

Objektové metody modelování

Metodika RUP, UP a agilní metodiky
tvorby software a informačních systémů

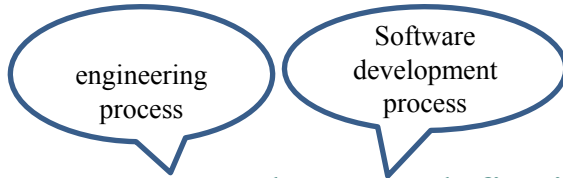
Tutoriál IV



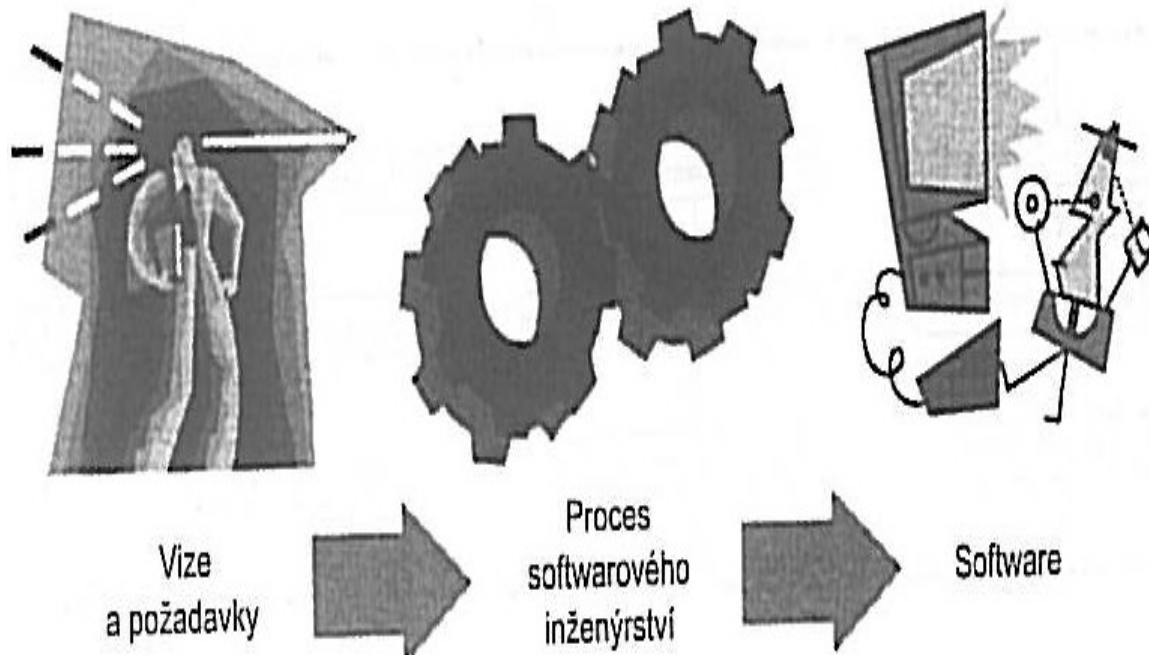
**SLEZSKÁ
UNIVERZITA**

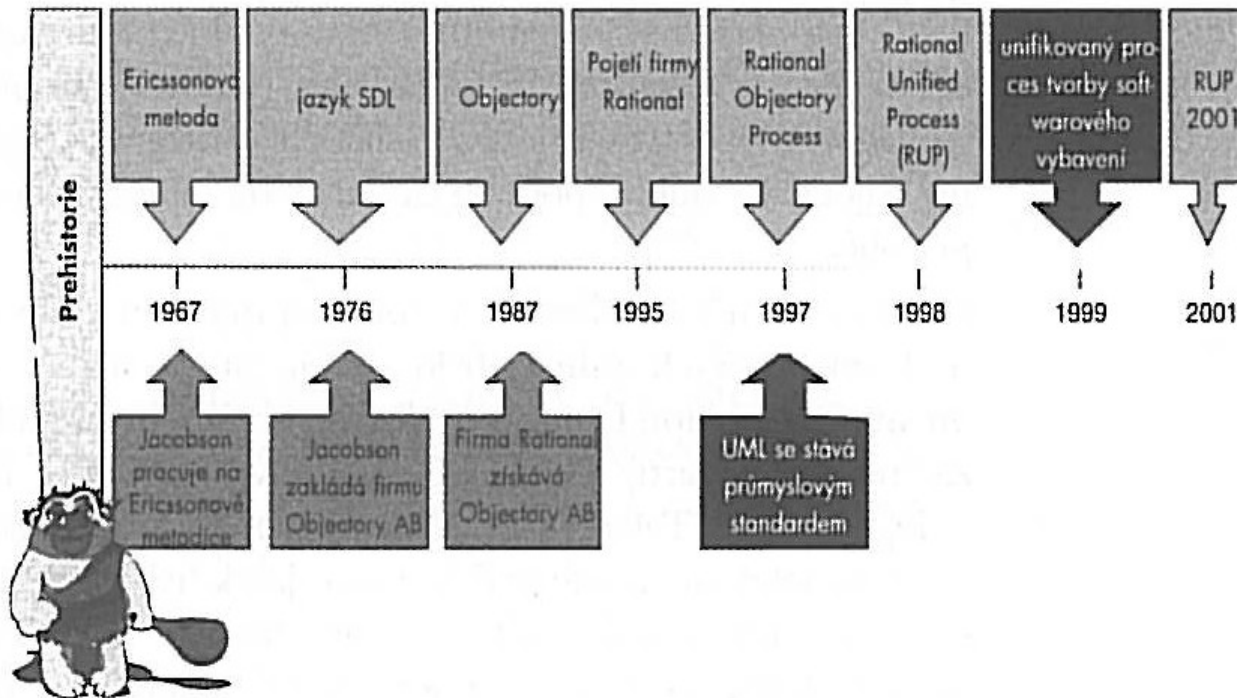
**OBCHODNĚ PODNIKATELSKÁ
FAKULTA V KARVINĚ**

RNDr. Zdeněk Franěk, Ph.D.





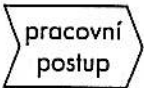
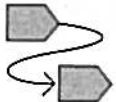


- SEP nebo SDP definuje při vývoje software otázky kdo, co, kdy a jak.
 - SEP je proces v němž jsou uživatelské požadavky realizovány vytvořeným softwarem viz obrázků další slide
 - Požadavek = vyjádření přání uživatele
 - UP – Unified Process je rozsáhlá, propracovaná objektově orientovaná iterativní 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 třeba 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
- Sématika elementů jednotlivých metod se nezměnila
- Mapování symbolů metodiky RUP na symboly metodiky UP

UP	RUP	Sloupec sémantika
 dělák	 role	<i>Kdo</i> – role, kterou v projektu hraje osoba nebo tým
 aktivita	 aktivita	<i>Co</i> – jednotka práce vykonaná dělníkem (role)
 pracovní postup	 pracovní postup	<i>Kdy</i> – 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 termínu „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 workers
 - Jasně definovaný účel ve vztahu k meziproduktu
 - Činnost se obvykle týká jednoho pracovníka a ovlivňuje několik málo meziprofuktů (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:

- 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)

POZN.

- 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:

