

IT a anatomie firmy

(Metody a metodiky v řízení a řešení IT)

(pracovní dokument)



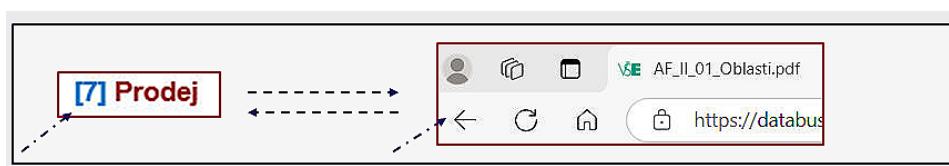
MBI tým

VŠE Praha, 2024

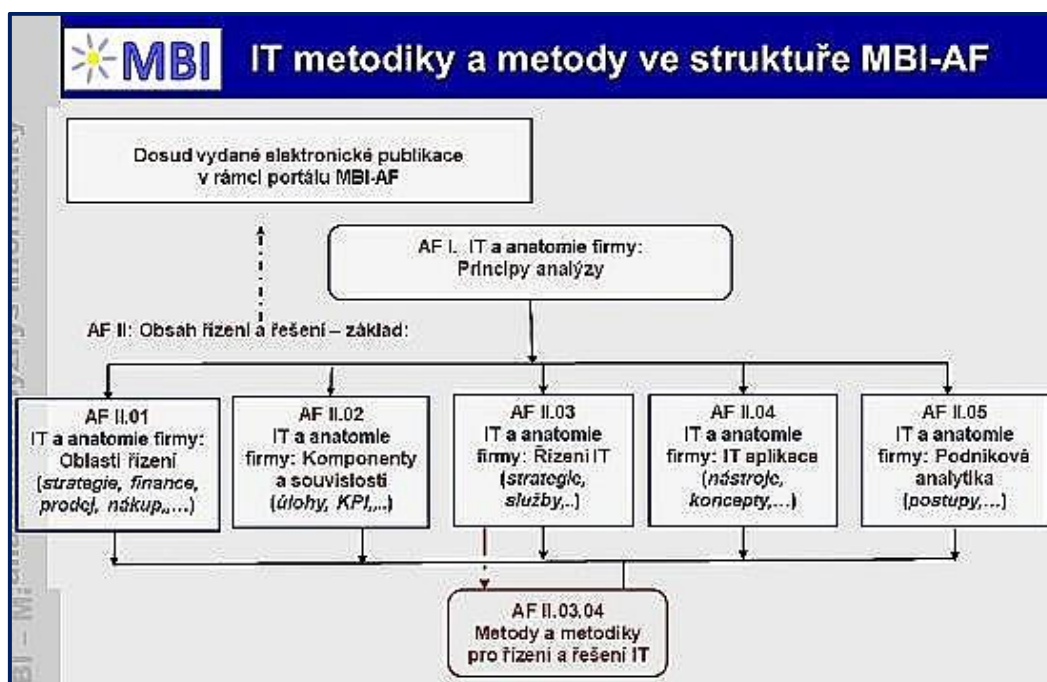


Poznámky k textu:

- V uvedeném schématu jsou v hranatých závorkách aktivní **odkazy na přechod na příslušné kapitoly**. Toto **platí v rámci celého dokumentu**. Návrat je na šipce v záhlaví stránky (viz obrázek):



- V dalším průběhu textu je k jeho oživení použita celá řada **obrázků a fotografií**. Ty jsou buď vlastní nebo „online obrázky“ firmy Microsoft, a to pouze s licencí **Creative Commons**, tj. jsou využity pouze pro studijní nikoli komerční účely.
- V textu jsou použity termíny „**podnik**“ i „**firma**“ ve stejném nebo obdobném smyslu. Termín „**firma**“ pokládáme za základní, ale v mnohém kontextu je využití termínu „**podnik**“, „**podnikový**“ apod. přirozenější. Využíváme je tak podle obvyklých použití v praxi.
- Účelem** tohoto dokumentu je vymezit **podstatné charakteristiky vybraných metod a metodik** v řízení a řešení IT. Kromě toho se obsah tohoto dokumentu chápe jako obecný základ pro definování a modifikace uplatnění těchto metodika a metod v různých typech firem a sektorech ekonomiky (strojírenská, maloobchodní a další).
- Dokument představuje doplnění** k dokumentu orientovanému na oblasti řízení IT. Jeho místo ve struktuře portálu MBI-AF ukazuje další obrázek:



Obrázek 1-1: Dokument metod ve struktuře dokumentů MBI-AF

Obsah

1. Úvodní poznámky a souvislosti	7
1.1 Komponenty řízení firmy	7
1.2 Řízení IT ve firmě	7
2. Metodiky, metody a rámce řízení IT	9
2.1 IT Infrastructure Library, ITIL	10
2.1.1 Celková charakteristika	10
2.1.2 Strategie služeb	11
2.1.3 Návrh služby	12
2.1.4 Přechod služeb	12
2.1.5 Provoz služeb	13
2.1.6 Neustálé zlepšování služeb.....	14
2.1.7 Struktura a principy ITIL 4	14
2.1.8 Guiding Principles.....	16
2.1.9 Governance	17
2.1.10 Service Value Chain.....	17
2.1.11 Continual Improvement (Neustálé zlepšování)	17
2.1.12 Practices (Praktiky)	18
2.1.13 General management practices.....	18
2.1.14 Service management practices	20
2.1.15 Technical management practices.....	21
2.1.16 Poznámky, reference	22
2.2 CobiT	22
2.2.1 Celková charakteristika	22
2.2.2 Poznámky, reference	22
2.3 CMMI	23
2.3.1 Celková charakteristika	23
2.3.2 Poznámky, reference	23
2.4 IT Balanced Scorecard	23
2.4.1 Celková charakteristika	23
2.4.2 Poznámky, reference	23
2.5 Test Maturity Model Integration, TMMI.....	23
2.5.1 Celková charakteristika	23
2.5.2 Vývoj a obsah TMMi	23
2.5.3 Komponenty TMMi	24
2.5.4 Poznámky, reference	24
2.6 Multidimensional Management and Development of Information System, MMDIS.....	24
2.6.1 Celková charakteristika	24
2.6.2 Poznámky, reference	25
2.7 TOGAF	25
2.7.1 Celková charakteristika	25
2.7.2 Poznámky, reference	25
2.8 Architecture Development Method, ADM	26
2.8.1 Celková charakteristika	26
2.8.2 Poznámky, reference	27
2.9 Zachmanův rámec	27
2.9.1 Celková charakteristika	27
2.9.2 Poznámky, reference	29
3. Metodiky a metody řešení IT.....	30

3.1	Project Management Body of Knowledge, PMBOK	31
3.1.1	Celková charakteristika	31
3.1.2	Obsah PMBOK	31
3.1.3	Procesní skupiny	31
3.1.4	Znalostní oblasti	31
3.1.5	Nástroje měření kvality	32
3.1.6	Tvorba plánů v PMBOK	32
3.1.7	Poznámky, reference	32
3.2	Projects in Controlled Environments, PRINCE2	33
3.2.1	Celková charakteristika	33
3.2.2	Principy	33
3.2.3	Parametry výkonnosti	34
3.2.4	Procesy	34
3.2.5	Zahájení projektu (SU, Starting up a Project)	34
3.2.6	Směrování projektu (DP, Directing a Project)	34
3.2.7	Nastavení projektu (IP, Initiating a Project)	34
3.2.8	Řízení etapy (CS, Controlling a Stage)	34
3.2.9	Řízení dodávky produktu (MP, Managing Product Delivery)	35
3.2.10	Řízení přechodu mezi etapami (SB, Managing a Stage Boundary)	35
3.2.11	Ukončení projektu (CP, Closing a Project), U405A)	35
3.2.12	Témata	35
3.2.13	Poznámky, reference	37
3.3	Scrum	37
3.3.1	Metodika Scrum	37
3.3.2	Pilíře	38
3.3.3	Základní pojmy a artefakty	38
3.3.4	Role	40
3.3.5	Činnosti	41
3.3.6	Škálování agilních metodik	43
3.3.7	Agile Scaling Model (ASM)	44
3.3.8	Large Scale Scrum (LeSS)	44
3.3.9	Poznámky, zdroje	46
3.4	Azure DevOps	48
3.4.1	Metodika DevOps	48
3.4.2	Azure DevOps	49
3.4.3	Azure DevOps Services vs. Azure DevOps Server	50
3.4.4	Funkce Azure DevOps	50
3.4.5	Azure Repos	50
3.4.6	Azure Pipelines	51
3.4.7	Azure Boards	51
3.4.8	Azure Test Plans	51
3.4.9	Zdroje	52
3.5	Rational Unified Process (RUP)	53
3.5.1	Vývoj a obsah RUP	53
3.5.2	Životní cyklus RUP	54
3.5.3	Fáze RUP	54
3.5.4	Poznámky, reference	55
3.6	Unified Modeling Language (UML)	55
3.6.1	Celková charakteristika	55
3.6.2	Typy diagramů UML	56
3.6.3	Use Case diagramy	56
3.6.4	Diagramy aktivit	60
3.7	Metodika CRISP-DM	61
3.7.1	Celková charakteristika	61
3.7.2	Porozumění problematice/zkoumané oblasti	61
3.7.3	Porozumění datům	61
3.7.4	Modelování	62

3.7.5	Využití výsledků	63
3.7.6	Poznámky, reference;	63
3.8	Datové modelování.....	63
3.8.1	Účel datového modelování.....	63
3.8.2	Relační databáze.....	64
3.8.3	Řešení datových modelů	65
3.8.4	Podstatné charakteristiky datových modelů	65
3.8.5	Normalizace databáze	66
3.8.6	Poznámky, reference	66
3.9	Procesní modelování	66
3.9.1	Procesní modelování: celková charakteristika	66
3.9.2	Procesní charakteristiky	66
3.9.3	Řízení podnikových procesů	67
3.9.4	Monitorování podnikových procesů.....	67
3.9.5	Business Process Reengineering, BPR.....	67
3.9.6	Poznámky, reference	68
3.10	Dimenzionální modelování	68
3.10.1	Dimenzionální modelování, celková charakteristika.....	68
3.10.2	Dimenzionální modelování, hrubý dimenzionální model	68
3.10.3	Dimenze a jejich charakteristiky	68
3.10.4	Ukazatelé a jejich charakteristiky.....	69
3.10.5	Návrh tabulek faktů.....	69
3.10.6	Návrh dimenzionálních tabulek.....	70
3.10.7	Poznámky, reference	70
4.	Závěry.....	72
5.	Zdroje.....	73

1. Úvodní poznámky a souvislosti



Pracovní metody a metodiky představují **významnou komponent řízení IT i řešení IT**. **Smyslem tohoto textu** je vymezit podstatu a hlavní charakteristiky vybraných metod a metodik a jejich místo v řízení firmy i v řízení a řešení IT.

Následující text obsahuje základní vymezení dokumentu a jeho vazby na ostatní dokumenty na portálu MBI-AF.

1.1 Komponenty řízení firmy

Dokument obsahuje pouze základní přehled komponent řízení ve firmě a vazby mezi nimi, jako např. úlohy, dokumenty, metody apod. Strukturu dokumentu představuje Obrázek 1-1:



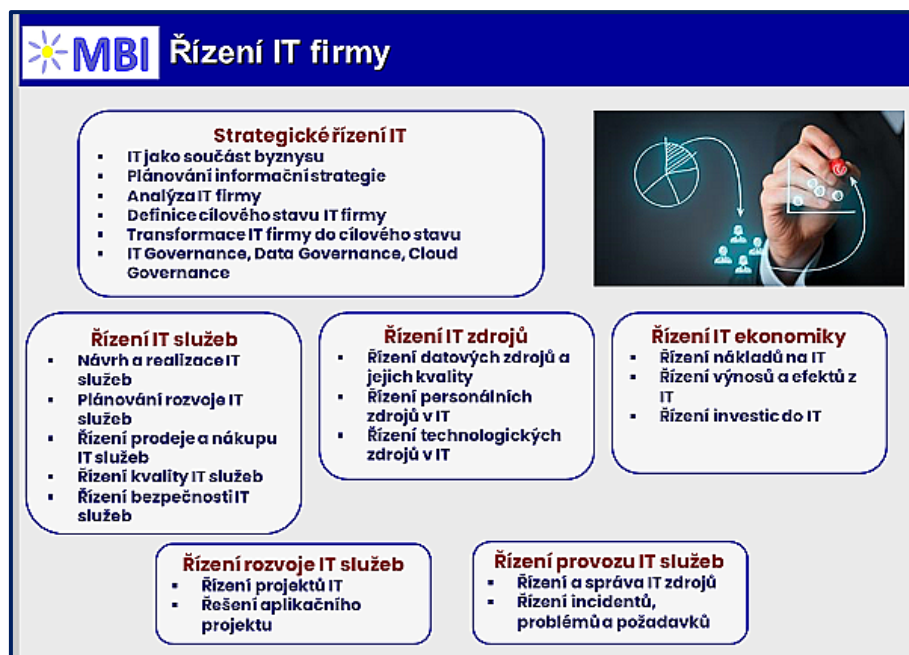
Obrázek 1-1: Komponenty řízení firmy a jejich souvislosti



[[Komponenty a souvislosti](#)]

1.2 Řízení IT ve firmě

Dokument je věnovaný jednotlivým oblastem řízení IT ve firmě firmy. V rámci tohoto dokumentu je rovněž definováno uplatnění uvedených metod a metodik. Strukturu dokumentu představuje Obrázek 1-2:



Obrázek 1-2: Domény a oblasti řízení IT ve firmě



[[Řízení IT](#)]

Další části dokumentu jsou věnovány **vymezení vybraných metod a metodik a jejich charakteristikám** spojených s řízením a funkcemi firmy, i IT. Podstatně komplexnější **přehled metod** je také např. na portále na adrese <https://www.valuebasedmanagement.net/>

2. Metodiky, metody a rámce řízení IT



Účelem kapitoly je:

- vymezit podstatu a základní charakteristiku **pouze vybraných metod a metodik**, které se váží k řízení IT,
- pro **detailnější charakteristiky** metodik a metod jsou uvedeny **reference** na některé další publikace.

Mapa metodik a metod řízení IT firmy (s odkazy)

[2.1] ITIL: IT Infrastructure Library			
[2.2] COBIT			
[2.3] CMMI	[2.4] IT Balanced Scorecard	[2.5] TMMI: Test Maturity Model Integration	[2.6] MMDIS
[2.7] TOGAF		[2.8] ADM	[2.9] Zachman

Další podkapitoly obsahují pouze rámcové **vymezení obsahu** vybraných **metodik, metod a rámců v řízení IT** firmy, a to:

- **IT Infrastructure Library, ITIL** představuje veřejný procesní rámec popisující nejlepší praktiky při správě služeb IT,
- **CobiT** obsahuje **popis sady procesů, návodů, cílů, hodnocení**, nejlepších praktik a ukazatelů,
- **CMMI** je model pro zlepšování procesů v oblasti rozvoje služeb,

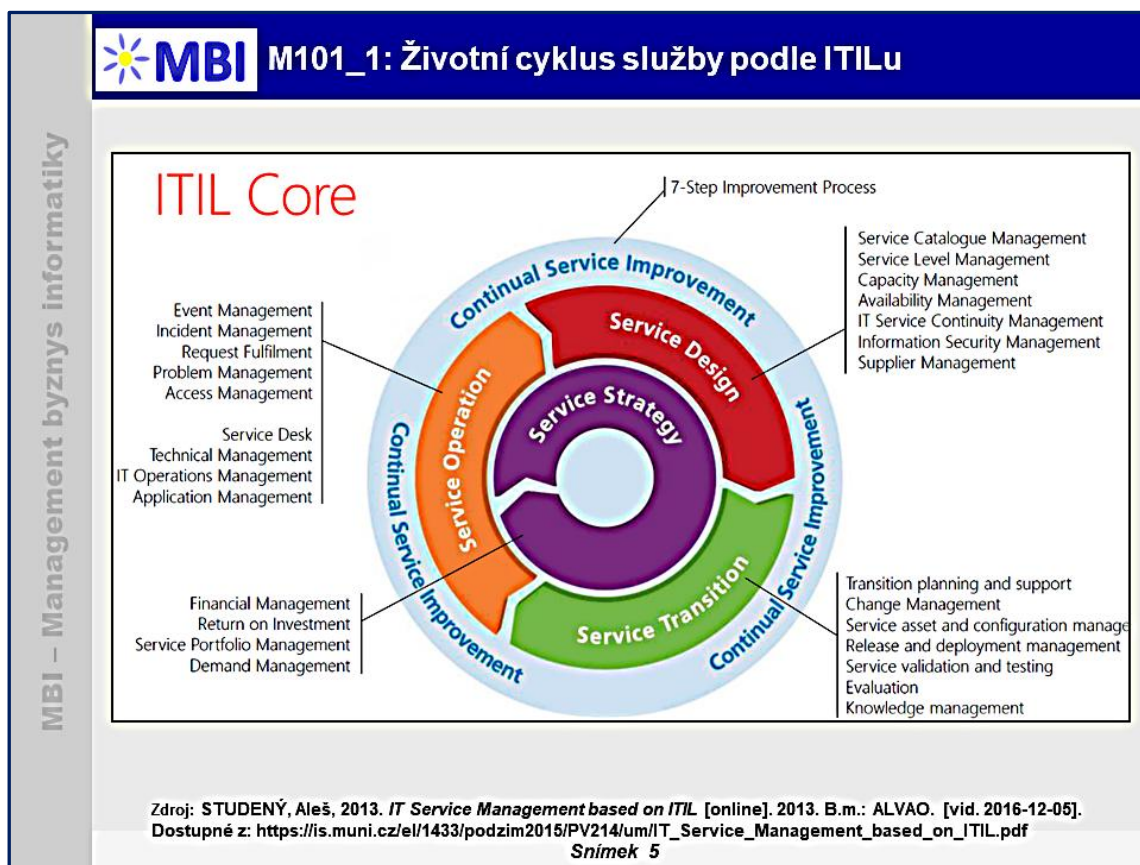
- **IT Balanced Scorecard** je nástroj pro řízení podnikové informatiky a vztahu mezi informatikou a byznysem.
- **Test Maturity Model Integration, TMMI** je jedním z nejnovějších modelů pro zlepšování procesů testování,
- **Multidimensional Management and Development of Information System, MMDIS** představuje vývoj, údržbu a provoz komplexního a integrovaného informačního systému podniku,
- **TOGAF** je nejpoužívanějším rámcem podnikové architektury,
- **Architecture Development Method, ADM** popisuje, jak vybudovat pro daný podnik specifickou podnikovou architekturu, která adresuje byznys požadavky,
- **Zachmanův rámec** akcentuje změny v náhledu na podnikovou architekturu i zkušenosti a chyby spojené se zaváděním podnikové architektury.

2.1 IT Infrastructure Library, ITIL

Svatošová, M. (KIT, VŠE)

2.1.1 Celková charakteristika

- ITIL představuje **veřejný procesní rámec** (Framework) popisující **nejlepší praktiky při správě služeb IT**.
- ITIL poskytuje **nástroje** pro IT governance, management a kontrolu IT služeb.
- Jako deskriptivní metodika poskytuje **flexibilitu pro přizpůsobení** v něm definovaných doporučení. Nejedná se o standard, který by musel být striktně dodržován. Jedná se **spíše o návod**, který by měl být pochopen a použit pro vytvoření přidané hodnoty pro poskytovatele služeb a jejich zákazníky. (Bucksteeg 2012).
- Je zaměřen na **proces kontinuálního zlepšování** u poskytovaných služeb. Na zlepšování služeb je nahlíženo **ze dvou perspektiv**, jak ze **zákaznické**, tak z **byznys** perspektivy.
- V květnu 2007 přichází **ITIL V3**, který sestává již pouze z **5 knih**. Klíčovou změnou je **postavení celého frameworku na komplexním životním cyklu služeb**, podle nějž je rozdělen i obsah jednotlivých publikací. Další změny se dotkly i aktualizace terminologie.
- V roce 2011 byla vydána aktualizovaná verze ITIL 2011. Tato aktuální verze nepřichází s žádnou výraznou úpravou ohledně filozofie celého frameworku, pouze reorganizuje některé části a doplňuje terminologii. (Bucksteeg 2012).
- Jádru ITILu se skládá z **pěti základních publikací**:
 - Service Strategy (Strategie služeb),
 - Service Design (Návrh služeb),
 - Service Transition (Přechod služeb),
 - Service Operation (Provoz služeb),
 - Continual Service Improvement (Neustálé zlepšování služeb).
- Každá z těchto knih popisuje **různé fáze životního cyklu služby**



Obrázek 2-1: Životní cyklus služby

2.1.2 Strategie služeb

- Strategie služeb stojí přímo v centru životního cyklu služby. Základem této knihy je **návrh, vývoj a implementace správy služeb jako strategického aktiva**, které je zde postaveno do strategického kontextu s podnikáním poskytovatele služeb. Strategie služeb ovlivňuje všechny životní fáze služby.
- Důraz je kladen na celkové **strategické pojetí nabídky služeb** s ohledem na výkon, potenciál a klíčové kompetence.
- Mezi **hlavní procesy** patří:
 - správa služeb IT,
 - správa portfolia služeb (Service Portfolio Management),
 - správa financí služeb IT (Financial Management),
 - správa požadavků (Demand Management),
 - správa vztahů s businesssem (Business Relationship Management).
- Další oblasti** jsou:
 - možnosti vytvoření tržního prostoru,
 - charakterizace typů externích a interních poskytovatelů služeb, aktiv služby,
 - výčet strategických rizik. (Bucksteeg 2012).
- Knihla poskytuje **návod** k nalezení odpovědí na klíčové **otázky ohledně architektury a formy služeb**, například:
 - Jaké služby budou poskytovány?
 - Komu budou poskytovány?

- Jak bude zákazník nebo stakeholdery vnímána a měřena poskytnutá hodnota?
- Jak bude měřena výkonnost služby? (TSO 2012).

2.1.3 Návrh služby

- Na základě požadavků stanovených v servisní strategii dochází k **návrhu charakteristik služby**. Hlavním cílem návrhu služby je zajistit to, aby byly nové, či změněné služby **navrženy v souladu s měnícími se požadavky byznysu**. Takové služby musí splňovat požadavky z hlediska účelnosti, výkonnosti a dostupnosti.
- **Klíčové aktivity** této fáze životního cyklu zahrnují plánování a koordinaci navrhovacích aktivit, jejímž prostřednictvím je zajištěn konzistentní návrh služby, správy služeb informačních systémů, architektury, použitých technologií, procesů, metrik a správy rozhraní.
- Mezi **základní principy** patří například tzv. princip 4p (The 4 Ps of ITIL), který pomáhá identifikovat základní zdroje pro zajištění poskytované služby. Tyto zdroje jsou: lidé (People), produkty (Products), procesy (Processes), partneři (Partners). (TSO 2013a).
- Kniha definuje **následující procesy**:
 - správa katalogu služeb (Service Catalogue Management),
 - správa úrovně služeb (Service Level Management),
 - správa kapacit (Capacity Management),
 - správa dostupnosti (Availability Management),
 - správa kontinuity služeb IT (IT Service Continuity Management),
 - správa bezpečnosti informací (Information Security Management),
 - správa dodavatelů (Supplier Management). (Bucksteeg 2012).

2.1.4 Přechod služeb

- Kniha se zabývá fází přechodu služby v rámci níž, je **navržená služba převedena do ostřejího provozu**.
- Hlavním cílem této fáze je zajištění, že **zaváděná služba bude vyhovovat očekáváním byznysu** tak, jak byly stanoveny v rámci procesu strategie služeb a návrhu služeb. Taková nová, či upravená služba musí reflektovat obchodní požadavky a cíle.
- **Klíčové aktivity** zahrnují plánování a řízení změn a releasů, řízení rizik, přenos znalostí, nastavení očekávání a zajištění, že očekávaná hodnota bude dodána.
- Při realizaci těchto aktivit je nutné dodržovat **následující principy**: porozumění užitečnosti, záruce a účelu u všech služeb, řízení komplexity spojené se změnami služeb, technologií a procesů, zavedení formalizované politiky a společného rámce pro implementaci změn, podpora přenosu znalostí a zajištění angažovanosti personálu podílejícího se na přechodu služby. (TSO 2013b).
- Pro tuto fázi jsou definovány **následující procesy**:
 - plánování a podpora přechodu (Transition Planning and Support),
 - správa změn (Change Management),
 - správa aktiv služeb a konfigurací (Service Asset and Configuration Management),
 - správa releasů a provozního nasazení (Release and Deployment Management),
 - validace a testování služby (Service Validation and Testing), vyhodnocení změny (Evaluation),
 - správa znalostí (Knowledge Management). (Bucksteeg 2012).

2.1.5 Provoz služeb

- Kniha se věnuje provozu služeb - správy a řízení služeb. Důraz je kladen na **zvyšování účinnosti a efektivnosti**, jak samotných poskytovaných služeb, tak i jejich podpory.
- **Cílem** je dodávání stabilní nákladově efektivní služby **poskytující zákazníkovi požadovanou přidanou hodnotu**. Jako v jediné ze všech částí životního cyklu v ní služba vytváří hodnotu pro byznys, a je tak prostředkem pro realizaci strategických cílů organizace, proto musí být této fázi věnována náležitá pozornost.
- Ze všech fází životního cyklu také nejvíce **přichází do kontaktu s koncovými zákazníky**. (TSO 2012).
- Kvalita provozu služeb zásadním způsobem **ovlivňuje spokojenost byznysu s celým podnikovým IT**. V knize jsou proto obsaženy návody, jakým způsobem **zajistit stabilitu služeb**, při současném zachování jejich připravenosti na změny v jejich návrhu, v měřítku, v rozsahu a změny v úrovni jejich poskytování. (TSO 2013c).
- ITIL definuje celkem **5 procesů** obsahujících návody pro podporu a údržbu IT služeb. (TSO 2012):
 - *Správa událostí* - hlavní úlohou správy událostí je identifikace a analýza různých typů událostí, na jejichž základě jsou vyvozena adekvátní opatření. Přičemž událostí lze rozumět skutečnost, kdy něco v systému funguje nesprávně a tato situace může potenciálně vést k vytvoření incidentu. Proces rozlišuje podle závažnosti následující typy událostí: ---informaci- – zpráva o běžné provozním stavu, varování – systém vykazuje podezřelý výkonnostní, či funkční charakteristiky, výjimku – funkčnost systému je podstatně ohrožena, nebo již došlo k jeho výpadku. Při správné implementaci slouží jako spouštěč (trigger) procesů správy incidentů případně problémů. (TSO 2013c).
 - **Správa incidentů** - proces správy incidentů má za úkol obnovení normální služby, tak rychle, jak jen to bude možné, a co s možná nejmenšími dopady na byznys. Incident je chápán jako: „Neplánované přerušení nebo snížení kvality služby IT. I výpadek konfigurační položky (CI) bez dosavadních dopadů na službu představuje incident.“ (Bucksteeg 2012) V rámci procesu dochází ke kategorizaci a prioritizaci příchozích incidentů, případně k jejich eskalaci vyšším úrovním podpory, pokud nemohou být vyřešeny přímo service deskem. (TSO 2013c).
- *Plnění požadavků* - v rámci procesu plnění požadavků dochází k vyřizování uživatelských požadavků nemajících přímou souvislost s poruchou, nebo přerušením služby. Pod pojmem požadavek si lze představit například dotazy na informace související s tím, jak ovládat aplikaci, či na poskytnutí jinak nepřístupných dat. Všeobecně lze říci, že až na výjimky se jedná o činnosti s nízkou důležitostí, ohledně technické funkcionality aplikací a s nimi souvisejícími službami. Nicméně mohou mít rozhodující vliv na spokojenost zákazníka s podporou. (TSO 2013c).
- **Správa problémů** - účelem tohoto procesu je předcházet chybám, které v konečném důsledku vedou k poruchám při dodávání služby. Evidencí a investigací opakujících se incidentů dochází proaktivně k jejich odstraňování, případně ke zmírnění dopadu těch, kterým není možné předejít. V rámci procesu vzniká databáze známých problémů (Known Error Database) s popsáními návody k jejich řešení (Workaround). Hlavním rozdílem mezi správou problémů a správou incidentů je to, že správa problémů se snaží nalézt primární příčinu výpadku služby, kdežto správa incidentů se spokojí „jen“ s obnovením poskytování služby, když je to nutné. (Bucksteeg 2012).
- **Správa přístupů** - hlavním cílem správy přístupů je poskytovat oprávněným uživatelům práva k využívání služby, nebo skupiny služeb. Zatímco neoprávněným uživatelům je tento přístup zamítnut. Správa přístupů chrání důvěrnost, dostupnost, integritu dat organizace stejně jako její intelektuální hodnoty. (TSO 2013c).

2.1.6 Neustálé zlepšování služeb

- Kniha se zaměřuje na **průběžné přizpůsobování služeb stále se měnícím obchodním požadavkům**. Neustálé zlepšování služeb kombinuje principy, praktiky a metody z managementu kvality, správy změn a modelů procesní zralosti.
- Stěžejním stavebním kamenem je **Demingův cyklus** (Plan-Do-Check-Act). (TSO 2013d).
- **Obsahem jeho jednotlivých fází je:** plánuj – kontrola potenciálu ke zlepšení aktuálního stavu, dělej – aplikace vylepšení, kontroluj – kontrola toho, zda změny byly aplikovány podle předpokladu, jednej – korekce zjištěných odchylek, změny plánů, nebo zlepšení správy kvality.
- ITIL tento cyklus nepřebírá zcela beze změny a **specifikuje 7 dílčích kroků:** identifikuj strategii pro zlepšení, definuj, co budeš měřit, shromáždíš data, přizpůsob data, analyzuj informace a data, prezentuj a použij data, implementuj zlepšení. (TSO 2012).

