

# JAK ČÍST KNIHY O METODIKÁCH ANEB ZOBECNĚNÁ RECENZE

**Pavel Drbal**

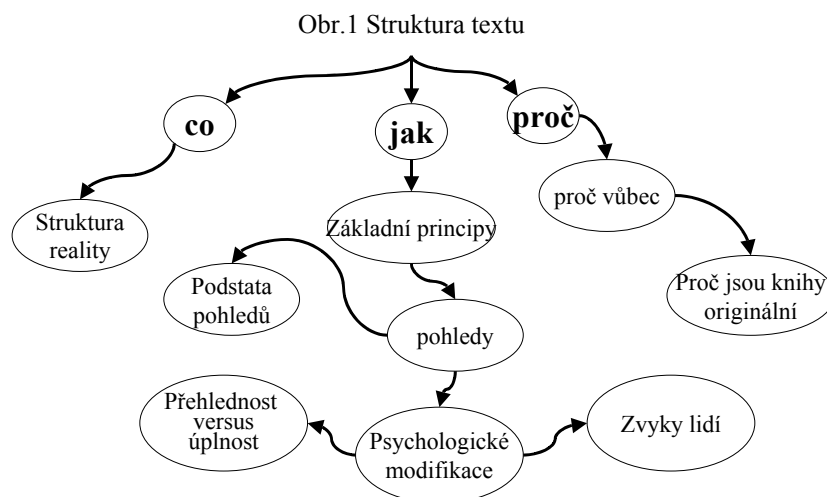
Vysoká škola ekonomická, nám. W. Churchilla 4, 130 67 Praha 3, ČR  
E-mail: drbal@vse.cz, <http://nb.vse.cz/~DRBAL>

## Abstrakt

Článek se zabývá přínosy popisů metodik a rozebírá jednotlivé přístupy.

## 1. Úvod

Každá kniha o metodikách sebe sama charakterizuje jako objevnou a hodnou následování. Jsem skeptik a jakékoliv propagaci ze zásady nevěřím, snažím se v každém textu najít racionální jádro. Na druhé straně mnoho čtenářů čte tyto knihy s očekáváními, která nemohou být splněna. V článku se rozebírá předmět těchto knih a určuje se, co může být přínosem. Struktura textu tohoto příspěvku je:



Nejdříve se budeme věnovat předmětu popisu, tj. uvedeme si schéma popisované reality, pak budeme diskutovat, jak je tato realita popisovaná v knihách o metodikách, jednak vypočteme základní principy, pak vysvětlíme jejich prezentaci v knihách. Nakonec se zmíníme o originalitě knih.

## 2. Struktura reality

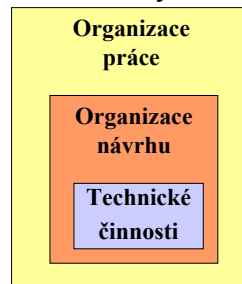
Realita sama o sobě je velmi složitá, chceme-li jakkoliv o ní mluvit, musíme sáhnout k abstrakci a typizaci. Na druhé straně, struktura určité reality vede k určité typizaci. V oblasti tvorby informačních systémů lze vytknout níže popisovanou strukturu:

Vymežíme si předmět zájmu – organizaci veškeré práce vedoucí ke vzniku informačního systému. Tyto práce jsou nejrůznějšího druhu – každý pes jiná ves. První krok je oddělit obecné práce, které jsou obvyklé u všech typů organizací (docházka, výplata, údržba místností apod.). Tyto typy činností označíme jako „organizace práce“. Zbude část, která je specifická pro vznik informačního systému.

I v této části lze vymezit dvě oblasti:

