POJMY A DOJMY

Ing Richard Bébr

Dojmy jsou (ve smyslu tohoto příspěvku) lasiné efektní chiméry, které někteří nekvalifikovaní a nekompetentní lidé vyvozují z obecně platných pojmů. Prosazování realizace takových dojmů může mít přinejmenším zlé důsledky ekonomické, provozní i společenské. Prostě je nebezpečné plést si pojmy s dojmy.

1. Úvod

Pro ilustraci toho, o čem má pojednat tento příspěvek si uvedeme příklad z doby nedávné. Osoby a obsazení:

V - vedoucí: vysokou školu absolvoval přes výkonnostní volejbal, odpovědného místa dosáhl pomocí včas získané červené knížky (pozn.: před nedávnem ji zahodil a nahradil trikolorou a libivým odznáčkem).

P - výkonná, solidní, pečlivá a spolehlivá pracovnice, která na rozdíl od vedoucího umí samostatně myslet (a je tudíž celkově mírně nežádoucí).

Úryvek dialogu z výrobní porady:

V: Je třeba okamžitě vymístit a odprodat náš zastaralý počítač EC 1057 a nahradit jej sítí PC !

P: Máme na ty PC dolary?

V: Koupíme moderní čs stroje PP 06 za koruny!

P: Bez winchesterů a bez tiskáren?

V: S tím už si musí poradit naši programátoři. Společnost jim umožnila vzdělání, tak ať teď ukáží, co se naučili!

P: A co bude s hotovými agendami, kterých jedeme poměrně dost?

V: Jste detailistka a kverulantka, narušujete svými výpady smělé perspektivy naší práce! Stávající agendy prostě přepíší pro PC pracovníci provozu v rámci závazku. Anebo se to udělá nějak jinak.

P: Myslím,pane inženýre, že si pletete pojmy s dojmy.

V: Úkoly jsou vám všem jasné, končím poradu.

Všichni víme, že takřka všude existují borci, kteří pochytili z nevin nebo jiných masmedií pár chimérických nápadů a ke své

větší slávě se je snaží bez ohledu na důsledky prosadit v praxi. A nedělejme si iluze, že naše soft-revoluce nějak zabrání další katastrofické činnosti těchto dovedů a rejdíků. Aby bylo jasno: nechci prosazovat přízemní realitu proti technickému pokroku. Podporuji však používání vzdělaného a zkušeného rozumu proti hurá-akcím, které by nám v neúprosných podmínkách tržního hospodářství tahaly z kapsy naše výplaty.

Proto uvedu namátkou několik příkladů dosti rozšířených dojmů a pokusím se je konfrontovat s obecně platnými pojmy.

2. Všechno známe a umíme

2.1 Analýza systémů:

Naprogramovali jsme u nás už spoustu různých systémů a určitá část z nich dokonce "chodí". Není tedy problémem programovat i systémy další, "na současné úrovni". Analýza systému je přece jednoduchá: uživatel řekne, co potřebuje, my to nějak rozhodíme do souborů a už se může programovat.

Když pak hotový výtvor prohlédne znalec, zjistí zajímavá a často žertovná fakta:
V evidenci pracovníků je i položka "barva očí", která se ovšem na žádné sestavě nevyskytuje. Dále jsou ve větě "osobní zálíby"; pečlivý analytik navrhl i výstup "přehled pracovníků dle zálib". Jelikož však není zajištěna aktualizace (žádný pracovník nám nepřijde sám od sebe chlásit, že nechal rybaření a věnuje se spíše slečnám), nemá výstup žádnou informační hodnotu.

To byly sice poněkud extrémní příklady, ale obecně je uchovávání dat bez informačního významu značně rozšířeno.

