

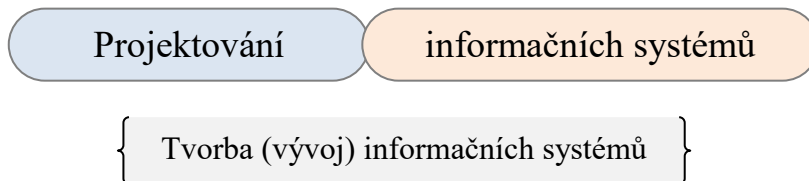
Předmět:	Projektování IS I
Téma:	Metodiky a metody vývoje IS

Vyučující:	dr. Dušan Kajzar	Školní rok: 2020/2021
------------	------------------	-----------------------

Obsah:

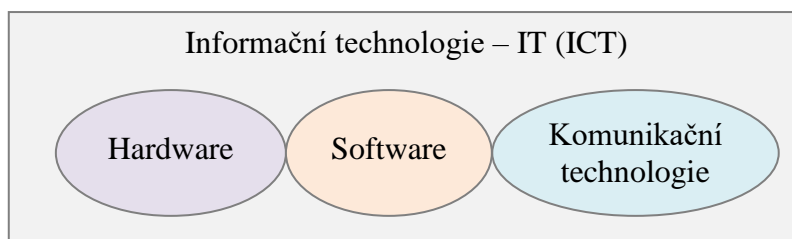
1. Základní pojmy	1
2. Metodiky, metody, techniky, nástroje	5
3. Životní cyklus IS	7
4. Typy životních cyklů IS	9
5. Obecné principy analýzy a návrhu IS	11
6. Druhy metodik a trendy v jejich rozvoji	13
7. Agilní metodiky vývoje IS	14
8. Projektování IS a řízení projektů IS	16
9. Závěr k tématu	16

1. Základní pojmy



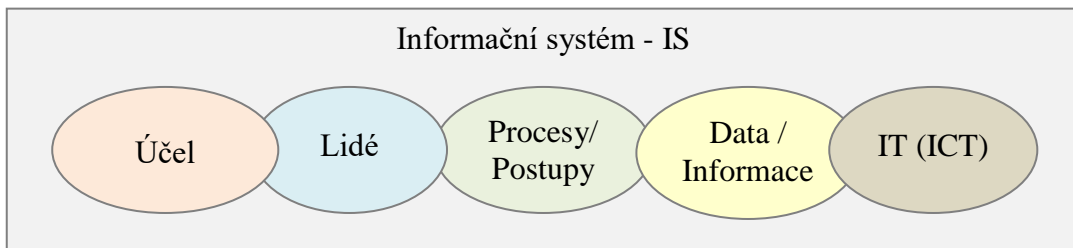
Informační technologie (IT, ICT):

- na **výpočetní technice** založené **prostředky a postupy** pro sběr, uchování, zpracování a přenosy informací,
- složky IT – viz obrázek.



Informační systém (IS):

- účelové využití informačních technologií pro zpracování určité třídy úloh (aplikací),
- složky IS – viz obrázek.



Příklad:

- IT ... PC + OS Windows + MS Excel,
- IS ... Zpracování plánu investic na příští rok.



Pojem aplikace (SW aplikace, úloha):

- funkční subsystém IS řešící danou oblast v podniku,
- uživatelský pohled na informační systém/subsystém,
- např. aplikace pro účetnictví, studijní agendu, ...



Projekt:

- pojem projekt
 - množina souvisejících činností,
 - společný cíl, strategický plán,
 - přidělené zdroje (materiál, lidé, finance, čas),
 - řízení (manažer), zadavatel (odběratel),
- výsledek projektu
 - materiální – budova, strojní vybavení, ...
 - nemateriální – kulturní akce, SW aplikace, ...
- charakteristické znaky projektu
 - definovaný začátek a konec,
 - projektový tým a jeho proměnlivé složení,

