4IT421 – Zlepšování procesů budování informačních systémů

Přehled rolí v jednotlivých metodikách

RUP pro velké projekty, RUP pro malé projekty, OpenUP, MMSP, Scrum, XP

Bc. Kamila Langrová (xlank10)
ZS 2013/2014

Obsah

1. Úvod

Tato práce se věnuje porovnání projektových rolí v jednotlivých vybraných metodikách. Těmito metodikami jsou RUP pro velké projekty, RUP pro male projekty, OpenUP, MMSP, Scrum a XP, tedy Extrémní programování.

V první části stručně popíši jednotlivé metodiky, jejich principy, vypíšu seznam rolí a každou z nich v pár větách popíšu.

V druhé části práce pak tyto role v jednotlivých metodikách budu porovnávat a to z několika hledisek. Jedním z těchto hledisek bude počet těchto rolí v daných metodikách, další hledisko bude porovnání z pohledu funkcí, tedy obsahu náplně práce těchto rolí a posledním hlediskem bude zhodnocení těchto metodik mezi sebou, co se týče obsažnosti, tedy jaké všechny potřebné oblasti projektového řízení obsáhnou a v čem jsou které metodiky nedostatečné. Zhodnotím jejich výhody a nevýhody.

2. Stručný přehled metodik a jejich rolí

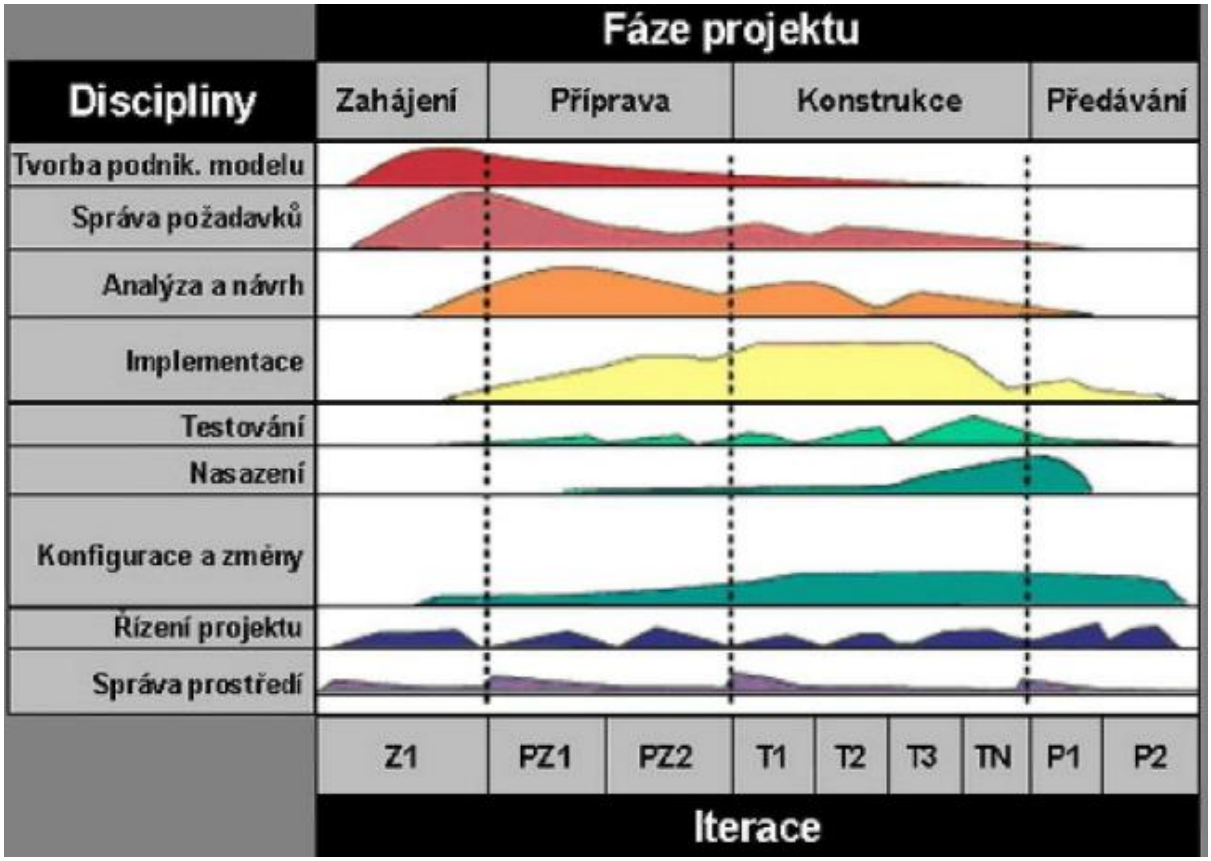
2.1. RUP pro velké projekty

2.1.1. Popis metodiky

„Rational Unified Process (dále jen RUP) je **objektově orientovaný iterativní přístup k životnímu cyklu software**. **RUP** náleží do skupiny tzv. přístupů řízených případy použití (use-casedriven approach), což znamená, že jako základní element je chápán případ použití definovaný jako posloupnost akcí prováděných systémem či uvnitř systému, která poskytuje určitou hodnotu uživateli systému. Pro modelování procesů se využívá prostředků jazyka UML (*Unifed Modeling Language*).
Hlavní myšlenka modelu je zjednodušena pro potřeby testování softwaru a zobrazena na obr 1. Na vodorovné ose jsou zaneseny fáze metodiky RUP. Fáze v tomto případě neodpovídají etapám životního cyklu produktu, ale dynamickému rozdělení metodiky RUP (jejich popis je uveden níže v této kapitole). V jednotlivých fázích se někdy uvádějí tzv. iterace, což jsou podmnožiny jednotlivých fází a představující jejích milníky. Svislá osa obsahuje vybrané procesy (které mají přímý vliv na testování softwaru v metodice) včetně důrazu, jaký je na ně kladen během metodiky RUP. Pojem „důraz“ v tomto kontextu chápejme jako objem pracovních kapacit rezervovaných v danou chvíli na konkrétní činnost. Ilustrace znázorňuje, jaký důraz je na daný proces v průběhu životního cyklu kladen.
RUP obsahuje celkem čtyři základní fáze. Každá fáze obsahuje několik dalších **iterací**.

1) Iterativní vývoj software
2) Správa požadavků
3) Architektura založená na komponentách
4) Vizuální modelování
5) Ověřování kvality software
6) Řízení změn software

Před započetím nové iterace musí být splněna dříve definovaná kritéria předchozí iterace. Fáze zahájení (inception) definuje účel, rozsah projektu a jeho obchodní kontext. Ve fázi projektování (elaboration) je potřeba analyzovat požadavky zákazníka, celého projektu a definovat základy architektury. Realizační fáze (construction) je nejdelší probíhá zde tvorba zdrojových kódů.