Máme evidovat zakázky a pracovníky, kteří na nich pracují; uživatel žádá i evidenci měsíčních náběhů hodin na jednotlivé zakázky. Ve větě zakázky tedy deklarujeme 12 položek pro měsíční náběhy, ve větě pracovníka odkazy na zakázky. Řada potenciálních informací (kolik udělal určitý pracovník na dané zakázce a pod.) je "mrtvá" - nelze ji z takto navržených souborů získat. Je ovšem tragické, že to celé se odehrává v relační databance, kde stačilo zřídit relaci "zakázka-pracovník-měsíc-hodiny" a ve "větě" zakázky ani měsíční náběhy odpustit. Z těchž dat by bylo možno získat všechny potřebné (dokonce všechny možné) informace.

Máme tedy v systémech data, která prakticky nenesou informaci i data, která díky nevhodnému strukturování dávají mnohem méně informací, než by mohla.

2.2 Systémový přístup:

Naštěstím teorie systémů je její zdánlivě jasný a jednoduchý základ: systém jsou prvky, spojené vazbami. Všichni vědí z tisku, že nebýt systémového přístupu nestanuli lidé na Měsíci apod., takže i politické strany systémově spracovávají své programy a systémově přistupují ke společenskému dění (výsledky bývají někdy tristní, jak jsme nedávno viděli).

Každý jen trochu ambicionální "odborník" uvede k jakémukoliv problému několik systémových aspektů a když je dovedně propojí s vazbami na téma "všechno se vyvíjí-všechno souvisí se vším-ve všem je rozpor" (dnes ovšem "demokratizace integruje realitu-trh stimuluje produkci-privatizace je báší kvality") máme odborně fundovaný spis, projev či "materiál", začasté vhodný i k získání několika písmen za jméno. Školy pak vykládají systémový přístup jako sled mnoha kroků, které musí student znát naspamět a u zkoušky je bezchybně v daném pořadí odříkat.

Už léta se např. mluví o "systému člověk-stroj". Konkrétně a věcně se toto téma probíralo pouze na seminářích "Programování" v příspěvcích i v diskusích. Ještě dnes nacházíme v odborné literatuře nezávazné obecné úvahy, ačkoliv již v r. 1983 jsme se v Ostravě celkem jednoznačně domluvili a dokonce stanovili soubor pravidel pro vedení dialogů (výsledky jsem zachytil v lit. /P84/ a Ing. Tvrdík se k problému vrátil v lit./1/); moderní software pro PC naše tehdejší úvahy do písmene potvrdil.

Zatím jsme také mnoho neslyšeli o tom, že dobrý analytik řeší tři okruhy vztahů v systému člověk-stroj, stroj-stroj, člověk-člověk. Jedním z uznávaných dojmů je názor, že systémový analytik je prostý technik a do lidí mu tedy nic není. Je však ověřeným pojmem, že systém bez dobře vyřešených vztahů mezi lidmi (jako jeho prvky) je systém nedořešený a v praxi mívá havarijní sklony. Systémový analytik si má přibírat k ruce odborníky na organizaci, psychology, lékaře, filozofy a sociology (i když poslední dvě profese jsme soustředěným úsilím prakticky zlikvi-

dovali) a s jejich pomocí řešit různé vztahy v systému.

2.3 Móda a legrace:

Utěšeně se šíří různé viry a jen co budeme mít nějaké ty sítě, nastoupí i hackeři (kteří se probourávají ochrannými systému a ničí data a programy; v lepším případě pouze zveřejní, jak zase převezli pány programátory nebo vám nechají někde na disku grafickou visitku).

Neodborný a často i "odborný" tisk píše o těchto jevech s obdivem. Četl jsem, jak geniální studentík celé noci oka nezamhouří, mobilizuje svůj neobyčejný důvtip a pomocí počítače typu Didaktik Gama vnikne do bankovního systému, kdež si nechá na své konto připsat symbolických 1000 $, neboť je tak ušlechtilý, že pohrdá miliónem.