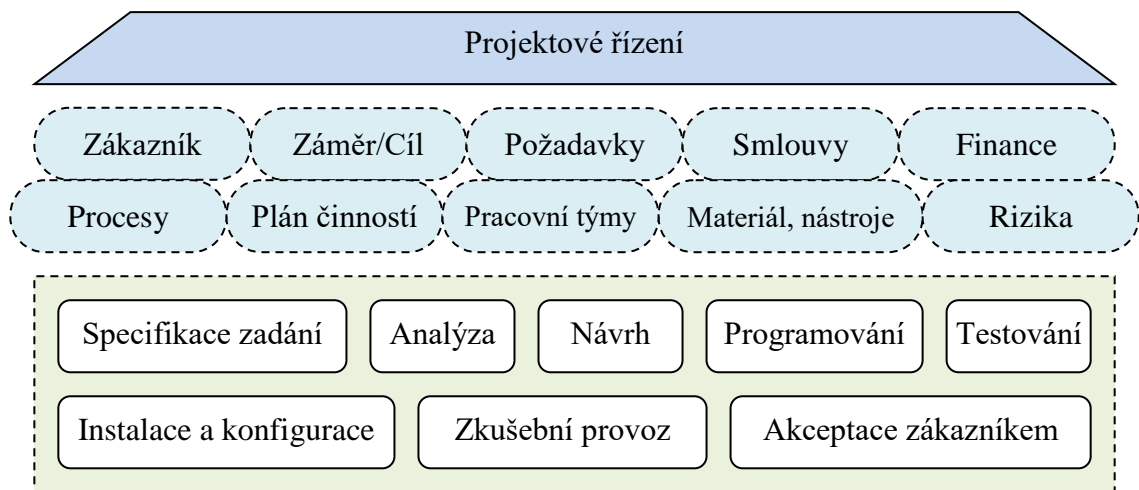


- určitá jedinečnost a neopakovatelnost činností,
- vysoká neurčitost, velká rizika,
- konstatování úspěchu – až při ukončení projektu,
- projektové řízení vs. provozní řízení (viz skriptum 5. ročník).

Projektování IS:

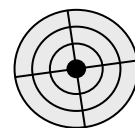
- množina činností s cílem vyvinout IS a zavést jej do používání v praxi,
- tvorba (vývoj) IS – probíhá v rámci projektu (jako projekt),
- řízení je zastřešeno tzv. projektovým managementem.

Vztah mezi řízením projektu a projektováním IS:



Proč proces vývoje IS jako projekt?

- definovaný cíl,
- časová omezenost vývoje (termíny začátku, ukončení),
- určitá „neopakovatelná“ činnost,
- omezené zdroje (lidé, finance, technika, ...),
- ...
- nikoliv permanentní „plichtění“ a „lepení“ něčeho.



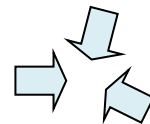
K historii pojmu „projektování IS“:

- analogie se stavebnictvím
 - koncepční + realizační práce,
- stírání hranic mezi koncepcí a realizací
 - iterativní charakter činností během vývoje IS,
 - používání moderních CASE nástrojů.



Co musíme brát v úvahu při projektování IS?

- různé **úhly pohledu** na proces vývoje IS
 - technický (HW, SW a síťová architektura),
 - technologický (provozní technologie = pracovní postupy),
 - organizační (organizace práce na pracovištích),
 - ekonomický (finance, návratnost investic),
 - sociální a psychologické vlivy,
 - => při vývoji IS nelze podcenit žádný z těchto aspektů,
- chápání IS jako systému „**sociálně-technického**“
 - nikoliv pouze „systém technický“ - HW, SW, komunikace, data, ...
 - sociální (společenské) prvky - účel použití, uživatelé, postupy práce, organizace práce, ...
- **různý význam** IS pro podnik
 - podpůrný, provozní, strategický.



Standardizované postupy během vývoje IS:

- z předchozího => vývoj IS by měl probíhat podle určitých pravidel, vhodných a ověřených postupů,
- Proč? Abychom něco významného neopomenuli!
Použijme osvědčené postupy (know-how). Ale jaké?
- **standardizované postupy** = předem definované, optimalizované, řízené,
- => obsažené v tzv. **metodikách a metodách vývoje IS.**

2. Metodiky, metody, techniky, nástroje

Základní pojmy:

- metodika
 - doporučený souhrn etap, přístupů, zásad, postupů, pravidel, metod, technik, nástrojů, dokumentů, metod řízení, ...
 - udělat - co, kdo, kdy, proč ...
- metoda
 - zaměřená na určitou etapu (etapy) vývoje IS,
 - obvykle spojená s přístupem (strukturovaným, objektovým),
- technika (jak?)
 - postupy kroků, jak se dostat k výsledku,
 - např. normalizace datového modelu, prototypování, ...
- nástroj (čím?)
 - prostředek k provedení něčeho, k zobrazení výsledku apod.,
 - např. modely systému, CASE nástroje.

