

## ОБУЧЕНИЕТО ПО СОФТУЕРНИ ТЕХНОЛОГИИ: ОЧАКВАНИЯ ЗА КРИЗА ИЛИ КРИЗА НА ОЧАКВАНИЯТА

**Нели Манева**

ИМИ на БАН, 1113 Акад. Г. Бончев, бл.8, София

*nelman@gbg.bg*

**Резюме:** Първият курс по Софтуерни технологии (СТ) беше подготвен и четен през 1984/1985 учебна година за студентите от ФМИ. Днес този курс присъства в програмите на всички бакалавърски степени на информатичните специалности, като при един и същ хорариум се забелязват различия най-вече в избора и задълбочеността на представяне на материала в лекциите, и начина на организиране на упражненията и практикумите. В статията се разглеждат някои организационни проблеми на обучението по СТ, анализирани от гледните точки на основните заинтересовани страни, като са споменати някои тенденции, бъдещи тревога. Описан е и формален подход, който може да се прилага в процеса на обучение, за да подпомага вземането на решения в критични ситуации.

**Ключови думи:** софтуерна криза, софтуерни технологии, обучение по СТ, сравнителен анализ

### 1. Въведение

Сред професионалистите в направление Софтуерни технологии и производство, възникнало за справяне със софтуерната криза, витае идеята за нейната непреходност. Но по дефиниция, „криза е повратна точка в развитието на нещо; решителен или критичен момент или етап, след който има подобрене или фатален край“. Затова се счита, че е необходимо терминологично уточняване – да не се говори за „софтуерна криза“, а за нещо, което **продължава дълго и/или се повтаря често, непрекъснато създавайки проблеми**. За български еквивалент на въведения от Пресман [3] термин „*chronic affliction*“ предлагаме „**хронична недостатъчност**“ и даваме следното определение: **Целта на СТ е да се разработва, предлага и използва софтуерът така, че да се преодолява хроничната недостатъчност и да се удовлетворяват очакванията на обществото**.

В раздел 2 на настоящата статия се дискутират някои тревожни тенденции, свързани не само с обучението по СТ, но и с по-нататъшната реализация на завършващите студенти. В раздел 3 са представени гледни точки на основните участници в процеса – студенти, преподаватели и работодатели от софтуерния бизнес. В раздел 4 се споделят някои идеи за подобряване на текущата ситуация. В раздел 5 е описано накратко как формалният метод на

Сравнителния анализ може да се използва в процеса на обучение по СТ и да подпомага вземането на решения. За илюстрация на приложимостта на метода е даден пример за сравняване на различни методи за усвояване на нови знания от софтуерните специалисти, за да се избират най-подходящите в контекста на конкретна ситуация.

## 2. Обучението по СТ – очакваме ли криза?

Обсъжданията с колеги и собствените ми наблюдения при преподаване в няколко университета показват **хронична недостатъчност на всички видове ресурси**, които са съществени за осигуряване на качествено обучение по СТ. Без да ги подреждаме по важност, ще споменем тези, намаляването на които изглежда критично.

Преди всичко се забелязва тенденция за **намаляване на студентите** в информатичните специалности. Причините са най-различни – демографският срив, „отвореността“ към чуждестранни университети и програми за мобилност, отлив от инженерно-технологичните специалности и пренасочване към други, по-лесни за следване и завършване хуманитарни специалности. Поради съществуващата сега система на финансиране на университети въз основа на броя на обучаваните студенти, се развихри борба между ВУЗ-овете за привличане и задържане на студенти. За съжаление, предприетите мерки за понижаване на критериите за прием и олекотяване на изучавания материал не съдействат за постигане на изискваното (и по европейските стандарти!) качество на обучение.

Дефицитът на кадри в софтуерната индустрия доведе до наемане за работа в софтуерните фирми на студенти от информатичните специалности, и то още от първите курсове. Справянето им с работата (поради факта, че им се възлагат стандартни, предимно несложни програмистки задачи или сервизни функции) поражда у работещите студенти измамното чувство, че за да си успешен софтуерен разработчик, не е необходимо да си дипломиран бакалавър или магистър. В резултат на това има голям брой недипломирани магистри, отлагащи с години завършването си. Незащитаването в срок създава не само административни трудности, но и проблеми с ритмичността на възлагането, ръководството и проследяването на работата на дипломанта, така че обемът и качеството на разработваната дипломна работа да удовлетворяват променящите се във времето критерии.