2.1.7 Struktura a principy ITIL 4

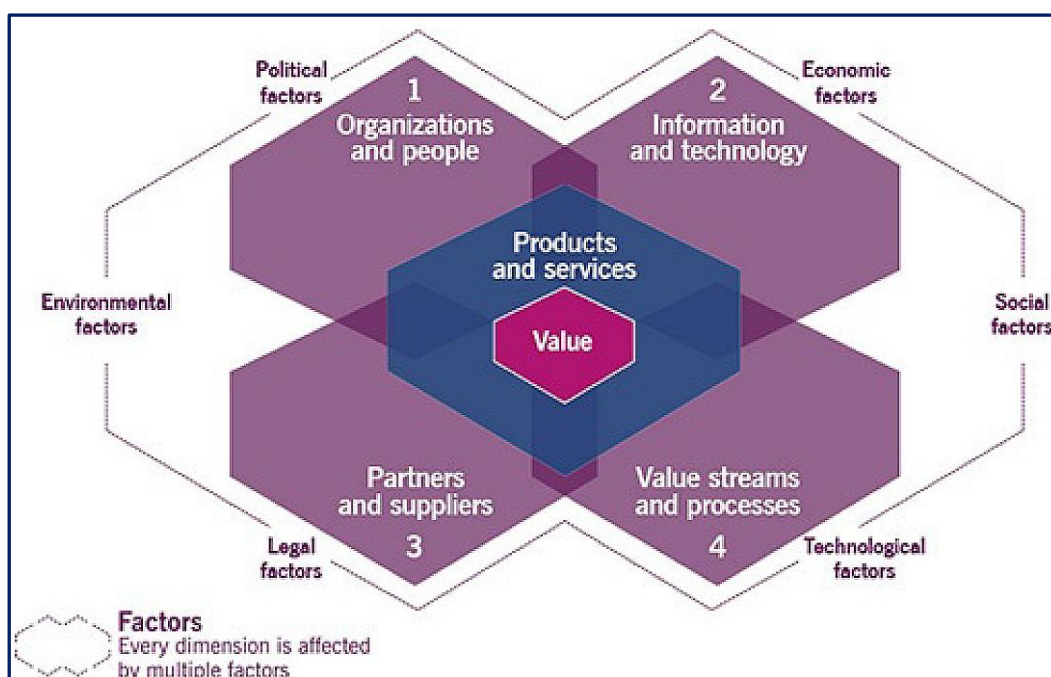
V této části je podrobně popsána struktura ITILu 4. Hlavním zdrojem pro tuto část byla publikace YASM. ITIL 4 vs. ITIL V3 | YaSM Wiki. YaSM [online]. 2019 [vid. 2020-05-20].

2.1.7.1 Dimenze

ITIL 4 definuje 4 hlavní dimenze, které jsou společně zásadní pro efektivního fungování služeb. Představují holistický a celostní přístup k řízení služeb. Na každou službu by se mělo nahlížet z pohledu všech těchto dimenzí. Pokud by nějaká z nich byla vynechána nebo jí byla věnována nedostatečná pozornost, mohlo by to vést k snížení kvality služby nebo k její nedostatečné efektivitě.

Dimenze jsou následující:

- Organizations and people (Organizace a lidé)
- Information and Technology (Informace a technologie)
- Partners and Suppliers (Partneři a dodavatelé)
- Value streams and processes (Hodnotové proudy a procesy)



Obrázek 2-2: 4 dimenze dle ITILu 4

Na obrázku jsou znázorněny jednotlivé dimenze a jejich propojení. Uprostřed celého znázornění je Value (Hodnota), kterou vytváření produkty a služby. Ty jsou ovlivňovány v 4 definovaných dimenzích. Zvenku působí vnější vlivy, které jednotlivé dimenze ovlivňují (politické, ekonomické, právní, technologické faktory a faktory prostředí.)

2.1.7.2 Organization and people (Organizace a lidé)

Dimenze se týká celé organizace jako celku (nejen IT). Každá organizace má mít organizační kulturu, která dostatečně podporuje její cíle. Organizace by měla mít definovanou jasnou vizi, misi a hodnoty, se kterými by její zaměstnanci měli být seznámeni a měli by se s nimi pokud možno ztotožňovat.

Důležité je mít nastavené správné procesy managementu, zvolenou organizační strukturu, definované role a odpovědnosti pro celou organizaci.

Klíčovým elementem v této dimenzi jsou lidé. Jedná se o stakeholdery služby, mezi něž patří zaměstnanci, dodavatelé i zákazníci. Co se týče zaměstnanců, ti by měli rozumět jejich roli a kompetencím a měli by být začleněni do organizace. Zaměstnanci jsou vybíráni nejen podle svých tvrdých dovedností (technické dovednosti, znalosti programů atd.) ale i podle svých tzv. měkkých dovedností (schopnost vést tým, komunikace atd.). Každá osoba v organizaci by měla mít jasnou představu, jakým způsobem přispívá do vytváření celkové hodnoty pro organizaci, zákazníka nebo další stakeholdery.

2.1.7.3 Information and technology (Informace a technologie)

Tato dimenze obsahuje jednak informace a znalosti, které jsou nezbytné pro řízení služeb a jednak potřebné technologie.

Potřebné informace jsou závislé na povaze dané služby. Organizace by si měla zodpovědět tyto otázky: jaké informace jsou ve službě používány a zpracovávány, jaké informace jsou potřebné k dodání služby a jakým způsobem je s informacemi nakládáno, jak se ukládají a jak se k nim přistupuje. Důležité je také určit, jak se informace předávají mezi jednotlivými službami. Informace potřebné pro provozování služby musí být dostupné, spolehlivé a přesné.

Technologická část této dimenze zahrnuje veškeré úrovně služby – aplikace, databáze, komunikační systémy a jejich integrace. Pokud organizace zvažuje zavedení nové technologie, musí si zodpovědět tyto důležité otázky:

- zda je daná technologie kompatibilní s již existující architekturou
- jaká nová rizika mohou vzniknout s použitím této technologie
- jestli neplynou z použití nové technologie nějaké regulační omezení (legislativa, nebo i předpisy v rámci organizace)
- jestli zaměstnanci mají potřebné znalosti, aby technologii mohli správně ovládat

2.1.7.4 Partners and suppliers (Partneři a dodavatelé)

Tato dimenze se zejména týká vztahu organizace s dalšími organizacemi, které jsou zahrnuty v návrhu, vývoji, nasazení, dodávání, podpoře nebo kontinuálního zlepšování služeb. Vztahy mezi organizacemi mohou být na různé úrovni formálnosti a integrace.

2.1.7.5 Value streams and processes (Proudý hodnot a procesy)

Tato dimenze se zabývá zejména tím, jak jsou jednotlivé části organizace integrovány a jak spolupracují na vytváření hodnoty pomocí služeb. V této dimenzi je využíván Service Value Chain, který dává organizaci návod, jak řídit služby efektivně. Jedná se o sérii kroků, které organizace používá pro dodávání služby zákazníkovi.

V této dimenzi by organizace měla prozkoumat všechny svoje aktivity a identifikovat hodnotové proudy. Díky tomu mohou lépe odhalit překážky v pracovním procesu nebo nadbytečné aktivity, které k vytváření hodnoty nepřispívají. Tyto aktivity by měly být eliminovány a tím se zvýší efektivita.

Optimalizace proudu hodnot může být například automatizace a zavedení nových technologií. Proudů musí být revidovány s ohledem na měnící se okolnosti, upravovány a neustále vylepšovány podle aktuálních potřeb organizace.

Další částí této dimenze jsou procesy. Proces je obecně řada kroků, které přetvářejí vstupy na výstupy. Procesy mohou být modelovány na různé úrovni detailu.

2.1.7.6 Externí faktory

Mimo 4 základní dimenze stojí externí faktory, které organizace ovlivňují zvenku a také působí na dodávání služeb. Metoda, kterou používá ITIL pro identifikaci těchto faktorů je PESTEL (také je nazývána PESTLE) analýza. Jedná se o analýzu makrookolí, která se zejména používá ve strategickém řízení a marketingu. Tato analýza rozděluje faktory do 6 kategorií. Jsou to: [19]

- politické faktory – jaké existující a potenciální vlivy na organizaci mohou působit (např. stabilita vlády, podmínky zaměstnání, omezení v podnikání atd.)
- ekonomické faktory – vliv ekonomiky (např. ekonomická recese, růst, daně, kurz měny atd.)
- sociální faktory – sociální změny a kulturní vlivy (např. demografie, životní styl, náboženství atd.)
- technologické faktory – nové technologie, trendy v technologiích
- legislativní faktor y– zákony, normy na státní i světové úrovni
- ekologické/enviromentální faktory – aktuální situace v ekologii, ochrana prostředí

2.1.7.7 Systém hodnoty

ITIL 4 představuje také nový pojem Service Value System (Systém hodnoty služby). Popisuje, jak aktivity a komponenty organizace společně tvoří systém, který tvoří hodnotu. Service Value System je flexibilní a netvoří jasně danou strukturu, kterou by se organizace musela nutně řídit. Vytváří také prostor pro různé způsoby práce a rámce, například agilní vývoj, DevOps nebo Lean.

Klíčovým vstupem tohoto systému je příležitost a poptávka. Příležitostí se rozumí možnost přidat hodnotu pro stakeholdery. Poptávka je potřeba produktu nebo služby.

Service Value System obsahuje tyto komponenty:

- Guiding principles (Vedoucí principy)
- Governance (Řízení na vyšší úrovni)
- Service Value chain (Řetězec hodnoty služby)
- Continual improvement (Soustavné zlepšování)
- Practices (Praktiky)

2.1.8 Guiding Principles

První komponentou SVS jsou Guiding Principles. Jedná se o doporučení, která vedou organizaci za všech okolností. Jsou to univerzální doporučení vhodná pro všechny organizace a představují hlavní myšlenku ITIL 4. Dají se použít v mnoha dalších rámcích. Je to těchto 9 principů:

- Focus on value – Zaměř se na hodnotu – V podstatě princip celého ITIL 4. Celá organizace musí primárně cílit na vytváření hodnoty pro stakeholdery. Stejně tak ICT služby vytváří hodnotu.
- Start where you are – Začni kde jsi – Není nutné celou organizaci přebudovat, je možné využít spoustu již existujících služeb, procesů, projektů atd. a případně je vylepšit tak, aby došlo k dosažení cílů a zefektivnění.
- Progress iteratively with feedback – Postupuj po iteracích se zpětnou vazbou – Nemělo se by dělat všechno najednou. Proces implementace a neustálého zlepšování je nutné rozdělit do menších sekcí. Je dobré po každé fázi sbírat feedback od všech stakeholderů, aby se zjistilo, jak byla daná fáze implementace úspěšná.
- Collaborate and promote visibility – Spolupracuj a podporuj transparentnost – Spolupráce jednotlivých stakeholderů je důležitá. Je zásadní sdílet důležité informace.
- Think and work holistically – Mysli a pracuj celistvě – Jednotlivé elementy se nedají oddělit, ale pracují jako systém. Na systém je třeba nahlížet jako celek.

- Keep it simple and practical – Zachovej jednoduchost a praktičnost – V procedurách používat nejmenší možný počet kroků, které dovedou k cíli. Eliminovat aktivity, které nevytvářejí hodnotu.
- Optimize and automate – Optimalizuj a automatizuj – Všechny zdroje by měly být využívány co nejefektivněji. Kde je to možné, je vhodné používat technologie pro automatizaci a automatizovat manuální a rutinní činnosti.

2.1.9 Governance

Každá organizace je řízena na nejvyšší úrovni člověkem nebo spíše skupinou lidí, kteří jsou zodpovědní za její výkonnost. [18 s. 80] Může se jednat o nejvyšší vedení. Governance definuje role, odpovědnosti, pravomoci a stanovuje cíle na strategické úrovni. Také se soustředí na vytváření hodnoty. Nejedná se od management, governance spíše management kontroluje a hodnotí jeho práce.

Na governance se vztahují vedoucí principy, které byly představeny v minulé podkapitole. Governance část společnosti také dohlíží na Service Value System. Governance a management spolu sdílí principy, na kterých organizace stojí. Jak Governance, tak management jsou neustále zlepšovány.

2.1.10 Service Value Chain

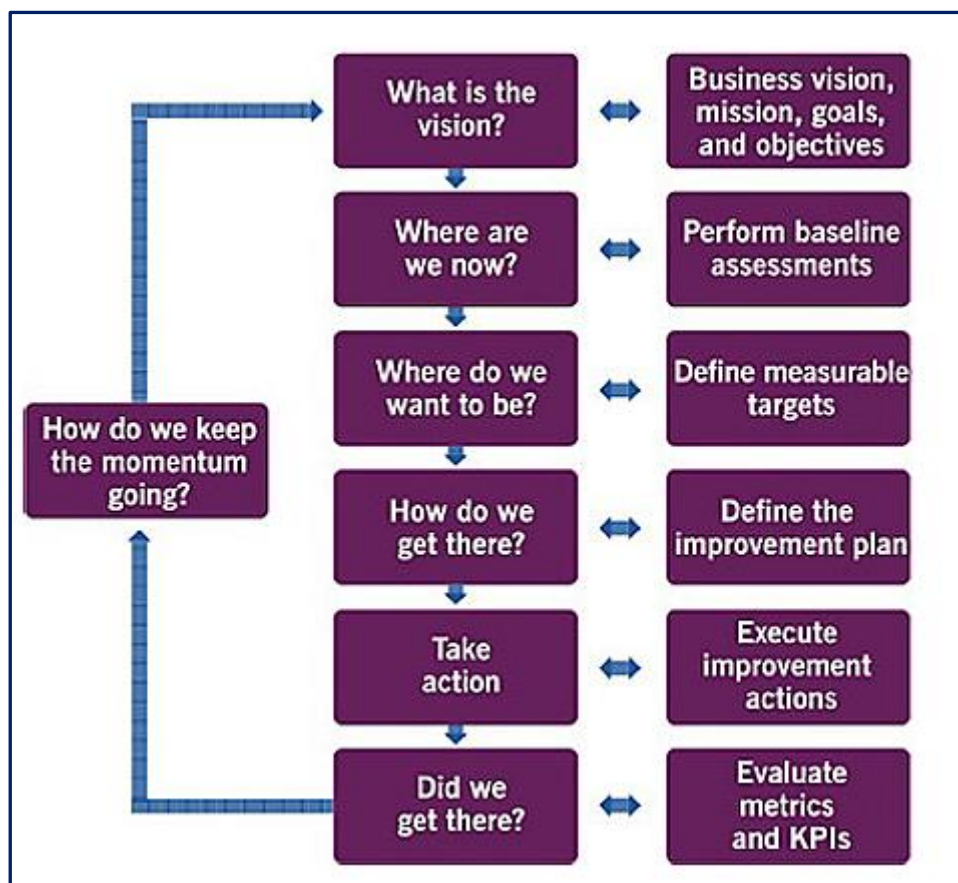
Hlavní částí Service Value Systém je Service Value Chain (Řetězec hodnoty služby). Jedná se o model, který definuje 6 klíčových aktivit, kterými se mění vstupy ve výstupy a cílem je vytvořit hodnotu.

- Plan – Plánuj – Účelem je, aby všichni rozuměli jednotné vizi, současnému stavu a směru vylepšování. Jedná se o celkový pohled z hlediska všech 4 dimenzí.
- Improve – Zlepšuj – Služby, produkty a praktiky se musí neustále vylepšovat, aby se dosáhlo co nejefektivnější tvorby hodnoty.
- Engage – Zapojení – Je zásadní dobře pochopit potřeby stakeholderů, zapojit je do tvorby hodnoty a udržovat dobré vztahy.
- Design and transition – Design a změny – Je nutné neustále zajišťovat, aby služby splňovaly požadavky stakeholderů na kvalitu, náklady a čas.
- Obtain/build – Dosažení/budování – Je kritické zabezpečovat, že služby jsou dostupné, když je potřeba, a splňují požadavky.
- Deliver and support – Dodání a podpora – Služby musí být dodávány a podporovány dle dohodnutých podmínek a očekávání stakeholderů.

2.1.11 Continual Improvement (Neustálé zlepšování)

Neustálé zlepšování se musí odehrávat na všech úrovních organizace – od nejvyšší (governance, strategie, vize, cíle) přes management, procesy až k jednotlivým praktikám. Díky neustálému zlepšování mohou být služby stále více efektivní.

Základní model neustálého zlepšování podle ITIL 4 je zobrazen na Obrázek 2-3.



Obrázek 2-3: Model neustálého zlepšování ITIL 4

Nejprve se určí vize, poté se řeší aktuální stav a čeho chceme dosáhnout, dále jak toho chceme dosáhnout. Následně se provede naplánovaná akce a nakonec se hodnotí, jak bylo zlepšení úspěšné. Tím ale proces nekončí, nicméně se neustále opakuje. Tento model se výrazně podobá známému PDCA cyklu (Plan – Do – Check – Act), který stojí na podobném principu neustálého zlepšování a opakování.

2.1.12 Practices (Praktiky)

ITIL 4 definuje praktiky jako „soubor zdrojů organizace navržených pro provádění práce nebo dosažení cíle“ [18 s. 105] Dělí praktiky do 3. kategorií:

- General Management Practices – Obecné praktiky řízení
- Service management practices – Praktiky řízení služeb
- Technical management practices – Praktiky technického řízení

V následující části jsou krátce popsány všechny praktiky. Praktika Service desk je více popsána v samostatné podkapitole, protože se nejvíce dotýká praktické části této práce.

2.1.13 General management practices

Obecných praktik řízení je v ITIL 4 celkem 14. Jedná se o tyto:

Architecture management

Cílem praktiky je zajistit porozumění různým elementům, které se spojují do organizace.

Continual Improvement

Cílem této praktiky je zajistit neustálé zlepšování všech praktik a služeb v rámci celé organizace. Základem této praktiky je model Continual improvement, tento model se dá použít na jakoukoli část

organizace (procesy, praktiky, cíle atd.). Všichni zaměstnanci organizace by měli být seznámeni a sami vykonávat neustálé zlepšování.

Information Security Management

Účelem této praktiky je zabezpečit informace, které jsou pro organizaci potřebné. Je nutné identifikovat a řídit rizika, která informacím hrozí – např. narušení jejich důvěrnosti, porušení integrity nebo přístupnosti. Praktika se skládá zejména z 3 aktivit – prevence, detekce a náprava. Při prevenci se organizace snaží zabránit bezpečnostním incidentům různými opatřeními (zabezpečení sítě, heslová politika atd.). Detekcí odhaluje případné incidenty a poté zajišťuje nápravu a podstupuje kroky, aby se podobné incidenty již neopakovaly.

Knowledge management

Zde je účelem, aby se zajistilo efektivní a vhodné použití dostupných informací. Znalosti a informace jsou velmi důležité aktivum, kterým organizace disponuje.

Measurement a reporting

Cílem této aktivity je podporovat správné rozhodování a neustálé zlepšování tím, že se sníží nejistota měření. K tomuto účelu se sbírají nejrůznější data, která jsou vyhodnocována na základě například CSF (Kritické faktory úspěchu) a KPI (Klíčové ukazatele výkonu). Reporty poté zobrazují výsledky a podporují rozhodování managementu. Měly by obsahovat informace, které jsou relevantní vzhledem k příjemci reportů.

Organizational Change Management

Zde je cílem zajistit, aby změny, které se v organizaci nevyhnutelně dějí, byly plynule a efektivně implementovány do organizace.

Portfolio Management

Cílem je zajistit, že organizace má správně nastavené soubory služeb, projektů, programů atd. aby dosáhla svých strategických cílů.

Project management

Účelem této praktiky je zajistit úspěšnost projektů správným plánováním, delegováním a monitorin- gem. ITIL zmiňuje dva přístupy k řízení projektů vývoje – vodopádový model (kde jsou požadavky dopředu definovány a neumožňuje se příliš změn v průběhu projektu) a agilní model (kdy vývoj probíhá iterativně a změny jsou na denním pořádku). Oba tyto modely jsou v souladu s ITIL.

Relationship management

Tato praktika slouží k nastavení vztahu mezi organizací a jejími stakeholdery. Zahrnuje pochopení potřeb stakeholderů, uspokojení stakeholderů, řízení stížností a konfliktů. Je důležité udržovat dobré vztahy s interními i externími stakeholdery.

Risk management

Praktika má za úkol správně identifikovat, analyzovat a vypořádat se s riziky. Rizika se v organizaci ohodnocují podle pravděpodobnosti a dopadu rizika. Podle těchto hodnot se pak rizika mohou prioritizovat a je určeno, jak se s nimi naloží.

Service financial management

Tato praktika se týká finančních zdrojů a investic a zajišťuje, že jsou v rámci řízení služeb využívány efektivně.

Strategy management

Záměrem této praktiky je definovat strategické cíle a alokovat zdroje, aby jich bylo dosaženo. Strategický management se má primárně zaměřovat na vytváření hodnoty pro stakeholdery.

Supplier Management

Zde je důležité, aby dodavatelé organizace dostatečně podporovali kvalitu služby. To znamená udržovat s nimi dobré a bližší vztahy a více s nimi spolupracovat.

Workforce and Talent Management

Praktika se zaměřuje na správný výběr lidí do organizace, kteří mají potřebné schopnosti a dovednosti, aby co nejvíce napomáhali k vytváření hodnoty a dosahování cílů. Součástí je také rozvoj zaměstnanců prostřednictvím vzdělávání.

2.1.14 Service management practices

Availability management

Cílem této praktiky je zajistit, že služby jsou dostupné dle dohodnuté dostupnosti a splňují potřeby zákazníků a uživatelů. Dostupnost služby závisí na tom, jak často služba selhává (MTBF) a jaký je čas obnovy služby (MTRS). Dostupnost se často uvádí v procentech a bývá součástí SLA.

Praktika zahrnuje dohodu o cílech dostupnosti, návrh infrastruktury, která zvládne požadovanou dostupnost (např. pokud máme dohodnutou dostupnost 99 %, budeme například potřebovat záložní server), zajištění monitoringu a reportingu dostupnosti a neustálé zlepšování směrem k vyšší dostupnosti služby.

Business analysis

Jedná se o analýzu businessu nebo nějaké jeho části a nalezení řešení, jakým způsobem vyřešit nějaký business problém.

Capacity and performance management

Tato praktika má za úkol zajistit, že služby splňují požadavky na výkon a že uspokojují potřeby zákazníků a uživatelů. Zároveň jsou efektivní z hlediska nákladů.

Change Control

Tato praktika zajišťuje, že změny, které musí být ve službách provedeny, budou správně řízeny a tím bude zajištěna jejich úspěšnost. ITIL rozlišuje tři druhy změn: standardní, normální a mimořádné. Standardní změny jsou změny, které jsou zdokumentované a mají nízké riziko. Normální jsou změny, které mají vyšší riziko, musí být správně naplánovány a schváleny. Také musí probíhat řádné testování a musí být vytvořena dokumentace. Mimořádné reagují na nestandardní události a musí být provedeny co nejdříve. Někdy je tak odsunuta tvorba dokumentace a testování až na dobu po provedení změny.

Incident Management

Cílem této praktiky je minimalizovat negativní následky incidentů. To je provedeno tím, že provoz služby je co nejdříve vrácen do normálního stavu.

Incident je definován jako neplánované přerušení služby nebo snížení její kvality. Incident Management má velký vliv na spokojenost uživatelů a zákazníků. Každý incident musí být zaznamenán a vyřešen v termínu, který odpovídá dohodě mezi poskytovatelem a příjemcem služby.

IT asset management

Podstatou této praktiky je řídit celý životní cyklus IT aktiv, aby se pomohlo organizaci zvýšit hodnotu, hlídat náklady, řídit rizika a podporovat rozhodování o nákupech aktiv. Aktiva obvykle obsahují všechno software, hardware, síťové prvky a cloudové služby.

Informace o IT aktivech se dají zjistit při auditu, ale lepší je udržovat seznam aktiv aktuální při každé změně aktiva (např. nákup HW, vyřazení HW, nákup nové licence SW nebo přeřazení licence). Hardware aktiva musí být označeny pro jednoznačnou identifikaci a musí být zaznamenáno, kde se nacházejí, aby se zamezilo krádeži nebo zneužití. Software aktiva musí být chráněny proti nedovolenému kopírování.

Monitoring and event management

Tato praktika slouží k systematickému sledování služeb a jejich komponent a zaznamenává události. Událost (event) je definována jako jakákoli změna stavu, která má význam pro řízení služby. Rozdělují se na informační, varovné a výjimečné události. Události jsou různé důležitosti. Některé události mohou být klasifikovány jako incidenty. Monitoring by měl probíhat co nejvíce automaticky, ale lidský zásah je také nezbytný.

Problem Management

Účelem této praktiky je snížit pravděpodobnost a dopad incidentů identifikováním aktuálních a potenciálních příčin incidentů. Problém je příčina nebo potenciální příčina jednoho nebo více incidentů. Tato praktika také definuje pojem známá chyba (Known error), což je problém, který byl zjištěn, ale ještě nebyl vyřešen.

Aktivity, které vedou k odhalení problému jsou například analýzy incidentů (např. podobné nebo duplicitní incidenty mohou mít stejnou příčinu). Problémy jsou prioritizovány zejména dle závažnosti rizika, které představují. Pokud problém nelze vyřešit rychle, vytváří se tzv. workaround, což je řešení, které umožňuje snížit dopad problému nebo incidentu.

Release management

Tato praktika má za úkol učinit nové služby nebo nové funkcionality přístupné uživatelům.

Service catalogue management

Účelem této praktiky je zajistit jednotné místo pro shromažďování informací o poskytovaných službách.

Service configuration management

Záměrem této praktiky je zajistit, že budou k dispozici informace o aktuálních konfiguracích služeb.

Service continuity management

Jedná se o praktiku, která zajišťuje, že bude zajištěna dostupnost služeb v případě nějaké katastrofy.

Service design

Tato praktika má za úkol navrhnout služby tak, aby splňovaly svůj účel a požadavky.

Service Desk

Účelem této praktiky je zachycovat poptávku po řešení incidentů a požadavků. Tato praktika je více popsána v podkapitole 4.5, protože na ní závisí praktická část této práce.

Service level management

Účelem této praktiky je nastavit jasné cíle pro úroveň služeb a zajištění, že dodávka služeb je správně řízena. Úroveň služby je metrika nebo více metrik, které definují požadovanou kvalitu služby.

Service request management

Tato praktika má za úkol podporovat dohodnutou kvalitu služby zpracováním požadavků uživatelů.

Service validation and testing

Praktika má zajistit, že nové služby nebo změny ve službách splňují definované požadavky. Požadavky mohou být definované například zákazníky, business cíli nebo legislativou. Testování je specifickou disciplínou, která zahrnuje jednotkové testy (unit testy), systémové testy a integrační testy.

2.1.15 Technical management practices

Deployment practices

Tato praktika se zabývá přenosem nového nebo změněného hardwaru, softwaru, dokumentace, procesu nebo jakékoli jiné komponenty do produkčního prostředí.

Infrastructure and platform management

Cílem této praktiky je řízení IT infrastruktury a platform, které organizace používá. Zahrnuje monitoring serverů, úložiště, sítě, klientských zařízení, middleware a operačních systémů.

Software development and management

Tato praktika má za úkol zajistit, že aplikace splňuje potřeby uživatelů, tedy že splňuje funkční i nefunkční požadavky.

2.1.16 Poznámky, reference

- YASM. ITIL 4 vs. ITIL V3 | YaSM Wiki. YaSM [online]. 2019 [vid. 2020-05-20]. Dostupné z: https://yasm.com/wiki/en/index.php/ITIL_4_vs_ITIL_V3
- BUCKSTEEG, Martin, 2012. *ITIL 2011: Stručný a srozumitelný výklad*. Brno: Computer Press. ISBN 978-80-251-3732-1.
- STUDENÝ, Aleš, 2013. IT Service Management based on ITIL [online]. 2013. B.m.: ALVAO. [vid. 2016-12-05]. [Dostupné z:](#)
- TSO, 2012. An Introductory Overview of ITIL® 2011 [online]. 2012. B.m.:
- TSO, The Stationery Office. [vid. 2016-12-05]. [Dostupné z:](#) ,TSO,2013a.
- ITIL:: [IT service management practices]. [2]: Service design. 2011 ed., 2. impr. London: TSO, The Stationery Office. AXELOS - global best practice. ISBN 978-0-11-331305-1.,
- TSO, 2013b. ITIL:: [IT service management practices]. [3]: Service transition. 2011 ed., 2. impr. London: TSO, The Stationery Office. AXELOS - global best practice. ISBN 978-0-11-331306-8.
- TSO, 2013c. ITIL:: [IT service management practices]. [4]: Service operation. 2011 edition, 2. impr. London: TSO, The Stationery Office. AXELOS - global best practice. ISBN 978-0-11-331307-5.
- TSO, 2013d. ITIL:: [IT service management practices]. [5]: Continual service improvement. 2011 ed., 2. impr. London: TSO, The Stationery Office. AXELOS - global best practice. ISBN 978-0-11-331308-2.

2.2 CobiT

2.2.1 Celková charakteristika

- **CobiT (Control Objectives for Information and Related Technology)** - vytvářen a distribuován **organizacemi ISACA a IT governance Institute**.
- Obsahuje **popis sady procesů, návodů, cílů, hodnocení, nejlepších praktik a ukazatelů**.
- CobiT 5 obsahuje **rámec (framework) pro řízení a správu** podnikové informatiky.
- Metodika CobiT 5 je **založena na pěti principech**:
 - naplnění potřeb zainteresovaných stran,
 - řízení IT napříč organizací,
 - použití metodiky CobiT 5 jako jediného integrovaného rámce pro řízení informatiky,
 - podpoře celostního přístupu,
 - rozlišení IT governance a IT řídicích činností.

2.2.2 Poznámky, reference

- [Voříšek, J. ; Pour, J. - Management podnikové informatiky - \(Professional Publishing 2012\) - ISBN9788074311024](#), s. 21-22, (kap. Chlapek, D.),
- [Voříšek, J. a kol - Principy a modely řízení podnikové informatiky - \(Praha, Oeconomia 2008\) - ISBN9788024514406](#), kap. 9.2,
- [IT Governance Institute. \(2007\). COBIT 4.1 - \(IT Governance Institute 2007\)](#),
- [IT Governance Institute. \(2008\). Enterprise value: Governance of IT Investments. - \(IT Governance Institute 2008\)](#),
- [IT Governance Institute. \(2009\). The Risk IT Framework. - \(IT Governance Institute 2009\)](#),
- [IT Governance Institute. \(2011\). Mapping of ISO/IEC 20000 With C BIT® 4.1. - \(IT Governance Institute 2011\)](#),

- [IT Governance Institute. \(2012\). Cobit 5: Enabling Processes - \(IT Governance Institute 2012\).](#)

2.3 CMMI

2.3.1 Celková charakteristika

- **CMMI (Capability Maturity Model Integration)** je model pro zlepšování procesů v oblasti rozvoje služeb vytvořený v **Software Engineering Institute (SEI)**. Obsahuje tyto **modely**:
 - **CMMI pro akvizice** (CMMI for Acquisition) zaměřená na zlepšení procesů při pořizování produktů a služeb,
 - **CMMI pro vývoj** (CMMI for Development), která představuje návod na zlepšení procesů při vývoji produktů a služeb,
 - **CMMI pro služby** (CMMI for Services) určená pro zlepšení procesů při poskytování služeb.