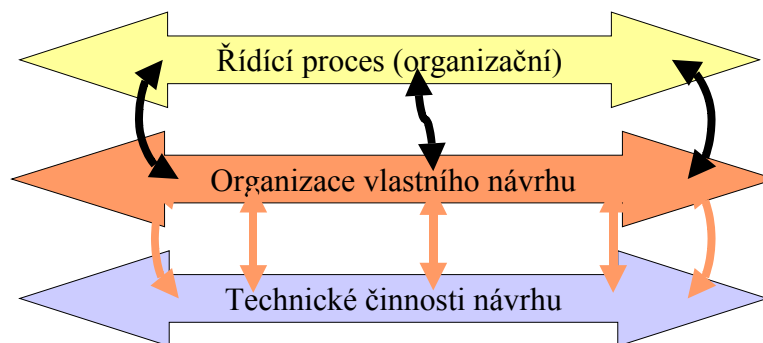
- Technické činnosti, které popostrkují tvorbu (výrobu) dopředu. Tato činnosti jsou obvykle spojeny s použitím konkrétních technik (například programování, modelování).
- Organizace návrhu, která určuje návaznost jednotlivých technik. Tato činnost je v celém procesu klíčová. (Rozdíl mezi zvládnutím a nezvládnutím organizace návrhu je stejný, jako mezi spisovatelem píšícím knihu a stádem opic mlátících do psacích strojů.)

Obr.2 Struktura reality



Z hlediska náplně se jedná o inkluzi, tj. „technické činnosti“ jsou součástí „organizace návrhu“, která je částí „organizace práce“. Z hlediska časového se jedná o tři souběžné procesy, které se navzájem ovlivňují.

Obr.3 Struktura procesů reality

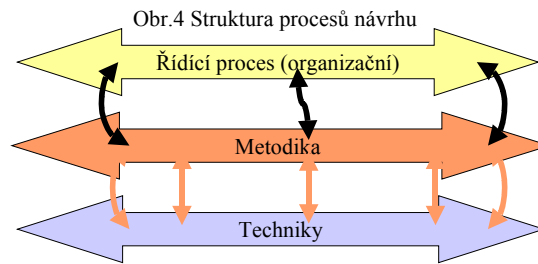


(Samozřejmě se jedná o tři skupiny souběžných procesů.)

Jedná se tedy o tyto procesy:

1. **Technické činnosti návrhu**, tj. vlastní výkonné práce. Proces tvorby informačního systému používá specifické techniky a prostředky (UML, programování, aplikační servery, modely aj.). pomocí těchto technik a prostředků se postupně vytváří konečný výrobek – informační systém.
2. **Organizace vývojového procesu**, tj. organizace technických činností. Řídí vlastní výkonné práce, zejména
  - určení pořadí prací,
  - kontrolu úplnosti a kvality,
  - určování iterací a znovupoužití.Tento proces je unikátní v tom smyslu, že pro různé procesy „Technické činnosti návrhu“ jsou různé procesy „Organizace vývojového procesu“ a vice versa, tj. různé procesy „Organizace vývojového procesu“ implikují různé procesy „Technické činnosti návrhu“ (odlišné techniky nebo jejich návaznosti).
3. **Řídící proces**, tj. obecná organizace výrobního procesu. Do tohoto procesu jsou vyděleny nespécifické činnosti, které jsou (v podstatě) stejné u různých typů výrob (povolování služebních cest nezávisí příliš na tom, jestli se vyrábí informační systémy nebo ledničky).

V informatické terminologii se obvykle používá pojem „metodika“ pro popis řízení vývojového procesu a pojem „techniky“ pro prostředky používané k vlastní práci.

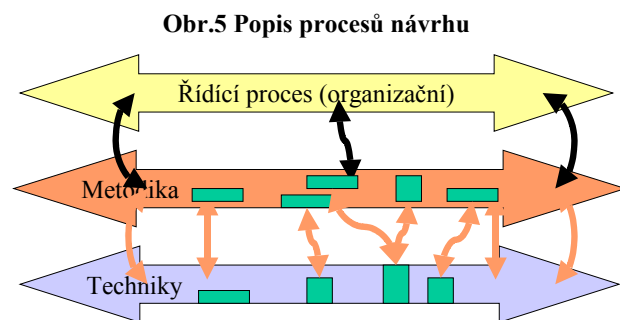


V tomto striktním přístupu se odděluje popis techniky od návaznosti technik (metodiky) do dvou různých procesů (podle mého mínění v soulase s praxí), kdežto v mnoha knihách se slovem „metodika“ označuje popis vybraných technik a jejich návaznosti.