V poslední fázi předání (transition) může být projekt předán zákazníkovi nebo do dalšího cyklu.“ [10]



Obrázek 1 - Metodika RUP. Zdroj: [10]

2.1.2. Jednotlivé role [11]

1) Analysts

   a. Business Architect
      Tato role je zodpovědná za celkovou obchodní architekturu, podílí se na všech významných rozhodnutích týkajících se struktury, hlavní činnosti a její realizace, rozhraní, omezení a kompromisů.
   b. Business Designer
      Tato role upřesňuje specifikaci určité části podniku.
   c. Business-Process Analyst
      Tato úloha vede a koordinuje business požadavky.
   d. Requirements Specifier
      Tato role určuje a udržuje podrobné systémové požadavky.
   e. Stakeholder
      Tato role představuje zájmovou skupinu, jejíž potřeby musí být splněny v rámci projektu. To může být každý, kdo bude významně ovlivněn výsledkem projektu.
   f. System Analyst
      Tato úloha vede a koordinuje požadavky na systémovou funkcionalitu a vymezení systému.

2) Developer

a. Capsule Designer
   Tato role navrhuje „pouzdra“, což zajišťuje, že systém může reagovat na události včas, v souladu s požadavky na návaznost.
b. Database Designer
   Tato úloha vede design datové struktury, který bude použit v systému.
c. Designer
   Tato úloha vede návrh části systému, v rámci omezení daných požadavků, architektury a procesu vývoje projektu.
d. Implementer
   Tato role se vyvíjí softwarové komponenty a provádí vývojářské testování pro integraci do větších subsystémů, v souladu s přijatými standardy projektu.
e. Integrator
   Tato úloha vede plánování a realizace implementace částí z integrace k vytvoření buildů.
f. Security Architect
   Tato úloha vede vývoj bezpečnostní architektury systému, který zahrnuje výběr bezpečnostních vzorů pro návrh systému, dokumentaci technické politiky týkající se implementace zabezpečení a podpůrné klíčové technické rozhodnutí, která omezují celkovou realizaci zabezpečení projektu.
g. Software Architect
   Tato úloha vede vývoj softwarové architektury systému, která zahrnuje podporu a vytváření podporu klíčových technických rozhodnutí, které omezují celkový design a realizace projektu.
h. User-Interface Designer
   Tato role koordinuje design uživatelského rozhraní. To zahrnuje shromažďování požadavků na použitelnost a prototypování návrhů uživatelského rozhraní ke splnění těchto požadavků.

3) General

a. Any Role

   Tato role charakterizuje úkoly, které může vykonávat každý člen týmu.

b. Review Coordinator

   Tato role umožňuje formální hodnocení a inspekce, zajistí, že se vyskytují v případě potřeby, a ověří, že jsou prováděny na dostatečné úrovni.

c. Reviewer
   Tato role poskytuje včasnou zpětnou vazbu k členům týmu projektu na pracovní produkty, které byly předloženy k posouzení.

d. Stakeholder
   Tato role představuje zájmovou skupinu, jejíž potřeby musí být splněny v rámci projektu. To může být každý, kdo bude významně ovlivněn výsledkem projektu.
a. Technical Reviewer
   Tato role přispívá technickou zpětnou vazbou na práci na projektu produktů. Tato zpětná vazba musí být včasná a vhodná.