2.3.2 Poznámky, reference

- [Voříšek, J. ; Pour, J. - Management podnikové informatiky - \(Professional Publishing 2012\) - ISBN9788074311024](#), s. 22-23, (kap. Chlapek, D.),
- [CMMI-DEV - \(Carnegie Mellon University. 2011\).](#)

2.4 IT Balanced Scorecard

2.4.1 Celková charakteristika

- IT Balanced Scorecard je nástroj pro **řízení podnikové informatiky a vztahu mezi informatikou a byznysem**.
- **Modifikuje** čtyři původní perspektivy (finanční, zákaznická, podnikové procesy a učení a růst) na **perspektivy důležité v oblasti podnikové informatiky** - orientace na uživatele, přínos pro podnik, provozní excelence, orientace na budoucnost.
- V každé perspektivě přístup definuje **misí, cíle a jejich metriky**.

2.4.2 Poznámky, reference

- [Voříšek, J. a kol - Principy a modely řízení podnikové informatiky - \(Praha, Oeconomia 2008\) - ISBN9788024514406](#), kap. 9.7.

2.5 Test Maturity Model Integration, TMMI

2.5.1 Celková charakteristika

- Test Maturity Model Integration (TMMi) je jedním **z nejnovějších modelů pro zlepšování procesů testování** a snaží se zlepšování procesů testování standardizovat.

2.5.2 Vývoj a obsah TMMi

- TMMi byl vytvořen mezi lety 2006 a 2008 nezávislou neziskovou organizací **TMMi Foundation**.
- **Cílem** této organizace bylo vyvinout a dále rozvíjet rozsáhlý, robustní a veřejně dostupný model pro **zlepšování procesů testování softwaru**.

- **Struktura TMMi** přímo vychází ze struktury CMMi a v některých částech se na model CMMi odkazuje (jako například u procesní oblasti configuration management, kde praktiky pro pracovní produkty dle CMMi jsou aplikovatelné na pracovní produkty testování).
- **Rozsah TMMi** pokrývá jak **testovací aktivity systémově inženýrského charakteru**, které pokrývají celkově vývoj systémů (což může, ale nemusí zahrnovat software), tak **testovací aktivity zaměřené na softwarově inženýrské disciplíny**, které pokrývají přímo vývoj softwarových systémů. (TMMi Foundation, 2012a, s. 7).
- Pro využití modelu TMMi ke zlepšování procesů testování a pro **hodnocení zralosti procesů** testování je třeba se dobře orientovat ve struktuře modelu.

2.5.3 Komponenty TMMi

- Komponenty modelu TMMi jsou rozděleny **do následujících třech kategorií**: Povinné (required), očekávané (expected) a informativní (informative):
 - **Povinné komponenty** popisují, čeho musí organizace dosáhnout, aby byla daná procesní oblast uspokojena, a bez kterých není dané úrovně zralosti dosaženo. Povinnými komponentami jsou v modelu TMMi specifické a generické cíle.
 - **Očekávané komponenty** popisují, co by měla organizace typicky implementovat, aby dosáhla požadovaných komponent. Očekávanými komponentami jsou v modelu TMMi specifické a generické praktiky, viz dále. Organizace se nemusí řídit přesně podle těchto praktik, ale pokud se těmito doporučovanými praktikami neřídí, musí alespoň aplikovat odpovídající alternativu.
 - **Informativní komponenty** jsou pro organizaci inspirací, jak dosáhnout povinných komponent a jak aplikovat očekávané komponenty. Informativní komponentami jsou v modelu TMMi sub-praktiky, vzorové pracovní produkty, poznámky, příklady a doporučení.
- **Vztahy mezi komponentami** modelu TMMi zachycuje obrázek (viz obr.1).

2.5.4 Poznámky, reference

- [Králová, I. - Metodika testování podle mezinárodních praktik a standardů - \(VŠE 2013\).](#)

2.6 *Multidimensional Management and Development of Information System, MMDIS*

Voříšek, J. (KIT, VŠE)

2.6.1 Celková charakteristika

- **Cílem MMDIS** je vývoj, údržba a provoz komplexního a integrovaného informačního systému podniku, který využívá potenciálu dostupných informačních technologií a informatických služeb k podpoře podnikových cílů.
- **Komplexní IS** je takový, který podporuje všechny činnosti podniku, pro něž je možné nalézt efektivní informatickou podporu.
- **Integrovaný IS** znamená, že informační systém je tvořen z celé řady hardwarových, softwarových a datových komponent, které jsou navzájem propojeny (integrovány) do jednoho systému. To, že IS optimálně využívá potenciálu dostupných ICT, znamená, že není nutné postaven na nejnovějších technologiích a nejsofistikovanějších ICT službách, ale vybírá z nich ty, které mají pro daný podnikový IS ekonomický smysl.
- **Hlavním kritériem efektivity** podnikové informatiky je to, do jaké míry se jí daří podporovat podnikové cíle a priority.

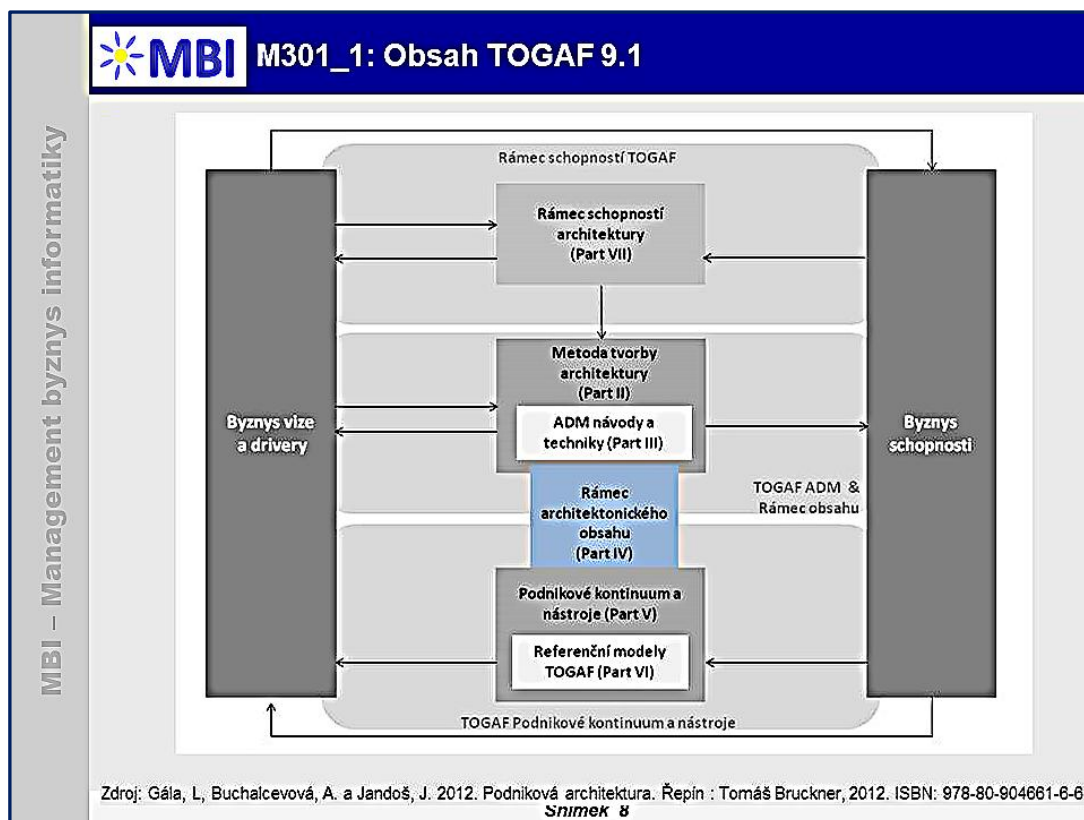
2.6.2 Poznámky, reference

- [Voříšek, J. a kol – Principy a modely řízení podnikové informatiky – \(Praha, Oeconomia 2008\) – ISBN9788024514406](#), kap. 6.1 a 6.2

2.7 TOGAF

2.7.1 Celková charakteristika

- TOGAF je **nejpoužívanějším rámcem podnikové architektury**.
- **Tvůrcem** TOGAFu je **Open Group Architecture Forum**, které v roce 2011 vydalo zatím verzi TOGAF 9.1.
- Podniková architektura **se zde skládá ze čtyř dílčích architektur**:
 - **Byznys architektura** (Business Architecture), která zahrnuje byznys strategii, governance, organizaci a klíčové byznys procesy,
 - **Datová architektura** (Data Architecture), která představuje logickou a fyzickou strukturu dat a řízení datových zdrojů,
 - **Aplikační architektura** (Application Architecture), která zahrnuje aplikace, jejich interakce a vazby na klíčové byznys procesy,
 - **Technologická architektura** (Technology Architecture), kterou tvoří softwarové a hardwarové schopnosti nutné pro nasazení byznys služeb, datových a aplikačních služeb.



Obrázek 2-4: Obsah TOGAF 9.1

2.7.2 Poznámky, reference

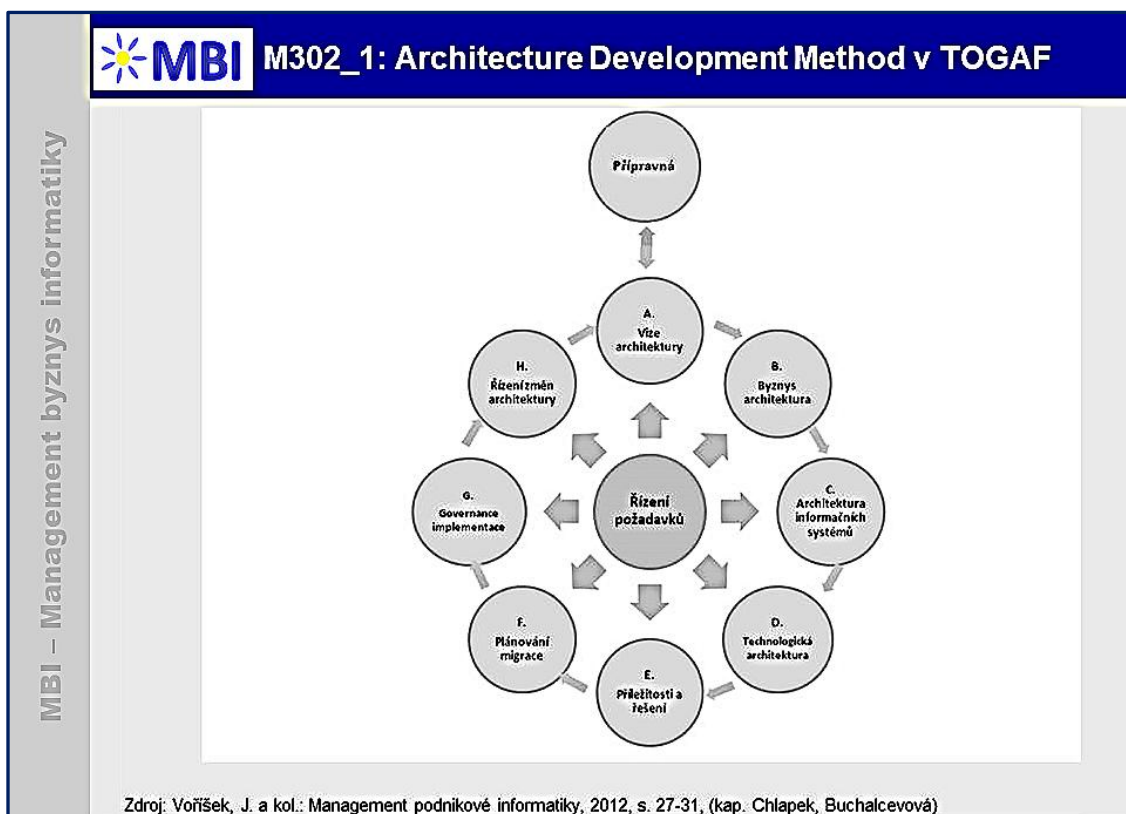
- [Voříšek, J. ; Pour, J. - Management podnikové informatiky - \(Professional Publishing 2012\) - ISBN9788074311024](#), s. 27-31, (kap. Chlapek, Buchalceková),

- [Gála, L. ; Buchalceková A. ; Jandoš J. - Podniková architektura - \(Bruckner, T. 2012\) - ISBN9788090466166,](#)
- [TOGAF Version 9.1. - \(Open Group 2011\),](#) pro MBI zpracovala: Buchalceková, A (katedra IT, VŠE).

2.8 *Architecture Development Method, ADM*

2.8.1 Celková charakteristika

- ADM popisuje, **jak vybudovat pro daný podnik specifickou podnikovou architekturu**, která adresuje byznys požadavky.
- ADM poskytuje **návody v několika rovinách**:
 - definuje několik fází vývoje architektury (Business Architecture, Information Systems Architectures, Technology Architecture) v cyklu jako procesní šablonu,
 - poskytuje popis jednotlivých fází – cíl, přístup, vstupy, kroky, výstupy,
 - podporuje řízení požadavků, které jde přes jednotlivé fáze.
- **Přípravná fáze** zahrnuje přípravné činnosti EA procesu:
 - Fáze A zahrnuje definici rozsahu, identifikaci zainteresovaných, vytvoření vize a schválení.
 - Fáze B je zaměřena na vytvoření byznys architektury.
 - Fáze C je zaměřena na vytvoření architektury IS včetně datové a aplikační architektury.
 - V rámci fáze D se vytváří technologická architektura.
 - Ve fázi E jde o identifikaci řešení a nástrojů, které povedou k dosažení cílové architektury.
 - Ve fázi F se definuje transformace.
 - Fáze G zajišťuje dohled nad implementací.
 - Ve fázi H se zavádějí procedury pro řízení změn.



Obrázek 2-5: ADM v TOGAF

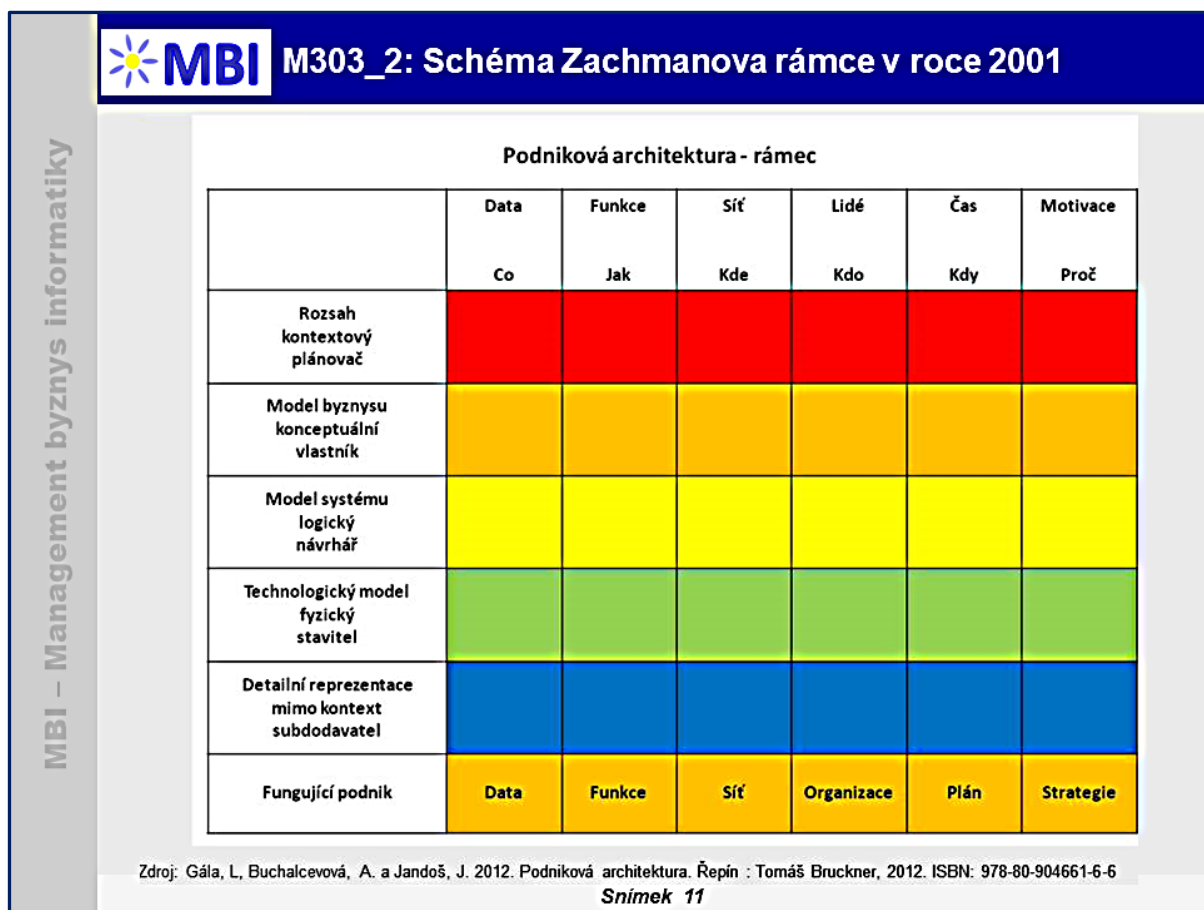
2.8.2 Poznámky, reference

- [Voříšek, J. ; Pour, J. - Management podnikové informatiky - \(Professional Publishing 2012\) - ISBN9788074311024](#), s. 27-31, (kap. Chlapek, Buchalceová),
- [Gála, L. ; Buchalceová A. ; Jandoš J. - Podniková architektura - \(Bruckner, T. 2012\) - ISBN9788090466166](#), pro MBI zpracovala: Buchalceová, A (katedra IT, VŠE).

2.9 Zachmanův rámec

2.9.1 Celková charakteristika

- Zachmanův rámec je **prvním rámcem podnikové architektury**. Je to představitel **klasifikačních rámců**.
- Zachmanův rámec se v průběhu času vyvíjel a akcentoval tak změny v náhledu na podnikovou architekturu i **zkušenosti a chyby spojené se zaváděním podnikové architektury**.
- Rámec pro architekturu informačních systémů (**Framework for Information Systems Architecture**) Zachman poprvé publikoval v roce 1987 v IBM Systems Journal.



Obrázek 2-6: Schéma Zachmanova rámce 2001

- Zachman použil **barvy pro zdůraznění myšlenky**, že každá řádka je jiná a je výsledkem transformace, nikoli dekompozice. Jsou použity barvy červená, oranžová, žlutá, zelená, modrá a oranžová. Oranžová barva je použita jak v řádce 2, tak v řádce 6, což má vyjádřit skutečnost, že to, co vlastníci (Owners) potřebují (řádka 2), by mělo být právě to, co se skutečně v podniku realizuje (řádka 6). Rámec se snaží zdůraznit abstrakci. Řádky 1 až 5 jsou abstrakcí, instance, tedy skutečné výskyty jsou až v řádce 6.
- V roce 2004 byla vydána další významná verze Zachmanova rámce, která byla z důvodu odlišení od předchozích verzí nazvána **The Zachman Framework2™**. Nejvýraznější změnou byl **přechod od IT terminologie k byznys terminologii**, který vedl ke zvýšení zájmu byznysu o rámec. Stále zde ale přetrvával jeden podstatný problém, a to že grafická reprezentace vůbec nezdůrazňovala myšlenku integrace v řádcích a transformace ve sloupcích matice. Postupujeme-li shora dolů ve sloupcích, jde o transformace nikoli zpřesňování. Stejně tak nedostatečné zdůraznění integrace v řádcích vedlo k tomu, že architekti vytvářeli složené modely místo toho, aby oddělili jednotlivé nezávislé pohledy a potom je integrovali.
- V roce 2011 byla vydána nová verze **Zachmanova rámce s označením 3.0**. Tato verze obsahuje **změny jak v grafické reprezentaci, tak zejména v obsahu a metamodelu**. Podstatné je zejména **označení rámce za ontologii**, což vyjadřuje podnadpis “The Enterprise Ontology.”



Obrázek 2-7: Zachmanovo schéma podnikové architektury

- Zachmanův rámec verze 3.0 představuje klasifikační teorii o vědecké podstatě podniku a typech entit, které v podniku existují.
- Je podnikovou ontologií. Je to schéma vyjadřující vzájemné působení dvou klasifikací, které existují tisíce let. První klasifikace je vyjádřena tázacími zájmeny Co (What), Jak (How), Kdy (When), Kdo (Who), Kde (Where) a Proč (Why). Integrace odpovědí na tyto otázky umožňuje úplné popsat komplexní systém. Druhá klasifikace představuje proces materializace, tedy transformace abstraktní myšlenky do reálné instance. (Zachman, 2008).

2.9.2 Poznámky, reference

- [Zachman, J. - The Zachman Framework Evolution - \(Zachman International, Inc. 2011\),](#)
- [Gála, L. ; Buchalceková A. ; Jandoš J. - Podniková architektura - \(Bruckner, T. 2012\) - ISBN9788090466166, pro MBI zpracovala: Buchalceková, A. \(katedra IT, VŠE\).](#)

3. Metodiky a metody řešení IT



Účelem kapitoly je:

- vymezit podstatu a základní charakteristiku **pouze vybraných metod a metodik**, které se váží k řešení IT,
- pro **detailnější charakteristiky** metodik a metod jsou uvedeny **reference** na některé další publikace.

Mapa metodik a metod řešení IT firmy (s odkazy)

[3.1] PMBOK		[3.2] PRINCE2	
[3.3] SCRUM			
[3.4] Azure DevOps	[3.5] RUP	[3.6] UML	[3.7] CRISP-DM
[3.8] Datové modelování	[3.9] Procesní modelování	[3.10] Dimenzionální modelování	

Další podkapitoly obsahují pouze rámcové **vymezení obsahu** vybraných **metodik a metod v řešení IT** firmy, a to:

- **Project Management Body of Knowledge, PMBOK** je standard projektového řízení,
- **Projects in Controlled Environments, PRINCE2** používá procesní přístup k řízení projektů,
- **Scrum** představuje rámec (angl. framework) pro vývoj a řízení komplexních produktů,
- **Azure DevOps** představuje oddělené postavení softwarového vývoje od softwarových operací.
- **Rational Unified Process (RUP)** sjednocuje různé metody a jazyky procesního inženýrství pomocí jednotného metamodelu,

- **Unified Modeling Language (UML)**, podstatou je jeho nezávislost na procesu vývoje, protože jazyk UML není svázán s žádnou konkrétní vývojovou metodikou
- **Metodika CRISP–DM** je metodika pokrývající kompletní proces data miningových úloh,
- **Datové modelování** je metoda, která slouží k navrhování struktury dat v databázích a k jejich dokumentaci,
- **Procesní modelování** je řešení změn, rozvoje a zavádění nových podnikových procesů ve vazbě na stanovenou podnikovou architekturu,
- **Dimenzionální modelování** je vymezení všech dimenzí, jejich obsahu, včetně vnitřní hierarchie prvků, a dílčích charakteristik, určení soustavy sledovaných ukazatelů (faktů) a jejich dílčích charakteristik a specifikace vazeb mezi ukazateli a odpovídajícími dimenzemi.

3.1 Project Management Body of Knowledge, PMBOK

3.1.1 Celková charakteristika

- PMBOK, celým názvem Project Management Body of Knowledge je **vytvářen americkou organizací PMI (Project Management Institute)**.
- PMBOK je **standard projektového řízení**, který je sepsaný v publikaci „**A guide to Project Management Body of Knowledge (PMBOK®)**“.

3.1.2 Obsah PMBOK

- PMBOK definuje **47 procesů** uskupených do pěti procesních skupin. Dále definuje **10 znalostních oblastí**. Každý proces má **definované vstupy, nástroje a techniky**, které mohou být v procesu použity a **výstupy**.
- Procesy jsou v projektech **používány opakovaně a iterativně**. Záleží na konkrétním projektu a **rozhodnutí projektového manažera** a jeho týmu, jaké procesy budou na konkrétní projekt aplikovány.
- Projekt je **rozdělený na jednotlivé fáze**, které slučují dohromady aktivity, jež vytvářejí jeden nebo více produktů nebo služeb.

3.1.3 Procesní skupiny

- **Zahájení** – procesy definující nový projekt nebo fázi.
- **Plánování** – procesy potřebné ke stanovení rozsahu projektu, upřesnění cílů a definování činností, které je potřeba provést, pro dosažení těchto stanovených cílů projektu.
- **Provádění** – procesy, které mají za cíl provádět práci stanovenou v projektovém plánu.
- **Monitorování a kontrola** – procesy, pro sledování, hodnocení, posuzování pokroku a výkonu projektu. V těchto procesech jsou identifikovány oblasti, ve kterých je požadována změna oproti plánu.
- **Zavření** – procesy, které je potřeba provést aby byly dokončeny veškeré aktivity potřebné k ukončení fáze nebo projektu.

3.1.4 Znalostní oblasti

- **Procesy jsou seskupeny** nejen do procesních skupin, ale také **do znalostních oblastí** (Knowledge areas).
- „**Znalostní oblast představuje komplexní sadu pojmů, termínů a činností, které tvoří profesionální oblast, oblast projektového řízení nebo specializaci**“ [PMBOK] Znalostní oblasti obsahují detailní popisy jednotlivých procesů, jejich vstupů, výstupů, nástrojů a technik. PMBOK **definuje 10 znalostních oblastí**:

- Management integrace (Project Integration Management),
- Management rozsahu (Project Scope Management),
- Management času (Project Time Management),
- Management nákladů (Project Cost Management),
- Management kvality (Project Quality Management),
- Management lidských zdrojů (Project Human Resources Management),
- Management komunikace (Project Communications Management),
- Management rizik (Project Risk Management),
- Management externích zdrojů (Project Procurement Management),
- Management zainteresovaných stran (Project Stakeholder Management).

3.1.5 Nástroje měření kvality

- Představení **sedmi základních nástrojů pro měření kvality** (Seven basic quality tools), zahrnujících:
 - diagram příčin a následků (též známý jako diagram rybí kosti),
 - vývojový diagram (označovaný také jako procesní mapa),
 - kontrolní listy (checklisty),
 - Paretův diagram,
 - histogram,
 - kontrolní grafy,
 - bodové grafy.

3.1.6 Tvorba plánů v PMBOK

- Každá znalostní oblast obsahuje:
 - na začátku tvorbu plánu, vztahujícího se k dané znalostní oblasti (například plán řízení kvality, plán řízení rizik apod.). Tvorba těchto plánů probíhá ve 4 procesech, které byly přidány do procesní skupiny plánování – plánování řízení rozsahu (Plan Scope Management),
 - plánování řízení časového harmonogramu (Plan Schedule Management),
 - plánování řízení nákladů (Plan Cost Management),
 - plánování řízení zainteresovaných stran (Plan Stakeholder Management).
- PMBOK také **obsahuje celou řadu výpočtů**, například pro výpočet odchylky nákladů, odchylky oproti časovému harmonogramu, výpočty indexů a dalších důležitých ukazatelů.

3.1.7 Poznámky, reference

- [Project Management Institute - A Guide to the Project Management Body of Knowledge \(PMBOK Guide\), Fifth Edition - \(Project Management Institute 2013\)](#),
- [Diplomová práce] [Matoušková, S. - Řízení IT projektů v praxi a obecně uznávané metodiky - \(Praha. VŠE 2014\)](#),
- [Voříšek, J. ; Pour, J. - Management podnikové informatiky - \(Professional Publishing 2012\) - ISBN9788074311024](#), s. 26, (kap. Chlapek, Buchalceová)

3.2 Projects in Controlled Environments, PRINCE2

3.2.1 Celková charakteristika

- Metodiku PRINCE2 (Projects IN Controlled Environments) vlastní **organizace UK Cabinet Office**. Udržuje a spravuje ji **APM Group Ltd**. Prošla několika revizemi.
- Usnadňuje **přizpůsobení metodiky** pro potřeby různých velikostí a věcných zaměření projektů.
- PRINCE2 **používá procesní přístup** k řízení projektů.
- Metodika je popsána v publikaci **“Managing Successful Projects with PRINCE2”** vydanou v roce 2009.
- Zahrnuje tyto **hlavní součásti**:*
 - **Principy** – představují základ pro efektivní využití metodiky v rámci projektů různého charakteru,
 - **Procesy** – jsou soubory činností podstatných pro efektivní směřování a řízení projektu,
 - **Témata** – popisují hlavní aspekty řízení projektů, pro které je charakteristické, že se musí řešit průběžně,
 - **Přizpůsobení** – nabízí možnosti modifikace uplatnění metodiky pro rozsahu a složitosti projektu, případně dle dalších faktorů (úroveň managementu, ekonomická situace apod.).

3.2.2 Principy

- **Kontinuální obchodní zdůvodnění** -
 - efekty projektu musí přesahovat náklady,
 - efekty mají být měřitelné, ale nemusí být finanční,
 - musí být definovány očekávané efekty a průběžně se musí vyhodnocovat jejich realizace,
- **Učení se ze zkušeností** – získané zkušenosti se musí zaznamenat a předávat následujícím projektovým týmům,
- **Definované role a odpovědnosti** – je nutné jasně definovat role v projektu, jejich povinnosti a odpovědnosti a současně i principy (cesty) pro vzájemnou komunikaci a reporting,
- **Řízení pomocí etap** –
 - termín „plánovací horizont“, tj. na jak dlouhé období se má plánovat v určité úrovni detailu,
 - projektový plán – hrubý, na celý projekt,
 - „etapy“ (resp. „manažerské etapy“) – jsou plánovány na podstatně vyšší úrovni detailu, obsahují „rozhodovací body“ pro posouzení dalšího pokračování projektu,
 - v jednom čase může Projektový manažer spustit pouze jednu etapu,
- **Řízení na základě výjimek** -
 - Projektový výbor může Projektovému manažerovi stanovit limity, resp. „tolerance“ vztahované k 6 základním parametrům výkonnosti projektu (čas, náklady, kvalita, rozsah, rizika, přínosy),
 - při řízení projektu může dojít k odchylkám v hodnotách parametrů – pokud jsou v rámci tolerance, není třeba revidovat stav projektu, pokud jsou mimo ni, pak musí výjimku řešit Programový manažer s Programovým výborem,

- **Zaměření se na produkty** – ty je třeba přesně identifikovat, musí odpovídat definovanému účelu vzhledem k požadavkům uživatele na jejich kvalitu,
- **Přizpůsobení projektovému prostředí** – vhodná míra přizpůsobení metodiky velikosti, komplikovanosti a významu projektu.