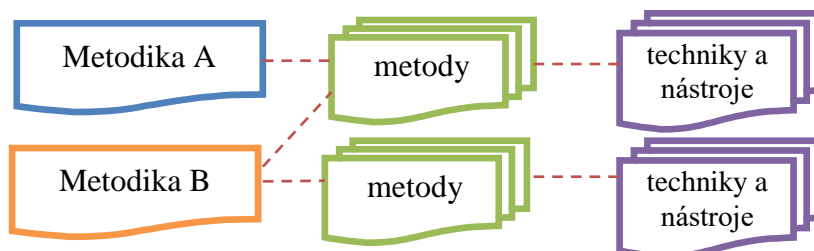


Poznámka k základním pojmům:

- **metoda** - z řeckého met-hodos, hodos (cesta), "za cestou", sledování, stopování, cesta za něčím,
- **metodika** - je obecně pracovní postup (metoda) nebo nauka o metodě,
- **metodologie** - je vědní disciplína, která se zabývá metodami, jejich tvorbou a aplikací.

Pojmy v našem slova smyslu:

- nejednoznačnost pojmů v literatuře (překlad termínu „methodology“),
- vazby mezi metodikou, metodami, technikami a nástroji.



Vznik a rozvoj metodik a metod:

- metodiky a metody jakožto doporučené a osvědčené postupy, pravidla, ...
- zobecněním z praxe, na akademické půdě,
- univerzálnost vs. specifická.

Možné zaměření metodik, metod, technik a nástrojů:

- na **různé typy** vyvíjených IS
 - vývoj aplikací v prostředí klient/server,
 - vývoj manažerských IS a IS v prostředí datových skladů,
 - modelování a optimalizaci podnikových procesů,
 - vývoj WWW aplikací,
- na kategorie IS „**obecný vs. speciální IS**“
 - specializované (zákaznické) IS – jednoúčelové nebo odvětvově zaměřené,
 - typové (krabicové) IS - výběr, kustomizace (přizpůsobení potřebám zákazníka).
- na velikost a organizaci práce **vývojových týmů**, např.
 - velké týmy – rigorózní (klasické, ortodoxní) metodiky,
 - malé týmy - agilní metodiky.

Konkrétní výběr vhodné metodiky a metod:

- v závislosti na typu a kategorii IS,
- na rozsahu IS (jednoduchá, složitá architektura),
- na významu IS pro podnik (podpůrný, klíčový),
- na velikosti a možnostech vývojového týmu, atd.

Shrnutí - metodika standardizuje proces vývoje IS:

- postup vývoje IS je předem definovaný, optimalizovaný, řízený,
- nikoliv „ad-hoc“ (jak koho napadne).

Z pohledu vedoucího vývojového týmu – použití standardizovaného postupu umožňuje:

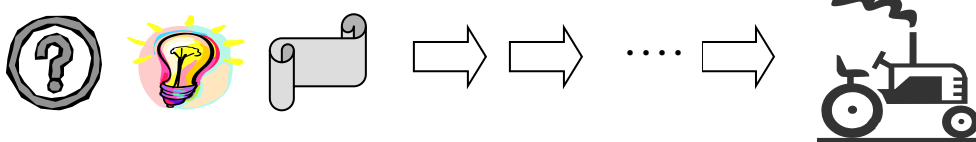
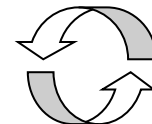


- vyšší kvalitu řízení prací na projektech vývoje IS
 - vedoucí má představu o celém procesu vývoje,
 - plánování, organizování, vedení, výběr a úkolování pracovníků, kontrola,
- naplánovat co, kdo a kdy bude provádět
 - => specializace projektových týmů na jednotlivé etapy vývoje IS,
 - => předpoklady pro vyšší produktivitu práce vývojových týmů,
- zajistit značnou nezávislost vyvíjeného IS na konkrétních řešitelích
 - dodržování standardů, evidence a dokumentace prací, ...
- definovat kritéria kvality pro každou etapu vývoje IS (základ pro řízení kvality procesu vývoje),
- stanovit pravidla pro tvorbu kvalitní projektové dokumentace,
- snížení rizik spojených s vývojem IS
 - z hlediska zdrojů, rozsahu řešení, času řešení,
- definovat kritéria pro výběr a použití CASE nástrojů.

