

тият пример за наука за движение, в който употребата на функции е непрекъснато контролирана от грижата за логическите основи. Именно този пример е онова, което не е преставало да бъде следвано.

Превел: Иван Чобанов

Осъвременяване на обучението по математика и физика

ОБУЧЕНИЕТО ПО СУФТУЕРНА ТЕХНОЛОГИЯ И ПРОИЗВОДСТВО ВЪВ ФАКУЛТЕТА ПО МАТЕМАТИКА И ИНФОРМАТИКА

А. Ескенази, В. Петрова, Н. Манева

1. Терминология. Налага се да започнем с терминологията. Предмет на разглеждания курс е това, което за първи път преди двайсетина години бе наречено Software Engineering (на английски), на френски стана Genie Logiciel и на немски — Software Entwicklung. Преводът на български трябва да държи сметка за вложеното в оригиналния термин съдържание. Това обаче е изключително трудно поради невероятното разнообразие от значения, давани от различните автори. Съществено се оказва например, дали той е: теоретик (склонен обикновено да остава около спецификацията и верификацията на програми и свързани с тях формализми и методи), практик (заинтересуван от години за използване модели на жизнения цикъл на програмните продукти, средствата за неговото планиране, организиране, както и осигуряването и управлението на качеството му) или пък ръководител (стигащ до проблемите на маркетинга на софтуера и комуникацията с потребителите). Във всеки случай, поради тези причини и поради езиковото несъвпадение на английското Engineering с българското инженерство, категорично следва да се откажем от софтуерно инженерство и като следствие — от софтуерен инженер. Ще приемем (без претенции, че това е идеалният термин на български) софтуерна технология и производство (СТП). Сериозно основание за това считаме и съвременната дефиниция от [1]: „СТП интегрира технически умения, професионални знания и дисциплина и административни способности, необходими за специфицирането, проектирането, внедряването и поддържането на надеждни и рентабилни системи, в които софтуерът е съществена компонента“.

2. Положението в света. Какви са основните особености на обучението по СТП във висшите учебни заведения по света?

2.1. СТП се разглежда като задължителна част от всяко обучение по информатика. Нещо повече — могат да се посочат учебни програми по други, близки специалности, в които СТП е застъпена. Един авторитетен пример е учебната програма на Компютърното дружество на IEEE [2] по Computer Science and Engineering (по същество информатика и въпроси на хардуера). Тя се състои от 9 основни предметни области, 4 лабораторни компоненти и 15 специализирани (advanced) тематични области. Предполага се, че всяка конкретна учебна програма ще бъде подмножество на предлаганата, като дадена предметна област се развива в един или няколко курса. СТП е сред малкото на брой предметни области, фигуриращи и в трите групи.

2.2. Като правило се предвижда не един, а няколко взаимно свързани курса по СТП в течение на няколко семестъра. Така в програмата на Тексаския християнски университет [3] те са 7 задължителни и 8 избираеми, а в тази на Сиатълския университет [4] — съответно 7 и 9.

2.3. Задължително е изпълнението на 2—3 софтуерни проекта, обикновено с продължителност на изпълнение от 6 до 12 месеца за всеки. При това по обясними причини се работи в групи от няколко студенти, върху сериозни реални задачи, с особена тежест върху работата в колектив (характерна за практическото софтуерно производство) и произтичащите от това проблеми на организация, управление, документирание, инспекции и т. н.

2.4. Непосредствено свързано с тези проекти е наличието на подходящ и достъпен хардуер и софтуер. Изглежда, че в развитите страни (а и не само там) въпросът с хардуера е решен окончателно. Пример 1: съгласно [3] на разположение на студентите е един компютър VAX 6320 със 128MB ОП(1)

и 6.5 GB дискова памет (ДП), един VAX 11/780 с 16MB ОП и 1.5GB ДП, значителен брой микрокомпютри и работни станции TANDY 3000, Macintosh + и SE, SUN 3/60, специализиран графичен хардуер. Пример 2: по наши лични наблюдения в Университета Чинггуа (Пекин) студентите по информатика разполагаха в началото на 1988 г. със значителен брой микрокомпютри, мини-машина Gould 32/2750, супермикро VV68000, станция Apollo, а заедно с други факултети ползваха мощни миникомпютри DRS-8 и Elxis 6400 с голям брой терминали.

