy. Hlavním cílem procesu je zajistit aby:

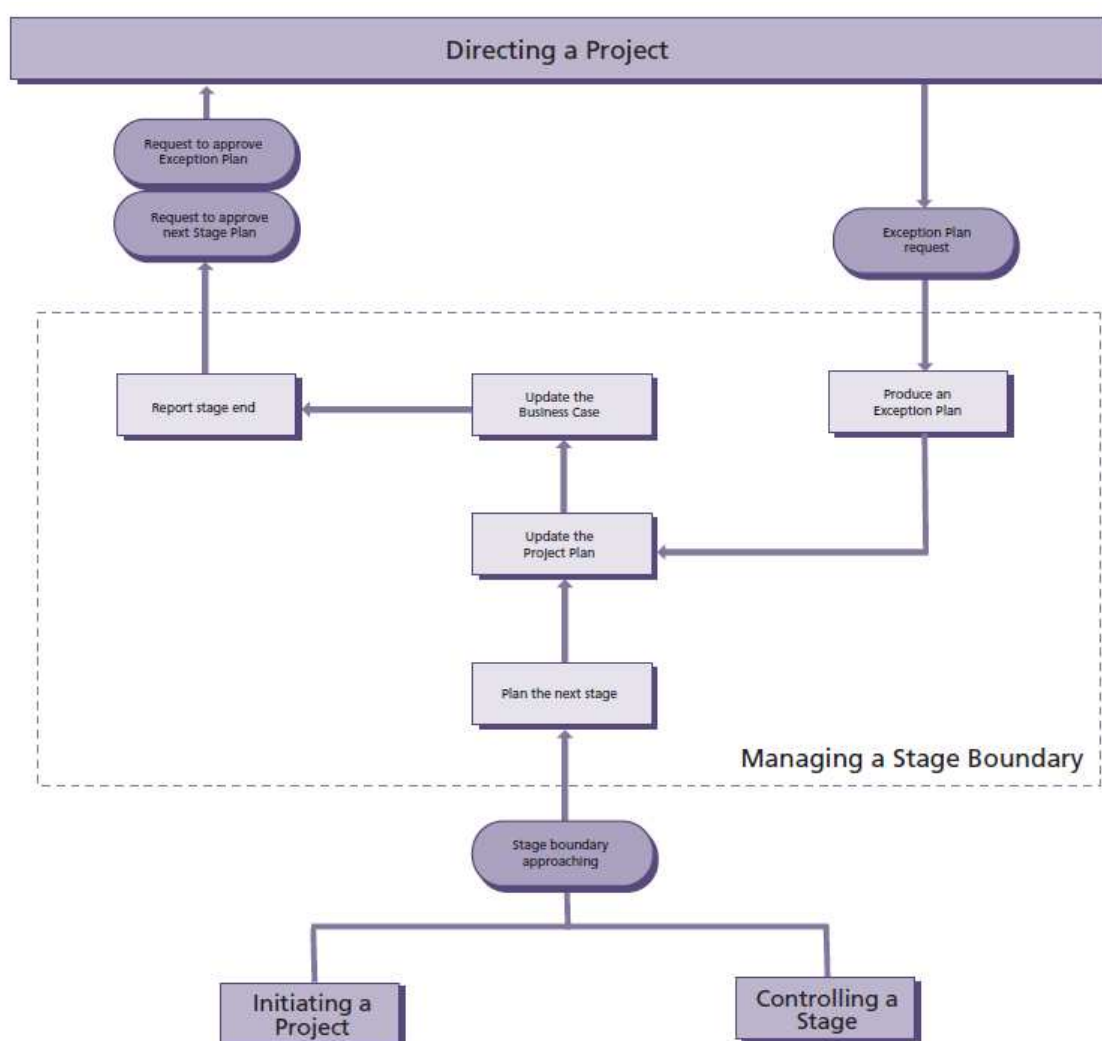
- Byl projektový výbor ujištěn, že všechny balíky práce naplánované pro současnou etapu byly dokončeny a akceptovány.
- Byl připraven plán pro další etapu.
- Byla revidována a v případě potřeby upravena dokumentace nastavení projektu (projektový plán, obchodní případ atd.).
- Byly dodány veškeré potřebné informace projektovému výboru, který na jejich základě rozhodne, zda je projekt i nadále životaschopný.
- Byly zaevidovány všechny informace a zkušenosti, které mohou pomoci realizaci následujících etap, celého projektu či nadcházejících projektů.

- Bylo požádáno o povolení začít novou etapou.

V rámci tohoto procesu jsou realizovány tyto činnosti:

- Plánování další etapy.
- Úprava projektového plánu.
- Úprava obchodního případu.
- Report o konci etapy.
- Evidence získaných poznatků.

(MURRAY;2009)



Obrázek 15 Proces řízení přechodu mezi etapami(MURRAY;2009)

### Proces ukončení projektu

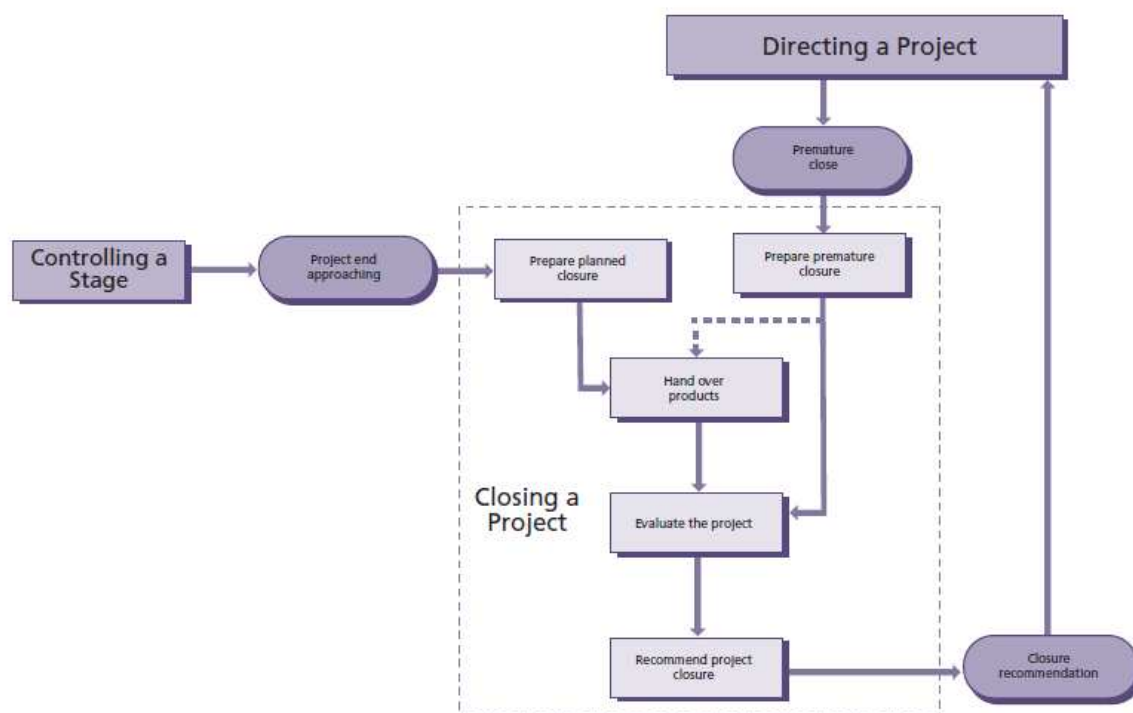
Účelem procesu ukončení projektu je poskytnout pevný bod, ve kterém jsou výsledné produkty akceptovány, a že cílů projektu vytyčených v dokumentaci nastavení projektu bylo dosaženo (včetně změn, které vyvstaly během běhu projektu a byly akceptovány). Hlavním cílem procesu je:

- Zkontrolovat akceptaci produktu uživatelem.
- Zajistit, aby zákaznicko prostředí podporovalo produkt, jakmile bude projekt ukončen.
- Revidovat výkonnost projektu oproti schváleným baselinám.
- Zhodnotit již realizované přínosy, aktualizovat předpověď zbývajících přínosů a plánovat zhodnocení těchto nerealizovaných přínosů.
- Zajistit pokrytí veškerých zbývajících otevřených bodů a rizik.

V rámci tohoto procesu jsou realizovány tyto činnosti:

- Příprava plánovaného uzavření projektu.
- Příprava předčasného uzavření projektu.
- Předání produktů.
- Vyhodnocení projektu.
- Doporučení projektu k uzavření.

(MURRAY;2009)



Obrázek 16 Proces ukončení projektu(MURRAY;2009)

## 2.2 Agilní metodologie

Agilní metodologie jsou asi nejdiskutovanějšími metodikami v posledních několika letech (alespoň co se IT týče). Nabízejí mnoho výhod oproti klasickým rigorózním metodikám, zejména v redukcí byrokracie v rámci celého projektu. Agilní přístupy mají mnoho zástupců, kteří pohlíží na rigorózní metodiky jako na dávno překonanou záležitost, která nemá v moderní době místo (u většiny projektů). Naopak své zastánce mají i rigorózní metodiky. Zástupci obou táborů mají různé důvody, proč zastávají tu či onu metodiku. Může jít zejména o negativní zkušenosti se zaváděním oné metodiky do provozu (např. při zavádění agilních metodologií se lze setkat s velkým odporem zejména u starších generací, které jsou zvyklé na zažité firemní procesy), neúspěch projektu pod danou metodikou, nebo se jim můžou přičít některé procesy a přístupy, které změna firmě přináší (např. u agilních metodik snížení byrokracie).

Níže je popsána historie agilních metodik, dále bude přiblížen základní dokument agilních přístupů a to Agilní manifest. Následuje výběr agilních metodik podle výzkumu Version-one a jejich popis.

## 2.2.1 Historie agilních metodologií

Stejně jako většina agilních metodik, přístupů a procesů sahá počátek agilních metodik již do 50. – 60. let 20. století, kdy se zaváděly různé nové přístupy pro zefektivnění a zeštíhlení výroby (většina metodik má své kořeny právě ve výrobním průmyslu). Tehdy např. vznikl Lean Management (který je předchůdce Lean Developmentu) ve firmě Toyota. Lean Management vznikl „jako alternativa k hromadné výrobě, v prostředí které vyžadovalo vysokou úroveň flexibility a postrádalo finance na nákladné investice. Provádí komplexní organizaci vývoje a výroby produktu, dodavatelů a kontakty se zákazníkem tak, aby při lepším plnění zákaznickova požadavku bylo zapotřebí méně lidského úsilí, prostoru, kapitálu a času - a přitom produkty mají mnohem lepší kvalitu než v hromadné výrobě.“ (BORDÁS, 2006). Podle (BORDÁS, 2006) jsou dva hlavní pilíře Lean Managementu JIT (Just in time) a JIDOKA. JIT (což se dá přeložit jako „právě v čas“) je princip výroby, kdy se materiál dostane na každé pracoviště zrovna, když je potřeba (nehromadí se, a to vede i ke snížení materiálu na skladě a tím pádem i ke snížení finanční náročnosti výroby). JIDOKA (což znamená „automatizace s lidskou inteligencí“) znamená, že stroj dokáže rozeznat špatný výrobek a umí na takovou situaci zareagovat, ať už zastavením výroby či odkloněním zmetků (např. pomocí čidel zjistí špatné rozměry, a tak se zmetek přesměruje do kontejneru).

Z těchto principů vychází právě agilní metodologie užívané v IT průmyslu, které se také snaží o zeštíhlení „výroby“ a zefektivnění nákladů. Základy agilních metodik v IT byly položeny již v 80 letech 20. století, ale nejmarkantnější rozmach zaznamenaly v roce 2001, kdy vznikl tzv. Agilní manifest, který sepisuje základní myšlenky agilního přístupu.

## 2.2.2 Agilní manifest

V roce 2001 mezi 11. - 13. únorem se v lyžařském letovisku Snowbird v Utahu sešlo 17 lidí, kteří se zabývali myšlenkou agilních přístupů, aby našli společnou řeč. Výsledkem jejich jednání byl Agilní manifest. Tito lidé reprezentovali různá odvětví agilu jako SCRUM, Crystal, FDD (Feature-Driven Development) a jiné a jejich záměrem bylo nalezení alternativního způsobu vedení dokumentace a změna přístupu dosavadních těžkých přístupů. Skupina sama sebe nazývala Agilní aliancí. Agilní metodologie mají čtyři základní hodnoty a dvanáct principů.

Hodnoty, které agilní metodologie vyznávají, jsou v (BECK, 2001) definovány takto:

- Lidé a jejich interakce oproti procesům a nástrojům.
- Fungující software oproti obsáhlé dokumentaci.
- Spolupráce se zákazníkem oproti vyjednávání o smlouvě.
- Reakce na změny oproti držení se plánu.

**Lidé a jejich interakce oproti procesům a nástrojům:**

Při praktikování agilních metodologií je nutné se zaměřit na všechny lidi v týmu a jejich interakci oproti procesům a nástrojům, což lze chápat jako zapojení více „lidštějšího“ přístupu k práci na projektu. Pokud je nutné něco vyřešit je lepší, když se vše řeší osobně (osobní styk také může působit lépe, než jen úkolovat prostřednictvím mailů či využíváním aplikací pro práci týmů), než posílat obsáhlé maily a do puntíku dodržovat nastavené procesy (které se mohou v krizových situacích jevit jako kontraproduktivní). Takto se lze ujistit, že člen týmu chápe, co se po něm chce, případně se zadání upřesní (či vyjasní). Díky tomu se ušetří i velké množství času, který by byl promrhan při zdlouhavé elektronické komunikaci. Také tento přístup pomáhá zlepšit klima na pracovišti, když se používá politika „otevřených dveří“ či v „open space“ prostorách.

**Fungující software oproti obsáhlé dokumentaci:**

Dodávaný software by měl být vždy dobře zdokumentován, což dokáže ušetřit čas a náklady při předávání (a konflikty, jestli to, co bylo dodáno, opravdu má být tak, jak to funguje v daném okamžiku a proto musí být vše uvedeno v dokumentaci). Nemělo by se však stát, že se samotné dokumentaci věnuje více prostoru a pečlivosti než aplikaci samotné. Dokumentace by měla být svědomitě vedena, ale ne na úkor produktu. Jelikož jsou agilní metodiky zaměřené na zákazníka, je cílem dodat fungující produkt v odpovídající kvalitě a v dohodnutém čase. Pokud se zákazníkovi předá bezchybná dokumentace bez kvalitně fungujícího produktu, nebude dílo s největší pravděpodobností akceptováno.

**Spolupráce se zákazníkem oproti vyjednávání o smlouvě:**

Dodavatel by se měl zaměřit na vybudování dobrého a produktivního vztahu se svým zákazníkem. Pokud mezi nimi zavládne téměř přátelský vztah, je velice pravděpodobné, že si zákazník opět zvolí jako dodavatele (ať už dalšího systému či jen zajištění aplikační podpory) společnost, se kterou má velice dobré vztahy. Dobrý vztah se zákazníkem má také za následek hladší průběh při řešení problémů (např. dodavatel se zpozdí s dodávkou další části dodávaného softwaru a toto zpoždění by mohlo být smluvně penalizováno. Díky dobrým vztahům nebude zákazník požadovat penále a vše se vyřeší diplomaticky. Nebo z opačné strany dodavatel se rozhodne dodat malou funkčnost navíc a v podstatě zadarmo pokud si zákazník uvědomil, že na tuto funkčnost při sestavování rozsahu zapomněl). Chování týmu vůči zákazníkovi by tudíž mělo mířit na neustále utužování vztahů, protože produkt se dodává zákazníkovi a tomu se musí vyhovět. Nemělo by se stát, že se vztahy pokazí neustálým vyjednáváním o smlouvě, když se snaží podchytit každou (byť jen teoretickou) možnost a riziko na úkor spolupráce. Pokud zákazník potřebuje aplikaci (nebo její část) do určitého data a podepisování další dílčí smlouvy se bude pravděpodobně protahovat, lze začít s realizací ještě dříve než je podepsaná samotná smlouva (toto platí jen u důvěryhodných a prověřených zákazníků, se kterými má společnost jenom dobré zkušenosti). Vedení dodavatelské společnosti se musí umět diplomaticky dohodnout na smlouvě a nedělat zbytečné obstrukce. Samozřejmě je nutné mít smlouvu perfektně připravenou a měla

by chránit jak dodavatele, tak i zákazníka (není smyslem této hodnoty agilních metodologií, aby bylo sestavení smlouvy nedbalé a jenom se slepě zaměřovalo na spolupráci se zákazníkem).

### **Reakce na změny oproti držení se plánu:**

Tým (či týmy), které pracují na projektu pro zákazníka, musí mít na paměti, že produkt má být pro zákazníka, takže musí naslouchat právě jeho přání. Je nutné mít dobře definovaný rozsah před zahájením samotného projektu, ale stává se velice často, že zákazník mění své priority. Některé funkcionality se vypustí (např. kvůli legislativním změnám je daná funkce již irelevantní), některé přibudou (zákazník zjistí, že onu funkci potřebuje až v pozdější fázi projektu) a některé budou změněny (toto může např. být zapříčiněné nečekanou změnou v jiné aplikaci, se kterou vyvíjená aplikace komunikuje, nebo když při testování bylo zjištěno, že výstup dané funkce úplně nevyhovuje potřebám uživatele). Pro spokojenost zákazníka je nutné správně reagovat na veškeré případné změny. Zde je hlavní rozdíl oproti rigorózním metodikám, které se drží striktně plánu a výstup se představí až na konci projektu. Agilní metodologie vyznávají inkrementální vývoj a dodávání v malých a pravidelných dílech produktu (v tzv. iteracích). Pokud se zákazník rozhodne pro změnu v produktu, dodavatel by měl být ochoten změnový požadavek přijmout. Jak bude se stávajícím rozsahem naloženo je již na každé společnosti. Nový požadavek se může realizovat na úkor jiného (nabízí se hlavně 2 možnosti a to: automaticky nerealizovat požadavek s nejnižší prioritou, nebo nerealizovat požadavek dle výběru zákazníka), může být navýšen rozpočet projektu (nebo nová funkcionality bude zaplacená zvlášť a vedena jako nový menší projekt), nebo pokud jde o funkcionality s triviální pracností, v tomto případě lze uvažovat i o realizaci na náklady dodavatelské společnosti za účelem zlepšení vztahu se zákazníkem.

Dvanáct principů definované (BECK, 2001) jsou tyto:

- 1) Uspokojení zákazníka díky brzkému a kontinuálnímu dodání hodnotného softwaru.
- 2) Uvítání změnových požadavků a to i v pozdějších fázích vývoje. Agilní procesy míří na lepší konkurenční výhodu zákazníka.
- 3) Časté dodávání fungujícího softwaru v rozmezí od několika týdnů až do několika měsíců, přičemž se preferuje kratší interval.
- 4) Business a vývoj musí spolupracovat na denní bázi po celou dobu trvání projektu.
- 5) Realizace projektu v okruhu motivovaných lidí (těmto lidem je nutné dodat odpovídající prostředí a podporu a věřit, že dokážou splnit svojí práci).
- 6) Nejeфективnější metodou pro zadávání úkolů a přenos informací je tvář v tvář.
- 7) Nejobektivnější metrikou pro měření stavu projektu je fungující software.