Усложняват се и проблемите с преподавателите. Наблюдава се устойчива тенденция на ежегодно увеличаване на реалната натовареност на преподавателите (като брой, „приравнени“ към аудиторни, часове за учебна година). Това се дължи както на необходимостта да се „покриват“ няколко информатични специалности и магистърски програми, така и на намаляване на броя на преподавателите по СТ, поради пенсионирането на действащи и

липсата на новопостъпващи. Разликата в заплащането във ВУЗ и софтуерните фирми е драстична и всъщност малък брой млади хора могат да бъдат привлечени само с изтъкване на „престижността“ на преподавателската и научно-изследователската работа.

Намаляването на финансовите средства (и бюджетни, и целево отпускани) за обучаващите организации – университети и научни организации с акредитация за обучение – налагат ограничения с дълготрайни последици. Преди всичко е невъзможно да се създаде и поддържа задължителната за обучението по СТ модерна технологична среда – добре оборудвани компютърни зали и съвременни софтуерни системи. Трудно е закупуването на учебна и научна литература, както и поддържането на абонамент за списания и достъп до световни информационни ресурси.

Съгласно най-новите мениджърски теории парите не са най-важният ресурс, но липсата им е пагубна за други два ресурса – човешкия и времевия. Това означава, че безпаричието във ВУЗ ще ни остави без добри студенти и без високо-квалифицирани, мотивирани и всеотдайни преподаватели, обречени да преподават на черни (и в най-добрия случай на бели!) дъски знания, които са морално остарели и неприложими в съвременния софтуерен бизнес.

Направените по-горе констатации не са нови, но отразяват факта, че посоката и темповете на развитие на симптомите за криза са такива, че застрашават качеството на обучението по СТ и могат да предизвикат сериозни сътресения в компютърно-зависимия общественоекономически живот в страната.

### 3. Криза на очакванията

Считаме, че по отношение на обучението по СТ сме свидетели на **криза на очакванията**, която се изразява в примиряване с текущата ситуация и минималистичност на поставяните цели. От психологическа гледна точка има обезверяване, чувство на безизходица и приемане на положението като непроменимо.

Голяма част от **студентите** искат диплома, а не придобиване на трайни знания и умения, с които да стартират успешна професионална кариера. Дефицитът на софтуерни специалисти на пазара на труда не налага сериозна проверка на нивото на кандидатите при наемането им на работа. Студентите не се стремят и към по-висок успех, защото той се взема предвид само при кандидатстване за стипендии (малки по размер, отпускани при сериозни ограничения за социален статус), или класации за обменни бакалавърски, магистърски или докторантски програми. В повечето случаи успехът от следването или по избрани дисциплини дори не е сред факторите, определящи избора на кандидат за дадена позиция в софтуерна фирма.

**Преподавателите** не са мотивирани да повишават нивото на професионалната си изява. Прекаленото (в сравнение с европейските норми) натоварване ограничава възможностите им за научно развитие и за обновяване на учебното съдържание така, че то да съответства на съвременните научно-приложни постижения в областта. Процедурите за промяна на хорариума и/или съдържанието на учебните планове и програми са ненужно сложни и бавни и не съответстват на динамиката в областта. Допълнителна демотивация у преподавателите се поражда и от липсата на интерес у студентите, измервана с ниска посещаемост или неактивно присъствие. Получава се порочен кръг – студентите не идват на лекции, защото работят, а работят зле, защото имат съществени пропуски в изучавания материал.

Голяма част от **работодателите** в софтуерния бизнес са разочаровани от съдържанието на учебните програми и общото ниско, по тяхно мнение, ниво на подготовка на завършващите студенти. Ръководителите на софтуерни компании, имащи предимно финансово-менеджърска квалификация, не могат да разберат и оценят значението на т.н. основополагащо знание (body of knowledge) в областта, и настояват за изучаване само на разпространените в момента софтуерни технологии, методи и средства. В много случаи те прибягват към организиране на собствено обучение, като се придържат към японския подход за корпоративна политика на подбор и „изграждане“ на подходящ професионален профил на служителите за всяка позиция в софтуерната организация.

Последно, но не и по значение за професионалната общност, е неактивната позиция на повечето **потребители** на софтуер. Специално за България, почти няма систематичен подход за проучване на мнението им и използване на резултатите от обратната връзка с тях, като елемент от добре обмислена и осъществявана политика на софтуерните фирми. За съжаление, притиснати от ограниченията, които финансовата криза налага, и потребителите на софтуер са се примирили и приемат безкритично предлаганите им софтуерни продукти. Силно занижени са изискванията им към качеството, като обичайните претенции са свързани само с надеждността на функциониране или с някои критични за използването на съответното приложение характеристики. Висококачествените ПП са скъпи за българските потребители от средна ръка и затова за разработване и внедряване се избират софтуерни системи с приемлив компромис на съотношението цена-качество. Оправданието е, че технологичното развитие е толкова бързо, че скоро тези системи ще остарееят и вече финансово замогналите се фирми ще успеят да ги сменят. В резултат на този подход в страната се наблюдава многогодишно използване на лоши ПП, които компрометират както

повсеместната компютризация, така и професионалната ни общност, приела отговорностите за осъществяването ѝ.