4) Manager

a. Change Control Manager
   Tato role určuje a dohlíží na řízení změn procesu.
b. Configuration Manager
   Tato role řídí celkový Configuration Management (CM) infrastruktury a prostředí vývojového týmu.
c. Deployment Manager
   Tato úloha vede přechod výrobku do komunity uživatelů, zajišťuje tyto plány, že jsou přiměřené, řídí problémy a sleduje pokrok.
d. Management Reviewer
   Tato role hodnotí plánování projektu a hodnotí výrobky v hlavních bodech v rámci životního cyklu projektu.
e. Project Manager
   Tato role plánuje, spravuje a přiděluje zdroje, určuje priority, koordinuje interakce se zákazníky a uživateli, a udržuje zaměření projektového týmu. Project Manager také zavádí soubor postupů, které zajistí integritu a kvalitu práce na projektu.
f. System Administrator
   Tato role udržuje rozvoj infrastruktury, a to jak hardware a software. To zahrnuje instalaci, konfiguraci, zálohování, atd.

g. Test Manager
   Tato role vede celkový testovací tým. To zahrnuje testování kvality, plánování a řízení zdrojů, a řešení problémů, které brání testování.

5) Production & Support

a. Course Developer
   Tato role se vyvíjí školicí materiály pro školení uživatelů o tom, jak produkt používat.
b. Graphic Artist

   Tato úloha vytvoří kresbu produktu, který je obsažen jako součást obalu výrobku.

c. Process Engineer
   Tato role je podpora rolí v RUP. Jejím hlavním cílem je vybavit projektový tým vhodným procesem rozvoje, a ujistit se, že členům týmu není bráněno vykonávat svou práci.

d. System Administrator

   Tato role udržuje rozvoj infrastruktury, a to jak hardware a software. To zahrnuje instalaci, konfiguraci, zálohování, atd.

e. Technical Writer
   Tato role vytváří materiální podporu koncových uživatelů, jako jsou uživatelské příručky, soubory nápovědy, poznámky k verzi, a tak dále.
f. Tool Specialist
   Tato role podporuje nástroje používané v rámci projektu. To zahrnuje výběr a získávání nástrojů, konfigurací a nastavením, a ověření, že fungují.

6) Tester

a. Test Analyst
   Tato role identifikuje a definuje požadované testy, sleduje detailní testovací průběh a výsledky v každém zkušebním cyklu a hodnotí celkovou kvalitu. Role představuje také subjekty, které nemají přímé či pravidelné zastoupení na projektu.
b. Test Designer
   Tato úloha vede definování testovacího přístupu a zajištění jeho úspěšné realizace. To zahrnuje identifikaci vhodných metod, nástrojů a pokynů k provedení požadovaných testů, a poskytnutí vodítko pro testování odpovídajících požadavků na zdroje.
c. Test Manager
   Tato role vede celkový testovací tým. To zahrnuje testování kvality, plánování a řízení zdrojů, a řešení problémů, které brání testování.
d. Tester
   Tato úloha provádí testy a zaznamenává výsledky testování.

2.2. RUP pro male projekty

Tato metodika vychází z metodiky RUP pro velké projekty (viz výše). Proto nebudu popisovat ani metodiku, ani role, pouze vypíši seznam rolí, které se v metodice vyskytují a kterých je výrazně méně než v RUP pro velké projekty (popis těchto rolí v předcházející kapitole).

2.2.1. Jednotlivé role

1) Analyst
   a. Requirements Specifier
   b. Stakeholder
   c. System Analyst

2) Developer
   a. Database Designer
   b. Designer
   c. Implementer
   d. Integrator
   e. Software Architect
   f. User-Interface Designer

3) General
   a. Any Role
   b. Review Coordinator
   c. Reviewer
   d. Stakeholder
   e. Technical Reviewer

4) Manager
   a. Change Control Manager
   b. Configuration Manager
   c. Management Reviewer
   d. Project Manager
   e. System Administrator
   f. Test Manager