Neustále se povídá a píše, jak je těžké a pracné vyrobit účinný virus a že jeho autor musí být vytečným programátorem. Uznávám, že je to pravda, ale odmítám se autorovi virů obdivovat. Řekněme si na rovinu, že tvorba virů je jen určitou formou vandalismu: jaký je rozdíl mezi zničením pracně vytvořených programů nebo informací a zničením sochy v parku? Sebegeniálnější hacker je obyčejný lump a terorista: jaký je rozdíl mezi tím, když se vbourám do systému (a něco tam ukradnu nebo poničím) a tím, když se vám vbourám do soukromého bytu? V čem se liší anonymní otravování počítačových systémů od anonymního otravování lidí telefonem?

Promiňte mé rozhořčení, pro sborník P'90 až nenáležité. Ale slyším z různých stran konejšivé řeči o inteligentním žertování, o milé klukovské recesi a dokonce i o tom, že "ten strach z virů je teď v módě, ale nesmí se to přehánět". Víme všichni, kde je kořen tohoto zlehčování a bagatelizování situace: kdybychom si skutečné nebezpečí virů a hackerů připustili, museli bychom veškerý software snad dokonce kupovat (!) a museli bychom určitý (nemalý) peníz investovat do ochrany systémů. A tak je lépe zůstat u optimistických dojmů a nepřipouštět si k tělu reálné drsné pojmy.

2.4 Umělá inteligence:

Po tristní úvaze o vandalech a teroristech něco veselého: problém umělé inteligence. Tady je pro dojmy a fikce široké a často neorané pole působnosti. Dokonce i technické zprávy (zejména výzkumné) s oblibou hovoří o řešení MTZ pomocí umělé inteligence, o programech pro evidenci základních prostředků s využitím heuristiky apod.

Napsal jsem o tom pár řádků v lit. /P89/. Podívejme se na problém blíže:

Nejprve se věřilo panu Turingovi, že inteligentní stroj je takový, se kterým si můžeme popovídat. Na truc (a schválně) napsal pan Weizenbaum z MIT program ELIZA (viz např. lit./2/), který zcela splňuje Turingův požadavek inteligence a přitom je neuvěřitelně pitomý.

Vytvořily se tedy obory jako "rozpoznávání obrazů", "porozumění řeči" apod., přičemž vrcholem snažení je simulovat určitou schopnost člověka, ke které podle běžných pojmů žádná inteligence není potřeba. I blbec v cukrárně rozpozná dort, avšak inteligentní program s tím bude mít moc práce, zvláště bude-li dort šišatě ukrojen. O spotřebě času na rozpoznání dortu u člověka a u počítače raději nehovořím. Prostě se zjistilo, že současné programovací a technické prostředky nějaké inteligentní uvažování nezvládnou. Je to asi dobře; o tom, co by se stalo, kdyby nám počítače začaly opravdu myslet existuje bohatá literatura v oblasti sci-fi.

Na druhé straně se však musíme rázně postavit proti dojmu, že práce na umělé inteligenci byly, jsou a budou zbytečné a nesmyslné. Mnohé výsledky dnes běžně využíváme (např. v oblasti grafiky, optimalizace, některých simulačních úloh atd).

I když dojem, že počítače nabudou značné inteligence je nereálný (dnes!!), musíme pojem "umělá inteligence" a jeho aplikace zevrubně studovat a v praxi využívat.

Je ovšem pozoruhodné - a odporuje to asi přírodním zákonům - že ač skutečná inteligence v umělé podobě neexistuje, podařilo se nám už dávno realizovat negaci: všichni víme, kolik se v našich programech vyskytuje umělé blbosti!

3. PC řeší vše

3.1 Automatizace - nic jednoduššího:

S příchodem PC vypukl nezřízený jásot v řadách počítačových hochštaplerů. Počítadla typu mainframe a mini už totiž přestala být exotickou hračkou a hrozilo nebezpečí, že někdo bude chtít, aby se na nich dělala pořádná práce. Oblíbeným trikem salonních systémových analytiků je tedy okamžitý výprodej stávající "velké" techniky (pokud možno do šrotu) s tím, že se časem pořídí 80 až 140 personálů. Když pak přízemní skonom neuvolní potřebné množství dolarů, zakoupí se dva výkonné stroje PP 01. Programátorům, kteří se k nim neprozřetelně přiblíží je uloženo urychleně na tomto vrcholu techniky zavést zpracování mezd. To se zpravidla nepodaří, mzdy se zadají do PVT a bystře se vyrobí program pro evidenci zlepšovacích návrhů, který je předváděn postupně všem vedoucím, publikován v tisku a popisován na seminářích. Týž program zůstane po dlouhou dobu jediným jakž takž chodícím agendou na naší progresivní technice.