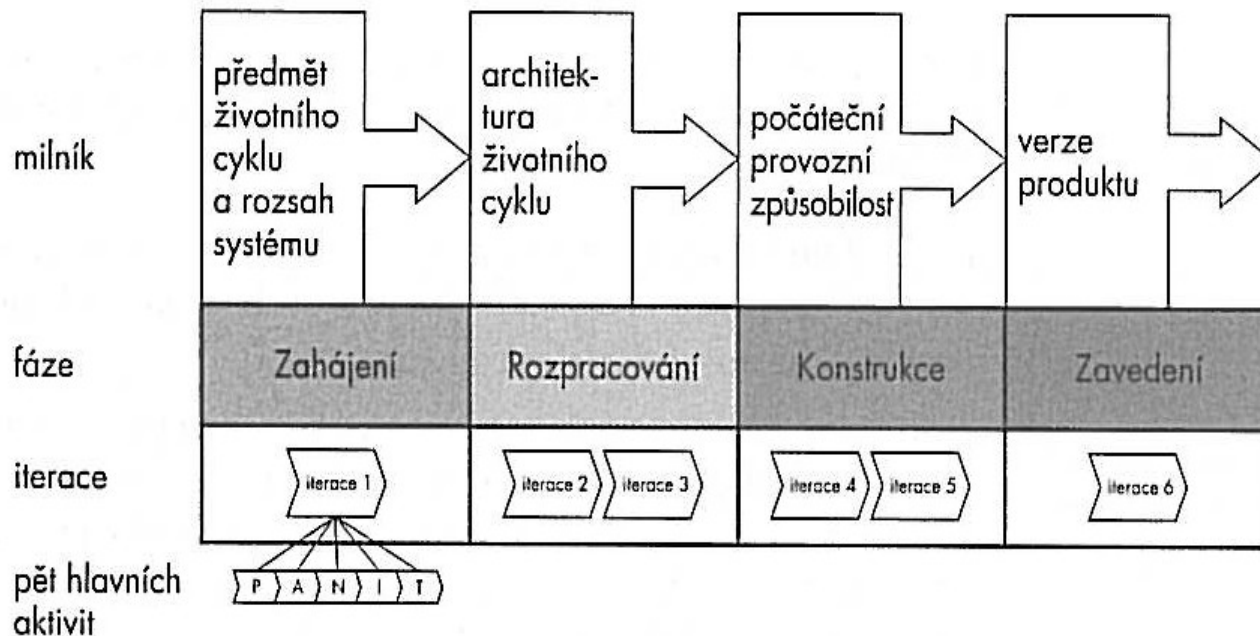
- *Požadavky.* Zachycují, co by měl systém dělat
- *Analýza.* Vybroušení požadavků a jejich strukturování
- *Návrh.* Realizace požadavků v architektuře systému.
- *Implementace.* Tvorba softwaru.
- *Testování.* Ověření, zda implementace funguje tak, jak se od ní očekává.



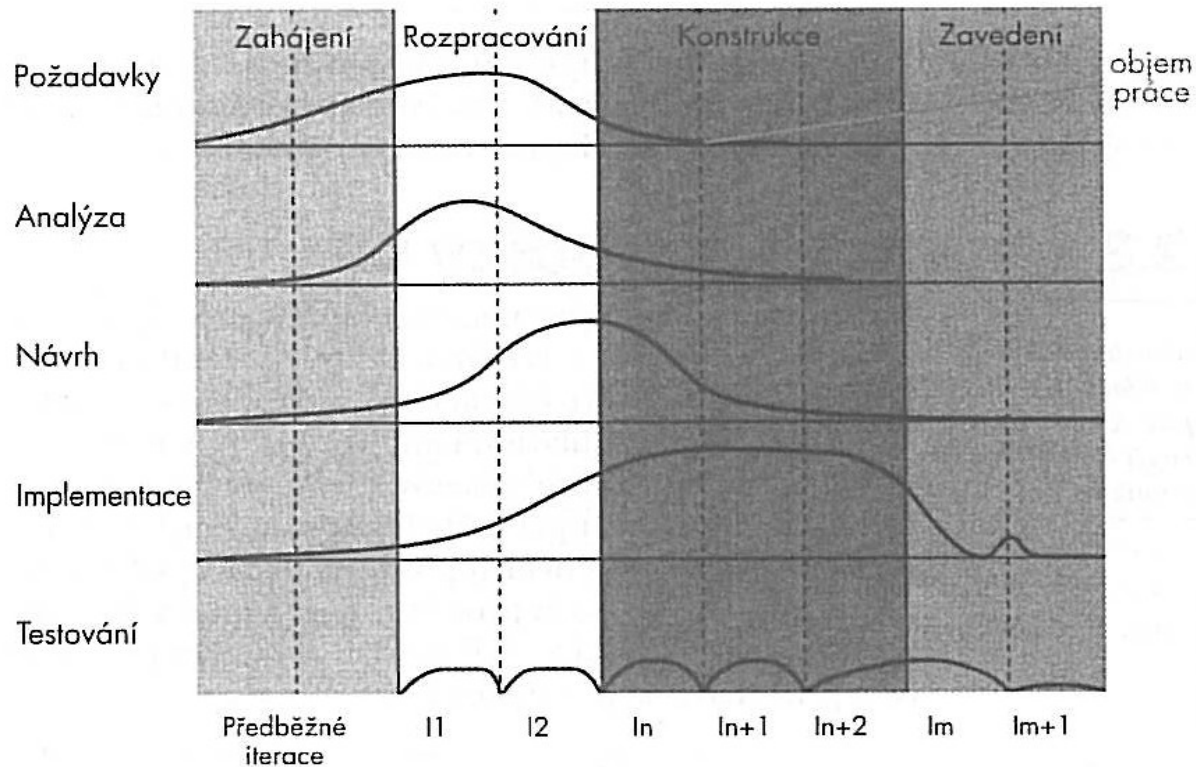
- **ITERACE**
 - Každá iterace generuje baseline.
 - Baseline - Poskytuje smluvený základ pro další přezkoumání a vývoj, může být měněna pouze prostřednictvím formálních metod správy konfigurace a změn