Není možné, aby se rozumně čitelná kniha zabývala všemi zmíněnými procesy, dokonce ani metodiky a použitelné techniky nelze popsat, protože to je velmi široká oblast. Je samozřejmě možné napsat přehledovou knížku s výčtem všeho, tu však nepovažuji za čitelnou. (Pokud jsem – zřídka – některou četl, bylo to k ničemu, přečetl jsem spoustu textu a nic jsem se nedozvěděl, žádné informace jsem nezískal.)

I jen popsat používané techniky je úkol příliš velkého rozsahu, navíc nelze dost dobře popisovat techniky bez jejich návaznosti. (Výjimkou je „Myslíme v UML“ [9].)

Jen výjimečně se vyskytují knihy, které se věnují výhradně jednomu tématu, například [6]. Obvyklejší je popis mozaiky prvků z těchto dvou procesů. (Například [7] se věnuje převážně organizaci návrhu, z části technice bussines analýzy; [3] se zabývá technikami a jejich návaznostmi; [1,2] je převážně o metodice, zčásti obecné organizaci, jen okrajově se věnuje technikám.



## 2.1 Dílčí závěr 1

Knihy o metodikách popisují jen části procesů, které je zapotřebí realizovat. Při jejich aplikaci si musíme uvědomit své potřeby (v které části procesů se chceme poučit) a zjistit, které z těchto potřeb kniha uspokojuje.

## 3. Jak jsou metodiky popisovány

Pro popis metodik (organizace vlastní práce) se používají běžné prostředky – tj. rozklad procesu celkového návrhu na fáze, dále rozklad fází na podřízenější procesy – celkem o třech či čtyřech úrovních. Nejnížší úroveň procesů je přiřazení atomárních činností jednotlivým lidským rolím.

Daleko bohatší je aparát pro popis jednotlivých technik a metod.

### 3.1 Základní principy

Podstatou problému je složitost řešených úloh, všechny metodiky pro tvorbu informačních systémů lze považovat za metodiky pro řešení složitých úloh.

Základní principy pro řešení složitých úloh jsou pouze tři:

1. **Parcelace** - dělení jednoho složitého problému na několik jednodušších.
2. **Iterace** – opakováním postupu získáváme přesnější řešení – například princip hierarchického rozkladu je vlastně použití parcelace a iterace.
3. **Abstrakce** – zanedbání odlišností – nahrazení několika unikátních předmětů zobecněným principem a odvozování konkrétních předmětů z tohoto obecného principu. Důsledným použitím tohoto principu je objektivě orientovaný přístup.

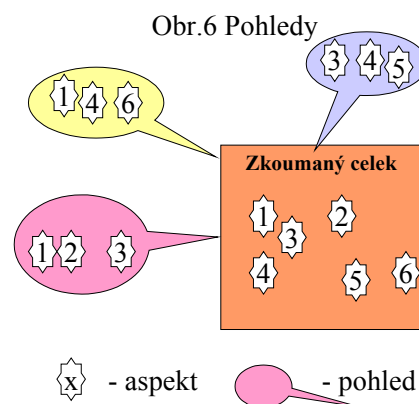
Všechny metody a techniky jsou aplikací těchto principů v určitém terminologickém hávu.

### 3.2 Pohledy

Pro zvládnutí velmi složitých problémů jsou nepostradatelné pohledy, které tvoří většinu technik návrhu informačních systémů.

Podstata pohledů je tato:

Mějme určitý složitý celek. V něm si vytipujeme několik aspektů, které ve svém souhrnu popisují na dané úrovni tento složitý celek. Abstrakcí vytvoříme několik pohledů tak, že některé aspekty zanedbáme, na jiné se soustředíme. K popisu celku je zapotřebí několik pohledů – aby pro každý aspekt existoval pohled, který se na něj soustředí. Je dobré, aby dva pohledy měly společný aspekt – lze pak dobře kontrolovat konzistenci pohledů.



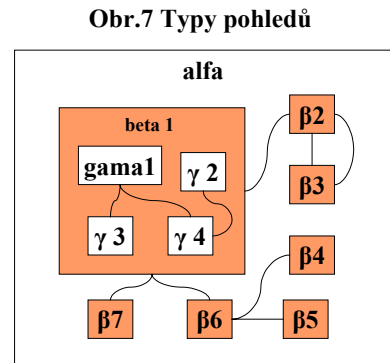
Zpracování jednoho pohledu (také se říká modelu) v nějakém jazyce (jako textový dokument, jako grafický výkres nebo kombinace obojího) se nazývá **artefakt**.

