

## МОДЕЛ ЗА СЪЗДАВАНЕ НА ОБРАЗОВАТЕЛЕН СОФТУЕР, ПРЕДНАЗНАЧЕН ЗА ОБУЧЕНИЕ НА КОМПЮТЪРНИ СПЕЦИАЛИСТИ

Теменужка Зафирова-Малчева, Пенчо Михнев

**Анотация:** Докладът представя модел интегриращ дизайн на обучение в курса Проектиране, разработка и оценка на образователен софтуер (ПРООС) за студенти от бакалавърските програми Компютърни науки и Софтуерни технологии на Факултета по математика и информатика (ФМИ) на Софийски университет „Св. Климент Охридски“ (СУ). Той описва общия модел за Проектиране и разработка на образователен софтуер (ПРОС), който съчетава етапи и елементи на образователния дизайн и софтуерния дизайн за създаването на качествени образователни продукти. Този модел се прилага при обучението в изборният курс ПРООС във ФМИ.

**Ключови думи:** модел за дизайн на образователен софтуер, образователен софтуер, образователен дизайн, софтуерен дизайн.

## A MODEL FOR INTEGRATION OF INSTRUCTIONAL DESIGN IN A UNIVERSITY COURSE FOR EDUCATIONAL SOFTWARE DESIGN AND DEVELOPMENT, TAUGHT TO COMPUTER SCIENCE STUDENTS

Temenugka Zafirova-Malcheva, Pencho Mihnev

**Abstract:** The paper presents a model for integration of Instructional design in teaching Educational Software Design and Development (ESDD) university course for students in Computer Science BSc programmes at the Faculty of Mathematics and Informatics (FMI) of Sofia University “St. Kliment Ohridski”. It describes the general model for ESDD that combine Instructional Design (ID) and Software Design (SD) stages and components to produce a qualitative educational software products. The model is used in the elective course “Design, Development and Evaluation of Educational Software” at FMI.

**Key words:** software design model, educational software, instructional design, software design.

### Въведение

Създаването на един софтуерен продукт е дълъг процес, преминаващ през различни фази от спецификацията на продукта през проектирането, реализацията, тестването до внедряването му в практически условия, както и последващото му модифициране, усъвършенстване и поддръжка. Успешното протичане на този процес изисква систематичен подход и е свързан със следване на определени утвърдени фази при разработката на софтуера. Роджър Пресман описва софтуерния процес с пет общи рамкови дейности – комуникация, планиране, моделиране, конструиране и внедряване (Pressman 2010). Съществуват множество различни модели, които описват последователността и начина на изпълнение на тези дейности и връзките между тях. В зависимост от типа, предназначението, обхвата и сложността на разработвания

софтуер може да бъде следван един или друг модел, определящ начина и последователността на протичане на тези фази.

Говорейки за образователен софтуер и вземайки под внимание, че по същество това е софтуер, който предоставя конкретно образователно съдържание и при разработката му са заложили специфични образователни цели (Zafirova-Malcheva 2014), то създаването на такъв тип софтуер включва елементи, които не са характерни за разработването на софтуер като цяло. Тези елементи включват на първо място дефиниране на обучителен проблем и образователни цели, които софтуерът ще решава, както и подбор на подходящо учебно съдържание. Това са елементи, които всъщност са характерни за дизайна на обучение, което води до извода, че проектирането и разработката на образователен софтуер изисква познания не само в областта на софтуерното инженерство, но и в областта на дизайна на обучение. Софтуерното инженерство предоставя методи и инструменти за изграждане на софтуерни продукти, но само по себе си това не е достатъчно за качествения дизайн и разработка на образователен софтуер. В отделните фази на разработката възниква нужда от знания, свързани с дизайна на обучение. Характерът на образователния софтуер предполага, че той има специфично образователно съдържание и е насочен към постигането на определени образователни цели. Присъствието на образователни елементи изисква при разработката на такъв тип софтуер да бъдат включени елементи от дизайна на обучение.