2.5. Както всяка специалност информатиката също се развива. При нея обаче това засега е много по-динамичен процес. СТП като част от информатиката е заела мястото си сравнително късно, но затова пък в момента има водеща роля. Пример: сред специализираните предметни области на учебната програма на IEEE от 1977 г. [2] СТП не фигурира; през 1983 г. СТП е вече на челна позиция.

2.6. Макар да не заемат винаги основно място специално сред курсовете, съставлящи учебната програма по СТП, трябва да се спрем на тези, посветени на проблемите на комуникацията. Защо? Оставяме настрана генералния дефект на нашата образователна система, породен от фактическата липса на обучение и възпитание по начините на общуване (писмено и устно, в различни формации от хора). Това е прекалено голям проблем за тази статия. Искане ни се обаче да обърнем внимание, че при производството на софтуер възниква непрекъснатата необходимост от общуване на производителите както помежду им, така и с ръководителите им, а също и с клиентите и потребителите. Не са ни известни български статии, още по-малко учебници, в които системно да се разглеждат проблемите на точния, недвусмислен, компактен и навременен обмен на информация. Няма научно обосновани препоръки, още по-малко методики за общуването с клиентите и потребителите на произвежданите програмни продукти, с тези, които ги поръчват (и следователно би следвало да формулират изискванията си към тях). Единствено по-добро е положението с видовете документация (потребителска и производствена). С това считаме, че поне сме очертали проблема на общуването в рамките на СТП.

3. Кратка история на преподаването на СТП у нас. Първият курс по софтуерна технология и производство беше четен в продължение на 3 години като спецкурс във ФМИ под името „Организация и производство на програмно осигуряване“ започвайки през летния семестър на 1984/1985 уч. г. Учебната програма и лекциите бяха подготвени от А. Ескенази и Вл. Занев и бяха предназначени за студентите от 4. и 5. курс към секторите по информатика на тогавашния Единен център по математика и механика. Хорариумът беше 45 часа (15 седмици по 3 часа) лекции. През 1986 г. същите автори подготвиха учебна програма по дисциплината „Производство и внедряване на програмно осигуряване“ за студентите от трети курс на ФМИ по специалността информатика. Курсът се запазил като едноместришен с хорариум 45 часа лекции, но и с 15 часа упражнения. За първи път той беше прочетен през 1987/1988 уч. г. от А. Ескенази, като упражненията бяха водени от В. Петрова и Н. Манева. От 1988/89 уч. г. лекциите се четат от В. Петрова и Н. Манева. През 1986 г. А. Ескенази и Вл. Занев подготвиха тезиси по лекциите, които бяха издадени на циклостил, преди всичко за нуждите на следипломната квалификация за учители по информатика към ФМИ. За тях курсът се чете поне веднъж годишно в съкратени варианти от 30 или 18 часа лекции. Отскоро курс по своя програма чете Г. Зеленков за част от студентите на ВИНС Варна.

4. Учебна програма.

4.1. Цели. През упоменатия вече 6-годишен период учебната програма е претърпяла изменения и по структура, и по съдържание. Причините са очевидни — натрупаният опит, появяването на 15 часа упражнения, измененията в областта (теоретични, практически, нормативни), по-дълбоко запознаване с чуждия опит, измененията на изискванията на пазара и др. Може обаче да се счита, че основата цел на курса остава в рамките на формулировката на учебната програма от 1986 година — да се запознаят студентите с основните проблеми на производството и внедряването на програмни продукти, както в теоретичен аспект (чрез разглеждане на подходящи модели на жизнения цикъл, на операциите и на документите), така и в практическо отношение (от гледна точка на действащите в България нормативни актове и стандарти). При това, специално внимание се отделя на въпросите на качеството на софтуера и на автоматизацията на софтуерното производство.

4.2. Предпоставки. Курсът по СТП не може да бъде сред началните курсове в обучението. Той изисква известна предварителна подготовка от обучаемия. Безспорно трябва да е преминал кратък уводен курс по основи на информатиката. В това се включват общи познания по алгоритми, начини на представянето им, представяне на данните в компютрите, архитектура на изчислителните системи. В рамките на такъв или по-добре на отделен кратък курс следва да са изучени основните структури данни, както и 1—2 езика за програмиране. Например, за отделни теми са необходими минимални знания по графи. Без да се разпростираме повече, можем да считаме, че от казаното следва възможността курсът или серията курсове да започнат още от трети семестър. Въпреки това неслучайно във ФМИ това става през шестия семестър. Картината по света е аналогична. Причината е в необходимостта от известен опит и зрялост у студентите. Някои американски университети изискват предварителен поне 2-годишен стаж в производството на софтуерни системи. Това, че студентът е написал десетина учебни програмки и още няколко по-големи, знае 1—2 езика за програмиране и основните неща от информатиката едва ли ще му позволи да обхване, осъзнае и почувства сложността и многообразието на дейностите по създаването на даден програмен продукт, взаимната

им обвързаност, значението на наглед странични и досадни операции (особено свързани с планирането, тестването и документирането), необходимостта от неприятни формални процедури.

