

Ing. Alexander Kirschner
PVT, Bratislava

Problémov v oblasti software je v Československu viac. Boli o.i. rozobrané napr. aj v /11/, /2/, či v /3/. Jedným z nich je určite aj otázka výuky a výchovy softwarových špecialistov. A práve tú chceme znova a s novou naliehavosťou v meniacich sa čs. podmienkach pripomenúť. Upozorňujeme, že tu nepojde o vzdelávanie v oblasti "čistej" matematickej informatiky /computer science/, ale len o vzdelávanie v oblasti softwarového inžinierstva /ďalej aj SE/.

Bez nároku na podrobnosť budeme tu softwarové inžinierstvo chápať ako aplikovanú matematickú informatiku /presnejšie sme sa o jeho vymedzenie pokúsili v /4/ a softwarového inžiniera ako odborníka v oblasti softwarového inžinierstva - predovšetkým navrhovateľa a implementátora hlavne aplikačných softwarových projektov /ale nie len ich/, keďže táto činnosť tvorí najčastejšie pracovné zaradenie absolventov fakúlt, ktoré sa na dnešnej "produkcii" softwaristov nejakto podieľajú. Vychádzame z toho, že takáto činnosť má dnes už prepracované základné postupy a svojím charakterom sa v zásade nelíši od iných projekčných činností, aj keď jej predmet je logicko-matematickej povahy.

V reálnych aplikačných softwarových projektoch vystupuje do popredia predovšetkým uplatnenie inžinierskych projekčných techník, ktoré sa do matematickej informatiky /computer science/, ako sa dnes vo svete všeobecne chápe /5/, nezačleňujú.

1. K obsahu výuky v oblasti softwarového inžinierstva

Podľa nášho názoru by výuka v oblasti SE mala čo najširšie naväzovať na teoretickú výuku matematickej informatiky. To po prvé. A po druhé by jej súčasťou mala byť konkrétna softwarová skúsenosť z reálneho /cvičného/ projektu. V zásade taká výuka musí pokrývať celú sféru tzv. vývojového cyklu software.

V ďalšom sa chceme pokúsiť špecifikovať v hlavných predmetových rysoch z dnešného hľadiska minimálny rozsah tejto výuky.

Usudzujeme, že vcelku možno teoretickú časť výuky rozdeliť na okruh technológií projektovania softwarových systémov a na okruh riadenia tvorby softwarových systémov.

V okruhu technológií by mal byť študent po skončení výuky schopný spracovať analýzu špecifikácií softwarového systému. Mal by poznať jazyky špecifikácií, a to ako voči zadávateľovi, tak i "voči" svojmu projektu.

Jadrom tvorivosti softwarového inžiniera sú techniky návrhu, a to ako softwarového systému, tak i programov. Ide hlavne o zácviak v disciplinovanom formulovaní projekčného zámeru s využitím metód postupnej dekompozície, abstrakcie a zjemňovania. V tejto fáze má zvláštnu dôležitosť nadobudnúť hľadisko modularity a štrukturovaného prístupu. Súčasťou návrhu je návrh štruktúry dát resp. celej bázy dát, dôležité miesto tu zastávajú metódy vyhodnocovania variantných riešení, a to hlavne s ohľadom na portabilitu či adaptabilitu výsledného produktu. Východiskom takýchto praktík sú základné časti matematickej informatiky /teória operačných systémov, základy hardware, systémové programovanie, znalosť programovacích jazykov atď. To je však už oblasť, ktorá hraničí s výcvikom vo sfére techník vývoja softwaru. Ide o informácie vzťahnuté ku operačným systémom istých počítačov, no popri prehľade by mala byť súčasťou výuky i aspoň jedna reálna sada techník /v našich podmienkach by asi najvhodnejšou bola taká, ktorá by bola orientovaná na počítače, JSEP, príp. SMEP/.

