

Objektové metody modelování

Metodika RUP, UP - Rigorózní metodiky
tvorby software a informačních systémů

Přednáška 9



**SLEZSKÁ
UNIVERZITA**

OBCHODNĚ PODNIKATELSKÁ
FAKULTA V KARVINĚ

RNDr. Zdeněk Franěk, Ph.D.

Ing. Tomáš Barčák, Ph.D.

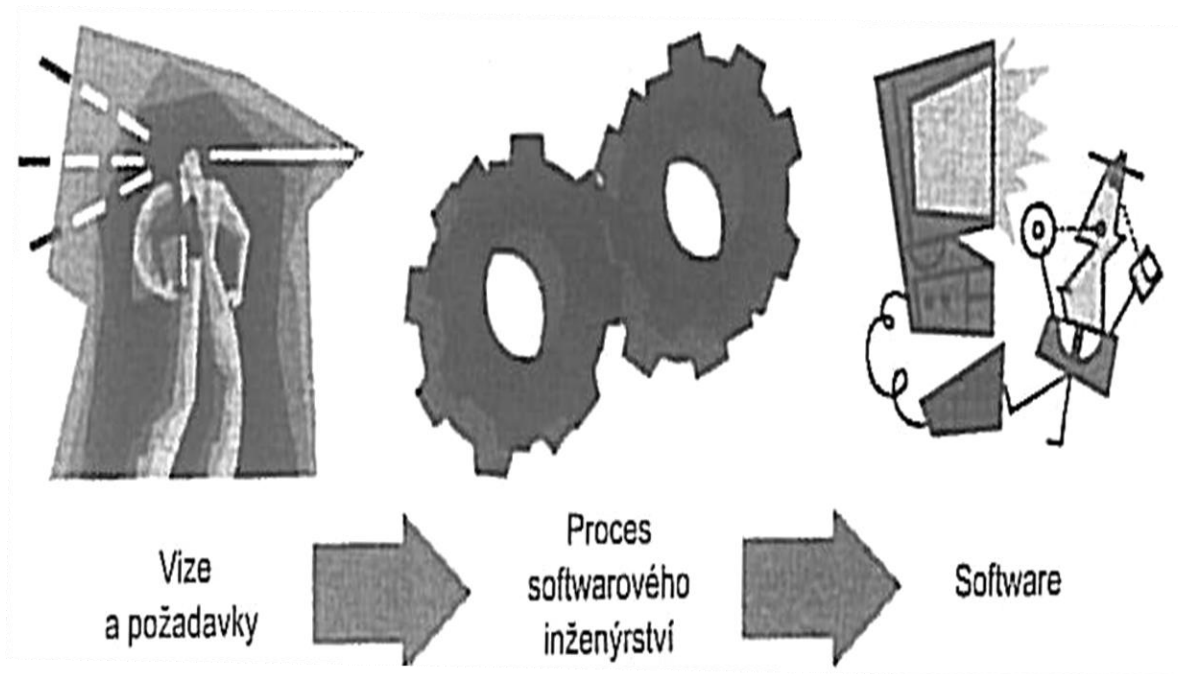
Co je to UP

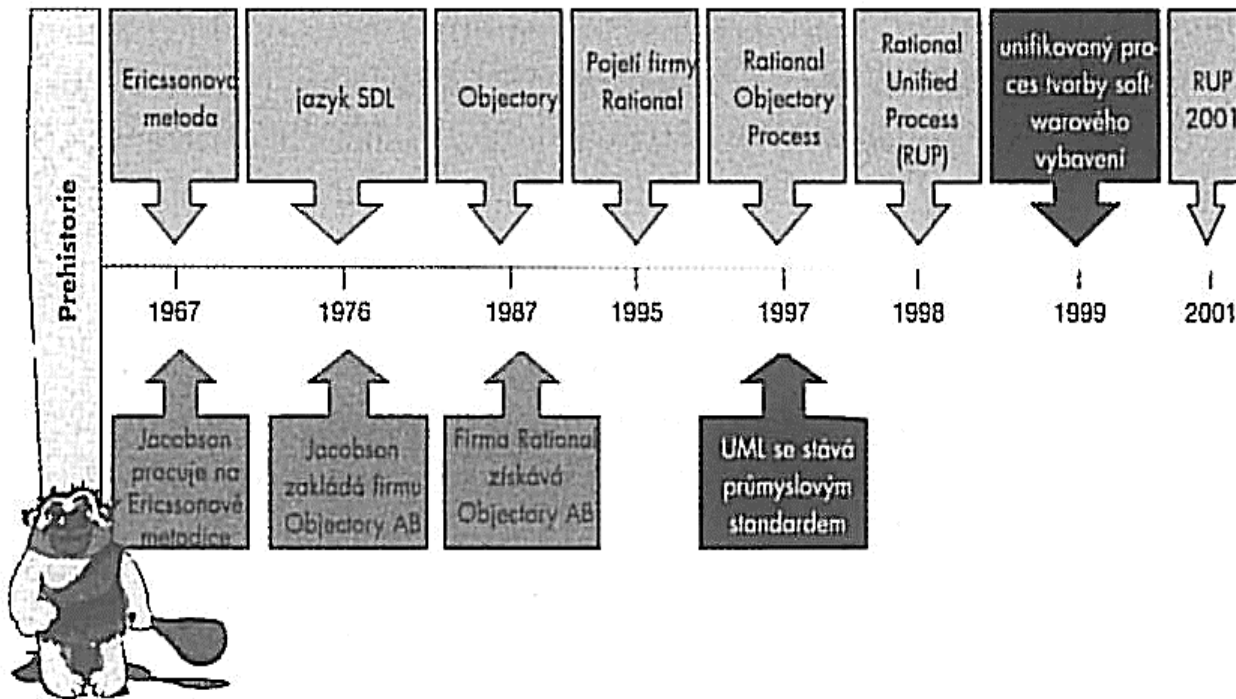


Software
engineering
process





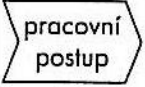
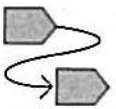
Software
development
process

- SEP nebo SDP definuje při vývoje software otázky kdo, co, kdy a jak.
- SEP je proces v němž jsou uživatelské požadavky realizovány vytvořeným softwarem viz obrázek další slide.
- Požadavek = vyjádření přání uživatele.
- UP – Unified Process, je rozsáhlá, propracovaná, objektově orientovaná, iterativní metodologie vývoje softwaru.
- UP je vyspělý otevřený standard procesu tvorby softwarového vybavení vyvinutý autory jazyka UML.
- RUP – Rational Unified Process je komerčním produktem, který rozšiřuje metodiku UP.
- Pro každý nový projekt je potřeba vytvořit novou instanci metodiky RUP; specifická komerční podtřída UP.





- Modelují aspekty kdo, kdy a co nepatrně odlišným způsobem.
- RUP vykazuje určité terminologické a syntaktické odlišnosti.
- Sémantika elementů jednotlivých metod se nezměnila.
- Mapování symbolů metodiky RUP na symboly metodiky UP.

UP	RUP	Sloupec sémantika
 dělník	← «trace» →  role	Kdo – role, kterou v projektu hraje osoba nebo tým