4.3. Съдържание. При определения в учебния план хорарен курсът по ПВПО не може да бъде нищо друго освен увод в основни въпроси на СТП. Както личи от 4.1. и става ясно по-долу, стремежът на авторите на учебната програма и на лекторите е да дадат идея за основния кръг проблеми на СТП и същевременно да подготвят донякъде студентите за бъдещата им дейност като производители на софтуер. Лекциите обхващат следните теми:

Тема 1. Същност на СТП. Концепция за програмен продукт (ПП). Програмна промишленост. Тази тема е уводна и целта ѝ е да представи основните понятия, принципи и проблеми на СТП. Съдържанието ѝ личи от заглавията на темите, включени в курса. Специално се очертава спецификата на програмното производство спрямо традиционното промишлено производство.

Тема 2. Живен цикъл на ПП. Модели на ЖЦ. Целта е да се представи пред студентите систематичен подход към разглеждането на целия процес на създаване и използване на ПП. Следвайки [5], класифицираме моделите по форма и по съдържание. Разглеждат се характерни особености и примери на хронологични, функционални и многомерни модели. Изяснява се динамиката на изискванията в ЖЦ — видове, еволюция, промяна, специфицирането и разпространението им. Темата се представя на две нива на сложност: в началото като увод в ЖЦ, етапите и дейностите на процеса, а по-нататък, след по-детайлно запознаване, представата за цялостния процес на създаване и използване на софтуер се обобщава чрез еволюционния [6] и спираловидния модел [7]. Подробно се разглежда двумерният модел на Гънтър [8], като се характеризира всяка от шестте фази и седемте функции.

Тема 3. Същност и реализация на основните дейности в програмното производство. Целта на темата е да представи същността и организацията на дейностите — както за отделна разработка, така и в предприятие за производство на софтуер.

3.1. Управление на проекти — съставяне на проектно задание, планиране на разработката, организация на колектива, времеви оценки и фактори за ентропията при реализацията на проектите.

3.2. Технологична същност на дейностите: планиране — предназначение и структура на общите и на някои конкретни планове; разработване — същност и подходи за организация на проектирането (предварително и подробно, средства и езици за описание на проекти, методи за проектиране) и на програмирането (структурно програмиране, модулност); изпитания — класификации в зависимост от целите, съдържанието и извършителя, настройване и тестване (обекти, стратегии, методи и организация на провеждането им); поддържане и съпровождане — специфика, организация и подпомагачи ги инструментални средства.

Тема 4. Икономика на програмното производство. Видове разходи в производството, икономически показатели. Разглеждат се задачите на анализа на икономическите показатели и подходи към изследване и управление на разходите, възможните принципи за ценообразуване при ПП и за предварително оценяване на разходите на производството. Представя се икономическият модел на Боем [9] и реализираната на негова основа в ИНТЕРПРОГРАМА експертна система РЕСУРС-ЭКСПЕРТ.

Тема 5. Качество на ПП.

5.1. Въвежда се понятието „качество“ на ПП, формулират се изисквания към качествените характеристики, модели на качеството и оценяването му. Разглеждат се йерархичните модели (като пример — моделът на СИВ [10]) и съответстващите им методи за оценяване. Представя се и класификационният метод [11].

5.2. Дефинират се задачите на управление на качеството и механизъмът на реализирането му. Обръща се внимание на подходите към стандартизиране на качеството — по отношение на продукта и процеса — илюстрирани чрез стандартизационните методи на ANSI/IEEE [12] и ISO 9000—9004 [13].

5.3. Метрики. Представят се метриките като средство за количествено изразяване на качествени свойства на ПП. Дават се класификации на метриките, изискванията към тях и начините за доказване на валидността им. Разглеждат се някои най-често цитирани в литературата и доказателно полезността си метрики — за четимост на документацията, за сложност на програми (Холстед, МакКейб, Гонг—Шмид), комплексна метрика на Рехенберг. Представят се и някои динамични метрики и метрики за сложността на управлението и информационния поток в многокомпонентни системи.