Съчетаването на тези две основни области на знания, от една страна, е основа за създаването на качествен образователен софтуер, а от друга – показва нуждата от систематичен подход, който ефективно да комбинира знания, методи и средства от тези области през отделните етапи от дизайна и разработката на образователен софтуер. Тази връзка е отразена при създаването на общия модел за проектиране и разработка на образователен софтуер (ПРОС), представен в този доклад.

Общият модел за ПРОС е разработен за целите на обучението във Факултета по математика и информатика на СУ „Св. Климент Охридски“. Моделът комбинира елементи от софтуерния дизайн с елементи от дизайна на обучение, като се базира на доказани модели от тези две области. Той се използва в изборната дисциплина „Проектиране, разработка и оценяване на образователен софтуер“ при обучението на студенти от специалностите Компютърни науки, Софтуерно инженерство, Информатика и Информационни системи в бакалавърските програми, които нямат познания в областта на образователния дизайн, съществени при разработката на качествен образователен софтуер.

### **Общ модел за ПРОС**

Идеята за вграждане на стратегии за учене и обучение при проектирането и разработката на образователен софтуер не е нова. В своето изследване „Pedagogy Embedded in Educational Software Design“, Иностроса и Мейар (Hinostraza & Mellar 2001) описват процеса на проектиране на образователен софтуер, базиран на експертно

участие на учители. Тяхното изследване обаче е фокусирано върху проектиране на софтуер на микро ниво. То се отнася до конструкцията, елементите и връзките между тях, свързани с представяне на учебното съдържание в конкретно софтуерно приложение за обучение. Това обаче е една съвсем малка част от дизайна на образователен софтуер и не е свързана с прилагането на систематичен подход, описващ целия процес от възникването на идеята до внедряването на готовия продукт.

Общият модел за ПРОС се базира на няколко основни модела за дизайн на софтуер, които с леки вариации следват общата рамка за разработка на софтуер, описана от Пресман (Pressman, 2010). Ясното структуриране и разграничаване на етапите в общия модел за ПРОС е заимствано от каскадния модел (Winston 1970). Но противно на него моделът за ПРОС позволява паралелно протичане на процесите, както в инкременталните модели (Pressman, 2010). Техниката за прототипиране също е интегрирана в новия модел с цел получаването на обратна връзка на възможно най-ранен етап от проектирането на софтуера и възможността за лесен редизайн (Pressman, 2010). Моделът за ПРОС комбинира изброените техники и методи от различни софтуерни модели, така че, когато ги съчетаем с методите на дизайн на обучение, да се получи хармонично работещ модел (Mihnev & Zafirova-Malcheva 2015).

Ключов елемент при дизайна на обучение, на който пада основният акцент при различните модели, е дефинирането на целите на обучението, като анализът на задачата допринася най-много за идентифицирането на тези цели. В своята книга “Task Analysis Methods for Instructional Design” (Jonassen et al. 1999) Йонасен, Тесмер и Ханъм определят фазата на анализ на задачата като „вероятно най-важната част от процеса на системния дизайн на обучение” (Jonassen et al. 1999) **Error! Reference source not found.** В своя модел за дизайн на обучение Смит и Раган (Smith & Ragan 2006) **Error! Reference source not found.** отделят особено внимание на този елемент от фазата на анализа, като разделят анализа на задачата на шест отделни стъпки. Първо се определят общите образователни цели. След това за всяка обща цел се прави анализ на информационните процеси, като се дефинират всички предварителни знания и умения, необходими за постигането на съответната цел. На базата на тези предварителни знания и умения за всяка обща цел се дефинират няколко конкретни образователни цели. Ето защо общият моделът за ПРОС се базира на модела на Smith & Ragan относно анализа на задачата и съответно дефинирането на общите и конкретните образователни цели на софтуера.

Базирайки се на общата рамка, следвана от повечето модели за дизайн на софтуер, моделът за ПРОС е разделен на няколко основни етапа, които включват: Анализ, Проектиране, Разработка, Тестване и оценка и Внедряване. Освен това този модел трябва да отчита характерните особености на крайния продукт – образователен софтуер. Ето защо един от основните елементи при анализа трябва да е насочен към анализ на задачата с цел определяне на образователните цели на софтуера, а това е ключов елемент от дизайна на обучение. Дефинирането на образователните цели се базира на съществуващ обучителен проблем, който пък възниква на базата на нужда от

определен тип обучение от страна на определена група от обучаеми. Т.е. анализът на задачата е строго обвързан с анализа на обучаемите и анализа на техните потребности.