Dnes prakticky vo výuke opomínané sú techniky testovania softwarových produktov, a to počínajúc ladením programov, cez plánovanie testov, ich systematické vyhodnocovanie, akceptačné, integračné a systémové testy. Úzko na to nadväzuje problematika testovania účinnosti konečného produktu, jeho informačné zabezpečenie /ochrana/. Prava, tu nám ide len o rámcové vymedzenie niektorých problémov.

V oblasti riadenia softwarového projektu by mal byť absolvent schopný hlavne plánovať na príslušnej úrovni proces tvorby projektu.

Mal by byť poučený o metódach tímovej spolupráce a delenia práce, metódach kontrol a oponentúr. Zčasti do takéhoto "riadiaceho minima" môžu vstupovať témy ekonómiky tvorby softwaru a riadenia práce projekčného tímu. Zvláštna dôležitosť musí byť venovaná údržbe vytvoreného projektu, jeho aktualizácii a s tým súvisiacej dokumentácie, ktorá je jednak pracovným, jednak záverečným obrazom stavu projektu a prelieta sa všetkými fázami projektovania.

Uvedený výpočet okruhov považujeme len za kmeňový, je doplňovateľný a modifikovateľný podľa potreby.

2. Možné formy výuky a výchovy

Je očividné, že výuka, ktorá by mala aj všetky vyššie naznačené zložky, ak by prebiehala len verbálne, bola by málo účinná. Jej nevyhnutnou súčasťou musí byť aspoň jeden - hoci menší a modelový - softwarový projekt. Myslíme, že je to možné realizovať formou istého skupinového softwarového laboratória, prebiehajúceho paralelne s výukou. Konečne, naliehavosť takejto požiadavky sa nezdorazňuje prvýkrát /napr. v /3//.

Pokiaľ ide o inštitucionalizovanú formu výuky, je základnou dilemou to, či ju orientovať ako vysokoškolskú, alebo ako postgraduálnu. V prvom prípade by to viedlo k pozdvihnutiu všeobecnej úrovne softwarových absolventov, kým výuka v rámci postgraduálneho štúdia môže byť adresnejšia. Tým otázkou formy štúdia neformulujeme ako alternatívu. V zahraničí už existujú návrhy študijných programov softwarového inžinierstva, a to ako podgraduálne, tak i postgraduálne /7/, /8/. Ďalej sa možno stretnúť aj s rámcovými študijnými programami výuky, ako je napríklad jednosemestrový kurz softwarového inžinierstva popísaný v /9/. Také kurzy sú však vhodné ako doplnkové k inej špecializácii, napríklad ku matematickej informatike, štúdiu informačných systémov apod.

3. Jeden model študijného programu

Usudzujeme, že ako inšpiratívny príklad vysokoškolského študijného programu špecializácie softwarové inžinierstvo môže poslúžiť

štvorročného štúdia popísaný v /6/.

Štúdium sa týka týchto oblastí:

- inžinierstvo a fyzikálne vedy,
- vedy o riadení,
- matematická informatika,
- samotné softwarové inžinierstvo.

Teoretická výuka je doplnená jednak cvičeniami, jednak vo 4. ročníku laboratórnym softwarovým projektom, ktorý je dôležitou a integrálnou súčasťou štúdia.

V tabuľke 1 sú uvedené predmety samotnej špecializácie v jednotlivých ročníkoch ako i počet hodín venovaný výuke istého predmetu v príslušnom ročníku.

ročník	predmet	hod.
1.	Úvod do softwarového inžinierstva	16
	Úvod do návrhu programov	40
	Komparatívne programovacie jazyky	40
	Počítače a spoločnosť	24
2.	Architektúra počítačov a assembly	48
	Návrh programov	48
	Štruktúry dát a návrh algoritmov	32
	Interpersonálna komunikácia	40
3.	Návrh a riadenie bázy dát	56
	Návrh číslicovej elektroniky	64
	Právne súvislosti softwarového inžinierstva	24
	Výpočtové systémy a komunikácia	64
	Manažérske plánovanie	24
	Riadenie softwarových projektov	32
4.	Systémy v reálnom čase	24
	Číslicové riadenie a mikroprocesory	64
	Ľudský činiteľ vo výpočtovej technike	24
	Laboratórny softwarový projekt	48

Tabuľka 1

Obsah ťažiskových predmetov tvoria tieto okruhy.

