

4IT421 – Zlepšování procesů budování IS

RUP disciplína Project management

Hanusová Lucie

Kulíšková Romana

2012

Obsah:

1.	Rational Unified Process.....	1
1.1	Best practices.....	2
1.2	Disciplíny.....	2
2.	Řízení projektu.....	3
2.1	Postup při řízení projektu	3
2.2	Role v řízení projektu	4
2.2.1	Project manager	4
2.3	Dokumentace v rámci řízení projektu	6
3.	Životní cyklus řízení projektu	9
3.1	Příprava projektu	9
3.1.1	Zahájení	9
3.1.2	High level plánování	9
3.2	Vykonávání	11
3.2.1	Detailní plánování	11
3.2.2	Vykonávání	13
3.2.3	Řízení a monitorování	13
3.3	Uzavření a zhodnocení projektu (iterace)	14
4.	Výhody a nevýhody	15
4.1	Výhody metodiky RUP obecně	15
4.2	Nevýhody metodiky RUP obecně	15
5.	Závěrem.....	17

Zdroje

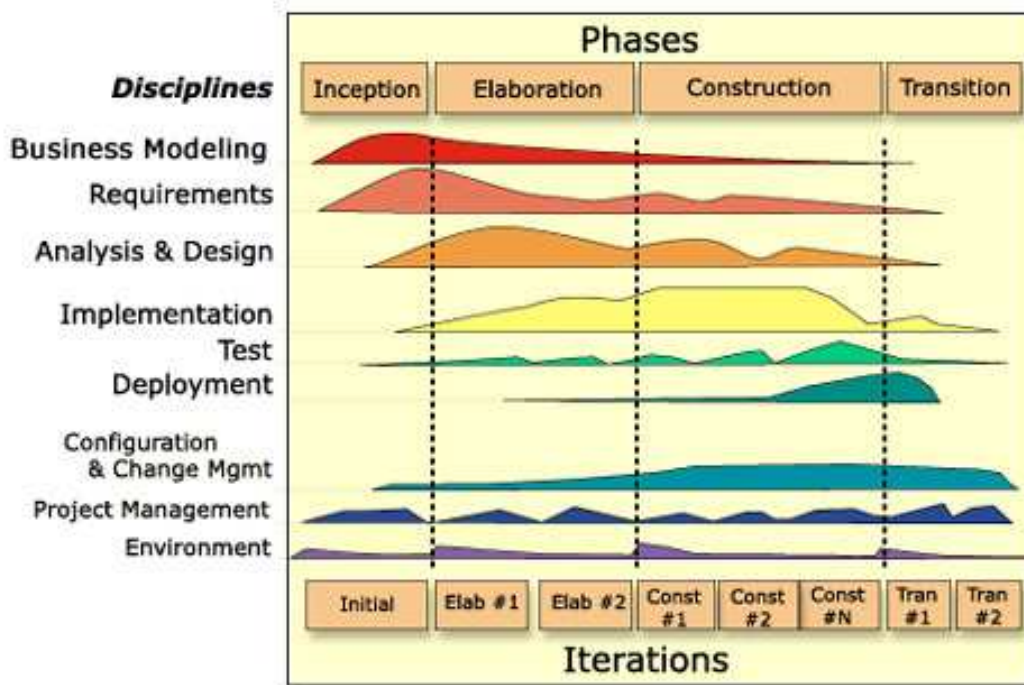
1. Rational Unified Process

Rational Unified Process (dále jen RUP) je metodika vývoje software vytvořená a používaná společností Rational Software Corporation (nyní divize IBM), která tuto metodiku distribuuje formou softwarového produktu. RUP představuje objektově orientovaný přístup k životnímu cyklu software. [ALDORF]

RUP se dělí se do čtyř hlavních fází: [Řízení projektů – Project management]

- Zahájení
- Příprava
- Budování
- Zavedení

Fáze projektu znázorňuje následující obrázek. Horizontální osa představuje fáze metodiky. Fáze mají své podmnožiny, tzv. iterace, které představují jejich milníky. Vertikální osa značí jednotlivé disciplíny uplatňující se během projektu a zároveň také jejich důležitost během jednotlivých fází.



Obr.1: Životní cyklus projektu metodiky RUP [Řízení projektů – Project management]

1.1 Best practices

RUP má šest tzv. nejlepších praktik, které se používají při vývoji software: [ALDORF]

1. Iterativní vývoj software
Projekt je rozdělen do čtyř fází. Každá fáze obsahuje několik iterací s „vodopádovým“ životním cyklem. Na konci každé iterace je výstupem spustitelná verze software.
2. Aktivní správa požadavků
Požadavky se sbírají, upřesňují, upravují a dokumentují během průběhu celého projektu. Pravidelné schůzky zadavatele a dodavatele.
3. Architektura založená na komponentách
Systém tak lze tvořit postupně, změny se zpracovávají snadněji a projekt je možné snadněji rozdělit do několika vývojářských týmů
4. Vizuální modelování
Grafické zobrazení zpřehledňuje návrh a poskytuje konzistenci mezi modelem a vlastní implementací.
5. Ověřování kvality software
Výsledný software je možné předat, jestliže splňuje požadovaná kvalitativní kritéria, což zajišťuje testování.
6. Řízení změn
Změny lze provádět během průběhu celého projektu a je nutné tyto změny řídit a dokumentovat.