- **PŘÍRŮSTEK (INKREMENT)**
 - Jsou pouze rozdílem jednou a další základní linií
 - Jsou to dílčí kroky k finální odevzdávané verzi systému

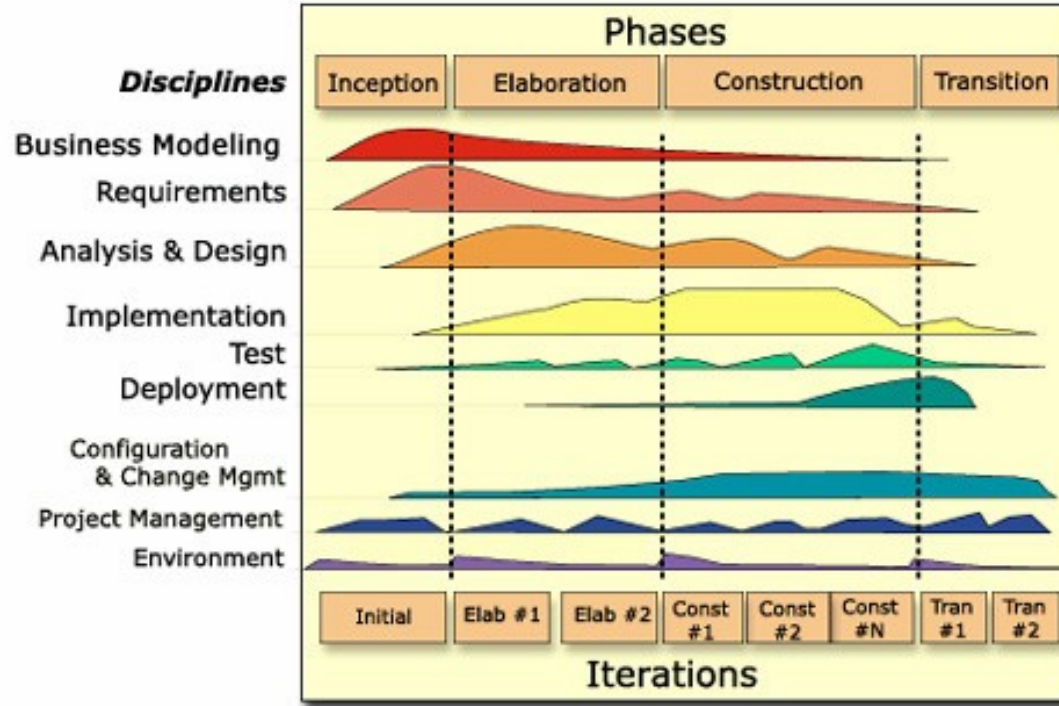
- 4 základní po sobě následující
 - Začátek (inception) – období plánování
 - Rozpracování (elaboration) – období architektury
 - Konstrukce (construction) – počátky provozuschopnosti
 - Zavedení (transition) – nasazení produktu do uživatelského prostředí