- 8) Agilní procesy podporují udržitelný rozvoj (sponzoři, vývojáři a uživatelé by měli být schopni udržet stále tempo).
- 9) Kontinuální snaha o udržení technické dokonalosti a dobrého designu pomáhá vylepšit agilitu projektu.
- 10) Zjednodušování je nezbytné (umění maximalizovat neodvedenou práci).
- 11) Nejlepší architektury, požadavky a design vychází ze samoorganizovaných týmů.
- 12) Tým v regulérních intervalech provádí reflexi svého fungování a upravuje podle toho své chování.

Níže jsou rozepsány jednotlivé principy.

- 1) Výhoda agilních metodologií spočívá v inkrementálním (přírůstkovém) vývoji. Raději než dodat celý produkt na konci projektu a riskovat větší množství chyb, nebo čelit všem změnovým požadavkům najednou, agilní přednosti upřednostňují kontinuální dodávání malých částí produktu. Produkt je dodáván postupně čímž se mohou dříve projevit skryté chyby, nebo naopak bude menší část fungovat bez problémů a lze stavět další rozvoj na pevných základech. Zákazník také dříve zjistí, zda dodanou komponentu potřebuje, nebo zda ji potřebuje pozměnit, aby lépe refletovala jeho potřeby. Tyto změny v projektovém plánu se tak nenahrnou najednou, ale budou rozprostřeny během celého projektu kontinuálně, čili bude menší tlak na jejich vypořádání. Velice důležitý je také fakt, že zákazník ihned vidí hodnotu investovaných peněz a lépe se odhaduje současný stav projektu (i časový trend projektu).
- 2) Agilní metodiky se zaměřují především na konkurenceschopnost zákazníka a jeho spokojenost. Tudíž se projekt přizpůsobuje aktuálním požadavkům a potřebám zákazníka. Agilně řízený projekt vítá změnové požadavky v průběhu celého projektu (i v pozdějších fázích projektu). Jak již naznačovaly výše rozepsané hodnoty agilních metodik, je nutné dodávat kvalitní produkt, který dokáže vyhovět aktuálním požadavkům zákazníka, které se mohly změnit od začátku projektu (ať už jsou způsobeny legislativními změnami, změnami na trhu nebo objevením nové technologie). Z tohoto důvodu je třeba produkt neustále upravovat podle přání zákazníka (je nutné, aby zákazník za takovou změnu zaplatil, ať už finančně či prostřednictvím vyškrtnutí jiného požadavku).
- 3) Jedním z určujících znaků agilních metodologií je vývoj v iteracích. Iterace (vývojový blok) může být dlouhá od jednoho týdne do několika měsíců. Obecně se doporučují kratší iterace ideálně 2 - 3 týdny. Na tuto dobu se určí objem práce, který by měl být splněn. Každý správný projekt by měl mít týmy rozdělené dle specializací (vývojový tým, analytický tým, testovací tým atd.). Díky iteračnímu vývoji lze na první iteraci zadat práci analytikům a ve druhé již nechat programátory vyvíjet onu zanalyzovanou část, zatímco analytici mohou pracovat na jiných požadavcích (ten-

to princip lze spíše používat v kratších iteracích, v delších se spíše rovnou dodává již otestovaná nová funkčnost, která je zanesená do dokumentace).

- 4) Agilní metodologie také vyžadují, aby pracovníci odpovědní za business komunikovali s vývojem na denní bázi. Pro zlepšení kvality dodávaného produktu je zapotřebí, aby programátoři zcela porozuměli, co se vlastně po nich chce. V porozumění tomu co si přeje zákazník a jak to vidí (či jak si to vyložil) vývojář, mohou vznikat rozkoly. Pro přeložení tohoto požadavku programátorům, nebo i ve druhém směru k zákazníkovi by měl sloužit business tým. Tento tým také pomáhá zlepšovat vztahy se zákazníkem a funguje jako prostředník. Z tohoto důvodu je důležité, aby se tyto týmy (respektive jejich zástupci) scházely každý den a ujasňovaly si zadání (upřesnění pro zákazníka, předání dotazu, zákazník může být částí business týmu).
- 5) Pro úspěch projektu je důležité mít krom jiných faktorů spokojené a motivované členy týmů, jelikož jsou to právě oni, kdo vytváří daný produkt. Pokud zaměstnanec není spokojený, je velice pravděpodobné, že se jeho nespokojenost promítne i do kvality odvedené práce. Je na vedoucím týmu (potažmo projektovém manažerovi), aby dokázal udržet své podřízené motivované a spokojené po celou dobu trvání projektu. Je důležité, aby bylo členům připomínáno, proč dělají, co dělají a jaký je smysl výsledného produktu. Dalším faktorem spokojenosti je příjemné prostředí, ve kterém je vše co může tým potřebovat. Pokud se tým nachází v tmavé kanceláři, nemá poblíž žádné občerstvení, může to mít neblahý dopad na morálku. Proto by mělo být pracoviště vybaveno jak funkčně (pokud lze průběh projektu vizuálně prezentovat, nebo pokud má tým k dispozici tabuli s rozdělenými úkoly na danou iteraci, bude se týmu lépe pracovat, jelikož budou mít lepší přehled o jejich postupu) tak esteticky. V poslední řadě by měl každý člen týmu cítit, že má plnou důvěru svých nadřízených. Pokud mu vedoucí zadá úkol, tak musí vědět, že má důvěru v jeho schopnosti a že dokáže svůj úkol zvládnout ke spokojenosti zákazníka.
- 6) Pokud se řekne agilní metodologie, mnohým se vybaví velice osobní přístup. Spíše než elektronicky se vše řeší ústně. Pokud vyvstane v průběhu projektu problém (např. nepochopení danému úkolu, či nejasnost v kompetencích mezi členy týmu), doporučuje se vše řešit tváří v tvář. Vyřešit obdobný problém elektronicky je zdoluhavější a výsledky nejsou vždy tak dobré, jako když se stejný problém řeší osobně. Elektronická forma komunikace navíc může budit zdání jakési anonymity a působí dosti neosobně. Osobní kontakt na pracoviště pomáhá také ke zlepšení vztahů na pracovišti (nevznikne dojem „skrývání se za monitorem“). Pro agilní přístupy je typická politika otevřených dveří.
- 7) Během každého projektu je nutné měřit, v jaké fázi se projekt nachází a kolik práce zbývá ještě odvést. Lze nastavit velké množství metrik, které budou mít různou vypovídací hodnotu, ale nic neřekne o aktuálním stavu, ve kterém se projekt nachází více než týmem dodávaný produkt. Pokud je cílem projektu dodat účetní systém, který má celkem obsahovat 5 modulů a byly již dodány (a úspěšně otestovány) 3

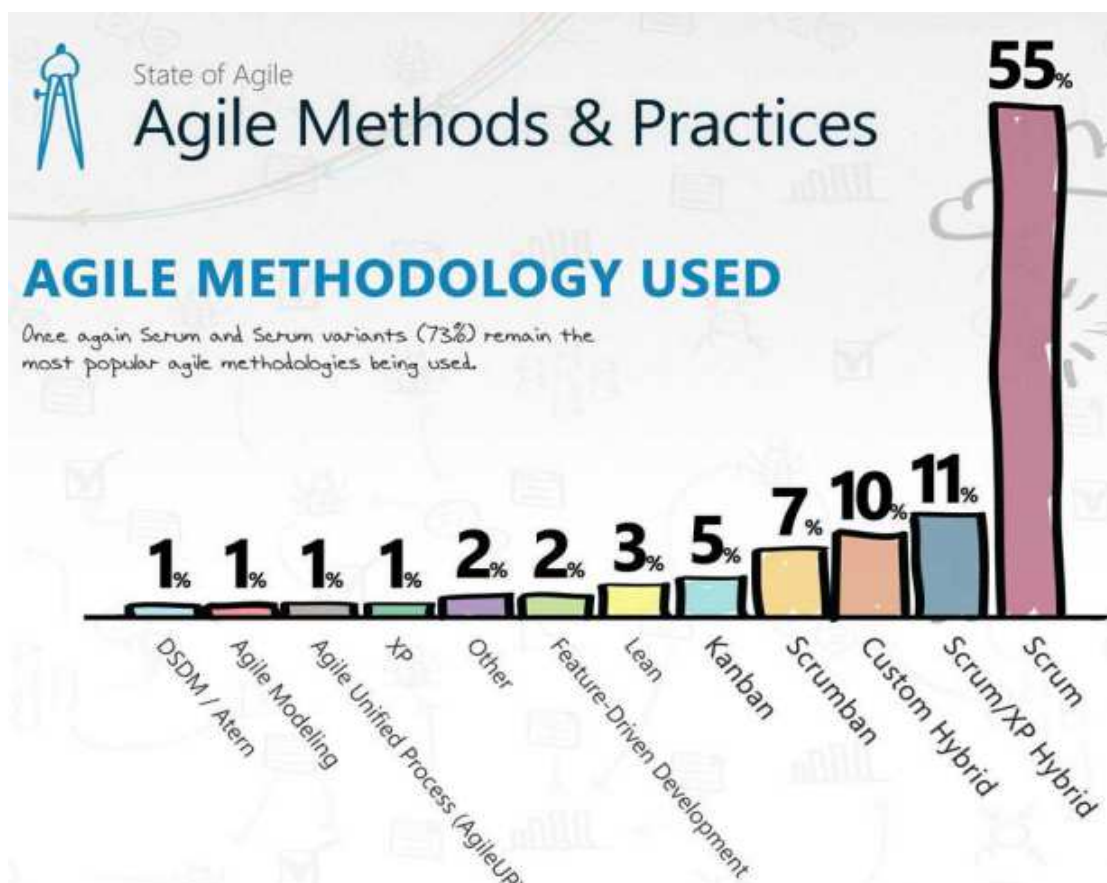
moduly, lze říci, že projekt je na 60% hotov. Tato informace má větší vypovídací hodnotu, než např. podepsaný předávací protokol vztahující se k nějakému fakturačnímu milníku. Podepsaný protokol může znamenat, že máme také 60% projektu hotovo, ale většinou se objeví další chyby, nebo testování není u konce, takže ve finále je produkt hotový z menší části.

- 8) Z principů inkrementálního a iteračního vývoje lze vyvodit, že agilní metodologie podporují udržitelný vývoj: Právě definované úkoly na každou iteraci a postupné dodávání pomáhá udržet vývoj v nastaveném tempu. Toto tempo by měla být schopná udržet každá role napříč projektem. Sponzor by měl být schopen stále stejně efektivně spolupracovat s business týmem, reagovat na požadavky ostatních týmů a vybírat požadavky do produktového backlogu. Vývojář by měl být schopen efektivně vyvíjet jednotlivé tasky atd.
- 9) Aby se dalo kvalitně reagovat na změny, a připravovat software na implementaci dalších funkcionalit je nutné, aby byl produkt kontinuálně udržován v perfektní technické kondici (okomentovaný kód, dodržování správných programátorských praktik). Dále je nutné udržovat i velice kvalitní design. Pokud nemá software dobře vytvořený design, zvyšuje se šance na nekvalitní produkt. Programátor musí mít kvalitní design, aby věděl, čeho se má držet a minimalizoval úpravy z vlastní hlavy, které se později mohou ukázat jako problematické.
- 10) Pro agilní metodologie často platí přísloví, že méně je někdy více. Při práci na projektu by mělo být cílem snaha o co nejefektivnější práci (čili nejlepší výsledky za co nejméně práce). Pokud se funkce softwaru dá udělat jednoduše, nebo je obecně její funkčnost známá (zdokumentována na veřejně přístupném místě, či realizována v rámci jiného projektu), nemá smysl pokoušet se jí vymýšlet znovu (může však nastat i případ, kdy se funkce značně zefektivní a taková změna je žádoucí), ale použije se (a potřebně upraví) již existující funkce. Stejně tak, pokud existuje v dané aktivitě nejlépe přijímaná metoda (tzv. Best Practices), je lepší použít tento přístup, než se snažit vymýšlet úplně nový přístup na úkor projektu. Tento přístup platí i pro přidělování úkolů členům týmů. Nemá smysl např. zadávat vytvoření datového modelu pracovníkovi, který není zvyklý tyto modely vytvářet, nýbrž zadat to nejefektivnějšímu členovi týmu. Toto samozřejmě nemusí platit vždy, např. pokud se snažíme zaškolit nového zaměstnance a počítáme s tím, že se nám vynaložený čas vrátí v podobně stejně efektivního člena týmu, nebo pokud jeden ze členů odchází z dané společnosti.
- 11) Tento princip opět vychází z osobního přístupu praktikovaného v rámci agilních metodologií. Každý tým by měl být zodpovědný za svoji práci, kterou by měl zvládnout bez většího dozoru (jistý dohled od projektového manažera je však nutný). Nejlepší výsledky tak dosahuje tým, do kterého není příliš zasahováno zvenčí, kde se dokážou členové týmu dohodnout a mají rozdělené role a povinnosti.

- 12) Posledním principem agilních metodologií je neustálé zlepšování. Po konci každé iterace je nutné provést reflexi odvedené práce. Zde je důležité poukázat na nedostatky v kvalitě produktu, ale také na nedostatky v komunikaci mezi členy týmu a mezi ostatními týmy. Pokud vyjde najevo, že nastavené procesy nefungují, je nutné je zlepšit či změnit. Neustálým zlepšováním procesů se zajišťuje trvale vysoká produktivita a lze tak lépe reagovat na změny v prostředí atd.

### 2.2.3 Výběr Best Practices agilních metodologií

Jelikož existuje celá řada agilních metodologií a neustále se objevují nové, je nutné vybrat pro účely této práce několik nejvhodnějších. Pro každou společnost či tým se hodí různé metodologie a každá z nich má své výhody a nevýhody. Pro tento výběr poslouží studie od společnosti VersionOne. Na svých stránkách (VERSIONONE;2015) se tato společnost prezentuje jako jediná společnost dodávající software, která je věrná agilním metodologiím již od jejich počátku. Zabývají se vytvářením nástrojů pro podporu agilních společností. Každý rok vydávají studii o využívání agilních metodologií. Je zde dotazováno 100 firem, které více či méně využívají agilní metodologie, což bude sloužit jako vzorek pro stanovení nejvyužívanějších metodologií. Z těchto vybraných agilních metodologií budou extrahovány Best Practices, které budou sloužit jako základ praktické části této práce.



Obrázek 17 Využívání jednotlivých agilních metodologií podle (VERSIONONE;2013)

Na první pohled je zřejmé, že nejvyužívanější agilní metodologií je Scrum a jeho klony. Pro Scrum hovoří jeho jednoduchost, efektivnost a zejména možnost použití v širokém spektru projektů. Dále se oproti ostatním rokům zvýšilo procento využívání Kanbanu (či jeho kombinace se Scrumem - Scrumban).

## 2.2.4 Popis vybraných Best Practices

Níže jsou rozepsány jednotlivé vybrané Best Practices agilních metodologií, vybrané podle průzkumu VersionONE (Scrumban není rozepsán, jelikož obsahuje kombinaci praktik ze Scrumu a Kanbanu a nepřichází s žádnou novou praktikou, taktéž AUP, protože se jedná pouze o metodologii RUP rozšířenou o praktiky z XP a agilního modelování). U každé agilní metodologie bude výpis hlavních praktik, kterými je metodologie definována a jejich stručný popis. Tyto praktiky budou posléze v další části práce integrovány do metodologie PRINCE2. Nebyly vybrány všechny principy a praktiky ze všech uvedených metodologií, a to z důvodu jejich přílišné obecnosti, vágnosti, nebo použitelnosti jen v určitém aspektu projektu (např. lze vztáhnout jen na programátory).

## Scrum

- **Sprint** – Sprint je časový úsek, ve kterém probíhá práce na vybraných User stories z backlogu (pro daný sprint). Cílem je potencionálně implementovatelný přírůstek produktu. Sprints v rámci projektu mají identickou délku a následující sprint začíná ihned po skončení sprintu předchozího. Sprints trvají zpravidla měsíc, nebo i kratší dobu.
- **Backlog** – V rámci metodologie Scrum je definováno několik backlogů a to především produktový (Product Backlog) a backlog pro aktuální sprint (Sprint Backlog). Jedná se o seznam požadavků (seřazený podle priority), který byl sestaven ve spolupráci s vlastníkem produktu (Product Owner). Jednotlivé požadavky jsou zpravidla vyjádřeny pomocí User Stories. Jednotlivé požadavky jsou poté z tohoto seznamu vybírány do backlogu pro následující sprint. Důležitou vlastností backlogu je fakt, že nikdy není úplný, a že se jedná o živý dokument.
- **Denní Scrum** – Jedná se o každodenní schůzky na začátku pracovního dne. Měly by být realizovány ve stoje (tzv. Stand up meetingy) a měly by trvat 15 minut. Hlavním účelem této schůzky je synchronizace členů týmů a jejich aktivit a naplánování práce na dalších 24 hodin. Provádí se kontrola realizované práce od poslední schůzky (tj. z minulého dne) a odhad pracnosti další práce. Každý člen týmu musí odpovědět na 3 otázky (Co jsem udělal včera? Co budu dělat dnes? Jaké jsou překážky v mé práci?).
- **Plánování sprintu** – Úzce souvisí s backlogem a sprintem. Jedná se o schůzku, při které se plánuje balík práce na následující sprint. Tato schůzka by měla být časově ohraničená a to při měsíčním sprintu maximálně 8 hodin. Jsou vybrány požadavky (na základě priorit v produktovém backlogu) a je vytvořen jejich odhad pracnosti. Výstupem je sprintový backlog, čili seznam požadavků, které budou realizovány během sprintu a utvoří implementovatelný přírůstek produktu.
- **Vyhodnocení sprintu** – Po skončení sprintu se koná schůzka, při které se vyhodnocuje právě ukončený sprint. Tato schůzka by měla při měsíčním sprintu trvat maximálně 4 hodiny. Na schůzce by měli být přítomni všechny zainteresované strany. Vlastník produktu poté popisuje, které požadavky z produktového backlogu byly při sprintu dokončeny. Vývojový tým poté sděluje dojmy ze sprintu (co se dařilo, co se nedařilo a jak byly problémy vyřešeny). Nakonec vývojový tým prezentuje dodaný přírůstek produktu.
- **Retrospektiva sprintu** – Po skončení vyhodnocení sprintu a před plánováním nového sprintu se koná retrospektiva sprintu. Tato schůzka by měla při měsíčním sprintu trvat maximálně 3 hodiny. Zde se jedná o schůzku Scrum týmu. Je zde rozebíráno, které aspekty minulého sprintu fungovaly dobře, které nefungovaly a u kterých je možné docílit zlepšení. Výstupem je vytvoření plánu pro implementaci navržených změn.