1.2 Disciplíny

Metodika RUP rozlišuje dva druhy disciplín – hlavní a podpůrné. Hlavní disciplíny se vztahují k vlastní práci na projektu, disciplíny podpůrné slouží k řízení a koordinaci prací na projektu. [ALDORF]

Hlavní disciplíny:

- Tvorba podnikového modelu
- Správa požadavků
- Analýza a návrh
- Implementace
- Testování
- Nasazení

Podpůrné disciplíny:

- Řízení projektu
- Řízení změn a konfigurace
- Správa prostředí

V rámci RUP každá disciplína stanoví role (kdo provádí úkol), činnosti (jak provádí úkol) a artefakty (čeho se činností dosáhne). Role definuje chování a odpovědnost jednotlivce nebo skupiny za činnosti a artefakty. Činnost je úkol, za který je odpovědná určitá role a popisuje kroky potřebné k vytvoření nebo aktualizaci jednoho nebo více artefaktů. Artefakt je vstup nebo výstup činnosti.

2. Řízení projektu

Řízení projektu je hlavní podpůrná disciplína, která zajišťuje plánování a koordinaci prací v rámci projektu. Cílem je nasměrovat aktivity, které jsou v rámci projektu vykonávány. Mezi důležité úkoly patří řízení rizik, zajištění a řízení zdrojů a to jak personálních, tak také materiálních. [ALDORF]

Kritickými činnostmi jsou:

- Zahájení nového projektu
- Posílení vztahů s externími týmy a zdroji
- Řízení rizik
- Odhad a plánování
- Správa iterací
- Ukončení fáze nebo projektu

2.1 Postup při řízení projektu

- Zahájení projektu
 - identifikace rizik
 - ekonomický plán
 - zahájení projektu
- Plánování vývoje software
 - vytvoření / aktualizace plánu vývoje software
- Plánování iterace
 - vytvoření plánu iterace (činnosti, využití zdrojů, kritéria pro zhodnocení)
 - aktualizace plánu vývoje software
- Řízení iterace
 - personální a materiální obsazení
 - provádění plánu iterace (zhodnocení a porovnání výsledků s plánem)
- Ukončení a zhodnocení iterace
 - vyhodnocení iterace (dodržování harmonogramu, plánované spotřeby zdrojů)
 - aktualizace seznamu rizik
- Monitorování a řízení projektu (po celou dobu trvání projektu)
 - monitorování projektu
 - plánování a přidělování úkolů
 - řešení nestandardních situací
- Ukončení fáze projektu (formální ukončení fáze vedoucím projektu)
 - příprava na uzavření fáze projektu (splnění hodnotících kritérií pro daný milník)
- Ukončení projektu (předání software zadavateli)
 - příprava ukončení projektu (zhodnocení) [ALDORF]

2.2 Role v řízení projektu

- Zadavatel

Organizace nebo osoba, která definuje požadavky na systém, podílí se na tvorbě systému a financuje projekt.

- Project manager

Osoba zodpovědná za celý projekt. Zajišťuje splnění požadavků zadavatele, řídí vývojářský tým a dohlíží na dodržení harmonogramu prací.

- Organization manager

Mnoho úkolů se v rámci organizací nebo různých projektů opakuje. Nejedná se o totožnou náplň úkolů, ale o rozsah. Příkladem mohou být lidské zdroje a jejich funkce řízení zásob, kdy toto vše má na starost právě organization manager.

- Acting project manager

Projekt někdy může začít bez určeného project managera a tak zkušený člen týmu převeze všechny nebo některé jeho povinnosti. Je lepší mít někoho alespoň na dočasné řízení projektu, než nemít nikoho, což s sebou nese značná rizika. Je důležité, aby se role acting project managera ujala osoba již se zkušenostmi a znalostmi.

- Technical project manager

Tato osoba se v praxi nejvíce osvědčila u projektů s menšími týmy – zpravidla s méně než deseti členy. Technical project manager je schopen produkovat rozumné plány, protože jsou založeny na jeho pochopení zadavatelských požadavků po technické stránce.

- Program manager

Některé projekty jsou příliš rozsáhlé a je lepší zavést hierarchickou strukturu řízení projektu s nejvyšší úrovní – program managerem. Teprve pod tuto osobu pak spadá project manager. Program manager je zvláštním druhem v řízení projektu, kdy se manager zabývá projektem komplexně. Je odpovědný za rozdělení rozsahu na zvládnutelné kousky (projekty), dokumentuje a řídí tyto jednotlivé projekty, dohlíží na jejich závislost a dílčí řízení těchto menších projektů pak deleguje. [TEMNENCO]

2.2.1 Project manager

Project manager by měl disponovat určitými dovednostmi a zkušenostmi, které závisí na velikosti a technické složitosti projektu. Project manager musí: [UPEDU]

- mít zkušenosti v oblasti vývoje software
- mít dovednosti v oblastech analýzy a řízení rizik, odhadu, plánování a rozhodování
- mít komunikační a vyjednávací schopnosti
- ukázat schopnosti s vedením a budováním týmu

- správně pracovat v rámci time managementu, dělat správná rozhodnutí a být schopen rychle pracovat pod tlakem
- mít dobré interpersonální dovednosti a zdravý úsudek při výběru členů týmu
- být objektivní při přidělování a hodnocení práce
- sdílet architektonické vize, ale zároveň být pragmatický při stanovení rozsahu a provádění plánu

Činnosti project managera: [UPEDU]

- Rozvoj plánu měření

Plán měření popisuje cíle, které musí projekt splňovat pro úspěšné dokončení, a metriky pro určení, zda je projekt na dobré cestě.