3.2.3 Parametry výkonnosti

- PRINCE2 rozlišuje **6 základních, převážně měřitelných parametrů výkonnosti** projektu, a to náklady (I502), čas (I503), rozsah (I504), kvalita (I531), riziko, přínosy (I506).

3.2.4 Procesy

- Každý proces je **součástí životního cyklu projektu** a obsahuje **aktivitu**, které jsou v rámci daného procesu prováděny, produkty, které jsou vytvářeny, upravovány nebo schvalovány a odpovědné osoby za jednotlivé produkty. PRINCE2 **definuje 7 procesů**.

3.2.5 Zahájení projektu (SU, Starting up a Project)

- Vstup – „Mandát projektu“ – obvykle definovaný v rámci programu více projektů (D124A),
- Je součástí předprojektové přípravy s cílem minimalizovat řešení chybně zadaných projektů,
- Hlavním výstupem je „Charta projektu“ (definice projektu, role, Řídící tým projekt, zadání pro Obchodní případ, popis Produktu, určení projektového přístupu) – bude podkladem pro Projektový výbor (D402A),
- Další výstupy – Deník projektového manažera, Přehled získaných poznatků, Plán etapy nastavení,
- MBI: (U401A).

3.2.6 Směřování projektu (DP, Directing a Project)

- Vstup – požadavek Projektového manažera na Projektový výbor,
- Projektový výbor rozhoduje o obchodním zdůvodnění projektu, zajišťuje komunikaci s korporátním managementem,
- Směřování projektu začíná po Zahájení projektu a je současně rozhoduje na konec o ukončení projektu,
- Výstupem jsou Schválení Dokumentace nastavení projektu, Schválení etapy, Požadavek na Plán realizace výjimky, Schválení výjimky, Schválení ukončení projektu,
- MBI: (U403A).

3.2.7 Nastavení projektu (IP, Initiating a Project)

- Vstupem je Schválení nastavení Projektovým výborem,
- Vytváří Dokumentaci nastavení projektu – hlavní výstup, základní řídicí dokument, obsahuje - aktualizaci Charty projektu, Strategii řízení rizik, Strategii řízení kvality, Strategii řízení komunikace, Strategii řízení konfigurace, Plán projektu, Projektové kontroly, Způsob přizpůsobení metodiky,
- MBI: (U402A).

3.2.8 Řízení etapy (CS, Controlling a Stage)

- Vstupem je schválený Plán etapy nebo Plán realizace výjimky,

- Výstupem jsou Balík práce, Zpráva o stavu etapy, Zpráva o otevřeném bodu, Zpráva o výjimce,
- MBI: (U404A).

3.2.9 Řízení dodávky produktu (MP, Managing Product Delivery)

- Vstupem je schválený Balík práce Projektovým manažerem a předán Týmovému manažerovi,
- Výstupem je Týmový plán, Zpráva o stavu Balíku práce, Specializovaný produkt, produkty,
- MBI: (U404A).

3.2.10 Řízení přechodu mezi etapami (SB, Managing a Stage Boundary)

- Při ukončení etapy se rozhoduje o pokračování, Projektový manažer předkládá Projektovému výboru podrobný plán na další etapu. Proces se používá i při vytvoření Plánu realizace výjimky,
- Výstupem jsou Plán (další) etapy, Aktualizovaná Dokumentace nastavení projektu (zejména Plán projektu a Obchodní případ a organizační struktura), Aktualizovaný Plán revize přínosů, Zpráva o ukončení etapy, Plán realizace výjimky,
- MBI: (U402A).

3.2.11 Ukončení projektu (CP, Closing a Project), U405A)

- Zajišťuje aktualizaci Plánu projektu, uvolnění zdrojů, akceptace a předání do provozu, vyhodnocení Plánu revize přínosů, vyhodnocení projektu, archivace dokumentace,
- Výstupem je Doporučení následných opatření, Odhady pracnosti, aktualizovaný Plán revize přínosů, Zpráva o ukončení projektu, Zpráva o získaných poznatcích, Oznámení o ukončení projektu,
- MBI: (U405A).

3.2.12 Témata

3.2.12.1 Obchodní případ

- Obchodní případ představuje **dokument zdůvodňující projekt z ekonomických, obchodních a dalších podnikových hledisek**. Projekt je řízen na základě Obchodního případu **ve 2 rovinách** – využívá se jako podklad pro rozhodování Projektového výboru, nebo obráceně - vyhodnocují se dopady problémů a změn v projektu na Obchodní případ.
- Obchodní případ se využívá v celém životním cyklu a jeho rámci se pravidelně aktualizuje. Vlastníkem Obchodního případu je sponzor. **Obsahem je:**
 - Shrnutí (manažerské shrnutí),
 - Důvody pro projekt,
 - Možnosti volby – varianta pro dosažení nejlepšího poměru cena / výkon,
 - Očekávané přínosy (měřitelné / neměřitelné),
 - Očekávané nevýhody (mimo nákladů a času),
 - Čas – celková doba projektu, nejbližší / nejvzdálenější datum počátku a ukončení, kdy vynaloženy náklady, kdy realizovány přínosy,
 - Náklady – celkové náklady podle druhů, způsob financování,
 - Návratnost investic – přínosy oproti nákladům, nevýhodám a rizikům,
 - Hlavní rizika.

3.2.12.2 Organizace

- **Vymezení rolí, odpovědností, organizační struktury. Ta zahrnuje:**
 - Korporátní / Programový management - zadání projektu (R111),
 - Projektový výbor - směřování, vedení projektu, zajištění zdrojů, rozhodování, schvalování plánů a produktů (R114),
 - Projektový manažer - řízení projektu, plánování, monitorování postupu, reporting Projektovému výboru (R103),
 - Týmový manažer - dodání výstupů, resp. produktů, řízení členů týmu, reporting Projektovému manažerovi (R115).
- **Řídící tým projektu** = Projektový výbor, Projektový manažer, Týmový manažer.
- **3 perspektivy:** Zákazník (byznys), Uživatel, Dodavatel.
- **Role řídicího týmu jsou:**
 - **Sponzor** (R113) – jediný člen Programového výboru, hlavní odpovědnost a rozhodovací pravomoc, zajišťuje finance, vlastní Obchodní případ, schvaluje výdaje, nastavuje tolerance,
 - **Hlavní uživatel** (RQ032) – požadavky, zajištění lidských zdrojů, schvaluje produkty a zajišťuje jejich využití a dosažení přínosů,
 - **Hlavní dodavatel** (R109) – zajišťuje dodavatelské zdroje, odpovídá za kvalitu produktů,
 - **Projektový dohled** (R116) – zajišťuje správný postup projektu, je nezávislý na Projektovém manažerovi, Projektové podpoře a Týmových manažerech,
 - **Změnová komise** (R117) – rozhoduje o změnách ovlivňujících rozsah a výstupy projektu, odpovědnost projektového výboru, lze delegovat,
 - **Projektový manažer** (R103) – řízení projektu, vydávání balíků práce, kontroly týmů, reporting,
 - **Projektová podpora** (R118) – administrativní podpora, poradenství, metodická pomoc, odpovědnost Projektového manažera, lze delegovat,
 - **Týmový manažer** (R115) – zajišťuje kvalitu produktů,
 - **Zainteresované strany** – jednotlivci nebo skupiny, které mohou mít vliv na projekt, nebo mohou být ovlivněny projektem.

3.2.12.3 Kvalita

- **Rozlišuje se** – produkty nedosahují požadovanou kvalitu, resp. produkty přesahují požadovanou kvalitu.
- **Specifikace** požadované kvality je v **Popisu produktu**. Řízení kvality **musí respektovat nastavený QMS** v organizaci, ISO normy apod. **Řešení kvality** zahrnuje plánování kvality a (operativní řízení kvality).

3.2.12.4 Plány

- Plánování představuje **3 úrovně plánu** – Plán projektu, Plán etapy, Týmové plány, případně Plán realizace výjimky.
- Využívá Techniku produktově orientovaného plánování.

3.2.12.5 Rizika

- Představují **nejisté události s dopadem na projekt** – hrozby, příležitosti.
- Řízení rizik **zahrnuje:**
 - Strategii řízení rizik,
 - Postup řízení rizik,

- Registr rizik.
- **Postup řízení rizik představuje:**
 - identifikaci rizik,
 - hodnocení,
 - plánování reakcí,
 - implementaci reakcí,
 - komunikování o rizicích.

3.2.12.6 Změna

- Představuje řízení změn, tj. **problémů a otevřených bodů** – v **Registru otevřených bodů** (požadavky na změnu, odchylky dodavatele od specifikace produktu).
- Součástí je **řízení konfigurace produktu, sledování konfiguračních položek** a jejich změn.
- Každá konfigurační položka má **záznam v Konfigurační knihovně**.

3.2.12.7 Progres

- Představuje **ovládací prvky, různé typy kontrol** – obsažené v **Dokumentaci nastavení projektu**.
- Manažerské etapy projektu znamenají **delegování pravomoci Projektovému manažerovi pouze na jednu etapu** s využitím principu podle výjimek, s rozhodovacími body o pokračování projektu.
- **Prvky kontroly zahrnují:**
 - tolerance,
 - časem řízení kontroly (Zprávy o stavu etapy, Zprávy o balíku práce),
 - událostí řízené kontroly (Zpráva o ukončení etapy, Zpráva o výjimce),
 - autorizace před zahájením prací (Schválení etapy, Autorizace Balíku práce).

3.2.13 Poznámky, reference

- Office of Government Commerce. Managing Successful Projects with PRINCE2 - (Office of Government Commerce)
- [Voříšek, J. ; Pour, J. - Management podnikové informatiky - \(Professional Publishing 2012\) - ISBN9788074311024](#), s. 26, (kap. Chlapek, Buchalcevoová),
- [Matoušková, S. - Řízení IT projektů v praxi a obecně uznávané metodiky - \(Praha. VŠE 2014\)](#),
- [Axelos. Examination Institutes. PRINCE2. \[Online\] \[Citace: 28. září 2014.\]](#),
- [Axelos. About Axelos. Axelos. \[Online\] \[Citace: 28. září 2014.\]](#).

3.3 Scrum

Grombířová, A. (KIT, VŠE)

3.3.1 Metodika Scrum

Cílem této kapitoly je charakterizovat agilní metodiku Scrum. Detailně popsat role, činnosti, artefakty a pravidla, která tato metodika definuje a používá. Za primární zdroj informací byl vybrán oficiální průvodce *The Scrum Guide* (Schwaber a Sutherland, 2017). Jako sekundární zdroj byla použita kniha od Josefa Myslína (Scrum: Průvodce agilním vývojem softwaru, 2016) a *Agilní metody řízení projektů* (Šochová a Kunc, 2014). Pokud není uvedeno jinak, kapitola čerpá právě z těchto zdrojů. Tato kapitola tvoří teoretické východisko pro praktickou část diplomové práce.

Podle nejnovějších výsledků průzkumu od společnosti VersionOne (12th Annual State of Agile Report, 2018) patří Scrum mezi nejvyužívanější agilní metodiky a spolu s metodikami ScrumBan a Scrum/XP Hybrid tvoří 70% podíl mezi ostatními používanými agilními metodikami.

Scrum představuje rámec (angl. framework) pro vývoj a řízení komplexních produktů. Jeho zakladateli jsou Ken Schwaber a Jeff Sutherland. Tato metodika se začala vyvíjet počátkem devadesátých let minulého století a je založena na teorii empirického řízení procesů. Empiristé tvrdí, že zkušenosti jsou tím nejdůležitějším zdrojem znalostí. Ve Scrumu je kladen důraz na samoorganizované týmy, zkušenosti, rychlost vývoje produktu a schopnost adaptovat se na změny. K optimalizaci predikce a řízení rizik využívá iterativní (opakující se) a inkrementální (přirůstkový) přístup (Schwaber a Sutherland, 2017).

3.3.2 Pilíře

Empirické řízení procesů se opírá o tři hlavní pilíře – **transparency** (transparentnost), **inspection** (inspekce) a **adaptation** (adaptace). Transparentnost představuje společné standardy, které používají všichni účastníci procesu. Prezентuje fakta, tak jak jsou. Všichni si důvěřují a společně usilují o dosažení cíle. Inspekce v této souvislosti znamená, že každý člen Scrum týmu kontroluje produkt, procesy a postupy za účelem neustálého zlepšování a dosažení cíle. Adaptace reprezentuje schopnost přizpůsobit se na základě výsledků inspekce. Všechny tři pilíře se vzájemně prolínají v průběhu celého procesu a pomáhají dodržovat pravidla a principy metodiky Scrum (Schwaber a Sutherland, 2017).

3.3.3 Základní pojmy a artefakty

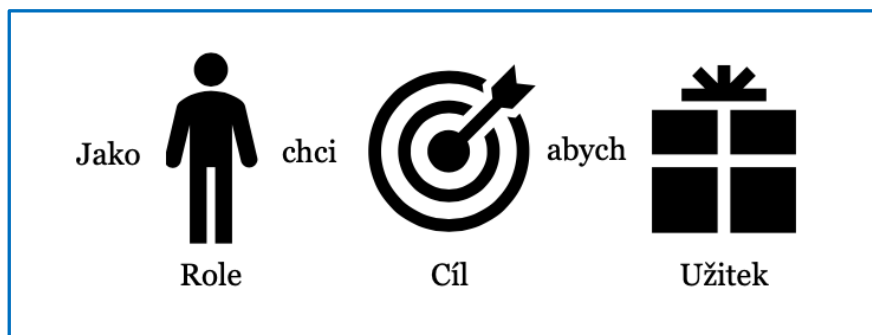
Scrum definuje, stejně jako jakákoliv jiná metodika, své vlastní pojmy a artefakty. Pro správné pochopení principu této metodiky jsou dále stručně uvedeny a objasněny alespoň základní pojmy a artefakty, se kterými se čtenář v průběhu čtení setká.

3.3.3.1 Story Point

Story Point je relativní jednotka, která se používá pro odhad velikosti jednotlivých požadavků. Odhad závisí na složitosti úkolu, riziku a množství práce, které je potřeba udělat. Většinou se Story pointy vyjadřují pomocí Fibonacciho posloupnosti (1, 2, 3, 5, 8, 13, 21, ...). Story pointy se nedají převádět na čas a odráží složitost pouze pro jeden daný tým. Nemá tedy smysl porovnávat Story pointy mezi jednotlivými týmy (Šochová a Kunc, 2014).

3.3.3.2 User Story

Jednoduše řečeno User Story představuje popis požadavku zákazníka. Měla by splňovat INVEST¹ kritéria, takže by měla být nezávislá, jednoznačně popsatelná, přinášet hodnotu, ohodnotitelná, malá a testovatelná. Nedílnou součástí každé User Story jsou tzv. akceptační kritéria (Šochová a Kunc, 2014). Akceptační kritéria pomáhají definovat a pochopit záměr uživatele. Nespecifikují konkrétní návrh řešení, jen upřesňují a rozvíjí, co by měl daný požadavek splňovat. Představují seznam podmínek, které musí User Story splňovat, aby byla akceptovaná jako dokončená. Pro zachování konzistence se při vytváření User Story používá jednotný formát „kdo, co, proč“:



Obrázek 3-1: Schéma User Story (autorka)

¹ INVEST – zkratka anglických slov Independent, Negotiable, Valuable, Estimable, Small, Testable

Jak lze vidět na Obrázek 3-1, User Story se skládá ze tří částí. První část definuje uživatele a jeho roli, druhá část funkcionalitu neboli cíl a poslední užitek (business hodnotu). Je tedy důležité si uvědomit jakou roli zrovna uživatel zastupuje, popsat jakou činnost musí systém vykonat a jaký to pro něj má účel. Příkladem správně napsané User Story je: „Jako zákaznice chci koupit lístky do kina online, abych nemusela stát frontu.“

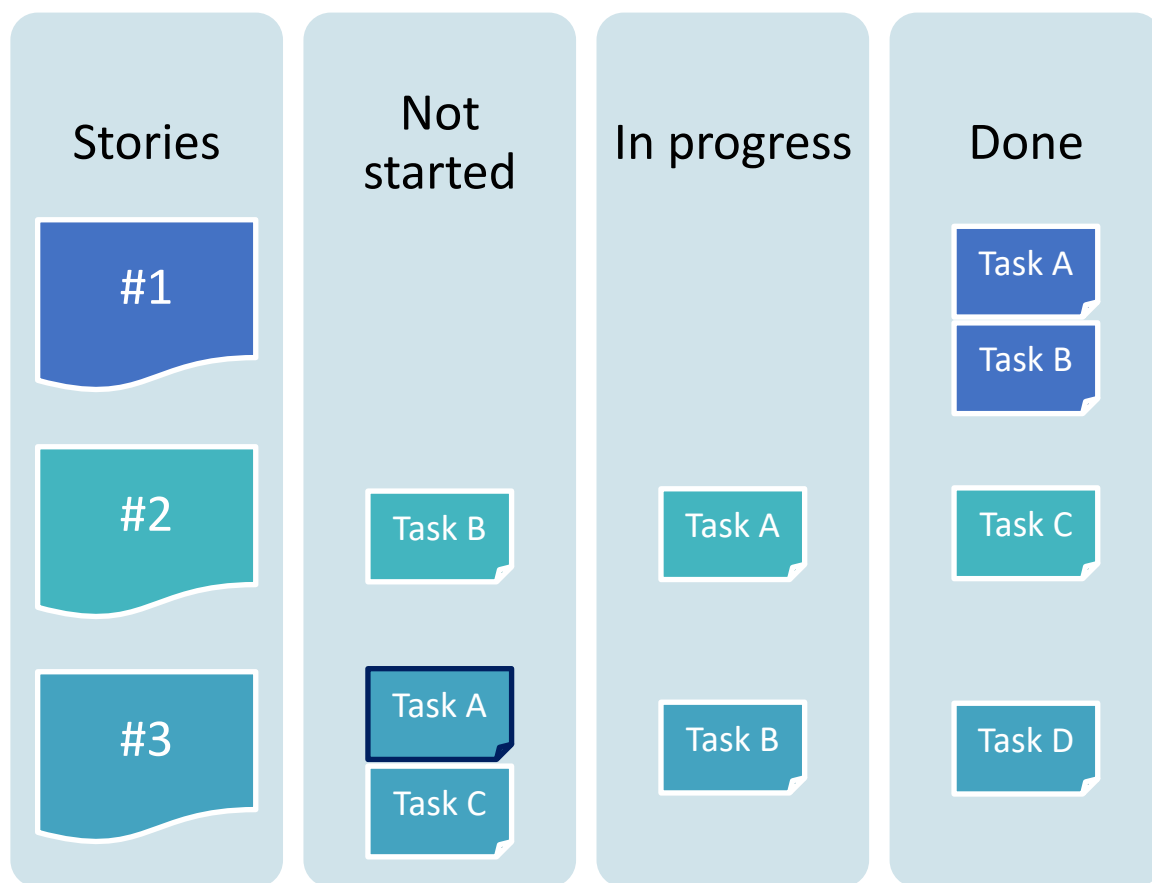
3.3.3.3 Product Backlog

Product Backlog neboli produktový backlog není nic jiného než seznam požadavků, které se musí v rámci vývoje softwaru implementovat. Obsahuje všechny výše zmíněné user stories včetně jejich popisu, odhadu náročnosti a priority. Představuje tedy seznam všech funkcí, vlastností, požadavků a chyb. Nejčastěji se eviduje ve formě tabulky nebo seznamu. Pro správu backlogu lze využívat nástroje pro řízení projektů jako jsou např. JIRA, TFS, VersionOne. Obsah produktového backlogu není stálý, aktualizuje se na základě změny požadavků od zákazníka a vyvíjí se souběžně s vývojem produktu. Správu backlogu má na starosti Product Owner. Jeho povinností je udržovat seznam aktuální, dostupný a seřazený dle priorit.

3.3.3.4 Sprint Backlog

Sprint Backlog obsahuje všechny úkoly, které je nutné splnit v aktuálním sprintu. Je tedy podmnožinou produktového backlogu. Stará se o něj celý Scrum tým a v průběhu sprintu se jeho obsah může měnit.

V praxi se lze setkat i s fyzickou podobou sprint backlogu ve formě tzv. Scrum Board. Příklad takové tabule je zachycen na Obrázek 3-2. Fyzická tabule slouží k lepší vizualizaci úkolů a používá se během Daily Meetingu.



Obrázek 3-2: Scrum Board (autorka)

3.3.3.5 Increment

Inkrement neboli přírůstek je suma všech dokončených položek z produktového backlogu. Koncem každého sprintu se tato hodnota navyšuje o počet nově naimplementovaných položek. Všechny tyto položky musí být potenciálně nasaditelné a v souladu s definicí „Done“, viz bod níže.

3.3.3.6 Definition of Done

Definice „Done“ představuje seznam pravidel a podmínek, které musí každá položka na konci sprintu splňovat, aby byla uznána jako doručená v požadované kvalitě. Je to soubor konvencí a standardů, které si každá vývojová organizace sama definuje. V průběhu projektu se tato definice rozrůstá, protože se zpřísňují nároky na kvalitu. Příkladem takové podmínky je, že na každou novou funkcionalitu je napsaný automatizovaný test.

3.3.3.7 Burndown chart

Pro zobrazení stavu sprintu a k predikci dokončení jednotlivých User Story se používá sloupcový graf, který má na ose x jednotlivé dny daného sprintu a na ose y sumu potřebných hodin k dokončení všech úkolů. Každý den by se měl počet zbývajících práce snižovat až k hodnotě nula k posledním dni sprintu.

Druhým příkladem použití grafu je pro zobrazení stavu celého projektu. Na ose x má jednotlivé sprinty a na ose y požadované funkcionality. Slouží jako pomůcka pro vlastníka produktu, aby věděl, kolik funkcionalit tým zvládne ještě dodat (Šochová a Kunce, 2014).

3.3.4 Role

Lidské zdroje jsou považovány za jeden z největších kapitálů softwarových projektů. Bez kvalitních lidí nemá projekt šanci na úspěch i za použití těch nejlepších technologií. Každý jedinec zastává v projektu určitou roli, která představuje soubor práv a povinností, očekávaného chování a jednání. To znamená, že každý pracovník má jasně definovanou roli, díky které ví, jaká práva a povinnosti zastává, zná své postavení v týmu, své úkoly a kompetence a jakou nese zodpovědnost. Je zřejmé, že v některých projektech může jedinec zastávat i více rolí najednou. To však vede ke konfliktu kompetencí a dopadá to tak, že ani jedna role není naplňována správně. Z toho důvodu je lepší, když každý jedinec má přiřazenou právě jednu roli.

V agilních metodikách se vyskytuje mnohem méně rolí než v těch tradičních. V metodice Scrum jsou definovány pouze tyto tři role (Schwaber a Sutherland, 2017): **Product Owner** (vlastník produktu), **Development Team** (vývojový tým) a **Scrum Master** (nemá český překlad). Všechny tyto role dohromady tvoří tzv. **Scrum Team** (Scrum tým).

Pro Scrum týmy je charakteristické, že jsou samoorganizované a multifunkční. To znamená, že si sami volí, jak budou pracovat a že mají všechny potřebné znalosti a dovednosti potřebné k dokončení práce a nejsou omezováni nikým zvenčí.

3.3.4.1 Product Owner

Product Owner definuje vizi celého projektu, včetně cílů a budoucího směřování. Je zodpovědný za maximalizaci hodnoty produktu. Představuje jakéhosi prostředníka mezi zákazníkem a vývojovým týmem. Zákazník musí říct, co si přeje a vývojový tým, zdali je možné to splnit.

Jeho další povinností je spravovat produktový backlog a jako jediný nese zodpovědnost za aktuálnost a správnost tohoto seznamu. Product Owner dále rozhoduje, které funkcionality se budou v příštím sprintu vyvíjet a celá organizace musí jeho rozhodnutí respektovat.

Mezi činnostmi spojené se správou produktového backlogu patří (Schwaber a Sutherland, 2017):

- Formulace jednotlivých požadavků na základě reprezentace zájmů zákazníka;
- Uspořádání položek a stanovení priorit na základě vize a cíle produktu;
- Optimalizace hodnoty práce vývojového týmu;
- Zajištění transparentnosti a dostupnosti požadavků;
- Zajištění toho, že vývojový tým rozumí položkám v produktovém backlogu;

Ačkoliv se může zdát, že je Product Owner vedoucím pracovníkem, jeho kompetence jsou omezené a sahají pouze do procesní stránky produktu. Rozhoduje sice o vizích, definuje úkoly a stará se, aby vyvíjený software odpovídal skutečným požadavkům zákazníka, ale nemá právo zasahovat do technické stránky produktu. Nerozhoduje o tom, jaké technologie se použijí a jak bude vývojový tým pracovat. V praxi začne tuto roli, po implementaci metodiky Scrum, většinou zastávat projektový manažer nebo vedoucí týmu.

3.3.4.2 Development Team

Vývojový tým se skládá z profesionálů, kteří nesou zodpovědnost za dodání inkrementu produktu na konci každého sprintu. Vývojový tým neobsahuje pouze programátory, ale i testery, analytiku, designéry a architekty. V rámci vývojového týmu se všem zpravidla říká vývojář, nehledě na to, jakou funkci zastává. Každý člen týmu by měl mít určitou schopnost, tak aby dohromady dokázali dosáhnout cíle sprintu. Ideální velikost týmu tvoří tři až devět členů. Tým musí být dostatečně flexibilní a schopný dokončit zřetelný kus práce. Pro vývojové týmy řízené podle metodiky Scrum je charakteristické, že jsou samostatně organizované. To znamená, že je nikdo neřídí. Sami se rozhodují, jak budou pracovat, řešit konflikty, kdy budou mít jednotlivé Scrum ceremonie apod. To vede k větší efektivitě práce.

3.3.4.3 Scrum Master

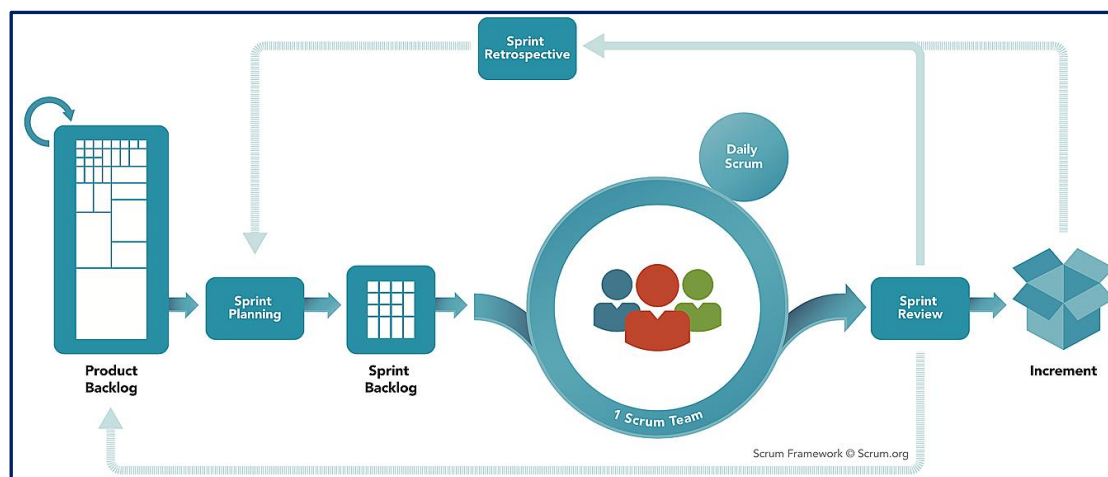
Scrum Master má na starosti kontrolu dodržování pravidel Scrumu, jejich osvojování a šíření. Je zodpovědný za to, aby všichni lidé ve Scrum týmu chápali praktiky a techniky Scrumu. Není vedoucím týmu, ale takovým pomocníkem. Snaží se pomáhat, jak vývojovému týmu, tak vlastníkovému produktu, za účelem maximalizace hodnoty jejich práce. Scrum Master by měl být komunikativní, vnímavý, trpělivý a tlumit případné konflikty.

Jeho náplní práce je (Schwaber a Sutherland, 2017):

- vést vývojový tým k samoorganizaci a multifunkčnosti;
- pomáhat vývojovému týmu k vytváření produktů s vysokou hodnotou;
- odstraňovat překážky se kterými se vývojový tým potýká;
- ochraňovat tým před vnějšími vlivy;
- motivovat tým k lepším výsledkům;
- moderovat všechny Scrum meetingy;
- vzdělávat vývojový tým v prostředí organizace, ve kterých ještě není Scrum plně osvojen;
- iniciovat změny, které vedou k vyšší produktivitě týmu;
- pomáhat product ownerovi s efektivním udržováním produktového backlogu;
- školit organizaci v osvojování Scrumu;
- pomáhat s implementací metodiky a spolupracovat s ostatními Scrum Mastery.

3.3.5 Činnosti

Činnosti představují časově ohraničené úseky, ve kterých se provádí určité úkoly. Scrum definuje pět hlavních činností, kterými jsou (Schwaber a Sutherland, 2017): **Sprint**, **Sprint Planning** (plánovací schůzka), **Daily Scrum** (denní schůzka), **Sprint Review** (vyhodnocení sprintu) a **Sprint Retrospective** (retrospektiva sprintu). Jejich vzájemná provázanost a posloupnost je zachycena na Obrázek 3-3. Všechny tyto události zajišťují pravidelnost jednotlivých úkolů a snižují potřebu vzniku nových meetingů. Každá činnost má pevně stanovený maximální čas, který by neměla překročit. Všechny činnosti jsou navrženy tak, aby přispívaly k transparentnosti, inspekci a adaptaci některého z aspektů Scrumu. Vynechání jakékoliv události vede ke snížení transparentnosti, ztrátě kontroly a znemožnění adaptace.



Obrázek 3-3: Scrum Framework (Scrum.org)

3.3.5.1 Sprint

Sprint je vlastně souhrnem těch třech zbývajících činností. Jedná se o časový úsek, který nesmí překročit délku trvání jednoho měsíce. Vždy záleží na projektu a počtu lidí ve Scrum týmu, ale nejčastější jsou dvoutýdenní sprinty (Ambler S. , 2018). Nový sprint začíná vždy hned po ukončení toho předchozího. Každý sprint má svůj cíl, ve kterém je specifikováno, co má být zrovna naimplementováno. Tento cíl bývá definován na plánovací schůzce. Během sprintu se pak vyvíjí inkrement produktu dle definice „Done“, to znamená, že se například programuje nová funkcionality, testuje se, fixují se bugy a dělá se vše pro to, aby byl splněn cíl sprintu.

