

Doc. Dr. M. Mikulík, CSc., ÚFM ČSAV Brno

**Abstrakt.** V příspěvku se pojednává o stadiích práce při vytváření programových systémů. Zdůrazňuje se, že základní podmínkou ekonomického přínosu programového systému je ekonomická oprávněnost úlohy.

### 1. Programové komplexy a jejich význam

Aplikace výpočetní techniky se stále více rozšiřují do všech oblastí lidské činnosti. Nedostatek dobrého programového vybavení snižuje efekt zavádění výpočetní techniky. Jednou z cest k nápravě je i vytváření komplexů dobře fungujících programů pro řešení specifických problémových oblastí uživatele. Při tom se razí požadavek usnadnit uživateli komunikaci s počítačem.

Jak upozorňuje J. Vaníček, lze v poslední době v oblasti aplikačního programového vybavení nejazykového charakteru vyzorovat tendenci k jakési polarizaci vytvářených prostředků ke dvěma krajním typům. Jedním z nich jsou ucelené, obvykle značně rozsáhlé, programové systémy, vytvořené s cílem komplexně řešit danou problémovou oblast. Druhou cestou výstavby aplikačního programového vybavení je vytváření dostatečné zásoby poměrně jednoduchých programových modulů, jakýchsi stavebních kamenů, realizujících často se opakující programové obraty, z kterých lze pak požadovaný program vytvářet nevelkým úsilím. Při tvorbě takovýchto typových prvků je klíčovou otázkou zajištění co nejjednoduššího spojování v rozsáhlé celky, jejich univerzality a dostatečné zásoby.

Typickými programovými systémy první skupiny bývají systémy z oblasti ASŘ. Existuje však i řada systémů z oblasti vědeckotechnických výpočtů. Příkladem je Computer-Graphics Three Dimensional Interaktive Applications System, označený CATIA. Jde o vysoce interakční 3-D grafický systém pro počítačem podporovaný návrh a výrobu konstrukcí (CAD/CAM). Systém umožňuje navrhovat různé geometrické útvary a s nimi efektivně pracovat. Pro navržené součásti umožňuje vytvářet programy pro NC-stroje, zajišťovat vazbu na banku dat, zajišťovat vazbu na mechanické výpočty metodou koneč-

ných prvků a na výpočty řešící vzájemnou polohu pohybujících se objektů. Do banky dat je možno ukládat i ty informace o navržené konstrukci, které jsou potřebné v procesu řízení podniku.

Jako příklad z druhé skupiny lze uvést programový komplex OPERACK, vypracovaný v ÚFM ČSAV v Brně. Obsahuje podprogramy a programy řady metod k nalezení uživatelem definované účelové funkce a to jak pro případ podmíněné, tak i pro případ nepodmíněné optimalizace. Dále obsahuje též řadu pomocných programových prostředků pro usnadnění práce uživatele.

Programové komplexy, zejména z druhé uvedené skupiny, bývají vzhledem k možnosti jejich prodeje často velmi popularizované. Bývají často produktem vysoce specializovaných softwarových firem. Použitý způsob řešení zahrnuje v sobě zpravidla nejnovější poznatky z příslušné aplikační oblasti. Představují velký objem práce a jejich vytvoření a uvedení do provozu bývá velmi nákladné. Údržba těchto komplexů vyžaduje velký počet kvalifikovaných pracovníků. Zaručují rychlé a spolehlivé zpracování úloh. Tyto vlastnosti umožňují uživateli poměrně brzy řešit problémy z příslušné problémové oblasti. Používání dodaných programových systémů má pro uživatele i řadu negativních stránek. Uživatel je v aplikacích omezen jeho možnostmi, musí se mu podřizovat a to někdy na úkor prospěšnosti. Většinou do systému nevidí a tedy jej nemůže rozšiřovat. Návaznost na jiné své systémy si musí často sám pracně zajišťovat. Přenos programového systému do jiného operačního systému bývá často mimo možnosti uživatele. Z toho je patrné, že rozhodnutí orientovat se na rozsáhlý programový systém, obzvláště od západní firmy, není vždy bez rizika. Tomuto rozhodnutí musí proto předcházet rozbor vlastních potřeb a možností a samozřejmě i zhodnocení příslušného programového systému, včetně kvality jeho dokumentace. Třeba provést i zhodnocení dodavatele, především z hlediska jeho podpory při zavádění systému do provozu.

