

GENERÁLNÍ ZÁZRAK

Ing. Richard Bébr

Příspěvek se zabývá neutěšeným stavem využití výpočetní techniky v ČSSR a snaží se rozebrat některé hlavní příčiny.

"Co naplat, když je židle krásně tvarovaná,
ale nedá se na ni sednout".

1. Uvedení do problému

1.1 Výchozí stav:

Převážná většina agend, zpracovávaných u nás na počítačích vykazuje oproti ručnímu zpracování tyto drobné rozdíly:

- je dražší
- potřebuje více pracovníků
- prodlužuje termíny.

Na jednom ze seminářů "Programování" bylo řečeno:

Kdyby počítače zítra zmizely ze světa, v řadě států se zhroutí hospodářství; u nás si oddychneme.

Generální zázrak spočívá v tom, že za tohoto stavu věci nás programatory ještě nikdo nenechal zavřít.

Možná, že čtenáři připadají tato tvrzení příliš drsná a ponurá. Nicméně v lit. /1/ může nalézt uvedené myšlenky ve formě elegantnější a (zdánlivě) méně aurové. Citujme:

- Zvedly počítače produktivitu řídicí práce?
...Ještě nikdy předtím jsme neměli tolik administrativy.
- Pomohly počítače k vyšší kvalitě řízení?
...Celkem vzato je řízení nyní horší než dříve; počítače úpadek řízení nezastavily.
- Výdaje na výpočetní techniku patří k největším investicím (za které bychom na př. obnovili celou energetiku).
...Přítom stěží budeme hledat případy dovršené počítačové automatizace.
- Počítač nepřispěchal pracujícímu člověku na pomoc, spíše si z něj udělal živý výkonný nástroj.

Nabízí se otázka: co dělat, aby se tento neblahý stav zlepšil? Jednoznačné rychlé efektivní řešení ovšem neexistuje. Můžeme však alespoň vytypovat některé základní příčiny, jejichž odstranění by věci prospělo. Příspěvek se snaží o popis těchto příčin, shusta formou poněkud drastických příkladů; autor poznamenává, že příklady jsou vzaty z konkrétní praxe, byť i jsou místy neuvěřitelné.

1.2 Základní problém:

Ač se to většinou programátorů nebude líbit, platí zcela objektivně tvrzení:

Za úroveň každé agendy (projektu, systému) odpovídá vždy projektant (programátor) a neexistují žádné výmluvy, které by ho této společenské odpovědnosti zprostily. Musíme si uvědomit, že programátor (projektant) je tvůrcem, navrhovatelem a vůdčí silou pokroku v oblasti výpočetní techniky. Uživatel žádá obecně: racionalizovat řízení, aprávu, výrobu, ... (tak jako kdysi žádala společnost tu efektivní zdroj světla, tu pohyblivé obrázky v každé domácnosti). Co a jak udělat musí říci odborník, profesionál (tak jako kdysi odborníci dodali tu žárovku, tu televizi). Uživatel/spotřebitel za nikoho nic vymýšlet nebude.

2. Všeobecné příčiny

2.1 Úroveň řízení programátorů:

Jak jsem již v řadě příspěvků pro seminář "Programování" uvedl, jednou z hlavních příčin neúspěchu aplikací výpočetní techniky je t.zv. katastrofický vedoucí řešitelského kolektivu. Blíže nám to osvětlí konkrétní příklad:

Progresivní vedoucí útvaru ASŘ (vzdělání, praxe i kádrové předpoklady splněny) nabyt své jediné znalosti z oboru výpočetní techniky před cca 15 lety (kdy vedl něco úplně jiného) ve tří denním rychlokurzu "základy práce vedoucího" (kde počítačům bylo věnováno 60 minut výkladu). Nyní řídí projekt - dejme tomu - "Personalistika"; vyžádá si tudíž od vedoucího personálního útvaru "zadání" (písemné, abychom byli kryti). Nebohý personalista a) nemá čas, b) o počítačích toho moc neví, c) bojí se odmítnout, aby nebyl nařčen z brzdění

pokroku. I pošle projektantovi formuláře všech výkazů, které musí personální odbor podle celostátních předpisů zpracovávat (a doufá, že je za něj udělá počítač).