 aktivita	← «trace» →  aktivita	Co – jednotka práce vykonaná dělníkem (role)
 pracovní postup	← «trace» →  pracovní postup	Kdy – posloupnost souvisejících aktivit, které přinesou projektu nějaký efekt

➤ **Kdo**

- Metodika UP nabízí koncepci dělníka (worker), popisuje roli jedince, týmu uvnitř projektu.
- Metodika RUP nabízí koncepci role.

➤ **Kdy**

- Aspekty kdy jsou v UP modelovány jako aktivity = úlohy, které budou dělníci týmy vykonávat – přijímají role.
- Posloupnost aktivit je v metodice UP i RUP označována jako pracovní postup - workflow.
- Tok pracovních činností a dokumentů.

➤ **Co**

- Aspekty „co“ procesu tvorby SW jsou v UP i RUP označovány jako artefakty (artefacts).
- Artefakty jsou předměty, které do projektu vstupují nebo jsou jeho produkty, používá se termín: „Meziprodukty“.
- Např. model, element modelu (např. třída, případ užití, subsystém), zdrojový kód, spustitelná aplikace.

➤ **Jak**

- Na otázku jak, odpovídají činnosti (activities).
 - Činnost je jednotkou práce, kterou má provést worker.
 - Jasně definovaný účel ve vztahu k meziprojektu.
 - Činnost se obvykle týká jednoho pracovníka a ovlivňuje několik málo meziprojektů (artefaktů).
-

Konkrétní aplikace metodiky UP v novém projektu



- Metodika UP je obecnou metodou tvorby software.
- Pro každou organizaci a pro každý jednotlivý projekt je třeba vytvořit novou instanci.
- Uznává se, že každý SW produkt se od ostatních liší.