Така, следвайки основните стъпки от системния дизайн на обучение и тези при проектирането на софтуер при общия модел за ПРОС, се обособяват две основни фази – Анализ и Проектиране и разработка, като всяка от тях се състои от няколко етапа:

- **Фаза на Анализ** – базира се на моделите за дизайн на обучение, включва три основни етапа: Анализ на потребностите, Анализ на аудиторията, Анализ на задачата;
- **Фаза на Проектиране и разработка** – базира се на моделите за проектиране на софтуер, включва четири основни етапа: Проектиране, Разработка, Тестване и оценяване, Внедряване и съпровождане.

Базирайки се на съвременните тенденции както при дизайна на софтуер, така и при дизайна на обучение, при този модел се извършва оценяване през целия процес на проектиране и разработка на софтуера под формата на обратна връзка, а в етапа на тестването и оценяването се прави общо крайно оценяване. В зависимост от резултатите от това оценяване в края на този етап може да се премине към внедряване на софтуера или да се върнем към някой от предходните етапи за редизайн и подобрения. Освен това моделът е цикличен и при необходимост дава възможност на всеки етап от процеса да се върнем към някой от предишните етапи. Разработката на крайния продукт е свързана със създаване на негов прототип, който в рамките на няколко итерации се обогатява и доразвива, до получаване на напълно функциониращ краен продукт.

На Фигура 1 е представена схема на общия модел за ПРОС, която илюстрира двете основни фази и техните етапи. Показани са дейностите, които включва всеки етап, както и връзките между тях и преходите между отделните етапи.



Фигура 1. Общ модел за ПРОС

## Общ модел за ПРОС етапи

Първият етап е свързан с анализа на потребностите от определен тип обучение и формулиране на обучителен проблем (Фигура 2), който може да бъде решен с разработката на специфичен софтуерен продукт. Проблемът трябва да бъде формулиран много внимателно, като трябва да може да се решава с помощта на някакъв тип обучение, което да позволява ефективното използване на компютри и софтуер.

Вторият етап е свързан с анализа на характеристиките на конкретна група от потребители и техните нужди (Фигура 2). Реално първите два етапа протичат паралелно, а резултатът от тях е формална дефиниция на обучителния проблем по отношение на необходимите знания, умения и нагласи (а в някои случаи – компетенции), които софтуерът трябва да решава (Mihnev & Zafirova-Malcheva 2015).



Фигура 2. Първи етап: Анализ на потребностите и Втори етап: Анализ на аудиторията

Финалният етап от фазата на анализа е свързан с анализ на задачата (Фигура 3). В този етап, изхождайки от формулирания обучителен проблем, се дефинират общите образователни цели на софтуера. За всяка обща цел чрез анализ на информационните процеси се определят предварителните знания и умения, необходими за нейното изпълнение, и се дефинират съответните конкретни образователни цели.

Анализът на задачата трябва да реализира резултати, адекватни на конкретните образователни цели, като се дефинира:

- учебното съдържание, което софтуерът ще предоставя;
- различните учебни дейности и задачи, чрез които ще се отработва съдържанието, техният тип, обем и степен на сложност;
- софтуерни изисквания.

Софтуерните изисквания, идентифицирани на този етап, са свързани с общите

настройки и характеристики, които софтуерът трябва да поддържа, и зависят от нуждите на потребителите. Например възможност за контролиране на нивото на звука; начин на структуриране и представяне на учебното съдържание; начин на контролиране на отделните компоненти, предоставяне на помощна информация и др. Тук трябва да бъдат взети под внимание различните изисквания, свързани с добрите практики при дизайн на интерфейса. Например възможност за връщане към главното меню на всяка стъпка от програмата; възможност за контрол на нивото на сложност и т.н. Тези изисквания са свързани със стандартите за софтуерен дизайн и са общовалидни независимо от характеристиките на потребителската група и обучителния проблем, който софтуерът решава.



