

**ПЛОВДИВСКИ УНИВЕРСИТЕТ „ПАИСИЙ ХИЛЕНДАРСКИ”
ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА
КАТЕДРА „ОБУЧЕНИЕ ПО МАТЕМАТИКА, ИНФОРМАТИКА И
ИНФОРМАЦИОННИ ТЕХНОЛОГИИ”**

ХРИСТО ТОШКОВ ХРИСТОВ

**МЕТОДИКА НА ПРЕПОДАВАНЕ
НА СЪВРЕМЕННИ ТЕХНОЛОГИИ
ЗА СЪЗДАВАНЕ НА СОФТУЕР**

АВТОРЕФЕРАТ

**на дисертационен труд
за присъждане на образователната и научна степен "доктор"
в област на висше образование: 1. Педагогически науки професионално
направление: 1.3. Педагогика на обучението по... докторска програма: Методика
на обучението по информатика и информационни технологии**

Научен ръководител: доц. д-р Христо Крушков

Пловдив, 2016 г.

Дисертационният труд е обсъден и насочен за защита на разширено заседание на катедра „Обучение по математика, информатика и информационни технологии” при Факултета по математика и информатика на ПУ „Паисий Хилендарски“.

Дисертационният труд се състои от 229 страници, от които 199 основен текст. Използваната библиография включва 165 източника, от които 84 заглавия на български, 2 на руски, 79 заглавия на английски език, като 12 от тях са официални интернет страници на висши учебни заведения, 20 са официални интернет страници на научни организации за работа и поддръжка на софтуерни технологии, а 5 са специализирани интернет справочници.

Оформени са 3 приложения: Приложение №1: Концепция за провеждане на активно, участващо наблюдение – тип интервю; Приложение №2: Въпросник за съставяне на анкетни карти; Приложение №3: Концепция за провеждане на педагогически експеримент за работа в екип чрез прилагане на метода „групова дискусия“ (фокус метод).

Приложени са списък с авторски публикации по дисертационния труд, списък с техните цитирания, списък с изследвания изготвени в изпълнение на задачи по научни проекти, списък с изследвания докладвани на научни сесии и конференции.

Защитата на дисертационния труд ще се състои на 21 юни от часа в Заседателната зала на новата сграда на ПУ „Паисий Хилендарски“, гр. Пловдив.

Материалите по защитата са на разположение на интересуващите се в секретариата на ФМИ, нова сграда на ПУ, каб. 330, всеки работен ден от 8:30 до 17:00 часа.

Автор: Христо Тошков Христов

Заглавие: „Методика на преподаване на съвременни технологии за създаване на софтуер“.

Университетско издателство „Паисий Хилендарски“
Пловдив, 2016 г.

СЪДЪРЖАНИЕ

Обща характеристика на дисертационния труд	4
Актуалност на проблема	4
Цели и задачи на дисертационния труд	5
Структура и обем на дисертационния труд	6
Кратко съдържание на дисертационния труд	7
Глава 1. Състояние на образованието и обучението в професионално направление „Информатика и компютърни науки“	7
1.1. Провеждане на качествено педагогическо изследване	7
1.2. <i>Статут и значение</i> на съвременното образование по софтуерни технологии	9
1.3. Състояние на обучението по информатика и софтуерни технологии	10
1.4. Тенденции и решения при преподаването на технологии за създаване на софтуер	12
1.5. Резултати от първа глава	13
1.6. Някои тези и изводи от първа глава	13
Глава 2. Разработване на <i>Методика на преподаване на съвременни технологии за създаване на софтуер</i>	14
2.1. Концепции и теоретични основи за разработване на <i>методиката</i>	14
2.2. Изграждане на <i>Общ йерархичен интеграционен модел</i>	16
2.3. Схема на <i>Общ йерархичен интеграционен модел</i>	22
2.4. Резултати от втора глава	23
2.5. Някои тези и изводи от втора глава	23
Глава 3. Внедряване на <i>Методика на преподаване на съвременни технологии за създаване на софтуер в процеса на обучение</i>	24
3.1. Подход на интегриране на <i>методиката</i>	25
3.2. Методи и експерименти	26
Заклучение	27
Резултати – научни приноси и постижения	27
Класификационна матрица на приносите	27
Сложност и трудности на изследванията	29
Приложение и перспективи	30
Апробация	30
Публикации по темата на дисертационния труд	31
Благодарности	31
Библиография	31

Актуалност на проблема

Разработването на *Методика на преподаване на съвременни технологии за създаване на софтуер*, която да е съобразена с концептуалните и технологични тенденции в софтуерното инженерство и нуждата от специалисти за развитие на информационното общество (ИО), е един актуален научен проблем. Актуалността на проблема се определя от нуждите на осъвременяване и синхронизация на образователната система на висшето образование с тенденциите на развитие в софтуерната индустрия.

Дисертационният труд е посветен на разработването на методика на преподаване предназначена за студентите изучаващи засилено програмиране; моделиране, създаване и приложение на софтуерни процеси; управление и реализация на софтуерни проекти. В работата са изследвани както **структурно-съдържателната страна на обучението**, като връзка *теории на съвременни технологии – тематично учебно съдържание*, така и процесуално-дейностната, като приложение на конкретни методи и средства приложени за определен вид учебно съдържание. По конкретно **научният проблем**, който изследваме, е: Реализиране (разработване и приложение) на *Методика на преподаване на съвременни технологии за създаване на софтуер*, базирана на концепцията за **внедряване на теории и практики от предметната област софтуерно инженерство¹ в учебните планове и програми на специалности „Информатика...“² и „Софтуерни технологии и...“³, посредством общ йерархичен интеграционен модел за структуриране, разпределяне, формализиране, адаптиране и актуализиране на тематично учебно съдържание, методи и средства на обучение.**

В педагогическата литература се разглеждат няколко класификации на научно-изследователските проблеми. Според Х. Саймън съществуват три типа проблеми: добре структурирани (количествено формулирани); неструктурирани (качествено изразени) и слабо структурирани (смесени) – съдържащи и количествени и качествени елементи. Според Р. Акоф проблемите могат да бъдат: по изразение – качествени и количествени; по обхват – частни и общи; и по специфика: оценъчни и на развитието[15]. В нашия случай научният проблем следва да се разглежда като **неструктуриран**, т.е. изразен е *качествено*, по обхват като разработка е **общ**, но и с конкретно частно приложение, а по специфика е **на развитието**.

За изследвания научен проблем съставянето на общ йерархичен интеграционен модел, разгръщането на образователна парадигма на обучението по софтуерни технологии, декомпозицията на сложността и намаляването на абстракцията на предвиденото за изучаване тематичното учебно съдържание, адаптирането на методи, подходи и средства за преподаване на заложеното учебно съдържание и др. основни въпроси, са свързани с разрешаването на множество трудности, които най-общо могат да се разделят на: **научно-теоретични** за областите на методиката на обучение и софтуерното инженерство; **нормативни**, касаещи законова уредба, университетските правилници, регламенти, квалификационни характеристики, учебни планове и програми и др.; и **приложни**, свързани с внедряването на теории и практики от софтуерното инженерство в учебните планове и програми на специалности „Информатика...“ и „Софтуерни технологии и...“. В този смисъл, научният проблем, предмет и обект на изследване могат да се разглеждат в научно-теоретичен, нормативен и приложен аспект.

Забележка¹: В световното образователно и научно пространство понятията *софтуерни технологии* и *софтуерно инженерство* често се използват като синонимни. Отнесено за страната, изследването на българската научна софтуерно-специализирана литература разкрива, че понятието *софтуерни технологии* най-често носи смисъл на *дисциплина, специалност* и *специализирана технология*, докато неговия синонимен еквивалент *софтуерно инженерство* обикновено се използва със специализирано значение в софтуерното производствено и индустрия. В настоящия труд използваме понятието *софтуерни технологии* със значение на дисциплина, специалност или специализирана технология, а *софтуерно инженерство* употребяваме в смисъл на научно направление и индустриална област.

Забележка²: Формулировката на понятието за специалност „Информатика...“ е обобщаваща за водещите бакалавърски специалности в страната, редовна форма на обучение, чиито названия варират в различните университети.

Забележка³: Формулировката на понятието за специалност „Софтуерни технологии и...“ е обобщаваща за бакалавърските специалности в областта на софтуерното инженерство, редовна форма на обучение, чиито названия варират в различните университети.

Обект на изследването е обучението на студентите от специалности „Информатика...“ и „Софтуерни технологии и ...“; бакалавър, редовна форма, като софтуерни архитекти, инженери и разработчици.

Предмет на изследването е влиянието на разработената методика за преподаване на съвременни технологии за създаване на софтуер за повишаване качеството на обучение и неговия потенциал на студентите от специалности „Информатика...“ и „Софтуерни технологии и...“; бакалавър, редовна форма.

Цели и задачи на дисертационния труд

Основната цел на дисертационния труд е: Реализирането (разработване и приложение) на *Методика на преподаване на съвременни технологии за създаване на софтуер*, чрез която да се повиши и усъвършенства качеството и ефективността на обучението по софтуерни технологии.

Въз основа на поставената цел и за нейното постигане определихме следните задачи и подзадачи:

Задача 1: Да се изгради постановка на научния проблем.

- 1) Да се определи научен проблем, обект, предмет, цел, задачи, хипотеза, изследователски методи и критерии за оценка на резултати;
- 2) Да се направи очерк, опише обхвата и ограниченията на научния проблем.

Задача 2: Чрез анализ на научно-теоретичната и професионално-приложна специализирана литература да се проучи развитието на методиките на преподаване на технологии за създаване на софтуер.

- 1) Да се изяснят теоретичните аспекти на основни за изследването понятия;
- 2) Да се проследи произходът, да се анализира развитието, да се оцени състоянието и да се направи обзор на методически въпроси за обучението по програмиране; по моделиране, създаване и прилагане на софтуерни процеси; по управление и реализация на софтуерни проекти за водещите специалности на направление „Информатика и компютърни науки“;
- 3) Да се анализира необходимостта от синхронизация между теориите и практиките при съвременните технологии за създаване на софтуер и тематичното учебно съдържание в плановите и програмите на специалностите „Информатика...“ и „Софтуерни технологии и ...“.

Задача 3: Да се разработи: *Методика на преподаване на съвременни технологии за създаване на софтуер*.

- 1) Да се изгради концепция, да се поставят теоретични основи на методиката и да се изведат особените положения при методите на провеждане на изследването специфични за дисертационния труд;
- 2) Да се анализират тенденциите и да се разкрият перспективите при: програмирането; традиционните софтуерни модели; популярните съвременни гъвкави методологии за моделиране, създаване и прилагане на софтуерни процеси; стандартите за управление и реализация на софтуерни проекти;
- 3) Да се построи общ йерархичен интеграционен модел.

Задача 4: Да се разработи концепция за внедряване на методиката.

- 1) Да се проведат методически и педагогически експерименти по методите на т. нар. активно, участващо наблюдение – тип интервю; фокус метод и анкетна карта;
- 2) Да се разработят подходи за структуриране, разпределяне, формализиране, адаптиране и актуализиране на тематично учебно съдържание в общия йерархичен интеграционен модел;
- 3) Да са разработят критерии и показатели за оценяване на постигнатите резултати;
- 4) Да се изведат насоки за усъвършенстване и да се представят перспективи за развитие и приложение на методиката.

Хипотеза на изследването: *Чрез предложената „Методика на преподаване на съвременни технологии за създаване на софтуер“ може да се повиши качеството на обучение и неговия потенциал, като се доближи и синхронизира тематичното учебно съдържание с теориите и практиките на съвременните софтуерни технологии.*

Методологическата роля на хипотезата се проявява в това, че тя е свързващо звено между теоретичната концепция и методиката на изследването[15]. При така определената хипотеза особено внимание заслужава формулировката „повишаване на потенциала на качество на обучение“. Тук под „повишаване на потенциал“ се има предвид, че поради динамичните изменения на съвременните технологии и тяхното относително кратко във времето съществуване **планираната закономерна съвременна актуализация и синхронизация** на *тематичното учебно съдържание с теориите на технологиите за създаване на софтуер* увеличава потенциала на качеството на обучение, в смисъл, че процесът на обучение е по-приближен до професионалните **актуални** (в крак с времето) практики на софтуерно производство.

За реализиране на основната цел и разрешаването на задачите на дисертационния труд са приложени следните **методи за провеждане на качествено изследване** в образователна област *софтуерни технологии* и индустриална област *софтуерно инженерство*:

- **анализ на съдържанието** на софтуерна, педагогическа и методическа литература;
- **анализ на съдържанието** на нормативни документи в това число ЗВО, правилници на ВУЗ, квалификационни характеристики, учебни планове и учебни програми, наредби и др.
- **активно участващо наблюдение – тип интервю, беседа и групово дискусия** за извличане, структуриране, актуализиране и адаптиране на тематично учебно съдържание, подходи, методи и средства на обучение;
- **анкетирание** на студенти за получаване на обратна връзка за качеството на обучението;
- Провеждане на **методически и педагогически експерименти** с изследователи, разработчици, преподаватели и студенти за анализ и окачествяване на процеса на обучение;
- Позоваване на **личен опит** в качеството на софтуерен разработчик и университетски преподавател;

В труда се изследват теории и практики на съвременните технологии за създаване на софтуер. Те се разглеждат като източник на актуално тематично учебно съдържание и като такива се изследват от гледна точка на педагогическата наука. Затова говорим за провеждане на педагогическо изследване над теориите на технологиите за производство на софтуер в софтуерното инженерство и индустрия, приложени в обучението на студенти подготвяни за софтуерни архитекти, инженери и разработчици.

Обхватът на научния проблем е определен от границите на: теориите при съвременните технологии за създаване на софтуер; тематичното учебно съдържание, методите и средствата на обучение по софтуерни технологии; нормативната уредба на специалности „Информатика...“ и „Софтуерни технологии и...“ и провеждането на обучение в тези специалности. Сечението на тези области, е **общата научно-теоретична рамка** на изследването.

Основна характеристика на научния проблем, е неговото степенуване по сложност на независими едно от друго нива: **ниво първо** – знания, умения и компетенции при стандартите за управление и реализация на софтуерни проекти; **ниво второ** – знания, умения и компетенции за приложение на традиционни модели и гъвкави методологии; **ниво трето** – знания, умения и компетенции за подходи, стилове, техники и практики на обектно ориентирано програмиране. Такова йерархично условно разграничение на обучението разкрива три нива (вида) подготовка: на *специалиста програмист*, на *специалиста моделиращ, създаващ и прилагащ софтуерен процес* и на *специалиста ръководител на софтуерен проект*.