Progresivní vedoucí ASŘ uloží svým podřízeným, aby sadané sestavy naprogramovali a to v interaktivním režimu (toto slovo někde zaslechl). Staříčský analytik, který je náhodou přítomen poznamená, že dávkový interaktivní tisk sestav je nesmysl a mele něco o přítulném systému, prototypovém přístupu a podobných pojmech, vedoucímu neznámých. Je tedy označen za kverulanta a nadále bude vyloučen ze všech porad; tak jsou operativně odstraněny překážky a vytvořeny předpoklady k provedení opatření k zajištění plnění rozhodnutí.

Výsledkem je roztomilá agenda, kdy personalisté sice interaktivně aktualizují, ale veškeré výstupy se pořizují na tiskárně ve středisku (pokud možno vzdáleném několik km od sídla personálního útvaru); žádný výstup na obrazovku přece nebyl zadán! Agenda neobsahuje program pro výběry a operativní výpisy (nebylo požadováno!) a neexistuje ani program pro obyčejný opis kmene (chudák personalista neví, že něco takového existuje a tudíž to nezadal).

Personalistům přibyla práce (sestavy jsou dodávány 4 dny po předepsaných termínech, takže výkazy se stejně dělají ručně), vedoucí ASŘ splnil úkol a organizace vykáže nadřízenému orgánu další realizovaný subsystém ASŘ.

Z příkladu plyne, že katastrofický vedoucí všechno přehrává na někoho jiného; sám věci nerozumí a svým podřízeným nevěří (jsou to počouchlá individua, snažící se vedoucího zesměšnit). V souhrnu tedy "odborník" žádá od laika návod, co a jak má odborně udělat. Kdyby takto postupoval Edison, svítíme dodnes loučemi.

2.2 Úroveň projektů:

Je nesporné, že jeden a tentýž problém lze řešit mnoha způsoby. Zabývejme se případem, kdy určitý tým programátorů předložil určitému uživateli řešení určitého problému. Pominěme situaci, kdy všem zúčastněným šlo pouze o splnění cíle (an sich), tedy o čárku ve zlaté knize (bez ohledu na

výsledek, ekonomiku, lidi a pod. - což bývá častěji, než si myslíme).

Uživatel by měl nyní posoudit, zda řešení je dobré. Pokud nemá ve svých řadách špičkové odborníky, nebývá toho schopen! Dobré řešení je totiž charakterizováno tím, že lepší řešení už neexistuje. Jak to má ale ubohý uživatel poznat, když nemá žádné srovnání? Na všeobecný i odborný tisk není spolehnutí (často vydává příšerné podvody za špičková řešení), programátor-"profesionál" tvrdí, že to jinak nejde udělat (a bývá o tom přesvědčen, neboť nezřídka vůbec neví, že to jde lépe: nikdo ho to nenaučil, nic jiného neviděl, k zahraniční literatuře se nedostal, v cizině nikdy nebyl - tam jezdí jen katastrofický vedoucí), pokud někde existují opravdu dobrá řešení, neví se o nich.

V celé branži platí výrok lorda Russela:

"Je to bída. Pitomci bývají obvykle plni sebejistoty, zatímco chytrí lidé jsou plni pochyb a nejistot".

Nelze se ani spolehnout na posudek expertů, neboť

- neví se, kdo je vlastně expert (diplomy vysokých škol už dávno nejsou zárukou kvality držitele)
- nikdo nedbá na "odbornou čest svého jména" (a ti, kdo na ní naivně trvali, byli jakožto nepohodlní rýpalové z odborné fronty vymístěni).

Případný posudek experta má tedy tuto osnovu:

"projekt je v rámci daných podmínek a omezení s přihlédnutím k současnému stavu poznání řešen vcelku bez závažných nedostatků některé aspekty řešení jsou sice v gnoseologické konsekvenci příliš artibilní, avšak v souhrnném hledisku věcně skufanditní toliko na stránkách 23 a 32 se vyskytují drobné překlepy v obr. 3 by snad bylo lépe nahradit kolečka čtverečky v práci jsem postrádal přehled použité literatury ... atd.