5) Production & Support
   a. Process Engineer
   b. System Administrator

6) Tester
   a. Test Analyst
   b. Test Designer
   c. Test Manager
   d. Tester

2.3.OpenUP a MMSP

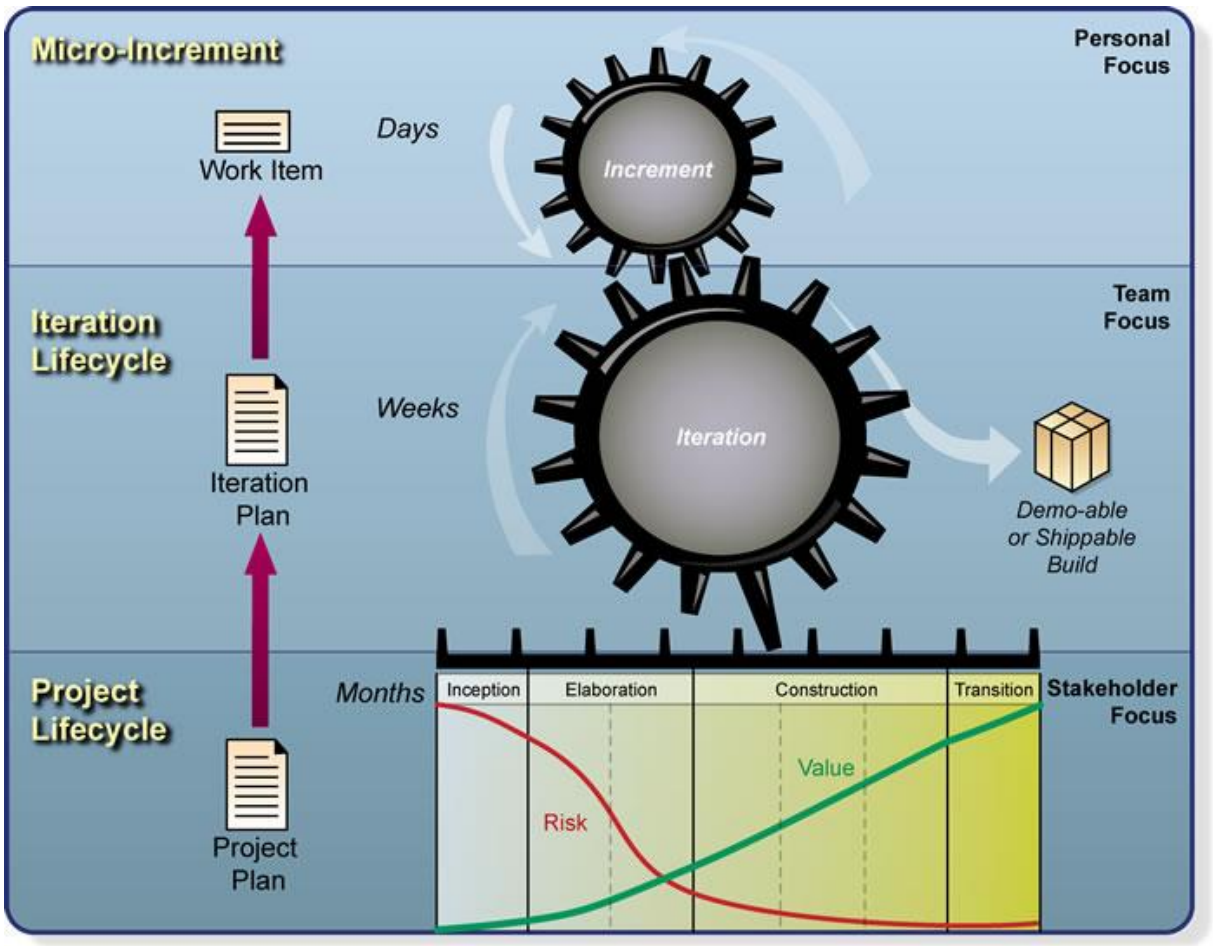
Jelikož metodika MMSP je lokalizovaná metodika OpenUP do češtiny, rozhodla jsem se je zařadit do jedné kapitoly a popsat pouze metodiku OpenUP.

2.3.1. Popis metodiky

„Metodika OpenUp vychází z metodiky RUP (**R**ational **U**nified **P**rocess), je vlastně modifikovanou - zásadně zeštíhlenou rigorózní metodikou RUP. Spolu s nástrojem Eclipse Process Framework je dostupná pod Eclipse Public License, tedy licenci na principu OpenSource. Každý tak může metodiku a její nástroje nejen využívat, ale i kopírovat, modifikovat a publikovat její upravené verze.
OpenUp je založena na iterativním a inkrementálním modelu životního cyklu. Je postavena na 4 základních principech, které vycházejí z Manifestu agilního vývoje [Buchalcevova, 2009]:

1) spolupráce pro sladění zájmů a porozuměni cílům projektu,
2) hodnoceni priorit s cílem maximalizace hodnoty pro zákazníka,
3) včasné zaměření na architekturu s cílem minimalizace rizik,
4) kontinuální zpětná vazba a zlepšovaní.

V rámci filosofie metodiky OpenUp je k vývoji softwaru přistupováno iterativně s průběžným testováním a průběžnou integraci. Výsledkem je možnost častých dodávek fungujícího softwaru a tím časté zpětné vazby zákazníka. Vzhledem k tomu se dá dobře reagovat na změny.
Součásti metodiky je metodicky obsah a procesní obsah. Metodický obsah je znovupoužitelný, použitelný pro veškeré projekty. Procesní obsah je aplikaci metodiky na konkrétní projekt.
Obsahuje časové hledisko a životni cyklus projektu. Pro konfiguraci metodiky slouží nástroj Eclipse Process Framework Composer, kde je možné si vybírat z jednotlivých metodických balíčků a metodiku si přizpůsobit přesně na míru. [Buchalcevova, 2009] Pote je jednoduché nakonfigurovanou metodiku publikovat například na firemním intranetu, kde k ní mohou přistupovat zaměstnanci pomoci webového rozhraní.
Metodika je rozčleněna tři vrstev. Mikropřirůstek, iterace, životni cyklus projektu.“ [2]