Структура и обем на дисертационния труд

Дисертационният труд е съставен от увод, 3 глави, заключение, библиографски източници и приложения.

В първа глава *Състояние на образованието и обучението в професионално направление „Информатика и компютърни науки“* е разработена постановката на дисертационния труд. Специално внимание е отделено на сложността и срещнатите трудности при протичането на изследванията, тъй като в направление „Информатика и компютърни науки“ за област *Методика на*

обучението по софтуерни технологии не са забелязани разработки на проведени качествени педагогически изследвания.

Във втора глава *Разработване на Методика на преподаване на съвременни технологии за създаване на софтуер* е разработена методиката като са анализирани общоприети и утвърдени в софтуерната индустрия и образователна практика стандарти, методологии, парадигми, стилове, подходи и пр. техники за производство на софтуер и практики на тяхното преподаване. Също така, съставени са *концепция за провеждане на качествено педагогическо изследване в областта на методиката на обучението по софтуерни технологии* и *концепция за реализиране на Методика на преподаване на съвременни технологии за създаване на софтуер чрез провеждане на качествено педагогическо изследване*. Чрез прилагането на двете концепции е проведено мащабно качествено изследване, в следствие на което е построен *общ йерархичен интеграционен модел за структуриране, разпределяне, формализиране, адаптиране и актуализиране на тематично учебно съдържание, методи и средства на обучението*. Концептуалният модел представя основи на теория на методиката на обучението по софтуерни технологии.

В трета глава *Внедряване на Методика на преподаване на съвременни технологии за създаване на софтуер в процеса на обучение* се разглежда въпросът на нейното интегриране в обучението. За тази цел е разработен подход на интеграция, като са разгледани проблемите за предпоставките и условията за приложение на подхода; ролята на метода активно, участващо наблюдение – тип интервю за интеграцията на методиката; окачествяването на процеса на обучение; проверка за достоверност на използваните методи и истинност на получените резултатите и др.

В *Заключението* са изведени същностите характеристики на изследването и научния проблем. Посочени са получените резултатите под формата на приноси и постижения чрез *класификационна матрица, списък на научните приноси и разширен списък на приносите групирани по постижения*. Също така, представени са перспективите на развитие пред *Методика на преподаване на съвременни технологии за създаване на софтуер*.

Дисертационният труд се състои от 229 страници, от които 199 основен текст. Използваната библиография включва 165 източника, от които 84 заглавия на български, 2 на руски, 79 заглавия на английски език, като 12 от тях са официални интернет страници на висши учебни заведения, 20 са официални интернет страници на научни организации за работа и поддръжка на софтуерни технологии, а 5 са специализирани интернет справочници.

Списъкът на авторските публикации се състои от 6 заглавия.

КРАТКО СЪДЪРЖАНИЕ НА ДИСЕРТАЦИОННИЯ ТРУД

Глава 1. Състояние на образованието и обучението в професионално направление

„Информатика и компютърни науки“.

1.1. Провеждане на качествено педагогическо изследване.

По същност решението на научния проблем представлява *преобразуване, класифициране и внедряване* на общоприети и утвърдени принципи, стандарти, подходи, теории и практики за създаване и разработване на софтуер, които, „извлечени“ от професионалното им приложение в софтуерното инженерство и индустрия и „преведени“ в образователното пространство, придобиват характер на ***теория на методика на обучение по софтуерни технологии***.

Методологията, която използваме и чрез която разрешаваме научния проблем е приложима за т. нар. **качествени изследвания**. Качествените изследвания са вид социални изследвания, към които спадат и качествените педагогически изследвания. За пионери на тези изследвания се приемат Ф. Бос и Б. Малиновски, които в началото на ХХ век провеждат първите културно-антропологически и етнологически изследвания, а представители на т. нар. Чикагска школа в социологията през 1920 г. изследват градската субкултура. По-съществени тласъци на развитие на тези изследвания се свързват с имената на П. Лазарсфелт и П. Мертон, които през 50-те години първи разработват т. нар. фокусирано интервю; К. Попър и Т. Адорно, които през 60-те развиват тематиката за „Логиката на социалните науки“; Б. Глейзър и А. Щраус, които през същия период разработват и популяризират своята „Обоснована теория“, още наричана „Теория, основана на данни“ [3].

За проведеното в настоящия труд качествено педагогическо изследване основна роля и влияние играе „Обоснована теория“ на Глейзър и Щраус.

Качествените изследвания се характеризират с проследяването на живота или дейността на определена категория хора. В нашия случай, за област *Методика на обучението по софтуерни технологии*, изследователски интерес представляват дейностите **разработване, преподаване, изследване и изучаване** на софтуерни технологии. Затова, макар и обектът на изследване да е обучението на студента, се изследват *софтуерни архитекти, инженери, разработчици, университетски преподаватели, изследователи на софтуер и самите студенти*.

Счита се, че „Обоснована теория“ на Глейзър и Щраус, е една от най-солидните теории за провеждане на качествени изследвания. В нея авторите правят опит да дадат отговор на въпроса: „Как чрез качествените изследвания могат да се създават теории в социалните науки?“. Според тях „Теория, основана на данни“ представлява качествен изследователски подход, систематичните техники и процедури на който дават възможност на изследователя да разработи теория, отнасяща се към една определена предметна област[3]. В настоящия труд е направен опит за поставяне на основи на теория за методиката на обучението по софтуерни технологии, наречена: *Методика на преподаване на съвременни технологии за създаване на софтуер*.

Според Глейзър и Щраус, обоснована теория е тази, която се извежда индуктивно чрез изучаването на даден феномен. Тя се създава, развива и верифицира чрез систематично натрупване и анализиране на данни, които се отнасят към проблема, т.е. събирането и натрупването на данни, техният анализ и създаваната теория се намират във взаимна връзка и взаимодействие помежду си[3]. Затова е особено важно да се обърне внимание, че тези изследвания не започват с определена теория, която след това да се обосновава. Точно обратното, при тях трябва да се започне от областта на изследване, представляваща поле за изследователска работа, от което да се изведат съществените за бъдещото развитие на областта въпроси, принципи и решения.

Известно е, че изследователският процес при качествените изследвания има отворен характер. Това означава, че неговото протичане не е подчинено на предварително създадени схеми и модели. Характерно за качествените изследвания е, възникването на обрати, открития и непредвидени ситуации. Тези ситуации са част от изследователския процес. За тях се формулират принципи, тези, изводи, следствия, правила и т.н., които се използват както за критичен преглед и проверка на съществуващи концепции и теории, така и за създаването и разработването на нови теоретични конструкции. В частност те служат за съставяне на нови твърдения, които са основополагащи за обособяването на теория в определена област.

Чрез качествените методи изследваните явления и процеси се анализират съдържателно. Посредством тяхното прилагане изследователят трябва да вникне задълбочено в изследвания процес, като тълкува и анализира отделни негови прояви. Затова при тези изследвания изследователят е т. нар. „участващ наблюдател“ в изследването, т.е. той използва методи, чрез които се „доближава“ до изследваните лица, събития, процеси или непосредствено участва в тях. Според У. Филстийд[51] качествената методология позволява на изследователя да получава информация от първа ръка, като го поставя по-близо до фактите. Според С. Уйлсън[64], ефективността на качествените изследвания зависи от способностите на изследователя да превърне себе си в изследователски инструмент.

При качествения подход погледът на изследователя е върху цялостното, комплексно, интегрално обхващане на изследваните явления, предмети, лица или действителност. Предварително формулираните теории, хипотези и очаквания се проверяват в тяхната взаимовръзка и естествена среда. Поради това получаването на данните и тяхната оценка не могат да се разделят[3]. Според Д. Рост „Работата на терен не е просто, механично събиране на предварително определени данни от предварително определени лица. Много повече, през цялото време се осъществява една постоянна диалектика между *получаване и анализ* на данни, т.е. налице е постоянна оценка на това, което се знае, в сравнение с това, което предстои да се изследва“[3]. При „качествения експеримент“ се проверява предмета на изследването, като се наблюдава какви промени настъпват. Това изменение обаче не засяга същността и структурата на изследователския обект. Проверката се извършва

относно изменението на структурата на предмета на изследването и се търси връзка с условията, при които е осъществено изследването[3].

Чрез проведеното в настоящия труд качествено педагогическо изследване формулирахме нови схващания, концепции, тези, изводи, модели и подходи, които обособихме в теория, наречена *Методика на преподаване на съвременни технологии за създаване на софтуер*.

1.2. Статут и значение на съвременното образование по софтуерни технологии.

Съвременните информационни технологии и средства оказват огромно влияние върху всички прояви на политическия, икономическия, социалния и културния живот. Влиянието е осезаемо във всички форми и нива на образование. Същевременно това влияние има краткосрочен и дългосрочен ефект върху подготовката на преподавателите, програмите, методите и съдържанието на образователния процес, върху обществените отношения, изградени вследствие на полученото образование[13]. Новите форми на образование се характеризират с интерактивност и сътрудничество в процеса на обучение[29]. Познанията, придобити в процеса на обучение и ползване на нови технологии, рефлектират върху всички умения, свързани с житейската реализация на личността и възможността ѝ за социална интеграция[26]. Изменението на формите на взаимоотношение в социалната сфера неизбежно води до изменение на образователната система, в частност до изменение на методите и средствата на обучение. Проблемът за иновациите в образованието, за навлизането на новите модерни технологии е в основата на редица държавни документи, които насочват вниманието към прилагането им на широка основа, за постигане на масовост и достъпност за всички обучаващи се[7].

В резултат на развитие на технологиите глобалните процеси придадоха ново качество на съвременната социална система, определена като постиндустриално, информационно, глобално общество[14], често назовавано с епитета на Маршал МакЛуън – „глобално село“[56]. Този преход, чрез който се извършват същностни обществено-културни и икономически промени, е известен в литературата като „преминаване от индустриално към информационно общество (ИО)“[14] [34]. Изминал е повече от половин век откакто за пръв път се е придал съвременен смисъл на понятието „информационно общество“ от д-р Фриц Маклуп, но въпросите за неговото развитие са все така актуални[50]. Информационното общество е степен на развитие на едно общество, в което съвременните информационни технологии проникват в тъканта на самите отрасли на икономиката, инфраструктурата, образованието, бита и останалите области на живота[43].

Благодарение на разработването на нови информационни и комуникационни технологии информацията се превърна в парична мерна единица (т. нар. bitcoin) и източник на икономически ресурс, започна да се разглежда като вид обществено взаимоотношение. Въпросите за същността и еволюцията на ИО се превърнаха в основа на почти всички подобласти на информатиката и компютърните науки, но двигателят, който води до тяхното разрешаване и развитие си остава в приоритетите на софтуерното производство. Създаването на нови технологии и софтуерни решения е производствена дейност, която формира динамични социални взаимоотношения, особено чрез съвременните форми на компютъра като смартфон, таблет, лаптоп и пр. Примерите са многобройни и от изключително значение за различните социални сфери. Темите за т. нар. електронно управление и идентификация като е-правителство, е-образование, е-култура, е-здраве, е-икономика и т.н. са част от ежедневието социален диалог на обществото ни. Технологиите промениха провеждането на научните изследвания. Съвременната наука е глобална, развива се в условията на изграждащото се ИО. Характерни за нея са междудисциплинарните изследвания, интегрирането на знанията[5]. В България тенденциите в обучението не правят изключение от тези глобални процеси, които променят системите на образование.

Процесът на обновяване и актуализиране на образователното съдържание, усъвършенстването на методите и средствата на преподаване, внедрявайки съвременни технологии за създаване на софтуер в учебните програми протича бавно, сравнено с революционното развитие на технологията. Това обстоятелство изправя образователната система пред нуждата от непрекъснато осъвременяване; от

разработване и интегриране на гъвкави педагогически и методически подходи на преподаване на съвременни технологии, които да отговарят на потребности на обучаемите. Както винаги, когато се усъвършенствала по-радикално системата на обучение, най-голямо е изменението в сферата на образователното съдържание, методите и средствата[1]. Днес, повече от всякога, промяната, която касае проблема за разработването на методики на преподаване, е нужно да се погледне през призмата на *софтуерната технология*. Съвременната технология е неизменна част и от образователното съдържание, и от методите на обучение, и от средствата на преподаване. Тя е част от стандартите за съвременна грамотност. Затова образованието по информатика, насочено към изграждане на култура на потребление на информация и технологии, е особено важно. Още по-важно от технологичното потребление обаче е производството на нови софтуерни решения, тъй като тези решения са двигател на технологичните промени и катализатор на иновации. Софтуерното производство е предшествано от **въпроса за обучението на софтуерните разработчици**, т.е. *от намирането на решения на проблемите за преподаването и усвояването на специализирани знания и умения за създаване и разработване на софтуер*.

Качественото обучение на критично количество софтуерни разработчици днес е от изключително значение. На първо място такова обучение подготвя тесни компютърни и научни специалисти, необходими за образованието на обществото; на второ място, това обучение подготвя специалисти, които, създавайки нови софтуерни технологии и решения, могат да влияят пряко върху процесите на промяна и развитие на останалите социални сфери; на трето място, това обучение произвежда интелектуален капитал, представен като компетенции на софтуерните разработчици, който е сред основните фактори за развитие на ИО, ведно с изграждането на неговата инфраструктура и облик. Поради тези причини обучението по програмиране, изграждането на знания и умения за създаване и прилагане на софтуерни процеси, придобиването на компетенции за реализация на софтуерни проекти и изобщо разработване на софтуерни приложения и услуги са фактор както в личностното развитие на отделния човек, така и за цялостното развитие на обществото и социалната система. Нещо повече, в днешно време подготвянето на необходимото количество качествени софтуерни специалисти може и би трябвало да се разглежда като национално интелектуално богатство.

В близкото минало разглеждането на проблема за обучението на „завършения“ софтуерен разработчик е бил приоритет предимно за дисциплината „Софтуерни технологии“. Днес въпросите на това обучение са заложили в учебните планове на различни компютърни специалности и в учебните програми на редица софтуерни дисциплини. С други думи, през последните десетилетия се наблюдава процес на разрастване на обучението по софтуерни технологии, излизашо от границите на отделната дисциплина и разпростиращо се в обособяването на цели специалности. Поради тази причина въпросът за разработването на *Методика на преподаване на съвременни технологии за създаване на софтуер* е от значение не само за дисциплината „Софтуерните технологии“.