Фигура 3. Трети етап: Анализ на задачата

След края на третия етап се преминава към втората фаза от модела, която е свързана с проектирането и разработката.

При проектирането на базата на резултатите от анализа се определя типът на образователния софтуер, който ще се разработва, и се създава неговият сценарий. Дефинират се дейностите и задачите, които потребителят ще изпълнява. На базата на дейностите, сценария и описаните по време на анализа на задачата софтуерни изисквания се дефинират функционалностите, които софтуерът трябва да поддържа. В резултат на този етап трябва да се създаде дизайнът на интерфейса на софтуера, в който са отразени неговите функционалности (Фигура 4).



Фигура 4. Четвърти етап: Проектиране

При работа със студентите на този етап се прилага техниката на сторибординг, чрез която се създава графичен модел на софтуера. Неформално той представлява нефункциониращ прототип на софтуера, който предоставя, от една страна, достъпен за потребителите начин да се запознаят предварително със системата и от друга, възможност разработчиците да получат обратна връзка още преди да е започнал етапът на кодирането (UsabilityNet 2006). Сторибордингът позволява също да се определят и опишат типът и мястото на елементите за всеки екран от програмата, какви ще са техните функционалности и връзките им с други елементи. Да се определи и опише как използването на даден елемент ще се отрази на поведението на програмата. Да се опишат действията, които потребителят ще може да изпълнява. Да се представи схемата за навигация в програмата и да се провери дали представянето е точно и пълно.

Следващият етап е Разработката е свързан с програмната реализация на продукта (Фигура 5). При нея на няколко итерации се създава прототип на софтуерния продукт, като след всяка итерация се допуска връщане към предходните етапи, където могат да се модифицират старите, да се добавят нови функционалности, или да се направят други промени. Това продължава до създаването на прототип, отговарящ на изискванията на крайния потребител. Преди да се премине към следващия етап, прототипът преминава през тестване и отстраняване на технически грешки. Този процес е цикличен и продължава докато софтуерът започне да работи коректно при определени, предварително дефинирани условия.

Последният етап от фазата на Проектирането и реализацията е свързан с тестването и оценяването на готовия прототип (Фигура 5). То е свързано, от една страна, с тестване на функционалностите на софтуера – дали отговарят на заложените при проектирането, и от друга, с експертно тестване и тестване с крайните потребители – дали разработеният софтуер отговаря на нуждите на потребителите.



Фигура 5. Пети етап: Разработка и Шести етап: Тестване и оценяване

След анализ на резултатите от тестването са прави оценка на софтуера, в зависимост от която може да се наложи връщане към някоя от предходните фази за отстраняване на възникнали проблеми или пускане на софтуера в експлоатация (внедряване).

След внедряването на софтуера в процеса на неговата експлоатация може да се наложат действия по неговото съпровождане (поддръжка). Ето защо моделът позволява дори и след внедряването да се върнем към някой от предходните етапи от разработката на софтуера. Това прави моделът изключително гъвкав, но въпреки това трябва да се прави предварителна оценка на рисковете и да не се допуска стигане до редизайн на продукта, налагаш връщане от заключителните в началните етапи от проектирането. Това предполага много внимателен и задълбочен анализ през първата фаза, какъвто моделът до голяма степен гарантира. Така, ако се наложат промени, те ще бъдат по-скоро във фазата на проектирането и разработката. Като създаването на неформален прототип чрез сторибординг и получаването на ранна обратна връзка до голяма степен намалява риска от големи пропуски.

## Приложение

Представеният общ модел за ПРОС се прилага успешно вече трета година при обучението на студенти в изборния курс „Проектиране, разработка и оценяване на образователен софтуер“ във ФМИ. Курсът е с продължителност един семестър и хорариум 30 часа лекции и 30 часа упражнения. Той се провежда в смесена форма на обучение, която освен присъствените занятия, включва и електронен курс. Той не само предоставя всички учебни материали, но се използва и за реализиране на разнообразните дейности в курса.