Nakonec se ukáže prostý fakt: je v zájmu všech zúčastněných projekt zavést, ať už je jakýkoliv (nesplnění úkolu a termínu je předem vyloučeno, takže jaképak mudrování o kvalitě). Přitom obyčejné pracovníky, jejichž údělem bude

se systémem pracovat netřeba považovat za zúčastněné. Nikdo nikdy nepoukáže (resp. netroufne si poukázat) na to, že by to šlo lépe (a poukáže-li, je to věčný nespokojenec, rýpal a kverulant; s takovými si už víme rady).

Zavede se tedy nakonec jakýkoliv projekt; šéfů se to netýká a "naše děvčata už to nějak zvládnou".

2.3 Rozhodující detaily:

Často zapomínáme jednu pěknou zásadu: princip systému bývá jasný, o úspěšnosti řešení rozhoduje detailní zpracování bodů styku s uživatelem. V praxi slyšíme: "na to není čas, jsou to stejně prkotiny!". I malá chybička v detailním detailu přinese však velké ztráty času při provozu (nebo ztíží obsluhu a zvýší chybovost, ale to také stojí čas). A čas jsou peníze.

Aby to bylo zcela jasně, ukážeme si na příkladech z jiných oborů, jaké paradoxy mohou při špatně navržené (nepromyšlené) interakci systém-uživatel vzniknout:

- Knoflíky u plynového sporáku se dříve dělaly ve tvaru šipky, aby jejich poloha byla jasná na první pohled a ověřitelná i hmatem; dnes jsou kulaté s nepatrnými značkami na čelní ploše, takže musím jít do dřepu, abych zjistil jejich nastavení (u plynového spotřebiče s nebezpečím výbuchu zvláště pikantní).
- V Praze se používá unifikovaná jízdenka pro metro i tramvaje; v tramvaji se zasunuje do strojku ve směru šipky, v metru proti směru šipky; návod k použití na jízdence není.
- Poštovní směrovací čísla nemají žádnou vazbu na meziměstská telefonní směrová čísla; člověk musí zatěžovat paměť dalšími údaji a zvyšuje se chybovost (která je v tomto případě dost nákladná).
- "Autoboty" jsou vynikající obuv pro řidiče motorových vozidel; dnes jsou k dostání jen pro děti do 10 let, což je z uživatelského hlediska poněkud absurdní.

- Přístup k pražské lanovce na Petřín je označen "piktogramem", na kterém není (jak by každý předpokládal) vozíček s lanem, ale idealizovaný čelní pohled na vozíčko se dvěma světly; žádného zájemce nenapadne, že by mohlo jít o lanovku a musí se na cestu doptat.

Analogie v projektech počítačových agend naleznete snadno sami. Jako typický příklad uveďme slavný dotaz uživateli (v jednom interaktivním systému), který zněl

SUBMIT (ANO-NE)?

a jehož hlavním významem zřejmě bylo důkladně poděsit nešťastnou dívčinu u terminálu. Stačí občas zalistovat časopisem MAA, kde bývají ukázky řešení dialogových úloh; jsou sice míněny jako prezentace vynikajících výsledků, ale při podrobnějším studiu dotazových textů a obrazkových hlaviček si v duchu říkáte - chudák uživatel.

3. Projektanti a programátoři

3.1 Všeobecně:

Firma Microsoft začínala s 25 pracovníky; z toho bylo 5 (slovy pět) programátorů. Počty pracovníků dnešních špičkových softwarových pracovišť se pohybují mezi desítkami a stovkami. Přitom většina těchto "softwarových pracovníků" se stará o servis produktů, zaváděcí služby, poradenskou činnost, marketing - prostě o prodejnost, užitečnost a uživatelský komfort.