1.3. Състояние на обучението по информатика и софтуерни технологии.

За всички специалности в университетите на територията на страната, част от професионално направление „Информатика и компютърни науки“, водеща обща характеристика е силно заложеното обучение по програмиране и разработване на софтуер. Специалностите, в които тази тематика най-обстойно се изучава, са „Информатика...“ и „Софтуерни технологии и...“. Нещо повече, задълбоченият прочит на учебните планове и програми на тези специалности показва, че в обучението на студентите на проблематиката разработване на софтуер се отделя най-голямо внимание. Затова сме изследвали процеса на обучение в тези специалности. Проучването обхваща девет специалности: седем „Информатика...“ и две „Софтуерни технологии и...“ в седем различни университета. Проведено е през учебната 2013/2014г. за периода: от октомври 2013 до април 2014г. За подбор на специалности „Информатика...“ сме се ръководили от следните критерии: 1. Специалността да е масова, т.е. със засилен кандидатстудентски интерес и исторически най-дълго просъществувала в учебното заведение; 2. Учебното заведение да е акредитирано да провежда обучение; 3. Квалификационната характеристика, учебният план и учебните програми да са

публикувани и достъпни чрез официален или специализиран уеб източник. На тези изисквания към момента на провеждане на изследването отговориха седем бакалавърски специалности.

Изследването е разделено на три категории: 1. Учебен план и учебна програма на специалност „Информатика...“; 2. Учебен план и учебна програма на специалност „Софтуерни технологии и...“; и 3. Учебни програми на свободно избираеми дисциплини. В отделните категории на деветте изследвани специалности е направено числово обобщение, представено в табличен вид, състоящо се от: „Общ брой изучавани дисциплини“, „Дисциплини с фокус на обучението върху разработването на софтуер“ и процентно съотношение между двете. Освен това, съобразявайки се с тенденциите на развитие при методите и средствата за създаване на софтуер в софтуерното инженерство и индустрия за частта „Дисциплини с фокус на обучението върху разработването на софтуер“, на всяка от изброените категории анализирахме три нива: „Подходи, техники, стилове и практики на програмиране“; „Методологии за моделиране, създаване и приложение на софтуерни процеси“ и „Стандарти за управление и реализация на софтуерни проекти“. Конкретно изследваните обекти на всяка специалност и дисциплина са образователно-структурните елементи: *учебен план, учебна програма, анотация и тематично учебно съдържание*. Източниците, които използвахме в проучването, са официални образователни документи, публикувани и достъпни от уеб сайтовете на съответните учебни заведения [24, 25, 26, 27, 28, 32, 33].

Таблица №1: Задължителни учебни дисциплини на спец. „Информатика...“.

Учебно заведение	ФМИ на СУ	ФМИ на ПУ	ПМФ на ЮЗУ	ФМИ на ВТУ	ФМИ на ШУ	НБУ	ЦИТН на БСУ
Специалност	И	И	И	И	КИ	И	ИКН
Общо дисциплини	30	41	30	28	29	20	44
С фокус върху ОП, МСПСП и УРСП	12	19	11	12	12	16	19
В проценти	~40%	~46 %	~37 %	~43%	~41%	80%	~43 %

Легенда: СУ – Софийски университет "Св. Климент Охридски"; ПУ – Пловдивски университет "Паисий Хилендарски"; ЮЗУ – Югозападен университет "Неофит Рилски"; ВТУ – Великотърновски университет "Св. св. Кирил и Методий"; ШУ – Шуменският университет "Еп. Константин Преславски"; НБУ – Нов български университет; БСУ – Бургаски свободен университет; ФМИ – Факултет по математика и информатика; ПМФ – Природо-математически факултет; ЦИТН – Център по информатика и технически науки; И – Информатика; КИ – Компютърна информатика; ИКИ – Информатика и компютърни науки; ОП – Обучение по програмиране; МСПСП – моделиране, създаване и прилагане на софтуерни процеси; УРСП – Управление и реализация на софтуерни проекти.

Таблица №2: Задължителни учебни дисциплини на спец. „Софтуерни технологии и...“⁴.

Учебно заведение	ФМИ на СУ	ФМИ на ПУ
Специалност	СИ	СТД
Общо дисциплини	33	40
С фокус върху ОП, МСПСП и УРСП	15	22
В проценти	~46 %	~55 %

Качеството на обучението по програмиране според публикуваните образователни документи на специалностите съответства на технологичните новости. Програмирането се изучава много обстойно, но това от своя страна, наред с плюсовете, има и негативно влияние върху обучението. За разлика от програмирането, подходи, техники, практики и т.н., които са част от жизнения цикъл на реализация на софтуерния процес не се изучават задълбочено в спец. „Информатика...“. За разлика от нея, при спец. „Софтуерно инженерство“ и „Софтуерни технологии и дизайн“ дейностите по разработването на софтуер, като събиране и анализиране на изисквания, проектиране на архитектура и дизайн, проверка на изходен код(тестване), интегриране на функционалности и пр. се изучават с

⁴ Забележка: В Таблица №2 са използвани съкращения описани в легендата на Таблица №1

необходимата задълбоченост и отговарят на много от критериите и изискванията за качество на обучението на софтуерната индустрия[4]. За високите стандарти на качеството в тези две специалности говори засиленият интерес на кандидат-студентите.

Таблица №3: Свободно избираеми учебни дисциплини⁵.

Учебно заведение	ФМИ на СУ	ФМИ на ПУ
Специалност	всички спец.	всички спец.
Избираеми дисциплини	113	76
С фокус върху ОП, МСПСП и УРСП	30	22
В проценти	~27 %	~29 %

Слабата страна на обучението и при деветте специалности „Информатика...“ и „Софтуерни технологии...“ е свързва с тяхната класическа форма, т.е. процесът на обучение протича като теоретични лекционни курсове и практични упражнения, в които рядко е възможно да се приложи екипен принцип на работа. Съвременните гъвкави методологии за създаване на софтуер обаче изискват формите на обучение да имат екипен характер, дори съобразно техните принципи това е задължително. Друг недостатък на обучението е свързан с липсата на тематично учебно съдържание по управление и реализацията на софтуерни проекти. Изучаването на тази тематика дори в специалност „Софтуерни технологии...“ е заложено слабо. Като проблем на обучението може да се тълкува липсата на разработени подходи и модели за определяне на тематични връзки между учебните програми. Независимо от тези слаби страни, трябва да се отчете, че студентите, завършили професионално направление „Информатика и компютърни науки“ и в частност специалности „Информатика...“ и „Софтуерни технологии...“, са сред най-търсените специалисти на трудовия пазар, при това не само в софтуерната индустрия, но и извън нея. В по-общ план това говори за високите стандарти на обучение в направлението.

1.4. Тенденции и решения при преподаването на технологии за създаване на софтуер.

В резултат на анализ на обучението по информатика и софтуерни технологии, в дисертационния труд, е направен обзор на развитието на методиката на обучение по софтуерни технологии. В проучването сме се придържали към подход, чрез който са изследвани различни образователни проблеми. За изготвянето на обзора са използвани методите **анализ на съдържанието** на софтуерна, педагогическа и методическа литература и **анализ на съдържанието** на нормативни документи, като са анализирани източниците [1, 2, 6, 8, 9, 10, 11, 12, 16, 17, 18, 19, 20, 21, 22, 23, 30, 31, 35, 36, 38, 37, 39, , 40, 41, 42, 44, 45, 46, 47, 52, 53, 54, 55, 57, 60, 61, 62, 77, 63]. Също така, проведени са **30 активни участващи наблюдения – тип интервю** по предварително подготвени въпросник и списък с концептуални модели и графики (виж Приложение№1 на дисертационния труд), в които са обсъдени множество мнения, позиции, становища и др. В резултат на проучванията в глава първа са обобщени в 3 таблици числовите данни за общо 484 учебни програми, формулирани са 7 тези и направени 15 извода. По-конкретно са изследвани въпросите за: значението, ролята и функциите на образованието по софтуерни технологии; състоянието на обучението по програмиране, моделиране, създаване и прилагане на софтуерни процеси и по управление и реализация на софтуерни проекти; Също така, представени са някои общи изходни положения за методиките на преподаване на софтуерни технологии съобразени със спецификите на висшето образование. Генералният извод, който се наложи в обзора е, че методиките на преподаване предназначени за обучението на студентите по проблемите на софтуерното производство са слабо развити. Въпреки че съществуват десетки изследвания, в повечето случаи, казано най-общо обучението по разработване на софтуер е ограничено до програмни примери тип учебна задача за работа в час. Нещо повече, в специализираната литература, с малки изключения, се анализират предимно неговите технологични

⁵ Забележка: В Таблица №3 са използвани съкращения, описани в легендата на Таблица №1

аспекти, а въпросите за т. нар. е-управление и идентификация като е-правителство, е-здравеопазване, е-банкиране и др. рядко са попадали в ползването на изследователите.

1.5. Резултати от първа глава.

- Разработена е постановка на изследванията в дисертационния труд;
- Оценена е степента на развитие на методиката на обучение по софтуерни технологии;
- Представени са развитието на образованието и обучението по информатика и софтуерни технологии;
- Изготвена оценка на състоянието на обучението по информатика и софтуерни технологии в седем от водещите университети на страната, актуална за учебната 2014/2015 година;

1.6. Някои тези и изводи от първа глава.

Теза №1: *Изграждането на софтуерни решения и технологии е производствена дейност с висока добавена стойност, чрез която се създават динамични и мобилни социални взаимоотношения посредством съвременните форми на компютърна техника.*

Теза №2: *Обучението по програмиране; моделирането, създаването и прилагането на софтуерни процеси; управлението и реализацията на софтуерни проекти и изобщо разработването на софтуерни приложения и услуги е от значение не само за образованието по информатика, а е фактор за цялостното развитие на обществото и социалната система.*

Теза №6: *Методиките на обучение по софтуерни технологии са изправени пред професионално педагогическо задължение и необходимост от честти, ежегодни промени на учебното съдържание.*

Извод №1: *Методиката на обучение по софтуерни технологии е слабо развита.*

Извод №6: *Методиката на преподаване по софтуерни технологии е слабо развита, като подходи на обучение, които да обхващат не само технологичните, но и различни административни, икономически, правни, образователни, на здравеопазването, по сигурността и др. аспекти на софтуерното инженерство не е забелязано да са разработени.*

Извод №9: *При прилагането на съвременни, модерни и не винаги традиционни методи и средства на обучение е подходящо да се използва комплексен подход от директни и индиректни форми на обучение, съобразени с конкретните контекстуални особености, сред които: моделиране на учебни планове и програми, прилагане на проектно-базиран подход, прилагане на проблемно-базиран подход на обучение, употребата на технологии за електронно обучение, използването на мултимедийни и симулационни учебни средства, разработване и адаптиране на стандарти и др.; подходи на преподаване стимулиращи както самостоятелната работа, така и уменията за работа в екип; подходи за намаляване на сложността и абстракцията софтуерните технологии; подходи за преподаване на концептуални познания за реализация на софтуерни проекти и др.*

Извод №10: *При разработването на методика на преподаване по софтуерни технологии е необходимо да се заложат директни и индиректни форми на обучение, които паралелно с технологичните знания, умения и компетенции да развиват и интелектуалните заложиби на студентите. При директните форми на обучение, характерни със строга структура на изложение на учебно съдържание, такова развитие се постига, чрез развиващото алгоритмично мислене и развиващото аналитично мислене. При индиректните форми на обучение, характерни със нестрога структура на изложение на учебно съдържание, такова развитие се постига, чрез развиващото моделиращо мислене и развиващото интуитивно мислене.*

Извод №12: *Сферата на приложение на методиката на преподаване на съвременни технологии за създаване на софтуер обхваща учебните дисциплини, които се четат в специалности „Информатика...“ и „Софтуерни технологии и...“, но нейната приложност не е ограничена единствено до курсовете, които се изучават в тези специалности.*

Извод №13: *В учебния план и програми на специалности „Информатика...“ съвременните техники и практики по програмиране са силно заложили и се изучават на високо ниво; същите по моделиране, създаване и приложение на софтуерни процеси са слабо застъпени и неравномерно*

разпределени, докато учебното съдържание по управление и реализация на софтуерни проекти не е предвидено за изучаване в отделна учебна дисциплина. В учебните планове на специалности „Софтуерно инженерство“ и „Софтуерни технологии и дизайн“ програмирането също както при специалност „Информатика...“ е силно застъпено и се изучава на високо ниво; техниките и практиките по моделиране, създаване и прилагане на софтуерни процеси са добре разпределени в учебния план и програми, но броят на предвидените теми е незадоволителен (в пъти по-малък от темите, предвидени по програмиране); докато предвиденото учебно съдържание за обучение по управление и реализация на софтуерни проекти е неудовлетворително предвид тенденциите на тяхната употреба в софтуерната индустрия.

ГЛАВА 2. Разработване на Методика на преподаване на съвременни технологии за създаване на софтуер.

2.1. Концепции и теоретични основи за разработване на методиката.

При подготовката и провеждането на изследванията, изложени в дисертационния труд, взехме под особено внимание обстоятелството, че в областта на методиката на обучението по софтуерни технологии не беше забелязана разработка на осъществено качествено педагогическо изследване. Затова разработихме две концепции: Концепция за реализиране (разработване и приложение) на Методика на преподаване на съвременни технологии за създаване на софтуер и Концепция за провеждане на качествено педагогическо изследване в областта на методиката на обучението по софтуерни технологии.

Изграждането на Концепция за провеждане на качествено педагогическо изследване по методика на обучението в областта на софтуерните технологии е повлияно от схващанията за методологията на качествените педагогически изследвания на Г. Бижков, и В. Краевски [3], И. Иванов [15] и Б. Глейзър и А. Щраус [48]. Концепцията е разделена на две части: **общи принципи и изисквания** за провеждане на качествени педагогически изследвания и **критерии за изпълнение** на реализираното в труда качествено изследване по методика на обучението по софтуерни технологии. Такова разделение се наложи, защото качествените педагогически изследвания по информатика и компютърни науки не са развити в българското научно пространство, а общите методи за тяхното приложение не са адаптирани за тесни научни подобласти.