- **Burndown chart** – Jedná se o diagram, který zachycuje trendy jednotlivých sprintů. Z tohoto diagramu se lze dozvědět, kolik User Stories bylo odbaveno za každý sprint a kolik jich celkově zbývá.
- **Definice toho co je hotovo** – Definice toho co je hotovo (Definiton of Done), je soubor standardů, vlastností a metrik, které určují, zda je možno User Story prohlásit za splněnou. Určuje jak požadovanou funkcionalitu, tak i kvalitativní nároky. Definice toho, co je hotovo, se používá jako minimální základ při tvorbě produktu.

(SUTHERLAND;2014), (SADDINGTON;2013)

## Kanban

- **Vizualizace workflow** – Pro lepší přehled toho, jak workflow funguje a pro jeho případnou inovaci je vhodné mít vizuální pohled na celou problematiku. Hlavní přínos této praxe je v tzv. Kanban tabuli. Ve sloupcích jsou jednotlivé části workflow, kterými prochází jednotlivé User Stories (nebo tasky). Myšlenka je taková, že každý kdo splnil předchozí úkol, si může vzít další z určitého sloupce.
- **Limitace rozdělané práce** – Práce v jednotlivých částech workflow (na Kanban tabuli reprezentováno sloupci) by měla být omezována na předem určený počet. Cílem je, aby nebylo najednou rozděláno příliš požadavků a na nových se začalo pracovat, až když jsou odbaveny předchozí požadavky
- **Řízení workflow** – Účelem Kanbanu je zavedení změn, tudíž jeden z hlavních aspektů bude řízení workflow. Jde o v podstatě nekonečný proces. Po monitoringu současného workflow je nutné identifikovat slabé místo a implementovat změnu. Po implementaci změny se opět provádí monitoring celého workflow, aby bylo možné zjistit, zda efekt implementovaného opatření měl pozitivní či negativní dopad.
- **Zajistit porozumění procesu** - Aby bylo možné zavést změnu, je nutné, aby proces či část procesu, který je nutné podrobit změnám, byl zcela pochopen. Proto je nezbytné, aby procesu rozuměli všichni členové týmu. Jakmile každý rozumí procesu, je potřeba, aby pochopili i to, co je cílem změny a proč je tato změna realizována.
- **Společně inovovat** – Pro co nejefektivnější inovaci je vhodné, aby se na procesu podíleli všichni jako tým s používáním vědeckých metod a přístupů. Důležitým aspektem je neustálá inovace (po malých částech, ale neustálá).

(WESTER;2016), (ANDERSON;2010)

**Lean**

- **Eliminace odpadu** – Odpadem se v tomto případě myslí redundantní funkcionality a kód, tj. vše, co z pohledu zákazníka nepřináší žádnou hodnotu. Dále je nutné eliminovat zbytečnou byrokracii a hlavně mít definované požadavky.
- **Rozvíjet znalosti** – Pro správný průběh projektu je nutné udržovat zpětnou vazbu směrem od zákazníka. Účelem je zjistit, zda nastavený systém je ten správný (a nikoli zda je současný systém správně používán). Jde o udržování a vytváření znalostí.
- **Rozhodovat se co nejpozději** – Tato praktika vychází z jedné z hlavních myšlenek agilních metodologií, a to mít spokojeného zákazníka. Není vhodné začít pracovat na co nejdetailnějším řešení problému, ale nechat si otevřené možnosti pro případ změny. Také čím později se začne pracovat na nejdetailnější úrovni, tím je vyšší šance se přiblížit představě zákazníka
- **Dodávat co nejdříve** – Každý zákazník preferuje rychlé dodání požadovaného produktu (nebo jeho části). Rychlé dodávky také redukují riziko změnových požadavků ze strany zákazníka (protože zákazník nemá téměř čas měnit názor). Zlepšuje se reakční doba na nové požadavky a je mnohem jednodušší plánovat následné balíky práce. Také se zmenšuje velikost připomínek k poslednímu nasazenému balíku práce.
- **Delegace pravomocí týmu** – Tato praktika je o důvěře v tým. Delegace pravomocí poskytuje týmu větší volnost a podporuje jejich sebevědomí a schopnost inovace. Krom zefektivnění práce týmu je tato praktika i výtečnou motivací.
- **Budování integrity** – Integrita produktu (rovnováha mezi použitelností, spolehlivostí, funkcionalitou a hospodárností) je pro každého zákazníka stěžejní. Tým se poučuje ze svých chyb během procesu a neustále (kdykoliv je k tomu vhodná příležitost) se snaží produkt vylepšovat.
- **Vidět projekt jako celek** – Platí pravidlo, že čím komplexnější projekt je, tím více se objevuje nebezpečí, že se bude tým snažit nahlížet na jednotlivé části produktu jednotlivě a podle toho je i řídit. Může tak docházet k optimalizaci jednotlivých funkcionalit na úkor celkového produktu. Proto je nutné nahlížet na celý projekt jako na celek.

(POPPENDIECK;2006)

**FDD**

- **Doménové objektové modelování** – Účelem této praktiky je dekompozice celkové domény (nebo problému) na menší objekty. Menší objekty znamenají menší problémy. Velice často se k tomuto účelu používá barevné modelování pomocí UML diagramů. Pokud jsou odhaleny řešení menších problémů (pomocí dekompozice), jejich spojením lze získat celkové řešení.
- **Vývoj podle užitečných vlastností** – Užitečnou vlastností se rozumí funkcionality, která přináší zákazníkovi nějakou přidanou hodnotu a lze ji implementovat během 2 týdnů. Tato praktika používá dekompozici funkcionalit, které jsou příliš komplexní na implementaci ve 2 týdnech na jednotlivé užitečné vlastnosti.
- **Individuální vlastnictví tříd** – Vymezuje vlastnictví tříd v rámci týmu. FDD používá individuální vlastnictví tříd, čili každá třída je přidělena k právě jednomu pracovníkovi.
- **Tým pro užitečné vlastnosti** – Jelikož každá užitečná vlastnost má zpravidla více než jednu třídu, je nutné seskupit veškeré vlastníky požadovaných tříd. Vlastník užitečné vlastnosti (Feature Owner) by měl poté řídit nově vzniklý tým za účelem vytvoření požadované funkcionality.
- **Inspekce** – Tým pro užitečné vlastnosti by měl být kontrolován, aby výsledek jejich práce nabyl požadované kvalitativní úrovně. Inspekce také mohou odhalit místa pro zlepšení a další vývoj kultury a týmové soudržnosti.
- **Pravidelné buildy** – Buildování by mělo být prováděno v pravidelných intervalech. Tím se mimo jiné zajistí i to, že v každé situaci bude připraven systém pro možnou ukázkou.
- **Řízení konfigurací** – Díky této praktice je možné sledovat nejnovější verze zdrojového kódu a historicky lze dohledávat veškeré minulé verze.

(PALMER;2002)

**XP**

- **Zajištění prostoru** – Tým by měl mít dostatečný prostor pro svoji práci. Dále by neměly existovat bariéry pro komunikaci, proto jsou nejlepší „open space“ prostory, kde se každý člen týmu cítí být na stejné úrovni a komunikace probíhá bezproblémově.
- **Točit lidi** – Zaměstnanci by měli být „protočeni“ mezi různými týmy, aby se lépe sdílely znalosti. Pokud se v celé společnosti nachází jediný zaměstnanec, který má určitou znalost (např. zkušenosti s určitou metodikou nástrojem atp.), obzvláště pokud je tato znalost pro danou společnost kritická, je vhodné, aby postupně pracoval

v jiných týmech a tímto způsobem trénoval ostatní zaměstnance (čímž se znalost rozšíří a společnost tak bude mít o použitelný zdroj navíc).

- **Opravení metodiky když se pokazí** – Při přijetí nové metodiky je snaha používat veškeré její procesy, nástroje a pravidla. Pokud nějaké pravidlo nefunguje, nemá smysl na něm lpět, a proto je nezbytné takové pravidlo (nástroj, proces atp.) přestat používat, nebo změnit tak, aby týmu více vyhovovalo.
- **CRC karty** – Praktika sloužící k efektivnímu návrhu systému jako tým. CRC (Class- třída, Responsibilities – odpovědnosti, Collaboration – spolupráce) karty pomáhají týmu odprostit se od klasického přemýšlení a evokují objektivně orientovaný pohled na problematiku. Díky této praktice lze také lépe nalézt alternativy k řešení.
- **Kolektivní vlastnictví** – Každý člen týmu může přicházet s novými nápady. Pokud jde o programátora, může kdykoliv přidat, nebo měnit kusy kódu. V takovém případě ale musí být předem daná akceptační kritéria (ať již akceptační testy v případě IT projektu, nebo kvalitativní požadavky v ostatních oborech).
- **Práce v páru** – V XP je tato praktika pojmenovaná párové programování, vzhledem k zaměření této práce však bude tato praktika brána obecněji. Myšlenkou je, že pokud programují (v tomto případě také analyzují, testují atp.) dva členové týmu na jednom počítači, je vyšší šance odhalení chyby a nalezení nového řešení.
- **Zákazník je vždy dostupný** – Zástupce zákazníka by měl být vždy dostupný, v případě nejasností, nebo nutnosti učinit rozhodnutí (dostupnost by měla být nejlépe v pracovní době dodavatele). Nejlepším řešením je zástupce zákazníka přímo na místě dodavatelovy společnosti.

(WELLS;1999)

### Agilní Modelování

- **Použití více modelů** – Při tvorbě modelů během vývoje softwaru je nutné používat více modelů, protože každý model zobrazuje různé aspekty systému.
- **Udržování co nejnižšího počtu objektů** – Ve vztahu k modelům je žádoucí udržovat co nejnižší počet potřebných modelů. Pokud jsou k úspěšnému vývoji zapotřebí 4 modely, není efektivní dělat jich více.
- **Okamžitá zpětná vazba** – Zpětná vazba je pro úspěšný projekt nezanedbatelná. Proto je důležité snažit se o co nejkratší zpětnou vazbu, čímž se docílí rychlejší reakce na požadavky zákazníka a lze tak efektivněji reflektovat aktuální potřeby zákazníka.

- **Předpokládat jednoduchost** – Zde platí známé pravidlo Occamovy břitvy. Nej-jednodušší řešení bývá i to nejsprávnější. Při přenesení tohoto pravidla do vývoje softwaru to znamená, že se jednotlivé funkcionality vytvářejí co možná nejjednodušeji (což krom přehlednosti a menší systémové zátěže přináší i snadnější úpravy v systému).
- **Přijímat změny** – Tato praktika je obsažena i v samotném agilním manifestu. Žádná funkcionality není nezměnitelná, zákazník i ostatní zainteresované strany v projektu se v čase mění a je v zájmu dodavatele tyto změny přijímat.
- **Inkrementální přístup** – Produkt se nedodává jako celek, ale po použitelných částech, takže zákazník vidí hodnotu za své vynaložené prostředky v průběhu celého projektu a nikoli pouze na jeho konci. Díky této praktice, má zákazník také možnost více reagovat na změny.

(AMBLER;2001-2014)

## DSDM

- **Účast uživatele** – Uživatel by jako jedna ze zainteresovaných stran neměl být přehlížen. Mnohdy je brán jako nejméně důležitý člen projektu (spíše se hledí na požadavky managementu zákazníka), ale vyvíjený systém budou používat právě uživatelé, a tak by měly být vyslyšeny i jejich názory na dodávaný produkt.
- **Zaměření se na kritickou funkcionality** - Iterativní dodávání produktu se musí zaměřit hlavně na kritické funkcionality systému, které vycházejí z aktuálních potřeb zákazníka. Spíše než mít funkcionality či modul co nejvíce, košatý je efektivnější se zaměřit na to, co zákazník doopravdy potřebuje.
- **Odvolatelnost změn** – Každá změna přijatá v průběhu projektu by měla být odvolatelná. Není žádoucí zůstat u zavedených změn, které se ukázaly být kontraproduktivní.
- **Komunikace zainteresovaných stran** - Je nezbytné, aby komunikace neprobíhala jen od zainteresovaných stran směrem k týmu, ale aby tyto strany komunikovaly i mezi sebou.

(SELECT BUSINESS SOLUTIONS; 1988-2016), (COBB;2015)

## Ostatní

Zde jsou uvedeny veškeré Best Practices, které nevznikly v rámci metodologie a jsou zpravidla využívány ve více metodologiích.

- **User Stories** – Jde o funkcionality systému, které lze nasadit do dvou týdnů. Každá User Story by měla být napsána z pohledu uživatele (či jiné role) daného systému.

Měla by obsahovat kdo potřebuje tuto funkcionalitu, o jakou funkcionalitu se jedná a proč jí potřebuje.

- **Planning Poker** – Jedná se o metodu odhadování pracnosti jednotlivých funkcionalit (většinou se používá v metodologiích, které užívají User Stories). Využívá se tzv. planning poker karet, které nabývají různých hodnot, a každý účastník vybere kartu, podle jeho odhadu pracnosti. Hodnota, která se objeví nejvíce je poté použita jako odhad pracnosti do plánu.

(WELLS;1999), (COHN;2006)

## 2.3 Best Practices

### 2.3.1 Definice Best Practices

Best Practices jsou v dnešní době v prostředí IT projektů velmi často zmiňovány. Při vyslovení tohoto slovního spojení se téměř každému pracovníkovi v IT oblasti vybaví něco, s čím má toto sousloví spojeno. Best Practices lze definovat jako ověřenou techniku (nebo praktiku), která byla ověřena praxí, bývá součástí metodiky nebo různých směrnic a lze jí aplikovat samostatně i bez nutnosti aplikace celé metodiky, ze které pochází. Best Practices jsou ve většině případů chápány dvěma způsoby.

Zprv se jedná o něco, co je ve společnosti (nebo projektu) běžně využíváno na denní bázi. Něco co je vnímáno jako naprostá samozřejmost (dalo by se mluvit i o zvyklosti, či zažité rutinně), kterou každý zná, a ví, o co se jedná. Tyto praktiky jsou většinou v podvědomí lidí, kteří s nimi přichází do styku, ale mnohdy je pak velice problematická jejich formulace a jejich popsání v dokumentaci.

Další náhled na Best Practices je, že se jedná o praktiky, které jsou zavedeny v dokumentacích různých metodik. Mohou tak být vnímány, jako osvědčené postupy, které byly dávno ověřeny nepřeberným množstvím společností a akademických institucí. Jsou tudíž brány jako ověřený fakt, který může být bezpečně využit k zefektivnění procesů. Může se však jednat také pouze o interní metodiku, která vznikla shromažďováním právě Best Practices. V tomto kontextu jsou Best Practices vnímány jako poznatky získané prostřednictvím zkušeností z projektů, procesů (zvláště při snaze o jejich zefektivnění), výsledky různých reportů a zpráv a množství rozhodnutí, které měly zásadní vliv na business společnosti. Poté se tyto praktiky stávají součástí interní dokumentace a metodik, jelikož se v daném prostředí nesčetněkrát ověřily a jejich výsledky jsou podloženy údaji ve zprávách a reportech. Z toho lze vyvodit, že rozličné metodiky, které vznikají napříč IT světem, začínají vždy u osvědčených praktik a technik, které jsou poté spolu s ostatními praktikami sjednoceny do jedné dokumentace. Právě díky tomuto druhému pohledu na problematiku Best Practices se společnosti snaží evidovat veškeré zkušenosti, rozhodnutí či techniky, aby mohly být použity v obdobných projektech a stát se tak neustále používanými praktikami. Tyto činnosti by měl mít na starosti tým znalostního managementu. (INFO ENTREPRENEURS;2009), (BUONO;2005)

### 2.3.2 Řízení Best Practices

Jak již bylo výše rozepsáno, dokumentaci možných Best practices, implementaci nových a ostatní činnosti spojené s vedením Best Practices má především na starosti tým znalostního managementu. Pokud jde o řízení Best Practices je nezbytné podrobovat analýze veškeré ukončené projekty (nejenom pokud je projekt jako celek ukončen, ale i na konci každé fáze), aby bylo možné izolovat nově nalezené praktiky (prostřednictvím lessons learned – učení se ze zkušenosti). Je nutné mít na paměti, že každá z bezpočtu Best Practices nemusí být tím nejlepším řešením pro daný projekt. Implementace každé jednotlivé praktiky by měla být analyzována z pohledu již použitých praktik (aby se zabránilo přílišné byrokracii a nepoužívalo se příliš mnoho praktik, což může být ve výsledku kontraproduktivní), ale také z pohledu projektu a jednotlivých členů. Proto by měl probíhat neustálý monitoring projektu, ze kterého lze vydedukovat, zda jsou potřeba nějaké nápravné kroky, aby projekt byl nahrazen zpět do správných kolejí, nebo byl více zefektivněn aktuální průběh projektu. Dále je nutné se zaměřit na Best Practices, které byly nasazeny v předchozí fázi, aby bylo možné zkontrolovat, zda jejich nasazení přináší požadované výsledky, nebo jestli je nutné od těchto praktik upustit z důvodu nízké nebo i záporné efektivity. K tomu lze použít např. KPI indikátory. Pomocí těchto indikátorů lze monitorovat současný stav businessu a plnění stanovených cílů. Z výsledků lze pak odvodit, jaké následující kroky je třeba učinit. Je nezbytné, aby měřené metriky byly pro získání potřebných informací jasně definované (a tudíž i dobře a konkrétně měřitelné). Další z možných nástrojů pro získání potřebných informací je např. benchmarking, plánování a výkonový monitoring. Pomocí těchto nástrojů lze porovnat výkonnost businessu před a po nasazení změn. Poslední část řízení Best Practice, ale neméně důležitá, je komunikace. Zde je myšlena komunikace na všech úrovních. Komunikací na úrovni týmu lze dosáhnout požadované efektivity nasazením Best Practices, vyjasněním a pochopením dané praktiky. Komunikace mezi projekty a projektovou kanceláří pomáhá sdílet využívané praktiky a případné doporučení. Komunikace na úrovni vedení pomáhá vybírat praktiky, které jsou v souladu s vizí a strategií společnosti.