3. Životní cyklus IS

Životní cyklus IS:

- průchod životními (vývojovými) etapami IS,
- hovoříme o tzv. etapách životního cyklu IS,
- význam slova „cyklus“.



Etapy (fáze) životního cyklu IS:

- poplatné používané metodice vývoje IS,
- metodika = doporučený postup vývoje IS,
- => členění celého procesu na tzv. životní etapy,
- etapy vývoje IS vs. etapy vývoje a používání IS.

Příklady etap životního cyklu IS:

- obecný příklad etap

specifikace zadání -> analýza systému -> návrh systému ->
implementace -> testování -> zavedení do provozu -> provoz systému,

- etapy podle MDIS

IST -> ÚST -> GAN -> DAN -> IMP -> ZAV -> PÚR,

- členění podle UP – dvojrozměrné schéma

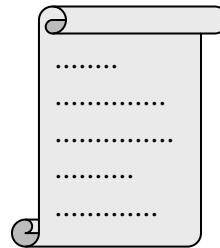
zahájení -> rozpracování -> konstrukce -> zavedení,

specifikace -> analýza -> návrh -> implementace -> testování,

- obsah metodiky => struktura životního cyklu IS + popis etap (fází).

Zpracování obsahu každé etapy:

- cíle (účel) etapy,
- předpoklady pro zahájení etapy,
- činnosti v průběhu etapy,
- dokumentace etapy,
- kritické faktory (rizika) etapy,
- kritéria ukončení etapy.



Otázka podrobnosti obsahu etap v dané metodice:

- vývojář ocení spíše úplnost principů, doporučení, postupů, ...
- co mám dělat od začátku vývoje IS do konce,
- nikoliv nutně detailní popis všech činností,
- ovšem
 - metodiky mnohdy končí návrhem IS či implementací,
 - v našem kurzu – projdeme cyklus až k „provozu a údržbě IS“.



Cyklické návraty a jejich příčiny:

- (nekvalitně provedené činnosti v některé etapě vývoje IS),
- věčné:
 - předmět analýzy je příliš složitý, nutno poznávat postupně,

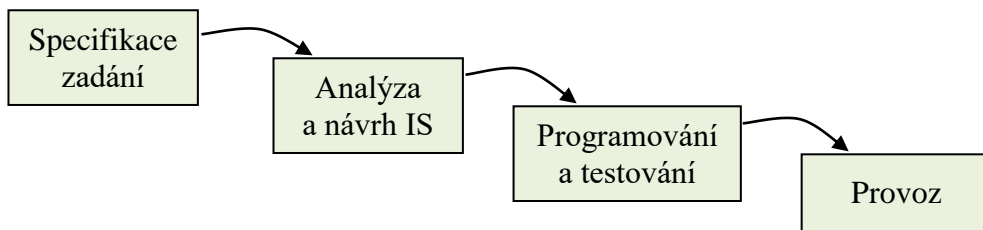
- mění se požadavky na IS - subjektivní i objektivní faktory,
- časové:
 - realizace IS jako celku – trvala by příliš dlouhou dobu,
- finanční:
 - jednorázová investice – příliš velká,
- organizační:
 - technologické a organizační změny – nutno provádět postupně.

4. Typy životních cyklů IS

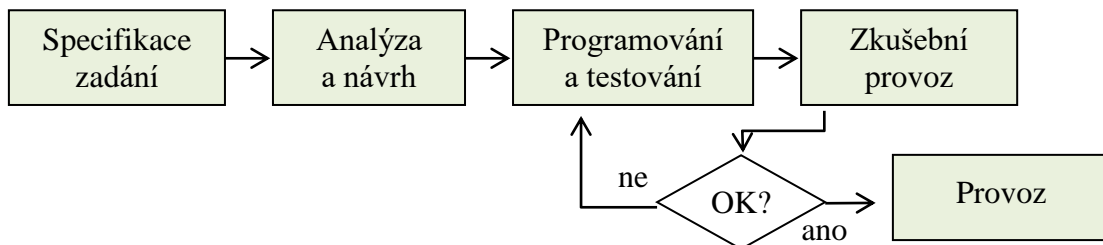
Metodika stanoví:

- rozdělení vývoje IS na etapy a obsah jednotlivých etap,
- povolené návraty k etapám předchozím a podmínky návratů,
- jinými slovy – doporučí základní model živ. cyklu IS,
- model živ. cyklu IS – schéma posloupnosti etap,
- základní vzorové modely – viz další obrázky.