Definice a začlenění do:

- vnitropodnikových standardů
- šablon dokumentů
- nástrojů – překladačů, nástrojů pro správu konfigurace
- databází – sledování chyb, sledování projektu
- úprav životnosti, např. kontrola kvality bezpečnostních systémů

- Zásada řízení případem užití a rizikem.
- Zásada soustředění se na architekturu.
- Zásada iterace a přírůstku (inkrementu).

POZNÁMKA

- Případy užití jsou způsobem, jak zachytit požadavky.
- Metodika UP posuzuje stavbu SW na základě analýzy možných rizik.
- Metodika UP je založena na návrhu a vývoji robustní architektury systému.
- Architektura popisuje rozklad systému na komponenty, ale také jak se ovlivňují a jak jsou nasazeny do hardware.
- Iterační proces znamená rozklad projektu na menší podprojekty (iterace) nebo na přírůstky (inkrementy).
- Tvoříme SW v procesu postupného upřesňování našeho konečného záměru.
- Např. k analýze se vracíme několikrát (dříve jen postupně analýza, návrh, tvorba).

Iterativní a přírůstkový proces



- Menší problémy se řeší lépe, než větší.
- Rozložení velkého softwarového projektu na řadu menších miniprojektů.
- Každý takový miniprojekt je považován za iteraci, která obsahuje všechny prvky normálního SW projektu:
 - Plánování
 - Analýza a návrh
 - Tvorba
 - Integrace a testování
 - Interní nebo externí uvedení

Baseline - každá iterace generuje vlastní základní linii, která se skládá z částečné kompletní verze finálního systému a z veškeré přidružené projektové dokumentace. Základní linie jsou postupně vrstveny, až je dosažena konečná podoba vytvářeného systému.

Inkrement (přírůstek) = rozdíl mezi základními liniemi.

UP JE OZNAČOVÁN ZA **ITERATIVNÍ** a **PŘÍRUSTKOVÝ**.



5 hlavních aktivit:

- *Požadavky.* Zachycují, co by měl systém dělat.
- *Analýza.* Vybroušení požadavků a jejich strukturování.
- *Návrh.* Realizace požadavků v architektuře systému.
- *Implementace.* Tvorba softwaru.
- *Testování.* Ověření, zda implementace funguje tak, jak se od ní očekává.

➤ ITERACE

- Každá iterace generuje baseline.
- Baseline - Poskytuje smluvený základ pro další přezkoumání a vývoj, může být měněna pouze prostřednictvím formálních metod správy konfigurace a změn.

➤ PŘÍRŮSTEK (INKREMENT)

- Jsou pouze rozdílem mezi jednou a další základní linií.
- Jsou to dílčí kroky k finální odevzdávané verzi systému.

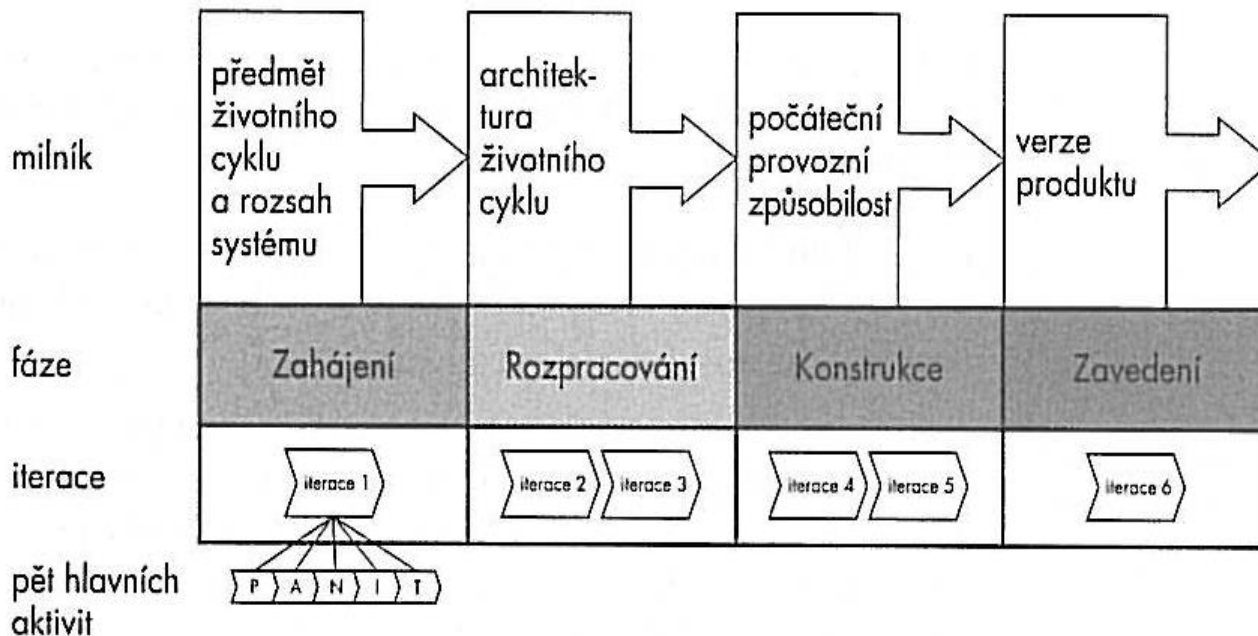
Viz publikace:

Arlow J., Neustadt I.: UML a unifikovaný proces vývoje aplikací, Computer Press a.s., 2003, str.30, ISBN 80-7226-947-X



4 základní období po sobě následující:

- Začátek (inception) – období plánování
- Rozpracování (elaboration) – období architektury
- Konstrukce (construction) – počátky provozuschopnosti
- Zavedení (transition) – nasazení produktu do uživatelského prostředí



Pět hlavních aktivit PANIT = Plánování, Analýza, Návrh, Implementace, Testování

Fáze a pracovní postupy a objem prací UP



SLEZSKÁ
UNIVERZITA
OBCHODNĚ PODNIKATELSKÁ
FAKULTA V KARVINĚ

Vysvětlivky:

I1 iterace 1

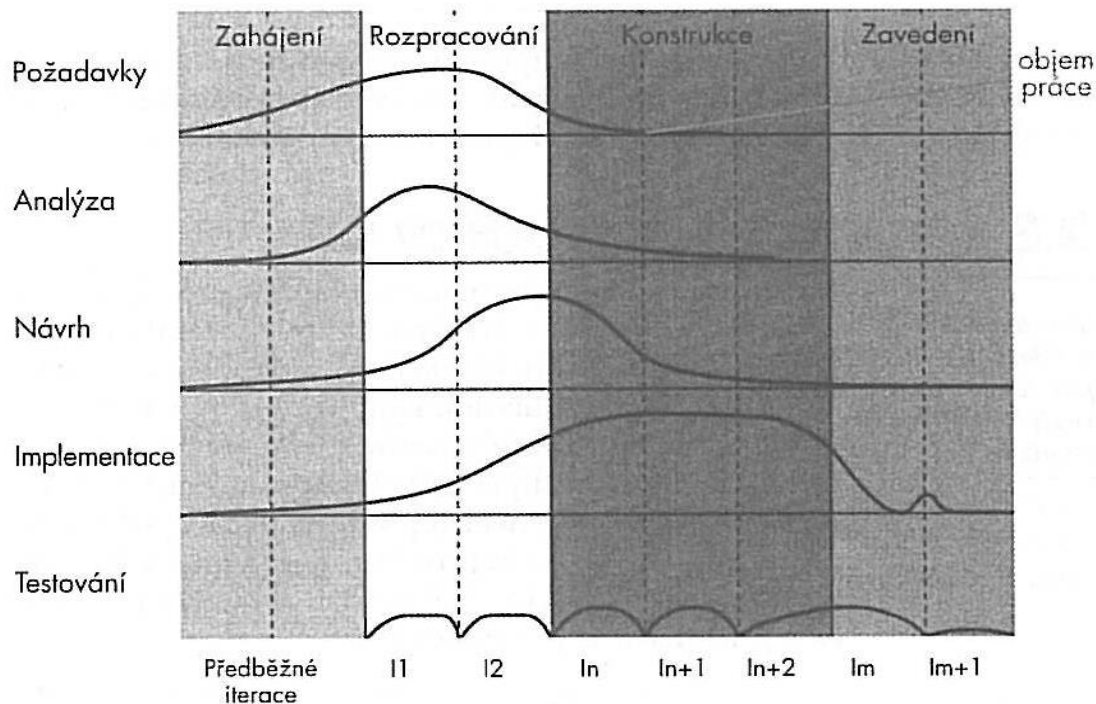
I2 iterace 2

In iterace n

In+1 iterace n+1

.....