#### **4. Какво да се прави?**

Посочихме някои проблеми, съществени за определяне на съдържанието и резултатите от обучението по СТ. Ще споделим някои идеи за това, което би могло да се направи – с не много финансови средства, но чрез добре замислени и обсъждани в професионалната ни общност програми, с реализацията на които биха могли да се ангажират представители на всички заинтересовани страни.

Считаме, че преди всичко е необходима целенасочена просветителска работа и обучение на участниците от всички нива за създаване на нагласи и за формиране на правилно отношение към проблемите на СТ, с отчитане на ролята на направлението в съвременното общество, основано на знания. Специално за качеството на обучението, освен системите за контрол във всеки ВУЗ, би било полезно да се усъвършенства и системата за акредитация на университетите и на специалностите, така че да се постига и регулиране чрез механизмите на конкуренцията.

Софтуерният бизнес, чрез отделните фирми или чрез браншовите си организации, би могъл да се включи заедно с университетите в кандидатстване за европейски и национални проекти, свързващи иновациите с обучението. Съществуващата сега система за спонсориране може да се разшири с нови форми – не само откриване на компютърни кабинети, но и предоставяне на фирмени разработки за целите на обучението в основните курсове и/или практикумите; финансиране на проекти за разработване на нови курсове, използване на преподавателите по СТ като консултанти във фирмени иновативни проекти и т.н. Има вече и добри практики за включване на подбрани представители на софтуерния бизнес в учебния процес като поканени лектори или консултанти за практикуми, разбира се, при ясно формулирани изисквания за квалификация, умения за преподаване и мотивираност.

#### **5. Един подход за повишаване на качеството на преподаване на СТ**

Направените по-горе констатации дават представа за последиците от хроничната недостатъчност в областта на софтуерното производство и отражението им върху процеса на обучение. За съжаление, повечето от причините за това състояние са външни и почти непроменими от общността на преподавателите, които могат само да се съобразяват с тях. Затова предлагаме един осъществим подход, влизащ в преподаването формален метод, който има потенциала да подобри както разбирането и осмислянето на материала, така и организацията на учебния процес.

Според Пресман [4], създаването на софтуер може да се разглежда като процес на решаване на проблеми (problem solving), при който във всеки момент се формулира проблем, анализират се алтернативите и се избира оптималното (според ситуацията) решение. Нашите изследвания показваха, че с подходящо формулиране на обхвата, сложността и контекста на разглеждания проблем, можем да подпомогнем процеса на вземане на решения така, че да се осигурява систематичност и обоснованост. Разработеният от нас метод за многокритериален избор е подробно описан в [4]. Тук ще представим накратко само основните му идеи, като се спрем най-вече на възможностите за използването му в преподаването на СТ.

### 5.1. Същност на Сравнителния анализ и процедура за прилагането му

В психологията вземането на решения се разглежда като основна човешка дейност, илюстрираща нивото на интелигентност. От прагматична гледна точка е важно, че успешното осъществяване на тази дейност определя професионалния и личностен просперитет на всеки индивид. Затова се насочихме към разработване на метод и съответно средство, подпомагащи процеса на вземане на решения.

Под **Сравнителен анализ** (СА) ще разбираме изследване на качествено съдържание на еднотипни обекти и съпоставянето им така, че да се избере най-добрият, да се подредят според качествено им съдържание или да се класифицира всеки обект към определена качествена категория.

Прилагането на СА се осъществява от **Аналитик**, отговарящ за всички аспекти на използването му и **Потребител**, поръчал анализа, за да вземе важно за него решение. В зависимост от ролята на потребителя, от дефинирания проблем и разглеждания момент, трябва да се опише **ситуация**, определяща контекста на провеждания сравнителен анализ. Всяка ситуация се задава с наредена шесторка:

**Ситуация** = {гледна точка, цел, обект, конкуренти, задача, ниво},

където **гледната точка** представя ролята на Потребителя; **целта** може да е характеризирание, оценяване, прогнозиране и всяка друга, зададена от Потребителя; **обектът** представя изследваното, **конкурентите** описват сравняваните обекти; **задачата** е да се определи обектът с най-висока стойност на измереното качество, да се подредят конкурентите по качество или да се класифицират; **нивото** определя сложността на провеждания анализ в зависимост от значимостта на решавания проблем и отделените за това ресурси.