Принцип №1: При провеждането на качествени педагогически изследвания изследователят трябва да е „на терен“, в близост до естествената среда на работа на изследваните лица и в активно взаимодействие с тях като „участващ наблюдател“.

Изискване №1: При прилагането на качествени методи за събиране на информация като анализ на съдържанието, участващо наблюдение, открито интервю и др. събраната информация няма статистически доказателствен характер, т.е. ако изследователят си служи с количествено изразени величини (като напр. честота, процент, средна величина и др.), те трябва да са единствено източник на информация за анализ, но не и средство за проверимост и достоверност на получения резултат.

Принцип №2: По време на провеждане на качествено педагогическо изследване протича диалектика между наличната и текущо получаваната информация.

Изискване №2: В методите на качествените изследвания, като част от инструментариума на тези експерименти, едновременно се прилагат процедури по събиране, оценка, анализ, интерпретация и систематизация на информацията.

Критерий №1: При провеждането на качествени педагогически изследвания в областта на методиката на обучението по софтуерни технологии да се предлага, наравно с допълването и разширяването на учебната тематика, подход за интеграция на напълно ново тематично учебно съдържание, в това число и нови учебни програми.

Критерий №2: Данните, получени от провеждането на качествено педагогическо изследване в областта на методиката на обучението по софтуерни технологии, трябва да са изведени от изследователско поле, представляващо обединение на областите методика на обучението и

софтуерно инженерство, а протичането на експеримента да не е „подчинен“ на предварително обособена теория.

Критерий №3: При създаването, развитието и верификацията на нова теория в методика на обучението по софтуерни технологии, чрез прилагане на качествено педагогическо изследване, процедурите по събирането, оценката, анализа, интерпретацията и систематизацията на информацията, включително и обосноваването на теорията, трябва да са в непрестанна взаимовръзка помежду си.

Критерий №4: За да е приложима в обучението по софтуерни технологии, теорията на методиката трябва да е **реалистична, разбираема, управляема и обобщена**.

Критерий №5: За да подлежат на верификация, използваните методи и процедури на качествено проведено изследване в методика на обучението по софтуерни технологии, е необходимо моделът на провеждане на педагогическия експеримент да съдържа компонентите: теоретична чувствителност, кодиране на данните, матрица на условията и теоретична извадка.

Критерий №6: За да се получат допълнително аргументи за валидност, вярност и истинност на получените резултати от провеждането на качествено педагогическо изследване за методика на обучението по софтуерни технологии, трябва при събирането на данните да са приложени способности като организационна ефективност, дисциплинарна субективност, аргументирана интерпретация, комуникативна валидизация и др.

Конструирането на **концепция за реализиране на „Методика на преподаване на съвременни технологии за създаване на софтуер“** е съставена като **система** от: 1. **Методи и подходи**, чрез които се извлича, класифицира, структурира и внедрява в процеса на обучение тематично учебно съдържание и педагогически подходи, методи и средства на преподаване; 2. **Структура** на общ йерархичен интеграционен модел, съставен от пет градивни елемента (виж фиг. №4) и четири йерархични компонента на базовия градивен елемент разгръщащ образователна парадигма на обучението по софтуерни технологии (виж фиг. №2); 3. **Критерии, показатели и параметри** за оценка на резултати (виж Таблица №4). Освен това, трябва да се има предвид, че: **концепцията за реализиране на методиката е частен случай на приложение на концепцията за провеждане на качествено педагогическо изследване**.

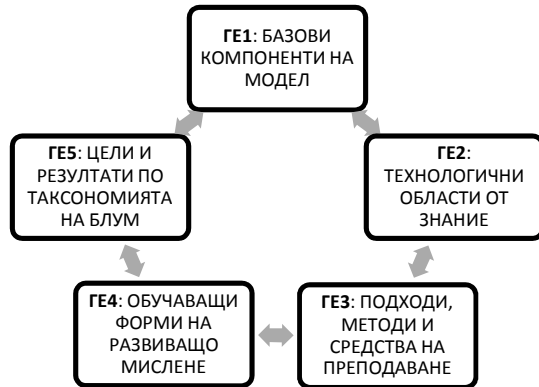
Таблица №4: Критерии, показатели и параметри за оценяване на резултати.

КРИТЕРИИ	ПОКАЗАТЕЛИ	ПАРАМЕТРИ
Извличане на актуално тематично учебно съдържание от теориите и практиките на съвременните технологии за създаване на софтуер в софтуерното инженерство и индустрия.	1. Обособяване на технологични области от знания. 2. Структуриране на области от знания.	1. Степен на актуалност за област от знание в софтуерното инженерство и индустрия. 2. Степен на приложност на област от знание в индустрията. 3. Степен на приложност на област от знание в обучението по софтуерни технологии.
Построяване на общ йерархичен интеграционен модел	1. Класифициране и структуриране на тематично учебно съдържание. 2. Класифициране на методи, прийоми, похвати и способности на преподаване. 3. Класифициране на технологични средства за преподаване.	1. Подбор на методи, прийоми, похвати и способности на преподаване. 2. Подбор на технологични средства за преподаване. 3. Актуализиране на учебно съдържание в модела.

Съставяне на учебни програми посредством общ йерархичен интеграционен модел.	1. Структуриране на тематично учебно съдържание от областите от знание. 2. Избор на методи и средства за преподаване. 3. Формализиране на учебната програма като образователен документ.	1. Поставяне на оценки за ключови знания, умения и компетенции в обособената област по Таксономия на Блум. 2. Определяне на обучителни форми на интелектуално развиващо мислене.
Внедряване на учебно съдържание	1. Принципи за внедряване на учебната програма.	1. Цели и задачи на обучението по учебната програма. 2. Провеждане на педагогически експеримент 3. Резултати от обучението.

2.2. Изграждане на *Общ йерархичен интеграционен модел*.

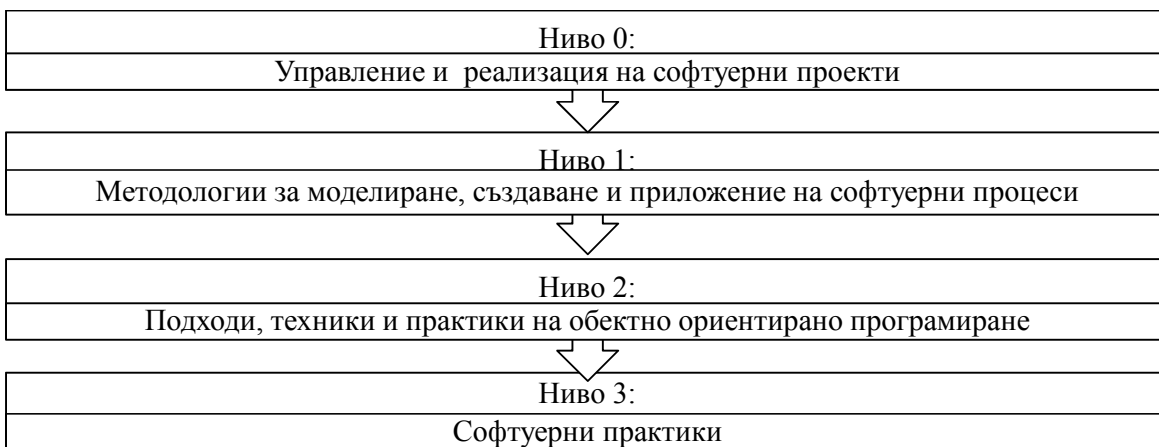
Концептуалният модел, представен на *Фиг. №1*, е съставена от пет градивни елемента. В началото моделът е „празен“, като запълването му със съдържание се извършва в хода на протичане на изследването. Последователността, в която са изследвани софтуерните теории и практики, е определена от историческото развитие на информатиката и софтуерното инженерство. Тези теории и практики, предварително, преди провеждането на изследванията сме йерархизирали по сложност и абстракция на три нива. Както може да видим на *Фиг. №2*, това са етапите на развитие на програмирането, еволюцията на моделите и методологиите за създаване на софтуерни процеси и приложението на стандарти за управление и реализация на софтуерни проекти. Към тях, от гледна точка на разгръщането на образователната парадигма на обучението по софтуерни технологии, добавяме четвърто ниво наречено *студентски софтуерни практики*.



Фиг. №1: Градивни елементи (ГЕ) на общ йерархичен интеграционен модел.

Построяването на общия йерархичен интеграционен модел преминава през няколко етапа свързани с провеждането на различни, независими едно от друго изследвания, сред които анализ на съдържанието на научна, нормативна и специализирана литература, провеждане на методически експеримент с изследователи, преподаватели и разработчици на софтуер по метод на активното, участващо наблюдение – тип интервю, анкетиране на студенти и провеждането на педагогически експеримент за работа в екип чрез прилагане на метода „Групова дискусия“ (Фокус метод). По време на различните етапи от построяването на модела се дискутират проблеми свързани със смяна на парадигмата в програмирането; контекст на софтуерната разработка при преподаването на обектно ориентираната парадигма; моделирането, създаването и приложението на софтуерни процеси при традиционните модели и гъвкавите методологии; управление и реализация на софтуерни проекти; обучението чрез т. нар. студентски практики, като връзка между наука, образование и бизнес;

конструирането на хранилище с технологични области от знания и области за подбор на подходи, методи и средства на преподаване; структуриране на учебен план и програми чрез Таксономия на Блум; проблема за връзката между различните дейности по разработване на софтуер и формите на развиващо мислене и др.



Фиг. №2: Образователна парадигма на обучението по софтуерни технологии.

При построяването на компонента „Подходи, техники и практики на програмиране“ специално внимание е отделено на въпросите за това: как през годините създаването на софтуер от отъждествяването си с дейността програмиране се е размножило в дейности като анализ на изисквания, проектиране на модели, моделиране, писане на изходен код, проверка за качество на кода, интеграция на софтуер и др.; как се е променила парадигмата в програмирането, как това е повлияло на обучението и др. В следствие от построяването на компонентата е разработен *подход за съставяне на учебна програма за изучаване на обектно ориентирана парадигма*.

При построяване на компонента „Моделиране, създаване и приложение на софтуерни процеси“ са анализирани тенденциите на развитие при моделите и методологиите за моделиране, създаване и приложение на софтуерни процеси. Анализът е разделен на два етапа: 1. Традиционни модели за реализация на софтуерни процеси; и 2. Гъвкави методологии за реализация на софтуерни процеси. В следствие на изследванията синтезираме техни принципи, характеристики, подходи, техники и практики подходящи за изучаване от студенти. Синтезът на подходите, техниките, практиките и т.н. е разделен на три категории: *концептуален модел* на софтуерен процес; *организационни и управленски подходи*; и *конструктивни техники, практики и средства*⁶. Изследвани са традиционните софтуерни модели, които в миналото са придобили най-широка популярност, употреба и приложение. Тъй като през последното десетилетие тези модели не се употребяват самостоятелно като средства за създаване на софтуер в софтуерното инженерство и индустрия, изследваме и анализираме тяхното съдържание от принципи, характеристики, подходи, техники и практики, които са повлияли на развитието или се използват в съвременните гъвкави методологии за създаване на софтуер.

След анализ на специализираната литература, проведените интервюта, беседи и анкетни карти, позовавайки се и на личния си професионален опит се състави **Списък №1: Подходи, техники и практики използвани за създаване на софтуер при „традиционните“ софтуерни модели, които са подходящи за приложение в обучението:**

- итеративен подход на работа;
- фиксиране на времето (time boxing) за работа по итерациите;
- кратко и неформално проектиране на дизайн;

⁶ Забележка: В тази категория под средството се има предвид: ръководства, указания, формуляри по образец и др. помощни материали на методологиите.

- идентификация, анализ и оценка на риска;
- управление на риска;
- минимизация на погрешни решения;
- гъвкава и адаптивна реакция към промените на изискванията към системата;
- бърза разработка на прототипи;
- тясно сътрудничество с клиента;
- ограничение на обхвата на разработката;
- установяване на междинни цели, преглед и анализ на текущото състояние;
- планиране на алтернативни решения.

Постепенното усъвършенстване и разрастване на софтуерните модели, като средство за организиране на работния процес и справяне със сложността и абстракцията на софтуерните разработки е довело до създаването и прилагането на много нови софтуерни решения, подходи и стандарти за разработка на софтуер. Прогресът на развитие на софтуерните техники и практики с течение на времето води до разглеждането на моделите като рамка за реализация на софтуерните процеси, а структурирането на процеса, по които се разработва софтуерът, започва да се обсъжда в научната литература и софтуерна практика като *методология*⁷. Методологиите намират отговор на потребностите на софтуерната индустрия през последните две десетилетия и се предпочитат пред традиционните подходи за разработка на софтуер, тъй като реализират динамичен софтуерен процес и по-бърза разработка в сравнение с тромавите, детайлно документирани традиционни софтуерни модели. Някои автори разглеждат този преход, като дават следното определение на понятието методология: МЕТОДОЛОГИЯ = МОДЕЛ + ТЕХНОЛОГИЯ(И)[63]. Други използват термина *традиционни(стандартни) методологии* като синоним на *софтуерен модел*, а за еквивалент на понятието *методология* си служат със съвременния термин *гъвкава методология* (*agile methodology*)[49], [65]. Понятието софтуерна методология е термин, включващ множество подходи, техники и практики за разработване на софтуер. Гъвкав метод (или методология) е обобщаващ термин, покриващ много процеси за създаване на софтуер, които имат общ набор от стандарти и принципи, дефинирани в „*Манифест за динамична разработка на софтуер*”[58].

От разширения модел на еволюция на методологиите, представен на *фиг. №12* в дисертационния труд, подробно сме изследвали *Унифициран процес (Unified Process, UP)*, *Скрум (SCRUM)* и *Екстремно програмиране (Extreme Programming, XP)*. Измежду всички методологии в модела сме подбрали тези три, тъй като, от една страна, те са сред най-прилаганите в софтуерното инженерство и индустрия, а от друга страна, сме взели предвид личния професионален и педагогически опит за работа с тях при реализацията и преподаването на софтуерни процеси.