Nechci naříkat nad likvidací větších počítačů (statistiky prodeje ukazují, že zájem o tuto zastaralou techniku v civilizovaném světě neklesá), ale spíše nad tím, s jakými dojmy se leckde přistupuje k nasazení PC. Sotva jsme se naučili integrovat úlohy a využívat databanky, již trháme systém a jeho data na malé kousíčky. Že existují sítě a distribuovaná data? Někde možná ano, ale když zjistíme, že u nás po velmi slušném spojovém okruhu přenášíme mizerný 1 MB prakticky 1 hodinu, přejde nás optimistický úsměv (pro silné povahy jsem spočítal, že po normální telefonní lince přenášíme týž 1 MB více než 11 hodin).

Stejně tak opomíjíme zabezpečení systému. Na streamer nezbyly peníze; když si ověříme střelivou pracnost pořízení kopie winchesteru na diskety, raději zavřeme oči a využívše uvolněných vztahů v oblasti náboženství dáme na modlení a věříme, že to s pomocí Boží nějak dopadne. A to už nemluvím o tom, že se nám po kancelářích povalují diskety (často ve funkci podšálků) s důvěrnými údaji, které si může kdokoliv kdykoliv opsat a zkopírovat pro svou potřebu.

3.2 Jazyky n-té generace:

Výrobci dBASů a spreadshitů (pravopis viz lit./P89/) horlivě ujišťují, že s jejich produktem si kdokoliv snadno naprogramuje cokoliv. Jistě: znám laiky-neprogramátory, kteří si pomocí Foxbase udělali dokonce telefonní seznam svých sleček!

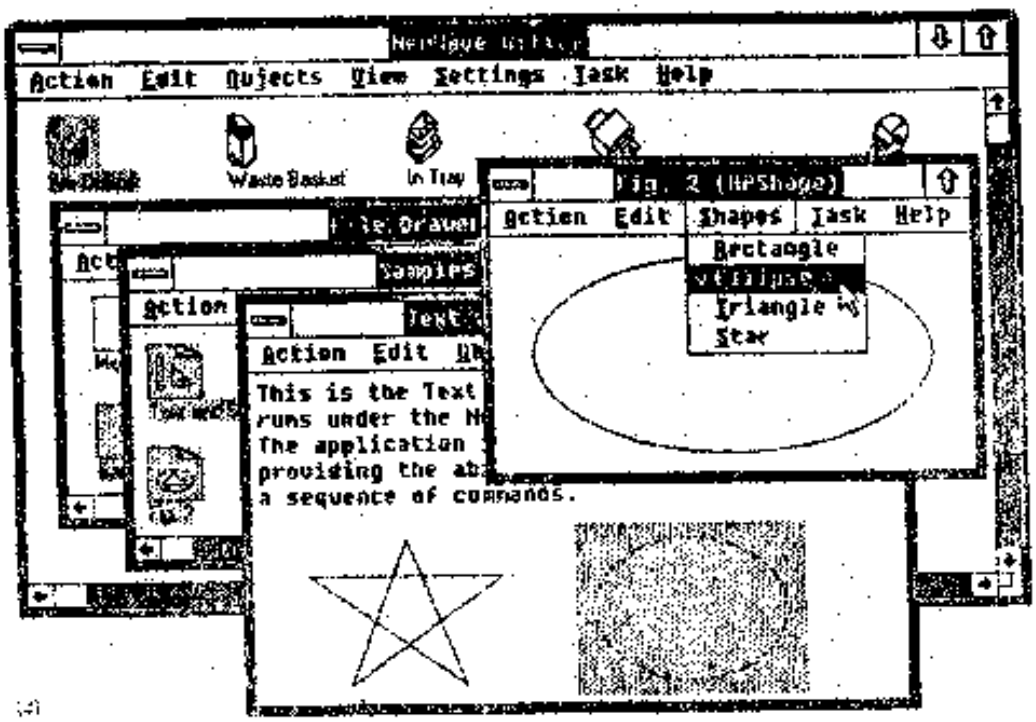
Protože však se chytré systémy rozrostly do značných rozměrů, běžný laik shlédnuv několik obézních manuálů (kterými jest se mu prokousati, aby zvládl základy systému) zbaběle prchá. Profesionál tyto systémy uvítá, rychle se je naučí a již mastí jeden program za druhým (většinou jako melouch), aniž by se zdržoval nějakými hlubšími analýzami. Aby to zas nevypadalo tak přízemně, prohlásí, že pracuje s jazykem čtvrté (ve společnosti obdivně vrkajících dam pak páté) generace, čímž zajisté povznese svou odbornou pověst.