(BUONO;2005), (INFO ENTREPRENEURS;2009)

### 2.3.3 Proč vyhledávat a implementovat Best Practices

Co je hlavním důvodem, který přiměje společnost (ať již se jedná o vedení společnosti, projektovou kancelář či jednotlivé projektové manažery), aby začala s vyhledáváním pro ni nejvhodnějších Best Practices a jejich následnou implementací? Odpověď na tuto otázku je zcela zřejmá. Jde o dosažení cílů společnosti. A tyto cíle jsou pro většinu společností stejné (pokud se nejedná o neziskové organizace, státní instituce atp.). Jedná se především o realizaci co největšího zisku, získání dominantního postavení na trhu či jeho segmentu. Toho je z velké části dosaženo náskokem před konkurencí. Tento věčný závod vyžaduje neustálou inovaci a výzkum v oblasti, v níž se daná společnost vyskytuje. Neustále se vyvíjející technologie a prudké změny trhu nutí společnosti pružně a efektivně reagovat na různé

podněty z okolí společnosti. Toto je také důvod, proč je součástí kultury většiny nejúspěšnějších společností inovace. Co se procesu inovace týče, zjevně nejdůležitějším pojmem je znalost. Inovace je v podstatě aplikace propojených aktivit začínajících u nápadů, až po výzkum. V průběhu tohoto procesu vznikají nové znalosti, které jsou během těchto aktivit hojně využívány. Také dochází k aplikaci starších znalostí. Na základě těchto skutečností lze říci, že inovace je závislá hlavně na znalostech. Ve vztahu znalostí k inovacím jsou velmi důležité 4 základní procesy (vytváření znalostí, sdílení znalostí, akvizice znalostí a dokumentace znalostí). Během těchto procesů se vytváří a sdílí nové znalosti a inovace je pak jen aplikace výstupů těchto procesů.

Při bližším zaměření na úlohu Best Practices v inovaci vyvstává otázka, jaké praktiky jsou pro daný projekt v dané společnosti v daném segmentu trhu nejlepší. V tomto případě je dobré zužitkovat zkušenosti, které vedoucí pracovníci nasbírali, jelikož je nutná jistá dávka predikce. Nikdo nemůže s určitostí říct, jak se bude trh vyvíjet, nebo zda implementace praktik přinese požadované efekty. Vždy existuje možnost na nepředvídatelný výsledek. Zkušenosti a odhad budoucí situace tuto šanci značně snižují. Např. pokud budeme uvažovat společnost, která se rozhodne k implementaci Best Practices, které nejsou dosud tolik využívány a ověřené. Může se jednat o novou metodologii (nebo novou verzi či odvozenou metodologii, která obsahuje nové praktiky), která není dosud hojně využívána, protože není dostatečně ověřená v praxi. V takovém případě společnost přijímá riziko neúspěchu, které může skončit i ztrátou většího podílu na trhu (velikost tohoto rizika je nutné identifikovat a poté se rozhodnout, zda při dané míře rizika a možného zisku danou metodiku přijmout), ale také může být v daném odvětví první společností, která dané praktiky adoptuje a získá tak náskok nad konkurencí. Dále se může jednat o metodologii či praktiku, která je léty ověřená a využívána, ale v jiném odvětví, než ve kterém figuruje modelová společnost. V tomto případě lze jako příklad uvést Lean, který začal ve výrobním odvětví a úspěšně se transformoval i do IT. I zde existuje identické riziko přijetí takovéto metodologie. Jako poslední příklad lze uvést metodologie či praktiky, které jsou hojně využívány v zahraničí, a pak je nutné zanalyzovat, zda lze tyto praktiky aplikovat i v dané společnosti (praktiky v jiných zemích nemusí fungovat v domácím prostředí, a to z mnoha důvodů jako např. legislativa, kulturní prostředí, technologická úroveň atp.).

(AKTHARSHA;2011);(ANDREEVA;2011), (INFO ENTREPRENEURS;2009)

### 2.3.4 Vztah Best Practices a znalostí

Pro získávání konkurenční výhody jsou také velice důležité znalosti a jejich aplikace. Může se praktika stát znalostí? Praktika jako taková nemůže být považována za znalost. Pro určení důvodu, proč se nemůže praktika stát znalostí, je nutné si definovat znalost. Znalost je vlastně zdroj dané společnosti, který je možné opakovaně využívat, inovovat a sdílet. Na rozdíl od ostatních zdrojů nemůže být znalost uložena v počítači, ani v jiném fyzickém úložišti. Jediné možné místo pro uložení znalosti je tedy lidský mozek. Naleznou se i tací,

kteří s tímto tvrzením nesouhlasí, ale je nutné si uvědomit, že pokud se daná znalost popíše do dokumentace, jedná se o informaci nikoli o znalost. Na základě této informace pak vzniká ona znalost. Znalost je tudíž informace, která je spojena se zkušeností (např. situace co člověk zažil), kontextem (např. prostředí a kultura společnosti), interpretací, reflexí, intuicí a kreativitou. Právě díky těmto aspektům se informace stane znalostí a zaměstnanec jí poté může aplikovat na potřebný proces. Z výše uvedených informací vyplývá, že Best Practice jako taková je pouze informace. Je patřičně zdokumentovaná, ale sama o sobě nemá užitek. Až její aplikace v daném kontextu spolu s kreativním nápadem na její realizaci a zkušenostmi (a ostatními aspekty, které dělají z informace znalost) vzniká znalost, která má pro společnost hodnotu a může být aplikována přímo do projektu (nebo procesu). Závěrem lze tedy konstatovat, že Best Practices a znalosti jsou úzce spojeny, jelikož požadovaná hodnota vybraných praktik vznikne jejich převedením na znalosti a jejich aplikací. (AKTHARSHA;2011);(ANDREEVA;2011)

## 3 Integrace Best Practices do PRINCE2 a nejčastějších problémů projektového managementu

V kapitolách níže bude rozepsána integrace jednotlivých definovaných Best Practices do jednotlivých částí PRINCE2. Dále bude definován souhrn častých problémů, které nastávají v průběhu projektů. Podle četnosti použitelnosti budou jednotlivé Best Practices ohodnoceny body.

### 3.1 Definice bodů

Aby bylo možné zjistit, které Best Practices jsou nejvhodnější pro integraci do metodologie PRINCE2, bude každé praktice přiděleno bodové ohodnocení na základě četnosti jejich použitelnosti. Jelikož se tato práce zaměřuje především na metodologii PRINCE2, budou praktiky, které lze uplatnit v jakémkoliv aspektu této metodologie ohodnoceny 5 body za každý takový aspekt. Pokud lze využít praktiku v případě nějakého nejčastěji se vyskytujícího problému, bude ohodnocena 1 bodem za každý takový problém (pokud lze praktiku použít jenom částečně, např. problém jako takový neodstraní, ale lze ji použít jako odstranění důsledku, bude bodový zisk ponížěn o půlku). Na závěr této práce budou jednotlivé Best Practices seřazeny v tabulce právě podle počtu bodů, které získaly v této části práce. Tato hodnota zanesená v tabulce bude vypovídat, jakou mírou lze danou praktiku integrovat do PRINCE2 a jakou mírou se dá využít řešení níže definovaných problémů.

### 3.2 Definice nejčastěji se objevujících problémů

Nyní budou definovány nejčastěji se objevující problémy při realizaci projektů. Tyto problémy byly převzaty z praxe autora a jeho kolegů (ze všech společností, kde za kariéru pracovali). Projekty byly řízeny ve střední české společnosti, která se zabývala především vývojem systémů pro veřejnou správu. Definované problémy pak budou konfrontovány s nejvhodnějšími praktikami, aby bylo možné zjistit, zda je některé agilní Best Practices mohou pomoci zažehnat.

- **#1 Neporozumění zadání** – Velice častý problém vyskytující se všech typech projektů. Problém neporozumění zadání může vzniknout v mnoha chvílích a na obou stranách. Zákazník nemusí problém dobře popsat, má na mysli něco jiného než se na první pohled může zdát, nebo ani on sám neví přesně, co chce. Další neporozumění může nastat na straně dodavatele, z důvodu nedostatečného dotazování zákazníka, nebo přílišné sebejistoty (problém se může zdát triviální nebo se podobá již řešenému problému, ale poté se může ukázat jako mnohem komplexnější). Neporozumění zadání může vést k větší časové i finanční náročnosti požadavku (musí být znovu uspořádána schůzka se zákazníkem, aby se zadání vyjasnilo, nebo se o nalezení správného řešení bude pokoušet tým dodavatele atp.), dodání nechtěné funkcionality, nebo může vyústit i v nefunkční dílčí produkt (a tím se projekt zpozdí minimálně o 1 sprint).
- **#2 Konflikt v týmu** – Projektový tým je tvořen lidmi. Jelikož jsou mezilidské vztahy velmi složité, je vysoká pravděpodobnost, že mezi jednotlivými členy týmu vznikne nedorozumění, nebo přímo konflikt. Konflikt může vzniknout např. zpochybněním přínosu (nebo jen konkrétního User Story, na kterém člen týmu pracoval) daného člena týmu, nebo rozdílným charakterem obou osob figurujících v konfliktu. Toto může mít velice neblahý dopad na celý projekt a to z několika důvodů. Členové týmu budou pracovat neefektivně, můžou se soustředit více na poškození svého „rivala“ než na samotnou práci a dokonce se mohou objevit naschvály či šikana na pracovišti. Takové problémy musí projektový manažer odstranit přímo v zárodku.
- **#3 Zákazník není přístupný kompromisům nebo nepochopil své povinnosti** – Mnohdy se stává, že zákazník se snaží zneužít své silné pozice v rámci projektu. Každý projekt provázají různé problémy či neočekávané situace. Část těchto problémů může být vyřešena jen ve spolupráci se zákazníkem. Jako příklad může posloužit situace, kdy dodavatel potřebuje součinnost třetí strany (např. aplikace třetí strany posílá jiná data, než bylo dohodnuto na poslední projektové schůzce), ale zákazník trvá na svém, že problém je právě na straně dodavatele. V takovýchto případech se někdy stane, že problém vyřeší až odkázání na smlouvu nebo zápis z jednání, kde je jasně deklarováno, kdo má za danou problematiku zodpovědnost (proto je důležité vést zápisy ze všech schůzek se zákazníkem a mít je podepsané).
- **#4 Člen týmu není tolik výkonný (nechápe účel projektu)** – Toto není příliš častý problém, ale občas některý člen týmu není natolik výkonný, na kolik by bylo potřeba. V tomto případě to pramení z nepochopení projektu, dílčího projektu nebo jen nějaké funkcionality. V takovém případě člen týmu pracuje s nechutí a mnoho času stráví také rozporováním svého úkolu. Tento problém by měl být stejně jako konflikt v týmu zaražen projektovým manažerem už v počátku (tyto dva problémy mohou také být úzce propojeny).

- **#5 Realizace příliš vysokého počtu projektů** – Každá společnost chce generovat co nejvyšší zisk. Pokud se jedná o dodavatele softwarových řešení, pak se budou přirozeně snažit získat co nejvíce zakázek. Z každé zakázky může vzniknout několik projektů s různou pracností. V této situaci by mělo vedení mít na paměti jaké zdroje má společnost k dispozici. Mnohdy se však stává, že se vedení chybně domnívá, že kapacity mají výkonnější, než reálně jsou (toto může nastat, když vedení nepochopí nebo nezná, co je přesně za účel projektu, nebo nezná daný okruh požadovaných systémů a až příliš pozdě se dozví, že žádný z jejich zdrojů nemá s touto problematikou žádné, nebo jen velice malé zkušenosti), že se zdroje doplní v průběhu projektu (z externích zdrojů, nebo interních po ukončení jiných projektů), nebo že zákazník bude svolný k průtahům projektů (tuto informaci se zákazník zpravidla dozví dříve při prvním akceptačním milníku). Toto může vést k nespokojenosti zákazníka, důsledkem čehož ztratí společnost možnost realizace dalších projektů (může přijít o stálého zákazníka, takže kvůli honbě za krátkodobým ziskem přichází o dlouhodobější zdroj obživy), nebo k přetěžování svých zdrojů čímž nejspíše utrpí kvalita produktu a o některé zdroje mohou i přijít.
- **#6 Špatná komunikace v rámci týmu** – Tento problém se mnohdy sám stává původcem ostatních problémů (jako např. konflikt v týmu, nepochopení zadání či nepochopení účelu projektu atp.). Platí, že v každé týmové činnosti je komunikace základ. Pokud je komunikace na špatné úrovni, nebo dokonce probíhá velice sporadicky, vzniká velké nebezpečí, že průběh projektu bude velice trpět. Člen týmu špatně pochopí svůj úkol, narazí na problém, který se bude snažit vyřešit sám bez další konzultace, projektový manažer se včas nedozví důležité informace atp. Problémem v tomto případě není jenom nedostatečná komunikace, ale také pravý opak, čili nadbytečná komunikace. Pokud členové týmu stráví více času konzultacemi a schůzováním než samotnou prací, je zřejmé že se projekt protáhne.
- **#7 Špatná komunikace směrem k zákazníkovi** – Zde se jedná v podstatě o kombinaci problému komunikace v rámci týmu a problému se zákazníkem (v tomto případě je to důsledek tohoto problému). Nedostatečná komunikace se zákazníkem může mít za následek zvýšenou pracnost (z důvodu delšího časového intervalu, kdy se dostane informace o nepochopení zadání nebo změnovém požadavku k členům týmu), nespokojenost zákazníka a i vážné důsledky na výstupy projektu.
- **#8 Vágně definovaný rozsah** – Při definování rozsahu projektu a všech jeho dílčích produktů je potřeba dát si pozor nejenom na porozumění zadání, ale i na dostatečnou úroveň detailů. Pokud se při analýze ve spolupráci se zákazníkem nepodchytí veškeré důležité detaily (v tomto případě je myšleno nezjištění důležitých detailů, které mají dopad na fungování výsledného produktu, který vzejde z prvních iterací), je zřejmé, že se budou muset tyto detaily doplnit, což stojí drahocenný čas (ať už se uspořádá další schůzka, nebo se vyžádají detaily jinou formou, projekt bude stát a zdroje nebudou využity). Vágně definovaný rozsah také může

vést k situaci, kdy si chybějící informace domyslí členové týmu, a pak můžou vzniknout nekonzistence v řešení.

- **#9 Příliš často měněný rozsah** – Agilní přístupy přinášejí do projektu krom pozitivních úprav i jistou dávku nejistoty. Právě kladný přístup ke změnovým požadavkům ze strany zákazníka je dobrý příklad. Pokud zákazník často mění rozsah projektu k obrazu svému, má to velký vliv na celkový čas projektu, na pracnost (některé požadavky již mohou být rozpracovány) a tím pádem i na morálku členů týmu. Může se také vytratit původní účel projektu a tím i smysl, proč projekt realizovat.
- **#10 Nízká flexibilita** – Jak společnost, tak i projektový tým by měly zvládnout flexibilně reagovat na vnitřní i vnější faktory, které ovlivňují projekt. Toto platí obzvláště v IT oboru, kde jsou nové technologie a metodologie představovány velice často. Tato flexibilita souvisí také s požadavky na změnu ze strany zákazníka. Pro toho je důležité, aby výsledný produkt odpovídal jeho potřebám. Např. při změně legislativy se při nízké flexibilitě může stát produkt nepoužitelný.
- **#11 Špatné odhady pracnosti** – Při odhadech pracnosti jednotlivých funkcionalit můžou nastat dva základní problémy. Funkcionalita může být velice nadhodnocena, takže na ní bude přiděleno více zdrojů, které poté budou chybět jinde (ať už při práci na jiném požadavku, nebo rovnou i na jiném projektu), nebo může být podhodnocena, důsledkem čehož se zpozdí plánované nasazení a blokuji se zdroje, které podle původního plánu měly být použity jinde.
- **#12 Příliš detailní a starostlivé řízení** - Tento problém se týká především projektového manažera. Projektový manažer by měl umět naslouchat svému týmu, který může být potenciálním zdrojem dobrých nápadů. Měl by více věřit svému týmu a neměl by mít zbytečně tvrdý přístup (neustále tlačení na termíny a výsledky, arogantní přístup, nebo dávat najevo nedůvěru v samostatnost týmu).
- **#13 Přílišné spoléhání na nástroje** – V průběhu projektu vyvstane celá řada problémů, na které se projektový manažer (potažmo projektová kancelář) snaží najít řešení. Mnohdy vidí řešení problému v použití softwarového nástroje. Použití takového nástroje může pomoci např. v projektu, kde jsou hlavním problémem špatně vedené požadavky. V takovém případě může pomoci opravdu správný nástroj na evidenci a vizualizaci. Ale mnohdy bývají původcem problému procesní záležitosti, nebo špatné pochopení (a vedení) metodologií. V tomto případě se problém pouze prohloubí, jelikož nástroj nedokáže na takovýto druh problému reagovat (a mimo jiné se zvednou náklady o nově pořízený software).
- **#14 Nedefinované (či málo definované) metriky** – Při akceptaci produktu (dílčího produktu nebo funkcionality) se v nepřítomnosti pořádných (nebo v nedostatečně podrobnosti definovaných) akceptačních metrik často stává, že je předání odmítnu-

to, nebo se řeší až příliš chyb, které znemožňují v další iteraci řešit potřebný počet požadavků.

- **#15 Nedostatečné kapacity** – Problém, který se objevuje v souvislosti s řešením příliš vysokého počtu projektů. Jak již bylo popsáno, důsledkem nedostatečných kapacit dochází k přetěžování zdrojů, snížení kvality a mnohdy i ke ztrátě zdrojů (pokud odejde zaměstnanec s pro společnost kritickým know-how, může to skončit i krachem společnosti)..
- **#16 Nedostatek kvalifikovaných zaměstnanců** – Jedná se o problém, velmi podobný nedostatku kapacit. Mimo již zmíněné přetěžování vzácných zdrojů (které mají kritické znalosti a zkušenosti), může dojít k snížení zisků společnosti (nedostatek potřebných zdrojů snižuje počet možných projektů). Jelikož jsou znalosti těchto zaměstnanců poměrně vzácné, je nutné se na tento problém zaměřit co nejdříve (ať už opatřením nových zdrojů, nebo rozšířením znalostí i mezi ostatní zaměstnance).
- **#17 Konflikt priorit** – V tomto případě je myšlena situace, kdy je zaměstnanec členem dvou a více týmů, přičemž každý tým je v rámci jiného projektu (nebo je členem jednoho týmu a celý tým řeší najednou více projektů). Velice často pak člen týmu pracuje na poslední přidělené prioritě, jelikož poslední projektový manažer, který mu zadal práci, daný úkol popsal jako nejurgentnější záležitost. Většina projektových manažerů vidí své projekty jako největší prioritu a potom se může lehce stát, že tým pracuje na méně důležitém úkolu. Takovou situaci by měl zpravidla řešit vedoucí oddělení nebo projektová kancelář (záleží na organizační struktuře, kultuře a nastavených pravidlech v rámci společnosti).
- **#18 Nejasně definované priority** – Každý tým při řešení zadaných úkolů řeší prioritně důležité věci a triviální záležitosti až na konec. Pokud však projektový manažer (vlastník produktu nebo kdokoliv, kdo zadává práci) zadává každý úkol jako nejvyšší prioritu, nebo naopak nedefinuje, jak prioritní úkol je, může dojít k situaci, kdy tým neví, na čem má pracovat. Pak vzniká nebezpečí, že si tým určí priority sám, a i když může odhadnout prioritu správně (napojení na databázi má určitě vyšší prioritu než přejmenování tlačítka), není to pravidlem. I když se priorita daného úkolu může zdát nejvyšší, tým může postrádat důležité informace. Může tak vzniknout situace, kde dostane nejvyšší prioritu opravit nefunkční aplikaci před migrací staré databáze na novou. Tým si bude myslet, že migrace může počkat, avšak ve skutečnosti je nefunkční okrajová aplikace (nebo ji využívá malý počet uživatelů, který se bez ní dokáže nějaký čas obejít), a na migraci čekají další dodavatelé.
- **#19 Nespolehlivý člen týmu** – Pokud člen týmu nepracuje jak má (případně nepracuje téměř vůbec), nemusí to být vždy zapříčiněno nepochopením úkolu, projektu atp. Člen týmu může být prostě jenom nespolehlivý a líný. V případě, že není zjednána náprava, může dojít k situaci, kdy daný člen týmu má na starosti kritickou aplikaci a když je nejvíce zapotřebí, tak chybí. Takové zaměstnance je zapotřebí co

nejdříve identifikovat a zjednat nápravu (zaměstnance propustit, nebo nastavit fungující systém).