Slouží pro definování cílů řízení, pokud jde o kvalitu, pokrok a zlepšení. Zjištění toho, co je třeba pravidelně měřit pro podporu daných cílů.

Výstupem jsou seznam rizik a plán měření.

Kroky: 1. Definování základních manažerských cílů

2. Potvrzení cílů
3. Definování dílčích cílů
4. Určení metrik potřebných pro splnění dílčích cílů
5. Určení primitivních metrik pro výpočet
6. Sepsání plánu měření
7. Vyhodnocení plánu měření
8. Zavedení mechanismu sběru

- Plánování etap a iterací

Je potřeba zajistit odhad celkového rozsahu projektu, pracnosti a nákladů. Hrubý plán projektu obsahuje hlavní milníky a klíčové výstupy v životním cyklu produktu. Definují se sady iterací v rámci jednotlivých fází projektu a určují se cíle pro každou iteraci. Dále se vytváří harmonogram, rozpočet, plánují se zdroje a definují se činnosti pro řádné dokončení projektu.

Výstupem jsou plán vývoje software a seznam rizik.

Kroky: 1. Odhad projektu

2. Definování milníků projektu
3. Definování cílů milníků
4. Definování počtu, délek a cílů iterací
5. Upřesnění milníku týkajícího se termínu a rozsahu projektu
6. Stanovení požadavků na zdroje
7. Plán pro ukončení projektu

- Plánování rozvoje iterace

Vytvoření detailního plánu pro každou iteraci, který obsahuje podrobný rozpis pracovních činností a odpovědností za přidělené úkoly, iterační milníky, výstupy a hodnotící kritéria pro iteraci.

Výstupem je plán iterace.

Kroky: 1. Určení rozsahu iterace
 2. Definování kritérií pro hodnocení iterace
 3. Definování činností v rámci iterace
 4. Přidělení odpovědností za činnosti

- Plánování a rozdělení práce

Pro přizpůsobení schválených změn pro produkt a postup projektu, které vznikají v průběhu iterace.

Výstupem jsou pracovní příkaz, plán vývoje software a plán iterace.

Kroky: 1. Schválení žádosti o změnu iterace
 2. Přiřazení odpovědnosti
 3. Popis práce a očekávaných výstupů
 4. Stanovení pracovních zdrojů a dalších zdrojů
 5. Sestavení plánu
 6. Přeplánování
 7. Vydání pracovního příkazu

2.3 Dokumentace v rámci řízení projektu

Zásadními dokumenty v rámci řízení projektu jsou: [CHARBONNEAU], [UPEDU]

- Ekonomický plán projektu

Dokument převádí vizi do ekonomické sféry. Software je v první řadě investice, takže je potřeba zaměřit se na ekonomickou stránku projektu. Dokument by měl obsahovat ekonomické přínosy a dobu návratnosti investice.

Odpovědnost za tento dokument nese vedoucí projektu. Ekonomický plán se vytváří ve fázi zahájení a slouží k rozhodnutí, zda projekt realizovat nebo ne. Během projektu se pak neustále aktualizuje.

- Seznam rizik

Dokument je seznamem stěžejních faktorů, které by mohly projekt ohrozit v jeho průběhu.

Odpovědnost za seznam rizik nese vedoucí projektu. Seznam je vytvořen ve fázi zahájení. Během projektu se při výskytu dalšího problému aktualizuje, minimálně ovšem po skončení a zhodnocení jedné iterace.

- Plán vývoje software

Plán vývoje tvoří soubor potřebné dokumentace k řízení projektu. Tvoří jej jednotlivé, během projektu aktualizované, dokumenty:

- Plán iterací
- Plán správy požadavků
- Seznam rizik
- Konfigurace RUP
- Pravidla postupů (modelování, programování, testování,...)

Za aktuálnost a úplnost souhrnu dokumentů zodpovídá vedoucí projektu. Plán vývoje se vytváří ve fázi zahájení a během projektu dochází k jeho aktualizaci. Kromě vedoucího projektu jej používají členové projektového týmu, aby porozuměli tomu, co je třeba udělat a věděli jak na sebe činnosti navazují. Aktualizuje se minimálně v rámci milníku, zpravidla však po dokončení iterace.

- Plán měření

Plán měření definuje cíle, metriky a primitivní metriky, které jsou shromažďovány v rámci projektu, a díky kterým se sleduje jeho průběh. Dokument určuje, jaké metriky by měly být vypočítávány v průběhu projektu ke sledování jeho pokroku, a vzhledem k souboru konkrétních cílů projektu.

Odpovědnost za obsah a měření plánu nese project manager. Plán je sestavován v počáteční fázi jako součást obecné plánovací činnosti.

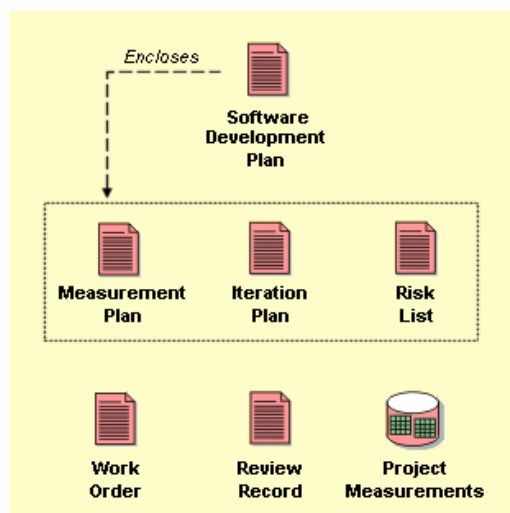
- Plán iterace

Dokument představuje cíle a plán činností pro danou iteraci.