Počet iterací je
různý podle
složitosti a
rozsahu
projektu.



Ve sloupcích
jsou fáze.

V řádcích pět
základních
pracovních
postupů.

Množství práce vykonané v pracovních postupech se liší v závislosti na dané fázi.



- **Každá fáze má svůj cíl.**
 - **Cílem fáze Začátek je „odstartování projektu“**
 - Tvorba podmínek proveditelnosti (virtual prototyping)
 - Nadnesení obchodního (podnikatelského případu)
 - Zachycení podstatných požadavků
 - Označení kritických rizik
 - Hlavní pracovníci: manažer projektu a systémový projektant
 - **Primární zaměření**
 - Hlavní důraz na pracovní postupy specifikace požadavků a jejich analýza
 - Technický prototyp, pak návrhářské a implementační práce
 - Jedinými sw artefakty jsou prototypy, není nutno testovat
 - **Milník (nastavuje určité cíle, kterých má být dosaženo)**
 - Milníkem počáteční fáze je předmět životního cyklu a rozsah systému, tzv. Life Cycle Objectives
-

Podmínky, které musí být splněny:

Hodnotící kritéria (metriky)	Je třeba dodat
Uživatelé a zainteresované osoby (zadavatel) souhlasí se záměry životního cyklu.	Dokument o představě, jenž obsahuje hlavní požadavky na projekt, funkce a podmínky.
S uživateli a zainteresovanými osobami byl dohodnut rozsah projektu. S uživateli a zainteresovanými osobami byly dohodnuty zachycené klíčové požadavky.	Počáteční případ užití (kompletní asi z 10 nebo 20 %). Slovník projektu.
Uživatelé a zainteresované osoby schválili náklady a pracovní rozvrh. Manažer projektu nadnesl obchodní případ.	Počáteční plán projektu. Obchodní případ.
Manažer projektu odhadl rizika.	Dokument nebo databáze obsahující odhad rizik.
Potvrzení proveditelnosti obsažené v technických studiích a v prototypech.	Jeden nebo více prototypů, které bude možno zahodit.
Náčrt architektury.	Počáteční dokument zachycující architekturu.



- **Cílem fáze Rozpracování je:**
 - Tvorba sputitelného architektonického základu.
 - Vylepšení odhadu rizik.
 - Definice atributů kvality.
 - ! Zachycení případů užití pro 80% funkčních požadavků !
 - Tvorba přesného plánu konstrukční fáze.
 - Formulace nabídky, prostředky, čas, vybavení, personál a náklady.
 - **Primární zaměření**
 - Požadavky – upřesnění rozsahu systému a požadavků na něj kladených.
 - Analýza – stanovení toho, co budeme tvořit.
 - Návrh – tvorba stabilní architektury.
 - Implementace – tvorba spustitelného architektonického základu.
 - Testování – testování architektonického základu.
 - **Milník této fáze je architektura (Life Cycle Architecture)**
 - Prověřujeme detailní předměty systému, architekturu a rozsah, výběr architektury a výsledek základních rizik.
-

Rozpracování (elaboration) – podmínky splnění



Hodnotící kritéria (metriky)	Co je třeba dodat
Byl vytvořen odolný robustní spustitelný architektonický základ. Spustitelný základ ukazuje, že byla rozpoznána a vyřešena důležitá rizika. Vize produktu byla stabilizována. Odhad rizik byl revidován. Obchodní případ byl revidován a odsouhlasen uživateli a zainteresovanými osobami. Projekt byl vytvořen do dostatečné hloubky, aby umožnil sestavení realistické nabídky zahrnující odhad času, peněz a prostředků pro nadcházející fáze. Uživatelé a zainteresované osoby souhlasí s plánem projektu. Plán projektu byl porovnán s obchodním případem. Bylo dosaženo dohody s uživateli a zainteresovanými osobami o pokračování v projektu.	Spustitelný architektonický základ. Statický model UML. Dynamický model UML. Model UML případu užití. Dokument o vizi. Aktualizovaný odhad rizik. Aktualizovaný obchodní případ. Aktualizovaný plán projektu. Obchodní případ a plán projektu. Konečný dokument.