V republice dnes máme asi 20 000 (slovy dvacet tisíc) evidovaných programátorů a analytiků. O prodejnost a užitečnost software se nestará prakticky nikdo.

Ve světě se ustálila zásada: programátor je ten, kdo umí programovat (a rozumí se tím: dělat takové programy, které jsou uživateli k užítku a tudíž i - v obrovské konkurenci - aplikovatelné a prodejné). U nás jsou hlediska pro výběr programátorů diametrálně odlišná (a netroufám si je ani vypsat, tím méně komentovat).

3.2 Tragedie extrémů:

Než něco užitečného dělat, je lépe se prosazovat na teoretické frontě, nejraději rozvíjením problému, který vlastně problémem není. Neobyčejně nesmyslné, nicméně vděčné téma je výběr programovacího jazyka.

Výraznou, specificky českou ukázkou je t.zv. "spor o BASIC". Byly popsány kilogramy papíru úvahami "BASIC-ANO, BASIC-NE". Autoři plní tiskové řádky spíláním a hrubými útoky na tento nebahý jazyk (viz na př. LAA 11/88, str.405). Vážne-li na nějakém semináři diskuse, je možno stanovený čas vyplnit hádáním o BASICu. Autority se shodují v tom, že BASIC je jazyk plaký, demoralizující, destruuující osobnost, prostě nemravný; měl by být zakázán a jeho užití stíháno hrdelními tresty. Ale: na všech počítačích posledních let je BASIC implementován (proč asi?), akademie věd SSSR (která - dá se říci - leccěmu rozumí) zavedla BASIC jako unifikovaný jazyk pro publikace atd.

K tomu si dovoluji autorský (ryze soukromý) komentář: moderní programátor se neobejde bez znalosti několika programovacích jazyků; dnešní systémový software dává širokou nabídku nástrojů pro řešení úloh (včetně BASICu). Volba jazyka musí být řízena pouze a jedině cílem a podmínkami řešení i zpracování dané úlohy.

Totéž ale platí i pro ostatní pomůcky a prostředky (Jackson-nejackson, blokáč-strukturák, UNIX-DOS, personál-sálový stroj,): použijme vždy to, co je pro daný účel nejvhodnější! Umění nslézt v obrovské paletě prostředků ty, které jsou pro zamýšlený účel nejlepší je součástí kvalifikace dobrého projektanta. Snahy redukovat a priori množinu disponibilních prostředků jsou - podle mého názoru - známkou jisté impotence řešitele.

Pozn.: autor je si vědom, že za rouhavou obhajobu ďábelského jazyka BASIC bude krutě napadán a není si ani jist; zda jej v náporu fundamentalistů obhájí; c'est la vie.

3.2 Vášnivý programátor - zhouba lidstva:

V úvodu budiž poznamenáno, že nebude řeč o vášni erotické, ale pouze o vášni programovací. Pro objasnění problematiky nebude na škodu citovat některé myšlenky z článku /2/:

- ...Existuje jistá souvislost mezi rozšířením nových komunikačních prostředků a změnami v psychologické organizaci člověka. Vynálezy telefonu, telegrafu a rozhlasu na př. způsobily, že prakticky zmizely některé tradice života minulého století; zároveň se zmenšila intenzita a rozmanitost emocionálních prožitků, které doprovázely kontakty mezi lidmi.
- ...Televize výrazně oslabila rozvoj obrazové paměti u dětí. V rozvinutých zemích prakticky zmizel eidetismus (schopnost uchovávat v paměti živé obrazy předmětů ještě dlouho poté, kdy zmizely ze sorného pole).
- ...Současné výzkumy se soustřeďují na "vášnivého programátora". Výzkumné práce popisují vážné deformace citové oblasti a mezilidských vztahů u lidí, kteří využívají rozmanitou výpočetní techniku; je to důsledek kontaktu lidské psychiky a informačních technologií.
- ...Důsledkem rozvoje informatiky se může stát omezení sociálních kontaktů. Zvětšení psychologické vzdálenosti od jiných jedinců vede k agresivitě a k tomu, že dominují čistě pragmatické motivy styku.
- ...Vznikají umělé jazyky, v nichž není žádná mnohoznačnost a smyslové podtexty, charakteristické pro lidskou řeč. Sotva lze očekávat, že by se v takových podmínkách mohly rozvíjet tvůrčí mechanismy našeho myšlení. Vzniká otázka, zda skutečně nová řešení nebudou nahrazena pouhou kombinatorikou, jak je tomu už v současné urbanistice.