- **#20 Velké množství připomínek a chyb** – Pokud chodí od zákazníka po každé implementaci nové verze aplikace vysoké množství chyb a připomínek, může to mít neblahé následky na další průběh projektu (chyby se budou řešit na úkor nových požadavků). V takovém případě musí být nastaveny procesy, které se postarají o co nejvyšší možné snížení chyb (téměř vždy jde o nedostatečnou kvalitu).

### 3.3 Best Practices a konkrétní problémy

#### 3.3.1 Neporozumění zadání

Pro vyřešení tohoto problému lze jako první praktiku použít **backlog**. V tomto případě se spíše než o vyřešení problému jedná o jeho prevenci. Podmínkou pro použití backlogu je adoptování agilních přístupů (nejlépe metodologii Scrum). To by totiž znamenalo, že zástupce zákazníka je členem týmu a tudíž byl přítomen vytváření backlogu. Při jeho tvorbě se využívají **User Stories** (backlog by bylo možné použít i bez nich, ale tato další praktika samotný úkol definice podporuje), takže musí být každá plánovaná funkcionality řádně vysvětlena. Při tvorbě backlogu tak téměř jistě dojde k přesné definici zadání (dalo by se zvážit, zda nezařadit i praktiku **plánování sprintu**, jelikož před každým sprintem bude znovu projednáváno zadání).

Další možnou praktikou by bylo **zajištění porozumění procesu** a to v případě, že jde o změnu procesu u zákazníka. Jelikož účelem této praktiky je zajištění porozumění procesu (proč se změna zavádí), budou o důvodu nutné změny informováni jak řešitelé na straně dodavatele, tak i zástupci zákazníka. Případné nejasnosti se tak téměř jistě podaří odstranit během realizace této praktiky.

Z praktik vybraných z metodologie Lean přicházejí v úvahu **rozvíjení znalostí, rozhodování se co nejpozději a vidět projekt jako celek**. Rozvíjení znalostí je založeno na velmi intenzivní zpětné vazbě od zákazníka. Během užívání této praktiky zjistíme, zda je používán správný systém, ale i zda je realizovaný produkt takový, jaký má být. Rozhodováním se co nejpozději docílíme toho, že zákazník si spíše uvědomí, co chce, dostane lepší náhled na celou problematiku a produkt je do detailu vytvořen až později. Díky tomu je vyšší šance na odstranění neporozumění mezi zákazníkem a dodavatelem. To že nebylo porozumění zadání lze snadno odhalit použitím praktiky vidět projekt jako celek. Pávě tehdy lze spatřit všechny souvislosti a tudíž je vysoce pravděpodobné, že vypluje na povrch, že zadání nebylo úplně správně pochopeno a že v celkovém pohledu nedává smysl.

Jako odstranění důsledku problému (nikoli jeho eliminace nebo prevence) lze použít **řízení konfigurací**. Škoda byla sice již napáchána, ale lze alespoň odstranit důsledky problémů vrácením se k předešlé verzi (ale ztráty při realizaci špatného zadání již nelze nahradit).

Pro odhalení špatného zadání lze použít i **CRC karty** a **dostupnost zákazníka**. K odhalení při využití CRC karet dojde již v průběhu plánování a modelování. Je sice možné, že nějaká práce před objevením problému bude ztracena, avšak produkt ještě nebude ve fázi realizace, takže škody budou nižší (záleží samozřejmě na rozsahu špatného zadání, zda se jednalo o celý modul nebo jen o malou funkcionalitu). Pokud je zákazník vždy dostupný (a nejlépe i přítomen na místě), může se případná nejasnost velmi rychle odhalit (ať již konzultací člena týmu se zákazníkem, nebo když si zákazník při plánování uvědomí, že nebyl zcela pochopen).

Z části by se dala použít i praktika **použití více modelů**, ale není zcela jisté, zda by použití více modelů znamenalo odhalení špatného zadání (může vyplynout z kontextu, ale také nemusí).

Pokud se využije **okamžitá zpětná vazba**, tak se špatné zadání odhalí velice rychle (např. při předložení prvního modelu atp.), jelikož jsou zákazník a dodavatel prakticky neustále v kontaktu.

**Komunikace zainteresovaných stran** odhalení nesprávného zadání velice pomůže, jelikož jedna z těchto stran je i sponzor projektu, pro kterého je projekt realizován (ať již jde o externího nebo interního zákazníka). Pokud se budou veškeré zainteresované strany pravidelně scházet nad daným projektem, při revizích zadání, nebo bodů z předešlé schůze, problém se dozajista objeví.

### 3.3.2 Konflikt v týmu

Jako první impuls pro odhalení konfliktu týmu může posloužit **denní scrum**. Pokud vzniká konflikt na základě nespokuprce nějakého člena týmu, nebo nedocnění práce a podobných problémů, může být odhalen právě na každodenní schůzce.

**Retrospektiva sprintu** také může velmi pomoci při odhalování konfliktů v rámci týmu. Po skončení sprintu je mimo jiné rozebíráno, co se povedlo a kde naopak vyvstaly problémy. Právě zde je velká pravděpodobnost, že se ozve některý člen týmu se svým problémem. Pokud danému členovi něco vadilo na přístupu či metodách jiného člena, téměř dozajista se ozve při retrospektivě.

Pravidelné **inspekce** pomáhají odhalovat nejen kvalitativní nedostatky, nýbrž i problémy mezi členy týmu. Při častých inspekcích velmi roste pravděpodobnost odhalení konfliktu (osoba provádějící inspekci může zaznamenat velké napětí na pracovišti).

Jako další z praktik, která pomáhá odhalit konflikt, může posloužit **zajištění prostoru**. Pokud tým pracuje v otevřeném prostoru, tak si případného konfliktu dvou členů všimne další člen, který může daný problém nahlásit projektovému manažerovi. Krom včasného odhalení může tato praktika posloužit i prevenci a následnému odstranění konfliktu. Konflikt mohl mít původ i ve stresu a nedostatečných pracovních podmínkách. Všichni členové týmu si v otevřeném prostoru připadají na stejné úrovni, takže se snižuje i počet možných důvodů konfliktů.

Pokud se budou **lidé točit** mezi různými týmy a pracovišti (je samozřejmě nezbytné nastavit přijatelné intervaly, protože obměňováním týmu každý měsíc by nebylo možné realizovat žádný projekt), rapidně se sníží počet konfliktů. Zaměstnanci budou pokaždé členové nového týmu (což ve větších společnostech může znamenat i poznávat pořád nové lidi), což eliminuje napětí mezi bývalými spolupracovníky. Také měnící se náplň práce (každý projekt má jiné problémy) má pozitivní účinek na morálku.

### 3.3.3 Zákazník není přístupný kompromisům nebo nepochopil své povinnosti

V případě, že zákazník odmítá akceptovat dílo a argumentuje nedostatečnou kvalitou (nebo že podle jeho názoru není produkt dokončen a požaduje přidat další funkcionality), lze uplatnit praktiku **definice toho, co je hotovo**. V nejlepším případě zákazník podepsal tuto definici pro každý dílčí produkt (nebo akceptaci). Poté se lze odkázat, že dodaný produkt opravdu splňuje, co bylo deklarováno v definici toho, co je hotovo, a zákazník je pak nucen akceptovat dodané dílo.

Pokud se aplikuje praktika **zajištění porozumění procesu**, bylo s vysokou pravděpodobností veškeré zadání ujasněno a také definovány podmínky, za jakých bude dílo akceptováno. Tato praktiku lze spojit i s výše uvedenou definicí toho co je hotovo.

V jistém pohledu může být užitečná i praktika **vidět projekt jako celek**. Díky tomu si dodavatel uvědomí, co je jeho odpovědností a co ne (v případě, že zákazník nechce připustit, že daný problém se netýká dodavatele). Pořád bude nutné dokázat, proč není daný problém v rukách dodavatele, ale díky této praxi si to může uvědomit i projektový tým dodavatele (stává se, že ani samotný tým na straně dodavatele si není jistý, zda problematickou oblast spravuje nebo zda problém se týká i jeho oblasti).

Lze použít i **řízení konfigurací**. To v případě, že zákazník obviňuje dodavatele, že jedna z posledních verzí jím dodávané aplikace způsobuje problémy v jiné aplikaci. Pak lze použít řízení konfigurací pro zjištění, zda v posledních verzích mohla nějaká nová či upravená funkcionality daný problém způsobit (u každé verze by mělo být evidováno, co její bylo součástí).

Pokud zákazník nezná meze působnosti dodavatele, je vhodné využívat praktiku **dostupnosti zákazníka**. Pokud je totiž zákazník neustále dostupný na pracovišti dodavatele, je součástí řešeného projektu a spíše porozumí, co má na starost dodavatel a co má na starosti

zákazník (případně další dodavatel). Podobné účinky může mít i **okamžitá zpětná vazba**. Zde je však šance na pochopení mezí, či odstranění problému s nespoluprací velmi nízká. Pomocí zpětné vazby spíše dojde k odhalení problému. V tomto ohledu může být velice podobná praktika **účast uživatele**. Pokud je uživatel přítomen na projektových schůzkách a je brán jako součást projektu, může si lépe uvědomit jaké je pole působnosti dodavatele (jelikož jsou to často uživatelé, kteří poté reportují chyby, a mnohdy mají problém rozlišit, komu chybu nahlásit).

Poslední využitelnou praktikou u tohoto problému je **komunikace zainteresovaných stran**. Obdobně jako u dostupnosti zákazníka, i tady může správně nastavenou komunikací dojít k pochopení, co je odpovědnost které strany.

### 3.3.4 Člen týmu není tolik výkonný (nechápe účel projektu)

Při **plánování sprintu** lze odhalit, zda některý ze členů týmů nechápe účel některých funkcionalit nebo i celého projektu. Při plánování se probírá, které **User Stories** je možné zařadit do dalšího sprintu a bude nejspíše vysvětleno i proč (a přejde se tak k vysvětlení jejich účelu). Pokud se nepochopení účelu odehrává jen v rovině funkcionalit, je možné využít i výše zmíněné User Stories. Z jejich formy je již patrné, proč daná role tuto funkcionalitu potřebuje.

Pokud je nepochopení účelu projektu po skončení sprintu, je možné, že člen týmu tento účel pochopí při **vyhodnocení sprintu**. Zde probíhá i ukázka dosud vytvořeného produktu (či posledního přírůstku). Poté co je vidět výsledek v praxi, zvyšuje se šance pochopit to, co se vlastně v posledním sprintu udělalo.

Pokud nízká výkonnost nepramení z nepochopení účelu projektu, ale z nepochopení role daného člena týmu (nebo i zainteresované strany), je nejvhodnější použít **vizualizaci workflow**. Právě ve vizuálně zpracovaném obraze workflow si může člen týmu uvědomit, jaká je vlastně jeho role v celém procesu.

**Zajištění porozumění procesu** je taky velice užitečná praktika v kontextu tohoto problému. Pokud člen týmu rozumí celému procesu realizace a plánu projektu, pochopí také proč je nutné projekt realizovat.

**Rozvíjení znalostí** přebírá zpětnou vazbu od zákazníka. Právě z těchto informací je možné reálně prezentovat dopady projektu a tím i vyjasnit jeho účel.

Nepochopení projektu může také pramenit z faktu, že člen týmu pracuje jen na jeho individuální části. Pokud dokáže **vidět projekt jako celek**, dokáže i pochopit účel oné individuální části, na které pracuje.

Použití **doménového objektového modelování** může velice ulehčit chápání projektu. Jelikož je každý objekt rozdělen na menší uchopitelné části, může člen týmu podle nich po-

chopit, jak vlastně funguje celek (nebo také naopak může chápat celek projektu, ale nechápe jednu či více funkcionalit, a tak se díky rozpadu na menší díly snáz pochopí, jak menší části fungují a interagují mezi sebou).

Pokud se bude praktikovat **vývoj podle užitečných vlastností**, člen týmu spíše pochopí menší části projektu (za předpokladu, že nepochopil jenom účel modulu či funkcionalit). Pokud se bude vyvíjet podle funkčních celků, bude jasnější, k čemu celý celek slouží, než kdyby se vyvíjely separátně metody atp.

**Inspekce** sice nenapomáhá porozumět účelu projektu, ale může být nápomocná při odhalování tohoto problému.

**CRC karty** jako součást plánování může pomoci osvětlit účel projektu, jelikož člen týmu může v propojení tříd vidět souvislosti, které mu před tím unikaly. Obdobný efekt bude mít i **použití více modelů**.

### 3.3.5 Realizace příliš vysokého počtu projektů

Pokud se použije praktika **retrospektiva sprintu**, je možné získat informace přímo od členů týmů, jak si s prací poradili. Pokud jsou příliš vytíženi, tato informace určitě bude vyřčena během schůzky. Taková informace pak slouží jako podklad pro rozhodnutí, zda množství již realizovaných projektů není příliš vysoké a zda je moudré přijímat další zakázky.

**Limitovat rozdělanou práci** může pomoci lepšímu odhadu kapacit. S takovouto informací již lze přednést managementu, že není možné realizovat takové množství projektů se současnými kapacitami.

**Řízení workflow** pomáhá získávat aktuální informace o stavu projektů. Z těchto informací lze vypořovat vytíženost jednotlivých článků a tak dojít k závěru, nakolik jsou kapacity vytížené a varovat tak, před akceptováním realizace nových projektů.

Při pravidelných schůzkách, kde by **komunikovaly všechny zainteresované strany**, se může probrat problematika kapacit, která vede k nemožnosti realizace vysokého počtu projektů. Jelikož zde jsou přítomny všechny důležité role projektu a dokonce i management, je pravděpodobné, že zde přednesenému problému bude věnována dostatečná pozornost.

### 3.3.6 Špatná komunikace v rámci týmu

Ke zlepšení celkové komunikace v týmu může napomocť **denní scrum**. Členové týmu pochopí, co který z nich dělá, jaké má odpovědnosti a povinnosti. Díky tomu ví, na koho se obrátit v případě problému či nutnosti konzultace.

Po skončení sprintu mohou ventilovat vzniklé problémy či nejasnosti během **retrospektivy** a **vyhodnocení sprintu**. Při obou schůzkách jsou rozebírány všemožné problémy, včetně těch komunikačních.

Pokud je komunikační problém zapříčiněn nepochopením rolí a odpovědností v týmu (tudíž členové nevědí s kým konzultovat různé moduly aplikace atp.), je dobré předvést si, jak funguje celý proces po rolích. K tomu slouží **vizualizace workflow**.

Pokud se aplikuje praktika **společné inovace**, zlepšuje se morálka týmu i jeho komunikace. Každý se cítí součástí týmu a jeho názor má váhu (a jeho nápady mohou být využity).

Pokud se používá **tým pro užité vlastnosti**, jsou spojení členové týmu pod jednou funkcionalitou. Při řešení stejné funkcionality mohou spíše najít společnou řeč a zlepšit tak komunikaci v rámci týmu. Také jsou přehlednější role a odpovědnosti v týmu, což napomáhá k efektivnější komunikaci.

**Zajištění prostoru** velice zefektivní komunikaci v rámci týmu. V otevřeném prostoru mají členové týmu větší pocit sounáležitosti. Takový prostor evokuje pocit, že všichni jsou na stejné úrovni. Jsou odstraněny bariéry a komunikace se stává přirozenější a bezproblémovou.

Také může pomoci praktika **točení lidí** v rámci ostatních týmů. Pokud je tým složený spíše z nekomunikativních typů, vymění se jeden z jejich členů za komunikativnějšího a ten může zlepšit situaci v týmu (bude se snažit neustále rozvést rozhovor a ostatní členy doslova „rozmluví“).

Pokud je realizováno **kolektivní vlastnictví**, je tým spíše nucen komunikovat. Jelikož kód je vlastněn všemi a každý může udělat jakoukoliv změnu, zvýší se i počet nutných konzultací a porad, což povede k efektivnější komunikaci.

Dále může velice napomoci **práce v páru**. Kromě kvalitativních přínosů budou nuceni oba členové týmu v páru mezi sebou více komunikovat, což podpoří jejich komunikační vlastnosti.

### 3.3.7 Špatná komunikace směrem k zákazníkovi

Při komunikaci se zákazníkem může pomoci, pokud se budou dílčí části výsledného produktu **dodávat co nejdříve**. Jelikož bude mít zákazník ve velice krátkých pravidelných intervalech produkt, ke kterému se bude muset vyjádřit, dojde jako vedlejší účinek k častější komunikaci mezi ním a dodavatelem.

Pokud je **zákazník vždy dostupný** na místě dodavatele, bude to znamenat i vysokou frekvenci komunikace mezi ním a dodavatelem. Zákazník bude v podstatě součástí týmu, a tudíž bude komunikace mnohem jednodušší a rychlejší.

Při **pravidelné komunikaci všech zainteresovaných stran** bude zákazník vždy informován o průběhu projektu a zlepší se celková komunikace. Jelikož jednou ze zainteresovaných stran je i **uživatel** (který je také ve většině případů zástupce zákazníka), jeho **účast** podpoří komunikaci.

### 3.3.8 Vágně definovaný rozsah

K řešení tohoto problému lze využít praktiku **rozhodovat se co nejpozději**. Tímto dostane zákazník čas, aby obsah dostatečně dodefinoval a mohlo se začít naplno pracovat.

K odhalení vágně definovaného rozsahu pomůže, pokud projektový tým **uvidí projekt jako celek**. Z celkového pohledu mohou být zřejmé nedostatky v definovaných požadavcích, odpovědností a obsahu.

Na nedostatečnou detailnost rozsahu se může přijít i při modelování celkového projektu. Ať se již použije **doménové objektové modelování**, nebo **použití více modelů**, vždy by mělo být alespoň z jednoho modelu jasné, že není kompletní.