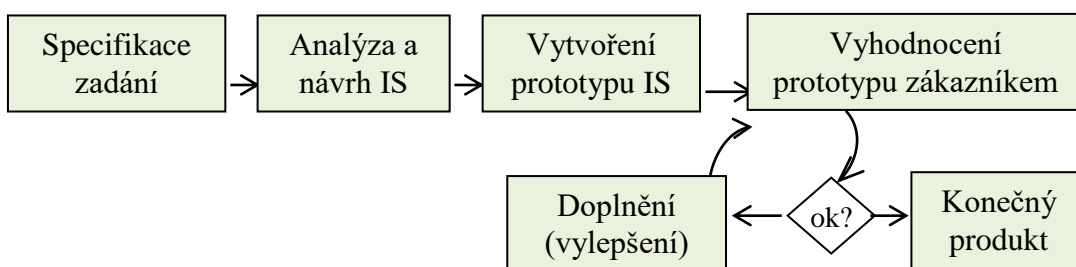
Model „vodopád“ (waterfall):



Postupné programové přírůstky (inkrementy):

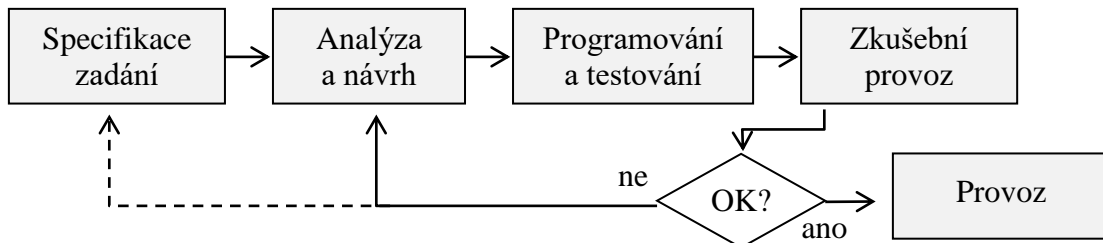


Model „prototyp“:



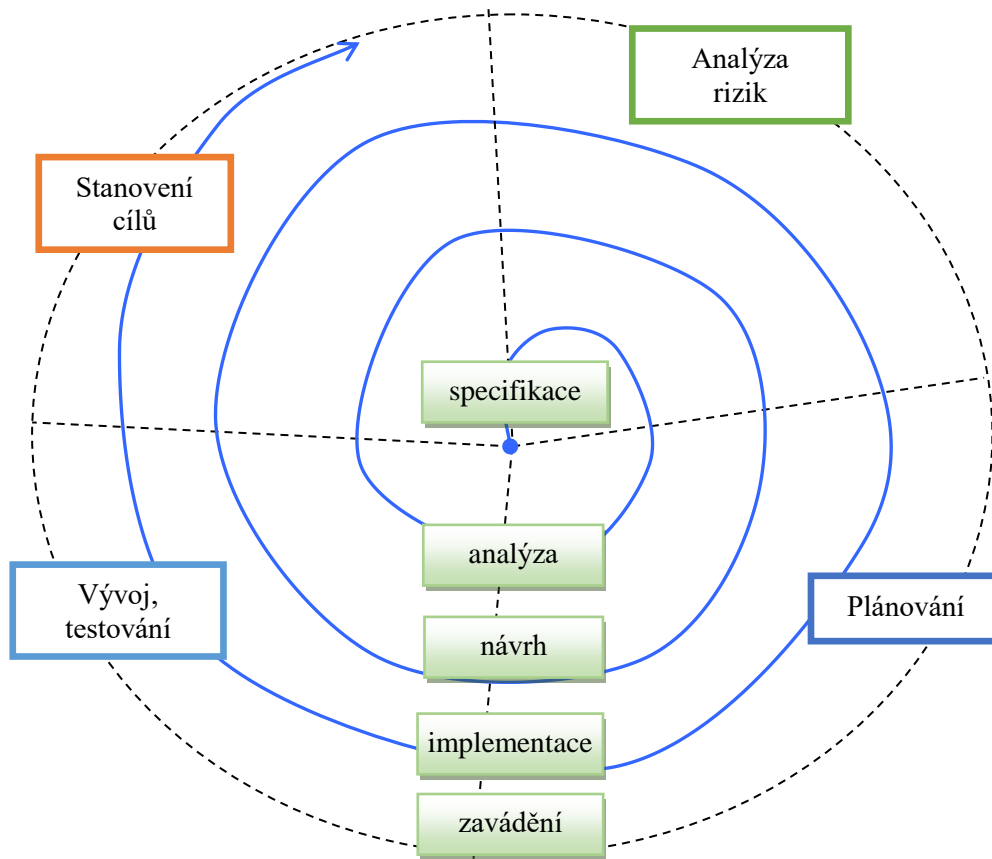
Model „výzkumník“:

- srovnajte s předchozím - co znamená rozdíl v praxi?

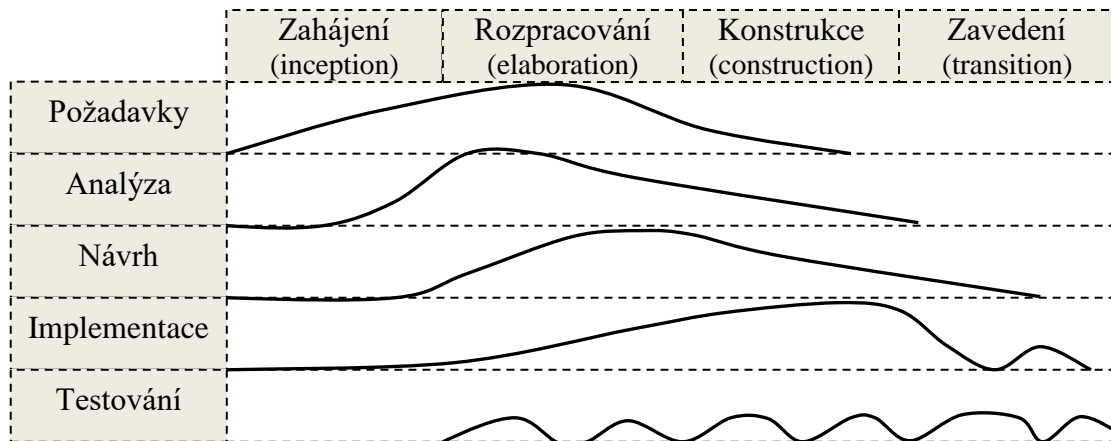


Model „spirála“:

- tzv. iterativní postup prací, s postupným prohlubováním řešení,
- model aplikovaný v současných moderních metodikách vývoje SW,
- doporučuji seznámit se s významem tohoto modelu a jeho kvadranty podrobněji (!),
- viz Kadlec V.: Agilní programování. Computer Press, 2004.



Model „UP – Unified Process“:



5. Obecné principy analýzy a návrhu IS

Základní prostředky pro boj se složitostí:

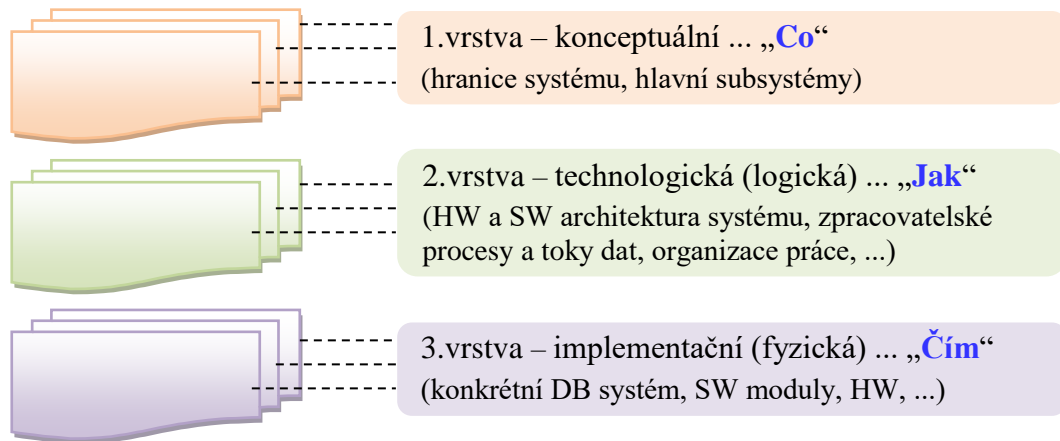
- hierarchický rozklad problematiky,
- etapizace a iterace postupu řešení,
- modelování a srovnávání modelů,
- použití grafických vyjadřovacích prostředků.