Успешното завършване на курса е свързано с разработването на работещ прототип на образователен софтуер. Курсът е структуриран, така че текущите задания, формиращи частично крайната оценка от курса, да са пряко свързани с реализацията на финалния проект. Тези задания следват отделните етапи от общия модел за ПРОС и представляват базови стъпки от разработката на финалния проект. Освен това курсът



залага на екипна работа, което, от една страна, развива важни социални умения у студентите, а от друга, е основен елемент от практиката при разработване на софтуер.

През последните две години курсът е завършен успешно от 64 студенти, които са разработили 20 добре функциониращи прототипа на образователен софтуер.

## Заклучение

Прилагането на систематичен подход при проектирането и разработката на образователен софтуер изисква съчетаване на методи и средства както от образователния, така и от софтуерния дизайн. Общият модел за ПРОС успешно съчетава тези елементи, като дава възможност на студентите да преминат през целия процес на разработка на образователния софтуер и да произведат работещ краен продукт. Това дава възможност по време на различните фази от разработката студентите да влизат в различни роли – на дизайнери, разработчици, тестери, оценители и т.н., което до голяма степен би било от полза при избора на направлението, в което да се развиват в бъдеще.

Вземайки предвид резултатите от приложението на общият модел за ПРОС, може да се твърди, че той успешно реализира своите цели и отговаря на нуждите на студентите, като им предоставя специфични познания в областта на образователния дизайн, които не са характерни за тяхното обучение, но са от съществено значение при разработката на образователен софтуер.

## Литература

- Hinostroza & Mellar 2001:** Hinostroza, E., Mellar, H., Pedagogy Embedded in Educational Software Design. Report of a Case Study // *Computers & Education*, 37 (1): 27–40, August 2001, [http:// Available from: <www.sciencedirect.com/science/article/pii/S036013150100032X>](http://www.sciencedirect.com/science/article/pii/S036013150100032X), (7 July 2015).
- Jonassen et al. 1999:** Jonassen, D. H., Tessmer, M., Hannum, W. H. Task Analysis Methods for Instructional Design. Lawrence Erlbaum Associates, Inc. Mahwah, New Jersey, 1999.
- Mihnev & Zafirova-Malcheva 2015:** Mihnev, P., Zafirova-Malcheva, T. A general model for educational software design and development // in Proceedings of EDULEARN15, *7th International Conference on Education and New Learning Technologies*, Barcelona, Spain, 6th - 8th of July 2015.
- Pressman 2010:** Pressman, R. Software engineering: a practitioner's approach. 7th ed., McGraw-Hill, 2010, ISBN 978-0-07-337597-7.
- Smith & Ragan 2006:** Smith, P. L., & Ragan, T. J. Instructional Design, 3rd Edition // Hoboken, NJ: John Wiley & Sons, Inc. 2005.
- UsabilityNet 2006:** A European Union project that provides usability and user centered design, Available from: <http://www.usabilitynet.org/home.htm>, (19 May 2015).
- Winston 1970:** Winston, R. Managing the Development of Large Software Systems, Proceedings of IEEE WESCON 26 (August), 1970.
- Zafirova-Malcheva 2014:** Zafirova-Malcheva, T. Step by step in design of educational software for children with special educational needs // in Proceedings of iCERi2014, *7th International Conference of Education, Research and Innovation*, 17th-19th November 2014, Seville, Spain. (pp. 1080-1089). ISBN: 978-84-617-2484-0, ISSN: 2340-1095, Depósito Legal: V-2632-2014.

**За авторите:** Ас. д-р Теменужка Зафорова-Малчева, Катедра „Информационни технологии“, Софийски университет „Св. Климент Охридски“, тел. 028161507, e-mail: [tzafirova@fmi.uni-sofia.bg](mailto:tzafirova@fmi.uni-sofia.bg)  
Ас. Пенчо Михнев, Катедра „Информационни технологии“, тел. 028161507, Софийски университет „Св.

Климент Охридски“, e-mail: pmihnev@fmi.uni-sofia.bg