Odpovědnost za plán a jeho následné dodržování nese vedoucí projektu. Plán iterace se vytváří vždy před zahájením dané iterace.

- Pracovní příkaz (smlouva)

Manažerovi slouží jako prostředek komunikace a stanovení toho, co je třeba udělat a kdo za to zodpovídá. Pracovní smlouva je sjednaná dohoda mezi project managerem a členy týmu, kteří vykonávají určité činnosti podle definovaného plánu, s určitými výstupy, pracností a omezenými zdroji.



Obr.2: Role a hlavní artefakty vytvořené v rámci projektového řízení [UPEDU]

3. Životní cyklus řízení projektu

Jak již bylo zmíněno v předchozí kapitole, řízení projektu jako takové je hlavní podpůrnou disciplínou, která zajišťuje plánování a koordinaci prací. Dalším významným úkolem této disciplíny je zajištění potřebných personálních a materiálních zdrojů. [ALDORF]

Životní cyklus projektu má několik základních fází. Mezi ně patří příprava projektu, vykonávání a uzavření a zhodnocení projektu. Tyto fáze jsou poté dále rozděleny na dílčí úkoly. Ty si podrobněji popíšeme v následujícím textu.

3.1 Příprava projektu

3.1.1 Zahájení

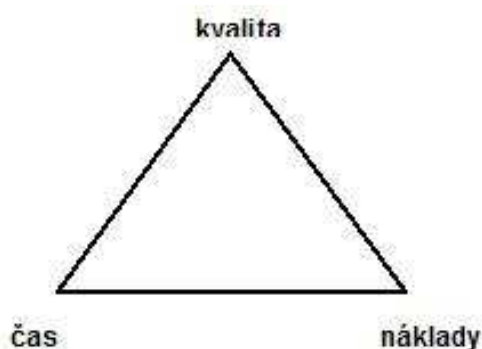
Na základě zadání se vytvoří úvodní seznam rizik, která jsou spojená s projektem. Zároveň se provede ekonomická analýza. Jejím výstupem je ekonomický plán. Na základě těchto podkladů se vedoucí projektu rozhodne, zda se vůbec vyplatí software vyvíjet či nikoliv. [ALDORF]

3.1.2 High level plánování

Projekt je plánovaná, řízená a časově ohraničená skupina činností, která má přesně dané vstupy a výstupy a spotřebované zdroje (lidské, technologické, finanční). Aby mohl projekt dobře fungovat, je potřeba všechny tyto aspekty „skloubit“ dohromady. K tomu slouží plánování projektu.

Projektové plány se liší v detailech a v rozsahu v závislosti na jejich účelu. I když se mluví o high level plánování, manažeři se obecně zabývají schválením projektu před jeho zahájením.

Existuje tzv. trojúhelník kvality [ŽÁČEK], ve které se projevuje vztah 3 hlavních faktorů. Času, nákladů a výkonu (kvality). Jinak řečeno, každý projekt se snaží vyvážit trojúhelníkový vztah času, finančních prostředků a rozsahu – nelze změnit jeden faktor, aniž by to neovlivnilo alespoň jeden další faktor. Úkolem vedoucího projektu je zachovat soudržnost trojúhelníku.



Obr.3: Projektový trojúhelník [CHVALOVSKÝ]

Jak toho docílit? Nejprve se při vzniku problému lokalizuje problém v rámci projektového trojúhelníku. Jedná se o faktor času (plán), prostředků (rozpočet) nebo rozsahu? Dále se určí, které strany trojúhelníku je možné změnit a které jsou fixní. Následně se upraví parametry tak, aby se problém vyřešil a projekt se optimalizoval.

Ovšem nastává tu ještě jeden problém a to je pohled zákazníka. Ten si samozřejmě vždy přeje co nejkvalitnější produkt v krátkém termínu a s co nejnižšími náklady. Ovšem aby byl projekt proveditelný, musí si sám tým určit jeden z předchozích faktorů. Například pokud bude zákazník trvat na zkrácení termínu, musí automaticky počítat s většími náklady nebo snížením kvality. Naopak při snaze kvalitu zvýšit, je potřeba navýšit náklady nebo prodloužit termín.

Plánování se zabývá přípravou projektového plánu. Projektový plán má za cíl popsat, co se má udělat, do kdy a za kolik a má následující vlastnosti: [POSPÍŠILOVÁ]

- identifikuje vše, co je zapotřebí k úspěšnému dokončení projektu
- obsahuje harmonogram pro načasování těchto úkolů a souvisejících milníků
- definuje potřebné zdroje se zárukou jejich dostupnosti v patřičnou dobu a zohledňuje nasazení těchto zdrojů a jejich řízení
- má rozpočet pro každý úkol
- obsahuje odpovídající rezervu pro nepředvídatelné události
- je věrohodný jak pro předpokládané realizátory, tak pro management

Do plánování je nutné zapojit osoby, které budou na projektu pracovat – jednak o úkolu vědí víc (jsou odborníkem), jednak pokud si práci naplánují, budou ji pravděpodobně i schopni vykonat a budou motivováni úkol splnit.

