

# CASE – obchodní trik, nebo naše spása

Karel Metzl

## 1 Úvod

Slovo CASE se v uplynulých dvou letech stalo nejpoužívanějším pojmem v analyticko-programátorských kruzích. Svědčí o tom i náš loňský seminář, kde se významná část příspěvků CASEM přímo nebo nepřímo zabývala. Přestože se v referátech velmi výstižně poukazovalo i na některá úskalí spojená se zaváděním CASE, valná většina z nich (s výjimkou referátu kolegy Běbra [1]) vyzněla značně optimisticky. Abych to vyvážil, budu se ve svém příspěvku zabývat hlavně problémy a nedostatky. Jsem si plně vědom toho, že takový jednostranný pohled může být zavádějící. Zároveň podotýkám, že nemám ani teoretické znalosti ani praktické zkušenosti s různými CASE na našem trhu a netroufám si proto obecně uvedené prostředky odsuzovat. Jedná se pouze o osobní postřehy a zkušenosti s omezenou platností.

## 2 Bude pro mne pořízení CASE přínosem?

Bobužel i pro prostředky CASE platí starý dobrý zákon o zachování práce. Abych mohl později t.j. za jeden, za dva roky nějakou práci ušetřit, musím teď (a právě teď se mi to vůbec nehodí) věnovat značné úsilí ke zvládnutí nového prostředku. A úsilí je to nemalé, zpravidla několikanásobně větší než je potřeba ke zvládnutí průměrného programovacího jazyka.

O užitečnosti CASE mě navíc přesvědčují hlavně distributoři t.j. lidi, kteří jsou na jeho rozšiřování hmotně zainteresováni a ke kterým, možná z určité osobní předpojatosti, nemám valnou důvěru. Jejich argumenty o počtu prodejů tu i hlavně cizozemských mě nechávají chladným. O tom vím své. Rovněž jejich ujištění, že až se to konečně naučím, přijde nová mnohem lepší verze [2], mě spíše popuzuje. Váhám. Neměl bych si na ni raději počkat?

## 3 Kdy je vhodný čas pro zavedení CASE?

V literatuře i od teoretiků se dovídám, že CASE mám poprvé nasadit na pokud možno jednoduchý projekt, který má mít navíc volné termíny. Jenže projekty, které mám zrovna teď realizovat nejsou jako z udělání jednoduché (kdyby byly, nepotřeboval bych asi CASE) a úlohy s volnými termíny se snad vyskytují jen v programátorských pohádkách.

Nutnost vyzkoušet si CASE nejprve na jednoduchém příkladu je pochopitelná, ale nese s sebou i jisté riziko. V praxi jsem se již mnohokrát přesvědčil, že některá řešení ať hardwarová nebo softwarová, která spolehlivě vyhovují pro malé objemy dat nebo rozsah složitosti se rázem stanou nepoužitelná, překročí-li určitou mez. Obávám se, že různé chvalozpěvy, které občas na prostředky CASE slyším, jsou právě od těch, kteří si je jen evičně vyzkoušeli na jednoduchých příkladech.

#### 4 Podpoří mě CASE po celou dobu vývoje?

Teoreticky jsou metodiky a jejich podpora CASE dobře rozpracovány tak, aby podporovaly vývoj od začátku až po realizaci, provoz a údržbu [3]. V praxi se však setkáváme spíše s různými byť užitečnými generátory kódu (CASE Tools, Lower CASE) nebo naopak s pomůckami vhodnými pro úvodní analýzu (Upper CASE).

Vybereme-li si nějaký prostředek z druhé skupiny, pak po úvodních fázích, ve kterých je nám CASE velice užitečný, dojde k ostrému zlomu. Analýza končí. Vytvořili jsme ve smyslu [3] modely funkční, datové, komunikační, řídicí někdy i organizační a prezentační, ale ten nejdůležitější, který by z nich vycházel a sjednocoval je: model konstrukční, nám jaksi chybí. Je třeba specifikovat, psát a ladit jednotlivé konkrétní moduly a programy a zde zůstáváme bez podpory, navíc v časové tísní často způsobené i tím, že jsme zbytečně ztratili neúměrně mnoho času hraním s novou hračkou.