Fáze a pracovní postupy a objem prací UP



Fáze a pracovní postupy a objem prací RUP





- **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:**
-

Začátek (inception) – podmínky splnění



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.	Počáteční případ užití (kompletní asi z 10 nebo 20 %).
S uživateli a zainteresovanými osobami byly dohodnuty zachycené klíčové požadavky.	Slovník projektu.
Uživatelé a zainteresované osoby schválí náklady a pracovní rozvrh.	Počáteční plán projektu.
Manažer projektu nadnesl obchodní případ.	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**
 - 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
-



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



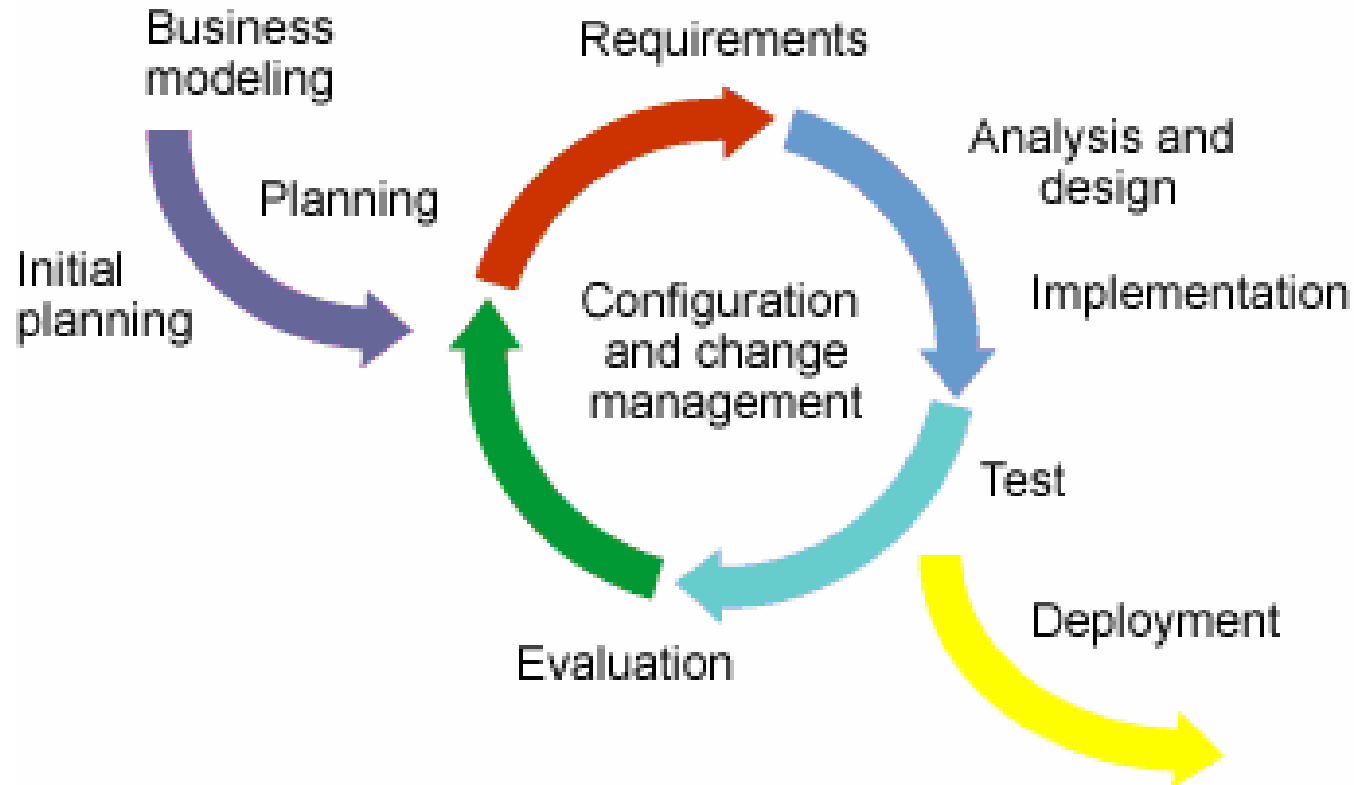
Hodnotící kritéria (metriky)	Co je třeba dodat
Softwarový produkt je dostatečně stabilní a na takové úrovni, aby jej bylo možno nasadit na počítače uživatele.	Softwarový produkt. Model UML. Testovací sadu.
Uživatelé a zainteresované osoby souhlasí s nasazením softwaru do svého prostředí a je k němu připraven.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