**Zaměření se na kritickou funkcionalitu** může pomoci odhalit nedostatečně definovaný rozsah. Při bližší analýze těchto funkcionalit se rychle projeví nedostatky rozsahu (ať už jde o samotnou funkčnost, nebo v případě více dodavatelů, co je součástí projektu a co není).

Při pravidelných schůzkách a **komunikaci zainteresovaných stran**, lze tento problém lehce vyřešit dodefinováním rozsahu (nebo vymezením co je předmětem dodávky a co již není).

### 3.3.9 Příliš často měněný rozsah

V této situaci je dobré využití **sprintů**, při kterých se realizuje pouze to, co je uvedeno v **backlogu** pro danou iteraci. Pokud se sprinty nastaví na měsíční trvání, může se redukovat změna rozsahu pouze na **plánování sprintu**.

Pokud se bude projektový tým **rozhodovat co nejpozději**, bude mít zákazník dost času si rozmyslet, co vlastně chce a tudíž by měl být rozsah již ustálený a neměl by se tolik měnit.

Pokud zákazník přichází se změnami rozsahu téměř každý týden, bylo by dobré zvážit **opravení metodiky**. Pokud nějaká část metodologie nefunguje dle představ týmu, neměla by se doslovně dodržovat, ale procesy se upraví podle situace. V této situaci by se nepřijímala každá změna, avšak zákazník by byl limitován počtem možných změnových požadavků a úprav rozsahu.

**Komunikace zainteresovaných stran** nabízí východisko i z této situace. Pokud zákazník neustále mění rozsah projektu, je nutné uspořádat schůzku, kde se rozsah pevně stanoví a potvrdí (samozřejmě změny budou možné, ale jen v nepatrném měřítku).

### 3.3.10 Nízká flexibilita

Pokud je problém s flexibilitou způsoben novými požadavky (způsobené změnou vnějších faktorů, jako např. změna legislativy), lze na ně efektivně reagovat pomocí **sprintů a inkrementálního vývoje**. Jelikož nedostane zákazník svůj produkt až na konci projektu, nýbrž dostává pravidelné dílčí dodávky, může lépe komunikovat změny a společnost na ně lépe reaguje.

Pokud se společnost rozhodne **limitovat rozdělanou práci**, může to pomoci flexibilněji reagovat na vnější i vnitřní změny. Pokud se nepracuje na více funkcionalitách najednou, nevzniknou ani takové škody (mnoho požadavků nebylo dosud realizováno, dokud nebudou hotovy již rozdělané), pokud se musí větší část projektu změnit (samozřejmě také záleží na fázi projektu, ve které daná změna nastane).

Při **rozhodování se co nejpozději** je velice redukováno riziko dopadu změn. Čím déle se čeká, než se začnou vyvíjet detaily funkcionalit, tím spíše dokáže tým reagovat na změnu v zadání.

Pokud se rozhodne projektový tým **dodávat co nejdříve**, rapidně snižuje nebezpečí, že daný produkt již zákazník nevyužije (čím déle trvá dodání produktu, tím je vyšší šance, že dodaný produkt již nebude pro zákazníka relevantní). Pokud se k této praxi přidají ještě **pravidelné buildy**, má tým možnost vytvářet ukázkové verze, s jejichž pomocí se může nechat od zákazníka potvrdit jejich aktuálnost.

Pokud zákazník vyžaduje změnu na starší verzi, protože si pozdě uvědomil, že danou úpravu funkcionality nevyužije, je vhodné využít **řízení konfigurací** společně s ochotou **přijímat změny a odvolávat již přijaté změny**.

Pro nejrychlejší reakci na změnu je potřeba, aby byl **zákazník vždy dostupný**. Způsob, jak dá zákazník vědět o potřebné změně, by měla být **okamžitá zpětná vazba**.

### 3.3.11 Špatné odhady pracnosti

Pro zpětný odhad pracnosti požadavku a následné poučení se do další iterace pomůže **retrospektiva sprintu**. Při rozebírání úspěšnosti právě ukončeného sprintu je nutné vzít si ponaučení, pokud se odhad nějaké z funkcionalit ukázal jako špatný. Pro sledování odhalených funkcionalit může posloužit **burndown chart**. Pomocí tohoto nástroje lze odhadovat pracnosti obdobných funkcionalit a zabránit tak nepřesnému odhadu.

Pokud se projektový tým snaží vyhnout špatnému odhadu pracnosti, je důležité vědět, na čem přesně se bude pracovat a nedělat nic navíc. Pokud se **eliminuje odpad**, je mnohem snadnější provést odhad (pokud by se funkcionality rozšiřovaly o nepotřebné metody, jejich pracnost by se zvýšila oproti původním odhadům). Toto je v souladu s praktikou **předpokládat jednoduchost**.

Pokud se jednotlivé balíky práce rozpadnou na menší části až na úroveň jednotlivých funkcionalit, značně to ulehčí jejich odhad. K tomu slouží **doménové objektové modelování**. Pro lepší vizuální zobrazení a pochopení vazeb již základních funkcionalit poslouží **CRC karty** a **použití více modelů**.

Nejpřesnější odhady může projektový tým získat využitím jedné z nejčastějších plánovacích technik v agilním světě a to **Planning Pokeru**.

### 3.3.12 Příliš detailní a starostlivé řízení

Přílišnou starost lze odstranit praktikováním **denního scrumu**. Tím získá projektový manažer představu, co bude který člen týmu dělat následujících 24 hodin.

Pokud projektový manažer **deleguje větší část povinností** na projektový tým a jeho členy, odpadne mu spousta starostí, a může se více soustředit na řízení. Dává tím také najevo důvěru jednotlivým členům týmů, což má mimo jiné i pozitivní dopad na morálku.

Aby projektový manažer dokázal, že má důvěru ve svůj tým a cení si jejich názoru, měl by používat praktiku **společné inovace**. Každý člen týmu může být potenciální studnice nápadů, a proto by se neměli podceňovat.

### 3.3.13 Přílišné spoléhání na nástroje

Pokud se nechce projektový manažer spoléhat na nástroje, které mají pomoci zlepšit proces, je výhodnější zaměřit se spíše na daný proces a zvážit možnosti jeho změny. K tomu lze využít praktiku **řízení workflow**. Aby bylo možné proces řádně sledovat a poté navrhnout změnu, je nutné, aby každému procesu rozuměl každý člen týmu, který se bude na návrhu změny podílet. Proto je důležité **zajistit porozumění procesu**.

Jelikož se změna nemusí jevit jako jasná, měl by projektový manažer zapojit i ostatní členy projektového týmu, jelikož každý člen může přijít s neotřelým nápadem, na který by jiní nemuseli přijít. Proto se na tento problém vyplatí použít **společnou inovaci**.

Problémem bývá v této situaci snaha nahradit nefungující proces nástrojem. Proto by měla být prioritou snaha o změnu procesu. Každá agilní společnost a každý agilní projektový tým by měly **přijímat změny**. Pokud se ukáže jako neefektivní, nebo dokonce kontraproduktivní, měla by se taková **změna odvolat**.

Pokud je chyba přímo v metodologii (špatně řízená, nebo se některé její části nehodí do dané společnosti), mělo by se realizovat **opravení metodiky**.

### 3.3.14 Nedefinované (či málo definované) metriky

Pro definování správných akceptačních kritérií (či metrik) by mělo být využito praktiky **definice toho co je hotovo**. Pokud je tato definice schválená zákazníkem, je velmi nepravděpodobné, že by zákazník odmítl akceptovat dílčí dodávku, pokud všechny funkcionality, které v ní byly obsažené, splňovaly všechny požadované atributy. Také je možné převzít tyto metriky z jiných projektů pomocí sdílení a **rozvíjení znalostí**.

Aby bylo možné definovat správné metriky, je nutné pochopit, co bude výsledkem. Jak bude vyvíjená funkcionalita plnit daný proces? Na tyto otázky pomáhá odpovědět praktika **zajištění porozumění procesu**.

Velice důležitá je také **okamžitá zpětná vazba**. Pokud se během projektu zjistí, že definované metriky již nejsou vyhovující, je potřeba nastavit proces jejich **změny**.

### 3.3.15 Nedostatečné kapacity

Při **retrospektivě sprintu** lze získat informace, které zdroje jsou přetížené. S touto informací je pak možné argumentovat u managementu o navýšení kapacit. Podobně lze tyto informace získat při **řízení workflow**, kde je možné monitorovat, v jaké části se požadavky nejdéle zdrží. Další možností zjištění přetíženosti kapacit je při pravidelných **inspekcích**.

Pokud dojde k přetěžování kapacit z důvodu jejich nedostatku, je nezbytně nutné o této skutečnosti informovat projektového manažera **okamžitou zpětnou vazbou**.

Management společnosti by se samozřejmě měl dozvědět informaci o stavu kapacit. Nejlépe se to dozví v rámci **komunikace všech zainteresovaných stran**.

Poslední možností jak odhadovat nedostatečné kapacity už při plánování požadavků je pomocí praktiky **Planning Poker**. Po odhadu pracnosti lze již jednoduše odvodit, zda má společnost dostatek kapacit na realizaci projektu (nebo iterace) v řádném termínu.

### 3.3.16 Nedostatek kvalifikovaných zaměstnanců

Nejlepším způsobem jak zvýšit počet kvalifikovaných zaměstnanců je sdílení interních znalostí (jelikož jsou to právě znalosti, které si na lidských zdrojích ceníme). Toho lze dosáhnout **rozvíjením znalostí** vně společnosti. Společné inovace mohou také (i když ne vždy) pomoci k distribuci znalostí.

Dále je vhodné **točit lidi** mezi týmy a v rámci společnosti. Tímto se budou sdílet znalosti s dalšími zaměstnanci a tím lze získat více kvalifikované síly (čímž se otevírá možnost přijímat více zakázek vyžadující tyto speciálně kvalifikované zdroje). Další možností sdílení znalostí je **kolektivní vlastnictví kódu**. Pokud bude nějaký zaměstnanec psát kód v souladu se znalostí, kterou ostatní zaměstnanci neznají (je to však potřeba pro úspěšné dodání produktu), bude muset ostatní relevantní zaměstnance do této problematiky zasvětit. Takto se může sdílet znalost i při **práci v páru**.

### 3.3.17 Konflikt priorit

Pro zajištění pořádku v prioritách se jeví jako nejvýhodnější způsob použití **backlogu** (ve spolupráci s použitím **plánování sprintu**). Zde jsou veškeré požadavky seřazeny právě podle priorit. Priority lze také aktualizovat na **denním scrumu**.

Pro lepší orientaci v prioritách lze použít **týmy pro užité vlastnosti**. Každý tým se bude věnovat jenom své přidělené užité vlastnosti, proto existuje jen velmi malá šance konfliktu priorit (také díky lepší znalosti dané užité vlastnosti mohou lépe odhadovat, která funkcionalita má vyšší prioritu).

### 3.3.18 Nejasně definované priority

Řešení tohoto problému je identické jako u konfliktu priorit, čili jde o využití praktik **backlogu**, **denního scrumu**, **týmů pro užité vlastnosti** a **plánování sprintu**.

### 3.3.19 Nespolehlivý člen týmu

Nespolehlivost člena týmu lze odhalit pomocí **denního scrumu**. Pokud se třetí den po sobě pořád potýká se stejným problémem, ukazuje to na jeho nízkou aktivitu.

Aby se nespolehlivý člen týmu nemohl schovávat, je dobré využít **zajištění prostoru**, kde bude na očích ostatním zaměstnancům, což ho může spíše motivovat k práci. Pomůže také **točit lidi**, jelikož se v novém kolektivu nebude chtít ukázat ve špatném světle. Podobný efekt může mít také **práce v páru**.

### 3.3.20 Velké množství připomínek a chyb

Pro snížení celkového počtu chyb je důležité nastavit patřičné metriky, které musí produkt (nebo jeho dílčí část) splňovat, než bude moci být implementována. K tomu lze využít praktiku **definice toho co je hotovo**.

Dobré je také **limitovat rozdělanou práci**. Velké množství rozdělaných požadavků může mít neblahý dopad na jejich kvalitu. Počet chybných požadavků se také sníží, pokud se bude vyvíjet jen to nejnnutnější, to znamená **soustředit se na kritické požadavky a eliminovat odpad**. Pokud se **dodává co nejdříve**, snižuje se objem dodaných požadavků a tím pádem i počet připomínek a chyb (není jich velké množství najednou, které je hůře uchopitelné). Dále je nutné se pro zvýšení celkové kvality vyvíjených požadavků zaměřit na **budování integrity**.

Pro zvýšení celkové kvality je nezbytné dobře porozumět zadání, čili používat **doménové objektové modelování, více modelů**, ale zároveň **limitovat počet objektů**, které se budou modelovat (příliš vysoký počet modelů může vést k matení členů týmu). Také by se mělo používat **individuální vlastnictví tříd**, čímž je každý člen týmu odpovědný za přidělené třídy (pokud bude kvalita nedostatečná, je tak snadnější najít a eliminovat slabý článek, např. proškolením daného člena týmu).

## 3.4 Principy PRINCE2

Níže jsou rozepsány veškeré praktiky, které lze integrovat do principů PRINCE2. U každého principu je v závorce uvedeno číslo, které slouží jako reference v tabulce Best Practices v závěru této práce.

### 3.4.1 Neustálé zdůvodňování opodstatněnosti (1)

Projekt by měl mít smysl po celou dobu realizace. Pokud již nelze zdůvodnit opodstatněnost projektu, nemá smysl v něm dále pokračovat. Pro tento princip je jako první vhodné integrovat praktiku **okamžité zpětné vazby**. Právě skrze zpětnou vazbu lze pravidelně aktualizovat obchodní případ a zajišťovat tak jeho stálou opodstatněnost.

Aby bylo možné zjistit, zda je projekt stále opodstatněný, je důležité udržovat pravidelný kontakt se všemi zainteresovanými stranami. Právě nejčastěji od nich i může přijít informace, že projekt již nemá opodstatnění a je tedy nutné ho ukončit. Proto je důležité udržovat stabilní, častou a čistou **kommunikaci se všemi zainteresovanými stranami**. Jednou ze zainteresovaných stran je i zákazník (ať se již jedná o interního či externího). Proto je vhodné mít **zákazníka neustále dostupného** jako člena projektového týmu. Jako poslední zainteresovaná strana je uživatel. **Účast uživatele** na projektu by měla být také zajištěna, neboť on bude tím, kdo bude výsledný produkt používat. A právě od uživatele je také možné dostat informaci, zda má projekt pořád opodstatnění.

### 3.4.2 Učení se ze zkušenosti (2)

Jako první krok při spuštění projektu, nebo jeho další fáze je potřeba identifikovat získané zkušenosti z minulého projektu či fáze. Z těchto zkušeností by se mělo vycházet při plánování další fáze (ať již jde o zahajovací fázi nového projektu, nebo jen o další iteraci v rámci stejného projektu). Tyto zkušenosti lze využít právě při **plánování sprintu** a za pomoci nástroje **Planning Poker** (poznatky z minulého sprintu lze využít při odhadování pracnosti, zejména podobné požadavky, které se např. ukázaly pracnější, než bylo odhadováno).

Získané znalosti by se měli sdílet a nově vytvořené znalosti by se měly zaznamenávat. **Rozvíjením znalostí** lze dosahovat změn, což je také hlavní účel učení se ze zkušeností. Sdílet znalosti lze několika způsoby. Např. pro to lze využít **společnou inovaci**, kde se dají aplikovat znalosti získané z různých projektů. Jelikož se při této aktivitě zapojuje celý tým, existuje vyšší šance na aplikaci více znalostí z obdobných projektů. Dalším možným způsobem sdílení znalostí je **točit lidi** z různých projektů, aby se jejich znalosti získané z obdobných projektů mohly zúročit i v jiných projektech. Jako členové nových týmů předají své zkušenosti ostatním členům a zefektivní tak průběh projektu (ať už implementací nové praktiky a metodologie nebo snížením dopadů rizik). Při promíchávání týmu je dobré mít na paměti, jaké znalosti převzaté z ostatních projektů mají být sdíleny a komu jsou určeny. Např. pokud má být předanou znalostí programovací praktika, lze využít **práci v páru**, a krom obvyklých výhod této praktiky dojde i ke sdílení oné znalosti.

Na konci každého projektu nebo iterace je nezbytně nutné předat získané znalosti dále (ať už jinému týmu na jiném projektu, nebo znalost zužitkovat hned v další iteraci). Pro tuto aktivitu jsou nejvhodnější schůzky po konci každé iterace. Jde o **vyhodnocení sprintu a retrospektivu sprintu**. Zatímco v retrospektivě budou dále sdíleny spíše interní znalosti, které se osvědčili např. při analýze, při vyhodnocení sprintu budou sdíleny znalosti týkající se zákazníka (politické chování směrem k zákazníkovi, interní předpisy zákazníka atp.).

Každý člen projektového týmu je povinen identifikovat potencionální znalosti, které by mohly být zdokumentovány a dále sdíleny. Snaha o identifikaci takovýchto znalostí by měla probíhat neustále, proto je velice důležité, aby se taková znalost při svém odhalení dostala dále do oběhu. Tudíž lze využít praktiku **okamžité zpětné vazby**.

### 3.4.3 Definované role a odpovědnosti (3)

Každý člen týmu musí mít jasně definovanou roli, odpovědnosti a povinnosti a znát ostatní role. Aby bylo možné dobře sledovat veškeré role v projektu, je třeba použít vizuálních pomůcek. K tomu může posloužit **vizualizace workflow**. Právě zde můžou být dobře znázorněny všechny role, které v rámci projektu vystupují.

Pro dosažení vyšší efektivity je možné na členy týmu delegovat různé pravomoci a odpovědnosti. **Delegací pravomocí** se tak projektový manažer zbaví pracovního vytížení, které

mohou převzít právě členové týmu. Samozřejmě je nutné, aby byli o delegaci pravomocí a odpovědností informováni ostatní členové týmu.

Různé projekty mají různé role a s různými povinnostmi a odpovědnostmi. Mohou např. používat **týmy pro užité vlastnosti**. Je nutné deklarovat všechny povinnosti vně každého týmu pro užité vlastnosti i mezi ostatními týmy. Při užití týmů užitečných vlastností se také často používá **kolektivní vlastnictví kódu**, čímž vznikají další odpovědnosti a pravidla s touto praktikou spojené.

PRINCE2 definuje několik základních rolí, které jsou součástí projektového týmu. Mimo jiné i uživatele, kteří by měli být součástí týmu. Mnohdy se **účast uživatele** v projektu pomíjí, i když jde o velmi důležitou roli, která přináší důležité informace o dodávaném produktu.

#### 3.4.4 Řízení po etapách (4)

PRINCE2 je řízen po etapách, takže v duchu agilních metod lze používat místo etap **sprinty**. **Plánování sprintů** by mělo být v takové úrovni, aby je bylo možné efektivně řídit efektivně. Plánování opět probíhá pomocí nástroje **Planning Poker**.