Praxe ukazuje, že ve vztahu k celku určité úrovně jsou čtyři základní typy pohledů:

1. **Statický** popis kontextu, ve kterém je zkoumaný celek zařazen.
2. **Dynamický** popis kontextu, do kterého je celek zařazen, tj. role zkoumaného celku v nadřazeném chování.
3. **Statický vnitřní** popis prvků, ze kterých se celek skládá.
4. **Dynamický** popis chování zkoumaného celku pomocí návaznosti chování vnitřních prvků.

K objasnění použijeme obrázek 7.

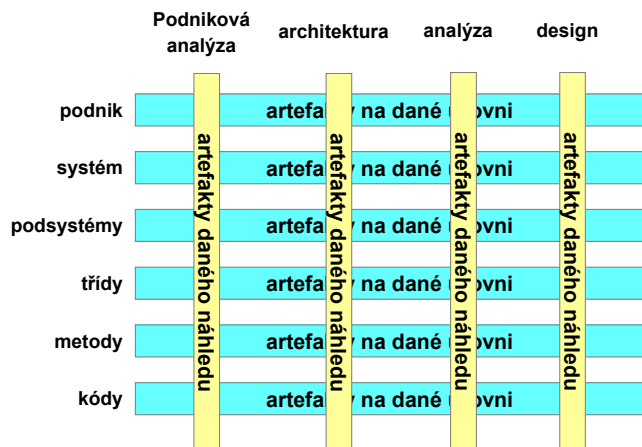
1. Statické vazby prvků úrovně beta v rámci nadřazeného celku alfa (vazby jsou znázorněny spojnicemi mezi čtverečky beta).
2. Dynamické vazby mezi prvky úrovně beta v rámci nadřazeného celku alfa (detailizace chování nadřazeného celku alfa. Pro popis se používají například DFD (diagramy datových toků), GA (grafy aktivit), OID (diagramy objektové interakce).
3. Výčet prvků úrovně gama jejichž souhrn vytváří prvek úrovně beta.
4. Popis chování prvků úrovně beta pomocí vazeb mezi prvky úrovně gama (příkladem je přechodový/stavový diagram).



### 3.3 Třírozměrný prostor projektování

Je zcela zřejmé, že používané prostředky se budou lišit podle toho, na jaké úrovni podrobnosti se pracuje. Také se budou lišit podle zpracovávané etapy. Z odlišnosti prostředků vyplývají i odlišnosti v metodice. Podrobně se tomuto tématu věnuje [4,5]. Zde jenom podotkneme, že znázornění na obrázku 8 je schématické, pojmenování sloupců jmény etap je nepřesné. Tyto styly rozboru můžeme například pojmenovat „náhledy“ – je to nutné, protože v jedné etapě bývá zapotřebí těchto náhledů více.

Obr.8 Třírozměrný prostor návrhu



Je zřejmé, že pro každý náhled na určité úrovni podrobnosti je možné použít artefakty výše uvedených čtyř typů. Tímto způsobem dostáváme třírozměrný prostor projektování. Každý bod tohoto prostoru představuje smysluplné použití artefaktů.

Na druhé straně je také zřejmé, že není možný postup, který by „na široké frontě“ realizoval artefakty ve všech bodech prostoru – ani to není fakticky možné (nebo vhodné, například realizovat kódy během podnikové analýzy je velmi zřídka). **Metodika je cesta v prostoru projektování.**

Také je zřejmé, že každý typ úlohy vyžaduje věnovat pozornost určitým náhledům na určitých úrovních. Zaplnění prostoru projektování určitými typy artefaktů jednak charakterizuje samotnou metodiku, ale také třídu úloh, pro které je metodika vhodná. Jestliže se podíváme na metodiky z hlediska zaplnění prostoru projektování, ukazuje se nám, že zaplnění prostoru se zvyšuje – rozsah záběru metodik se zvětšuje (ukázky viz [4]). Samotné metodiky se rozrůstají v monstra, která přestávají být rozumně zvládnutelná. Chceme-li řešit konkrétní úlohu, pak obvykle nevyžaduje tvorbu všech artefaktů, jinými slovy, při řešení konkrétní úlohy musíme použít podmnožinu metodiky.

