

Vilém Holán p.m.
KRS Ostrava

Ing. Petr Miklica
VOKD Ostrava

PROGRAMÁTOR Z POKLEDU UŽIVATELE

Jako pracovníci výpočetních středisek jsme si navykli posuzovat vztah uživatel - výpočetní středisko výhradně ze svého postavení. Vycházeli jsme s faktu, že uživatel je nepříjemný druh člověka, který se musí řádně vycvičit k používání výpočetní techniky, neboť její naprostou nutnost ještě zcela nepochopil a pokud ji pochopil, tak zase neustále obtěžuje požadováním úprav v ukončených programech. Je věčně nespokojen s termíny zpracování. Sám dává pozdě podklady a při tom je netrpělivý při očekávání výsledků. S vychytralým úsměškem komentuje drobná nedopatření ve výstupech a při tom všem má ještě programátory za lidi, kteří mu vlastně zkomplikovali život.

Všechno má ale i druhou stránku. Uživatel má své problémy. Někdy ho dokonce samotného napadne, že by mu výborně pomohl počítač. Třeba nestandardním výstupem z nějaké datové základny. Svůj problém si v mezích možností ujasní a vydá se do výpočetního střediska. Tam však - zejména u velkých středisek narazí na neprůhledné a nezdolatelé monstrum. Je v plánovací oddělení poučen, že jeho požadavek bude řešen komplexně v rámci toho či onoho podsystému ASŘ v příští pětiletce. V nejllepším případě bude jeho požadavek zařazen do plánu jako samostatná úloha na příští kvartál. Zakázkové myšlení velkých středisek mu vezme dech. Pustí se do práce s kapesní kalkulačkou, papírem a tužkou a za týden je hotov. Bohužel je

také nadlouho hotov s výpočetní technikou.

Jaké postavení má v tomto problému programátor ? Různé. V mnoha střediscích je uživateli docela blízko, v jiných má k němu dál, v jiných je od uživatele oddělen na jedné straně bariérou analytiků, na druhé provozem a implementátory.

Postoj programátora bude ovšem záležet také na pochopení jeho vlastní role v celém procesu zavádění automatizace informatiky. Může ji chápat jako úzce vymezenou odbornost v soukolí budování ASŘ, ale může své postavení chápat také velmi pružně - bude se prostě považovat za odborníka výpočetní techniky, kterému současná úroveň poznání resp. vědy vymezuje právě tento způsob práce. Odborníka, který poučen vývojem řekněme v posledních patnácti letech ví, že nitro může být v naší profesi způsob práce jiný, dělba práce jiná, stejně jako bude i naše technika jiná. Co zůstane stejné, co je invariantní vůči těmto změnám - je cíl naší práce. Totiž poskytnout uživateli co nejrychleji, nejpřehledněji takové informace nebo takové jejich zpracování, které právě konkrétní uživatel v právě dané chvíli potřebuje. Míra, v níž to dokážeme zůstane vždycky skutečnou mírou naší profesionální zdatnosti.

A právě tento cíl jako by se paradoxně ztrácel se uživatele. Samozřejmě, že zde můžeme uplatnit námitku v tom smyslu, že automatizace se provádí plánovitě podle projektu ASŘ. Jsou stanoveny priority, schváleny harmonogramy. Uživatel může své požadavky uplatňovat během projekce. Může se k projektu vyjádřit při oponentuře a dokonce ještě při zkušebním běhu. I když to všechno je jisté pravda, není to pravda celá. V zásadě se jistě takto postupovat musí, ale navíc musíme zcela vážně respektovat i život. Chcete-li věcněji - respektovat stálý vývoj uživatelských systémů.