Děje se ale něco horšího: naše software-housy vedeny dojmem rychlých zisků bez investic počaly chrlit účelové programy dle objednávek (psané samozřejmě s použitím "jazyků n-té generace") a tak trochu zapomněly na vlastní výzkum a vývoj.

Je tedy naprostá většina našich uživatelských programů (pro PC) vytvořena pomocí produktů zámořského původu. Charakter a styl těchto prostředků se zřetelně promítá i do výsledných programů. Naskýtá se otázka: je to tak dobře?

Není závažné, když program občas na uživatele bafne nějaké to "ADD NEW RECORDS (Y/N)" nebo "CANNOT OPEN FILE". Zajímavější je otázka pojetí zmíněného již vztahu "člověk-stroj". Např. systém "oken" podle mého názoru vyhovuje americké mentalitě, založené na jakémsi dryáčnictví a smyslu pro efektní show, kteréžto vlastnosti nejsou sice našemu uživateli přílišblízké. Spousta oken s různými menu a ještě k tomu s "ikonami" Američana nadchne, kdežto náš uživatel se nad takovou obrazovkou spíše zasmušší (viz ukázka na následující stránce). Nebylo by dobře trochu se zamyslet nad mentalitou našeho uživatele a vytvořit takové produkty, které by jí vyhovovaly?

Obrázek k výkladu na minulé stránce:



Jako příklad z této sféry uvedu ještě geniální hru TETRIS: Američanům připadala příliš strohá a suchá i doplnili ji pestrými obrázky v křiklavém provedení, objevujícími se překvapivě bez jakéhokoliv zjevného smyslu; průzkumem jsem zjistil že naši hráči dávají přednost původní jednoduché a elegantní sovětské verzi. Poláci - jejichž mentalita je nám bližší než americká - převedli TETRIS do prostoru (!) bez laciné vnější okázalosti se sympatickou profesionální střídmostí; hráč není nijak rušen a může se soustředit na vskutku ďábelský obsah hry.

Suchý, střízlivý a profesionálně realistický mladý ekonom (např. má vlastní dcera) okamžitě rozmetá tyto romantické úvahy věcnou poznámkou, že na otevřeném softwarovém trhu nejsme schopni konkurovat americké razanci, podepřené špičkovými znalostmi a rutinovanou praxí; slovanská jemnost a něha bude převáloována nájezdem drsných cowboyů. V současném neruby obráceném světě přísluší optimizmus a idealizmus zřejmě pouze starcům; dovolte tedy autorovi vyslovit hříšnou myšlenku: až se přeženou různé bouře, náš Programátor (tak jako již mnohokrát v historii) ohromí svět a přinutí softwarové giganty obdivně zmlknout

(nedávno to např. učinilo několik čs frajerů v oblasti CAD a ochrany programů) - konečně viz též lit./P88/.