Obrázek 2 - Principy metodiky OpenUP a MMSP. Zdroj: [6]

2.3.2. Jednotlivé role

1) Architekt

„Architekt představuje na projektu roli zodpovědnou za definování softwarové architektury včetně přijímání klíčových rozhodnutí určujících technická omezení návrhu a implementace IS/ICT.“ [8]

2) Projektový manažer

„Klíčovou a mnohdy nezastupitelnou rolí na projektu je Projektový manažer. Jedná se o člena vývojového týmu, který je zodpovědný za vedení a plánování projektu, koordinaci spolupráce se všemi zainteresovanými stranami a soustředění celého týmu na dosažení projektových cílů.“ [8]

3) Analytik

„Analytik představuje v projektovém týmu roli, která je zodpovědná za sběr zákaznických a uživatelských požadavků od všech zainteresovaných stran, jejich porozumění, zachycení a určení jejich priorit. Vzhledem k tomu, že představuje „spojovací článek“ mezi vývojovým týmem určitého IS/ICT a jeho budoucími uživateli, je na něj kladena řada nároků.“ [8]

4) Tester

„Osoba v pozici Testera je na projektu zodpovědná za testování. „[8]

5) Nedefinovaná role

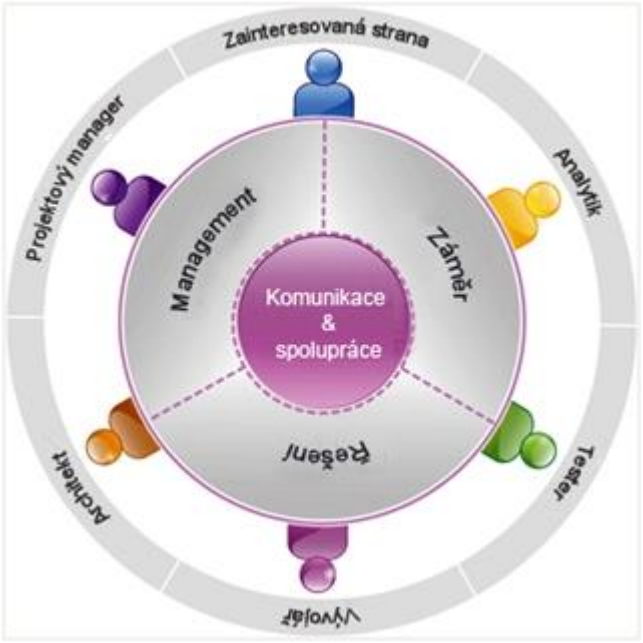
„Kromě šesti nejdůležitějších rolí mohou na různých projektech spolupracovat i osoby, jež není možné do vytyčených kategorií zařadit. Jejich role je pro účely této metodiky označována jako “Nedefinovaná“.“ [8]

6) Vývojář

„Role Vývojář je zodpovědná za vývoj jednotlivých komponent IS/ICT a jejich sjednocení do jednoho integrovaného řešení, včetně zajištění jeho souladu s navrhovanou architekturou. Může se podílet i na návrhu uživatelského rozhraní a jednotkovém testování (unit-testing).“ [8]

7) Zainteresovaná strana

„Tato role reprezentuje všechny, kteří mají zájem na úspěšném dokončení projektu, protože budou jeho výsledkem jakýmkoliv způsobem ovlivněni. Patří sem především budoucí uživatelé vyvíjeného IS/ICT, investoři, zákazníci či sponzoři.“ [8]



Obrázek 3 - Role a jejich zaměření. Zdroj: [7]

2.4.Scrum

2.4.1. Popis metodiky

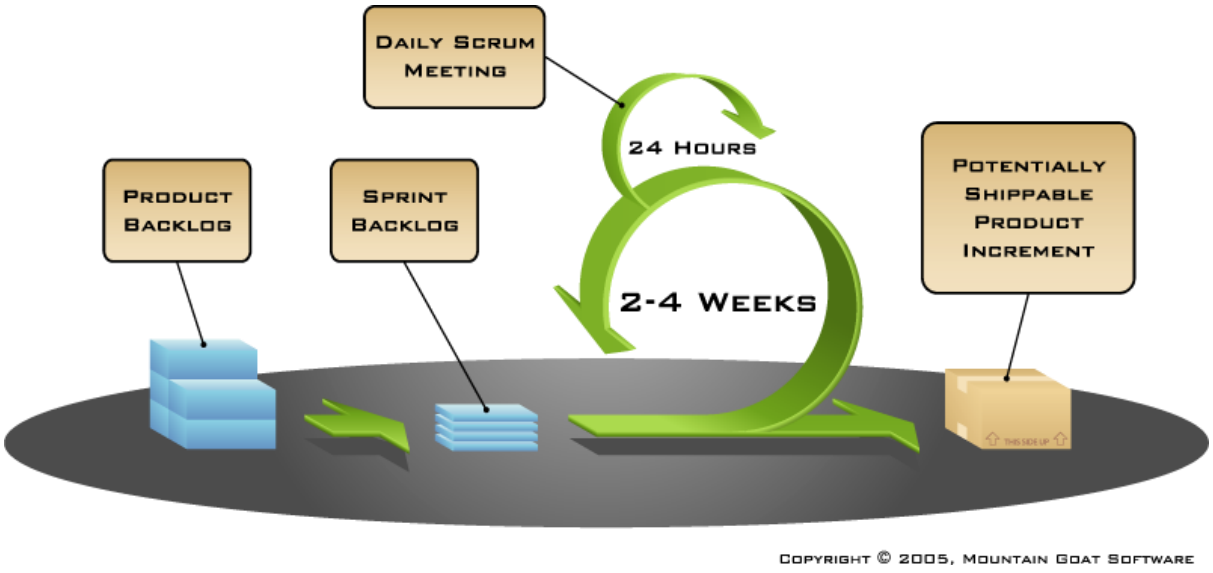
„Scrum je jednoznačně jednou z nejznámějších agilních metodik, někdy dokonce bývá považován za synonymum agilního vývoje. Jedna se o *lehkou* metodiku, jejími autory jsou Jaff Sutherland a Ken Schwaber.