Jak už bylo zmíněno výše, každý sprint se tedy skládá z plánovací schůzky, denních stand-upů, samotné práce vývojářů, vyhodnocení sprintu a závěrečné retrospektivy. Během sprintu platí různá pravidla, která se musí dodržovat. Prvním pravidlem je, že se neprovádí žádné změny, které by mohly ohrozit naplnění cíle sprintu. Zároveň se nesmí snižovat kvalita cíle sprintu, ale může se upravit jeho rozsah.

Ke zrušení sprintu může dojít v situaci, kdy se cíl sprintu stane zastaralým a neplatným. Právo zrušit sprint má pouze Product Owner, ale může to udělat i na základě pobídky od ostatních. V takovém případě se všechny nedokončené položky vrátí zpět do produktového backlogu a ty hotové se vyhodnotí, zdali je možné je nasadit.

3.3.5.2 Sprint Planning

Sprint planning neboli plánovací schůzka slouží, jak už název napovídá, k naplánování všech úkolů a činností, které je potřeba v rámci sprintu udělat. Této schůzky se účastní Product Owner a vývojový tým. Scrum Master má za úkol meeting zprostředkovat, moderovat a dohlížet na to, že se strany pochopily. Pro měsíční sprint by měla tato schůzka trvat maximálně 8 hodin. Pokud je sprint kratší, tak se čas úměrně sníží.

Vstupem pro tuto událost je seřazený produktový backlog, naplánována kapacita vývojového týmu na další sprint a velocita (rychlost) týmu za uplynulý sprint (kolik Story pointů doručili). Výstupem je Sprint Backlog neboli seznam požadavků, které se vývojový tým zavázal v příštím sprintu doručit a nadefinovaný cíl sprintu.

Během této schůzky tedy dochází k naplánování a odhadu náročnosti nových funkcionalit a k rozdělení úkolů mezi jednotlivé členy týmu.

3.3.5.3 Daily Scrum

Tato událost je známá i pod pojmem „Daily Stand-up“, protože by se u ní správně mělo stát. Měla by trvat maximálně 15 minut a každý den se na stejném místě a ve stejný čas opakovat. Probíhá tak, že všichni členové vývojového týmu po sobě odpovídají na tyto tři otázky:

- Co jsem udělal/a včera, abych pomohl/a týmu splnit cíl sprintu?
- Co budu dělat dnes, abych pomohl/a týmu splnit cíl sprintu?
- Existují nějaké překážky, které mi brání ve splnění cíle sprintu?

Není povinné, aby se této schůzky účastnili i Scrum Master a Product Owner. Scrum Master má pouze za úkol dohlédnout na to, aby se schůzka konala a naučit vývojový tým dodržovat maximálně možný limit. Product Owner se samozřejmě může přijít podívat, ale není tam v pozici kontrolora nebo manažera. Daily Scrum slouží hlavně členům vývojového týmu, aby naplánovali práci na dalších 24 hodin a aby věděli, na čem kdo zrovna pracuje a zdali nemá nějaké potíže.

Správně prováděný meeting vede ke zvýšení produktivity a transparentnosti, zlepšení vzájemné komunikace a spolupráce a podporuje týmovost. To, že se při meetingu stojí vede k větší efektivitě a soustředění jednotlivých členů. Tato schůzka zajišťuje kontrolu (zdali je směřováno k naplnění cíle sprintu) a adaptaci (upravit činnosti v případě překážek) týmu.

3.3.5.4 **Sprint Review**

Sprint Review je určené pro vývojový tým, vlastníka produktu, zákazníka a ostatní zainteresované strany. Scrum Master je zde ve funkci zprostředkovatele a moderátora. Vyhodnocení se provádí vždy na konci každého sprintu. Při měsíční délce sprintu tato činnost trvá nejvýše čtyři hodiny. Pokud je sprint kratší než měsíc, tak se čas úměrně zkrátí.

Tento meeting slouží k diskuzi výsledků sprintu. Provádí se demo, kde vývojáři ukazují (demonstrují), jaké funkcionality se podařilo vyvinout. Případně odpovídají na vzniklé dotazy. Dále diskutují, co se během sprintu dařilo, k jakým problémům docházelo a jak byly následně odstraněny. Product Owner má za úkol objasnit, které úkoly byly dokončeny a které ne a z jakých důvodů. Je zde bezprostřední prostor k získání zpětné vazby od zákazníka a k úpravě dalších požadavků.

Výsledkem vyhodnocení sprintu je upravený produktový backlog s vytipovanými položkami, které se s největší pravděpodobností budou vyvíjet v příštím sprintu.

3.3.5.5 **Sprint Retrospective**

Retrospektiva je považována za jednu z nejpřínosnějších a nejdůležitějších činností, i když se velmi často autorka setkává s názorem, že je to pouze ztráta času. V rámci této schůzky se schází celý Scrum tým a pracuje na věcech, které je potřeba do příště zlepšit. Vzniká zde prostor ke vnitřní kontrole týmu, toho, jak pracují a navzájem mezi sebou komunikují. Retrospektiva je poslední činností v rámci sprintu a trvá maximálně tři hodiny (měsíční sprint). Samozřejmě i u této schůzky se doba zkracuje dle trvání sprintu.

Scrum Master zde zastává důležitější roli, protože se musí na každou retrospektivu předem připravit. Vybrat techniku/formát, kterým bude schůzka vedena. Jeho náplní je zajistit, aby schůzka proběhla a aby účastníci pochopili její účel. Scrum Master má za úkol povzbuzovat ostatní ke změnám, které by mohli vést ke zlepšení Scrum procesu.

Smyslem retrospektivy je identifikovat věci, které se členům Scrum týmu líbily a dobře fungovali. Stejně tak odhalit záležitosti, které je potřeba do příštího sprintu zlepšit. Občas je součástí i tzv. appreciation, kdy si členové navzájem vyjadřují dík a uznání za něco, co se jim během sprintu povedlo.

Výstupem tohoto jednání je seznam akcí, které si musí Scrum tým osvojit a stanovení metrik, jejichž cílem je sledovat, zdali se naplánované akce dodržují a dochází ke kýženému zlepšení (výsledku).

Mezi další praktiky patří tzv. **Product Backlog Refinement** neboli Grooming. Během této schůzky diskutuje Product Owner se zbytkem týmu vybrané požadavky a odhaduje se jejich složitost (nejčastěji ve Story pointech). Zároveň upřesňuje specifikaci jednotlivých User stories a případně je dělí na menší části. Pořádá se pravidelně jednou nebo dvakrát za sprint. Výstupem je připravený seřazený produktový backlog na dva až tři sprinty dopředu.

3.3.6 **Škálování agilních metodik**

Cílem této kapitoly je uvést čtenáře do problematiky škálování agilních metodik. Popsat Agile Scaling Model, LeSS framework a Spotify model. Jako hlavní zdroj informací byla použita kniha od autorů C. Larman a B. Vodde (Large-Scale Scrum: More with LeSS, 2016) a práce od Scotta Amblera (The Agile Scaling Model (ASM): Adapting Agile Methods for Complex Environments, 2009) pro společnost IBM.

Původně byly agilní metodiky navrženy pro menší nezávislé týmy. V posledních letech se však začaly uplatňovat i pro vývoj velkých komplexních systémů, kde mnohdy pracuje více týmu najednou na

jednom společném produktu. A to ani nemusí sedět ve stejné pobočce. Při takových podmínkách je velmi důležitá vzájemná spolupráce, komunikace a koordinace napříč všemi skupinami.

V souvislosti s tím vznikl pojem *škálování agilních metodik*, který představuje úpravu a přidání nových praktik dle aktuálních potřeb daného projektu. Tyto změny nastávají ve chvíli, kdy se neustále zvětšuje počet objektů v projektu a jejich objem práce. Základní agilní metodiky tyto předpoklady nejsou schopny pokrýt, proto vznikají nové rámce, které praktiky rozšiřují a upravují.

3.3.7 Agile Scaling Model (ASM)

Agile Scaling Model představuje obecný rámec pro škálování agilních metodik. Definuje plán, jak efektivně přijmout agilní strategie a přizpůsobit je komplexním projektům. Jedná se o průvodce, který organizaci připravuje na změnu. Jeho autorem je uznávaný metodolog Scott W. Ambler.

ASM dělí agilní metodiky a přístupy do tří hlavních kategorií (Ambler S. W., Agile Scaling Model: Be as Agile as You Need to Be, 2011):

- Agile Development (agilní vývoj)
- Agile Delivery (agilní dodávka)
- Agility at Scale (škálovaná agilita)

Představitelem první kategorie jsou metodiky Scrum, Extreme Programming (XP), Agilní modelování a Feature-Driven Development (FDD), které vyznávají principy samo-organizovaných a spolupracujících týmů. Soustředí se pouze na část životního cyklu, a to na vývoj softwaru, který je řízen hodnotou. Jsou vhodné pro malé, společně umístěné týmy vyvíjející jednoduchý software.

Druhá kategorie rozšiřuje agilní vývoj tak, aby řešil celý životní cyklus systému, od zahájení projektu až po uvedení do provozu. Životní cyklus je kromě hodnoty řízený i rizikem. Zavádí milníky, které zvyšují transparentnost projektu a zvyšují šanci na úspěch projektu. Reprezentantem jsou agilní metodiky Dynamic System Development Method (DSDM) a OpenUP. Také jsou vhodné pro menší lokální týmy, které poskytují jednoduché řešení.

Poslední kategorie se zaměřuje na disciplinovaný agilní vývoj softwaru, kde je použitelný jeden nebo více faktorů. Mezi škálovací faktory patří: velikost vývojového týmu, geografické rozložení, složitá organizační struktura, soulad s předpisy, kulturní nebo organizační komplexita, technická složitost a celopodniková disciplína. Všechny tyto faktory vedou k tomu, že se agilní praktiky upravují, aby se eliminovala rizika spojená se škálováním metodik a aby se adaptovaly na nově vzniklou situaci. Takovým rizikem mohou být například bankovní regulace, které nařizují dodatečné požadavky na systém.

Prvním krokem ke škálování agilních přístupů je podle Amblera (Agile Scaling Model: Be as Agile as You Need to Be, 2011) „*přijmout disciplinovaný agilní životní cyklus dodávek a rozpoznat, které faktory škálování jsou použitelné pro projektové týmy a přizpůsobit přijatou strategii tak, aby řešila rozsah složitostí, kterým týmy zrovna čelí.*“

Mezi nejznámější škálovací rámce patří Scaled Agile Framework (SAFe), Scrum of Scrums, Large Scale Scrum, Nexus a Disciplined Agile Delivery. V posledních letech se do podvědomí dostává i Spotify model.

3.3.8 Large Scale Scrum (LeSS)

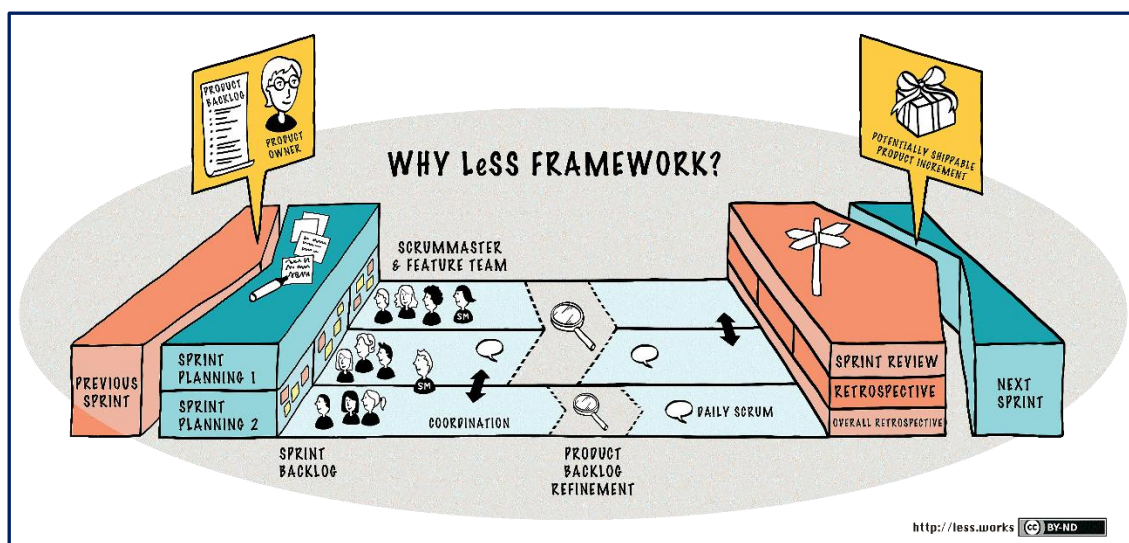
Jak už bylo zmíněno výše, jedním z rámců, který se používá pro škálování agilních metodik je Large Scale Scrum. Začal se vyvíjet v roce 2005 a je dostupný ve dvou variantách (Larman a Vodde, 2016). Jedná se o jednoduchý framework, který staví na základních principech metodiky Scrum. Používá se v případě, kdy vzniká stále více týmů, které pracují na stejném produktu a když se zvyšuje objem a rozsah jejich práce.

První variantou je samotný **LeSS** framework, který poskytuje prostředí pro maximálně osm týmů, přičemž v každém z nich by mělo ideálně být 8 lidí. Pokud spolupracuje na projektu více jak osm týmů, používá se druhá varianta takzvaný **LeSS Huge**. Liší se složitější strukturou produktového backlogu, kdy jsou jednotlivé položky rozděleny do tzv. **Requirement areas** (oblasti požadavků). Za každou oblast pak zodpovídá a spravuje **Area Product Owner** neboli takový zástupce vlastníka produktu. Rámec LeSS Huge tak definuje minimálně o jednu roli navíc než klasický Scrum. Každý Area Product Owner má na starosti svou klíčovou oblast požadavků (může jich být i víc), které rozumí a komunikuje

s vývojovým týmem a zadává jim požadavky na výslednou funkcionalitu (The LeSS Company B.V., 2019).

Jak lze vidět na Obrázek 3-4 LeSS framework se od klasického Scrumu liší tím, že má navíc **Overall Retrospective** a **Sprint Planning Two** meeting. Při celkové retrospektivě se schází zástupci všech týmů a probírají záležitosti a problémy, které mají vliv i na ostatní týmy. Stejně jako u Scrum retrospektivy je schůzka řízena Scrum Masterem, který má za úkol shromažďovat podněty z klasických retrospektiv a upozornit na možné problémy, které je potřeba vyřešit. Řeší se zde například, jak dobře spolu týmy spolupracují, zdali se od sebe navzájem učí a jestli dodržují stanovená pravidla.

Plánovací schůzky jsou rozděleny do dvou kol. V prvním kole se schází zástupci týmů s vlastníkem produktu a vybírají si požadavky (User Story), které si vezmou do nadcházejícího sprintu. Poté už se sejdou samotné týmy a diskutují jednotlivé položky ze svých Sprint backlogů. Rozdělují si práci a rozkládají požadavky na jednotlivé úkoly (tasks).



Obrázek 3-4: LeSS Framework (The LeSS Company B.V., 2019)

Autorka zmiňuje vybraný framework, protože v současnosti už bylo potřeba rozšířit a upravit agilní praktiky i ve společnosti, pro kterou pracuje. Během jejího působení se zvýšil počet týmů a rozsah projektu. Takže se začíná nasazovat a uplatňovat škálovací rámec LeSS.

Začátkem léta se chystá restrukturalizace banky a přechod na agilní principy řízení na úrovni celé organizace, takže je příhodné zde popsat i Spotify model, který posloužil jako předloha pro novou organizační strukturu.

3.3.8.1 Spotify Tribe

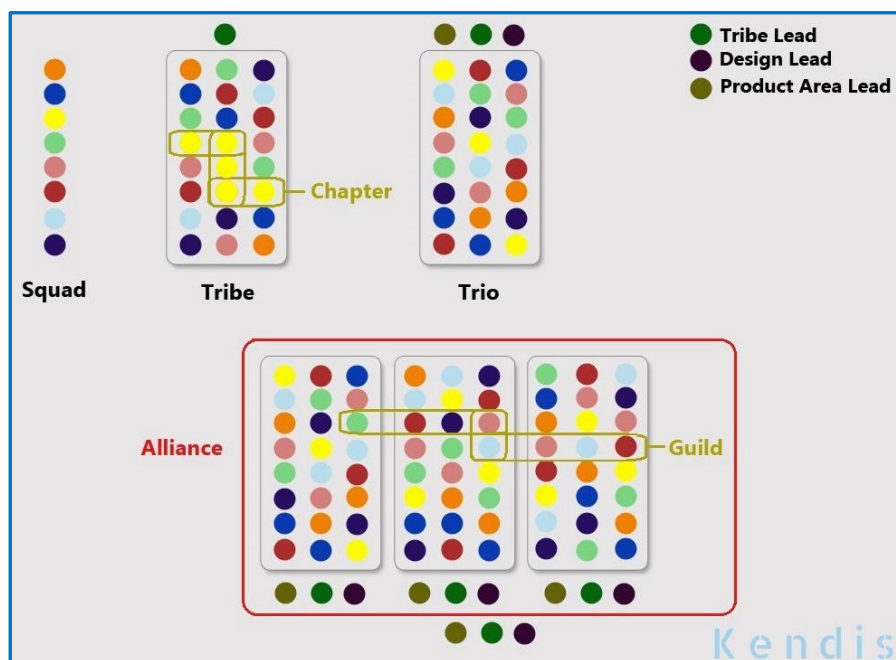
Za zrodem tohoto modelu stojí firma Spotify, která poskytuje službu umožňující streamovat a podcastovat hudbu. K lednu 2018 měla společnost více jak 140 milionů uživatelů v 61 zemích (Kamer, 2018). Z toho 70 milionů placících uživatelů (Koutský, 2018). V současné době zaměstnává jen oddělení vývoje a výzkumu přes 1800 zaměstnanců, kteří jsou rozmístěni do 180 týmů. K efektivnímu řízení práce, a tak velkého počtu lidí byli nuceni si vymyslet vlastní škálovací agilní rámec, který nese název Spotify Tribe a vychází z metodiky Scrum.

Jejich firemní kultura staví na důvěře a autonomii jednotlivých týmů, což vede k větší motivovanosti zaměstnanců. Upřednostňují agilní principy před praktikami Scrumu. Na rozdíl od metodiky Scrum nepoužívají roli Scrum Mastera, ale roli agilního kouče (Agile Couch). Spotify Tribe dále definuje tyto klíčové elementy (Kniberg, 2014):

1. Squad – agilní tým podobný Scrum týmu, který je nezávislý, multifunkční a samoorganizovaný. Většinou má méně jak 8 lidí, kteří spolu sedí v jedné místnosti a kteří mají společnou dlouhodobou misi a sdílenou zodpovědnost za výslednou část produktu. Každý Squad si sám rozhoduje o tom, jak bude pracovat a dle jaké agilní metodiky bude řízen (Scrum, Kanban, XP, mix). Má k dispozici agilního kouče, vlastníka produktu a je v kontaktu se stakeholderem.

2. Tribe – množina agilních týmů, které pracují v souvisejících oblastech. Každý kmen má svého lídra, který je zodpovědný za poskytování co nejlepšího prostředí pro skupiny v tomto kmenu. Kmeny jsou navrženy tak, aby pokud možno nepřesáhly 100 lidí.
3. Chapter – odborník na horizontální úrovni organizace, který má na starosti lidi z různých týmů uvnitř jednoho kmene. Má za úkol je podporovat v osobním růstu a specifických výzvách.
4. Guild – neformální skupina tvořená lidmi z různých kmenů, kteří mají zájem spolu tvořit guild. Součástí guildu se může stát osoba z jakéhokoli týmu, kmenu či jiného uskupení. Jejich úkolem je řešit problémy a držet týmy v rovnováze.
5. Trio – každý kmen, který má vlastní design, produktovou oblast a svého vůdce.
6. Alliance – 3xTrio
7. Chief Architect – klíčový článek organizace. Hlavní architekt, který definuje vizi a architekturu systému. Každý systém má svého vlastníka, který sleduje závislosti mezi jednotlivými systémy a udržuje kontrolu nad jejich releasy. Za účelem zmírnění rizik jsou povinnosti architektů rozdělené na Dev a Ops.

Grafická provázanost výše uvedených elementů je znázorněna na **Obrázek 3-5**.



Obrázek 3-5: Spotify Tribe Engineering Model (Kendis Team, 2018)

3.3.9 Poznámky, zdroje

Ambler, Scott W. The Agile Scaling Model (ASM): Adapting Agile Methods for Complex Environments. místo neznámé : IBM Rational, 2009.

Ambler, Scott W. Agile Scaling Model: Be as Agile as You Need to Be. Agile Alliance. [Online] 2011. Dostupné z: <https://www.agilealliance.org/wpcontent/uploads/2016/01/Agile-Scaling-Model.pdf>.

Ambler, Scott W. 2018 IT Success Rate Survey Results. [Online] 2018. Dostupné z: <http://www.ambysoft.com/surveys/success2018.html#Results>.

Ambler, Scott W. 2018 State of Software Development Survey Results. [Online] 2018. Dostupné z: <http://www.ambysoft.com/surveys/softwareDevelopment2018.html>.

Atlassian. DevOps: prolomení bariéry mezi vývojem a provozem. Atlassian.com. [Online] 2019. Dostupné z: <https://cs.atlassian.com/devops>.

Balada, Jakub. Agilní metodiky v komplexním prostředí. Praha : Disertační práce. Vysoká škola ekonomická v Praze, Fakulta informatiky a statistiky. Vedoucí práce Alena Buchalcegová, 2015.

Beck, Kent a Beedle, Mike. Manifesto for Agile Software Development. agilemanifesto.org. [Online] 2001. Dostupné z: <http://agilemanifesto.org/iso/cs/manifesto.html>.

Buchalcegová, Alena. Metodiky vývoje a údržby informačních systémů: kategorizace, agilní metodiky, vzory pro návrh metodiky. 1. vydání. Praha : Grada Publishing, 2005. ISBN 80-247-1075-7.

Buchalcegová, Alena. Metodiky budování informačních systémů. Praha : Oeconomica, 2009. ISBN 978-80-245-1540-3.

Černý, Jan. Analýza problémů agilních projektů firmy. Praha : Vysoká škola ekonomická v Praze, Fakulta informatiky a statistiky, 2015.

Derby, Esther a Larsen, Diana. Agile Retrospectives. Making Good Teams Great. Raleigh : The Pragmatic Bookshelf, 2006. ISBN 0-9776166-4-9.

Gurecková, Veronika. Vedení agilní transformace týmu ve společnosti IBM. Brno : Masarykova univerzita, Fakulta informatiky, 2017.

Hendl, Jan. Kvalitativní výzkum: základní teorie, metody a aplikace. 2. aktualiz. vyd. Praha : Portál, 2008. ISBN 978-80-7367-485-4.

Hendl, Jan. Kvalitativní výzkum: základní teorie, metody a aplikace. Čtvrté, přepracované a rozšířené vydání. Praha : Portál, 2016. str. 440. ISBN 978-80-262-0982-9. 80

Jesson, Jill K., Matheson, Lydia a Lacey, Fiona M. Doing Your Literature Review: Traditional and Systematic Techniques. Londýn : SAGE Publications Ltd, 2011. ISBN 978-1-84860-153-6.

Kadlec, Václav. Agilní programování: Metodiky efektivního vývoje softwaru. 1. vydání. Brno : Computer Press, 2004. ISBN 80-251-0342-0.

Kamer, Jurriaan. How to Build Your Own "Spotify Model". Medium.com. [Online] 9. Únor 2018. Dostupné z: <https://medium.com/the-ready/how-to-build-your-own-spotify-model-dce98025d32f>.

Kendis Team. Exploring Key Elements of Spotify's Agile Scaling Model. Medium.com. [Online] 23. Červenec 2018. Dostupné z: https://medium.com/@media_75624/exploring-key-elements-of-spotifys-agile-scalingmodel-471d2a23d7ea.

Kniberg, Henrik. Spotify engineering culture (part 1). Spotify Labs, 2014.

Komori, Michi. Thematic Analysis. Design Research Techniques. [Online] [Citace: 3.. Leden 2019.] Dostupné z: <http://designresearchtechniques.com/casestudies/thematicanalysis/>.

Koutský, Zdeněk. Spotify si měsíčně předplácí více než 70 milionů uživatelů. Dotekomanie.cz. [Online] 6. Leden 2018. Dostupné z: <https://dotekomanie.cz/2018/01/spotify-si-mesicne-predplaci-vice-nez-70-milionuuzivatelu/>.

KPMG. Přínosy agilní transformace. KPMG. [Online] 2018. Dostupné z: https://assets.kpmg/content/dam/kpmg/cz/pdf/summary_CZE_Prinosity_agilni_transfo rmace_2018-09.pdf.

KPMG. Agilní přeměna - 8 doporučení z Nizozemska. KPMG. [Online] 29. Březen 2019. [Citace: 31. Březen 2019.] Dostupné z: <https://home.kpmg/cz/cs/home/clanky-aanalzy/2019/03/agilni-premena-8-doporuceni-z-nizozemska.html>.

Kugler, Filip. Dopady zavedenia agilnej metodiky Scrum v burzovej spoločnosti. Praha : Vysoká škola ekonomická v Praze, Fakulta informatiky a statistiky, 2018.

Kulhánková, Barbora. Behaviour Driven Development a Scrum v korporátním prostředí. Praha : Vysoká škola ekonomická v Praze, Fakulta informatiky a statistiky, 2016.

Larman, Craig a Vodde, Bas. Large-Scale Scrum: More with LeSS. místo neznámé : Addison-Wesley Professional, 2016.

Liniový manažer. Osobní rozhovor. Praha, 2. 4. 2019. ManagementMania. Člověkodén (Man-day). ManagementMania.com. [Online] ©2011-2016. Dostupné z: <https://managementmania.com/cs/clov-ekoden-manday>. 81

Mersino, Anthony. Agile Projects are More Successful than Traditional Projects. Vitalitychicago.com. [Online] 1. Duben 2018. Dostupné z: <https://vitalitychicago.com/blog/agile-projects-are-more-successful-traditional-projects/>.

Myslín, Josef. Scrum: Průvodce agilním vývojem softwaru. 1. vydání. Brno : Computer Press, 2016. ISBN 978-80-251-4650-7.

Ozierańska, Aneta, et al. The Critical Factors of Scrum Implementation in IT Project - the Case Study. Katowice : Journal of Economics and Management, University of Economics in Katowice, 2016. Journal of Economics and Management. ISSN 1732-1948.

Popelák, Martin. Využití metodiky SCRUM ve vývojových týmech. Praha : České vysoké učení technické v Praze, 2016.

Scrum.org. The Scrum Framework Poster. Scrum.org. [Online] © 2019. Dostupné z: <https://www.scrum.org/resources/scrum-framework-poster>.

Seppanen, Jaana. Scrum - from Theory to Practice in Software Development. Tampere : Tampere University of Technology, 2016.

Schwaber, Ken a Beedle, Mike. Agile Software Development with Scrum. Portland : Prentice Hall, 2002.

Schwaber, Ken a Sutherland, Jeff. The Scrum Guide. Scrum.org. [Online] 2017. Dostupné z: <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-GuideUS.pdf#zoom=100>.

Šochová Zuzana. Super User Story, User Story a Epics. Zuzi's blog. [Online] 24. Listopad 2011. Dostupné z: <https://soch.cz/blog/management/agile/scrummanagement/super-user-story-user-story-a-epics/>.

Šochová, Zuzana a Kunc, Eduard. Agilní metody řízení projektů. 1. vydání. Brno : Computer Press, 2014. ISBN 978-80-251-4194-6.

Šochová, Zuzana. Agilní transformace. Sochova.cz. [Online] © 2010-2019. Dostupné z: <https://sochova.cz/novinky/novinky/agilni-transformace.htm>.

Šochová, Zuzana. Skvělý ScrumMaster. 1. vydání. Brno : Computer Press, 2018. ISBN 978-80-251-4927-0.

Techopedia. User Acceptance Testing (UAT). Techopedia.com. [Online] © 2019. Dostupné z: <https://www.techopedia.com/definition/3887/user-acceptance-testing-uat>.

The LeSS Company B.V. Product Backlog Refinement. Less: More with Less. [Online] © 2014 ~ 2019. Dostupné z: <https://less.works/less/framework/product-backlogrefinement.html>.

The LeSS Company B.V. LeSS Framework. Large-Scale Scrum: More with LeSS. [Online] © 2014 ~ 2019. Dostupné z: <https://less.works/less/framework/index.html>. 82

The Standish Group. The CHAOS Report 1994. Standishgroup.com. [Online] 1994. Dostupné z: https://www.standishgroup.com/sample_research_files/chaos_report_1994.pdf.

The Standish Group. The CHAOS Report 2015. Standishgroup.com. [Online] 2015. Dostupné z: https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf.

Tománek, Martin. Řízení projektů agilního vývoje softwaru na základě PRINCE2 a Scrum. Praha : Diplomová práce. Vysoká škola ekonomická v Praze, Fakulta informatiky a statistiky. Vedoucí práce Zora Říhová, 2015.

VersionOne. 12th Annual State of Agile Report. Collab.net. [Online] 9. Duben 2018. Dostupné z: <https://explore.versionone.com/state-of-agile/versionone-12th-annual-stateof-agile-report>.

3.4 Azure DevOps

3.4.1 Metodika DevOps

Pojmenování DevOps představuje složeninu dvou pojmů: Development a Operations. Oddělené postavení softwarového vývoje od softwarových operací. Toto oddělené postavení může způsobit

například zpoždění odhalení chyb. Pokud se část tohoto procesu nepředvídatelně prodlouží, může to vést až k opoždění celé iterace. Nové pojetí DevOps má vytvořit propojení a společné fungování obou celků. Na obrázku (Obrázek 3-6) je graficky zpracované propojení obou částí vývoje a operací. Na levé straně obrázku jsou provázané aktivity vývoje. S přechodem na pravou část si můžeme zaznamenat změnu na aktivity, které jsou spíše operačního charakteru.