За разлика от „традиционните“ модели при съвременните гъвкави методологии както концептуалните модели, така и организационните и управленските подходи, техники и практики притежават по-ясно изразени характеристики. Затова, преди анализа на трите методологии, формулираме характеристиките за концептуален модел, организационен и управленски подход и конструктивна техника, практика и средство на методология:

- **Концептуален модел** наричаме схематичното описание на концепцията на работния цикъл на реализация на софтуерния проект или модела на жизнения цикъл на софтуерния процес;

- **Организационни и управленски подходи** са тези подходи, които се отнасят до моделирането, създаването и/или прилагането на способ или форма на *организация и управление* на жизнения цикъл на разработка на софтуерен процес или до определянето на *взаимоотношенията* между участниците и заинтересованите страни (обикновено разглеждани като актьори на роли) по реализация на софтуерния проект;

Забележка ⁷: В научната и специализирана литература по софтуерно инженерство често като синоним на понятието *методологии* се използва термина *метод*.

- **Конструктивен елемент** е техника, практика или средство за реализация на софтуер, която е част от моделирането, създаването или прилагането на софтуерен процес.

Списък №2⁸: *Организационни и управленски подходи и конструктивни техники, практики и средства при Унифициран процес, Скрум и Екстремно програмиране:*

- итеративен подход на разработка;
- емпиричен подход за контрол на работните процеси;
- дисциплинираност, т.е. стриктно спазване на правила за *прозрачност (transparency)*, *инспекция (inspection)* и *адаптация (adaptation)*;
- споделяне на общи ресурси;
- използване на централизиран достъп до информацията;
- споделяне на общи стандарти за документиране на работата;
- планиране на ключови събития (milestones);
- управление на промени, качество на софтуерната разработка, ресурси, рискови фактори и др.;
- степенуване на отговорностите;
- тясно сътрудничество между специалистите с различни отговорности
- постоянно проследяване на прогреса;
- установяване на отклонения от плана и работните спецификации;
- коригиране на отклоненията;
- пренареждане на работния поток
- програмиране по двойки (pair programming);
- тясното сътрудничество на разработчика с клиента (т.е. възложителя и потребителя);
- кратки срокове на доставка на функционалност;
- визия за работата на приложението;
- разработване, базирано на тестове (test-driven development);
- рефакторинг (refactoring);
- постоянна интеграция (continuous integration);
- малки доставки на функционалност (small releases);
- организационна концепция „роля, отговорност, задача, продукт“;
- разделения на събраните изисквания на функционални и нефункционални;
- приоритетизиране на изискванията по т. нар. метод MoSCoW (Must-Should-Could-Would);
- визуално проектиране на дейност, случай на употреба, клас, комуникация, последователност и др.;
- подход за идентифициране на обекти с т. нар. CRC⁹ (Class-Responsibility-Collaboration) карти;
- поддържане на готови шаблони за различни документи
- и др.

При построяването на компонентата „Управление и реализация на софтуерни проекти“ са изследвани два *стандарта за управление и реализация на софтуерните проекти* – Системата от знания за управление на проекти (СЗУП, Project Management Body of Knowledge – PMBOK[59]) и Интегриран модел за зрялост на възможностите (ИМЗВ, Capability Maturity Model Integration – CMMI). Също така, анализирани са повечето от добре познатите гъвкави методологии по три показателя: дейности и практики за ръководене на софтуерен проект; дейности и практики за реализация на софтуерен процес и задачи, роли и продукти за изграждане на софтуерен процес, които наричаме *конструктивни елементи*. В следствие от анализа на стандартите и методологиите е

⁸ В дисертационни труд подходите, техниките, практиките и средствата на трите методологии са описани отделено.

⁹ CRC (Class-Responsibility-Collaboration) изобретена от Уорд Кънингъм в края на 80-те не е част от стандартите на RUP, но е широко използвана техника за дизайн на обекти.

направена съпоставка на конструктивните елементи между едните и другите. Съществена черта на съпоставка между гъвкавите методологии и стандартите за управление е степента на зависимостта им от технологията. Употребата на стандарти за управление на проекти е в пълна степен технологично независима, докато при прилагането на гъвкави методологии поради контекста на софтуерната разработка съществува *степен на технологична зависимост* на съответния подход, техника и практика.

При реализацията на софтуер в концептуалната перспектива работата е фокусирана върху изграждането на софтуерния проект, докато в софтуерна перспектива тя се свързва с реализацията на софтуерен процес. При съвременните методологии за разработване на софтуер дейностите, които са част от обсега и на двете перспективи, са обединени, но от изграждането на софтуерния проект зависят всички следващи действия по реализацията на софтуерния процес. В този смисъл *изграждането на софтуерния проект* е основополагащ за *реализацията на софтуерния процес*. От тази гледна точка е подходящо, преди да се приложи интегрирано планиране на управлението на софтуерния проект и реализацията на софтуерния процес, като част от някоя методология – параметрите на софтуерния проект да се изследват независимо от конкретната реализация на софтуерния процес.

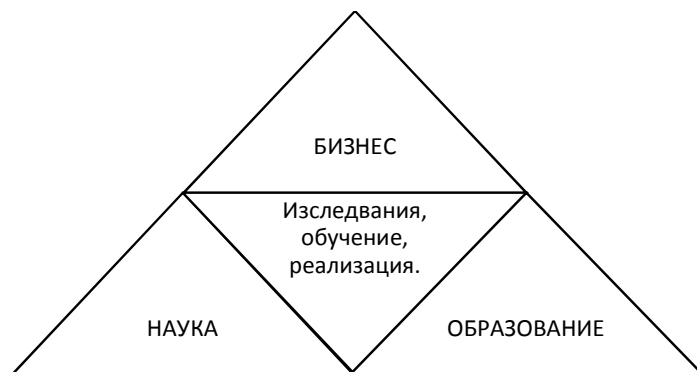
В някои случаи на разработка на софтуер употребата на стандарти за управление на проекти с подробната и детайлна документация е недостатък, но за целите на обучението на начинаещия софтуерен разработчик концептуалната технологично независима теория е предимство при усвояването на управленски знания, умения и компетенции за планиране на софтуерни проекти.

При построяване на компонента „Студентски практики – връзка между наука, образование и бизнес“ е изследван проблемът за провеждането на студентски софтуерни практики като сътрудничество между наука, образование и бизнес. Този въпрос не може да се разреши изолирано, без участието на всяка от страните. Същевременно организацията на такива практики надхвърля обхвата на дисертационния труд. Затова в работата сме се ограничили единствено до неговото теоретично решение, като предлагаме **подход на обучение чрез студентски практики**, състоящ се от:

1) **Обща рамка за изготвяне на образователна форма на обучение в условия на:**

- a. *сътрудничество между наука, образование и бизнес;*
- b. *бизнес среда на работа и обучение.*

2) **Теоретично-идеализирана визия** за тясна специализация на обучение по софтуерни технологии.



Фиг. №3: Релационни процеси между наука, образование и бизнес.

Следващата стъпка от построяването на *Общия йерархичен интеграционен модел* е изграждането на **структура** и съставянето на **съдържание** на градивните елементи ГЕ-2, ГЕ-3, ГЕ-4 и ГЕ-5, съответно: *Хранилище с технологични области от знания; Области за подбор на подходи, методи и средства на преподаване; формите на развиващо мислене и цели и резултати по Таксономия на Блум.*

Вследствие на анализа на съдържанието на специализираната литература и в резултат на синтеза на отговорите от интервютата обособихме **четири области на приложни технологии: софтуерни архитектури; среди и средства за разработване на софтуер; сървърни и уеб технологии и технологии за работа с бази от данни.** Отделно разглеждаме пета област, която наричаме **базови софтуерни концепции.** Тази област я отделяме от останалите четири, тъй като нейното съдържание е съставено от градивния елемент ГЕ-1, чрез който се разгръща образователната парадигма на обучението по софтуерни технологии. Характерно за базовите софтуерни концепции е, че те **притежават трайност на съществуване и приложение в софтуерната индустрия,** която е по-продължителна във времето от концепциите на приложните технологии в останалите области.

За разлика от голямото разнообразие и динамика при технологиите съпътствани от чести изменения и промени, при обособяването на методите на обучение е минал бурният период на развитие.

В дисертационния труд е представена *Класификация на методите на обучение* по компютърни науки, разработена от Д. Дурева (виж Гл.1, т.1.3.2.3). Към тази класификация могат да се добавят и някои нови и неразработени методи на обучението, които са породени от специализирани дейности в софтуерната и преподавателска практики при преподаването на специфични технологии. Такъв метод, частен случай на метода дедукция, е *абстрахирането*, което се използва при преподаването на сложни и високо абстрактни софтуерни технологии. Друг метод, вариация на метода демонстрация, е *софтуерната симулацията*, която, например, успешно се използва за онагледяването на интернет комуникация при моделите клиент-сървър, а и изобщо при разработването на по-сложни мрежови софтуерни приложения. Трети метод, който спада към групата методи, класифицирани според основната дидактическа цел е *моделирането*, като част от обучението на студенти по проектиране на софтуерни модели. Проблемът за класификацията на *абстрахирането, софтуерната симулацията и моделирането* като нови методи на обучението е отговор на въпроса за начина и подхода на преподаване и усвояване на специфично познание в тясно научна област, което не е познато за педагогическата практика и за което по необходимост се налагат нови подходи за неговото преподаване.

Въпреки че методиката на обучение е преминала през интензивния период на развитие за основните си методи на преподаване, образователната практика показва, че в обучението по софтуерно технологии за прилагането на почти всеки метод на обучение е необходимо да се разработи подход за негова адаптация. Такива подходи на адаптация правят методите на преподаване актуални и адекватни на спецификите на обучението по софтуерни технологии. Поради тази причина разглеждаме проблема за развитието на методите на обучение, предвидени за преподаването на софтуерни концепции и приложни технологии, като въпрос на обогатяване на техните възможности на приложение чрез подбор и разработване на подходи и средства за адаптация. Тези подходи и средства в преобладаващата си част, са разработени в научната литература по методика. Те от своя страна, когато са поставени в конкретен контекст на определена подобласт на софтуерните технологии, също така се нуждаят от приспособяване специфично на тематиката. Затова говорим за **подходи и средства за адаптация на методите на обучение,** които от своя страна е необходимо да се приспособят към тематичния контекст. Поради това към *Класификацията на методите на обучение* съставихме **Списък №3: Подходи и средства за адаптиране на методите на преподаване в обучението по софтуерни технологии:**

Подходи:

- Проектно-базиран подход на обучение;
- Проблемно-базирани подходи за провеждане на проблемно-базирани упражнения;
- Подходи за актуализиране и усъвършенстване на учебно съдържание и преподавателски практики на преподаване посредством осъществяването на обратна връзка със студенти, които да оценяват качеството на преподаване чрез провеждането на анкетни карти и беседи;
- Подходи за провеждане на активно обучение (active learning), което поставя обучаемия в центъра на образователния процес;

- Педагогически похвати за повишаване на мотивацията на обучаемите;
- Участие в различни видове образователни игри;
- Експериментиране със симулационни модели;
- Използването на компютърни системи за електронно обучение;
- Подходи на *конструктивизма* за учене чрез експериментиране и откриване; обучение чрез запитване и търсене на информация (Inquiry Learning); обучение към критично мислене (Critical Thinking Learning) и др.;
- Подходи на *конективизма* за съчетание на формално и неформално образование в мрежова и технологизирана среда; свързване на специализирани елементи от информационни източници; откриване на връзки между области, идеи, концепции и др.;
- Защитаване на концепции, тези и самостоятелни разработки от обучаем пред професионална и научна аудитория (т. нар. публична защита);

- и др.

Средства:

- *Технологията като среда за обучение* – т. нар. интегрирани среди за разработка на софтуер;
- *Технологията като средство за работа* – в това число езици за проектиране, моделиране и програмиране;
- Използването на графични нотации за онагледяване на сложни теоретични концепции;
- Употребата на технологии и системи за електронно обучение;
- Използване на модулни интеграционни компютърни системи в помощ на обучението при генериране на въпроси и тестове, визуализиране на алгоритми, автоматизиране на оценяването и др.;
- Използването на мултимедийни и симулационни компютърни системи;
- Прилагане на технологични прототипи като учебни обекти
- и др.

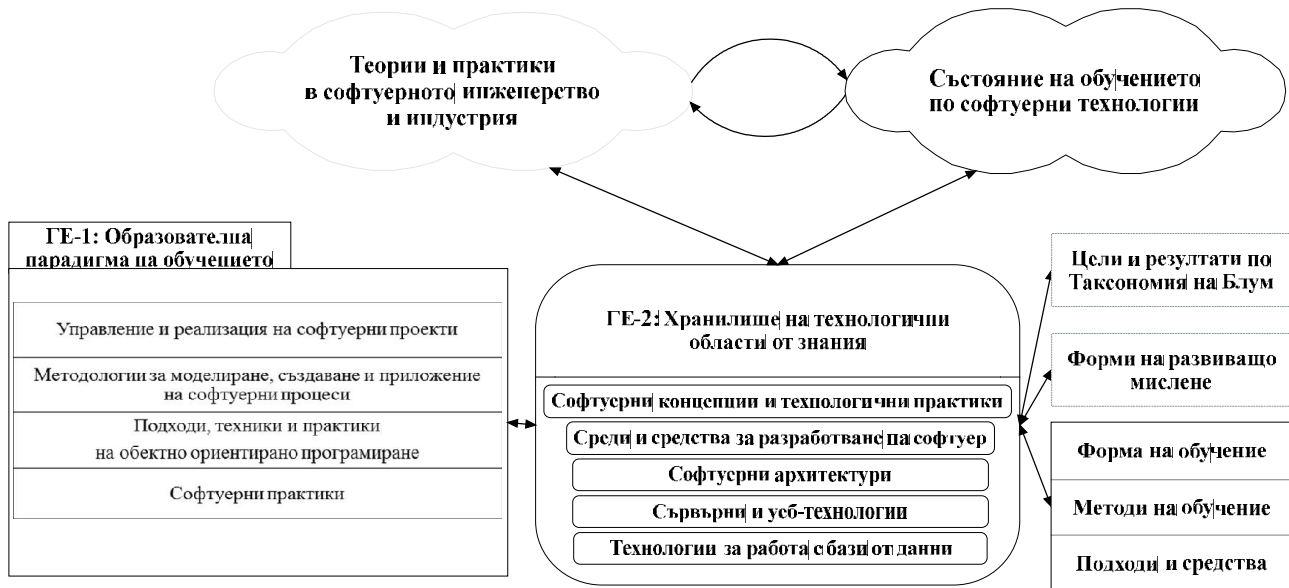
При изграждането на *ГЕ-4 – форми на развиващо мислене*, след задълбочен анализ на специализираната литература по проблема и дискусия с интервюираните при активното, участващо наблюдение – тип интервю стигнахме до заключение, че в областта на софтуерното инженерство, формите на мислене, които най-точно отговарят на критерия за обобщеност, като начин на мислене и на конкретност, по отношение на съвременните техники и практики за създаване на софтуер, са: алгоритмичното, аналитичното, моделиращото и интуитивното мислене.