#### 5 CASE a letitý spor analýza versus programování

V analytiko programátorské veřejnosti se již po léta vede boj mezi dvěma školami. Zastánci první tvrdí, že analytik by neměl být zatěžován nějakými zbytečnými znalostmi z programování a operačních systémů. Takové znalosti prý omezují jeho tvůrčí rozlet. Zarputilými zastánci této školy bývají hlavně Malí analytici viz klasifikace [6].

Druhá škola naopak učí, že už od samého začátku vývoje, by se mělo přihlížet ke konkrétnímu prostředí, ve kterém bude systém provozován. V drtivé většině případů je prostředí již předem dáno rozhodnutím zadavatele nebo různými často komerčními důvody.

Prostředky typu CASE dávají za pravdu škole první. Úvodní analýza by měla probíhat výlučně na logické úrovni a teprve po důkladném rozboru na této dosti abstraktní úrovni, by se mělo vybrat technické řešení (ze tří až šesti alternativ?) a do tohoto prostředí transformovat dosavadní logický vývoj. A právě tato transformace představuje zlom popsáný výše v bodu 4. Patřím spíše k zastáncům druhé školy, a proto z tohoto hlediska pohledím na prostředky Upper CASE trochu s nedůvěrou.

## 6 Je CASE všelékem pro každého?

Jakmile se objeví v analyticko-programátorské praxi nový prostředek, okamžitě všichni zajásají, že s ním i ti méně schopní (podle klasifikace [6] ženská od Cobolu, malí odborníci, Cvachové atd.) budou nyní rychle řešit i ty nejsložitější problémy. Takovou euforii jsem už zažil mnohokrát. Opakovala se s nástupem COBOLU, PL/1, dBASE, SQL atd. Bohužel ani v jednom případě se očekávání nespěnilo. Je až s podivem, jak jsme nepoučitelní a jak stále věříme, že se jednou objeví kámen mudrců.

Ačkoliv se to zdá podivné, platí neúprosné pravidlo: čím je ten který prostředek promyšlenější, úspornější v zápisu, automatizovanější, více neprocedurální atd., tím větší klade nároky na uživatele a jeho znalosti a tím větší škoda vznikne jeho neodborným používáním. Můžeme tedy zcela jistě očekávat, že s CASE prostředky dosáhnou dobrých výsledků jen dobří uživatelé, výborných jen výborní a neschopní s ním mohou napáchat nedozírné škody. Zatím vím pouze z doslechu, že nejlepší, ale také nejdražší CASE dokáže generovat kód v cílovém jazyku. Živě si dovedu představit monstra, která by při neodborné manipulaci takovou automatizací mohla vzniknout.

## 7 Nestane se CASE byrokratickou brzdou?

Prostředky CASE a s nimi spojené metodiky nejsou ani ve světě, který v současnosti pokládáme za náš vzor, přijímány všude bez výhrady. Martin [4] mluví o zdi mezi uživatelem a programátorem. Vyjmenovává její jednotlivé cihly a za velmi významný prvek pokládá formalizovaný vývoj a sáhodlouhé sepisování požadavků. Ještě ostřeji vystupuje proti CASE R. I. Polise [5]. Řadí ho mezi „pasti“, které číhají na nebohé východoevropské programátory a kterým by se měli vyhnout. Píše doslova: „Další skupina nepoužitelných řešení souvisí s aplikací strukturované analýzy a návrhu, třeba v současnosti s prostředky CASE. Metody tohoto typu spolehlivě zajistí, že náklady porostou: ani v nejmenším nezlepšují rychlost či efektivitu návrhu a implementace, zato však garantují širokou papírovou stopu, kterou může sledovat naprosto kdokoli s odvoláním, že dodržuje správný postup. Za případný neúspěch není nikdo odpovědný, protože všichni postupovali podle pravidel“.