Projektový plán:

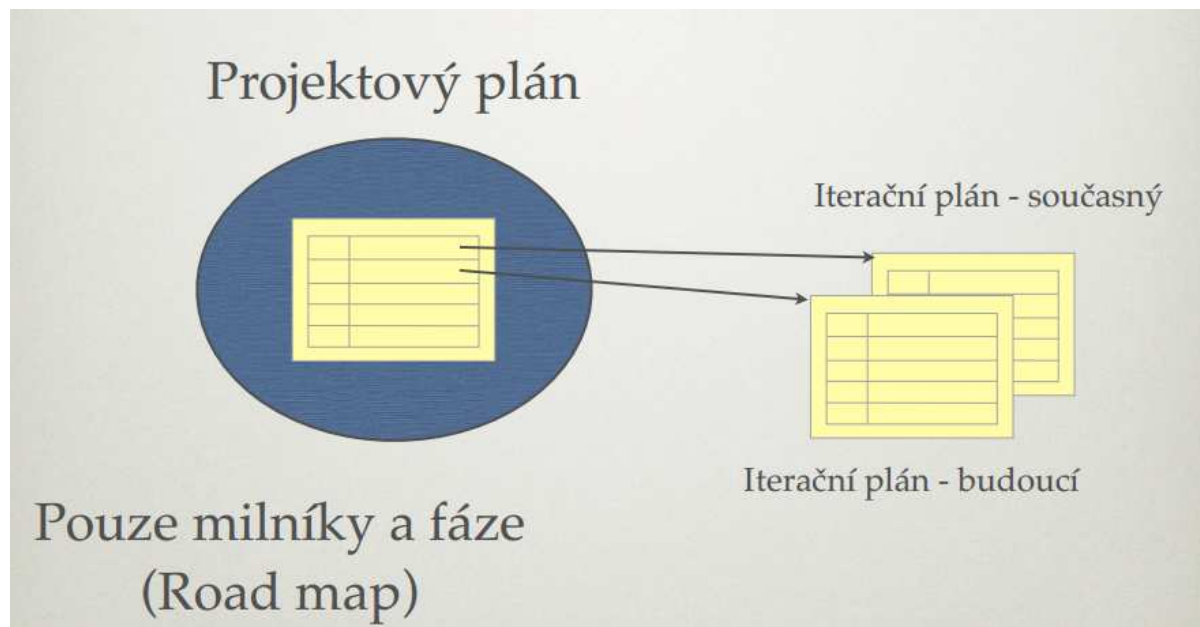
- simuluje skutečnou realizaci projektu
- pomáhá koordinaci a komunikaci – informuje každého na projektu, co se od něj očekává, co dělají jiní
- poskytuje základ pro sledování průběhu projektu – ukazuje (šéfovi, zadavateli, ostatním manažerům) kam projekt směřuje, co právě děláte, jak dlouho se to bude ještě dělat
- motivuje lidi k lepší práci

Každý projekt by měl mít v projektovém plánu zabudovanou rezervu času, nákladů, případně i kvality provedení.

3.2 Vykonávání

3.2.1 Detailní plánování

Existují dvě úrovně plánování. Jedná se o projektový plán a iterační plán (současný nebo budoucí). Projektový plán obsahuje pouze milníky a fáze, obecně se mu říká Road map.



Obr.4: Projektový vs. iterační plán [ŽÁČEK]