Obrázek 3-6: - DevOps (ZDROJ: DevOps metrics)

S příchodem vznikají odlišné názory na definici a vlastnosti DevOps. Někteří nový přístup považují za abstraktní rámec, zatímco jiní se přiklání k tomu, že se jedná spíše o popis postupu práce nebo sadu dovedností.

Existuje podobnost metodologie DevOps s agilním přístupem. Shodné pojetí spočívá především v zaměření na jednotlivce, iterace a spolupráce mezi zainteresovanými osobami. Velký důraz je kladen především na automatizaci. DevOps funguje jako rozšíření agilního rámce a principy aplikuje na celou organizaci. Steve Ingall dodává, že „DevOps není nový proces ani nástroj, takže to ve skutečnosti není další typické „agilní“ řešení.“ (Zdroj: Is DevOps another Project Management Methodology?)

Probereme si hlavní dvě oblasti, na kterých je DevOps postavený. V první řadě je fungování založeno na týmech. U zrodu DevOps stálo přesvědčení a

Druhá část, kterou jsem vybral je tématika nástrojů pro využívání konceptu DevOps.

V poslední části bych odkázal na metriky, které byly důkladně rozebrány s tematikou Agilních metodik. Metriky jsou ve velké míře podobné i v této problematice DevOps.

3.4.2 Azure DevOps

Samotná aplikace Azure DevOps naplňuje hlavní cíl textu, proto je nutné se na začátek detailněji zaměřit na její části. V úvodu si řekneme něco základní informace o aplikaci. V následujících částech se přesuneme k charakteristikám jednotlivých funkcí, které budeme využívat při vývoji Business intelligence. Tato kapitola má sloužit jako zdroj informací pro čtenáře, aby se v jednotlivých částech aplikace dokázal orientovat.

Azure DevOps je produkt společnosti Microsoft, která je založená na principech cloudu. Mimo jiné slouží jako místo pro ukládání kódu, se kterým nepřetržitě pracují všichni vývojáři a testéři. Hlavním cílem aplikace je podpora práce k dosažení kvalitních výsledků.

Aplikace od společnosti Microsoft pokrývá následující funkce:

- Máme možnost využít uložení přímo integrované do toho řešení nebo možnost využít Git pro ukládání zdrojového kódu
- Podpora průběžné integrace a dodávky vytvářených aplikací
- Využití agilních přístupů pro podporu plánování a sledování práce
- Metodika Scrum a Kanban je zde používána pro jednoduché plánování iterací
- Testovací nástroje pro zajištění kvality výstupů
- Nastavitelné dashboardy pro sledování progresu, výkonnosti a mnoho dalšího
- Možnost uložení dokumentů a manuálů, které budou dostupné pro všechny členy

3.4.3 Azure DevOps Services vs. Azure DevOps Server

Máme možnost si na začátek vybrat mezi dvěma možnostmi, jak Azure DevOps využívat. Aplikace nám poskytuje dvě základní řešení, která nabízejí integrované prostředí, agilní nástroje pro plánování a další společné funkcionality popsané výše. Existuje však několik zásadních rozdílů v jednotlivých pojetích.

První případ nese označení „Azure DevOps Services“, který byl dříve pojmenovaný jako „Visual Studio Team Services (VSTS)“. Hlavním rozdíl je viditelný v tom, že poskytované řešení funguje na principu cloudu. Druhá možnost pojmenovaná jako „Azure DevOps Server“ je poskytovaná jako on-premise řešení. Stejně jako v prvním případě došlo v rámci aktualizace a rozšiřování funkcí k přejmenování. Původně se Azure DevOps Server nazýval „Visual Studio Team Foundation Server (TFS)“.

Azure DevOps Services je hostovaná verze souboru Azure DevOps Server. Jak již bylo zmíněno, obě řešení poskytují určitý výčet společných funkcionalit. Na druhou stranu můžeme nalézt několik zásadních odlišností.

Základní rozdíly se projevují v následujících oblastech:

- Rozsah dat
- Autentizace
- Uživatelé a skupiny
- Bezpečnost a ochrana dat

3.4.4 Funkce Azure DevOps

Funkcionality aplikace je potřeba vybrat tak, abychom se nejvíce přiblížili našim cílům. Jako většina aplikací i Azure DevOps poskytuje mnoho možností, jak si komponenty seskládat. V kontextu s tématem je výběr jednotlivých částí závislý na tvorbě Business intelligence ve výrobní společnosti.

3.4.5 Azure Repos

V první řadě je důležité mít spolehlivé uložení pro vytvářený kód. Azure Repos poskytuje sadu nástrojů pro správu kódu. V současné době nezáleží na tom, jestli vytvářený projekt je velkého rozsahu nebo se jedná o jednoduchý program. V obou případech je vhodné využívat verzování a zálohování každé části práce.

Systémy pro správu verzí nám umožňují sledovat a řídit změny, které jsou prováděny v uplynulém čase. Při jakékoli úpravě kódu říkáme systému, aby vytvořil „snímek“ našich nových souborů a bezpečně je uložil. Snímky jsou ukládány natrvalo, proto v případě potřeby je možné se k daným úpravám kdykoli vrátit. Pomocí správy verzí dokážeme zajistit jednoduchou práci mezi jednotlivými členy týmu. Nejdůležitějším požadavkem na správné řešení je to, aby všichni členové pracovali s nejaktuálnějším jednotným kódem.

Azure Repos poskytuje přehlednou organizaci verzí daného vývoje. V případě problémů se můžeme okamžitě vrátit na verzi, která fungovala bez problémů. Aplikace má přehledně graficky zpracované uložení kódů, kdy jsou chronologicky řazené jednotlivé změny. Ke každým změnám je přiřazeny člověk, který ji uložil a můžeme také nahlédnout, o jakou změnu se jednalo.

Správa verzí v aplikaci Azure DevOps je poskytována ve dvou možnostech:

- Git

- Team Foundation Version Control (TFVC)

3.4.5.1 Team Foundation Version Control

Pro analýzu aplikace a ukázkou funkcionalit budeme využívat právě tuto možnost pro správu verzí. V následujících řádcích si charakterizujeme tuto část.

Tento systém správy verzí je centralizovaný a využívání je možné jak pro malé, tak i pro rozsáhlejší vývojové projekty. Pro rozhodování ve výběru tohoto systému napomáhají faktory, že členové týmu mají pouze jednu verzi každého souboru. Další skutečností je to, že historická data jsou ukládána pouze na serveru a zde jsou také vytvářena větvění projektu.

3.4.6 Azure Pipelines

Tento balíček služeb dodává řešení hlavní význam, v čem spočívá smysl DevOpsu. Azure Pipelines je služba, která umožňuje automatické vytváření a testování kódu. Přínos pro vývojové týmy spočívá především ve dvou hlavních oblastech. Těmito oblastmi jsou takzvané „continuous integration“ a „continuous delivery“. Popis částí je rozepsán níže. Výsledek implementace tohoto souboru má zajistit kvalitní kód, který bude snadno dostupný pro uživatele.

3.4.6.1 Continuous Integration

Continuous integration, zkráceně jen CI, se využívá pro automatické budování projektu a následné testování. CI mimo jiné napomáhá k zachytávání vývojových problémů ve fázích, kde oprava je rychlá a není tak náročná. Testy vytvářeného kódu probíhají pravidelně a kontinuálně. Soubory známé jako artefakty (v původním znění „artifacts“) jsou vytvářeny právě díky CI systémům.

3.4.6.2 Continuous delivery

V druhé části se zaměříme na continuous delivery, v praxi více využívané zkrácené označení CD. Tato služba se zabývá automatickým nasazováním a testováním vývojového kódu. Tento proces probíhá ve více fázích, aby se zvýšila výsledná kvalita. Kvalita spočívá také v tom, že se využívá kód, který byl ve fázi CI otestován. V předchozím odstavci jsme zjistili, že CI produkují artefakty. Na základě „spotřeby“ těchto artefaktů jsou vytvářeny nové verze vytvářeného softwaru.

Tato řešení mohou pracovat se všemi jazyky nebo platformami a je možné nasazovat na všech operačních systémech (Windows, Linux a Mac).

3.4.7 Azure Boards

Azure Boards přináší do vývojového prostředí jednoduché a rychlé sledování úloh a funkcí. Řešení poskytuje řízení práce pomocí principů Scrum, Kanban, backlogů, týmových dashboardů a vlastního reportingu. Sledování dílčích úkolů může být prováděno napříč celým týmem nebo u jednotlivých členů odděleně.

Řešení poskytuje více procesů, podle kterých je možné sledovat a řídit týmovou práci. Můžeme si vybrat mezi základními procesy, agilními, Scrum nebo CMMI.

3.4.8 Azure Test Plans

Velmi významným faktorem při tvorbě systému je kvalita výstupu. Testování vyvíjeného softwaru nám zjišťuje stav, v jakém se projekt nachází. Azure DevOps poskytuje bohaté nástroje pro několik typů testování. Hlavní myšlenkou toho řešení je jednoduchost a dostupnost pro všechny členy týmu. Hlavním cílem je právě kontrola kvality vytvářeného softwaru.

Zaznamenáváme pojem manuální testování, který se v této problematice objevuje. Jak je již zřejmé z názvu, spočívá to v tom, že všechny testy jsou vytvářeny a prováděny ručně.

V oblasti manuálního testování Azure DevOps poskytuje čtyři druhy testovacích metod.

- Plánované manuální testování
- Uživatelem přijaté testování
- Průzkumné testování
- Feedback stakeholderů

Zaměříme se na stručný popis manuálního testování, protože na základě této metody budeme zkoumat funkcionality tohoto rozšíření.

3.4.8.1 Plánované manuální testování

Řešení Azure DevOps poskytuje rozšíření pro vytváření bohaté nástroje pro vytváření kvalitních testovacích scénářů. Hlavní myšlenkou je opět kooperace napříč celého vývojového procesu za účelem zajištění kvalitního výstupu. Princip manuálního testování spočívá v organizaci testů do určitých sad a plánů. Testování je poté přiděleno jednotlivým testerům, kteří jsou za provedení testu zodpovědní.

Na začátek je nutné vymezit si tři základní pojmy, se kterými se budeme setkávat. Těmito pojmy jsou testovací plány, testovací sady a testovací případy. Testovací plány vystupují jako vytvořený soubor složený z jednotlivých testovacích sad. Testovací sada obsahuje soubor testovacích případů, které byly definovány.

3.4.9 Zdroje

ANON, 2017. Microsoft Visual Studio 2017 Released for Public Download. ICT Monitor Worldwide; Amman [online]. [cit. 2019-03-30]. Dostupné z: <http://search.proquest.com/docview/1875258135/abstract/E6663F95D7794218PQ/7>

BANICA, Logica, Magdalena RADULESCU, Doina ROSCA a Alina HAGIU, 2017. Is DevOps another Project Management Methodology? [online]. 21(3), 39–51 [cit. 2019-03-30]. ISSN 14531305. Dostupné z: <http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=125625305&lang=cs&site=ehost-live>

COBB, Charles G., 2015. The Project Manager's Guide to Mastering Agile. B.m.: Wiley. ISBN 978-1-118-99104-6.

DAVIS, Jennifer a Katherine DANIELS, 2016. EffectiveDevOps. 1. vyd. B.m.: O'Reilly. ISBN 978-1-4919-2630-7.

FORSGREN, Nicole a Mik KERSTEN, 2018. DevOps Metrics. [online]. 61(4), 44–48 [cit. 2019-03-28]. ISSN 00010782. Dostupné z: <https://dl-acm.org/zdroje.vse.cz/citation.cfm?id=3182626>

GÁLA, Libor, Jan POUR a Zuzana ŠEDIVÁ, 2015. Podniková informatika. Třetí vydání. Praha: Grada. ISBN 978-80-247-5457-4.

GARTNER, 2019. Continuous Integration (CI). Gartner IT Glossary. [online]. [cit. 2019-04-05] Dostupné z: <https://www.gartner.com/it-glossary/continuous-integration-ci/>

HOUSTON, Dan X., 2014. Agility beyond software development. In: Proceedings of the 2014 International Conference on Software and System Process [online]. [cit. 2019-03-23]. ISBN 978-1-4503-2754-1. Dostupné z: <http://dl.acm.org/citation.cfm?id=2600821.2600837>

JOHNSON, Bruce, 2017. Professional Visual studio 2017. B.m.: Wiley. ISBN 978-1-119-40459-0.

KATARIA, Nikita, 2016. Implementing Agile Development at Scale: An Industry Case Study [online]. [cit. 2019-03-30]. Dostupné z: <https://digitalcommons.du.edu/cgi/viewcontent.cgi?article=2125&context=etd>

KIMBALL, Ralph a Margy ROSS, 2013. The Data Warehouse Toolkit. 3. vyd. B.m.:Wiley. ISBN 978-1-118-53080-1.

KUPIAINEN, Eetu, Mika V. MÄNTYLÄ a Juha ITKONEN, 2014. Why are industrial agile teams using metrics and how do they use them? In: Proceedings of the 5th International Workshop on Emerging Trends in Software Metrics [online]. [cit. 2019-04-05]. ISBN 978-1-4503-2854-8. Dostupné z: <http://dl.acm.org/citation.cfm?id=2593868.2593873>

MBI, 2015. MBI. Management Byznys informatiky [online]. [cit. 2019-03-16]. Dostupné z: www.mbi.vse.cz

MICROSOFT, 2018a. Exploratory and manual testing overview - Azure Test Plans [online]. [cit. 2019-03-26]. Dostupné z: <https://docs.microsoft.com/enus/azure/devops/test/overview>

MICROSOFT, 2018b. Choose a process such as Basic, Agile, Scrum, or CMMI - Azure Boards [online]. [cit. 2019-03-30]. Dostupné z: <https://docs.microsoft.com/enus/azure/devops/boards/work-items/guidance/choose-process> MICROSOFT, 2018c. Plan, code, collaborate, ship applications - Azure DevOps Services [online]. [cit. 2019-03-23]. Dostupné z: <https://docs.microsoft.com/enus/azure/devops/user-guide/what-is-azure-devops-services>

MICROSOFT, 2018d. What is Azure Pipelines? - Azure Pipelines [online]. [cit. 2019-03-30]. Dostupné z: <https://docs.microsoft.com/en-us/azure/devops/pipelines/getstarted/what-is-azure-pipelines>

MICROSOFT, 2019a. Collaborate on code - Azure Repos [online]. [cit. 2019-03-26]. Dostupné z: <https://docs.microsoft.com/en-us/azure/devops/repos/get-started/what-is-repos>

MICROSOFT, 2019b. Download SQL Server Data Tools (SSDT) - SQL Server [online]. [cit. 2019-05-05]. Dostupné z: <https://docs.microsoft.com/en-us/sql/ssdt/download-sqlserver-data-tools-ssdt>

MICROSOFT, 2019c. Team Foundation Version Control (TFVC) index to content - Azure Repos [online]. [cit. 2019-03-30]. Dostupné z: <https://docs.microsoft.com/enus/azure/devops/repos/tfvc/>

MICROSOFT, 2019d. Understand differences between Azure DevOps Services and Azure DevOps Server - Azure DevOps [online]. [cit. 2019-03-23]. Dostupné z: <https://docs.microsoft.com/en-us/azure/devops/user-guide/about-azure-devopsservices-tfs>

MICROSOFT, 2019e. Understand what you get with Azure Boards and the Basic process - Azure Boards [online]. [cit. 2019-03-26]. Dostupné z: <https://docs.microsoft.com/enus/azure/devops/boards/get-started/what-is-azure-boards-basic-process>

RADINGER, Wolfgang a Karl Michael GOESCHKA, 2003. Agile software development for component based software engineering. In: Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications [online]. B.m.: ACM, s. 300–301 [cit. 2019-04-30]. ISBN 978-1-58113-751-4. Dostupné z: <http://dl.acm.org/citation.cfm?id=949344.949422>

ROY, Utpal a Susmita DAS, 2011. Agile Software Development Methodology and its Cost Estimation Technique. National Journal of System and Information Technology; New Delhi. [online]. 4(2), 209–225 [cit. 2019-03-30] ISSN 09743308. Dostupné z: <http://search.proquest.com/docview/1478064437/abstract/97BF728361A644C3PQ/1>

STAMELOS, Ioannis G a Panagiotis SFETSOS, 2007. Agile Software Development Quality Assurance. B.m.: IDEA GROUP PUBLISHING. ISBN 978-1-59904-216-9.

VERMA, Nitya a Amy VOIDA, 2016. Mythologies of Business Intelligence. In: Conference Extended Abstracts on Human Factors in Computing Systems [online]. [cit. 2019-03-16]. ISBN 978-1-4503-4082-3. Dostupné z: <http://dl.acm.org/citation.cfm?id=2851581.2892379>

3.5 *Rational Unified Process (RUP)*

3.5.1 Vývoj a obsah RUP

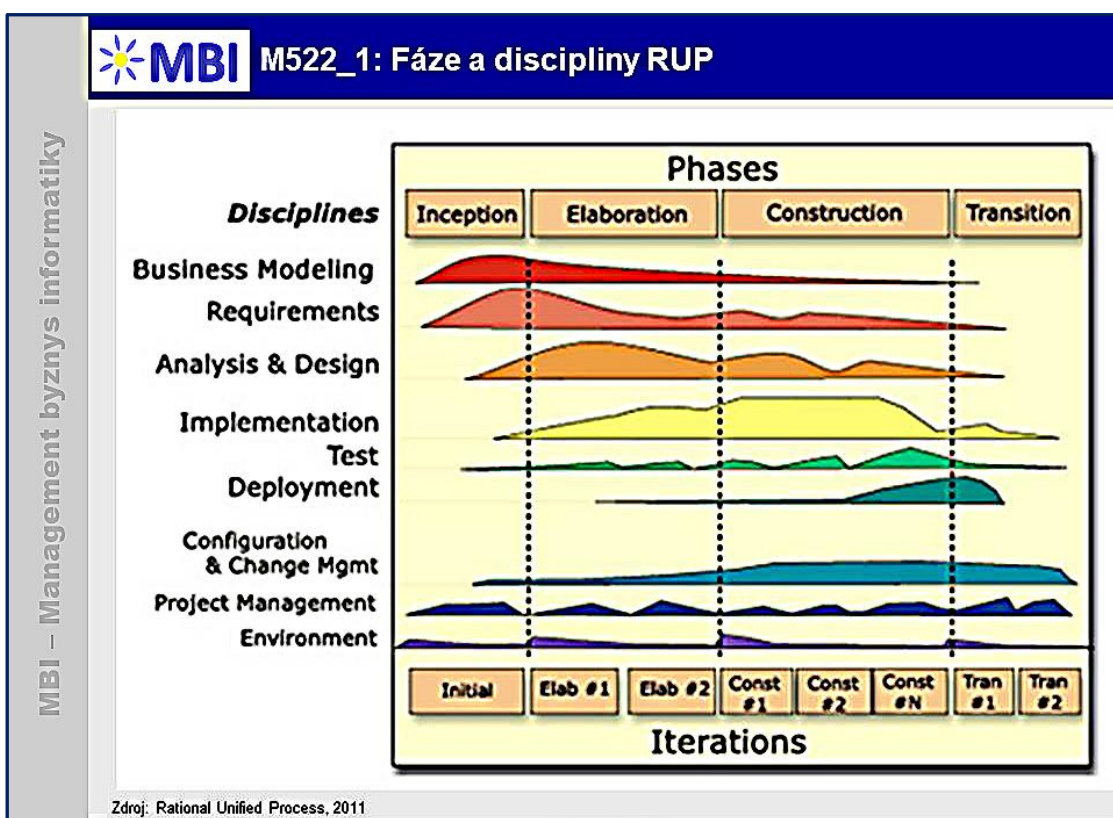
- Metodika Rational Unified Process vznikla v roce 1995, později byla vytvořena její zjednodušená verze Unified Process, kterou popsali Ivar Jacobson, Grady Booch a James Rumbaugh v knize „**The Unified Software Development Process**“. Od roku 2003 je metodika RUP **doplněována o agilní praktiky**.
- Poslední verze je postavena na konceptu **Unified Method Architecture (UMA)**. Koncept UMA **sjednocuje různé metody a jazyky procesního inženýrství** pomocí jednotného meta-modelu, který definuje terminologii a strukturu uspořádání metodického obsahu a procesů

vývoje. Je postaven na oddělení znovupoužitelného metodického obsahu od jeho konkrétní aplikace v procesech určitého projektu.

- Metodika Rational Unified Process je deskriptivní, **obsahuje dobře definovaný proces vývoje systému**, který se často používá pro vývoj systémů založených na objektových a komponentových technologiích. Je založen na takových principech softwarového inženýrství jako je iterativní, požadavky řízený a na architektuře založený přístup k vývoji softwaru.

3.5.2 Životní cyklus RUP

- Scott Ambler [AMBLER, 2005] charakterizuje **životní cyklus RUP** jako:
 - sekvenční ve velkém,
 - iterativní v malém,
 - s přírůstkovými dodávkami SW v pravidelných cyklech,
 - prováděný podle nejlepších praktik.
- Proces vývoje softwaru** je v metodice RUP popsán **v rámci dvou dimenzí**.



Obrázek 3-7: Fáze a disciplíny RUP

- Horizontální osa** představuje dynamický pohled na proces, který je vyjádřen pomocí cyklů, fází, iterací a milníků.
- Vertikální osa** reprezentuje statické hledisko procesu, popis činností, artefaktů, pracovníků a pracovních toků, které jsou seskupeny do disciplín.
- Graf ukazuje **podíl jednotlivých disciplín** v různých fázích projektu.

3.5.3 Fáze RUP

- Sekvenční aspekt** je zachycen pomocí fází, **iterativní charakter RUPu** je reprezentován **disciplínami**, které se provádějí všechny v rámci iterací.

- Každá **fáze má definovanu množinu cílů**. Splnění těchto cílů **se ověřuje** na konci fáze při dosažení milníku, který je pro každou fázi definován.
- Pro každý milník jsou **specifikována kritéria splnění cílů fáze**, na jejichž základě se rozhoduje o postupu do další fáze.
- RUP **definuje 4 fáze**:
 - Počáteční fáze (Inception),
 - Elaborační fáze (Elaboration),
 - Konstrukční fáze (Construction),
 - fáze Nasazení (Transition).
- V rámci **Počáteční fáze** **definují cíle projektu, požadavky, vytváří se harmonogram projektu** (plán iterací), odhadují se náklady projektu a definují rizika. Součástí této fáze může být vytvoření modelu nebo jednoduchého prototypu, na kterém se ověří, zda je možné se zvolenou technologií a pomocí zvolených nástrojů klíčové požadavky splnit. Počáteční fáze končí rozhodnutím, zda je projekt za daných požadavků, dostupných technologií, zdrojů a rozpočtu možné realizovat.
- **Cílem Elaborační fáze je definovat architekturu systému**. V této fázi by měl být vytvořen prototyp, který ověří všechny architektonické principy a umožní zpřesnění plánu realizace systému. Měly by být definovány komponenty, které je třeba vyvinout pro opakované použití.
- **Obsahem Konstrukční fáze je návrh a realizace systému včetně testování**. Prosazuje se pokud možno paralelní vývoj.
- **Cílem fáze Nasazení je zajistit, aby uživatelé mohli systém používat**. Součástí této fáze je školení uživatelů, předání dokumentace, vytvoření help-desku atd.
- RUP je **založen na iterativním vývoji**. V každé fázi probíhá **jedna nebo více iterací**, přičemž v jedné iteraci se prochází všechny disciplíny.
- Iterace **se věnují jen části** vyvíjeného systému - přírůstku. Pro každou iteraci se vytváří detailní **Plán iterace**.
- **Každá iterace**, zejména ve fázi Konstrukce, **končí dodáním fungující části systému**.

3.5.4 Poznámky, reference

- [Bruckner, T. ; Voříšek, J. ; Buchalceková, A. ; kolektiv - Tvorba informačních systémů: Principy, metodiky, architektury - \(Grada Publishing 2012\) - ISBN9788024779027](#), pro MBI zpracovala: Buchalceková, A. (katedra IT, VŠE)

3.6 Unified Modeling Language (UML)

Rahm, V. (KIT, VŠE)

3.6.1 Celková charakteristika

- Jazyk UML vznikl v polovině 90. let a je **na něm postaveno mnoho objektově orientovaných metodik**. O rozvoj jazyka se stará skupina **Object Management Group (OMG)**.
- Podstatnou výhodou jazyka UML je jeho **nezávislost na procesu vývoje**, protože jazyk UML není svázán s žádnou konkrétní vývojovou metodikou.
- Jazyk UML prošel během svého rozvoje **několika změnami a rozšířeními**. Příkladem je **Object Constraint Language (OCL)**, který je využíván pro definici omezení.
- Pro ukládání modelů UML existuje standardizovaný formát **XML Metadata Interchange (XMI)**. Pomocí XMI je **možné přenášet UML modely mezi různými CASE nástroji**.

- Za pomoci diagramů umožňuje jazyk UML **zachytit systém na různé úrovni abstrakce**. UML obsahuje **14 typů diagramů**, které jsou **rozděleny na diagramy chování a diagramy struktury**.
- Mezi **diagramy chování** patří diagram aktivit, stavový diagram, diagram případů užití a všechny diagramy interakcí. Do diagramu interakcí patří diagram komunikace, diagram přehledu interakcí, sekvenční diagram a diagram časování.
- Do **diagramů struktury** patří diagram tříd, diagram vnitřní struktury, diagram komponent, diagram nasazení, objektový diagram (někdy též nazýván diagram instancí), diagram profilů a diagram balíčků.
- **Diagramy chování** jsou určeny k **zachycení chování systému**. Diagramy chování zachycují dynamické chování objektů v systému, a to včetně **jejich metod, spolupráce, aktivity a stavové historie**. Dynamické chování systému může být popsáno jako série změn systému v určitém čase (Object Management Group, 2015).
- **Diagramy struktury** jsou určeny k **zachycení elementů nezávislých na čase**. Diagramy struktury zachycují **statickou strukturu objektů v systému** a to znamená, že zachycují elementy ve specifikaci, která je nezávislá na čase. Elementy v diagramu struktury **reprezentují koncept aplikace a mohou zahrnovat koncept z reálného světa, abstraktní koncept a implementační koncept**. Strukturální diagramy nezachycují detail dynamického chování, který je zachycen pomocí diagramů chování. (Object Management Group, 2015).

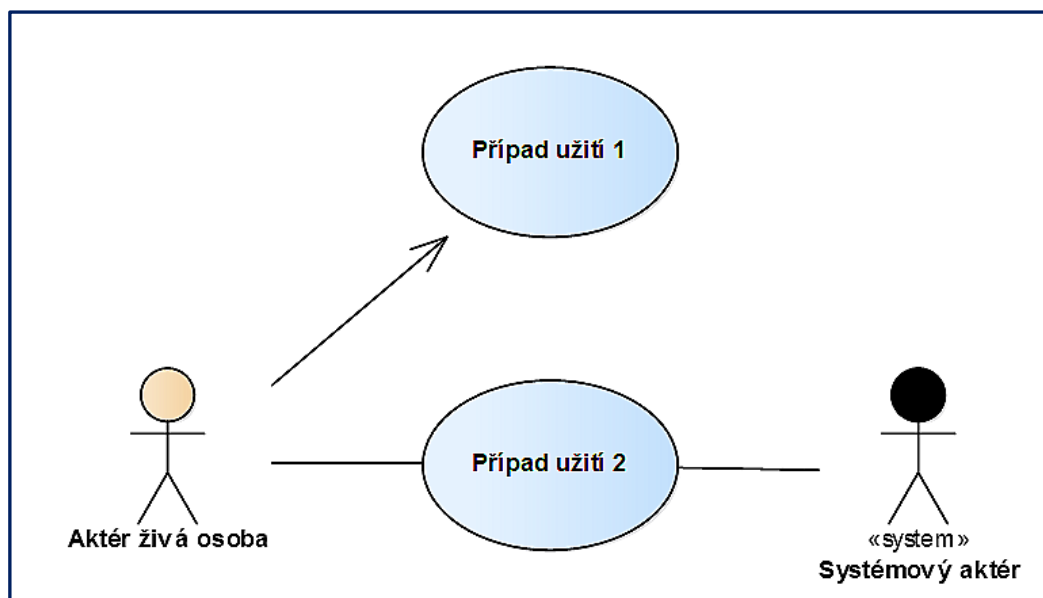
3.6.2 Typy diagramů UML

- **Diagram profilů:** Metamodel k zachycení stereotypů jako tříd a profilů jako balíčků,
- **Diagram tříd:** Určen k zachycení tříd, atributů, metod a vztahů mezi objekty,
- **Diagram vnitřní struktury:** K zachycení interní struktury prvku a spolupráce, kterou daná struktura umožňuje,
- **Diagram komponent:** Pro zachycení vztahů jednotlivých komponent, které tvoří větší komponenty anebo celý informační systém. Určeno pro svévolné zachycení komplexních systému.,
- **Diagram nasazení:** Určen k zachycení hardwarových zdrojů a softwarových komponent. Diagram umožňuje zachycení jejich spolupráce, lokality a další.,
- **Objektový diagram:** Zachycuje kompletní nebo částečnou strukturu systému ve specifický čas.,
- **Diagram balíčků:** K zobrazení obsahu a závislostí jednotlivých balíčků.,
- **Diagram aktivit:** Reprezentace posloupnosti aktivit s podporou rozhodování, iterace a souběžnosti.
- **Diagram případů užití:** Zachycuje uživatelskou interakci se systémem a chování systému,
- **Stavový diagram:** Určen k zobrazení jednotlivých stavů objektů a přechodů mezi nimi.,
- **Sekvenční diagram:** Pro zachycení interakce mezi jednotlivými objekty a její posloupnosti,
- **Diagram komunikace:** Zobrazením toku zpráv a vzájemných vztahů zachycuje komunikaci třídy, sekvence a případu užití,
- **Diagram přehledu interakcí:** Stejný význam jako má diagram aktivit s tím, že jednotlivé aktivity představují diagramy interakcí,
- **Diagram časování:** Pro zachycení chování objektu ve specifický časový úsek.

3.6.3 Use Case diagramy

- **Diagram případů užití** zachycuje **funkční požadavky na systém**.
- Diagram případů užití umožňuje zachytit, jaký typ uživatele používá systém a jaké činnosti vykonává.