Pokud se uživatel rozhodne budovat programový systém vlastními silami, musí si být vědom velkého rozsahu práce s tím spojené a tedy i vysokých pořizovacích nákladů.

## 2. Stadia práce při vytváření programových systémů

Vytváření programového systému prochází postupně několika stadii:

- přípravné stadium
- vypracování úvodního projektu
- vypracování prováděcího projektu a vlastní naprogramování
- testování
- předání systému do provozu

### 2.1. Přípravné stadium

V tomto stadiu se provádí

- formulace projektového úkolu a stanovení cílů řešení
- vytváří se pracovní tým

Postavené problémy se musí jasně formulovat a musí se zdůvodnit potřeba jejich řešení. Je zapotřebí přesně stanovit cíle řešení. Problémy musí být zařazeny do souvislostí s jinými problémy, řešenými počítačově nebo i jinými prostředky. Formulace úkolu musí vytvářet podmínky pro zavedení progresivních řešení v dané aplikační oblasti. Zautomatizování zastaralého způsobu práce prodlužuje jeho životnost a zabráňuje přechodu na modernější způsob. Dále je zapotřebí stanovit technické prostředky, kterých se má při řešení použít. Je nutno vymezit tu část problému, která se má řešit počítačově. Dále je zapotřebí vymezit kategorie uživatelů, kteří budou s příslušným programovým systémem rutinním způsobem pracovat. S ohledem na profil budoucích uživatelů programového systému je zapotřebí řešit i otázku styku člověk-počítač. Důležitým bodem přípravného stadia je stanovení celkového kritéria pro posouzení úspěšnosti vyřešení úlohy a provedení odhadu ekonomických přínosů, které vyřešení daného úkolu přinese. Základní podmínkou pro dosažení ekonomického přínosu je ekonomická oprávněnost dané úlohy a dobrá formulace. Na řešení úkolu je zapotřebí stanovit kapacity a termíny. Navržený projektový úkol by měl projít oponenturou.

Po vytvoření projektového úkolu se vytvoří pracovní tým. V něm by měli být zastoupeni jednak tvůrčí pracovníci různých profesí, jednak pracovníci, kteří budou zajišťovat pomocné prá-

ce. Pracovníci týmu musí mít zajištěn styk s řediteli a s uživateli souvisejících problémů. Vedoucí pracovník týmu musí být nejenom odborně zdatný, ale musí mít i dobrý přehled o vývojových trendech v příslušných oblastech, dobré styky a v neposlední řadě musí mít organizační schopnosti. Musí mít i dostatečnou autoritu u ostatních pracovníků týmu. Je důležité, aby počet pracovníků týmu byl optimálně navržen. V žádném případě zde neplatí přímá úměra: čím více pracovníků, tím bude úkol dříve splněn. Vnitřní organizace týmu musí zajistit, aby případným odchodem kteréhokoliv pracovníka, dokonce ani odchodem vedoucího týmu, nebylo včasné splnění úkolu ohroženo.

## 2.2. Úvodní projekt

V tomto stadiu se provádí

- shromáždění podkladů a jejich zhodnocení
- analýza problému
- návrh architektury systému a jednotlivých funkcí
- návrh způsobu pořizování dat a jejich kontroly

Po důkladném seznámení se s projektovým úkolem musí pracovníci týmu shromáždit dostupné informace o přístupech jiných organizací, našich i zahraničních, k řešení stejných a příbuzných problémů a o jejich zkušenostech s rutinním provozem těchto řešení.

V analýze celého problému se zkoumají souvislosti s návaznými úlohami a to jak z pohledu věcně-technického, tak i z hlediska počítačového řešení. Do provádění analýzy musí být zapojeni pracovníci všech potřebných profesí, včetně pracovníků, kteří jsou velice dobře seznámeni s operačními systémy uvažovaných výpočetních systémů. Výsledkem analýzy by měl být již poměrně podrobný návrh celého postupu řešení.

Pro programový systém je nutno navrhnout jeho architekturu a základní filozofii pro práci s ním. V této fázi je nutno též vyspecifikovat všechny hlavní funkce, které se mají programovým systémem zajišťovat. Je třeba navrhnout vazby na jiné systémy.