- **Cílem fáze Zavedení**
 - Začíná v okamžiku dokončení testování a konečné nasazení systému
 - Oprava chyb
 - Příprava uživatelského pracoviště
 - Přizpůsobení sw na pracovišti uživatele
 - 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, jsouli 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



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 živ.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

- 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



- 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.

Hodnota pro zákazníka

jestliže SW musí dodávat hodnotu pro zákazníka, kdo může nejlépe určit, jaká hodnota to je?

vývojáři ne, ale zákazníci ano,

přesun zodpovědnosti na zákazníka

zákazník určuje a mění priority funkcí

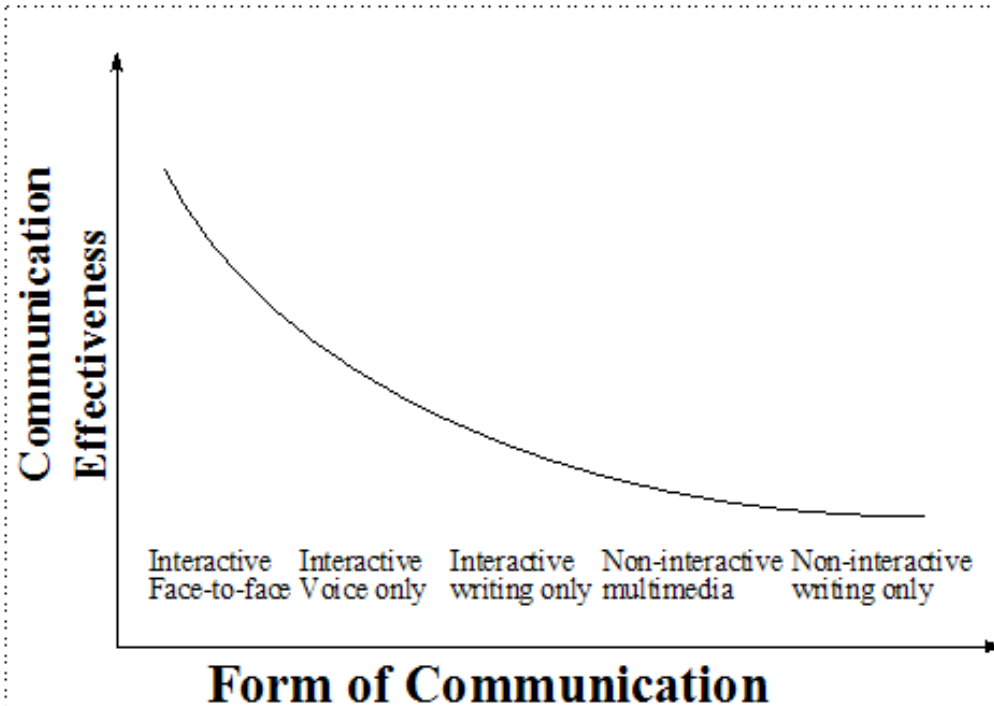
zákazník nevidí do budoucna,

proto my, vývojáři musíme do systému zabudovat funkce a data , které by mohl zákazník v budoucnu potřebovat

X

je lepší vytvořit systém tolerantní ke změnám, který umí akceptovat změny v budoucnu

Osobní komunikace





Rigorózní metodiky

- postaveny na nedůvěře.
- *Nevěřím ti, že uděláš práci správně, tak tě musím neustále sledovat a kontrolovat.*

Agilní metodiky

- vůdcovství a spolupráce - jsou formovány na důvěře a respektu.
- *Věřím ti, že uděláš práci dobře, a tak budeme spolupracovat, abychom dosáhli výsledku.*



Rigorózní metodiky

- standardizují lidi v organizaci
- snaží se vykázat lidi do role zaměnitelné součástky
- čím větší projekty se realizují, tím více specialistů je třeba zapojit

Agilní metodiky

- využívají individualit a silných stránek lidí
- požadují integraci znalostí, stálá interakce a kooperace, sdílení znalostí v týmu, týmové řešení problému.