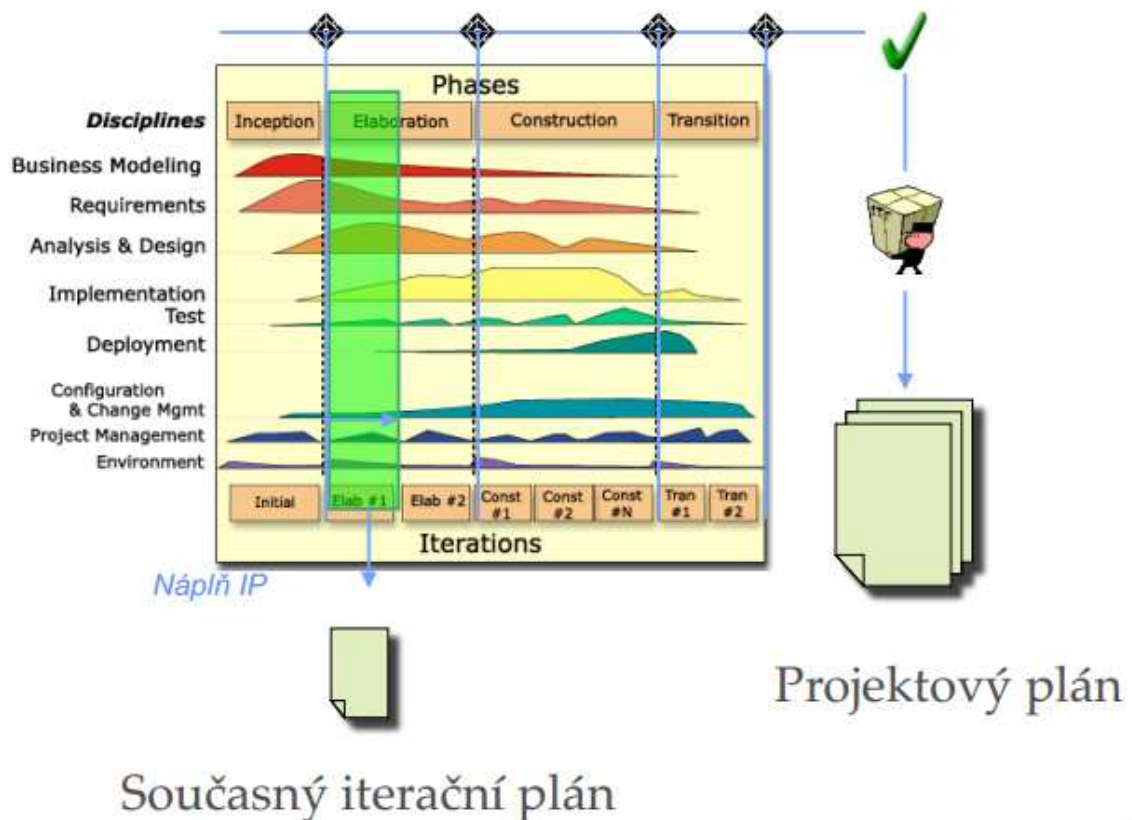
Iterační plánování

Iterační plán je detailní plán, který identifikuje, čeho má být dosaženo v dané iteraci. Zahrnuje seznam rizik, která je potřeba prozkoumat nebo se kterými je nutné se vypořádat. Dále obsahuje podmnožinu požadavků, které by měly být implementovány, a pravděpodobně také nějaké požadavky na změny, které by měly být ošetřeny.

Vyhodnocení iterace je vyvážené pravdivé zhodnocení, které porovnává dosažené cíle s plánem iterace. Zjišťuje, zda byly cíle iterace naplněny, či nikoliv. Není to pouze kontrola splnění implementačních požadavků stanovených pro danou iteraci. Je velmi důležité, zda byla rizika odhalena během iterace úspěšně odstraněna nebo zmírněna. Také je zapotřebí si všimnout, zda nebyla identifikována nová rizika nebo nevyvstal nějaký nový problém. Výsledky dané vyhodnocené iterace jsou využity v plánu následujících iterací. [JULINEK]

Vývoj projektu probíhá v opakujících se iteracích s typickou délkou (většinou to bývá 1-4 týdny). Na začátku každé iterace zákazník vybere podle svých aktuálních obchodních priorit požadavky. Na jejich základě se vytvoří iterační plán, kde jsou jednotlivé požadavky rozčleněny na jednotlivé úkoly pro vývojové týmy. Vývojové týmy se potom mohou soustředit na cíle iterace a tak dosáhnout maximální produktivity. [ECCAM]

Grafické zobrazení jednoho iteračního plánu je možné vidět na Obr.5.



Obr.5: Grafické znázornění iteračního plánu [ŽÁČEK]

Iterační plán obsahuje:

- Počátek a konec iterace
- Cíle iterace a úkoly pro jejich naplnění
- Kritéria kvality
- Akceptační kritéria

Další otázkou může být, kolik iterací je potřeba a především, jak mají být dlouhé. Toto nelze obecně říci, jelikož délka a počet iterací jsou závislé na několika faktorech. Záleží především na doméně, zkušenostech týmu, zda se jedná o novou či známou technologii a v neposlední řadě, zda požadavky, které jsou na projekt kladeny, jsou také již známé či nikoliv.

3.2.2 Vykonávání

Podle projektového a iteračních plánů každý pracovník ví, co má udělat a v jakém čase. Činnost vykonávání řídí projektový manažer, který také zodpovídá celkově za průběžné plnění úkolů. Hlavním úkolem projektového manažera je projektové práce řídit a ne je vykonávat.

V této fázi se postupně naplňují projektové cíle. Cílem projektu je dosáhnout požadovaných parametrů provedení a kvality, v daném termínu, v rámci rozpočtovaných nákladů. Všechny těchto cílů musí být dosaženo současně. Je nutné znát, který ze základních cílů projektu (kvalita provedení, čas, náklady) jsou nejdůležitější nebo jestli se klade důraz na všechny základní cíle.

Dále je velmi důležitá komunikace. Všichni členové týmu musí mít dostupné informace o projektu – plán, zápisy ze schůzek, podpornou dokumentaci. Podporu musí mít všechny formy komunikace, které informují o projektu a zviditelňují projekt. Rozhodování týmu by mělo být nejlépe skupinové a členové týmu by měli být za jejich realizaci zodpovědní. Manažer projektu musí neustále získávat zpětnou vazbu o všech možných záležitostech projektu, předávat je na správné místo a zajistit případně nápravnou akci. [POSPÍŠILOVÁ]

3.2.3 Řízení a monitorování

V této fázi se provádějí činnosti, které vykonává vedoucí projektu po celou dobu jeho trvání. Jedná se především o monitorování projektu, řešení nestandardních situací, kontrola rozsahu, termínů a rozpočtu. Dále to je komunikace během projektu a v neposlední řadě rozdělování úkolů a plánování.

Řízení a monitorování v průběhu projektu je velmi důležité, jelikož průběžné monitorování změn vede k odhalení problémů, na něž je možné následně ihned reagovat. V některých situacích je nutné, aby byl systém po nějakou dobu velmi pečlivě monitorován, aby významná část vývojářů a uživatelů byla připravena vyřešit klíčové problémy, které se mohou objevit.