При изграждането на *ГЕ-5 – Цели и резултати по Таксономия на Блум* е предложен подход за определяне на междудисциплинни връзки. Такива връзки могат да се определят между дисциплини, в които: има надграждане на тематичното учебно съдържание; по-задълбочено се изучава определена концепция; или се усъвършенстват знанията и уменията за работа с приложна технология изучавани в друга дисциплина. Така в резултат на определяне на тематичните връзки между дисциплините се откриват възможности за синхронизиране, адаптиране и оптимизиране както на учебните програми, така и на тяхната взаимосвързаност. Част от този подход беше изграждането на **общ списък със софтуерни концепции и приложни технологии**. Когато в този списък попаднат всички софтуерни концепции и приложни технологии, изучавани в дисциплини от една специалност, ясно си проличава кои от тях се надграждат. Това са дисциплините, в които могат да се определят междутематични връзки за целите и резултатите по Таксономия на Блум.

В разработената методика на преподаване най-съществената функция на *ГЕ-5* е *връзката между поставената образователна цел на входа на висшето образование, процесът на обучение като комплекс от дейности за всички тематично свързани учебни дисциплини и постигнатия образователен резултат на изхода на висшето образование*.

2.3. Схема на Общ йерархичен интеграционен модел.

От гледна точка на разработването на методиката моделът има концептуален характер. Той може да послужи както за анализирането на общи проблеми, така и за намирането на конкретни решения за обучението по софтуерни технологии.



Фиг. № 4: Общ йерархичен интеграционен модел.

2.4. Резултати от втора глава.

- Разработване и прилагане на: Концепция за провеждане на качествено педагогическо изследване по методика на обучението в областта на софтуерните технологии;
- Разработване и прилагане на: Концепция за реализиране на *Методика на преподаване на съвременни технологии за създаване на софтуер*;
- Построяване на общ йерархичен интеграционен модел за структуриране, разпределяне, формализиране, адаптиране и актуализиране на тематично учебно съдържание, подходи, методи и средства на обучение по софтуерни технологии;
- Обобщение на подходящи за приложение в обучението: *организационни и управленски подходи и конструктивни техники, практики и средства* използвани за създаване на софтуер при гъвкавите методологии УП, Скрам и ЕП; *подходи, техники и практики* използвани за създаване на софтуер при софтуерните модели; *образователни подходи и средства* за адаптиране на образователните методи на преподаване в обучението по софтуерни технологии;
 - Разкриване на образователна парадигма на обучение по софтуерни технологии;
 - Разработване на разширен модел на развитие на гъвкавите методологии за създаване на софтуер;
 - Разработване теоретично решение: *подход на обучение – студентски софтуерни практики: обща рамка за изготвяне на образователна форма на обучение и теоретично-идеализирана визия за тясна специализация*;
 - Категоризиране на формите на развиващо мислене в обучението по софтуерни технологии.
 - Разработване на подход за съставяне на учебна програма за преподаване и изучаване на обектно ориентирана парадигма.

2.5. Някои тези и изводи от втора глава.

Теза №8: *Смяната на парадигмата на програмиране* към настоящия момент не е довела до смяна на парадигмата на преподаване.

Принцип на подход №1: *Преподаването на обектно ориентираната парадигма е основано на изучаването на три дейности: изучаване на обектно ориентиран анализ (ООА), изучаване на обектно ориентиран дизайн (ООД), т.е. проектиране и моделиране на обекти и изучаване на обектно ориентирано програмиране (ООП).*

Правило на подход №1: Тематичното учебно съдържание за изучаване на дейностите ООА, ООД и ООП трябва да се разработва и представя в посочената последователност, като се взема под внимание съвместимостта на технологиите между тях.

Принцип на подход №2: Преподаването и изучаването на ООА е независимо от изучаването и прилагането на технологиите за проектиране, моделиране и програмиране. ООА е технологично независима дейност.

Правило на подход №2: Да се разработва така тематичното учебно съдържание, че в процеса на обучение да се постигне плавен преход от технологичната независимост при ООА към технологичната зависимост при ООД и ООП.

Извод №16: При преподаването на обектно ориентирана парадигма на програмиране съществува практика, повлияна от структурни разсъждения, при която организацията на учебното съдържание се изгражда с цел изучаване на синтаксис и семантика на език за програмиране, а не на основа характеристиките на подхода за програмиране.

Извод №17: При преподаването на обектно ориентирана парадигма на програмиране независимо от етапа на реализация на процеса на създаване на софтуер, е допустимо да се изграждат и моделират обектни модели с различна степен на сложност и абстракция.

Извод №18: Предвид технологичните предимства, пазарните тенденции и възможностите за универсалност на моделиране с високо (различно) ниво на абстракция е безспорно, че обектно ориентираната парадигма е за предпочитане пред алтернативните стилове на разработване на софтуер.

Извод №19: В софтуерното инженерство и индустрия през последните две десетилетия се предпочитат прилагането на гъвкави методологии за реализация на софтуерни процеси, пред дисциплинираните, детайлно документирани и стандартизирани „традиционни“ модели. За целите на обучението по софтуерни технологии обаче подходите, техниките и практиките на „традиционните“ модели с детайлни и ясно теоретизирани концепции имат място и могат да се използват като източник на тематично учебно съдържание.

Извод №21: Масовото приложение на гъвкавите методологии в софтуерната индустрия и слабото тяхно изучаване във висшето образование изискват търсенето на решения и „налагат“ предприемането на стъпки, които да доведат до синхронизация между методологии за създаване на софтуер и образователни методи и практики на тяхното преподаване.

Извод №24: Планирането, управлението и реализацията на софтуерни проекти разкрива концептуалната перспектива на разработване на софтуер.

Извод №25: Моделирането, създаването и приложението на софтуерни процеси чрез съвременна гъвкава методология разкрива софтуерната перспектива на разработване на софтуер.

Извод №29: За провеждането на студентски софтуерни практики чрез т. нар. проектна форма на обучение е необходимо да съществува орган, който да изпълнява функции на стимулатор, регулатор, организатор и координатор на сътрудничество между представители на науката, образованието и бизнеса.

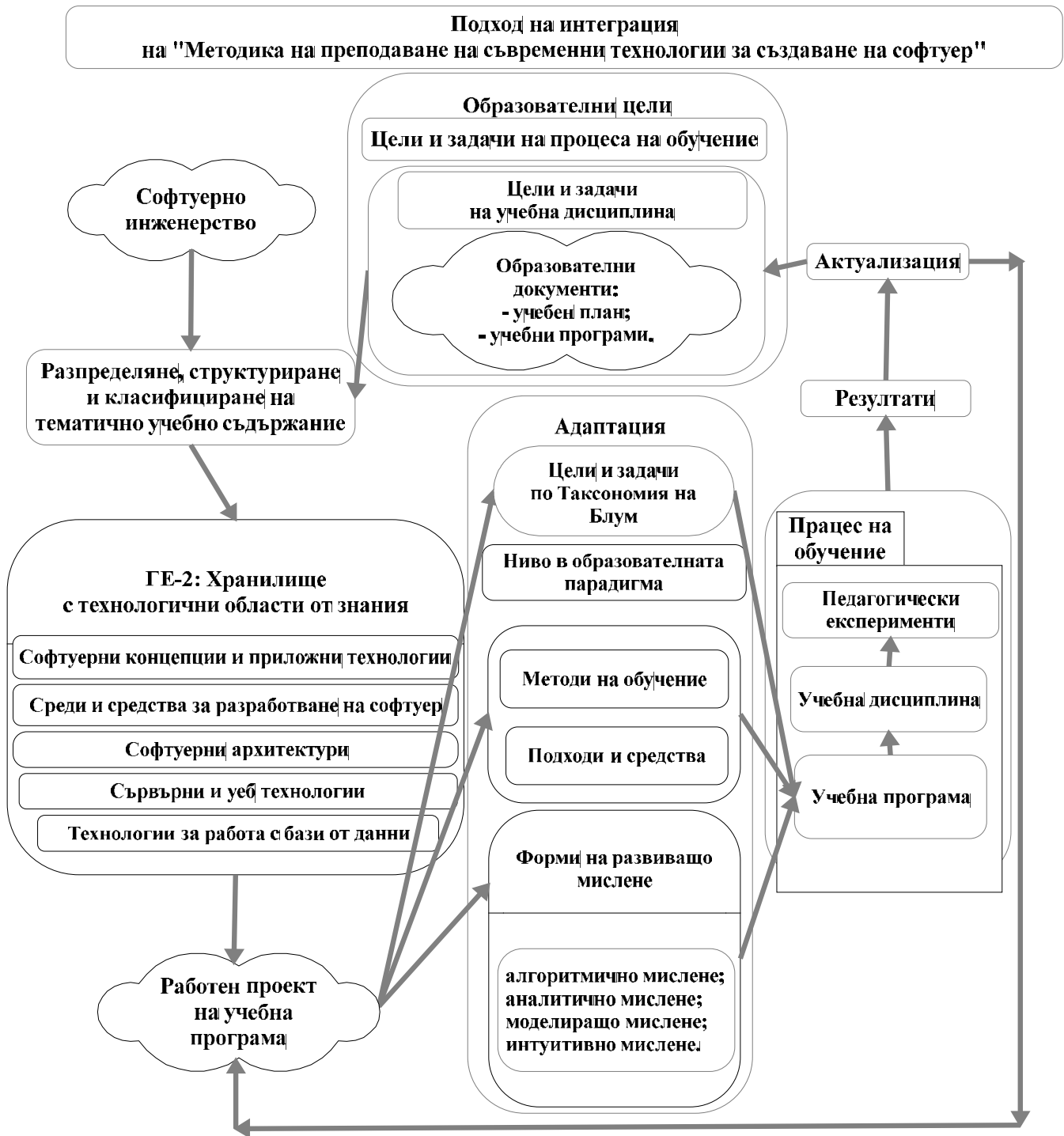
Извод: №30: При осъществяването на проектна форма на обучение чрез провеждане на студентски софтуерни практики в професионална бизнес работна среда и в условия на сътрудничество между наука, образование и бизнес студентът освен тясно професионалната специализация трябва да придобива социални знания, умения и компетенции за работа в екип, организираност, отговорност към работата, за провеждане на формална и неформална комуникация, креативност, вземане на решения при непредвидени обстоятелства и др.

ГЛАВА 3. Внедряване на Методика на преподаване на съвременни технологии за създаване на софтуер в процеса на обучение.

Реализацията на методиката на преподаване на съвременни технологии за създаване на софтуер се характеризира с два процеса: разработване и внедряване. Начинът на внедряване на методиката

разкриваме чрез разработването на **Подход за интегриране на Методика на преподаване на съвременни технологии за създаване на софтуер**.

3.1. Подход на интегриране на методиката.



Фиг. №5: Концептуален модел на подход на интеграция на Методика на преподаване на съвременни технологии за създаване на софтуер.

Концепцията за реализиране на *Методика на преподаване на съвременни технологии за създаване на софтуер*, разгледана изолирано извън въпроса за нейното приложение в процеса на обучение, притежава единствено научно-теоретичен характер. За да придобие теорията приложен методически характер, трябва тя да е част от образователната практика на обучението по софтуерни технологии. Привеждането на *Методиката* от „теория“ към „практика“ извършваме чрез подход на интеграция. В контекста на концепцията за реализация на методиката и от гледна точка на нейната структура това означава видът на *общия йерархичен интеграционен модел* да придобие форма, подходяща за внедряване в процеса на обучение. Тази **трансформация** на *общия йерархичен модел* в *подход за интеграция на методиката* ясно може да се види, като се съпоставят концептуалните графики на модела и подхода (*виж фиг. №4 и фиг. №5*). Разработването на две структури: *общ йерархичен модел* и *подход за интеграция на методиката* е обусловено от необходимостта на моделиране на два различни процеса. Чрез общия йерархичен модел се моделира процесът на извличане на теория от софтуерното инженерство и привеждането ѝ в подходящ вид за висшето образование, докато чрез подхода на интеграция на методиката се моделира процесът на обучение.

Подходът за интеграция на *Методиката на преподаване* притежава две нива на приложение: 1. Процес на обучение по учебен план на специалност; и 2. Процес на обучение по учебна програма на дисциплина. В този смисъл е важно да уточним, че приложението на подхода на интеграция, в частта си обучение по учебна програма, е реализиран на практика, докато за обучението по учебен план на специалност решенията не са приведени в практиката, а са останали развити в научно-теоретичен аспект.

Подходът на интеграция на *Методика на преподаване на съвременни технологии за създаване на софтуер* е приложен за разработването на пет нови за процеса на обучение дисциплини, като две от тях изцяло са разработени чрез него, а за останалите три той е използван частично.

3.2. Методи и експерименти.

От използваните методи и извършени експерименти особено внимание заслужават: изготвянето на оценка за качеството на проведения процес на обучение; провеждането на активното участващо наблюдение – тип интервю; и провеждането на педагогически експеримент за работа в екип чрез прилагане на метода „Групова дискусия“. Последните, активно наблюдение и педагогически експеримент са обособени в отделни приложения на дисертационния труд, съответно Приложение№1 и Приложение№3.

Системата за окачествяване на проведеното обучение е съобразена със спецификата на областта и общата тенденция на девалвация на образователната оценка. Тя е съставена от три компонента:

- 1) Категории за оценка;
- 2) Методи за диагностика;
- 3) Резултат от окачествяване на обучението.

В *Методика на преподаване на съвременни технологии за създаване на софтуер системата за окачествяване на процеса на обучение* следи за показатели в четири категории:

- 1) Оценката на тематично учебно съдържание;
- 2) Категория (ниво) в Таксономия на Блум;
- 3) Резултати от самостоятелно и екипно свършена работа;
- 4) Образователната оценка, в това число отделен случай и среден успех.