- **Cílem fáze Konstrukce**
 - Splnění požadavků analýzy a návrhu a vyvinout ze spustitelného základu konečnou verzi systému.
 - Zachování integrity vytvářeného systému !!!
- **Primární zaměření**
 - Důraz na pracovní postup implementace.
 - Požadavky – odhalit požadavky, které byly v dřívějších fázích přehlédnuty.
 - Analýza – dokončit analytický model.
 - Model – dokončit návrh modelu.
 - Implementace – zajistit počáteční provozní způsobilost (Initial Operational Capability).
 - Testování – testovat počáteční funkční variantu.
- **Milník – sw systém je připraven pro testování**
 - tzv. beta verze.



Hodnotící kritéria (metriky)	Co je třeba dodat
Softwarový produkt je dostatečně stabilní a na takové úrovni, aby jej bylo možno nasadit na počítače uživatele.	Softwarový produkt. Model UML. Testovací sadu.
Uživatelé a zainteresované osoby souhlasí s nasazením softwaru do svého prostředí a je k němu připraven.	Uživatelské příručky. Popis verze.
Poměr skutečných výdajů vůči plánovaným je přijatelný.	Plán projektu.

Fáze METODIKY UP - Zavedení (transition) – nasazení produktu do uživatelského prostředí



- **Cílem fáze Zavedení**
 - Začíná v okamžiku dokončení testování a konečné nasazení systému.
 - Oprava chyb.
 - Příprava uživatelského pracoviště.
 - Přizpůsobení SW na pracovišti uživatele a úprava software v případě problémů.
 - Tvorba manuálů a dokumentace.
 - Konzultace s uživateli.
 - Koncová revize.
 - **Primární zaměření – důraz na implementaci a testování**
 - Požadavky a analýza se už nepoužívá.
 - Návrh – úprava návrhu, jsou-li během testování nalezeny chyby.
 - Implementace – přizpůsobení SW pracovišti uživatele a oprava chyb.
 - Testování – beta-testy a přijímací testy na pracovišti uživatele.
 - **Milník – PRODUKT JE UVOLNĚN A PŘIJAT DO UŽÍVÁNÍ NA PRAC. UŽIVATELE**
 - Poslední milník - beta-testy, přijímací testy a oprava chyb jsou dokončeny.
-