"Vášnivý programátor", tvor zběsile oddaný svému počítači se v poslední době prudce množí zásluhou různých osobních a domácích počítačích hraček. Je nekomunikativní, nezná humor, odmítá romantiku a poezii, nesnáší drobné nesmysly, které jsou kořením života. Povýšil svou exaktní logiku

(i sebe) nad běžné lidské stádo.

Pozor: "vášnivý programátor" je terminus technicus; nevztahuje se na ty, kteří mají rádi své povolání programátora a přitom zůstávají lidmi!

"Vášnivý programátor" je nebezpečný nejen lidské společnosti pro svou nekomunikativnost, ponurost ducha a odstup od života. Pokud je i zaměstnán jako programátor, stává se - ač je to s podivem - škůdcem této profese. Oplývá pecky a poky, hrabe se v paměti, je u vytržení nad různými flagy a znovu křísí bitové žonglérství. Program je pro něj cílem a smyslem života a zcela mu uniká základní pravda, že program je pouze prostředkem k uspokojení uživatelských potřeb. Ve všech oborech platí, že profesionál má znát perfektně řemeslo a "být nad věcí". Vášnivec je naopak otrokem věci. Se znalostí řemesla to u něj nebývá nijak slavné. Neboť - jak jsme si v Ostravě již mnohokrát řekli - součástí programátorského řemesla je na př. i dovednost napsat programy jednoduché, čtivé a srozumitelné. Dobrý profesionál (v každém oboru) si může nad perfektním řemeslem dovolit i nějaké ty parády. Vášnivý (š)moula řemeslem pohrdá a dělá jen ty parády; podle toho vypadá výsledek (laik žasne, odborník zděšeně prchá).

Nechceme přece nutit uživatele, aby přestali být lidmi a přizpůsobili se strohému sucharství "rychlých blbců" (jak nazval počítače Mr. Aitken). Program, který sám o sobě je nelidský, musí se ve svých projevech přizpůsobit člověku. A to lidské může do programu vložit jen programátor, který sám zůstal člověkem.

3.3 Profesionální čest:

Motto: Jednomu kolegovi z kapitálie bylo nabídnuto, aby si okopíroval jakýsi progránek, který na jejich trhu stojí mnoho set jednotek tamní měny. Podivil se: "ale to by přece byla krádež" a zdvořile odmítl s tím, že si raději ten drahý produkt koupí.

Představme si, že existuje u nás výborný programátor, znalý všech špičkových trendů, chytrý, systémově myslící atd. A teď je nucen katastrofickým šéfem, šíleným uživatelem nebo jinou "soubou" nadanou světskou mocí, aby projektoval něco, o čem je přesvědčen, že je to špatné. Jak se má náš programátor zachovat? Nebo jinak: koupíme si čs personální počítač, který je samozřejmě dodáván jen s nejzákladnějším programovým vybavením. Kolega "vášnivý programátor" sežene pokoutně plno foxbasů, editorů, spreadsheetů (to není překlep, to je špek pro angličtináře) a je vedoucím pochválen za to, kolik ušetřil podniku finančních prostředků.

Má nyní Pochtivý Programátor odmítnout pracovat s kradenými tovary a raději sám napsat (v assembleru a ve volném čase) svůj vlastní tabulkový procesor?

Nebo: Dělán oponentní posudek na zcela slaboduchý projekt. Mám ho zabít, když

- ho zpracovala parta mých kamarádů, kteří by přišli o premie?
- řešitelský tým vedl člověk, který má až příliš dlouhé prsty?
- vím, že náměstek už rozhodl, že se to bude realizovat?
- je mi jasné, že by mě už nikdy oponentem nejmenovali a já bych přišel o občasných 135,- Kčs?