Na druhé straně je určitá užitečná „byrokracie“ nezbytná. Nedovedu si bez ní představit tvorbu a provozování jakéhokoliv již trochu složitějšího systému. Stavět pouze na tvořivé práci několika „superprogramátorů“ [6], kteří udržují systém ve svých hlavách, smrtelných a v dnešní době často fluktuujících, s jejichž odchodem systém padá, je více než ošidné.

O užitečnosti formalizace svědčí i to, že veškeré státní zakázky ve Velké Británii mají přímo předepsanou metodiku SSADM a připravují se i závazné normy pro software informačních systémů pro EHS [7].

## 8 Mohu si CASE finančně dovolit?

Hlavní výhodou prostředků CASE vidím v tom, že všechny údaje o projektu jsou soustředěny na jednom místě (REPOSITORY) a každý pracovník projektu k nim má přístup, může je sdílet a podle svých přístupových práv i doplňovat. Podmínkou takového ideálního využívání je, že všichni vývojoví pracovníci mají na stole výkonný počítač (Windows) napojený na silný server se síťovou verzí CASE. I pro nevelký tým to pak představuje několikamilionovou investici, která navíc při dnešním překotném vývoji může morálně zastarat ještě dříve než se začne naplno využívat.

Pokud si takovou investici nemůžeme dovolit a zakoupíme jedinouživatelskou verzi na izolovaném PC, pak se v nejlepším případě bude CASE používat jako grafický editor a ještě na něj budou stát uživatelé frontu.

O tom, že ani na vyspělém západě nejsou prostředky CASE obecně dostupné jsem se mohl přesvědčit vloni. V rámci přípravy na automatizaci naší daňové soustavy nám expert z Velké Británie vysvětloval jak je u nich zpracovávána silniční daň. Přivezl si s sebou dokumentaci vypracovanou podle závazné normy SSADM. Celá dokumentace byla velice přehledná, expert byl v metodice náležitě sběhlý, ale k našemu velkému překvapení byla celá provedena v tužce a navíc od ruky. Předpokládám, že firma, která získala tak významnou státní zakázku, nebyla žádný outsajdr. Vzal jsem si z toho poučení, že metoda je důležitější než prostředek.

## 9 Závěr

V závěru by se slušelo odpovědět na otázku položenou názvem referátu. Rozhodně nelze pokládat CASE za spásu. Je to prostředek, který může být velice užitečný, ale také být nemusí. Velice záleží na vhodné volbě a na tom, kdo a jak ho bude používat. V každém případě už nyní bychom se měli na nástup CASE připravit a nejlepší přípravou, která je navíc investičně nenáročná, je studium příslušných metodik.

## Literatura

- [1] Běbr, R.: MODES ROBES nebo ZASE CASE  
Sborník PROGRAMOVÁNÍ '92
- [2] Peška, J.: TRH, OBCHOD, programátoři  
Sborník PROGRAMOVÁNÍ '92
- [3] Drbal, P.: CASE  
Sborník PROGRAMOVÁNÍ '92

- [4] Martin, J.: Application Development without Programmers  
Prentice – Hall, 1982
- [5] Polis, R.I.: Zavádění informačních technologií ve východní Evropě  
Computer Echo 1/93
- [6] Dvořák, L.: Superprogramátoři.... a ti druzí  
Sborník PROGRAMOVÁNÍ '92
- [7] Šlegr, J.: Programová tvorba z pohledu světových norem  
Sborník PROGRAMOVÁNÍ '92
- 

**Autor:** Ing. Karel Metzl  
AR SYSTÉM  
akciová společnost  
Gorkého 2  
702 00 Ostrava 1  
Tel: 069/23 11 23 kl. 175