Ačkoliv se neznámými aspekty zabývá přímo řízení rizik (tzv. Risk Management), je nutné se jimi zabývat i v této fázi a mít eventuelní plán pro případ, že by riziko opravdu nastalo. Rizika jsou většinou nejistá, či dokonce neznámá a proto mohou zabránit úspěšnému dokončení software v dohodnutém čase. U každého rizika by měly být zaznamenány minimálně následující informace: [JULINEK]

- identifikátor
- název
- popis
- závažnost
- dopad na projekt v případě projevu rizika
- příčiny rizika
- způsob monitoringu
- způsob eliminace či přijetí
- krizový plán v případě projevu rizika
- stav rizika

3.3 Uzavření a zhodnocení projektu (iterace)

V této poslední fázi si klademe následující základní otázky: [ŽÁČEK]

- Co fungovalo, co se osvědčilo v průběhu předchozí iterace (projektu, releasu)?
- Co moc dobře nefungovalo, co se neosvědčilo v průběhu předchozí iterace (projektu, releasu)?
- Co bychom měli dělat jinak, jaké akce bychom měli podniknout za účelem zlepšení v další iteraci (projektu, releasu)?

Cílem projektu je splnit „trojimperativ“ a dosáhnout toho, aby zákazník výsledky projektu akceptoval. Akceptační kritéria musí být definovány již na začátku projektu a musí se na nich shodnout obě strany – zákazník a dodavatel. Pokud kritéria nebyla definována, vznikne na konci projektu zpravidla neshoda – zákazník požaduje víc. Během projektu může vzniknout patent, který podléhá ochraně duševního vlastnictví. Vlastnická práva je nutné ošetřit ve smlouvě mezi dodavatelem a zákazníkem.

4. Výhody a nevýhody

4.1 Výhody metodiky RUP obecně

Obecnost a rozsáhlost

RUP je založen na objektovém přístupu a současně je zde velmi silné propojení a vazba na různé CASE nástroje. Díky tomu je možné se stereotypy RUP pracovat v mnoha vyspělejších CASE a přizpůsobit si tak RUP svým potřebám.

Jako hlavní výhody této metodiky vidíme především obecnost a rozsáhlost. Jelikož RUP začíná už od úplných počátků. RUP nám radí, čím bychom měli začít, jak dále pokračovat krok po kroku, až nás dovede k nasazení výsledného produktu do užívání. RUP je také velmi univerzální, jelikož je možné tuto metodiku využít v jakémkoliv typu projektu softwarového vývoje (např. logistický software, vývoj internetového bankovníctví, systém pro státní správu,...)

Systematičnost

Další výhodu vidíme v systematičnosti a detailní propracovanosti kroků, které je třeba učinit. Model, na kterém je tato metodika založena (role, aktivita, artefakt) je velmi dobře pochopitelný. Je zde přehledné rozdělení procesu na fáze, pracovní procesy a iterace. Toto zajišťuje snadnou orientaci a rozdělení životního cyklu vývoje. Díky propracovanosti a popisu jednotlivých kroků každý pracovník ví, co a jaké výstupy se od něj čekají. RUP díky nadefinovanému workflow všech svých kroků říká, co je za každý krok od každé role očekáváno a také jaké jsou vstupy a výstupy.

Doplňkové nástroje

Kromě popisu vývojového procesu je samozřejmě součástí metodiky RUP celá řada softwarových nástrojů, které dokáží usnadnit práci. Jedná se o nástroje pro sdílená úložiště, sběr požadavků, testování. Vše je provázáno na definované procesy RUP. Dále firma Rational (nyní divize IBM) poskytuje doplňkové nástroje ve formě návodů, konceptů a studií. Především množstvím a kvalitou těchto materiálů a návodů se RUP výrazně liší od podobných metodik. Kromě komerčně dodávaných materiálů, které jsou přidávány k metodice nebo k jejím nástrojům, poskytuje IBM také množství nově vzniklých dokumentů, které jsou volně dostupné ve formě tzv. „white papers“.

4.2 Nevýhody metodiky RUP obecně

Abychom se nebavili pouze o výhodách metodiky RUP, uvedeme také několik nevýhod. Ty vlastně velmi úzce souvisí s výše uvedenými výhodami. Jelikož z nich vycházejí.

Rozsáhlost může být na škodu

I přes to, že navigace v metodice je velmi intuitivní, může se v ní nezkušený člověk velmi rychle a snadno ztratit. Je to tím, že každý proces dále odkazuje na další místa, ke každému procesu je definovaný seznam kroků, které je nutné mít hotové ještě před zahájením. Tomuto problému ovšem můžeme předejít tím, že budeme mít pro implementaci RUP člověka, který bude zbytek týmu řídit.

Příliš velká univerzálnost

Velkou nevýhodou může být i univerzálnost. Je to způsobeno tím, že RUP má za úkol obsáhnout veškeré možné případy ve vývoji softwaru, popisuje příliš mnoho kroků, které v mnohých projektech nemusí být vůbec zapotřebí a celkovou dobu trvání projektu mohou jen zbytečně prodloužit. Ovšem i tento problém se dá poměrně snadno vyřešit a to rozdělením projektu v základní verzi na velké a malé projekty.

Pokrytí pouze části celého životního cyklu

Jelikož je RUP metodika vývoje, je samozřejmé, že už neřeší procesy po zavedení software, jako například následnou údržbu, provoz, další možné rozšiřování a vyřazení z provozu. Tato nevýhoda je opět řešitelná a to pomocí vlastních rozšíření – metodikou Enterprise Unified Process, která do RUPu přidává navíc fáze provozu a údržby a následně i vyřazení z provozu.

Stejně tak můžeme říci, že RUP také žádným způsobem nedefinuje způsob řízení lidských zdrojů, jejich motivaci, školení a výběr.