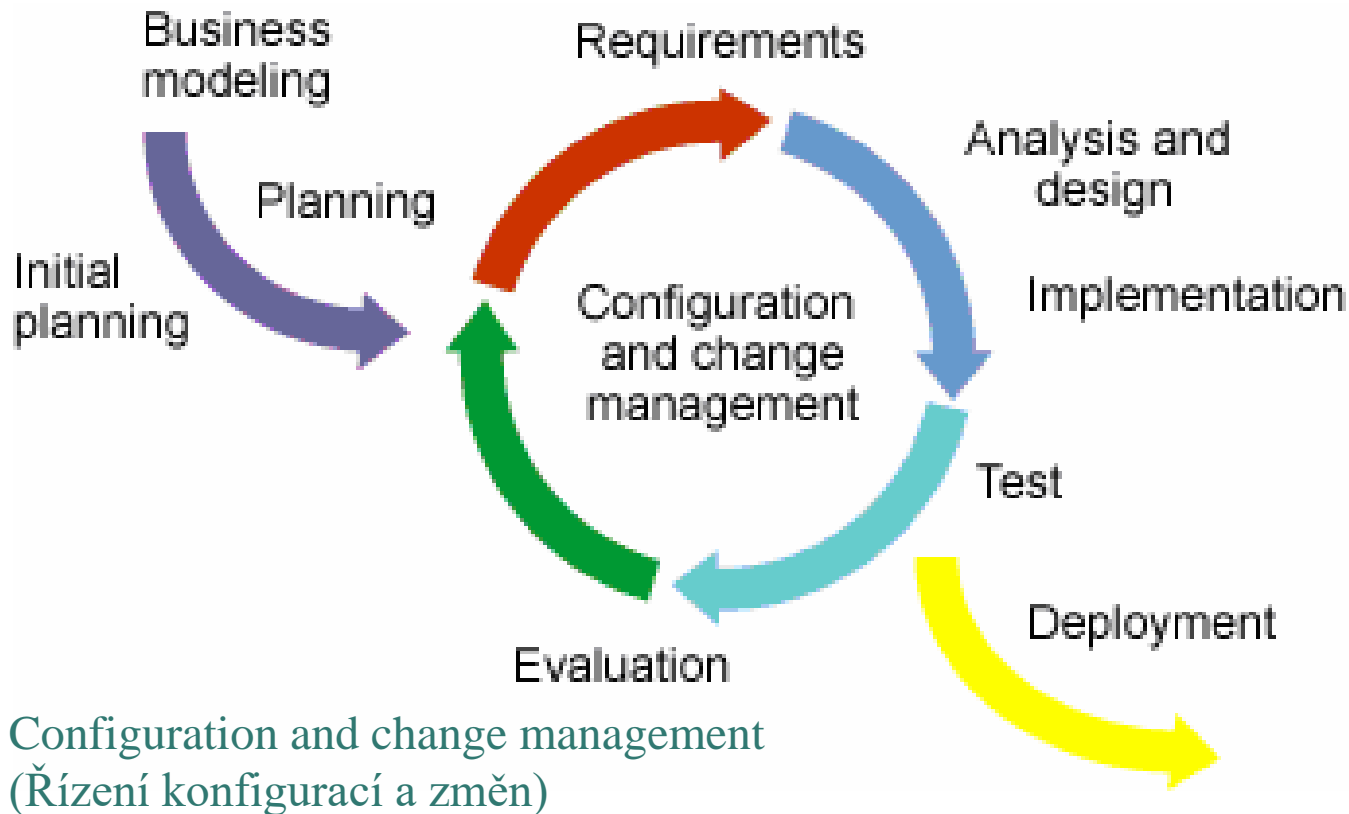


Hodnotící kritéria (metriky)	Co je třeba dodat
Beta-testy jsou dokončeny, byly provedeny nezbytné změny a uživatel souhlasí s faktem, že systém byl úspěšně nasazen. Pracovníci uživatele produkt aktivně využívají.	Softwarový produkt.
Strategie podpory produktu byla nejprve s uživateli dohodnuta a později implementována.	Plán uživatelské podpory. Uživatelské příručky.

Diagram pracovního procesu projektu vývoje SW



SLEZSKÁ
UNIVERZITA
OBCHODNĚ PODNIKATELSKÁ
FAKULTA V KARVINĚ



PŘI PRACOVNÍM
PROCESU VÝVOJE SW
SE VYUŽÍVÁ
MODELOVÁNÍ.

Fáze pracovního procesu:

Prvotní plánování
Obchodní modelování
Plánování
Požadavky
Analýza a návrh
Realizace
Testování
Vyhodnocení
Nasazení



Problémy:

- metodik je velké množství a nejsou jednotně popsány
- obtížně je lze srovnávat a vyhledat vhodnou metodiku
- mnohé metodiky se zaměřují jen na určité aspekty vývoje, jen na určité fáze životního cyklu
- většinou jsou zaměřeny jen na vývoj nového SW
- tradiční rigorózní metodiky pro současné projekty nevyhovují
- většina metodik je v angličtině



- **Objektivní důvody:**
 - různé technologie vyžadují různé techniky
 - organizace se liší firemní kulturou
 - každý jedinec je jedinečný
 - každý tým je jedinečný
 - projekty se liší velikostí
 - projekty se liší důležitostí
 - snaha prezentovat se
 - komerční důvody

Kritérium přístupu k vývoji a kritérium způsobu vývoje



Kritérium přístupu k vývoji

- Strukturované metodiky
 - RAD metodiky
 - RAD nástroje, prototypování
- Objektově orientované metodiky

Kritérium způsobu vývoje

- tradiční metodiky s vodopádovým životním cyklem
- metodiky pro iterativní a přírůstkový vývoj

Charakteristika rigorózních technik



- Těžké metodiky:
 - podrobné, hodně formalit, direktivní řízení
- Předpokládají:
 - opakovatelnost procesů
 - možnost definovat všechny požadavky na řešení předem
- Příklady:
 - OPEN, RUP, EUP, OOSP
- Metodiky pro hodnocení SW procesů:
 - (Software Process Assessment)
 - Capability Maturity Model (CMM)



Společné principy

- iterativní vývoj s velmi krátkými iteracemi
- zaměření na fungující SW, který má hodnotu pro zákazníka
- lidé jsou prvořadým faktorem – důraz na spolupráci a komunikaci
- tolerantní ke změnám
- automatizované testování



Děkuji za pozornost

Otázky?