Rigorózní metodiky

- standardizují lidi v organizaci
- snaží se vykázat lidi do role zaměnitelné součástky
- čím větší projekty se realizují, tím více specialistů je třeba zapojit

Agilní metodiky

- využívají individualit a silných stránek lidí
- požadují integraci znalostí, stálá interakce a kooperace, sdílení znalostí v týmu, týmové řešení problému.

Které metodiky řadíme mezi agilní?



- Adaptive Software Development (ASD),
- Dynamic Systems Development Method (DSDM),
- Feature-Driven Development (FDD),
- Extreme Programming (XP),
- Lean Development,
- SCRUM,
- Crystal metodiky,
- Agile Modeling

Porovnání rigorózních a agilních metodik



**SLEZSKÁ
UNIVERZITA**
OBCHODNĚ PODNIKATELSKÁ
FAKULTA V KARVINĚ



rigorózní metodiky

předpoklady

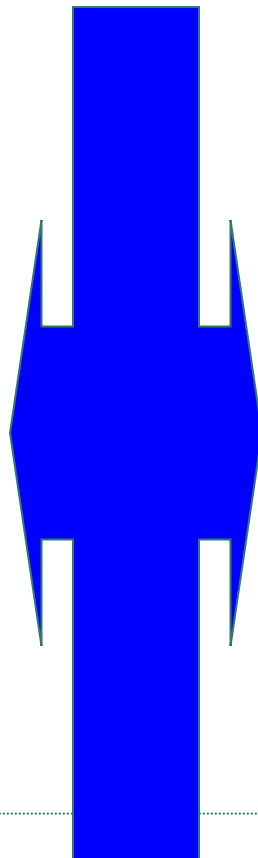
- SW procesy lze popsat
- požadavky je možné definovat předem

obsah

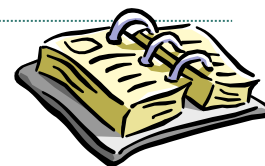
přesně definované
procesy, činnosti,
artefakty

použití

standardní projekty,
velké projekty



agilní metodiky



- SW procesy nelze popsat
- předem jen hrubé požadavky

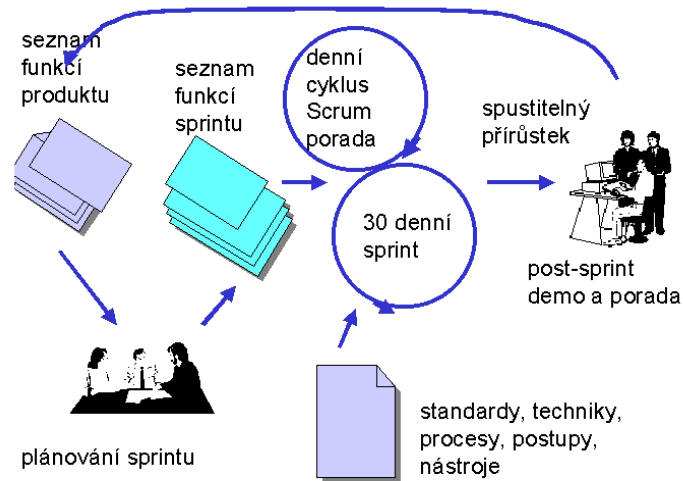
jen generativní pravidla,
praktiky a principy

výzkumné projekty
time-to-market

menší týmy

- vývoj SW není definovaný proces, který je možné přesně popsat a opakovat, ale empirický proces
- empirický proces vyžaduje odlišný styl řízení - vyžaduje **konstantní**