Základní principy analýzy a návrhu IS:

- princip abstrakce a konkretizace,
 - stupně abstrakce (kategorizace, agregace, generalizace),
- princip modelování IS (zobrazení IS),
 - model IS jako prostředek k dorozumění:
 - mezi odborníky vývojového týmu,
 - mezi vývojáři a zákazníkem,
 - model jako nástroj k analýze, optimalizaci a simulaci,
 - použití vhodných výrazových prostředků a CASE nástrojů,
- princip rozlišovacích úrovní v zobrazení IS,
 - od nejhrubší – znázornění interakce systému s okolím,
 - přes – systém jako množina spolupracujících subsystémů,

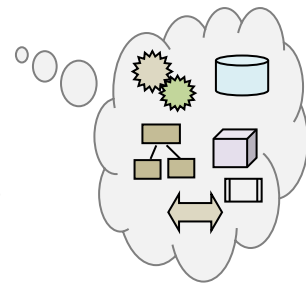
- až po detaily – potřebné k praktickému řešení,
- princip tří architektur v zobrazení IS.

Princip tří architektur zobrazení systému:



Analyzované oblasti (dimenze) vyvíjeného IS:

- funkční (procesní)
 - funkce (volby) v systému – „co systém umí“,
 - procesy probíhající v systému (jejich algoritmy),
- datová
 - struktura dat systému, návrh struktury databáze,
 - datové toky (přenosy dat) mezi procesy a subsystémy,
- řídicí
 - řízení systému podle dané provozní technologie,
 - stavy systému, příčiny přechodů mezi stavy,
- organizačně technologická
 - organizace práce pracovišť s využitím IS,
 - zachycení celého procesu zpracování (provozní technologie),
- systémově technická
 - HW a SW architektura systému.



6. Druhy metodik a trendy v jejich rozvoji

Zdroje informací o metodikách a metodách:

- literatura + Internet,
- školení u specializovaných firem,
- konzultanti specializovaných firem – firemní metodiky.

Druhy metodik:

- státem podporované metodiky a mezinárodní metodiky,
- agilní metodiky,
- firemní metodiky,
- další komerční a nekomerční (volně využitelné) metodiky.

Příklady metodik:

- historicky významné
 - MDIS (Multidimensional Development of Information System),
 - SSADM (The Structured Systems Analysis and Design Method),
 - SDM (System Development Methodology),
 - Euromethod,
 - SE (System Engineering),
 - SIV,
- v současnosti používané a známé
 - RUP, UP,
 - SCRUM.

Poznámka - vyhledejte na internetu:

- informace o současném stavu a využití výše uvedených metodik,
- informace o současných v praxi používaných metodikách.

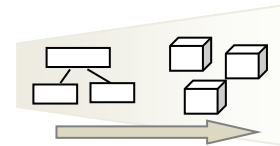
Trendy, jež se uplatnily v rozvoji a používání metodik:

- používání **iterativních** postupů
 - model „spirála“, prototypování,
- zaměření na **objektové** metody, techniky a nástroje

- obecně – objektový přístup ke tvorbě IS,
- modelování a optimalizace **procesů** jakožto součást vývoje IS
 - nejprve zmapujeme a optimalizujeme procesy,
 - pak na ně nasadíme inf. systémy,
- posun od "hard" k "soft" metodám
 - organizační, sociální, psychologická dimenze,
 - školení, konzultační služby, servis, ...,
 - chápání IS jako systému sociálně technického,
- zaměření na implementaci typového aplikačního SW (**TASW**),
- použití tzv. **agilních** metodik.