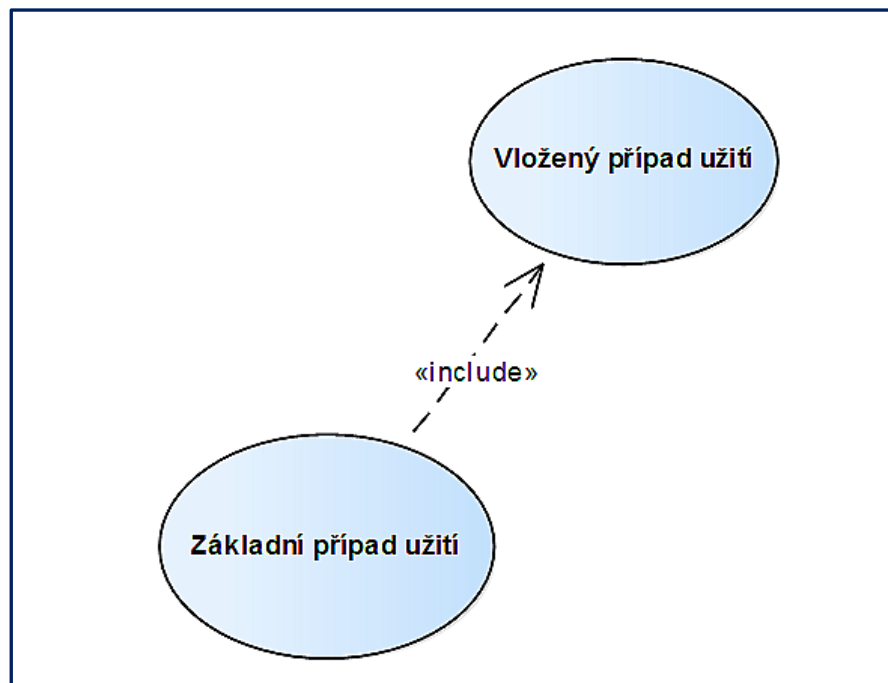
- Nejvýznamnějšími prvky diagramu případů je případ užití, aktér a vztah.
- Na modelu případů užití jsou zobrazeny zejména jednotlivé případy užití a jejich slovní popis. Případy užití představují funkční požadavky na systém.
- Prvek diagramu s názvem aktér zachycuje uživatele, který komunikuje se systémem. Aktérem může být buď živá osoba, jiný systém anebo hardware se kterým modelovaný systém komunikuje. Aktér je v jazyce UML zobrazen symbolem, který u sebe obsahuje název modelovaného aktéra. Živým aktérem může být například operátor anebo uživatel bankomatu. Neživým aktérem může být například core banking systém,



Obrázek 3-8: Případ užití, příklad

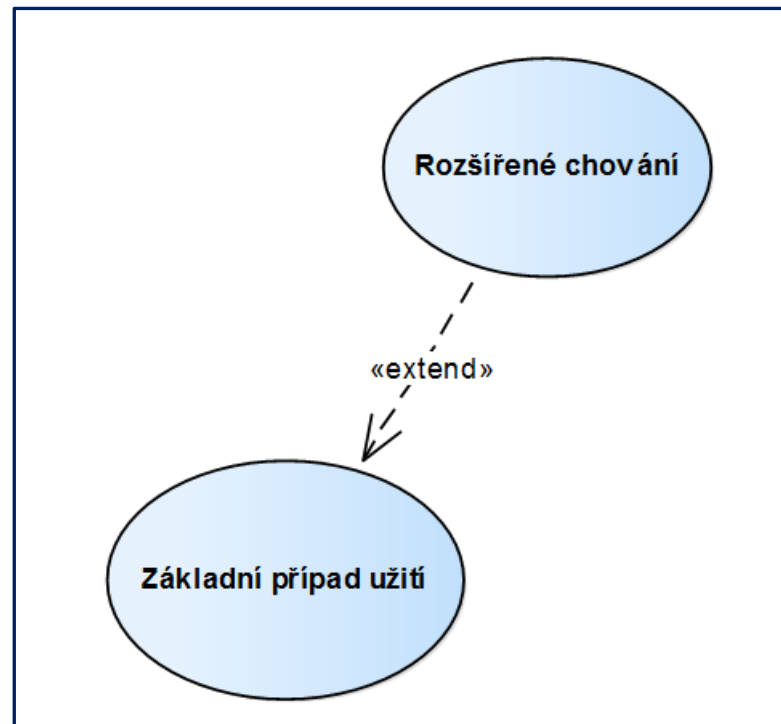
- Významným prvkem diagramu je případ užití, který je určen pro charakteristiku funkcionality systému plnící určitý cíl a je používán aktérem.
- Každý případ užití dále obsahuje slovní popis. Samotná funkcionality případu užití je specifikována jako posloupnost interakcí mezi aktérem a systémem. Posloupnost interakcí je pro případ užití označeno jako scénář případů užití (dále scénář). Scénář je určen jednomu průchodu případem užití od začátku do konce neboli jedné instanci případu užití. Jednotlivé případy užití začínají určitou spouštěcí událostí a pokračují, dokud není cíl případu užití splněn anebo dokud není případ užití přerušen. Obrázek obsahuje schématické znázornění případů užití,
- Na obrázku jsou znázorněny různé průchody případem užití neboli různé toky událostí, které jsou popsány různými scénáři. Prostřední přímá čára je zobrazením pro úspěšný scénář, který reprezentuje základní tok událostí. Úspěšný scénář vede k dosažení cíle případu užití.
- Scénář z obrázku s označením „nesprávné zadání“ odpovídá chybně zadaným vstupním datům. V takovém případě je nutné se vrátit zpět a zadat údaje znovu. Pokud dojde k chybě či výjimce, tak skončí scénář dříve a v takovém případě není splněn cíl případu užití. Jednotlivé případy užití jsou množinou možných scénářů, které jsou popsány ve specifikaci daných případů užití. Tento popis by měl být sepsán bez používání detailní technické terminologie, jelikož je určen pro diskusi nad požadavky na systém i s uživateli.
- Doporučená struktura slovního popisu případu užití obsahuje:
 - Identifikace - jednoznačné označení případu užití,
 - Název případu užití,
 - Cíl – určení byznys cíle případu užití,
 - Primární aktér (aktéři) - pomocí případu užití plní cíl.
 - Pomocný aktér (aktéři) - poskytuje službu pro splnění úkolu primárního aktéra.

- Vstupní podmínky - musí být splněny, pokud má případ užití začít.
- Výstupní podmínky - definují stav systému po ukončení případu užití.
- Scénáře případu užití - hlavní scénář - může být rozdělen na dílčí alternativní scénáře, které popisují výjimečné situace.
- Dalším prvkem diagramu je vztah, který vyjadřuje tok informace mezi vnějším aktérem a případem užití. Vztah bývá označován také jako komunikační asociace a na diagramu je zobrazen čarou mezi případem užití a aktérem. Zobrazená čára může naznačovat, která strana iniciuje interakci, a to zakončením pomocí šipky. Uvádět šipku není povinné a v takovém případě může iniciovat interakci libovolná strana. Obrázek zobrazuje příklad aktéra, případů užití a zobrazenými vztahy mezi zmíněnými prvky,
- Vztahy mohou nabývat různých typů, které jsou známé pod názvy „include“, „extend“ a „generalizace“ či „specializace“.
- Vztah „include“ je určen pro znázornění vloženého případu užití do jiného případu užití. Použití nalezne zejména u opakující se činnosti, která je vyjmuta a vytvořena jako nový případ užití navázaný na původní případ užití pomocí vazby „include“. Vložený případ se provede vždy. Obrázek zobrazuje použití vztahu „include“



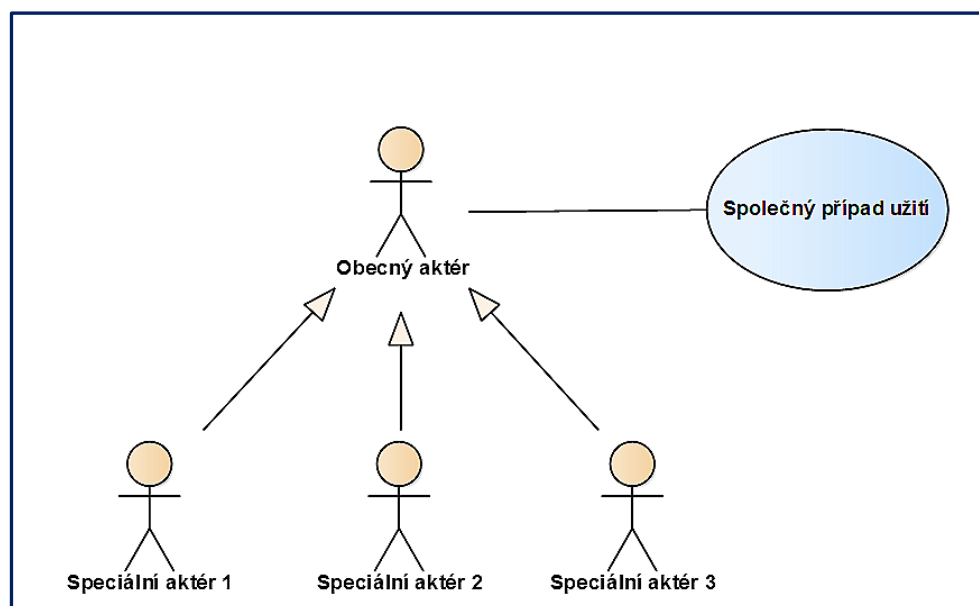
Obrázek 3-9: Příklad vloženého případu užití

- Vztah „extend“ je určen pro znázornění rozšíření chování základního případu užití. K rozšíření dochází pouze v případě, že je splněna podmínka rozšíření a pro vztah „extend“ dále platí, že základní případ může existovat sám o sobě (na rozdíl od vztahu „include“). Použití vztahu „extend“ nalezne uplatnění zejména v případech, ve kterých je zapotřebí vytknout podmíněné nebo výjimečné chování anebo je zapotřebí přidat chování, které rozšiřuje původní funkcionality, například v případech, kdy bude rozšiřující chování přidáno až v další verzi systému. Obrázek zobrazuje použití vztahu „extend“,



Obrázek 3-10: Příklad rozšířeného chování případu užití

- Vztah „generalizace“ či „specializace“ je určena pro zachycení obecného a speciálního chování. Tento vztah není určen jen pro případy užití, ale může být používán i pro aktéry. Vztah může způsobit nepřehlednost v modelu, a proto se jeho používání spíše nedoporučuje. Uplatnění vztahu nalezne zejména jako zjednodušení pro aktéry v případě, že množina aktérů vykonává jeden případ užití. V takovém případě je vhodné vytvořit společného předka pro jednotlivé aktéry a tohoto předka napojit na daný případ užití. Může se jednat například o přihlášení do aplikace. Obrázek obsahuje použití vztahu „generalizace“ u aktérů, kteří využívají společný případ užití,

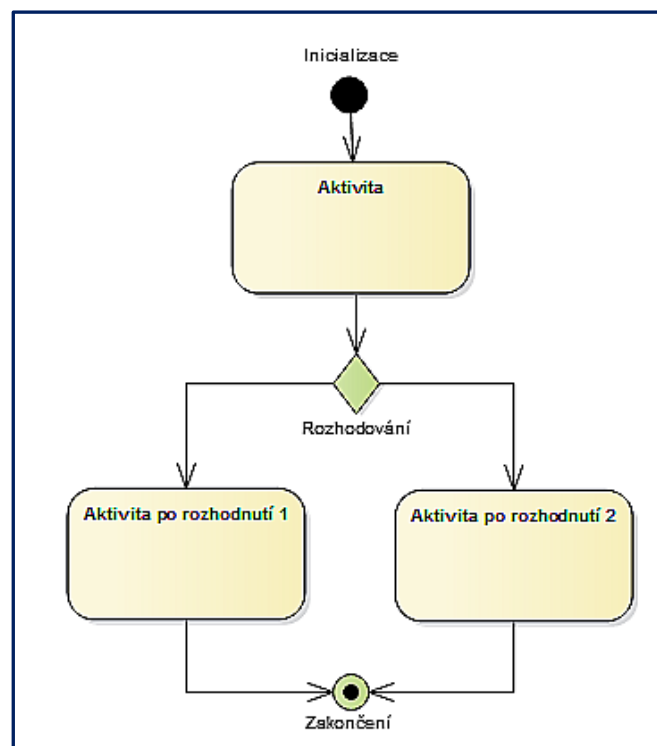


Obrázek 3-11: Příklad obecného aktéra se společným případem užití

- Modelování případů užití představuje techniku analýzy, která pomáhá identifikovat a vyjasňovat jednotlivé požadavky na systém. Modelování bývá poměrně pracné a v průběhu procesu vývoje dochází k upřesňování podoby modelu.

3.6.4 Diagramy aktivit

- Diagram aktivit je určen pro zachycení aktivit se sekvenčním i paralelním průběhem. Modelování pomocí diagramu aktivit je možné využít pro zachycení logiky scénáře případu užití, zachycení byznys procesů a zachycení logiky byznys pravidel. Mezi prvky diagramu aktivit patří inicializace (zahájení), aktivita, tok, rozhodování, větvení, spojení, plavecké dráhy a ukončení.
- Aktivita je modelována jako obdélník s uvedením názvu aktivity, který by měl být vyjádřen slovesnou vazbou, např. přihlásit se do aplikace. Aktivitu není možné dále dělit a nelze ji přerušit. Jednotlivé aktivity mohou mít pouze jeden vstupní bod a jeden výstupní tok. Pro zobrazení toku je určena čára se šipkou.
- Prvek rozhodování je určen pro přerušení lineárního provádění aktivit. Vstupem je pro rozhodování jeden tok a výstupní počet toků není omezen, ovšem každý výstupní tok musí mít definovanou výstupní podmínku, které musí být definovány tak, aby bylo možné splnit pouze jednu.
- Paralelní zpracování je možné zachytit pomocí prvku rozvětvení a spojení. Prvek rozvětvení má jeden vstupní tok a několik výstupních toků, které probíhají paralelně. Když dojde k rozvětvení, tak běží toky paralelně bez ohledu na pořadí, které není podstatné. Po dokončení paralelního zpracování dojde ke spojení běžících toků pomocí prvku pro spojení.
- V diagramu aktivit je dále možné využít „plavecké dráhy“ k zachycení toho, kdo danou aktivitu provádí. Aktivity se pro tento účel seskupí do svislých pruhů dle nositele aktivity, který danou aktivitu provádí, např. dle osoby či oddělení.
- Obrázek obsahuje příklad s prvky inicializace, aktivity, rozhodování a zakončení.



Obrázek 3-12: Prvky pro diagram aktivit

3.7 Metodika CRISP–DM

3.7.1 Celková charakteristika

Cross-Industry Standard Process for Data mining (CRISP-DM) je metodika pokrývající kompletní proces data miningových úloh (Rauch, Šimůnek 2014, str. 19). Metodika, která je při použití nezávislá na konkrétním odvětví společnosti zkoumající data, na použitých softwarových nástrojích a také na aplikaci konkrétní metody či algoritmu v oblasti data miningu. (Chapman et al. 1999, 2000, str. 1)

Chapmanovo CRISP-DM představuje standardizovanou a volně dostupnou formu vhodného přístupu k řešení data miningových problémů definovaných v rámci obchodních či výzkumných oddělení společnosti. Dle CRISP-DM sestává každý data miningový projekt z šesti vzájemně navazujících fází. Posloupnost těchto fází je adaptivní, což znamená, že následující fáze je obvykle velmi silně ovlivněna výstupy vytvořenými v rámci fáze předchozí.

CRISP-DM je znázorněna šipkami (viz základní slide) se zpětnou orientací, kdy je možné se po zpracování jedné fáze vrátit k fázi předchozí a provést potřebné, z následné fáze plynoucí, změny.

Vnější kruh v rámci diagramu symbolizuje cyklický charakter celého procesu, u kterého není neobvyklé, že po vyhodnocení výstupů úlohy (stejně tak i po využití získaných znalostí) je vhodné se vrátit na začátek celého procesu a překonfigurovat vstupní předpoklady tak, aby došlo k co možná největšímu zlepšení celého procesu a jeho výstupů. Jednotlivé fáze procesu dobývání znalostí z databází dle metodiky CRISP-DM jsou v dalším přehledu.

3.7.2 Porozumění problematice/zkoumané oblasti

V rámci této fáze je kladen důraz na porozumění požadavkům a cílům projektu z manažerského pohledu, dále probíhá hodnocení rizik a přínosů, kalkulace potřebných zdrojů a nákladů. Je také stanoven předběžný plán průběhu prací. Je také nutné porozumět zkoumané oblasti, ze které pocházejí zkoumaná data, v rámci které je jejich analýza prováděna.

Data miningové úlohy je možné provádět v rámci různých doménových oblastí, které se mohou v mnohém lišit, jako jsou telekomunikace, marketing, strojírenství, doprava atd. Je třeba disponovat znalostí specifik zkoumaného odvětví k rozlišení mezi novými zajímavými a již známými znalostmi plynoucími z datové analýzy.

3.7.3 Porozumění datům

Autoři metodiky uvádějí v její dokumentaci několik kroků, které je nutné s daty v rámci této fáze provést: (Chapman et al. 1999, 2000, str. 18)

- **Sběr vstupních dat** – v tomto kroku je nutné získat data nebo alespoň přístup k nim ze zdrojů definovaných v rámci projektu. Výstupem tohoto kroku je seznam získaných datových sad a metod, pomocí kterých je možné data získat, a seznam problémů, které se v průběhu sběru dat vyskytly.
- **Popis dat** – popisem získaných dat je myšleno prozkoumání jejich hrubých a povrchových vlastností jako jsou datové typy, počty záznamů jednotlivých datových sad, významy jednotlivých polí apod. Důležitou částí tohoto kroku je zhodnocení možnosti využití získaných dat v souladu s relevantními požadavky projektu.
- **Zkoumání dat** – ke zkoumání dat jsou nejčastěji využívány deskriptivní a vizualizační techniky, v rámci kterých jsou data zobrazována a zkoumána. Zkoumána je například frekvence výskytu hodnot v rámci jednotlivých atributů, průměry, mediány, maximální a minimální hodnoty atd.
- **Ověření kvality dat** – tento krok je zaměřený na zjištění kvality získaných dat. Jeho cílem je odpovědět na otázky typu: Pokrývají data kompletně zkoumanou oblast v potřebné míře? Jsou data správná, nebo obsahují chyby? V jakém jsou chyby zastoupení? Obsahují data

chybějící hodnoty? V jaké míře a kde se vyskytují, jak jsou reprezentovány chybějící hodnoty? Jestliže se v tomto kroku projeví datové chyby, je doporučeno vytvořit seznam jejich možných řešení. Tato řešení jsou silně závislá na zkoumaných datech a obchodních znalostích.

- **Příprava dat** – Berka popisuje tuto fázi následujícím způsobem: „Příprava dat zahrnuje selekci dat, čištění dat, transformaci dat, vytváření dat, integrování dat a formátování dat.“ (Berka 2003, str. 26). V této fázi probíhá výběr konkrétních atributů, ale i jednotlivých záznamů použitých v analytické úloze. Mezi kritéria výběru těchto dat patří relevance ve vztahu k cílům data miningové úlohy, kvalita a možná technická omezení využitých nástrojů jako je množství a datové typy vybraných dat. V rámci čištění dat je snaha o získání takové kvality dat, která je vyžadována vybranou analytickou metodou. Může se jednat o vložení vhodných výchozích hodnot nebo nahrazení chybějících hodnot adekvátní metodou apod. Vytváření dat zahrnuje operaci s daty vytvářející odvozené atributy, celé nové záznamy nebo transformované hodnoty již existujících atributů. Odvozenými atributy jsou myšleny atributy, které jsou vytvořeny kombinací některých, v datech, již existujících atributů v rámci jednotlivých záznamů. Příkladem může být odvozený atribut ‚marže‘, který odpovídá rozdílu atributů ‚tržby‘ a ‚náklady‘. Pokud jsou různá data týkající se konkrétního objektu uložena ve více tabulkách či datových sadách, jsou v rámci kroku integrace dat kombinována tak, aby dohromady tvořila nový záznam složený kombinací relevantních dat týkajících se tohoto objektu. V rámci této integrace může být provedena i agregace dat, pokud není nutné pracovat s detailními záznamy. Příkladem takové integrace může být spojení několika tabulek obsahujících údaje o zákazníkovi, kdy v jedné tabulce jsou osobní údaje zákazníka a ve druhé údaje demografické. Spojením těchto tabulek vznikne nová tabulka obsahující osobní i demografické údaje v rámci jednoho záznamu. Pod agregací dat je možné si představit situaci, kdy není třeba zkoumat jednotlivé nákupy konkrétního zákazníka, ale zkoumaným atributem může být například počet provedených nákupů zákazníka v určitém období. Formátováním dat jsou myšleny operace s daty, které data upraví v syntaktickém smyslu. Tyto úkony jsou prováděny v rámci požadavků využitých nástrojů k následnému modelování dat. Požadavkem takového nástroje může být například pozice zkoumaného atributu v rámci datové sady, kdy může být vyžadováno umístění tohoto atributu na poslední pozici v datové sadě, nebo nutnost označení každého záznamu unikátním identifikátorem apod. Data mohou být také transformována k účelu splnění specifických požadavků jednotlivých algoritmů data miningu. Může se jednat například o diskretizaci (rozdělení numerických dat do intervalů) či binarizaci (převod kategoričkových dat na data numerická). Výstupem této fáze je upravená datová sada vyhovující požadavkům konkrétního data miningového řešení a využitým algoritmům zpracovávajícím data.

3.7.4 Modelování

Larose popisuje tuto fázi pomocí čtyř stručných bodů, které jsou: (Larose c2005, str. 7):

- Výběr a aplikace vhodných modelovacích technik.
- Kalibrace parametrů vybraného modelu (data mining algoritmu) za účelem jeho optimálního nastavení a získání relevantních výsledků.
- Mít na paměti, že často k řešení jednoho data miningového problému je možné využít několik rozdílných technik a modelů. Obecně se doporučuje využít více různých technik a jejich výsledky kombinovat.
- Z předchozích kroků je možné, že vyplyne potřeba vrátit se zpět k fázi přípravy dat a jejich modifikaci tak, aby bylo pracováno s co nejvhodnějšími daty přizpůsobenými konkrétní zvolené data miningové technice.#SEC2#Vyhodnocení výsledků;
- V této fázi dochází k vyhodnocení získaných znalostí, které jsou relevantní z pohledu data miningových metod. Tyto znalosti je však nutné vyhodnotit z pohledu manažerů, pro které jsou získané znalosti určeny a kteří určí, zda došlo k naplnění cílů úlohy definovaných v jejím zadání.

V rámci evaluace výsledků klasifikační úlohy je možné zjistit například následující: „Výsledky testování klasifikačních znalostí ukázaly, že systém byl příliš přísný, tedy rozpoznával klienty rizikové, ale v určitých případech (obzvláště u vyšších půjček) za rizikové označil i klienty bonitní. Bylo tedy rozhodnuto, že ve všech pobočkách banky bude využíván program, který bude rozhodovat o úvěrech do určité částky.“ (Berka 2003, str. 27). Nakonec dochází k určujícímu rozhodnutí o způsobu využití získaných výsledků.

3.7.5 Využití výsledků

V poslední fázi dochází k sumarizaci využitelných výsledků úlohy a k definici způsobu využití výsledků jednotlivými koncovými uživateli. Výstupy data miningové úlohy mohou být ve formě závěrečné zprávy o dosažených výsledcích nebo ve formě softwarového řešení automatizujícího proces celé data miningové úlohy v uživatelsky využitelné podobě. Nejdůležitější fáze je porozumění problému, která zabere 20 % času, ale má 80% význam. Časově nejnáročnější je fáze přípravy dat, která zabírá 80 % času s 20% významem, přičemž modelování a analýza zkoumaných dat zabere 5 % času a má 5% význam. (Berka 2003, str. 28).

3.7.6 Poznámky, reference;

- BERKA, Petr. Dobývání znalostí z databází. Praha: Academia, 2003. ISBN 80-200-1062-9.,
- RAUCH, Jan a Milan ŠIMŮNEK. Dobývání znalostí z databází, LISp-Miner a GUHA. Praha: Oeconomica, nakladatelství VŠE, 2014. Odborná kniha s vědeckou redakcí. ISBN 9788024520339.,
- SKALSKÁ, Hana. Data mining a klasifikační modely. Hradec Králové: Gaudeamus, 2010. Recenzované monografie. ISBN 9788074350887.
- CHAPMAN, Pete, et al. CRISP-DM 1.0: Step-by-step data mining guide. In: The Modeling Agency [online]. Pittsburgh: One Oxford Centre, Copyright © 1999, 2000 [cit. 2017-11-21]. Dostupné z: <https://www.the-modeling-agency.com/crisp-dm.pdf>,
- LAROSE, Daniel T. Discovering knowledge in data: an introduction to data mining. Hoboken, N.J.: Wiley-Interscience, c2005. ISBN 9780471666578.

3.8 Datové modelování

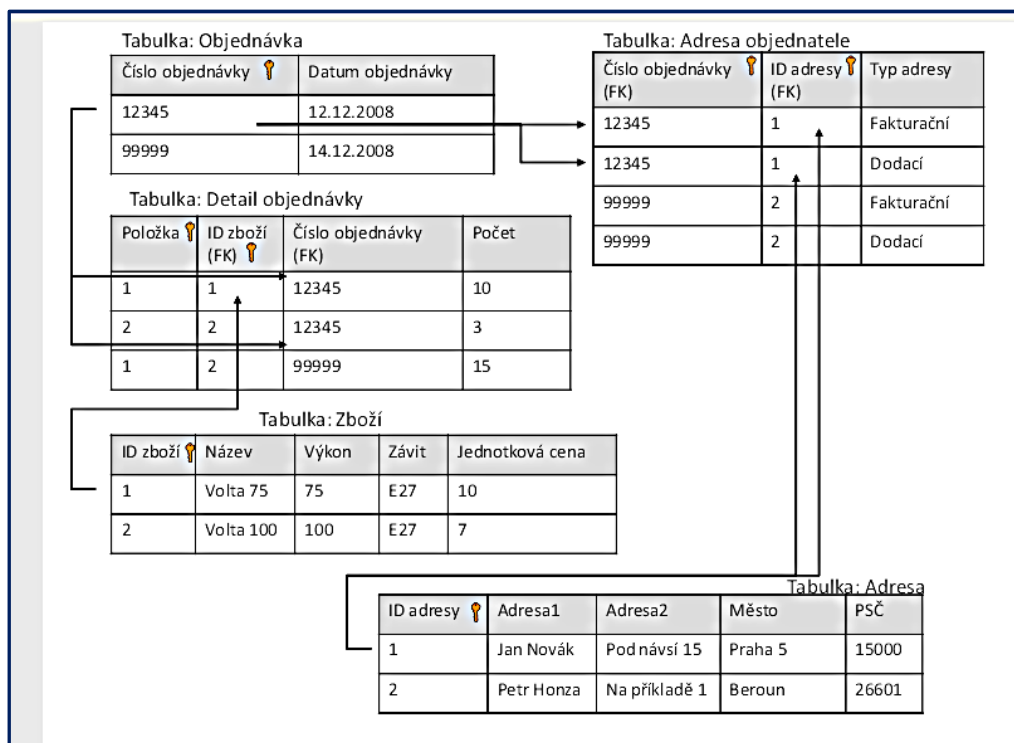
3.8.1 Účel datového modelování

- Datové modelování je **metoda**, která slouží k **navrhování struktury dat v databázích** a k jejich dokumentaci. **Účelem** datového modelování je přispívat **ke kvalitnímu návrhu databází, jak transakčních, zdrojových, tak analytických**. To znamená, že:
- databáze jsou **konstruovány efektivně s minimalizací redundancí** dat (tj. jejich duplicit, či multiplicit), se zajištěním požadovaného výkonu a doby odezvy při nejčastějších dotazech a dalších operacích,
- databáze se navrhují tak, **aby poskytovaly data uživatelům, která odpovídají jejich aktuálním požadavkům**, ale i požadavkům, které lze v budoucnu očekávat. To znamená tak, aby databáze bylo možné relativně bezpečně rozšiřovat, udržovat a modifikovat. Úpravy databáze nesmí znamenat nepřiměřená rizika pro kvalitu dat,
- v relační databázi je **řešena integrita na úrovni položek (atributů), tabulek i vazeb mezi tabulkami** zajišťující přesnost a konsistenci uložených dat po celou dobu provozu,
- **aplikace postavené na databázi lze pak řešit efektivně** a návrh databáze neznamena významné omezení pro jejich další údržbu a rozvoj,

- **efekty a problémy** datového modelování shrnuje faktor "Využití datového modelování" (F059).

3.8.2 Relační databáze

Datové modelování je spojené obvykle s **relačními databázemi**. Je účelné v tomto kontextu **zrekapitulovat** několik základních termínů.



Obrázek 3-13: Tabulky a jejich vazby v relační databázi

3.8.2.1 Relace

Relace - dvourozměrná datová struktura představovaná **tabulkou**, kde jeden řádek tabulky odpovídá jednomu záznamu (např. jednomu záznamu zboží) a sloupec odpovídá položce (např. *ID zboží*, *Název*, *Jednotková cena* atd.).

V databázovém pojetí se **sloupec tabulky** označuje jako **atribut**, jehož hodnoty spadají do tzv. **domény**.

3.8.2.2 Integrita

Integrita, což je pravidlo omezující možné hodnoty atributů, případně možné manipulace se záznamy, které existují ve vazbě k záznamům jiné tabulky.

K významným integritním omezením patří **existence primárního klíče** (tzv. **entitní integrita**). Tím se zajišťuje **jednoznačná identifikace záznamu** v tabulce (žádné dva záznamy nemohou mít stejnou hodnotu primárního klíče). S termínem primární klíč souvisí i **umělý primární klíč (surrogate key)**, nemající nějaký specifický podnikový význam (např. *ID zboží*), ale vzniká postupným načítáním stanovené hodnoty (většinou 1) při přidávání dalších záznamů do tabulky databáze,

Další typ integritního omezení je **doménová integrita**, která zajišťuje, aby údaj uvedený jako hodnota atributu byl vybrán z množiny definovaných přípustných hodnot. Například pro sloupec (doménu) *Typ adresy* v tabulce *Adresa objednatele* může být takovým doménovým omezením možnost uvedení buďto hodnoty *Fakturační* nebo hodnoty *Dodací*.