Po konci každé etapy je nutné zapsat veškeré získané znalosti, Mimo znalostí je také nutné reportovat managementu a projektovému výboru. Tyto aktivity se dají praktikovat v rámci **retrospektivy sprintu** a **vyhodnocení sprintu**. V reportech pro management by mělo být obsaženo také množství odbavených požadavků za poslední sprint, za celou dobu projektu a také množství zbývajících požadavků. K tomu slouží reporty odvozené z **burndown chartu**.

#### 3.4.5 Řízení výjimkami (5)

Jelikož se tento princip řídí především nastavením tolerancí pro každou etapu, lze použít **definici toho, co je hotovo**. V tomto případě nejde o definici kvalitativních ukazatelů, které musí funkcionality splnit, aby ji bylo možno implementovat. Praktika bude vnímána jako nastavení hranic, které když vybraný aspekt projektu překročí, bude vzniklá situace reportována managementu. Pro kontrolu těchto nastavených hranic, je nutné provádět pravidelné kontroly. Pomocí pravidelných **inspekcí** týmu lze přesažení těchto hranic brzy zaregistrovat (při použití delegace pravomocí na nižší stupně managementu budou odpovědní členové týmu reportovat aktuální problémy).

Součástí tohoto principu je i delegace co nejvíce pravomocí na nižší stupně managementu. Proto je na místě použít praktiku **delegace pravomocí týmu**.

Aby bylo možné včas přijmout opravné (nebo preventivní) opatření, je nutné nastavit efektivní způsob dodání informací na ty správná místa. K tomu nejlépe poslouží užití **praktiky okamžité zpětné vazby**.

Pokud např. odhadovaná měsíční pracnost přesáhne očekávanou hodnotu o 20%, je na managementu učinit rozhodnutí, jak bude dále projekt pokračovat. Pokud je tato pracnost predikována i na další měsíce, je nutné přijmout změnu, která vrátí pracnost na požadovanou hodnotu (a tím se může zamezit zvýšení celkových nákladů projektu). Přesáhnutí této meze může samozřejmě plynout z nově zavedené změny, která se špatně implementovala, nebo byla jen nevhodně zvolena pro danou situaci. V takovém případě je nutné změny zvrátit. Obě výše zmíněné zkušenosti podchycují praktiky **přijímání změn** a **odvolatelnost změn**.

Pro rozhodování o dalším osudu projektu musí existovat efektivně nastavená **kommunikace mezi zainteresovanými stranami** (které jsou také součástí managementu, který dané rozhodnutí přijímá).

### 3.4.6 Zaměření se na produkty (6)

PRINCE2 je metodologie, která se spíše než na aktivity zaměřuje na výstupy. Proto je velice důležité dbát na produkt a jeho kvalitu. Aby byl výsledný produkt kvalitní, je nutné definovat kvalitativní metriky, které musí splňovat každá jeho dílčí dodávaná část. Pro tuto činnost se nejlépe hodí praktika **definice toho co je hotovo**.

Také je velmi důležité mít definováno, co bude výsledný produkt (jak bude dodáván, jaké budou jeho dílčí produkty atp.). Pro rozpis všech dodávaných částí je dobré využít **backlog**.

Pokud je realizováno příliš práce najednou (myšleno, že je najednou rozděláno příliš funkcionalit), vzniká nebezpečí snížení kvality produktu, jelikož se musí každý člen týmu myslet na příliš úkolů (a s nimi spojených problémů) najednou. Proto je žádoucí **limitovat rozdělanou práci**. Pro zlepšení kvality jednotlivých dílčích produktů je vhodné realizovat pouze to, co je nezbytně nutné pro jeho funkci (jinak řečeno je nutné **eliminovat odpad**). Dále pokud má být produkt co nejkvalitnější, je nutné zajistit neustálý přísun informací od zákazníka, jak je nejnovější verze produktu přijímána, a zda je nutné zvýšit kvalitu. Proto je žádoucí, aby byl **zákazník neustále dostupný** a fungovala **okamžitá zpětná vazba**. Aby mohl zákazník provádět kontrolu po jednotlivých funkcích, a aby se zamezilo velkému přísunu připomínek, využije se **inkrementálního vývoje**. Společně s touto praktikou se využije užívání **pravidelných buildů** (možnost prezentace současné verze zákazníkovi), **řízení konfigurací** (pro možnost vrátit se ke starší verzi, pokud nová postrádá požadovanou kvalitu), **budování integrity** (pro snížení počtu chyb), **doménového objektového modelování** (pro lepší návrh produktu), **vývoje pro užité vlastnosti** (pokud se využívá týmů pro užité vlastnosti), **individuálního vlastnictví tříd** (každá třída je vlastněna jedním členem týmu a tím se dá lépe kontrolovat kvalita), **CRC karet**, **kolektivního vlastnictví**

**kódu** (pro lepší distribuci nápadů), **párového programování**, **použití více modelů** (pro návrh produktu), **udržování co nejnižšího počtu objektů** (pro zamezení neefektivity), **předpokládání jednoduchosti** (nesnažit se aby byl produkt složitější, než je doopravdy zapotřebí), **účasti uživatele** (produkt bude využíván uživateli, tudíž je nutné znát jejich názor na konkrétní funkcionality) a užívání **User Stories**.

Poslední a jednou z nejdůležitějších použitelných praktik je **zaměření se na kritickou funkcionality**. A to z toho důvodu, že pro zákazníka je nejdůležitější funkční a použitelný produkt, než produkt, který má mnoho možných nastavení a funkcí, ale tu nejdůležitější nemá funkční.

### 3.4.7 Uzpůsobení řízení prostředí projektu (7)

Každá metodologie by měla být uzpůsobena konkrétní společnosti, její kultuře a organizační struktuře. Aby bylo možné lépe takovou metodologii implementovat, je nutné nejdříve **zajistit porozumění procesům** uvnitř společnosti (procesů, kterých se nasazení dané metodologie týká). Aby bylo možno co nejefektivněji porozumět všem relevantním procesům, je dobré zajistit **vizualizaci workflow**. Pokud je správně pochopeno, jakým způsobem práce v rámci procesů funguje, implementace nové metodologie bude mít vyšší šanci na úspěch. Samozřejmě uzpůsobení řízení prostředí projektu neznamena uzpůsobovat pouze metodologii, nýbrž také provést změny v procesech (většinou se jedná samozřejmě o drobné změny). Proto je důležité **přijímat a odvolávat změny** (zvláště odvolávat ty, u kterých se ukáže jejich neefektivita). Zda jsou tyto změny úspěšné lze zjisit pomocí řízení workflow. Pokud je detailně sledováno a **řízeno workflow** a funguje **okamžitá zpětná vazba**, je společnost schopna flexibilně reagovat na tyto změny.

## 3.5 Témata PRINCE2

Níže jsou rozepsány jednotlivé praktiky, které lze integrovat do témat PRINCE2. U každého tématu je v závorce uvedeno číslo, které slouží jako reference v tabulce Best Practices v závěru této práce.

### 3.5.1 Obchodní případ (1)

Pro použití obchodního případu jsou velice dobře uzpůsobeny **sprinty**. Pokud se sprinty ztotožní s etapami projektu (po konci každé etapy by měl být obchodní případ zaktualizován), lze po konci každého sprintu při realizaci **retrospektivy sprintu** (nebo i **vyhodnocení sprintu**) upravit obchodní případ.

Jelikož je důležité stále se zabývat opodstatněností projektu, je důležitá **neustálá dostupnost zákazníka a účast uživatele**. Z těchto dvou zdrojů lze získat velmi důležité informace týkající se další životaschopnosti projektu a na základě těchto informací poté upravit obchodní případ. Tyto informace mohou také přijít od ostatních zainteresovaných stran. Proto je podstatné udržovat **komunikaci všech zainteresovaných stran**.

### 3.5.2 Organizace (2)

Jelikož se u projektu jedná o dočasnou organizační strukturu, je nutné vědět, jaké role budou v projektu použity. Pro definici všech těchto rolí je nutné **vidět projekt jako celek**, aby bylo možné predikovat, které role opravdu budou potřeba, a které nikoliv. Jako další krok v definici rolí je nutné **zajistit porozumění procesu** (pokud není procesu porozumění, nelze určit veškeré v něm vystupující role). Role mohou přibývat či ubývat s postupem projektu, proto je důležité mít přehled, jak jednotlivé procesy fungují. K tomu poslouží **vizualizace a řízení workflow** (pro eventuální změny rolí). Při definici odpovědností je nutné se zamyslet, jaké odpovědnosti bude mít na starosti projektový manažer a jaké lze **delegovat na členy týmu**.

Při určování organizační struktury a rolí, je nutné se zamyslet, jakým stylem bude projekt veden. Pokud se vývoj produktu zaměří na užité vlastnosti, je vhodné používat i **týmy pro užité vlastnosti**. Dále pokud je nějaká role pro každý tým nepostradatelná, avšak ve společnosti je nedostatek kvalifikovaných lidí, lze tento nedostatek vyřešit za pomoci **točení lidí** s potřebnou kvalifikací napříč týmy.

V rámci tohoto tématu jsou také definované veškeré zainteresované strany a je tudíž při formování projektového týmu nutné, aby veškeré **zainteresované strany pravidelně komunikovaly**.

### 3.5.3 Kvalita (3)

Téma kvalita se zaměřuje výhradně na definici kvalitativních požadavků produktu. Tím pádem lze využít veškeré praktiky (týkající se kvality) v principu jako zaměření se na produkty. Jedná se o využití **inkrementálního vývoje, pravidelných, řízení konfigurací, budování integrity, doménového objektové modelování, vývoje pro užité vlastnosti, týmů pro užité vlastnosti, individuálního vlastnictví tříd, CRC karet, kolektivního vlastnictví kódu, párového programování, použití více modelů, udržování co nejnižšího počtu objektů, předpokládání jednoduchosti, účasti uživatele, User Stories, , backlogu, definice toho co je hotovo, zaměření se na kritickou funkcionalitu, eliminace odpadu a limitace rozdělané práce**.

Dalším aspektem tohoto tématu je přijímání změn. Proto je důležité **zajistit porozumění procesu** (aby bylo snáze uchopitelné, co se má změnit). Pro efektivní fungování tématu je

však důležité, aby byly **přijímány nové změny** a **odvolávány neefektivní změny**. Pro zjištění, zda tyto změny dosáhly kýženého efektu lze využít **řízení workflow** (které je dobré pro lepší představu mít **vizualizované**), a poté předat zjištěné informace **okamžitou zpětnou vazbu**.

### 3.5.4 Plány (4)

Pokud má být projekt správně naplánován, je nutné nejdříve vědět, co je všechno potřeba naplánovat a odhadnout pravděpodobné vnější faktory, rizika a příležitosti, které budou mít na projekt vliv. Proto se vyplatí znát celý projekt a to jak jeho dílčí části, tak i **vidět projekt jako celek**.

Pokud se plánuje projekt, je dobré znát více náhledů na problematiku. Tyto náhledy je nejlepší získat z různých modelů, a proto pokud se **použije více modelů**, má projektový manažer i lepší podklady pro odhady pracnosti projektu.

Před každou etapou se koná **plánování sprintu** (nebo etapy). Pro určení, které funkcionality budou, realizovány slouží **backlog** a pro odhad jejich pracnosti **Planning Poker**. Na plánování v rámci každého dne etapy je neefektivnější použít **denní scrum**. Pro přehled minulých etap a lepší predikce následující etapy by měl projektový manažer udržovat aktuální **burndown chart**.

### 3.5.5 Riziko (5)

Pro řízení rizik, je výhodné využití **sprintu**, jelikož pravidelné iterace jsou efektivním nástrojem pro řízení rizik (díky krátké době iterace). Při **plánování sprintu** lze vzít v potaz rizika identifikovaná při **vyhodnocení** a **retrospektivě** posledního ukončeného sprintu.

Pro správné identifikování rizik je nutné **zajistit porozumění procesu**, neboť u procesu, který je dobře zmapován a pochopen se lépe odhadují možná rizika a příležitosti.

Pravidelné **inspekce** pomáhají odhalit rizika včas a při použití **okamžité zpětné vazby** je dostatek času na realizaci preventivních opatření.

Následkem rizika je nějaké negativní událost. Pro odstranění následků takové události, nebo snížení rizika, případně jeho dopadu je důležité **přijímat změny**, nebo **odvolávat změny**, které se ukázaly jako neefektivní. Pro sledování vývoje rizik během projektu pomáhá **řízení workflow**.

### 3.5.6 Změna (6)

Změna je v téměř každém projektu nevyhnutelná. Pro **přijetí** a **odvolání** již nasazených změn je nutné, aby byl v projektu nastaven kontrolní mechanismus. Pro to nejlépe poslouží **vizualizace** (pro stanovení, která část procesu potřebuje změnu) a **řízení workflow**. Pro okamžité zjištění účinnosti změn (nebo potřeby zavedení nové) se použije **okamžitá zpětná vazba**. Každou změnu musí schválit veškeré relevantní zainteresované strany. Aby se mohly kvalifikovaně rozhodnout je nutné, aby **pravidelně komunikovaly**. Lze použít i praktiku **sprintu** (poté bude mít projektový manažer možnost implementovat změny v pravidelných intervalech).

Změnu může vyžadovat také samotná **metodologie**, **pokud nefunguje** jak má (nebo se od ní odchýlilo).

### 3.5.7 Průběh (7)

V tomto tématu se nastaví kontrolní mechanismy, které určí, zda aktuální průběh projektu koresponduje s plánem. Pro toto téma lze tudíž využít **definice toho, co je hotovo** (ve formě, která definuje překročení meze). Pro kontrolu, zda se daří odbavovat požadavky podle plánu, se použije **burndown chart**. Tyto informace budou analyzovány vždy na konci iterace a to při **vyhodnocení** a **retrospektivě sprintu**, případně při **denním scrumu** (nejaktuálnější průběh projektu). Pro nejaktuálnější obraz skutečnosti je vhodné praktikovat **pravidelné inspekce** týmů. Pokud je odhalena odchylka od plánu projektu, je nutné informovat management prostřednictvím **okamžité zpětné vazby**.

## 3.6 Procesy PRINCE2

Níže jsou rozepsány jednotlivé praktiky, které lze integrovat do procesů PRINCE2. U každého procesu je v závorce uvedeno číslo, které slouží jako reference v tabulce Best Practices v závěru této práce.

### 3.6.1 Proces zahájení projektu (1)

Při zahájení projektu je důležité, nastavit kontroly, které budou určovat, zda je projekt v průběhu projektu stále životaschopný (pomocí upravené praktiky **definice toho co je hotovo**). Stanoví se role a zainteresované strany (a nastaví se **kommunikace zainteresovaných stran**). Zvolí se také míra **delegace povinností mezi členy týmu** a nižší stupeň managementu.

### 3.6.2 Proces řízení projektu (2)

Celý proces řízení projektu lze plně substituovat za jakoukoliv agilní metodologii. Z tohoto faktu vyplývá, že v rámci tohoto procesu lze aplikovat **veškeré praktiky** definované v této práci. Aplikují se změny, zákazník neustále komunikuje s dodavatelem, vývoj je realizován v iteracích a plány se používají ke kontrole, že se projekt drží plánu.

### 3.6.3 Proces nastavení projektu (3)

Při nastavení projektu je důležité, aby bylo **vizuálně zobrazeno a řízeno workflow**. Díky tomu pak lze lépe plánovat v průběhu projektu. Jako další krok je **zajistit porozumění všech procesů** a i celého projektu. Dále se tvoří celkové plány projektu, čemuž dozajista pomůže **vidět projekt jako celek** (detailní plánování bude probíhat před každou realizační fází). Použijí se veškeré metody pro tvorbu modelů jako **doménové objektové modelování** (použijí se **CRC karty** a **více modelů**, ale přitom se bude **udržovat co nejmenší počet nutných modelů**) a vytvoří se kompletní **backlog**. Pokud se projekt bude zaměřovat na vývoj podle užitečných vlastností, je zformován tým pro **užité vlastnosti**. Všechny týmy jsou poté seskupeny do **společných prostor**.

### 3.6.4 Proces kontroly etapy (4)

Jako etapy mohou posloužit **sprinty**. Před začátkem sprintu proběhne **plánování sprintu**. V průběhu toho plánování jsou definovány kvalitativní požadavky na funkcionality (**definice toho co je hotovo**) a je vytvořen **backlog** (pomocí **User Storeis**) pro daný sprint. Odhad pracnosti je realizován pomocí **Planning Pokeru**. Jsou **přijímány změny** pro další sprint.

Před začátkem sprintu jsou v rámci prováděných změn **opraveny** veškeré nefunkční techniky dané metodologie.

**Zákazník by měl být neustále dostupný**, zvláště při plánovací fázi a při vyhodnocení sprintu)

### 3.6.5 Proces řízení dodání produktu (5)

Během tohoto procesu jsou kontrolovány a stanoveny kvalitativní požadavky (**definice toho co je hotovo**). Je určena také nutná **integrita** produktu. Každý manažer a všichni členové týmu rozumějí tomu, co má být vytvořeno a to je zaneseno do **backlogu**. Veškeré důležité informace jsou **sdělovány všem zainteresovaným stranám**. **Uživatel** je zapojen do procesu a dodá informace týkající se současné kvality. Pro udržení potřebné kvality je

nutná neustálá kooperace se zákazníkem (který by tudíž měl být **neustále dostupný**). Postup řešení všech funkcionalit lze sledovat pomocí **burndown chartu**.

### 3.6.6 Proces řízení přechodu mezi etapami (6)

Po konci etapy je **sprint vyhodnocen** a proběhne **retrospektiva**. Jsou vyhodnoceny výsledky z **burndown chartu** a pomocí **okamžité zpětné vazby** jsou veškeré důležité informace předány zainteresovaným stranám, které **komunikují** o potřebných změnách. Jsou **odvolány změny**, které nepřinesly žádaný účinek. Veškeré získané znalosti jsou zdokumentovány (v podobě informace) a jsou dále **rozvíjeny**.

### 3.6.7 Proces ukončení projektu (7)

Při ukončení projektu je nutné analyzovat, zda byly všechny plány splněny. Pomocí **burn-down chartu** lze zjistit, zda rozsah projektu byl realizován, jak bylo plánováno (i s možnými změnami požadavků, které se objevily během realizace projektu). V tom případě by v **backlogu** měly zůstat jen funkcionality s nejnižší prioritou, které nebyly realizovány na úkor změnových požadavků. Jsou zaznamenány veškeré znalosti, které budou **rozvíjeny** v dalších obdobných projektech. **Zainteresované strany komunikují**, jsou předány veškeré dokumenty a podepsány zákazníkem. Během poslední fáze projektu je důležité, aby byl **zákazník neustále dostupný** a fungovala **rychlá zpětná vazba** (aby bylo možné již hotový projekt uzavřít a použít zdroje jinde).