Metodika Scrum je koncept, který je tvořen několika součástmi, kterými jsou: Scrum tým, role, činnosti, artefakty, pravidla.

Scrum tým je složen z vlastníka produktu, vývojového tymu a Scrum mastera. Tyto jednotlivé role jsou charakterizovány níže.

Scrum tým je samoorganizující se jednotkou - postup při provádění prací na projektu si volí sami členové týmu, nejsou přímo řízeni osobou z venku. Dále je tým samostatnou, multifunkční jednotkou, díky čemuž může dokončit svou práci, aniž by musel čekat na někoho z venku. Díky iterativnímu a inkrementálnímu přístupu se týmu dostava zpětná vazba.

Podle Scrumu je ideální vývojový tým dost malý na to, aby zůstal flexibilní a zároveň dost velký na to, aby byl schopen dokončit smysluplnou prací. Tymy s počtem menším než tři lide nemohou naplno využit interakci mezi členy, naopak tymy větší než 9 lidi jsou náročné na koordinaci. Optimální velikost Scrum tymu je tedy 3-9 členů.“ [2]



Obrázek 4 - Grafické znázornění metodiky Scrum. Zdroj: [3]

2.4.2. Jednotlivé role

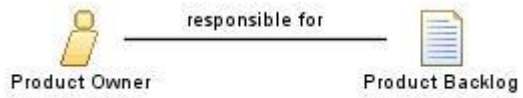
1) Scrum Master

„Scrum master je osoba zodpovědná za osvojování a dodržování pravidel Scrumu; trvá na dodržování jak ducha teorie Scrum, tak jeho technik a pravidel. Role Scrum mastera by se dala charakterizovat jako vedoucí tymu, avšak Scrum master by měl (nejen) tymu hlavně sloužit. Scrum master je kouč

tymu, vede ho směrem k sebeorganizovanosti a multifunkčnosti a kromě toho například odstraňuje překážky, které vývojovému týmu brání v prací na produktu.“ [2]

2) Product Owner

„Vlastník produktu nese zodpovědnost za prací vývojového tymu a maximalizaci hodnoty produktu. Nikdo jiny, než vlastník produktu není oprávněn úkolovat vývojový tým; naopak vývojový tým nemá povoleno plnit úkoly jiné osoby než vlastníka produktu.“ [2]



Obrázek 5 - Product Owner. Zdroj: [4]

3) Scrum Team

„Jen a pouze členové vývojového tymu vytvářejí přírůstky produktu. Tým pracuje jako celek, nevytvářejí se podtýmy.“ [2]

Obrázek 6- Scrum Team. Zdroj: [5]

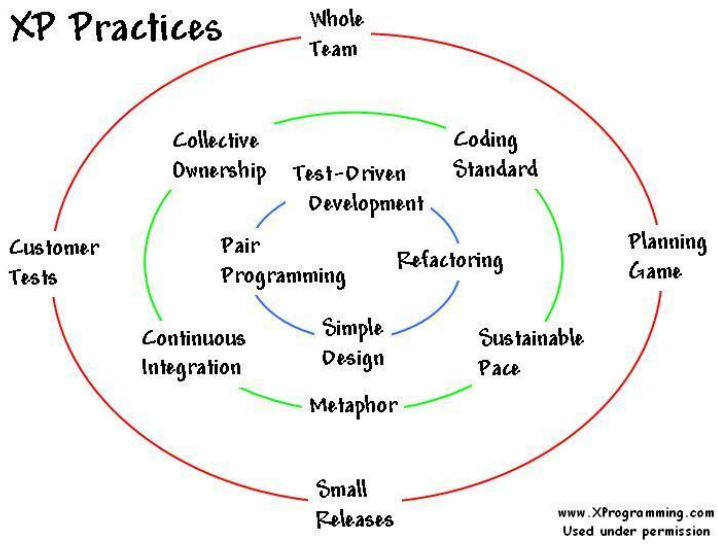
2.5.XP (Extrémní programování)