Тема 6. Разработване на ПП в България. Излага се действащата нормативна уредба за разработване и разпространение на ПП у нас — Наредба 8 (ДКНТП, 1982), БДС ЕСПО (структура, съдържание и задължителност на съпровождащите и експлоатационни документи), Наредба 1 (КИ, 1987). Прави се сравнение на дефинираните в тях ЖЦ и процеси с вече изучаваните. Представят се някои чужди и международни стандарти, като се сравняват с действащите у нас. Разглеждат се и въпроси, свързани с правото на собственост и защитата на софтуера като интелектуална собственост, както и с фирмената организация в софтуерното производство у нас.

Тема 7. Маркетинг и софтуер. Дефинират се основните понятия, свързани с маркетинга — основна маркетингова концепция, сегментиране на пазара, маркетингов комплекс, позициониране на продукт, пазарен ЖЦ на ПП. Подчертава се необходимостта от прилагане на маркетингови методи в условията на пазарна икономика с отчитане на спецификата на маркетинга на софтуер.

Тема 8. Автоматизация на софтуерното производство. Разглежда се необходимостта от автоматизацията на софтуерното производство и препоръчителните стъпки за осъществяването ѝ. Коментират се двата най-често използвани подхода — на индивидуални средства и мощни интегрирани системи. Привеждат се примери за реално прилагани средства. Съпоставят се автоматизирани средства, класифицирани като подпомагащи СТП (CASE), среди за СТП (SEE) и използвани в софтуерните фабрики.

4.4. Упражнения. Целта на упражненията е конкретизиране и илюстриране с примери на разглежданите в курса въпроси и извършване на самостоятелна работа по създаване на учебни проекти. За съжаление предвиденият в учебния план хорариум от 15 часа (7 двучасови упражнения, провеждани през седмица), позволява реализирането само на илюстративен подход, и то само по някои от разглежданите теми. Поради големия брой студенти в група (средно 30 души) и невъзможността да се осигурят нормални условия за работа с компютри, упражненията са преди всичко семинарни. Тематично упражненията се групират по следния начин:

I. Нормативно определени документи за разработване и реализация на ПП у нас. Избран е необходимият минимум такива документи — договор, технико-икономическо задание (ТИЗ) и задание за разработка (ЗР) като основни и инициализиращи всяка конкретна разработка. Разглеждат се предназначението, структурата, процедурите на съставянето и утвърждаването им. Експериментирани са няколко варианта на тази група упражнения — с различна степен на подробност и върху различни примери.

II. Управление на проекти. Разглежда се механизъмът на административното управление на проекти, възможностите за автоматизиране и изискванията към реализиращите го средства. Правят се демонстрации и студентите работят с такъв ПП.

III. Модели на качеството. Представа се моделът на СИВ за качеството на софтуер и методиката за оценяване по модела. Илюстрира се йерархичната структура в дълбочина чрез избрани фактори.

IV. Метрики. Прилагат се някои от разглежданите в курса метрики за изследване на текстове на програми върху обозрими конкретни примери.

Упражненията завършват с контролна работа, включваща въпроси от тематиката на упражненията и задачи — върху дадени програмни фрагменти да се приложат разглежданите метрики.

5. Изводи от натрупания опит. При увеличаване на хорариума на курса, той би следвало да включи формални методи в ранните етапи на ЖЦ — например езици за описание и средства за реализация на проекти. Изложението следва да се обогати с повече примери от производството на софтуер в нашата страна, ако се намери начин за достъп до такава информация. За упражненията е най-добре да се формират програмни колективи от 5—10 души. На всяка такава група да се възлага проектно задание в обозрим обем, което да бъде проектирано, разработено и документирано в рамките на семестъра и защитено в края му. На ефективността на провеждане на лекциите и упражненията отрицателно се отразява липсата на размножителна техника, която би спестила диктуването на съдържанието на някои нормативно-определени документи, справочна информация за метриките и др. Многообразен и голям е броят на използваните литературни източници, повечето от които не са лесно достъпни. Това обуславя необходимостта от учебно помагало, което да отразява систематично и подробно включения в курса материал. Би могло да бъде обмислена процедура, по която ФМИ да получава от водещите фирми — производители на софтуер у нас, програмни продукти, които да се използват за демонстрации пред студентите. Както показва световната практика, това е не само начин да се демонстрира полезността на някои препоръчвани от теоретичните изследвания подходи, но и да се осъществи предварително запознаване на студентите със стила и насочеността на отделните фирми, в които те биха постъпили на работа.