Druhy přístupů k analýze a návrhu IS:

- strukturovaný,
- objektově orientovaný (objektový),
- hybridní.



7. Agilní metodiky vývoje IS

Principy agilních metodik:

- přírůstkový (iterativní) vývoj IS s velmi **krátkými iteracemi**,
- intenzivní **zpětná vazba**,
- důraz na **komunikaci** mezi zákazníkem a vývojářem,
- přísné automatizované **testování SW**.

Hlavní myšlenky „Manifestu agilního vývoje SW“:

- nebránit se (mnohdy) změnám v průběhu vývoje IS,
- umožnit změnu SW je mnohem efektivnější, než snažit se jí zabránit,
- během vývoje SW je třeba se učit reagovat na nepředvídané události.

Příčiny vzniku agilních metodik:

- vznik většinou koncem 90. let 20. století,
- klasické přístupy jsou někdy administrativně náročné a nepružné,
- tj. neúměrně vzhledem k rozsahu a typu vyvíjeného IS,

- zadání mnohdy není zcela jasné, často se mění.

Nejznámější agilní metodiky:

- ASD (Adaptive Software Development),
- FDD (Feature-Driven Development),
- XP (Extreme Programming), tzv. „Extrémní programování“,
- Lean Development,
- SCRUM,
- Crystal metodiky.

Trade-off „něco za něco“:

- „administrativní lehkost“ agilních metodik,
- mnohdy vysoké riziko pro vývojovou firmu i zákazníka,
- => důraz na (až extrémní) způsoby vykrytí rizik.

Vhodnost agilní metodiky:

- menší týmy, menší projekty,
- před nasazením agilní metodiky je nutno zvážit, jaká je vyspělost (zkušenost) vývojového týmu (jak striktní vodítko tým potřebuje),
 - tj. zda je opravdu vhodné zvolit metodiku agilní,
 - či raději nasadit striktnější metodiku,
- pokud nasadit agilní metodiku, pak dobře zvážit kterou, např.
 - XP – volnější řízení, vhodné pro organizačně vyspělejší tým,
 - SCRUM – pevnější vedení týmu,
 - ...

Další informace viz:

- Kadlec V.: Agilní programování,
- Beck K.: Extrémní programování,
- Beck K.: Programování řízené testy,
- Myslín J.: SCRUM. Průvodce agilním vývojem softwaru,
- WWW informační zdroje.

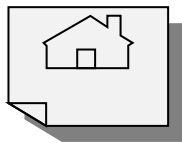
Příklady metodik zaměřených na vývoj WWW aplikací:

- metodika Jennifer Fleming,
- WebWAVE Development Process,
- WebWAVE Ongoing Development Process,
- Getting Real (f. 37signals), ...

8. Projektování IS a řízení projektů IS

Projektování IS:

- koncepční a realizační práce na vývoji IS,
- vedoucí vývoje IS (hlavní řešitel).



Řízení projektů (Project Management):

- organizační a ekonomická stránka vývoje IS,
- Project Manager,
- plánování prací, organizování týmů, vedení lidí, kontrolní činnosti, ...

Existence metodik (a CASE nástrojů):

- pro projektování IS (vývoj IS),
- pro řízení projektů (např. SAFE/PM).

9. Závěr k tématu

Seznámili jsme se:

- se [základními pojmy](#)
 - IS/IT, projektování IS, metodika, metoda, techniky, nástroje,
- s pojmem [životní cyklus IS](#)
 - a s významem členění životního cyklu IS na etapy,
- s obecnými [principy](#) analýzy a návrhu IS,
- se základními (historicky významnými) druhy [metodik](#),

- se vztahem mezi **projektováním** IS a **řízením projektu** vývoje IS.

Nejdůležitější pro praxi:

- **prostudování a pochopení** principů analýzy a návrhu IS,
- **přizpůsobení** metod konkrétním podmínkám použití
 - konkrétním charakteristikám navrhovaného IS,
 - potřebám a možnostem vývojového týmu,
- **aktivní aplikace** metodik, metod, technik a nástrojů,
- **!! nikoliv** „mechanické dodržování pravidel“ bez porozumění jejich účelu.