4. Programování je pouze logika a rutina

4.1 Ani logika:

V lit. /P88/ jsem se ozval proti dojmu, že "programátor potřebuje ke své práci exaktní logiku". K hlubšímu rozboru nyní použijeme ukázku logiky z lit. /4/:

"Vyslovíme větu: Tato věta je nepravdivá.
Jsou nyní dvě možnosti:
a) předpokládáme-li, že věta je pravdivá, vyvodíme, že je nepravdivá, neboť to sama o sobě tvrdí; tento předpoklad je tedy vyvrácen
b) je-li věta nepravdivá, vyplývá z ní, že je pravdivá a to je předpoklad "a", který byl vyvrácen dříve.
Jde o paradox, vzniklý tím, že věta mluví o sobě samé."

Každý praktik uzná, že uvedená úvaha je sice chytrá, elegantní a zajímavá, v programátorské praxi však stěží nalezne uplatnění.

Řada matematiků slušně programuje; je to však hlavně tím, že využívají své matematické pečlivosti. Dobře programuje spousta lidí humanitního zaměření, u kterých by to málokdo čekal. Dobře programují i zarytí nepřátelé matematiky a logiky (např. moje již zmíněná dcera). Na druhé straně známe matematiky, logiky a exaktní pumprdenty, kteří nedají dohromady tři řádky chodivého programu.

Dojem o exaktní logice nabraďme platnými pojmy: Programování je zcela specifickou kombinací fantazie a pečlivosti, tvůrčího rozletu a přízemního prtání, neustálého dodržování a současného porušování řádu věcí. Programátor musí být v jedné osobě bohém a pedantský úředník. Kdo neumí soustavně střídat obrazotvornost, hodnou Isaaca Asimova s počítáním písmenek, řádek a sloupců není dobrým programátorem.

Bébrův zákon (lit. /P86/) to říká jasně: všechny přístupy, které užívá muž když si podmaňuje ženu a všechny přístupy, které užívá žena pro svedení muže jsou vždy v programování přínosem.

Tyto přístupy jsou - jak známo - založeny na imaginaci celkové koncepce a současně na pečlivém provedení detailů.

Tím považuji za vyvrácené veškeré dojmy o exaktní logice v programování. Citlivým znalcům pak doporučuji nevrubné studium jemných rozdílů v programech, psaných programátorskými samci a samičkami.

4.2 Ani rutina:

Mnoho lidí podléhá dojmu, že rutina = profesionalita. Rutinovaný číšník vám prostě přinese to, co jste chtěli, nezarmoutí vás, ale zůstává "mimo hru". Profesionál přidá v přesně odhadnuté chvíli tu úsměv, tu slovo, tu milé gesto; váš pobyt v hospodě povznese na velmi příjemný zážitek. Stejně tak profesionální programátor nepadá do stereotypní rutiny; každý program řeší tak, aby vyhověl konkrétnímu uživateli jeho neblahou činnost.

Rutinér napíše program, jaký se po něm žádá. Profesionál dbá na své odborné jméno a snaží se dodržet morálku profese; odmítá psát programy, falšující skutečnost, deformující informace, degradující data; odmítá vyrábět děsivé kreatury jenproto, aby se někde mohl vytahovat splněným nesmyslným plánem. Profesionál se snaží, aby se na něj vztahoval ten dávno zapomenutý pojem "na slovo vzatý odborník". Trvá vždy na tom, že jeho podpis je zárukou kvality toho, pod co se podepsal.

4.3 A co psycha?

Ač je to s podivem, bývá programátor často i člověk. Je tedy třeba studovat též jeho duševno (donedávna panoval obecně rozšířený dojem, že takovéto úvahy jsou k ničemu; tento dojem byl silně propagován, neboť co kdyby někoho napadlo zabývat se psychou některých vědních oborností a "specialistů").