Литература

1. J. W. Brackett. The Boston University Software Engineering Graduate Program. *Journ. of Systems and Software*, 10 (1989), 267-269.
2. J. T. Cain, G. G. Langdon, M. R. Varanasi. The IEEE Computer Society Model Program in Computer Science and Engineering. *Computer*, 17 (1984), 8-22.
3. J. Comer, D. Rodja k. Software Engineering Education at Texas Christian University: Adapting a Curriculum to Changing Needs. *Journ. of Systems and Software*, 10 (1989), 235-244.

4. E. Mills. The Master of Software Engineering (MSE) Program at Seattle University. *Journ. of Systems and Software*, 10 (1989), 245-252.
5. А. Ескенази, Вл. Занев. Организация и технология на софтуерното производство. София, 1986.
6. H. Krasner. Requirements Dynamics in Large Software Projects. *Information Processing '89, Proc. of the 1989 IFIP World Congress*, p. 211-216.
7. B. W. Boehm. The Spiral Model of Software Development and Enhancement. *Computer*, 21 (1988), 61-72.
8. Р. Гантер. Методи управления проектированием программного обеспечения, М., 1981.
9. Б. Боэм. Инженерное проектирование программного обеспечения, М., 1985.
10. Обща методика оценки качества программных средств. *МИК ВС Бюллетень*, вып. 1 (37), М., 1988.
11. A. Eskenazi, V. Angelova. A New Method for Software Quality Evaluation. *Journ. New Generation Comput. Systems*, 3 (1990), 47-53.
12. ANSI/IEEE 730-1983 Standard for Software Quality Assurance Plans.
13. ISO 9000-9004 Standards for Quality Systems.

ИСТОРИЯТА НА ЕДИН ЕКСПЕРИМЕНТ*

Л. П. Бенезет**

През 1987 г. в нашето списание се появи превод на доклада на известния американски математик Хаслер Уитни, изнесен на конгрес на Международния математически съюз (*Физ.-мат. спис.*, 29 (1987), 43—48). В тази работа се цитира една статия, излязла преди повече от половин век, автор на която е един инспектор по математика в щата Масачузетс — Л. П. Бенезет. Благодарение на любезността на проф. Уитни се снабдихме с копие на текста, публикуван в мъчно достъпно у нас списание. Надяваме се, че читателите ще посрещнат с интерес разказа на Бенезет и ще се убедят, че, от една страна, няма нищо ново под слънцето и освен това, че за да победи разумът, са необходими дълги и настойчиви усилия, особено в областта на образованието. — *Бел. ред.*

1. През пролетта на 1929 г. покойният Франк Д. Бойнтън, главен инспектор на училищата в Итака (Ню Йорк) и председател на главния инспекторат, изпрати на някои свои приятели и колеги статия върху програмата на държавните училища. Основната идея в нея беше, че към учебната програма постоянно се добавят нови неща (правила за безопасност, здравни инструкции, начини за икономия на финансови разходи и др.), но никой не предлага да се съкрати нещо от нея. Призивът му сякаш беше: „Покажете ми как да изхвърлим нещо от учебния материал!“ Това ми напомни известното сравнение на МакАндрю, че програмата на началното американско училище е като тавана в къщата на семейство Джоунс — откакто са се нанесли в тази къща преди 50 години не са изхвърлили нещо от нея.

Един месец по-късно написах на Бойнтън писмо от осем страници, в което му казах кое според мен би могло да се премахне от нашата действаща програма. Ще цитирам два параграфа от него:

На първо място, струва ми се, че пратихме много време в началното училище за неща, които децата трябва да прескочат или отложат, докато не почувстват нужда да ги изучат. Ако имаше начин, щях да премахна аритметиката от първите шест класа. Вместо това щях да оставя децата да играят на „покупки и продажби“ с имитирани пари, защото къде другаде освен в магазина на едно единадесетгодишно дете му се налага да използва аритметиката?

Чувствувам, че е безсмислено да отнемаме осем години от основното училище, за да преведем децата през обикновената програма по аритметика на началния курс. Каква нужда има едно десетгодишно дете от уменията да дели многоцифрени числа? Цялата аритметика би могла да бъде отложена до седмата година в училище и всяко нормално дете ще я овладее за две години.

* The Story of an Experiment. *Journ. Nat. Educ. Ass.*, 24 (1935), 241-243, 301-303; 25 (1936), 7-8. — *Бел. ред.*

** L. P. Benezet (Manchester, New Hampshire). — *Бел. ред.*