Projekční práce včetně schvalovacích procedur mohou trvat měsíce i léta. Zvolený postup je nepochybně dostačující na příklad ve stavební projekci. Tam ale projektujeme něco, co potom slouží padesát, sto i dvě stě let. Náš informační systém bude sloužit - i když dobře - tak deset let. Za tuto dobu se změní počítače, způsob pořizení dat, přenosný sys-

tém a především dojde k pokroku v uživatelské oblasti. Znání se organizace, předpisy, řídicí algoritmy, technologie. Už během projekce se náš přednátk výrazně mění. Takže i když bona fides považují všichni oponenti a recenzenti projekt za dokonalý, zpravidla už neodpovídá realitě, nad níž je vystaven. Při provozu se neobejdeš bez podstatných zásahů už vůbec.

Uživatelé se vyskytnou řada problémů - až už v rámci vyprojektovaných a provozovaných systémů či mimo ně, které by potřeboval řešit ihned. Víme, že jsou to často opravdu racionální požadavky. Můžeme si však bohužel přislanat, že si s touto situací prostě nevíme rady. V každém případě se to tak uživatelé, který má dobrý vztah k výpočetní technice, musí jevit.

Uživatel, jak jsme sami zjistili, formuluje pak takto své závěry:

1. Počítači lze svěřit jen droby, a nichž můžeme riskovat termínové skluzy. Příčinou skluzu bývá nejčastěji nespolehlivost technického zařízení a komplikace při převodu dat.
2. Počítači lze svěřit jen dílky, a nichž očekáváme okamžitě výsledky. Výpočetní střediska se při dávkovém zpracování chovají jako dodavatelé služeb. Řadí požadavky do fronty a postupně plní.
3. Výpočetní střediska plní velmi těžkopádně požadavky na změny v provozovaných systémech, vyplývající ze změny předčtené reality.
4. Výpočetní střediska usilují o instalaci stále větších počítačů. To vyvolává potřebu větších prostor, větších investičních prostředků, více lidí. Úvedené potíže se však tím nezmenšují, nýbrž zvětšují.
5. Výpočetní střediska produkují stále velké množství v projektu předem připravených standardních sestav a brání se požadavkům na nestandardní sestavy.

Uživatel přislanvá řadu kládů a výhod, které získal do - pčnými systémy, ale opředeně vytyká nedosažení těch kvalit, které výpočetní technika v sobě v každém případě skrývá:

1. Souhrnná rychlost celého procesu zpracování informací.
2. Přístupnost informací.
3. Pohotovost ke změně.
4. Jednoduchost ovládání uživatelem.

Shrneme-li tedy to, co až dosud bylo řečeno, žádá se po nás spolehlivost, rychlost, správnost, přizpůsobivost. Každý z nás musí uznat, že jsou to požadavky oprávněné. Ale považují první tři z nich jsou na nás ve větší míře nezávislé, zůstává jen u posledního - u přizpůsobivosti. Sem patří schopnost vyhovět nepředvídanému požadavku a spolehlivá údržba, to znamená průběžné přizpůsobování se měnící se realitě.

Programátor se nyní může zeptat, proč se má jen on zamýšlet nad těmito úskalími výpočetní techniky. Neříkáme ovšem že jenom on, ale hlavně on. Programátor je pro uživatele už tradičně ústřední postavou výpočetního střediska, ne-li dokonce jedinou postavou, o níž vlastně něco bližšího ví. Má to své pochopitelné kořeny. V počátcích výpočetní techniky se uživatel poměrně snadno dohodl s programátorem o svém problému. Programátor každý úkol nejen pomohl upřesnit, projektoval, programoval, ale také prováděl pravidelný provoz a mnohdy dlouho na provoz dozíral. Údržba byla potom pro něho snadnou záležitostí. Programátor je také pro uživatele představitel preciznosti profese. Uživatel ví, že programátor se nezájímá ani v interpunkčním znaménku.

Je jsou to ovšem jen uvedené důvody historické, ale fakt, že průběžná reakce na požadavky uživatele a požadavky údržby je především v rukou programátora. Uživatel se domnívá, že tkví v ochotě a v produktivitě.