Finanční náročnost

Finanční náročnost je způsobena úzkou vazbou na nástroje výhradně z rodiny IBM, které podporují pracovní procesy. Tyto nástroje jsou prodávány samostatně, poměrně za vysoké ceny (několik set až několik tisíc dolarů za licenci k jednomu nástroji). Jde o poměrně promyšlené obchodní jednání, kdy samotná metodika je prodávána zvlášť a nástroje také. Pro plné využití RUPu je tedy nezbytné si dokoupit i tyto nástroje. Ovšem v rámci konfigurace metodiky a její implementace do konkrétní organizace je samozřejmě možné vytvořit vazbu i na nástroje od jiných výrobců.

5. Závěrem...

I přesto, že má RUP své „mouchy“, stále více se rozšiřuje ve firmách, které se zabývají vývojem softwaru. Je téměř jasné, že v rukou nezkušeného člověka se může tato tradiční metodika proměnit spíše v nekontrolovatelný problém, proto je velmi důležité, aby byla v rukou zkušeného lídra. Poté se stane velmi mocnou zbraní, kterou lze poměrně dobře využít v boji s konkurencí. Důležitá je činnost implementátora metodiky, který rozhodne, co z dané metodiky jeho tým využije a co nikoliv a co je potřeba nahradit třeba jinou metodikou.

Pokud můžeme shrnout náš názor, myslíme si, že metodika RUP je velmi užitečná a dokáže projektovému týmu velmi pomoci sdělením Best Practices, jak by měl kdo, kdy a jakým způsobem postupovat. Metodika RUP je užitečná pouze v rukou zkušeného implementátora.

Zdroje:

ALDORF, Filip. *Objekty - Objekty - Metodika RUP (Rational Unified Process)*[online]. 2005 [cit. 2012-10-20]. Dostupné z: <http://objekty.vse.cz/Objekty/RUP>

BUCHALCEVOVÁ, Alena. *Metodiky budování informačních systémů*. Vyd. 1. Praha: Oeconomica, 2009, 205 s. Vysokoškolská učebnice (Oeconomica). ISBN 978-80-245-1540-3.

ECCAM. Agilní vývoj software. In: *Eccam* [online]. 2012 [cit. 2012-11-20]. Dostupné z: http://www.eccam.com/agile_process.cz.php

GIBBS, R. *Project management with the IBM rational unified process: lessons from the trenches*. Upper Saddle River: IBM Press, 2006, 287 s. ISBN 03-213-3639-9.

CHARBONNEAU, Serge. *Software Project Management -- A Mapping between RUP and the PMBOK*. IBM[online]. 2004 [cit. 2012-11-10]. Dostupné z: <http://www.ibm.com/developerworks/rational/library/4721.html>

CHARVAT, Jason. *Project management methodologies: selecting, implementing and supporting methodologies and processes for project*. Vyd. 1. New Jersey: John Wiley, 2003, 264 s. ISBN 04-712-2178-3.

CHVALOVSKÝ, Václav. *Řízení projektů: aneb překážkový běh na dlouhou trať*. 1. vyd. Praha: Wolters Kluwer ČR (Aspi), 2005. ISBN 978-80-7357-085-8.

JULINEK, Pavel. *Použití RUP pro malé SW projekty*. Brno, 2008. Dostupné z: http://is.muni.cz/th/72639/fi_m/diplomova_prace.pdf. Diplomová práce. Masarykova univerzita.

KROLL, Per a Philippe KRUCHTEN. *The rational unified process made easy: a practitioner's guide to the RUP*. Boston: Addison-Wesley, c2003, xxxv, 416 p. ISBN 03-211-6609-4.

KROLL, Per a Bruce MACISAAC. *Agility and discipline made easy: practices from OpenUP and RUP*. Upper Saddle River, NJ: Addison-Wesley, c2006, xxii, 417 p. ISBN 978-032-1321-305.

POSPÍŠILOVÁ, Radka. Projektový management. In: *Katedra telekomunikační techniky: ČVUT v Praze, Fakulta elektrotechnická* [online]. [cit. 2012-11-20]. Dostupné z: <http://www.google.cz/url?sa=t&rct=j&q=&esrc=s&source=web&cd=12&ved=0CEkQFjABOAo&url=http%3A%2F%2Fwww.comtel.cz%2Ffiles%2Fdownload.php%3Fid%3D2833&ei=qt-0ULbnFI754QSz7oDoAw&usg=AFQjCNECpdhdYyBtArd4OVYq1Y5fkg4JTg>

Řízení projektů - Project management [online]. 2005 [cit. 2011-04-22]. *Co je projekt a jaké má vlastnosti*. Dostupné z: <http://rizeni-projektu.cz/view.php?cislocclanku=2005091201>

SHUJA, Ahmad K. a Jochen KREBS. *IBM Rational Unified Process reference and certification guide: solution designer*. Upper Saddle River, NJ: IBM Press/Pearson, 2008, 336 s. ISBN 978-013-1562-929.

TEMNENCO, Vitalie. A Project Manager's RUP in review. *IBM* [online]. 2008 [cit. 2012-11-10]. Dostupné z: <http://www.ibm.com/developerworks/rational/library/edge/08/apr08/temnenco/index.html>

UPEDU [online]. Montréal: École Polytechnique, 2011 [cit. 2012-11-01]. Dostupné z: <http://www.upedu.org/>

ŽÁČEK, Jaroslav. RUP - disciplíny. In: *WWW server uživatelů na Ostravské univerzitě* [online]. Ostrava, 2009 [cit. 2012-11-22]. Dostupné z: <http://www1.osu.cz/~zacek/infos1/04.pdf>