### 3.7 Výsledné tabulky

Tabulka 1 Tabulka problémů

| Číslo problému | Název problému                                                        | Best Practices                         |
|----------------|-----------------------------------------------------------------------|----------------------------------------|
| 1              | Neporozumění zadání                                                   | 1;2;3;6;8;15;17;20;22;23;<br>31;35;39  |
| 2              | Konflikt v týmu                                                       | 7;25;28;32;46                          |
| 3              | Zákazník není přístupný kompromisům<br>nebo nepochopil své povinnosti | 1;3;4;11;15;21;35                      |
| 4              | Člen týmu není tolik výkonný (nechápe<br>účel projektu)               | 5;6;12;15;17;20;21;22;23;<br>25;39;40  |
| 5              | Realizace příliš vysokého počtu projektů                              | 2;7;9;33                               |
| 6              | Špatná komunikace v rámci týmu                                        | 2;3;5;7;11;12;16;24;26;28;<br>32;45;46 |
| 7              | Špatná komunikace směrem k zákazníkovi                                | 2;3;11;34                              |
| 8              | Vágně definovaný rozsah                                               | 2;17;21;22;31;38                       |
| 9              | Příliš často měněný rozsah                                            | 9;8;18;31;37                           |
| 10             | Nízká flexibilita                                                     | 1;3;10;13;18;33;34;35;<br>42;44        |
| 11             | Špatné odhady pracnosti                                               | 7;14;17;21;23;27;36;                   |
| 12             | Příliš detailní a starostlivé řízení                                  | 19;28;45                               |
| 13             | Přílišné spoléhání na nástroje                                        | 9;10;13;15;37;45                       |
| 14             | Nedefinované (či málo definované) metriky                             | 1;4;10;15;20                           |
| 15             | Nedostatečné kapacity                                                 | 1;2;7;9;25;27                          |

|    |                                        |                                  |
|----|----------------------------------------|----------------------------------|
| 16 | Nedostatek kvalifikovaných zaměstnanců | 20;24;26;32;45                   |
| 17 | Konflikt priorit                       | 6;8;16;28                        |
| 18 | Nejasně definované priority            | 6;8;16;28                        |
| 19 | Nespolehlivý člen týmu                 | 24;28;32;46                      |
| 20 | Velké množství připomínek a chyb       | 4;17;21;29;30;33;34;36;<br>38;41 |

Tabulka 2 Principy PRINCE 2

| Číslo principu | Název principu                        | Best Practices                                                               |
|----------------|---------------------------------------|------------------------------------------------------------------------------|
| 1              | Neustálé zdůvodňování opodstatněnosti | 1;2;3;11                                                                     |
| 2              | Učení se ze zkušenosti                | 1;6;7;12;20;24;27;32;45                                                      |
| 3              | Definované role a odpovědnosti        | 5;11;16;19;26                                                                |
| 4              | Řízení po etapách                     | 6;7;12;14;18;27                                                              |
| 5              | Řízení výjimkami                      | 1;2;4;10;13;19;25                                                            |
| 6              | Zaměření se na produkty               | 1;3;4;11;16;17;21;23;24;26;<br>29;30;31;33;34;35;36;38;39;<br>40;41;42;43;44 |
| 7              | Uzpůsobení řízení prostředí projektu  | 1;5;9;10;13;15                                                               |

Tabulka 3 Témata PRINCE 2

| Číslo tématu | Název tématu    | Best Practices                                                                                  |
|--------------|-----------------|-------------------------------------------------------------------------------------------------|
| 1            | Obchodní případ | 2;3;7;11;12;18                                                                                  |
| 2            | Organizace      | 2;5;9;15;16;19;22                                                                               |
| 3            | Kvalita         | 1;3;4;5;8;9;10;11;13;15;16;<br>17;21;23;24;26;29;30;31;33;<br>34;35;36;38;39;40;41;42;43;<br>44 |

|   |        |                           |
|---|--------|---------------------------|
| 4 | Plány  | 6;8;14;17;22;27;28        |
| 5 | Riziko | 1;6;7;9;10;12;13;15;18;25 |
| 6 | Změna  | 1;2;5;9;10;13;18;37       |
| 7 | Průběh | 1;4;7;12;14;25;28         |

Tabulka 4 Procesy PRINCE 2

| Číslo procesu | Název procesu                       | Best Practices                    |
|---------------|-------------------------------------|-----------------------------------|
| 1             | Proces zahájení projektu            | 2;4;5;19                          |
| 2             | Proces řízení projektu              | Všechny praktiky                  |
| 3             | Proces nastavení projektu           | 5;8;9;15;16;17;21;22;23;30;<br>46 |
| 4             | Proces kontroly etapy               | 3;4;6;8;10;18;37                  |
| 5             | Proces řízení dodání produktu       | 2;3;4;8;11;14;29                  |
| 6             | Proces řízení přechodu mezi etapami | 1;2;7;8;12;13;14;20               |
| 7             | Proces ukončení projektu            | 1;3;14;20                         |

Tabulka 5 Tabulka Best Practices – celkové hodnocení

| Číslo praktiky | Praktika                          | Metodologie | Body PRINCE2 | Body Problémy | Body Celkem |
|----------------|-----------------------------------|-------------|--------------|---------------|-------------|
| 1              | Okamžitá zpětná vazba             | AM          | 60           | 4,5           | 64,5        |
| 2              | Komunikace zainteresovaných stran | DSDM        | 45           | 7             | 52          |
| 3              | Zákazník je vždy dostupný         | XP          | 40           | 4             | 45          |
| 4              | Definice toho co je hotovo        | Scrum       | 40           | 3             | 43          |
| 5              | Vizualizace workflow              | Kanban      | 40           | 2             | 42          |
| 6              | Plánování sprintu                 | Scrum       | 35           | 5             | 40          |
| 7              | Retrospektiva sprintu             | Scrum       | 35           | 5             | 40          |
| 8              | Backlog                           | Scrum       | 35           | 4             | 39          |
| 9              | Řízení workflow                   | Kanban      | 35           | 3             | 38          |
| 10             | Přijímat změny                    | AM          | 35           | 3             | 38          |

|    |                                        |         |    |     |      |
|----|----------------------------------------|---------|----|-----|------|
| 11 | Účast uživatele                        | DSDM    | 35 | 2   | 38   |
| 12 | Vyhodnocení sprintu                    | Scrum   | 35 | 2   | 37   |
| 13 | Odvolatelnost změn                     | DSDM    | 35 | 2   | 37   |
| 14 | Burndown chart                         | Scrum   | 35 | 1   | 36   |
| 15 | Zajistit porozumění procesu            | Kanban  | 30 | 5   | 35   |
| 16 | Tým pro užité vlastnosti               | FDD     | 30 | 3   | 33   |
| 17 | Použití více modelů                    | AM      | 25 | 4,5 | 29,5 |
| 18 | Sprint                                 | Scrum   | 25 | 2   | 27   |
| 19 | Delegace pravomocí týmu                | Lean    | 25 | 1   | 26   |
| 20 | Rozvíjet znalosti                      | Lean    | 20 | 4   | 24   |
| 21 | Doménové objektové modelování          | FDD     | 20 | 4   | 24   |
| 22 | Vidět projekt jako celek               | Lean    | 20 | 3,5 | 23,5 |
| 23 | CRC karty                              | XP      | 20 | 3   | 23   |
| 24 | Práce v páru                           | XP      | 20 | 3   | 23   |
| 25 | Inspekce                               | FDD     | 20 | 2,5 | 22,5 |
| 26 | Kolektivní vlastnictví                 | XP      | 20 | 2   | 22   |
| 27 | Planning Poker                         | Ostatní | 20 | 2   | 22   |
| 28 | Denní Scrum                            | Scrum   | 15 | 6   | 21   |
| 29 | Budování integrity                     | Lean    | 20 | 1   | 21   |
| 30 | Udržování co nejnižšího počtu objektů  | AM      | 20 | 1   | 21   |
| 31 | Rozhodovat se co nejpozději            | Lean    | 15 | 4   | 19   |
| 32 | Točit lidi                             | XP      | 15 | 4   | 19   |
| 33 | Limitovat rozdělanou práci             | Kanban  | 15 | 3   | 18   |
| 34 | Dodávat co nejdříve                    | Lean    | 15 | 3   | 18   |
| 35 | Řízení konfigurací                     | FDD     | 15 | 2,5 | 17,5 |
| 36 | Eliminace odpadu                       | Lean    | 15 | 2   | 17   |
| 37 | Opravení metodiky když se pokazí       | XP      | 15 | 2   | 17   |
| 38 | Zaměření se na kritickou funkcionalitu | DSDM    | 15 | 2   | 17   |
| 39 | User Stories                           | Ostatní | 15 | 2   | 17   |
| 40 | Vývoj podle užitečných vlastností      | FDD     | 15 | 1   | 16   |
| 41 | Individuální vlastnictví tříd          | FDD     | 15 | 1   | 16   |
| 42 | Pravidelné buildy                      | FDD     | 15 | 1   | 16   |
| 43 | Předpokládat jednoduchost              | AM      | 15 | 1   | 16   |
| 44 | Inkrementální přístup                  | AM      | 15 | 1   | 16   |
| 45 | Společně inovovat                      | Kanban  | 10 | 3,5 | 13,5 |
| 46 | Zajištění prostoru                     | XP      | 5  | 3   | 8    |

## 4 Závěr

Na základě výše realizovaného výzkumu lze konstatovat, že Best Practices agilních metodologií lze uplatnit i ve velmi popisné metodologii, jakou je PRINCE2. Díky integraci těchto praktik tak dojde k odlehčení celé metodologie, čímž lze dosáhnout efektivnějšího řízení v duchu agilních principů definovaných v Agilním manifestu. Vybrané praktiky byly s úspěchem integrovány do praktik, témat i procesů PRINCE2. Díky univerzálnosti těchto praktik lze docílit zefektivnění v rámci celého projektu, aniž by se projektový manažer (potažmo celá projektová kancelář) odchýlit od řízení v duchu PRINCE2. Všechny principy a témata zůstávají zachovány, naopak jsou podpořeny vhodnými praktikami. Při integraci praktik do procesů, bylo zjištěno, že druhý proces (proces řízení projektu) dokáže obsáhnout všechny vybrané praktiky. Z toho lze vyvodit, že celý tento proces může být nahrazen kompletní agilní metodologií (ať už si projektový manažer vybere kteroukoliv z nich), nebo jen zvolenou směsicí praktik, které nejlépe budou odpovídat situaci a kultuře společnosti.

Při pokusu identifikovat možné praktiky, které by pomohly vyřešit definované nejčastější problém projektového řízení, bylo odhaleno, že tyto praktiky mohou být velice snadno použity na odstranění těchto problémů, nebo alespoň na snížení rizika jejich výskytu nebo i zmírnění následků.

V tabulkách jsou uvedeny vybrané praktiky, které jsou seřazené podle své míry použitelnosti a provázanosti s jednotlivými problémy. Použitelností se v tomto případě myslí počet aspektů PRINCE2 a počet problémů, ve kterých lze tuto praktiku aplikovat. Je však potřeba mít na paměti, že každý problém je unikátní a nelze tudíž jednoznačně prohlásit, že použití dané praktiky přinese požadovaný efekt. Zda je v dané situaci možno praktiku použít musí rozhodnout projektový manažer na základě kontextu. Výše uvedené tabulky tak lze použít jako doporučení, které praktiky by měl projektový manažer zvážit jako možnou podporu při řešení konkrétního problému, nebo jakými praktikami odlehčit projekt řízený podle metodologie PRINCE2.

## Seznam literatury

- AKTHARSHA, U Syed. A Theory of Knowledge Management. Journal of Contemporary Research in Management. Coimbatore: PSG Institute of Business, 2011///Jan-Mar, 6(1), 103-119. ISSN 09739785. Dostupné také z: <http://search.proquest.com/docview/920826624?accountid=17203>
- AMBLER, Scott. The Principles of Agile Modeling (AM). In: Agile Modeling [online]. Calgary, Copyright2001-2014 [cit. 2016-04-17]. Dostupné z: <http://agilemodeling.com/principles.htm>
- ANDERSON, David J, David J Anderson and Associates INC. The Principles & general practices of the Kanban method [online]. Seattle WA:, 2010 [cit. 2016-04-16]. Dostupné z: <http://www.djaa.com/>
- ANDREEVA, Tatiana a Aino Kianto. Knowledge processes, knowledge-intensity and innovation: a moderated mediation analysis: a moderated mediation analysis. Journal of Knowledge Management. Kempston: Emerald Group Publishing, Limited, 2011, 15(6), 1016-1034. DOI: <http://dx.doi.org/10.1108/13673271111179343>. ISSN 13673270. Dostupné také z: <http://search.proquest.com/docview/902237592?accountid=17203>
- BARTOVIČOVÁ, Slavomíra. Zavedení agilního přístupu projektového managementu ve vybrané firmě. Brno 2014. Diplomová práce. Vysoké učení technické v Brně.
- BECK, Kent, James GRENNING a Mike BEEDLE et. al. *Manifesto for Agile Software Development* [online]. Ward Cunningham, ©2001 [cit. 2015-10-25]. Dostupné z: <http://agilemanifesto.org/>
- BORDÁS, Robert. Historie. Lean Company [online]. ©2006 [cit. 2015-10-16]. Dostupné z: <http://www.leancompany.cz/historie.html>
- BUONO, Anthony F a Flemming POULFELT. Challenges and issues in knowledge management. Greenwich, Conn.: Information Age Pub., c2005. ISBN 15-931-1419-2.
- COBB, CHARLES G. The Project Manager's Guide to Mastering Agile: Principles and Practices for an Adaptive Approach. John Wiley & Sons, 2015. ISBN 1118991761.
- COHN, Mike. Agile estimating and planning. Upper Saddle River, NJ: Prentice Hall Professional Technical Reference, c2006. ISBN 01-314-7941-5.
- ESKEROD, Pernille a Martina HUEMANN. Sustainable development and project stakeholder management: what standards say: what standards say. International Journal of Managing Projects in Business. Bingley: Emerald Group Publishing, Limited, 2013, 6(1), 36-50. DOI: <http://dx.doi.org/10.1108/17538371311291017>. ISSN 17538378. Dostupné také z: <http://search.proquest.com/docview/1268780803?accountid=17203>
- HONSOVÁ, Veronika. Projektové řízení v konkrétní společnosti - agilní vs. tradiční přístupy. Praha, 2013. Diplomová práce. Vysoká škola ekonomická v Praze

INFO ENTREPRENEURS. BEST PRACTICE [online]. Montreal: Info entrepreneurs, 2009 [cit. 2016-04-15]. Dostupné z: <http://www.infoentrepreneurs.org/>

MURRAY, Andy. Managing successful projects with Prince2. 5th ed. London: TSO, 2009, xii, 327 s. ISBN 978-011-3310-593.

PALMER, Stephen R a John M FELSING. A practical guide to feature-driven development. Upper Saddle River: Prentice Hall PTR, c2002. Coad series. ISBN 01-306-7615-2.

POPPENDIECK, Mary a Tom POPPENDIECK. Lean software development: an agile toolkit. 10. print. Boston: Addison-Wesley, 2006. ISBN 978-032-1150-783.

SELECT BUSINESS SOLUTIONS. What is DSDM? [online]. Trumbull, ©Copyright1988-2016 [cit. 2016-04-17]. Dostupné z: <http://www.selectbs.com/>

SADDINGTON, Peter. Agile pocket guide: a quick start to making your business agile using Scrum and beyond. Hoboken, N.J.: John Wiley, c2013. ISBN 9781118438251.

SUTHERLAND, Jeff a Ken SCHWABER. The Scrum Guide. In: Scrum Guides [online]. 2014 [cit. 2016-04-15]. Dostupné z: <http://www.scrumguides.org/scrum-guide.html>

WELLS, Don. The Rules of Extreme Programming. In: Extreme Programming:: A gentle introduction [online]. 1999 [cit. 2016-04-17]. Dostupné z: <http://www.extremeprogramming.org/rules.html>

VERSIONONE: Agile Project Management Software for Agile Development. About Us: Pioneers in Agile Programing & Management [online]. ©2015 [cit. 2015-11-03]. Dostupné z: <https://www.versionone.com/>

WESTER, Julia. What is Kanban? In: EVERYDAY KANBAN [online]. Seattle WA, 2016 [cit. 2016-04-16]. Dostupné z: <http://www.everydaykanban.com/what-is-kanban/>

ZATLOUKAL, Filip. Moderní metodiky vedení projektů (pro implementace zakázkového SW). Praha 2014. Diplomová práce. Vysoká škola ekonomická v Praze

## Seznam obrázků a tabulek

### Seznam obrázků

|            |                                                                     |    |
|------------|---------------------------------------------------------------------|----|
| Obrázek 1  | Obchodní případ (MURRAY;2009)                                       | 23 |
| Obrázek 2  | Čtyři úrovně řízení v PRINCE2(MURRAY;2009)                          | 25 |
| Obrázek 3  | Role v projektovém týmu a jejich závislosti(MURRAY;2009)            | 26 |
| Obrázek 4  | Kvalita v PRINCE2(MURRAY;2009)                                      | 28 |
| Obrázek 5  | Plány v PRINCE2(MURRAY;2009)                                        | 29 |
| Obrázek 6  | Řízení Rizik v PRINCE2 (MURRAY;2009)                                | 31 |
| Obrázek 7  | Procedura kontroly otevřených bodů a změn(MURRAY;2009)              | 33 |
| Obrázek 8  | Analýza možností(MURRAY;2009)                                       | 33 |
| Obrázek 9  | Procesy a Etapy v PRINCE2(MURRAY;2009)                              | 35 |
| Obrázek 10 | Proces zahájení projektu(MURRAY;2009)                               | 37 |
| Obrázek 11 | Proces řízení projektu(MURRAY;2009)                                 | 38 |
| Obrázek 12 | Proces nastavení projektu(MURRAY;2009)                              | 39 |
| Obrázek 13 | Proces kontroly etapy(MURRAY;2009)                                  | 41 |
| Obrázek 14 | Proces řízení dodání produktu(MURRAY;2009)                          | 42 |
| Obrázek 15 | Proces řízení přechodu mezi etapami(MURRAY;2009)                    | 43 |
| Obrázek 16 | Proces ukončení projektu(MURRAY;2009)                               | 45 |
| Obrázek 17 | Využívání jednotlivých agilních metodologií podle (VERSIONONE;2013) | 53 |

### Seznam tabulek

|           |                                            |    |
|-----------|--------------------------------------------|----|
| Tabulka 1 | Tabulka problémů                           | 91 |
| Tabulka 2 | Principy PRINCE 2                          | 92 |
| Tabulka 3 | Témata PRINCE 2                            | 92 |
| Tabulka 4 | Procesy PRINCE 2                           | 93 |
| Tabulka 5 | Tabulka Best Practices – celkové hodnocení | 93 |