Pakud se týká ochoty, považoval by uživatel za ideální, kdyby výpočetní středisko a jeho pracovníci, především programátoři byli asimilováni jeho prostředím. Uvítá spíše strojařského, stavežského, nemocničního programátora. Pracovníka, který kromě své "řemeslné zdatnosti" (v tom nejkrásnějším smyslu) bude "cítit" a uživatelskou problematikou, vezme ji za svou a bude se zde považovat za specialistu. Takoví lidé (a dodáváme, že existují) nemají pak potřebu utíkat

od systému, zůstávají u jeho údržby a rozvoje a nikterak to nepovažují za ztrátu odborné prestiže. Toto pojetí koinciduje i s technickým vývojem (minipočítače, terminály) a bude tedy na něm určitě něco pravdy.

My sami budeme považovat za stěžejní problém přetížení. Poněvadž úkoly porostou určitě rychleji než počet programátorů, je jediná cesta ve zvýšení produktivity programování. Na každém setkání programátorů, v odborných časopisech, v osobních setkáních, v každém středisku se tímto problémem zabývájí. Ale sami považujeme, že změna může přinést jen podstatný zisk, takový, který by přecnost běžných programů snížil třeba na třetinu. Myslíme, že k tomu jsou cesty. Na příklad generátory běžných operací na souborech s parametrickým ovládním. Přitom máme se však u našich softwarářů, aby budovali tyto prostředky i za cenu funkčních omezení a takovým vnějším ovládním, které je blízko malé násobilce. Nechtě se na příklad připomeň jen jedno číselné srovnání, jen vzestupné třídění atd. Devadesát procent díků lze většinou vždy této jednoduchosti přizpůsobit.

Jestliže tyto prostředky dáme do rukou analytiků, resp. projektantů, nebo sami použijeme pro řešení náročných úkolů uživatele, vytvoříme si potřebný časový prostor. Bude to možná na úkor elegance řešení a na úkor strojového času. Ale tyto kvality můžeme spíše obětovat než dosažení cíle - vyhovění uživateli. S tím souvisí do jisté míry i délka práce. Od programátora analytika šel vývoj k diferenciaci analytické a programátorské práce, poněvadž se vydělili ještě softwaráři (o ostatních funkcích zde nemluvíme). Domníváme se, že v blízké budoucnosti se těžištěm programování stane software. Bude zapotřebí " vyrábět " prostředky, o nichž jsme mluvili výše, bude zapotřebí " vyrábět " prostředky pro ovládním datových základů, pro komunikaci atd. Tyto práce pohltí asi všechny schopné programátory. Tito pracovníci budou sestavovat také programy, které nebude možno zvládnout standartizovanými prostředky. Standardní úlohy zvládne nabídnutými prostředky analytik či projektant jen jako dodatek, či formu zá-

pisu své práce.

Zbývá ještě dodat, že sami bychom k otázce ochoty a produktivity přidali ještě otázku orientace. Je obtížné, aby se člověk orientoval v krátkém čase v tolika úkolech. Z části by byl problém orientace řešen návrhem uživatele na asimilaci výpočetních technik jeho prostředím. Z velké části však problém orientace souvisí s dokumentací. Tak jako jsme dříve řešili na nedostatek dokumentace, začneme v blízké době asi kritizovat její přebujelost. Něco nám může odbourat již zmíněná standardizace nejméně opakovaných souborových operací. Dále si však musíme uvědomit, že náloky jsou k něčemu procedurální popisy, zvláště pak ty, které si libují v individuální symbolice. Postačí stručný slovní popis funkce a řádná specifikace dat. Dokumentace stejně zastarává a pro údržbu postačí to, co jsem právě uvedl. Stostránkové dokumentace systémy, která změni svou podobu během 5 - 10 let jsou příliš velkými přepečem.

Pracovníci výpočetní techniky jsou zvyklí na určitou přesnost. V tomto sdělení jsme se jí nechránili. Nedefinovali jsme například, kdo přesně programátorem rozumíme. Užili jsme vlastně nevyhnutelného obrazu, který má o něm uživatel. Svá tvrzení jsme statisticky neprokázali. Vzalo by to mnoho času a nebyl by to rozumně využitý čas. Berte je proto jako výsledek pozorování a určitého náhledu. A berte je, prosím, jako výzvu.