### 3.4 Dílčí závěr 2

Metodiky pokrývají množinu možných úloh návrhem řešení určitého stylu. Skýtá to dvě potíže:

- **Metodika nabízí prostředky, které v dané situaci nepotřebujeme. A to musíme být schopni rozeznat.**
- **Metodika nenabízí prostředky, které potřebujeme. Příslušné postupy musíme být schopni doplnit.**

Podstatný je bod prostoru projektování, který je zapotřebí pokrýt. Odlišnosti zpracování tohoto bodu mohou být jen formálně odlišné (z důvodů chtěné originality) nebo se mohou lišit ve využívání psychických vlastností člověka (jejich odlišnost je dána psychikou člověka, není věcná) – o tom jsou následující kapitoly.

## 4. Unikátnost technik

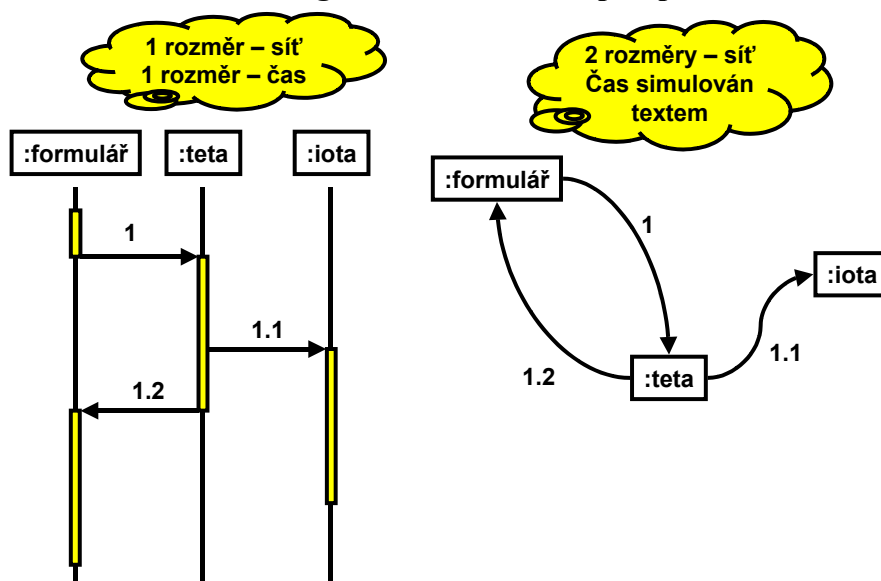
Máme čtyři typy artefaktů, ale z hlediska technik máme jen dva typy: **statický** a **dynamický**. Dvojice statických a dynamických pohledů se liší úrovní podrobností. Na druhé straně v grafickém jazyce UML (který je specializován na zachycení těchto pohledů) je osm či devět typů diagramů a mimo UML jich je jistě alespoň dvojnásobek. Je dobré znát důvody jejich odlišnosti, abychom se neučili něco, co již vlastně umíme.

### 4.1 Věcné důvody odlišnosti diagramů

Pohybujeme se v oblasti objektově orientovaného myšlení, je zřejmé, že pohledy se budou lišit úrovní abstrakce. Skutečně jsou dvojice grafů, které se liší v tomto aspektu – zobrazují buď abstrakce, nebo jejich instance. Pro statický pohled to je dvojice „Diagram tříd“ a „Diagram objektů“. **Diagram tříd** se (pomocí tříd) snaží popsat všechny možné objekty (všechny možné instance tříd), kdežto **diagram objektů** zachycuje jen jeden speciální případ. Diagram tříd popisuje vytvářený systém, kdežto diagram objektů je pomůcka k tomu, abychom dobře rozuměli diagramu tříd.

Podobné ale daleko vyhocenější to je u dynamických pohledů. (Vyhocenější proto, že dynamika se zachycuje podstatně hůře než statika.) **Diagram aktivit** zachycuje (může zachytit) všechny možné průběhy daného procesu (i všechny možné procesy), kdežto diagram objektové interakce (resp. diagram sekvenční a totéž platí pro diagram spolupráce) zachytí pouze část konkrétního průběhu jednoho procesu. Účel je zřejmý, abychom dobře zachytili (obecný) proces, musíme dobře porozumět jeho jednotlivým konkrétním průběhům.