Един от най-интересните (и сложни!) проблеми в прилагането на сравнителния анализ е създаването на **модел на качеството** за изследвания обект. Обикновено се прилага инкрементален подход – при поява на нов обект се създава негов първоначален модел, който се съхранява и

може да бъде разширяван за целите на всеки следващ анализ. Така в резултат се получава **базов** модел с йерархична структура, при който **коренът на дървото** представя качеството на обекта; на **първото ниво** са **факторите** – потребителско-ориентирани характеристики на обекта, с определено тегло; а на **второто и следващите нива** са **критериите** – характеристики на качеството, присъщи на разглеждания обект. В зависимост от обстоятелствата, може да се конструира и **частичен** модел, който да представя изследваното в момента качествено съдържание на обекта.

За да се улесни използването на метода е разработена и постъпкова процедура, която описва последователността на извършваните действия:

**Стъпка 1.** Идентифициране на проблем и определяне на контекста на сравнителния анализ – дефиниране на ситуация.

**Стъпка 2.** Построяване на модел на съответното за ситуацията качествено съдържание на обекта.

**Стъпка 3.** Определяне на множеството от сравнявани обекти.

**Стъпка 4.** Оценкаване – попълване с данни на двумерна таблица, редовете на която съответстват на сравняваните обекти, а стълбовете – на разглежданите фактори. Процедурата за оценкаване се осъществява от Аналитика с помощта на експерти, като се използва създаденият йерархичен модел на качеството с определени тегла за всяка характеристика и метрики за всяко листо. Оценкаването се извършва възходящо. Започва се с прилагане на метриците от последното ниво, след това се изчисляват мерките на характеристиките от следващото по-високо ниво и така, докато се изчислят мерките на факторите, с които се попълват съответните елементи на двумерната таблица.

**Стъпка 5.** Изпълнение на задачата на СА – въз основа на оценките на факторите за всеки от изследваните обекти и теглата им може да се реши поставената задача – да се определи кой е най-качественият обект, да се получи сортиран списък на обектите, подредени по качество или да се определи принадлежността на обект към предварително зададени класове на качеството. Ще отбележим, че е съставена съвкупност от различни по теоретичната си сложност методи за многокритериален избор, които са програмно реализирани и могат да се прилагат чрез разработено инструментално средство, с отчитане на дефинираното в ситуацията ниво на сложност.

**Стъпка 6.** Анализ, интерпретиране и обобщаване на резултатите. Подготовка на отчети и представянето им на Потребителя.

**Стъпка 7.** Планиране и осъществяване на следващи дейности.

Разработеният метод е вграден в цялостна методология INSPIRE [3], като е създадена и концепция за интегрирана система, подпомагаща

систематичното прилагане на тази методология в теорията и практиката на софтуерните технологии и производство.

## 5.2. Използване на Сравнителния анализ в обучението по СТ

За доказване на осъществимостта на разработения подход, той беше проверен за няколко дейности в софтуерното производство с различна значимост, обхват и сложност. Досега проведените експерименти са за дейностите тестване, осигуряване на качеството, осигуряване на ползваемостта, управление на човешкия фактор, управление на софтуерно разработване с подизпълнители, оценяване на иновативен метод в СТ и осъществяване на програма за подобряване на качеството на процеса на разработване. Наред с конструирането на модели за съответните изследвани обекти, са идентифицирани и конкретни моменти от жизнения цикъл, в които СА може да подпомага вземането на решения в критични ситуации. Допълването на стандартните лекции за тези дейности с получените резултати от прилагане на СА биха показали на студентите как цикълът на разработване може да се съчетае с проверки и анализи, подпомагащи вземането на решения в критични моменти.

В досегашното прилагане на СА са създадени модели на различни обекти в СТ – продукти, процеси и ресурси. От методическа гледна точка запознаването на студентите с модели на качеството на даден софтуерен обект дава възможност те да разберат кои характеристики на изследвания продукт (програма, спецификация, проект, тестов сценарий, документ, и др.) или процес (технологичен, управленчески) могат да бъдат постигнати при систематично и добре организирано разработване, според изискванията на програма за осигуряване на качеството. Преподаването на разработената методика за оценяване на качеството на моделите в СТ пък дава възможност да се илюстрира как съвременния *model-driven* подход за разработване може да се използва в практиката.