При събирането на данни за проведеното обучение може да се използва произволен диагностичен метод стига информацията да се групира по посочените категории на системата за окачествяване. За събраната информация се прави теоретичен анализ, в следствие на който се изготвя оценка за проведеното обучение. Крайният резултат на окачествяване е съставен от три конкретни оценки:

- 1) Степен на актуалност на проведеното обучение;
- 2) Степен на усвоимост на учебното съдържание;
- 3) Степен на необходимост от актуализация.

ЗАКЛЮЧЕНИЕ

В заключение представяме резултатите от труда. Те са разделени на две категории – приноси и постижения. Постиженията са по-значимите резултати от изследванията, които обединени по групи формират научен, научно-приложен или приложен принос. С други думи един научен принос е съставен от няколко постижения, които са резултат на различни изследвания.

Резултати – научни приноси и постижения.

Характерът, видът и мащабът на проведените изследвания доведоха до обособяване на основи на теория наречена *Методика на преподаване на съвременни технологии за създаване на софтуер* или назована по-общо *Теория на методика на преподаване на съвременни софтуерни технологии*. В аксиоматичен смисъл основите на теорията са поставени от две концепции в резултат на чието приложение се построиха два концептуални модела и формулираха пет принципа, девет тези, тридесет и един извода, шест правила, две изисквания и десет критерия. От тази гледна точка, близките по смисъл и значение научни твърдения и техния анализ и интерпретация представляват постижения на дисертационния труд. Обединението на постиженията по групи съставят научните научно-приложните и приложни приноси. Те са представени чрез **Списък №4: Научни, научно-приложни и приложни приноси** и **Таблица № 5: Класификационна матрица на приносите**. Същевременно обаче, за да се разкрие по детайлно съдържанието на *Теория на методика на преподаване на съвременни софтуерни технологии* разширяваме Списък №4, като добавяме групирани постижения, които съставят отделните научни, научно приложни и приложни приноси.

Списък №4 – Научни, научно-приложни и приложни приноси на дисертационния труд:

1. Изготвяне на оценка за степента на развитие на методика на обучението по софтуерни технологии.
2. Разработване и приложение на: Концепция за провеждане на качествено педагогическо изследване по методика на обучението в областта на софтуерните технологии.
3. Разработване и приложение на: Концепция за реализиране на *Методика на преподаване на съвременни технологии за създаване на софтуер* в професионално направление „Информатика и компютърни науки“ чрез провеждане на качествени педагогически изследвания.
4. Построяване на *Общ йерархичен интеграционен модел* за структуриране, разпределяне, формализиране, адаптиране и актуализиране на тематично учебно съдържание, подходи, методи и средства на обучение по софтуерни технологии.
5. Съставяне на подход на интеграция на *Методика на преподаване на съвременни технологии за създаване на софтуер*.
6. Разработване на изследователски метод: *активно, участващо наблюдение – тип интервю* за провеждане на качествени педагогически изследвания по методика на обучението в областта на софтуерните технологии.

Класификационна матрица на приносите.

В Таблица №5 – *Класификационна матрица на приносите* са посочени следните характеристики: видът на приноса, разрешената задача, точката в труда, научната формулировка и публикацията. В тези взаимовръзки може по-задълбочено да се вникне когато приносите се разгледат като съставка от група постижения.

Таблица №5: Класификационна матрица на приносите.

Принос	Характеристика	Задача/ подзадача	Точка/ подточка	Научна формулировка	Публикация
1	научен	1.2., 2.1., 2.2., 2.3.	1.1., 1.2., 1.3., 1.4.	Т№1, Т№2, Т№3, Т№4, Т№5, Т№6, Т№7, Т№8, И№1, И№2, И№3, И№4, №5, И№6, И№7,	1

				И№8, И№9, И№10, И№11, И№12, И№13, И14, И№15, И№16, И№17, И№18.	
2	научно-приложен	3.1.	2.1.1.	П№1, П№2, Из№1, Из№2, К№1, К№2, К№3, К№4, К№5, К№6.	
3	научно-приложен	3.1., 4.3.	2.1.2.	Табл.№4.	
4	научен	3.3., 4.2.	2.2., 2.3., 2.4., 2.5., 2.6.	Мфиг.№20, Мфиг.№12, И№19, И№20, И№21, И№22, И№28, И№29, И№30, И№31, С№1, С№2, С№3, С№4.	2, 3, 4, 5
5	приложен	4.1, 4.2.	2.2., 3.1., 3.2., 3.3.	Мфиг.№21, Т№9, ПП№1, ПП№2, ПП№3, Пр.П№1, Пр.П№2, Пр.П№3, Пр.П№4, Пр.П№5, Пр.П№6.	2
6	научно-приложен	4.1., 4.4.	3.1., 3.2.		

Легенда: П – принцип; ПП – принцип на подход; Пр.П – правило на подход; Т – теза; И – извод; Из. – изискване; К – критерий; М – модел; С – списък, Табл. – таблица.

Разширен списък №4 – Научни, научно-приложни и приложни приноси по групи от постижения:

1. Изготвяне на оценка за степента на развитие на методика на обучението по софтуерни технологии:

- Проследяване на произхода и развитието на образованието и обучението по информатика и софтуерни технологии в професионално направление „Информатика и компютърни науки“;
- Изготвяне на количествената и качествена оценка на състоянието на обучението по информатика и софтуерни технологии в седем от водещите университети на страната посредством анализ на 484 учебни програми в девет специалности, актуални за учебната 2014/2015 година;
- Извеждане на тенденциите на развитие при обектно ориентирания подход на разработка на софтуер – смяна на парадигмите на разработване и преподаване.

2. Разработване и приложение на: Концепция за провеждане на качествено педагогическо изследване по методика на обучението в областта на софтуерните технологии:

- Изготвяне на сравнителен анализ между количествените и качествените изследвания;
- Разработване на система от принципи, изисквания и критерии за провеждане на качествено изследване по методика на обучението.

3. Разработване и приложение на: Концепция за реализиране на *Методика на преподаване на съвременни технологии за създаване на софтуер* в професионално направление „Информатика и компютърни науки“ чрез провеждане на качествени педагогически изследвания:

- Прилагане на подходи за извличане, класифициране и структуриране на тематично учебно съдържание, педагогически методи и средства на преподаване;
- Прилагане на концепцията за построяване на общ йерархичен интеграционен модел;
- Разработване на критерии, показатели и параметри за оценка на резултатите.

4. Построяване на *Общ йерархичен интеграционен модел* за структуриране, разпределяне, формализиране, адаптиране и актуализиране на тематично учебно съдържание, подходи, методи и средства на обучение по софтуерни технологии:

- Обобщаване на подходящи за приложение в обучението: *организационни и управленски подходи и конструктивни техники, практики и средства* използвани за създаване на софтуер при гъвкавите методологии УП, Скрум и ЕП; *подходи, техники и практики* използвани за

създаване на софтуер при традиционните софтуерни модели; образователни *подходи* и *средства* за адаптиране на методи на преподаване в обучението по софтуерни технологии;

- Разкриване на образователната парадигма на обучение по софтуерни технологии;
- Разработване на разширен модел на еволюция на гъвкавите методологии;
- Разработване на подход на обучение – студентски софтуерни практики;
- Категоризиране на формите на развиващо мислене в обучението по софтуерни технологии.

5. Съставяне на подход на интеграция на *Методика на преподаване на съвременни технологии за създаване на софтуер*:

- Разработване на подход за съставяне на учебна програма за преподаване и изучаване на обектно ориентирана парадигма;
- Изготвяне на системата за качествяване на процеса на обучение;
- Разработване на: Приложение №3: Концепция за провеждане на педагогически експеримент за работа в екип чрез прилагане на метода „Групова дискусия“ (Фокус метод).

6. Разработване на изследователски метод: *активно, участващо наблюдение – тип интервю* за провеждане на качествени педагогически изследвания по методика на обучението в областта на софтуерните технологии:

- Методически експеримент – Приложение №1: Концепция за провеждане на активно участващо наблюдение – тип интервю.

Чрез изведените, структурирани и обединени в обща теория научни твърдения основната хипотеза на труда, че предложената *Методика на преподаване на съвременни технологии за създаване на софтуер* може да повиши качеството на обучението и неговия потенциал, се потвърждава.

Сложност и трудности на изследванията.

Основните трудности при преведените изследвания се свързват с техния мащаб. На практика при подбора на подходи, техники, практики и др. способи за създаване на софтуер се изследва цялата област на софтуерното инженерство. Двата основни метода, които използвахме при подбора са анализ на специализирана литература и провеждане на т. нар. активно участващо наблюдение – тип интервю. Вторият метод се оказва особено трудоемък, тъй като освен провеждането на самите интервюта, добитите чрез него мнения, разбирания, позиции и др. водеха до „отклонения от плана“, преоценка на вече формулирани тези и изводи, по-задълбочено вникване в намерените решения, изправяне пред нови, нерешени проблеми от науката и т.н. Друга основна трудност която срещнахме е свързана с вида на изследването. При качествените педагогически изследвания съществуват редица нерешени проблеми. Затова в хода на проучванията се наложи да разработим концепция за провеждане на качествено педагогическо изследване в областта на методика на обучението по софтуерни технологии и едва след това да пристъпим по същество към самите изследвания. Друга, трудност свързана с обема на работата представляваше изготвянето на оценката за състоянието на обучението по информатика и софтуерни технологии при анализа на 484 учебни програми.

Сред трудностите трябва да отделим **сложността на протичане** на самото изследване като приложение на концепция за реализиране на *Методика на преподаване на съвременни технологии за създаване на софтуер* чрез провеждане на качествено педагогическо изследване. Както се вижда от Таблица №5 – *Класификационната матрица на приносите* и Разширен списък №4 – *Научни, научно приложни и приложни приноси по групи от постижения* в резултат на проведените изследвания се създаде сложна система от концепции, принципи, модели, тези, изводи, правила, списъци, изисквания и критерии, която носи характер на основи на теория на методика на обучението по софтуерни технологии.

Приложение и перспективи пред *Методика на преподаване на съвременни технологии за създаване на софтуер*.

Методиката е използвана за съставянето на пет учебни дисциплини. Изцяло чрез нея са разработени два избираеми курса: „Анализ, проектиране и разработване на софтуерни приложения“ и „Проектиране, стилизиране и изграждане на уеб страници“. Също така чрез нея частично е разработено учебното съдържание по избираемата дисциплина „Въведение в обектно ориентираното програмиране с Java“ и редовните дисциплини „Увод в уеб програмирането“ и „Уеб дизайн“.

Перспективите пред *Методика на преподаване на съвременни технологии за създаване на софтуер* са свързани с нейното приложение. Двете ѝ най-съществени достойнства са *Общ йерархичен интеграционен модел за структуриране, разпределяне, формализиране, адаптиране и актуализиране на тематично учебно съдържание, подходи, методи и средства на обучение и Концептуален модел на подхода на интеграция*. Предназначението на моделите е да се използват за съставяне на нови учебни програми на независими една от друга учебни дисциплини или на взаимосвързани дисциплини, които предполагат надграждане на знания и умения. Общият модел обаче е приложим при разработването на учебни планове, а не само на учебни програми. Освен това отделните елементи на двата основни модела на методиката могат да се използват за окачествяване на съществуващи учебни планове. В случая под окачествяване на учебен план се има предвид разработването на общ технологичен речник на изучаваните технологии и определяне на връзките между учебните дисциплини.

Като перспектива пред методиката трябва да отбележим възможностите, които тя предоставя за разработване на учебен план на нова магистърска програма за обучение на ръководители на софтуерни проекти. Както е добре известно, към настоящия момент на територията на страната не съществува специалност, в която да се извършва обучение на софтуерни мениджъри (project managers).

Сред приложението и перспективите на *Методика на преподаване на съвременни технологии за създаване на софтуер* трябва да отделим **гъвкавостта и устойчивостта на методиката**. Това може да видим от образователната парадигма на обучението по софтуерни технологии в общия йерархичен интеграционен модел. Така например, на ниво три имаме обучение чрез студентски софтуерни практики, чиято форма през последните три години започва да навлиза в обучението, макар и все още да не е известно какъв вид ще придобие; На ниво две имаме обучение чрез обектно ориентирани технологии, за чието перспектива като приложност и употреба може да се прогнозира устойчивост в порядъка на едно-две десетилетия; На ниво едно имаме обучение при съвременните гъвкави методологии за разработка на софтуер, за чието приложение може да се направи смела прогноза на масова използваемост за повече от две, дори три десетилетия; На ниво нула имаме обучение чрез софтуерни проекти, чиято актуалност ще е в сила докато софтуерните технологии е масово привлекателна научна и образователна област.

АПРОБАЦИЯ

Части от някои изследвания на дисертационния труд са изготвени в изпълнение на задачи по следните проекти:

1. Научен проект с номер на договор НИ11-ФМИ-004/30.05.2011 към фонд „Научни изследвания“ при ПУ „Паисий Хилендарски“ на тема „Разработка и приложение на иновативни ИКТ за провеждане на качествени конкурентноспособни научни изследвания и цялостно осъвременяване процеса на обучение във ФМИ“.

2. Научен проект с номер на договор НИ13-ФМИ-002/19.03.2013 към фонд „Научни изследвания“ при ПУ „Паисий Хилендарски“ на тема: „Интеграция на ИТ в научните изследвания по математика, информатика и педагогика на обучението“.

3. Научен проект с номер на договор НИ15-ФМИИТ-004/24.04.2015 към фонд „Научни изследвания“ при ПУ „Паисий Хилендарски“ на тема „Иновативни, фундаментални и приложни научни изследвания по компютърни науки, математика и педагогика на обучението“.

4. Научен проект с номер на договор ИТ15-ФМИИТ-004/24.04.2015 към фонд „Научни изследвания“ при ПУ „Паисий Хилендарски“ на тема: „Изследвания в областта на иновационни ИКТ с ориентация към бизнеса и обучението“.

Части от някои изследвания на дисертационния труд са докладвани на следните научни сесии и конференции:

1. Научно-практически форум „Иновации в обучението и познавателното развитие“, 2-4.09.2015, гр. Бургас.
2. XXII Международна научна конференция на Съюза на учените в Стара Загора, 7-8 юни 2012.
3. Международна конференция „Взаимодействието теория – практика: Ключови проблеми и решения“, 24-25 юни 2011, Бургас.
4. Anniversary International Conference REMIA 2010, December 10-12, 2010, Plovdiv.