Obr.9 Diagram interakční ↔ spolupráce

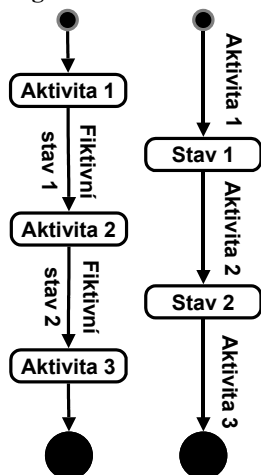


Dalším hezkým příkladem odlišnosti je dvojice interakční diagram a diagram spolupráce (viz obr.9). Oba diagramy zobrazují totéž, konkrétní průběh procesu (tj. instanci diagramu aktivit). **Interakční diagram** jeden rozměr zobrazení věnuje síti (celá síť je v něm zobrazena jako řada), druhý rozměr věnuje časové posloupnosti dějů v této síti. Oproti tomu **diagram spolupráce** zobrazuje síť ve dvou rozměrech (skutečně jako síť), posloupnost dějů zaznamenává speciálním číslováním přechodových šipek. Je vám asi zřejmé, proč se při projektování real-time úloh dává přednost interakčnímu diagramu.

#### 4.2 Psychologické důvody odlišnosti diagramů

Na pomezí věcných a psychologických důvodů je odlišnost v zachycovaném předmětu. Tak například **diagramy tříd** (objektů), **komponent** a **rozmístění** jsou téhož typu – jsou statické, zachycují uzly a spojnice mezi uzly vyjadřují jejich vztahy. Výrazové prostředky v diagramu tříd jsou velmi složité (jsou složité vzhledem k obvyklým diagramům), kdežto výrazové prostředky diagramů komponent a rozmístění jsou skoro triviální. Popisovat vztahy mezi třídami pomocí triviálních prostředků nejde a popisovat jednoduché vztahy pomocí hodně složitých prostředků je matoucí (snižuje míru porozumění čtenářem diagramu). Dále si myslím, že jediný důvod odlišnosti diagramů komponent a rozmístění je to, že se vyskytují v téže etapě a že je výhodné na prvý pohled poznat, co diagram zobrazuje (ostatně, v jednoduchých případech se pak dají výrazové prostředky umístit do jednoho diagramu).

Obr.10  
Diagram aktivit ↔ stavový



Typickým příkladem pro psychologickou odlišnost diagramu je **diagram jednání** (UseCase). Není věcný důvod, proč by model jednání nemohl být zobrazen diagramem tříd. Aktor je totéž co třída, typ jednání je popis chování, což je možno vyjádřit jak třídou, tak operací (metodou). Jediný nesoulad je ve spojnici aktora a typu jednání – je to trochu jiné než obvykle chápaná asociace – ale asociace je pojem velmi obecný. Je však silný psychologický důvod k formálnímu odlišení. Zacházení s modelem jednání je jiné než s modelem tříd a zaměnit to je velmi nebezpečné pro úspěch projektu. Jakákoliv zábrana této záměny je dobrá.

Rozdíl mezi **diagramem aktivit** a **stavovým diagramem** je po výtce formální – je to duální vztah. Co je v jednom diagramu znázorněno šipkou, je v druhém diagramu znázorněno oválem a opačně. Jediný pozorovatelný rozdíl jsem zjistil v zadání paralelních procesů – v diagramu aktivit je názornější.

Někteří autoři zavádějí graf pro znázornění procesů s bezrozměrnými šipkami, je to graf, který má aktivity i stavy jako uzly, aktivity oválné, stavy obdélníkovité, šipky znázorňují následnost a jsou bez rozměru. Nutno poznamenat, že takové grafy jsou pochopitelnější (čitelnější).

Z hlediska odlišnosti a shodnosti je velmi zajímavý vztah mezi diagramem aktivit a diagramy datových toků (DFD). Do objektově orientovaného projektování diagramy datových toků zavedl Rumbaugh [8], v prvních vydáních UML však nebyly (ani tam nebyly diagramy aktivit). Pak se zavedly diagramy aktivit a postupně se obohacovaly tak, že nyní pokrývají vše, k čemu sloužily diagramy datových toků. Důvody jsou zřejmě psychologické (kompetenční), objektová metodika nemůže mít stejné diagramy jako strukturovaná, byť mají dělat totéž.