От педагогическа гледна точка, най-приложими за обучението по СТ са резултатите, получени от използване на сравнителния анализ в извънкласната работа по информатика [1]. Идентифицираните основни и допълнителни обекти (обучаем, учебна програма, тема, задача, обучаващ, форма на проверка, учебно помагало) и създадените за тях модели могат сравнително лесно да се модифицират така, че да отразят особеностите на процеса на обучение и по СТ. Считаме, че при наличие на такива модели, всеки преподавател може да идентифицира *ситуации*, в които Сравнителният анализ би бил полезен.

Резултатите от проведените досега експерименти за включване на създадените модели на обекти в съдържанието на един основен (за бакалаври) и един специализиран (за магистри) курс по СТ са окуражаващи.



Конкретно приложение на метода ще илюстрираме с пример за сравнителен анализ на формите за обучение на софтуерните специалисти.

### 5.3. Пример

Сред добрите практики в областта на софтуерните технологии, разгледани в [2], има и такива за определяне на съдържанието и формите за обучение на софтуерните специалисти. За идентифицираните 17 форми на обучение е определено подреждане по четири критерия – цена на обучението чрез съответната форма, ефикасност, ефективност и актуалност на получаваната информация. Критериите са разгледани като равностойни и въз основа на оценките им се пресмята обобщена оценка, определяща поредното място на всяка форма в крайната класация. Резултатите от изследването са показани в следната таблица [2]:

Score	Form of education	Cost	Efficiency	Effectiveness	Currency
3.00	Web-browsing	1	1	9	1
3.25	Webinars / e-learning	3	2	6	2
3.50	Electronic books	4	3	3	4
5.25	In-house training	9	4	1	7
6.00	Self-study CD/DVD	4	3	7	10
7.25	Vendor training	13	6	5	5
7.25	Commercial training	14	5	4	6
7.50	Wiki sites	2	9	16	3
8.25	Live conferences	12	8	8	5
9.00	Simulation web sites	8	7	13	8
10.25	Self-study from books	5	13	12	11
10.25	Journals	7	11	14	9
10.75	On-the-job training	11	10	10	12
11.75	Mentoring	10	12	11	14
12.00	Books	6	14	15	13
12.25	Undergraduate training	15	15	3	16
12.25	Graduate training	16	16	2	15

Табл. 1. Подреждане на формите на обучение

Използвайки предложената в [2] методика, подготвихме провеждането на Сравнителния анализ за обект „форма на обучение“. При моделирането му се спряхме на прост линеен модел със същите четири характеристики на качеството, оценявани по скала от 1 (най-добра) до 10. Планирахме провеждане на СА с различни варианти на стандартната ситуация, като стойностите на следните елементи, определящи ситуацията, бяха постоянни:

- <цел> - оценяване
- <обект> - форма на обучение
- <задача> - ранжиране
- <ниво> - просто

Останалите два елемента, определящи ситуация, бяха с променливи стойности. За елемента <гледна точка> бяха разгледани варианти с две стойности: на организатора на обучението и на обучаем. За множество от <конкуренти> беше избирано подмножество на гореописаните 17 форми на обучение. В процедурата за оценяване на всеки обект се използваха различни теглови коефициенти за четирите характеристики на качеството, чрез което се постигаше настройване на провеждания анализ към контекста, определен от конкретно разглежданата ситуация.

Резултатите от проведените експерименти показаха как класацията на формите за обучение се изменя в зависимост от гледната точка на оценяващия и избраните от него конкуренти и тегла на критериите.

## 6. Заключение

Наред с поставянето на съществуващите проблеми в обучението по СТ и споделяне на идеи за разрешаването им от институции на различно ниво, чрез положителния ни опит от прилагането на Сравнителния анализ показахме, че всеки от преподавателите може да допринесе за по-качествен процес на обучение чрез премислени подобрения в съдържанието на курса и осъществяване на разумен избор в ситуации, свързани с организацията на преподаването.

## Литература

1. Христова, Пл., *Обектен подход за организиране на извънкласната работа по информатика за ученици до 5 клас*. Дисертация за получаване на образователната и научна степен “доктор” по научна специалност 05.07.03, Русе, 2010.
2. Jones, C., *Software Engineering Best Practices*, McGraw Hill, 2010.
3. Maneva, N., INSPIRE: A Software Engineering Methodology. *Journal of Information Technologies and Control*, № 1, 2005, pp.31-37.
4. Maneva, N., Comparative Analysis: A Feasible Software Engineering Method. *Serdica Journal of Computing*, vol.1, No 1, 2007, pp.1-12.
5. Pressman, R., *Software engineering – A Practitioner's Approach*, Sixth edition, McGraw Hill, 2006.