Публикации по темата на дисертационния труд:

1. Христов, Хр, Хр. Крушков, „Състояние на обучението по информатика и софтуерни технологии“, Образование и технологии, Бургас, година VI, кн.6, 2015г., стр. 388. http://itlearning.bg.com/magazines/Spisanie2015/resources/spisanie_e_book_2015.pdf
2. Христов, Хр, Хр. Крушков, „Архитектура „Модел-изглед-контролер“ в помощ на преподаването на уеб-технологии“, Математика и информатика, София, година LVI, кн. 4, 2013г., стр. 368-383.
3. Христов, Хр. А. Христова, Роля и място на програмирането в обучението по информатика, като част от съвременните методологии за създаване на софтуер, Наука и технологии, том II, 2012, <http://journal.sustz.com/VolumeII/Number3/Papers/HristoHristov2.pdf>
4. Христов, Хр. А. Христова, „Модели и гъвкави методи в обучението по компютърни науки“, Наука и технологии, том II, 2012, <http://journal.sustz.com/VolumeII/Number3/Papers/HristoHristov1.pdf>
5. Христов, Хр., „Трудности и решения при смяна на парадигмата. Преподаване на обектноориентиран анализ, дизайн и програмиране, Доклади на международна конференция“, Взаимодействието теория - практика: Ключови проблеми и решения“, т. III, Бургас, 24-25 юни 2011, стр. 303-310.
6. Hristov, H. Review and Outlooks of the Means for Visualization of Syntax Semantics and Source Code. Procedural and Object-oriented Paradigm – Differences, Proceeding of the Anniversary International Conference REMIA, Plovdiv, December 10-12, pp. 443-451, 2010.

БЛАГОДАРНОСТИ

Изказвам сърдечна благодарност на научния си ръководител доц. д-р. Христо Крушков за проявената подкрепа по време на работата върху дисертацията и на проф. д-р. Коста Гъров за оказаните професионална помощ и съдействие.

Библиография

- [1] Андреев, М., Дидактика, ДИ „Народна просвета“ С., 1981.
- [2] Андреев, М., Процесът на обучението. Дидактика, УИ "Св. Климент Охридски", С., 1996.
- [3] Бижков, Г., В., Краевски, Методология и методи на педагогическите изследвания, шесто издание, УИ „Св. Климент Охридски“, С., 2007.
- [4] Браншови организации в сферата на ИКТ, *Стратегически изисквания на софтуерната индустрия за реформа на образователната система*, БАСКОМ, версия 1.0, София, 2012.
- [5] Бърнев, П., Информационното общество и преходът към него, Асоциация „Развитие на информационното общество“, Пловдив, 2008., стр. 30.
- [6] Георгиева, А., Н., Николов, Н., Николова, Л. Георгиев, Методи на обучение във висшите учебни заведения, Годишник на минно-геоложкия университет “Св. Иван Рилски”, Том 50, част IV, Хуманитарни и стопански науки, 2007.

- [7] Георгиева, Ц., Информационно-комуникационните технологии в образованието - парадоксалният ефект на стимула, приближаващ ни към реалната среда, Сборник доклади на Национална конференция "Образованието в информационното общество" Пловдив, АРИО, 31 май - 01 юни, 2012, стр.309-314.
- [8] Гроздев, С., Т., Чехларова, Т., Терзиева, За необходимостта от развитие на алгоритмично мислене в обучението по информатика, Сборник доклади на Национална конференция "Образованието в информационното общество", Пловдив, АРИО, 27-28 май 2010, стр.102-108.
- [9] Гъров, К., Някои методически аспекти на обучението по информатика и информационни технологии., Монография, УИ „Паисий Хилендарски“, Пловдив, 2013.
- [10] Донева, Р., Е., Сомова, Обектно ориентирано програмиране., УИ, П., 2000.
- [11] Дурева-Тупарова, Д., Г., Тупаров, За подготовката на студентите - бъдещи учители по информатика и информационни технологии, Сборник доклади на Национална конференция „Образованието в информационното общество“, Пловдив, АРИО, 26-27 май 2011, стр.337-341.
- [12] Дурева-Тупарова, Д., Проблеми на методиката на обучение по информатика и ИТ, УИ Югозападен Университет „Неофит Рилски“, Благоевград, 2003.
- [13] Захариев, З., Информационният свят и образование – предимства и рискове., Списание „Наука“, бр. 1, 2013, стр.26.
- [14] Захариев, З., П., Балкански, Ж., Миланов, Т., Чавгова, В., Митев, И., Гарвалов, Г., Дерменджиева, Глобализацията между двете хилядолетия, Гражданско образование - част I, Национален институт по образование, София, 2001, стр. 9.
- [15] Иванов, И., Педагогическа диагностика, УИ „Еп. Константин Преславски“, Шумен, 2006.
- [16] Иванов, И., Теории за образованието, УИ „Еп. Константин Преславски“, Шумен, 2004.
- [17] Крушков, Хр., М., Крушкова, Д., Крушкова, Как да подобрим обучението по програмиране, Сборник доклади на Национална конференция “Образованието в информационното общество“, Пловдив, АРИО, 27-28 май 2010, стр.110-116.
- [18] Крушков, Хр., М., Крушкова, В., Атанасов, М., Крушкова, Компютърна система за обучение по програмиране, Сборник доклади на 39-та Пролетна конференция на СМБ, Математика и математическо образование, Албена, 6 - 10 април, 2010, стр. 354-358.
- [19] Крушков, Хр., М. Крушкова, Д. Коленцов, Д. Крушкова, Й. Стойнова, "Разработване на интелигентни компютърни методи и средства за обучение", Научни трудове на СУБ, т.VII, ISSN 1311-9419, Пловдив, 05-06.11. 2009 г., стр.412-415.
- [20] Крушкова, М., Методика за активно обучение по програмиране чрез използването на ИКТ, Автореферат на дисертационен труд за присъждане на образователна и научна степен „доктор“, УИ „Паисий Хилендарски“, Пловдив, 2014.й
- [21] Манева, Н., К., Манев, Тристранно партньорство в обучението по софтуерни технологии - предизвикателства и резултати, Сборник доклади на Национална конференция „Образованието в информационното общество“, Пловдив, АРИО, 31 май-01 юни 2012, стр.205-215.
- [22] Манева, Н., Обучението по софтуерни технологии: очаквания за криза или криза на очакванията, Сборник доклади на Национална конференция „Образованието в информационното общество“, Пловдив, АРИО, 26-27 май 2011, стр. 54-63.
- [23] Орозова, Д., Някои проблеми на обучението по информатика във висшето училище, Сборник доклади на Национална конференция „Образованието в информационното общество“, Пловдив, АРИО, 13-14 октомври 2006, стр.44-05.
- [24] Официален сайт на Бургаски свободен университет, ЦИТН, <http://www.bfu.bg/index.php?q=node/95>, [последно посещение 24.10.2014].
- [25] Официален сайт на Великотърновски университет, ФМИ, <http://www.uni-vt.bg/1/Default.aspx?zid=164>, [последно посещение: 29.10.2014].
- [26] Официален сайт на МОН, Национална програма - ИКТ в училище, <http://internet.mon.bg/ikt/?m=1>, [последно посещение: 11.04.2014].

- [26] Официален сайт на Нов български университет, <http://www.nbu.bg/entrance.php>, [последно посещение: 24.10.2014].
- [27] Официален сайт на Шуменски университет, ФМИ, <http://info.fmi.shu-bg.net/index.php>, [последно посещение: 29.10.2014].
- [28] Официален сайт на Югозападен университет "Неофит Рилски", Природо-математически факултет <http://www.swu.bg/university-profile/faculties-and-colleges/mathematics-and-natural-sciences.aspx>, [последно посещение: 28.10.2014].
- [29] Павлова, И., Информационните технологии в образователния процес: еволюция към ново качество на образованието, Сборник доклади на Национална конференция „Образованието в информационното общество“, Пловдив, АРИО, 13-14 октомври 2006, стр. 161-164.
- [30] Петров, П., Дидактика. Вѳда Словена, С., 1994
- [31] Сандалски, М., А., Стоянова, Среда за обучение по софтуерни технологии, Сборник доклади на Национална конференция "Образованието в информационното общество", Пловдив, АРИО, 26-27 май 2011, стр. 064-072.
- [32] Специализиран сайт на ФМИ при ПУ, <http://fmi-plovdiv.org/>, [последно посещение: 28.10.2014].
- [33] Специализиран сайт на ФМИ при СУ, <http://www.fmi.uni-sofia.bg/>, [последно посещение: 28.10.2014].
- [34] Старибратов, И., Компютърът - новият администратор в образованието, Сборник доклади на Национална конференция, „Образованието в информационното общество“, Пловдив, АРИО, 27-28 май 2010, стр.193-199.
- [35] Старибратов, И., Е., Ангелова, Методически подходи за обучение чрез използване на електронни учебни ресурси, Сборник доклади на Национална конференция „Образованието в информационното общество“, Пловдив, АРИО, 26-27 май 2011, стр.329-336.
- [36] Стоименова, В., Обучението по вероятности и статистика за педагогическите специалности във Факултета по математика и информатика на Софийски университет, Доклади на Четиридесет и втора Пролетна конференция на Съюза на математиците в България, Боровец, 2-6 април, 2013., стр.437-446.
- [37] Теодосиев, Т., Обучението по програмиране на новацие - защо и как?, Сборник доклади на Национална конференция „Образованието в информационното общество“, Пловдив, АРИО, 31 май-01 юни 2012, стр.195-204.
- [38] Теодосиев, Т., Проблеми на обучението по програмиране, Сборник доклади на Национална конференция „Образованието в информационното общество“, Пловдив, АРИО, 13-14 октомври 2006, стр. 51-56.
- [39] Терзиева, Т., За усвояването на алгоритми в обучението по програмиране, Сборник доклади на Национална конференция „Образованието в информационното общество“, Пловдив, АРИО, 27-28 май 2010, стр.223-230.
- [40] Терзиева, Т., Развитие на алгоритмично мислене в обучението по информатика, Автореферат на дисертационен труд за присъждане на образователна и научна степен „доктор“, УИ „Паисий Хилендарски“, Пловдив, 2012.
- [41] Тодорина, Д., Съдържание на обучението, Тема от учебник по педагогика от българо-руски колектив, ЮЗУ „Н. Рилски“ (България) - ЕГУ „И. А. Бунин“ (Русия), 2010.
- [42] Тодорова, Е., Рефлексията в обучението по информационни технологии, Автореферат на дисертационен труд за присъждане на образователна и научна степен „доктор“, УИ „Паисий Хилендарски“, Пловдив, 2014.
- [43] Христева, К., С., Влдова, Използване на информационни и комуникационни технологии в обучението по български език. Запетая в сложното съставно изречение, Сборник доклади на Национална конференция „Образованието в информационното общество“, Пловдив, АРИО, 13-14 октомври 2006, стр. 114-119.
- [44] Христов, Хр. А. Христова, Роля и място на програмирането в обучението по информатика, като част от съвременните методологии за създаване на софтуер, Наука и технологии, том II, 2012.

- [45] Христов, Хр., Трудности и решения при смяна на парадигмата. Преподаване на обектноориентиран анализ, дизайн и програмиране, Доклади на международна конференция „Взаимодействието теория - практика: Ключови проблеми и решения”, т. III, Бургас, 24-25 юни 2011, стр. 303-310.
- [46] 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.
- [47] Шестдесет години БАН-ИМИ основан 1947, БАН, С., 2007.
- [48] Barney G., A., Strauss, The Discovery of Grounded Theory: Strategies for Qualitative Research, Aldine Transaction, December 31, 1999.
- [49] Boehm, B., R., Turner, Management challenges to implementing agile processes in traditional development organizations. IEEE Software, vol.22, Issue 5, 2005, p.30 - 39.
- [50] Crawford, S., The Origin and Development of a Concept: The Information Society, Bull. Med. Libr. Assoc., vol. 71(4), Oct.1983, pp. 380-385.
- [51] Filstead, W., Qualitative Methodology: Firsthand Involvement with the Social World, Markham Publishing Company, Chicago, 1971.
- [52] Hristov, H. Review and Outlooks of the Means for Visualization of Syntax Semantics and Source Code. Procedural and Object-oriented Paradigm – Differences, Proceeding of the Anniversary International Conference REMIA, Plovdiv, December 10-12, pp. 443-451, 2010.
- [53] Jones, C., Software Engineering Best Practices, McGraw Hill, 2010.
- [54] Knuth, D., Algorithmic thinking and mathematical thinking, The American Mathematical Monthly, Vol. 92, No. 3, Mar., 1985, pp. 170-181.
- [55] Maneva, N., Comparative Analysis: A Feasible Software Engineering Method. Serdica Journal of Computing, vol.1, No 1, 2007, pp.1-12.
- [56] McLuhan, M., L. Lapham, Understanding Media: The Extensions of Man, MIT Press, 1994.
- [57] Naur, P., B., Randell, (Eds.). Software Engineering: Report of a conference sponsored by the NATO Science Committee, Garmisch, Germany, 7-11 Oct. 1968, Brussels, Scientific Affairs Division, NATO (1969) pp. 231.
- [58] Official site of: Manifesto for Agile Software Development, 2001, <http://agilemanifesto.org/>, [last visit of 08.12.2015].
- [59] Project Management Institute, A Guide to the Project Management Body of Knowledge (PMBOK® Guide), 4th Edition, ISBN 978-1-933890-51-7, 2008.
- [60] Siemens, G., Connectivism: A learning Theory for the Digital Age, International Journal of instructional technology and distance learning, January 2005, Vol. 2, Issue 1, pp.2-10.
- [61] Specialized website of Brian Randell dedicated to „The NATO Software Engineering Conferences“, <http://homepages.cs.ncl.ac.uk/brian.randell/NATO/index.html>, [last visit on 02.10.2015].
- [62] Sommerville, I., Software Engineering 9th edn., Addison-Wesley, Boston, MA, 2011.
- [63] Sorensen, R., A Comparison of Software Development Methodologies, CrossTalk, Jan, 1995, 12-18.
- [64] Wilson, S., The Use of Ethnographic Techniques in Educational Research, Review of Educational Research, Vol. 47, No.1, Winter 1977, p. 245-265.
- [65] Wysocki, R., K., Effective Project Management: Traditional, Agile, Extreme, 6th Edition, Wiley, 2011.