## 5. Originalita

*Není nad originalitu,  
Každý po ní touží,  
Všichni chodí přes most,  
Jen já půjdu louží.*

Nových myšlenek je málo, původní práce sestávají z toho, že se vyloví idea cestující po člancích a seminářích, upraví se pro současné podmínky a zařadí se do systému procesů, který je potřebný. Kromě toho existuje i „originalita“ (chtěná i nechtěná), která čtenáře mate. Zdroje této originality jsou:

- tlak akademického prostředí,
- neznalost,
- časová nebo prostorová dislokace.

V určitých zaměstnáních existuje silný tlak na častou publikaci originálních výsledků, přičemž jejich hodnota je určována počtem kusů. Není nic snazšího, než vzít málo frekventovanou oblast, znovu ji přepsat se zavedením vlastní originální terminologie a mít vlastní grafickou notaci. (Takový je i tento článek, který se snaží připomenout již známé věci v souvislosti s tím, jak se učit metodikám.)

Často je potřeba si vymyslet potřebnou věc, o které se později ukáže, že byla již dříve publikována. Ovšem vlastnoručně vymyšlená věc mezitím byla publikována se svou vlastní terminologií, takže nakonec autor vypadá jako purista, který si zavádí svůj vlastní jazyk.



Občas není možné dostat se k relevantním informacím. To se například týká knihy [7], jejíž autoři si zavedli grafickou notaci dávno předtím, než vznikl standard UML. Použitím standardní notace by hodnota knihy zůstala, ale čtenáři by odpadlo mnoho práce s učením se nové notace (respektive u začátečníků – učení se notaci, která se nepoužívá nikde jinde než ve skupině autorů knihy).

## 6. Závěr

Tvorba informačních systémů je komplexní proces a dobře čitelná kniha nemůže být ničím jiným, než popis mozaiky částí tohoto procesu.

Chceme-li tyto knihy chápat a řídit se jejich doporučeními, nezbyvá, než si vytvořit procesní kostru, brutálně řečeno: převzít (a modifikovat) postup některé technologie a do tohoto postupu vkládat jednotlivé techniky tak, jak jsou v jednotlivých knihách popisovány.

V současné době považuji za dobrý popis kostry metodiky knihy [1,2], techniky lze získat v [6,3,7,8].

Při hledání technik si je třeba uvědomit jejich redundanci a zástupnost, diferencovat je podle (výše uvedeného) třírozměrného fázového prostoru [4,5]

### Literatura:

1. Ambler S. W.: Process Patterns; Cambridge University Press, Cambridge 1998, ISBN 0 521 64568 9
2. Ambler S. W.: More Process Patterns; Cambridge University Press, Cambridge 1999, ISBN 0 521 65262 6
3. Drbal, Pavel: OOMT - Objektově orientované metodiky a technologie 1. VŠE, Praha 1997. 193 stran. ISBN 80-7079-740-1.
4. Drbal P.: Metodika klasifikace metodik (The Method of Methods Classification), proceeding of the TVORBA SOFTWARE'2001, Ostrava, 2001, ISBN 80-85988-59-3
5. Drbal P.: Jak vytvořit a zkontrolovat vlastní metodiku (The Creating and Testing of The Special Method), minicourse, proceeding of the OBJEKTY'2001, Praha, 2001, ISBN 80-231-0829-X
6. Gamma E., Helm R., Johnson R., Vlissides J.: Design Paterns, Addison-Wesley, Singapore 2001, ISBN 020-163-3612
7. Polák J., Merunka V., Carda A.: Umění systémového návrhu, Grada, Praha 2003, ISBN 80-247-0424-2
8. Rumbaugh J.: Object-Oriented Modeling and Design, Prentice-Hall 1991
9. Schmuller J.: Myslíme v jazyku UML; Grada, Praha 2001, ISBN 80-247-0029-8.
10. Jacobson I., Booch G, Rumbaugh, J.: The Unified Software Development Process, Addison-Wesley, 1999. ISBN 0-201-57-169-2
11. <http://nb.vse.cz/~drbal>