V tomto stadiu je nutno navrhnout způsob získávání a pořizování dat a navrhnout způsob kontroly jejich správnosti.

Úvodní projekt by měl být znovu opakován a to za přítomnosti zadavatele úkolu, budoucích uživatelů a dalších odborníků.

### 2.3. Prováděcí projekt a vlastní naprogramování

V tomto stadiu se provádí

- rozčlenění systému do hierarchicky spojených modulů
- vlastní naprogramování

Pro vypracování návrhu se používá řada metod. Jednou z nich je t.zv. návrh "shora-dolů". Při jejím použití se vychází z celkové architektury problému. Hlavním výsledkem návrhu je pak rozčlenění systému do hierarchicky spojených modulů, obvykle jeden hlavní řídicí modul a několik modulů podřízených. Každý z nich, pokud možno, realizuje jednu funkci systému. Jednotlivé moduly mohou mít podřízené pomocné moduly, realizující pomocné funkce. Takováto struktura může být různě hluboká. Pro formalizaci návrhu se používají tabulky, grafy a diagramy, které přehledným způsobem informují o funkci, postavení a interakci každého modulu v hierarchii.

Pro vlastní napsání programu je nutno zvolit programovací jazyk. Tvorba programu vychází z jeho hierarchické struktury. Při používání metody "shora-dolů" začíná se zápisem nejabstraktnějších úrovní. Současně s tvorbou jednotlivých modulů lze provádět testování jejich správnosti metodou "shora-dolů". Lze tedy testovat dříve, než je celý programový systém ukončen.

Je zapotřebí, aby celý komplex byl zapsán v jednotném duchu. Tím se docílí vyšší přehlednosti a zlepší se orientace v jednotlivých komponentách. K tomu se vydávají metodické pokyny pro daný úkol.

### 2.4. Testování

Testování je rozsáhlý, často diskutovaný problém. Vzhledem k tomu, že se v praktických případech musíme omezit jenom na vybraná data, nelze testováním prověřit správnost testovaného systému. Pro testování programových systémů se používá řada metod. Jedna z nich je označena jako metoda "shora-dolů". Testování za-

činá od nejvyššího řídicího modulu. Místo skutečných, zatím ještě neotestovaných modulů nižší úrovně, se použijí "slepé" moduly. Ty se při testování postupně nahrazují již otestovanými moduly. Jako výhody tohoto způsobu testování se uvádí, že se napřed testuje hierarchie celého systému a jeho nejdůležitější řídicí funkce. Testování vrcholí v testování celého systému.

### 2.5. Předávání programového systému do provozu

Při předávání systému do provozu se celý systém znovu testuje za přítomnosti zadavatele a uživatele. Někdy se ukáže, že je nutno některé funkce systému pozměnit. Při předávání systému do provozu se předává i dokumentace systému a návod na použití.

Po předání systému do provozu je zapotřebí zajistit opravování chyb, které se zjistí během provozu. Nutno zajistit, aby se provedení všech oprav dostalo do dokumentace.

### 3. Závěrečné poznámky k řešení projektového úkolu

Během všech etap práce je nutno vytvářet průběžně dokumentaci. Doporučuje se, aby se o průběhu řešení vedly záznamy, v nichž by se dokumentovaly i neúspěšné pokusy o řešení. Jsou to často důležité poznatky, které mohou být později užitečné při řešení jiných problémů.

Je řada vlastností, které by měl mít dobrý softwarový produkt. Uvedme alespoň některé z nich. Měly by se v něm především odrážet nejnovější poznatky z příslušné aplikační oblasti. Měl by být vytvořen s ohledem na technické prostředky uživatele. Vzhledem k údržbě by měl být dobře dokumentován. Pro uživatele musí být vypracován návod, který mu umožní rychle pochopit způsob použití při řešení konkrétních úloh. Celková koncepce a způsob zápisu programu by měl umožňovat lehké provádění změn. Důležitým znakem je i snadná přenositelnost systému na výpočtové systémy pod jiný operační systém.

Výsledná kvalita softwarového produktu je značně ovlivněna nutnými kompromisy mezi požadavky na kvalitu, náklady a termíny.