2.5.1. Popis metodiky

„Metodika e**X**treme **P**rogramming, (XP, extrémní programování) je jednou z nejznámějších agilních metodik, kterou představil Kent Beck v roce 1999. Někdy je považována téměř za synonymum pro agilní vývoj. Není až tak „extrémní " v pravém slova smyslu, její název je odvozen od toho, že základní principy programování dovádí do úplnosti, řekněme do extrému.
Hlavním cílem extrémního programování je snaha o maximální jednoduchost řešeni.
V rámci filosofie extrémního programování jsou uznávány čtyři hodnoty: Komunikace, jednoduchost, zpětná vazba, odvaha, respekt.
Metodika definuje čtyři základní činnosti, které nemají pevně danou posloupnost, všechny jsou spolu vzájemně provázané a zabývají se jimi všichni vývojáři každý den. Těmito činnostmi jsou:

1) Testování
2) Psaní kódu
3) Poslouchání
4) Návrh

Extrémní programování dále popisuje dvanáct postupů, které doporučuje použit pro dosažení hlavního cíle, tedy jednoduchého, kvalitního řešení. Těmito postupy jsou plánovací hra (plánování formou dialogu), malé verze (časté dodávky, buildy), metafora, jednoduchý návrh, testování, refaktorizace, párové programování, společné vlastnictví kódu, nepřetržitá integrace, čtyřicetihodinový pracovní týden, zákazník na pracovišti, standardy pro psaní zdrojového kódu.
Produktem extrémního programování je aplikace, která je maximálně jednoduchá a obsahuje pouze to, co je požadováno. Jsou definovány jednotlivé role, které jednak pomáhají vymezovat pravomoci, tak zapojuji aktivně zákazníka do procesu vývoje aplikace.“ [2]



Obrázek 7 - Diagram metodiky XP. Zdroj: [9]

2.5.2. Jednotlivé role

„XP definuje jednotlivé projektové role: Programátor, tester, kouč, stopař, velký šéf, konzultant a zákazník.

1) Programátor

Programuje, testuje, komunikuje s ostatními programátory a neustále refaktorizuje.

2) Tester

Zajišťuje pravidelné spouštění automatizovaných testů, případně může spolupracovat se zákazníkem na sestavení funkčních testů.

3) Kouč

Řídí a motivuje tým, dodává mu potřebný nadhled. Je zodpovědný za proces vývoje
jako celek.

4) Stopař

Zprostředkovává tymu zpětnou vazbu, v případě potřeby upravuje harmonogram práce. Vede protokol s výsledky funkčních testů.

5) Velký šéf

Dodává tymu odvahu. Neplete se do cesty, když tým funguje, ale je k dispozici v případě problému.

6) Konzultant

Externista s vyšší mírou odborných znalosti, který je povolán v případě problémů.

7) Zákazník

Zákazník ví, co programovat. Měl by se naučit, jak psát zadání a funkční testy.“ [2]

3. Porovnání rolí dle počtu

	Management	Development	Testing
RUP lp	Managers (7) General (5)	Developers (8) Analysts (6) Production/Support (6)	Testers (4)
RUP sp	Managers (6) General (5)	Developers (6) Analysts (3) Production/Support (2)	Testers (4)
OpenUP **MMSP**	Project manager Stakeholder Any role	Architect Analyst Developer	Tester
Scrum	Scrum Master Product Owner	Team (5-7)	-
XP	Manager Coach (Customer)	Developer Tracker	Tester (Tracker)

Celkový počet rolí v daných metodikách:

1) RUP pro velké projekty: 36
2) RUP pro malé projekty: 26
3) OpenUP/MMSP: 7
4) Scrum: 3
5) XP: 7

Výše vidíme, že agilní metodiky mají výrazně méně rolí než ty klasické metodiky. Vychází to už z jejich definice. Sice mezi RUP pro velké projekty a pro malé projekty je rozdíl celých 10 rolí, ale 26 rolí v RUP pro malé projekty je stále hodně například oproti Scrum rolím, které jsou 3.

Ve větších metodikách vychází početně největší podíl rolí na development, zatímco u malých metodik je více rolí spíše v managementu. Zajímavý je fakt, že obecně není kladen velký důraz na testerské role. Pokud úplně nechybí, pak většinou bývá uveden pouze tester, pouze u RUP metodiky jsou čtyři role jen na testing.

Nicméně velký počet rolí v metodice je pochopitelný u velkých projektů, ale o to větší nedorozumění, „komunikační šum“ a ztrátu informací může způsobit.

4. Porovnání rolí dle funkcí

Co se náplně práce jednotlivých rolí týče, tak se metodiky od sebe velmi liší. Je logické, že metodiky pro větší projekty mají více rolí, než metodiky pro menší projekty, nicméně v některých případech se v agilních metodikách vyskytují, i v tom malém počtu rolí, nějaké role, které velkým metodikám chybí.