Posledním typem integritního omezení je tzv. **referenční integrita**, která je **reprezentována cizím klíčem (foreign key)**. Cizí klíč zajišťuje provázanost tabulek, např. mezi tabulkou *Zboží* a *Detail*

objednávky je cizím klíčem *ID zboží* umístěný v tabulce *Detail objednávky*. Na obrázku (Obrázek 3-13) je položka, která je cizím klíčem označena zkratkou FK. **Princip referenční integrity zajišťuje:**

- **aby nemohla být vložena jako hodnota atributu**, který je označen jako **cizí klíč**, hodnota, jež nemá odpovídající hodnotu atributu v tabulce s primárním klíčem. Tak je např. zajištěn vztah mezi tabulkou *Objednávka* a *Detail objednávky*, kdy platí, že v tabulce *Detail objednávky* se jako hodnota atributu *Číslo objednávky* nesmí objevit jiné číslo než to, které je uvedeno u některého ze záznamů v tabulce *Objednávka*,
- aby z tabulky **nebylo možno odstranit záznam**, jehož hodnota je uvedena jako cizí klíč v tabulce jiné. To např. znamená, že nelze odstranit záznam z tabulky *Zboží*, jehož ID zboží je rovno 1, protože se tato hodnota vyskytuje jako hodnota cizího klíče v tabulce *Detail objednávky*.

3.8.3 Řešení datových modelů

Uvedené základní **principy relačních databází** se promítají do řešení datových modelů. Návrh datových modelů se realizuje na několika úrovních, resp. v několika základních krocích. Tyto kroky následují obvykle od konceptuálního návrhu, přes logický k fyzickému:

- **Konceptuální návrh** databáze - prvotní návrh databáze - její **struktury, vazeb a všech potřebných charakteristik**
 - je zaměřen na **věcnou podstatu** dat a nezohledňuje fyzické, resp. technologické charakteristiky databáze, ale pouze její požadovanou strukturu a logiku uspořádání dat, které jsou **nezávislé na konkrétním softwaru** pro řízení databáze,
 - **účelem** konceptuálního návrhu databáze je specifikovat typy datových objektů (**typy entit**), jejich vlastnosti (**atributy**), přirozenou **identifikaci**, **vazby** mezi entitami a jejich vlastnosti,
 - - v entitách **se nemodelují cizí klíče**, vazby mezi entitami jsou vyjádřeny pouze graficky.
- **Logický návrh** databáze - na základě konceptuálního návrhu databáze se řeší logický návrh databáze:
 - **cílem** je *detailně vymezit strukturu* databáze založené již na určitém logickém schématu dat, tj. v našem případě relačním,
 - především definovat **struktury databázových tabulek**, jejich atributy, primární klíče a relace mezi tabulkami prostřednictvím cizích klíčů,
- Vytvoření, případně generování **fyzického návrhu** databáze - na základě logického návrhu databáze se definuje fyzický datový model:
 - specifikuje již potřebné **implementační charakteristiky** ve vztahu k příslušnému **databázovému prostředí**, v němž bude databáze realizována,
 - např. **formáty jednotlivých datových položek**, jejich rozsah, resp. délka, definování cizích klíčů a vztahů mezi tabulkami a další charakteristiky.

3.8.4 Podstatné charakteristiky datových modelů

Entita:

Entita představuje z pohledu konceptuálního datového modelu dále **nedělitelnou jednotku reálného světa**, o které se budou v databázi vést příslušná data:

- v databázi entitě obvykle odpovídá **jeden řádek** v databázové tabulce,
- **typ entity** pak představuje zobecnění pro všechny entity se stejnými vlastnostmi (např. typem entity je *Zákazník* a entitou jeden každý zákazník),
- typu entity odpovídá v databázi jedna **databázová tabulka**,

- typ entity má **vlastní název**, **identifikátor** a její **obsah** je dále definován skupinou atributů, které jsou následně, v rámci logického návrhu databáze, základem pro definování struktury databázové tabulky,
- v rámci logického návrhu databáze je základem pro **definování struktury** databázové tabulky,
- **Poznámka:** Entita a Typ entity se oba běžně používají pro termín „typ entity“, Výskyt pak pro jednu konkrétní entitu, tj. data o konkrétním sledovaném objektu.

Vazby:

Vazby - se definují mezi dvěma entitami použitím symbolu vazby s doplněním názvu vazby a dalších jejích charakteristik, včetně kardinality.

Kardinalita vazby - násobnost a povinnost znamená vyjádření **možnosti výskytu počtu entit** na obou stranách vazby:

- existují základní vztahy 1:1, 1:M, M:N,
- pro vazbu **1 : M, (One to Many)**, to znamená, že na příklad jeden obchodník odpovídá za více objednávek zboží a jedna objednávka zboží je vyřizována pouze jedním obchodníkem,
- vazba **M : N (Many to Many)** to bude znamenat, že jeden obchodník vyřizuje více objednávek, ale na druhé straně na jedné objednávce se může podílet více obchodníků,

Povinnost vazby vyjadřuje **nutnost existence** dané vazby ke každému výskytu entity, například obchodník nemusí být odpovědný za žádnou objednávku, ale objednávka musí mít obchodníka, který je za ní odpovědný.

3.8.5 Normalizace databáze

Normalizace databáze se chápe jako postup, kdy je **struktura dat** v relační databázi **přeorganizována** tak, aby byly **minimalizovány duplicity nebo multiplicity** uložení dat v databázi tak, aby ideálně jedna změna v realitě vyžadovala pouze jednu změnu v datech v databázi.

3.8.6 Poznámky, reference

- GÁLA, L., POUR, J., ŠEDIVÁ, Z.: *Podniková informatika*. Praha, Grada, 2015. ISBN 978-80-247-5457-4
- LABERGER, R.: *The Datawarehouse Mentor*. New York, McGraw Hill, 2011. ISBN: 978-0-07-174532-1

3.9 Procesní modelování

3.9.1 Procesní modelování: celková charakteristika

- Řízení podnikových procesů (**BPM, Business Process Management**) zahrnuje aktivity spojené s optimalizací podnikových procesů a jejich maximálním přizpůsobením potřebám podniku.
- **Jádrem BPM** je řešení změn, rozvoje a zavádění nových podnikových procesů ve vazbě na stanovenou podnikovou architekturu, což představuje komplex úloh (projektů) reengineeringu procesů (**BPR, Business Process Reengineering**). Tyto projekty zahrnují formalizované vyjádření, resp. modelování podnikových procesů a další optimalizační kroky.

3.9.2 Procesní charakteristiky

- Efektivita a výkonnost podnikového řízení **závisí na úrovni takových procesních charakteristik**, jako jsou:

- vlastní **výkonnost** procesů, tj. jejich časová a finanční náročnost, např. u procesů řízení zakázek, řízení vztahů k zákazníkům atd.,
- **flexibilita** procesu, tedy schopnost reakce na změněné ekonomické prostředí, na změněné legislativní podmínky, nové požadavky a změny v technologiích. To znamená, že procesy řízení musí být nastaveny tak, aby poskytovaly prostor pro uplatnění uvedených změn a vlivů. Např. procesy pro plánování a řízení zakázek musí počítat s výkyvy v technických i personálních kapacitách, s různou dostupností materiálových zdrojů a finančních prostředků,
- výkonnost podniku je z procesního hlediska ovlivňována jejich **úrovní zralosti**, tj. úrovní jejich dokumentace, formalizace či optimalizace.

3.9.3 Řízení podnikových procesů

- Řízení podnikových procesů **zahrnuje celou škálu metod a přístupů s cílem zvýšit efektivnost a výkonnost firmy**, v tomto případě především **procesní výkonnost**.
- **Rychlost, s jakou podnik schopen reagovat na požadavky** zákazníka je, vedle ceny, kvality, inovativnosti, jeden z rozhodujících faktorů jeho konkurenceschopnosti.

3.9.4 Monitorování podnikových procesů

- S uvedenými charakteristikami souvisí **potřeba sledovat průběh podnikových procesů** a jejich činnosti **průběžně měřit**, tj. jejich časové hodnoty a další výkonové parametry (objem produktů, chybovost apod.), např. sledované časy na vyhodnocení objednávky zákazníka, časy naplánování zakázky, časy přípravy a kompletace zakázky atd.
- To je předmětem **metody monitorování podnikových procesů (BAM, Business Activity Monitoring)**.
- BAM je postavena **na celé řadě úloh a principů**, k nimž patří:
 - **stanovení cílů** BAM a výběr procesů, které budou předmětem jeho nasazení,
 - správné **vymezení míst** v procesu, tj. klíčových aktivit, resp. uzlových bodů, kde je měření časových a výkonových charakteristik nejúčinnější,
 - stanovení **měřených charakteristik**, včetně jejich obsahového a formálního vymezení, měrných jednotek apod.,
 - stanovení **periodicity měření** (každou hodinu, denně, náhodně apod.),
 - stanovení **zodpovědných pracovníků** za měření hodnot,
 - stanovení **funkcionality** analýz realizovaných se získanými, resp. naměřenými hodnotami,
 - určení způsobu **využití** měřených hodnot, tj. v rámci BPR, případně organizačních a technických opatření pro zlepšení sledovaných procesů.
- Aplikace BAM využívají **specifických technologií zařazovaných do middleware** (F836), které slouží k řízení a dohledu nad událostmi. Tyto technologie **aplikacím BAM poskytují vhodná data** o událostech v podniku, např. elektronicky došlá objednávka, přerušení funkce výrobní linky apod.

3.9.5 Business Process Reengineering, BPR

Zásadní změny, resp. **vytvoření zcela nových procesů** (Business Process Reengineering, BPR) vychází ve větší či menší míře **z předpokladu, že současné podnikové procesy již nevyhovují** situaci na trhu a potřebám podniku a je nezbytné je úplně přehodnotit a změnit.

V tomto případě se projekty BPR realizují v následujících krocích, či etapách:

- **definice rozsahu** BPR projektu, tj. jaké oblasti podnikového řízení má pokrývat, jaké organizační jednotky, např. včetně dislokovaných závodů a poboček, jaké mají procesy postavení ve firmě, např. pouze základní (core) procesy, nebo i podpůrné a řídicí procesy,
- **analýza potřeb** a možností podniku vzhledem k procesům, jaké nároky jsou na ně kladeny z hlediska např. vztahů k zákazníkům, ke kooperacím s partnery, jaké jsou potřeby podnikového managementu na procesy zahrnující podnikové analýzy, plánování, operativní rozhodování atd.,
- **vytvoření zcela nové soustavy procesů**, procesního modelu, tj. všech vymezených procesů a jejich vzájemných vazeb (jejich vstupů a výstupů) a definování vnitřní logiky procesů, tj. sítí činností a dalších charakteristik,
- **plánování přechodu** na nový procesní model, vytvoření harmonogramu přechodu, určení zodpovědností a kompetencí pro pracovníky podniku,
- **implementace nového procesního modelu**, tj. verifikace nové procesní dokumentace, promítnutí do organizačních norem a předpisů, případně úpravy, nebo nasazení nových podpůrných technologií (workflow) a vyškolení příslušných pracovníků.

3.9.6 Poznámky, reference

- ŘEPA, V.: *Podnikové procesy*. Praha, Grada 2007.
- GÁLA, L., POUR, J., ŠEDIVÁ, Z.: *Podniková informatika*. Praha, Grada, 2015. ISBN 978-80-247-5457-4

3.10 Dimenzionální modelování

3.10.1 Dimenzionální modelování, celková charakteristika

Dimenzionální modelování **vychází z poznání a zhodnocení potřeb řízení** dané organizace. Jeho **obsahem je**:

- **vymezení všech dimenzí**, jejich obsahu, včetně vnitřní hierarchie prvků, a dílčích charakteristik jednotlivých dimenzí,
- určení **soustavy sledovaných ukazatelů** (faktů) a jejich dílčích charakteristik,
- **specifikace vazeb** mezi ukazateli a odpovídajícími dimenzemi.

3.10.2 Dimenzionální modelování, hrubý dimenzionální model

- Určení všech analytických **dimenzí**,
- Určení soustavy sledovaných **ukazatelů** (faktů) a jejich dílčích charakteristik,
- Specifikace **vazeb** mezi ukazateli a odpovídajícími dimenzemi.

3.10.3 Dimenze a jejich charakteristiky

Vymezení všech dimenzí, jejich **obsahu a dílčích charakteristik**, zejména:

- identifikace dimenze – např. „D_Zboží“,
- plný název dimenze, např. „Zboží nabízené a prodávané podnikem“,
- obsah dimenze – detailní textový popis obsahu dimenze,
- typ dimenze: časová dimenze, SNOWFLAKE, STAR, degenerovaná,
- hierarchie dimenze - hierarchické úrovně dimenze,
- zdroj dat pro dimenzi, resp. její prvky, tj. databázová tabulka, textový soubor apod. s příslušnou standardní identifikací datového zdroje,
- popis dimenze, poznámky k vytvoření a využití dimenze.

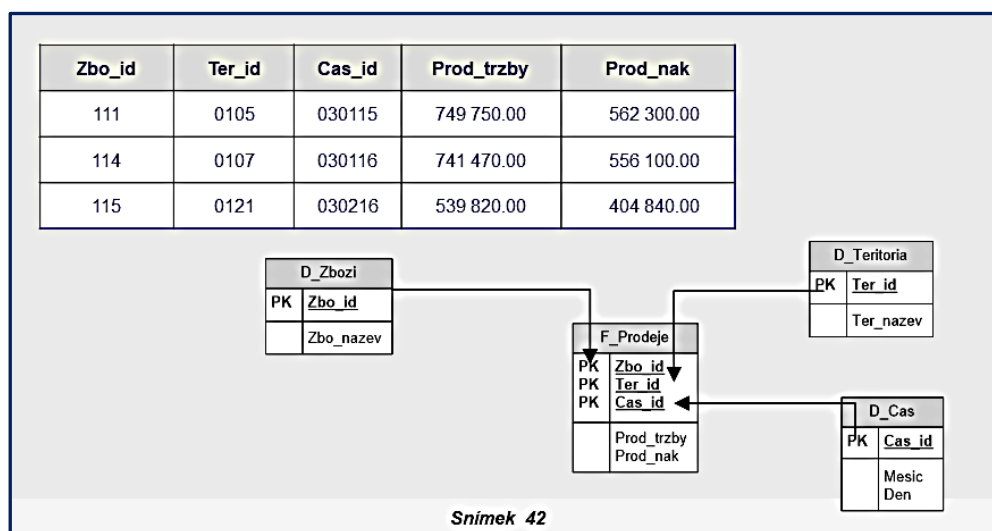
3.10.4 Ukazatelé a jejich charakteristiky

Vymezení všech sledovaných ukazatelů, jejich obsahu a dílčích charakteristik, zejména:

- identifikace ukazatele – např. „Prodej_Id“,
- symbolická identifikace ukazatele, např. „Prodej_objem“,
- plný název ukazatele, např. „Objem prodeje v Kč“,
- obsah – detailní obsahové vymezení ukazatele,
- zdroj dat pro zdrojové ukazatele, nebo výpočty pro ukazatele kalkulované,
- formát, např. Numeric (10,2),
- jednotka vyjádření ukazatele – Kč, kusy, % apod.,
- možnosti agregace ukazatele ve vztahu k dimenzím: aditivní, neaditivní, semiaditivní,
- KPI – zda má ukazatel charakter KPI, KGI, nebo žádný z nich,
- poznámky – např. určení tzv. analytických pravidel.

3.10.5 Návrh tabulek faktů

Tabulka faktů uchovává **hodnoty sledovaných ukazatelů** (Obrázek 3-14).



Obrázek 3-14: Tabulky faktů

- **Sloupce** jsou buď klíčové atributy (Zbo_id, Ter_id, Cas_id), nebo hodnoty ukazatelů (Prod_trzby, Prod_nak),
- **Klíčové atributy – dimenze**, jejich hodnoty - jsou prvky dimenzí,
- **Řádky** - jednotlivá měření (v obchodě, výrobě apod.), většinou přiřazovány na co nejnížší úroveň detailu, tj. pouze na úrovni listů ve strukturách použitých dimenzí,
- **Granularita** - úroveň podrobnosti údajů – faktů - je přímo závislá na počtu a úrovni podrobnosti dimenzí odpovídajících příslušné tabulce faktů,
- Pokud to technické kapacity dovolují, měla by být data uložena s nejvyšší možnou granularitou.
- Data vstupující z různých zdrojů je účelné transformovat na stejnou nebo srovnatelnou granularitu.
- Nedoporučuje se kombinovat fakta s různou granularitou do jedné faktové tabulky,

3.10.6 Návrh dimenzionálních tabulek

- De facto **podnikové číselníky** se všemi možnostmi a problémy (Obrázek 3-15)
- Klíčovým problémem je většinou **sjednocení číselníků**, resp. dimenzí v rámci celého podniku,
- Jedna **řádka tabulky** je vymezena pouze pro **jeden prvek dimenze s primárním klíčem**, podle něj by měla být tabulka i indexována,
- **Velký počet atributů** (50 – 100 atributů) – pro dokumentační a reportovací účely, úrovně agregací, podmínky v dotazech,
- Zahrnout všechny relevantní **popisné charakteristiky**,
- „**Property members**“, neboli vlastnosti prvků, což jsou takové atributy, resp. položky, které se nepodílejí na vymezení nebo hierarchii dimenze,
- **Kvalita datového skladu** dána kvalitou a hloubkou atributů dimenzí,
- Atributy vyjádřené v plných výrazech, **ne ve zkratkách**,

informatiky	 M504_16: Tabulky dimenzí – principy návrhu					
	Produkt_Id	Skupina_Id	Produkt_Skupina	Kateg_Id	Produkt_Kateg	Nazev
	111	11	notebook	1	počítač	Acer Travelmate 292
	114	12	PDA	1	počítač	Pocket LOOX 410
	116	21	monitor LCD	2	monitor	LG L1730S

Obrázek 3-15: Tabulky dimenzí – principy návrhu

Další principy a doporučení:

- V rámci jedné dimenze se mohou tvořit tzv. **alternativní struktury**, tj. např. pro různé organizační struktury v rámci jednoho podniku apod. - nejnižší úrovně hierarchie – listy obsahují sloupce pro identifikátory, resp. klíče pro více nadřazených struktur,
- **Degenerované dimenze**, tzn. že existuje dimenze pouze na základě příslušného atributu v tabulce faktů a nemusí pro ni existovat dimenzionální tabulka (např. číslo položky nákupu na POS),
- **Časová dimenze** – po dnech – se všemi podstatnými charakteristikami dní (pracovní den, svátek, ..) – podnikový kalendář,
- **Časová dimenze - hodiny** – jako zvláštní, kombinovat s denní, jednodušší manipulace,
- **Kombinované dimenze** - výhody – nižší nároky na prostor, efektivnější prohlížení a zpracování, nevýhody – nižší pochopitelnost a průhlednost řešení pro uživatele,
- Je-li sledovaná veličina měřitelná a měnící se v čase – pak patří do fakt tabulky, či zda je diskrétní a vystupuje spíše jako konstanta – pak jde o položku z dimenzionální tabulky,
- „**Parent-child**“ dimenze, což je např. dimenze zaměstnanců, kde na úrovni listu je vždy pouze jeden „zaměstnanec“, na vyšší úrovni je „manager“, který je většinou nadřazený pro několik zaměstnanců, ale ten se odkazuje zpět na jednoho konkrétního zaměstnance – managera,
- **Sběrná dimenze (Junk Dimension)** – smyslem je vytvořit společnou, sběrnou dimenzi (junk dimension), která bude v jednotlivých záznamech obsahovat kombinace příznaků, kde každý záznam s takovou kombinací bude mít svůj umělý klíč.

3.10.7 Poznámky, reference

- POTANČOK, M., POUR, J., CHRAMOSTOVÁ, V. Podniková analytika pro manažery, Oeconomia, Praha, 2021,

- POUR, J., MARYŠKA, M., STANOVSKÁ, I., ŠEDIVÁ, Z.: Self Service Business Intelligence, Praha. Grada, 2018. ISBN 978-80-271-0616-5
- LABERGER, R.: *The Datawarehouse Mentor*. New York, McGraw Hill, 2011. ISBN: 978-0-07-174532-1
- KIMBALL, R., ROSS, M.: Relentlessly Practical Tools for Data Warehousing and Business Intelligence. Indianapolis, John Wiley Publishing 2010. ISBN 978-0-470-56310-6

4. Závěry

Text tvoří jeden z textů řady II „IT a anatomie firmy“ v tomto případě zaměřený na specifikaci jednotlivých komponent řízení, které jsou využity v ostatních textech na úrovni AF II i AF III. Obdobně jako v ostatních textech **je cílem prezentovat obsah řízení firmy pohledem a potřebami analytika**, případně manažera nebo analytika vývojáře. Zatímco díl I se zaměřil na vymezení a **objasnění principů analýzy** orientované na firemní obsah, pak díl II prezentoval **základní obsah jednotlivých oblastí a komponent řízení** právě na základě zmíněných principů analýzy, ale bez zohlednění specifík jednotlivých odvětví.

Je zcela zřejmé, že **text dílu II** vytváří **základ** pro všechny navazující odvětvově orientované texty. V souvislosti s ostatními texty jsme uvedli, že smyslem uvedeného pojetí a přístupu k analýze je přispět ke **zvyšování kvality a výkonu** práce analytiků, manažerů a analytiků vývojářů v reálné praxi. V případě tohoto textu to platí nemalou měrou. Jestli i tento text takový příspěvek představuje, pak se jeho smysl podařilo naplnit.

5. Zdroje

- AALST, WIL van der. Process Mining Data Science v akci. Berlín : Springer, 2016. 978-3-662-49850-7.
- ANGEL, I.O., SMITHSON, S.: Information Systems Management - Opportunities and Risks, Macmillan, 1991.
- BACAL, R.: *Manager's Guide to Performance Management*. New York, McGraw-Hill 2012. ISBN 978-0-07-177225-9.
- BERKA, P. 2003. Dobývání znalostí z databází. Praha: Academia, 2003. str. 366. ISBN 80-200-1062-9,
- BRUCKNER, T. VOŘÍŠEK, J., BUCHALCEVOVÁ, A. a kolektiv: Tvorba informačních systémů: Principy, metodiky, architektury, Grada, 2012, ISBN 978-802477-9027.
- BUCHALCEVOVÁ, A.: Metodiky vývoje a údržby informačních systémů. Praha, Grada 2004. ISBN 80-247-1075-7.
- BUCHALCEVOVÁ, A., PAVLÍČKOVÁ, J., PAVLÍČEK, L.: Základy softwarového inženýrství, materiály ke cvičením. Praha, VŠE 2008.
- CAO, L.: Data Science Thinking. Springer, 2018. ISBN 978-3-319-95091-4.
- CIMLER, P., ZADRAŽILOVÁ, D. a kol.: Retail management. Praha, Management Press, 2007. ISBN: 978-80-7261-167-6.
- DOHNAL, J., POUR, J.: IT v řízení podniku, Praha, Professional publishing 2016. ISBN 978-80-7431-160-4.
- DOHNAL, J., PŘÍKLENK, O.: CIO a podpora byznysu. Praha, Grada, 2011. ISBN 978-80-247-4050-8.
- DOHNAL, J.: Řízení vztahů se zákazníky – procesy, pracovníci, technologie. Praha, Grada 2002.
- DRESNER, H.: Profiles in Performance. New York, John Wiley and Sons, 2010. ISBN: 978-0-470-40886-5.
- ECKERSON, W., W.: Performance Dashboards. New Jersey, John Wiley & Sons 2006.
- ENGLISH, L. P.: *Improving Data Warehouse and Business Information Quality: Methods for reducing costs and increasing profits*. New York, John Wiley & Sons 2003. ISBN 0-471-25383-9.
- FIBÍROVÁ, J., ŠOLJAKOVÁ, L., WAGNER, J., PETERA, P.: Manažerské účetnictví. Nástroje a metody. Praha, Wolters Kluwer, 2015. ISBN: 978-80-7478-743-0.
- HALAMA, J.: Řízení datové kvality v Hadoop Ecosystem, DP, VŠE, Praha, 2021.
- HOLTSNIDER, B., JAFFE, B.D.: IT Manager's Handbook. Amsterdam, Elsevier 2012. ISBN 978-0-12-415949-5.
- CHANDLER, N.: The CPM Scenario. Gartner BI Summit 2008.
- CHRAMOSTOVÁ, V., POTANČOK, M., POUR, J.: Byznys analytika pro manažery, Oeconomia, Praha, 2020.
- JUROVÁ, M. a kol.: Výrobní a logistické procesy v podnikání. Praha, Grada Publishing, 2016. ISBN 978-80-247-5717-9
- KEŘKOVSKÝ, M., VALSA, O.: Moderní přístupy k řízení výroby. 3 doplněné vydání. Praha, C.H. Beck pro praxi, 2012. ISBN 978-80-7179-319-9.
- [ISO certifikace – \(MANages, a.s., \[Online\] @2009. \[Citace: 26. září 2014.\] 2009\)](#),
- KIMBALL, R., ROSS, M.: Relentlessly Practical Tools for Data Warehousing and Business Intelligence. Indianapolis, John Wiley Publishing 2010. ISBN 978-0-470-56310-6.
- KOVÁŘ, M.: Informatická podpora výroby vozu, prezentace pro VŠE Praha. Škoda Auto, 2022.
- KOTTER, J.P. – Vedení procesu změny: osm kroků úspěšné transformace – (Management Press 2000) – ISBN 8072610155,
- KRÁL, B. a kol.: Manažerské účetnictví. Praha, Management Press 2006. ISBN 80-7261-141-0.

- KRÁL, Bohumil, Jana FIBÍROVÁ, Ondřej MATYÁŠ, Michal MENŠÍK, Jakub STRÁNSKÝ, Libuše ŠOLJAKOVÁ, Jaroslav WAGNER, Martin ZRALÝ a Zbyněk HALÍŘ, 2012. Manažerské účetnictví. 3. doplněné a aktualizované vydání. Praha: Management Press. ISBN 978-80-7261-217-8.
- KULHAVÝ, L. - Praktické uplatnění technologií Data Mining v pojišťovnictví - (VŠE, Praha 2011),
- KUNSTOVÁ, R.: Efektivní správa dokumentů. Praha, Grada, 2009. ISBN 978-80-247-3257-2.
- LABERGER, R.: The Datawarehouse Mentor. New York, McGraw Hill, 2011. ISBN: 978-0-07-174532-1.
- LABERGER, R.: Datové sklady. Agilní metody business intelligence. Praha, Computer Press, McGraw Hill, 2012. ISBN 978-80-251-3729-1.
- LANEY, D., B.: Infonomics, Bibliomotion, Inc., New York, 2018. ISBN 978-1-138-09038-5.
- MÁŠA, Petr. 2008. 4IZ560 - Data mining - praktické aplikace. 2008.
- NOVOTNÝ, O., POUR, J., BASL, J., MARYŠKA, M.: Řízení výkonnosti podnikové informatiky. Professional Publishing, Praha, 2010. ISBN 978-80-7431-040-9.
- NOVOTNÝ, O., VOŘÍŠEK, J. a kol.: Digitální cesta k prosperitě. Praha: Professional Publishing 2011. ISBN 978-80-7431-047-8.
- PALADINO, B.: Innovative Corporate Performance Management: Five Key Principles to Accelerate Results. Indianapolis, Wiley Publishing, 2011. ISBN: 978-0-470-62773-0.
- PARMENTER, D.: Key Performance Indicators (KPI): Developing, Implementing, and Using Winning KPIs,
- PETERKA, M.: Řízení výrobních firem, prezentace pro VŠE Praha. Seyfor, 2022.
- POUR, J., MARYŠKA, M., STANOVSKÁ, I., ŠEDIVÁ, Z.: Self Service Business Intelligence, Praha. Grada, 2018. ISBN 978-80-271-0616-5.
- PROVOST, F., FAWCETT, T.: Data Science for Business. What You Need to Know About Data Mining and Data-Analytic Thinking. O'Reilly Media. Sebastopol. 2013. ISBN: 978-1-449-36132-7.
- ŘEPA, V.: Podnikové procesy. Praha, Grada 2007.
- SCHIESSER, R.: IT Systems Management. New York, Prentice Hall 2010. ISBN 978-0-13-702506-0.
- SIEGEL, E: Predictive Analytics. New York, John Wiley & Sons, 2016. ISBN 978-1-119-14567-7.
- SLÁNSKÝ, D.: Data and Analytics for the 21st Century: Architecture and Governance, Professional Publishing, 2018. ISBN 978-80-88260-16-5.
- ŠEDA, J.: IOT A PRŮMYSL 4.0, prezentace pro VŠE Praha. Škoda Auto, 2022.
- ŠOLJAKOVÁ, L. FIBÍROVÁ, J.: Reporting. Praha, Grada, 2010. ISBN 978-80-247-2759-2.
- SYNEK, M. a kol.: Manažerská ekonomika. Praha, Grada 2011. ISBN 978-80-247-3494-1.
- SYNEK, M., KISLINGEROVÁ, E. a kol.: Podniková ekonomika. Praha, C H Beck 2015. ISBN 978-80-7400-274-8.
- TOGAF: Architecture Development Method – (The_Open_Group).
- ŠVECOVÁ, L., VEBER, J. Produkční a provozní management. Grada, 2021. ISBN 978-80-271-1385-9.
- TOMEK, G., VÁVROVÁ, V.: Průmysl 4.0 aneb nikdo sám nevyhraje. Praha, Professional Publishing, 2017. ISBN 978-80-906594-4-5.
- TOMEK, G., VÁVROVÁ, V.: Řízení výroby a nákupu. Praha, Grada 2007. ISBN 978-80-247-1479-0
- TOMEK, G., VÁVROVÁ, V.: Integrované řízení výroby. Praha, Grada 2014. ISBN 978-80-247-4486-5
- UČEŇ, P.: Zvyšování výkonnosti firmy na bázi potenciálu zlepšení. Praha, Grada 2008. ISBN: 978-80-247-2472-0.
- VAN DECKER, J, CHANDLER, N.: Top Processes for Corporate Performance Management, Gartner, 2011.

VEBER, J. a kol.: Management - základy, prosperita, globalizace. Praha, Management Press 2000.

VOŘÍŠEK, J., POUR, J. a kol.: Management podnikové informatiky, Professional Publishing, 2012, ISBN 978-80-7431-102-4.

VOŘÍŠEK, J. a kol.: Principy a modely řízení podnikové informatiky. Praha, Oeconomia 2008. ISBN: 978-80-245-1440-6.

ŽŮRKOVÁ, Hana, 2007. Plánování a kontrola - klíč k úspěchu. 1. vydání. Praha: Grada Publishing. ISBN 80-247-1844-8.