Pokud by existovalo něco jako profesionální čest, nemusel bych v této kapitole napsat tolik otazníků.

Dovolují si připojit tvrzení, že rehabilitace pojmu jako "profesionální čest", "odborné jméno" a pod. by výrazně pomohla ke zlepšení úrovně našeho software (a nejen software). Možná, že by pomohlo i prosté opatření (dodnes běžné v dělnické profesi): za zmetek se nebere mzda.

Ale: máme vůbec na to? přejeme si to vlastně? obětujeme nějakému ideálu pohodlí zaběhané rutiny?

4. Softwarová krize

4.1 Ve světě:

Jak bylo řečeno v úvodu, je situace v našem software dosti neutěšená. Ostatní svět je sice zaplaven nejružnějšími programovými prostředky, nicméně všechno je uděláno

tak nějak "klasicky". Takové skoky, jakými byly kdysi "strojový kód - jazyk symbolických adres - COBOL" už jsme dlouho nezažili. Slibovanou umělou inteligencí se jaksi nedaří probudit; s hrůzou se zjistilo, že není jasné ani co to je "normální" inteligence, natož umělá. Redukoval se tedy rozsah pojmu na porozumění řeči a rozpoznání obrazů. Ale řekněte sami: když někdo porozumí tomu, co říká druhý a dívá se na autobus rozpozná, že je to autobus - je už kvůli tomu inteligentní? Zkuste říci větu "to je fór" jako oznamovací, tázací, s přízvukem na 1., 2. nebo 3. slově (nebo na 1. a 3. slově současně) - to je významů! Může si s tím někdy vůbec počítač poradit? Takže významný pokrok se nedá jen tak hned očekávat.

Shrnuto: softwarově vyspělý svět prožívá stagnaci v bohatství.

4.2 U nás:

Když se podíváme po našich softwarových luzích a hájích, můžeme shrnout předem: stagnace v bídě.

Soustavné a velkorysé kradení programů potlačilo vlastní tvorbu i rozvoj. Softwarový trh přešlapuje takřka na místě, zcela nám chybí instituce poradců. Programy jsou k uživateli bezohledné a uživatel tápe bez možnosti vybrat si z konkurenční nabídky.

Ale pozor:

Novela Hospodářského zákoníku, přijatá zákonem č. 98 /1988 Sb. s účinností od 1.7.1988 vymezuje zásady využívání programových produktů a nakládání s nimi. Akcentuje ekonomický význam nehmotných statků a jejich využití prostřednictvím směnných vztahů (lit. /3/). V novelizaci předpisů FCÚ je zakotvena možnost volné tvorby cen v oblasti programového vybavení. Zásady přestavby hospodářského mechanismu podporují podnikavost a rozvoj tržních vztahů.

Podnikání v oblasti software je velice efektivní, neboť je založeno pouze na duševním potenciálu "výrobce". Potřebný základní kapitál i provozní náklady jsou relativně nepatrné.

Jsou tedy vytvořeny všechny podmínky pro to, aby se produkce software rozvinula do tržního modelu jako jedna z prvních oblastí našeho hospodářství. Zdravé konkurenční vztahy by mohly přinést zásadní obrát v kvalitě našeho programového vybavení, což by prospělo především uživatelům. Je reálná naděje, že touto cestou by mohly být likvidovány neradostné skutečnosti, uvedené na začátku tohoto příspěvku.

5. Závěr

Situace je špatná, příčiny jsou známé, podmínky pro zlepšení jsou dány. Budeme je umět využít?

That's the question, přátelé programátoři !

6. Literatura

- /1/ Jirásek, J.: Přestavba a počítač. Moderní řízení 8, 1988.
- /2/ Značka -zr- : Počítač - člověk. Rudé právo 31.10.1988.
- /3/ Bányaiová, A.: Programy - součást majetku organizace a ochrana zpracovatele. MAA č. 12, 1988.
- /4/ Sborníky seminářů "Programování" z let 1976 až 1988.