monitorování a adaptaci



SCRUM meetings



- umožňují monitorovat stav,
- konají se ve stejný čas na stejném místě
- trvají méně než 30 minut (cílem je 15 minut)
- vede je *Scrum master*
- účastní se jich všichni členové týmu (vývojáři, uživatelé , testeři,..)
- navštěvují je manažeři, aby věděli o stavu, ale aktivně se neúčastní
- slouží ke zjištění problémů, ale ne k jejich řešení
- každý účastník musí zodpovědět 3 otázky
 - co udělal od poslední scrum porady
 - co bude dělat do příští scrum porady
 - jaké překážky mu stojí v cestě

- modelování základní částí vývoje SW at' v lehkých nebo těžkých metodikách
- **dva primární důvody, proč modelovat:**
 - abyste pochopili podstatu toho, co vytváříte,
 - abyste mohli komunikovat v týmu
- je bohužel obětí mýtů a nepochopení

Nesprávné tvrzení o modelování



Modelování je vždy svázáno s dokumentací

- model je abstrakce, která popisuje jeden nebo více aspektů problému a možné řešení problému
- tradičně jsou modely chápány jako diagramy a odpovídající dokumentace

Realita

- nevizuální artefakty jako CRC karty, textový popis byznys pravidel jsou považovány také za modely
- při modelování nejde o model jako takový, ale o proces modelování
- většina modelů není součástí oficiální dokumentace systému, ale některé modely je dobré uchovat - persistentní modely

Nesprávné tvrzení o modelování)



Je možné vše namodelovat předem a správně

- snaha namodelovat vše předem a správně a zmrazit požadavky před začátkem kódování

Realita

- nemá smysl modelovat ve všech podrobnostech
- programátoři příliš neuznávají modely
- základem vývoje SW je iterativní přístup - trochu modelování, trochu kódování, trochu testování a hlavně nasazení fungující verze SW a zpětná vazba od zákazníka

Nesprávné tvrzení o modelování



Při modelování je nutné používat CASE nástroj

Ten nejlépe zachytí všechny aspekty modelu a zajistí konzistenci mezi modely

Realita

- daleko efektivnější je vytvářet jednoduché modely, které zachycují jen podstatné informace
- a používat jednoduché nástroje

Nesprávné tvrzení o modelování



Při vývoji softwaru je třeba zmrazit požadavky

Při zmrazení požadavků na začátku životního cyklu nejspíš dodáte to, co bylo požadováno, ale pravděpodobně ne to, co je třeba.

Realita

- dochází ke změnám, mění se požadavky
- protože se mění byznys priority
- protože zákazník po dodání části systému jej lépe pochopí
- lepší než zmrazit požadavky a riskovat neúspěch je uchopit změnu a odpovědět na ni

Návrh je vytesán do kamene

Realita

- nikdo, ani nejlepší návrhář není dokonalý a stejně tak jeho dílo, nemá smysl fixovat chyby
- pokud nechceme zmrazit požadavky, není možné zmrazit návrh
- návrh je hotov, až po dodání kódu

Nesprávné tvrzení o modelování



Modelování je ztráta času

- častý názor nových vývojářů, kteří se orientují jen na kódování

Realita

- často se produktivita vývojáře zvýší náčrtnem diagramu, vytvořením prototypu UI atd.
- produktivní vývojáři modelují před psaním kódu

Mýty o modelování

Základem je datové modelování

- datová komunita má politickou sílu

Realita

- datové modelování je důležitý, ale sekundární úkol modelování



Nesprávné tvrzení o modelování

Všichni vývojáři umí modelovat

Realita

- Umění modelovat vyžaduje velké zkušenosti

Nesprávné tvrzení o modelování



**SLEZSKÁ
UNIVERZITA**
OBCHODNĚ PODNIKATELSKÁ
FAKULTA V KARVINĚ

Modelování je spojeno s rigorózními metodikami

Realita

- je možné modelovat agilním způsobem – používat jednoduché modely a jednoduché nástroje



Děkuji za pozornost

Otázky?