Nejprve bych ráda upozornila na absolutní absenci testovacích rolí v metodice Scrum. V dnešní době to považuji za velký nedostatek a z vlastních pracovních zkušeností vím, že je snaha vždy nějak testery „našroubovat“ i to projektu řízeného Scrum metodikou. Osobně mi i počet testovacích rolí v některých metodikách připadá velmi slabý, i když to jsou agilní metodiky, tak testování považuji za jednu z těch velmi důležitých částí projektu, proto se domnívám, že vhodná je role nejen Testera ale minimálně Test manažera a v ideálním případě i Test analytika.

Další, z mého pohledu, zajímavou rolí je rozhodně Konzultant z metodiky XP. Nejen, že se nevyskytuje u ostatních agilních metodik, ale tato role není takto konkrétně definována ani v RUP metodikách. Částečně jeho práci v ostatních metodikách přebírá role Analytika, případně by se dala tato role dát na roli Nedefinovaná/Any role, ale osobně si myslím, že kdyby každý projekt měl nějakého externistu odborníka, pak by se problémy při projektovém řízení daly vyřešit rychleji a hlavně efektivněji.

Za velmi praktické považuji rozdělení manažerů na 7 (popř. 6) rolí v metodice RUP, jelikož se domnívám, že technické a jiné specifikace projektu jsou často velmi složité a náročné (a to i u malých projektů) a ne vždy (obzvláště u metodiky Scrum je to viditelné) je jeden (či dva) lidé schopni pojmout celý rozsah této problematiky. Takto rozdělením rolí se dosáhne toho, že každý bude odborníkem na danou část této problematiky a dohromady toho budou vědět mnohem více, než jeden člověk, který obsáhne celou tuto problematiku.

A závěrem bych ráda upozornila na tzv. Nedefinovanou roli (Any role). Domnívám se, že v každé metodice by měla být taková „role“, jelikož každý projekt je něčím specifický, a když pak je potřeba nějakého člověka navíc, kvůli této jedinečné funkcionalitě/potřebě. Tato role se z uvedených metodik vyskytuje v RUP (obou metodikách), OpenUP a MMSP. Chybí ve Scrum a metodice XP.

5. Závěr

Na závěr bych zhodnotila, že celkově nelze jednoznačně říci, která z metodik je nejvhodnější, které role jsou důležité, a které nikoliv. Každá role má své klady a zápory, větší množství rolí přispívá k větší odbornosti, ale k větší ztrátě informací a přílišné administrativě a byrokracii, zatímco menší počet rolí může způsobovat problémy při potřebě vysoké odbornosti konkrétní části projektu.
Z vlastní zkušenosti tedy doporučuji spíše nějakou kombinaci více různých metodik. Například u nás v bance pro vývoj nové mobilní aplikace používáme Scrum v kombinaci s lístečky z Kanbanu a testování funguje na principu RUP. Nicméně je vždy vhodné zvolit metodiku podle specifik daného projektu a to nejen podle rozsáhlosti, budgetu a požadavků, ale i dle možností firmy dodat lidi do týmu a podle best practices dané firmy. Což je důvod, proč si velké firmy vyvíjejí vlastní metodiky projektového řízení, které většinou tedy vychází k některé již známe metodiky, a je „přiohnutá“ na konkrétní specifika dané firmy (více o tématice vlastních metodik: LANGROVÁ, Kamila. Analýza projektových rolí v bankopojišťovacím sektoru ČR. Praha, 2013. Bakalářská práce. Vysoká škola ekonomická v Praze.).

6. Zdroje

[1] KULHÁNKOVÁ, Barbora. Výběr metodiky řízení vývoje softwaru pro prostředí mikropodniku. Praha, 2013. Bakalářská práce. Vysoká škola ekonomická v Praze.

[2] Scrum Overview: Main Description. [online]. [cit. 2013-12-19]. Dostupné z: http://kitscm.vse.cz/scrum/index.htm

[3] Role: Product Owner: Relationships. [online]. [cit. 2013-12-19]. Dostupné z: http://kitscm.vse.cz/scrum/index.htm

[4] Role: Scrum Team: Relationships. [online]. [cit. 2013-12-19]. Dostupné z: http://kitscm.vse.cz/scrum/index.htm

[5] Introduction to OpenUP: Main Description. [online]. [cit. 2013-12-19]. Dostupné z: http://kitscm.vse.cz/openup_published/index.htm

[6] Role Set: Role: Main Description. [online]. [cit. 2013-12-19]. Dostupné z: http://mmsp.czweb.org/

[7] Roles. [online]. [cit. 2013-12-19]. Dostupné z: http://mmsp.czweb.org/

[8] Getting Started: Main Description. [online]. [cit. 2013-12-19]. Dostupné z: http://kitscm.vse.cz/XP_published/Publish/index.htm

[9] RUP – Rational Unified Process. [online]. [cit. 2013-12-19]. Dostupné z: http://testovanisoftwaru.cz/manualni-testovani/modely-zivotniho-cyklu-softwaru/rup/

[10] Role Sets. [online]. [cit. 2013-12-19]. Dostupné z: https://kitscm.vse.cz/RUP/SmallProjects/index.htm#core.base_rup/customcategories/additional_resources_E6845C6F.html

7. Seznam obrázků