Často propadáme dojmu, že nezbytnými vlastnostmi programátora jsou chytrost a píle. Již Bismarck zavedl pro vojáky klasifikaci:

líný + chytrý = velitel
pilný + chytrý = štábní důstojník
líný + blbý = řadový voják
pilný + blbý = zastřelit za úsvitu

Uvedené členění můžeme snadno aplikovat na programátory. Nesmíme tedy roli chytrosti a píle absolutizovat.

Opomíjeným pojmem bývají i základní povahové rysy a mentalita programátora. Ukažme si na jednoduchém příkladu chování lidských typů. Příklad zapíšeme v prokletém jazyku BASIC (na který kdekdo nadává, ale všichni ho umíme):

V závislosti na hodnotě proměnné I má být vykonán určitý příkaz. Programátor-flegmatik nehne brvou a píše

```
1Ø ON I GOTO 2Ø,3Ø,4Ø atd
2Ø bflm : GOTO 1ØØ
3Ø hohkr: GOTO 1ØØ
atd.
```

Sanguinik s lehkým srdcem napíše poněkud frivolně

```
1Ø IF I=1 THEN bflm
2Ø IF I=2 THEN hohkr
3Ø IF I=3 THEN mlfb        atd.
```

Naproti tomu cholerik se na problém oboří

```
1Ø IF I>1 THEN IF I>2 THEN IF I>3 THEN   atd.
```

a po chvíli psaní narazí na vlnobití KLSů, ve kterých se zcela ztratí; i zařve silným hlasem, roztrhá vše co dosud napsal a případně prokopne dveře u skříně.

Programátor-melancholik dosud nenapsal nic, neboť smutně dumá o tíži úkolu, který na něj dolehl.

Při ladění programů se povahy projeví ještě výrazněji: Flegmatik pečlivě fajfkuje větve, které už prozkoumal a pomocí dvanácibarevných fixů značí do listingu opravy a úpravy.

Sanguinik se nesmírně baví tím, že tiskárnou proletěla jako blesk celá krabice papíru 1+2, aniž by se cokoliv vytisklo (omylem se mu do jakéhosi nulovacího cyklu dostal příkaz PAGE).

Melancholik sedí s hlavou v dlaních nad poněkud zmatenou sestavou a ujišťuje sám sebe, že tohle nemůže nikdy odladit.

Cholerik při výskytu první chyby zastaví stroj, vymaže, vymuluje a prokleje všechno, roztrhá listing, vytiskne nový, zjistí, že zapomněl o jakou chybu vlastně šlo a vyřítí se na chodbu, aby se uklidnil cigaretou. V půli cigarety na chybu

přijde (v duchu), zahodí hořící pálku do krabice s makulaturou, spustí editor, opraví chybu, zapomene opravený source uložit, kompiluje a linkuje, při spuštění vyjede táž chyba a celé divadlo se několikrát opakuje; přitom divý muž stihne ještě uhasit vzniklý požár a založit nový.

Jak vidno, je užitečné zabývat pojmy z oblasti psychy programátorů, neboť tak lépe pochopíme jejich projevy a můžeme jim např. přidělovat práci podle individuálního naturelu.

5. Závěr

Upustím od pedantického shrnování závěrů, neboť mi na minulém semináři bylo v kuloárech vytýkáno, že mé příspěvky pro sborník "Programování" nekončívají happy-endem. Chci žadatelům vyhovět a protože jsem si vědom, že každé vyprávění má končit svatbou, sděluji všem čtenářům, že se teta Kateřina znovu bohatě provdala (lit. /3/).

6. Literatura

Odkazy na mé vlastní příspěvky pro seminář "Programování" jsou uvedeny v textu jako /Pnn/, kde nn je ročník semináře.

Dále bylo použito této literatury:

/1/ Tvrdík,J.: Interaktivní programy a pravidla dialogu. Sborník "Programování '88".

/2/ Martin: Design of Man-Computer Dialogues. Prentice Hall,1973.

/3/ Jirotka,Z.: Saturnin.

/4/ Zich,O.: Moderní logika. Orbis,1958.