Úvod do SE: základné problémy SE; základy inžinierskeho riešenia úloh; história SE; rola softwarového inžiniera v spoločnosti; základy výpočtovej techniky; úvod do programovania.

Úvod do návrhu programov : princíp postupného zjemňovania a jeho aplikácia na návrh štruktúrnych programov ; funkčná dekompozícia; riadiace štruktúry; notácia návrhu; dokumentácia; techniky verifikácie.

Návrh programov: štruktérne metodológie návrhu programov; analýza požiadavok; metódy Constantineho, Jacksona a Warniera; štruktúry dát a programov; techniky verifikácie.

Riadenie softwarových projektov : aplikácia teórie organizácie a riadenia projektov na vývoj software; plánovanie a riadenie projektu; analýza požiadavok a zdrojov; psychológia riadenia tvorivých tímov.

Manažérske plánovanie : plánovanie a účtovanie softwarových projektov; ekonomické analýzy pre rozhodovanie; kontrakty; odhad cien; plánovanie a rozpočtovanie.

Súčasťou štúdia sú ďalej predmety spoločného základu, ako napr. vybrané časti matematiky, ekonomiky, ďalej fyzikálnochemické a spoločensko-vedné základy. Záver tretieho a 4. ročník je zo značnej časti venovaný výberovým predmetom, ktoré si príslušná fakulta zavedie podľa svojich potrieb a možností.

Usudzujeme, že asi takto koncipovaný študijný plán špecializácie pokrýva základné požiadavky na absolventa-softwaristu a poskytuje dostatočne široký základ pre jeho rôzne pracovné zaradenie.

4. Záverečné poznámky

Východiskovým problémom pre zavedenie systematickej výuky v oblasti softwarového inžinierstva je zrejme to, že samotný výukový základ matematickej informatiky je dnes neveľmi u nás rozpracovaný. Aj to je dôvod, prečo sme tu formulovali požiadavky na výuku len rámcovo, nie ako konkrétny plán. A nehovoriac o ďalších okruhoch, ktoré by už dnes bolo potrebné do softwarovej výuky zahrnúť, ako napr. teóriu bánk dát, distribuovaného spracovania atp. Avšak to sa nám z dnešného "predštartovného bodu" javí ako okrajové, doplniteľné neskôr.

Literatúra

- /1/ Král J., Štunc J.: Softwarové inžinierstvo - teória a prax.
In: SOFSEM '76, VVS Bratislava 1976, str. 389-396
- /2/ Král J.: Současné problémy software. Informačné systémy, roč.6/77,
č.2, str. 177-187
- /3/ Brodský J., Staudek J.: Profil absolventa - softwaristky.
In: SOFSEM '76, VVS Bratislava 1976, str. 383-388
- /4/ Kirschner A.: Softwarové inžinierstvo - porovnanie prístupov
a definícií. Mechanizace a automatizace administrativy, roč.
1982, č. 6, str. 227-229
- /5/ Curriculum '78. Communications of the ACM, vol. 22/1979/, No.3,
pp.147-166
- /6/ Jensen R.W., et al: A proposed 4-year software engineering cur-
riculum. SIGCSE Bulletin, vol.3/1978/, No.3, pp. 84-92
- /7/ Fairley R.E.: MSE 79: First draft of a master curriculum in
software engineering. Software engineering notes, vol.4/1979/,
No.1, pp.12-17
- /8/ Fairley R.E.: MSE 79: Second draft of a master curriculum in
software engineering. Software engineering notes, vol.4/1979/,
No.2, pp.13-16
- /9/ Kant E.: A semester course in software engineering.
Software engineering notes, vol.6 /1981/, No.4, pp.52-76