



БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ
ИНСТИТУТ ПО ИНФОРМАЦИОННИ И
КОМУНИКАЦИОННИ ТЕХНОЛОГИИ

Елисавета Димитрова Тричкова

ОПТИМИЗАЦИЯ НА ПРОЦЕСИ В ИНФОРМАЦИОННИ СИСТЕМИ

ДИСЕРТАЦИЯ

за придобиване на образователната и научна степен «доктор»

по специалност 02.21.10. „Приложение на принципите и методите на
кибернетиката в различни области на науката”

в професионално направление 5.3. „Компютърна и комуникационна
техника”

Научен ръководител:
/доц. дтн. Красимира Стоилова/

София, 2014 г.

Съдържание

Въведение.....	5
Глава първа. Информационни системи и процеси. Електронно обучение	8
1.1 Увод	8
1.2 Същност и основни понятия на Информационни системи (ИС).....	8
1.3 Основни понятия от софтуерните технологии	8
1.3.1 Определения	9
1.3.2 Цели	9
1.3.3 Критерии за качество на софтуерни продукти.....	10
1.3.3.1 Характеристики	10
1.3.3.2 Класификация на критериите за качество	11
1.3.3.3 Модели и стандарти за оценка на качеството	11
1.4 Разработване на информационна система	14
1.4.1 Същност и основни понятия	14
1.4.2 Модели на жизнения цикъл.....	18
1.5 Процеси в информационни системи.....	19
1.6 Методи за многокритериална оптимизация.....	22
1.6.1 Парето-оптимални решения	23
1.6.2 Методи на скаларизацията и компромисните решения	24
1.7 Web-базирани образователни информационни системи	27
1.8 Изводи	33
1.9 Цел и задачи на дисертационния труд	34
Глава втора: Синтез на процеси в информационни системи	36
2.1 Увод	36
2.2 Методология за проектиране и разработване на информационна система.....	36
2.2.1 Етапи.....	36
2.2.1.1 Анализ и дефиниция на изискванията към системата	38
2.2.1.2 Проектиране на системата.....	39
2.2.1.3 Разработване на системата	48
2.2.1.4 Тестване на системата.....	48
2.2.1.5 Внедряване на системата.....	49
2.2.1.6 Оценяване на системата.....	49
2.2.2 Описание на избора на технологии и средства за реализация.....	49

2.3 Възможност за прилагане на методи за многокритериална оптимизация към подход за оценка на качеството на система за Web-базирано обучение	51
2.3.1 Определяне на тегловни коефициенти чрез ранжиране на целевите параметри	52
2.4 Изводи	55
Глава трета: Разработване на информационна система за Web-базирано обучение	56
3.1 Увод.....	56
3.2 Реализация на системата.....	56
3.2.1 Дефиниране на изискванията към системата	57
3.2.2 Създаване на базата от данни.....	58
3.2.3 Проектиране на Web архитектурата на системата.....	60
3.2.4 Проектиране на системата като съвкупност от модули	61
3.2.4.1 Описание на потребителския интерфейс на системата	62
3.2.4.2 Описание на административен интерфейс на системата.....	72
3.3 Сравнение на разработената система с други български и чужди системи в областта на електронното обучение	77
3.4 Изводи.....	82
Глава четвърта: Оптимизиране на информационни процеси.....	84
4.1 Увод.....	84
4.2 Подход за оценка качеството и ефективността на системата за Web-базирано обучение	84
4.2.1 Етапи на реализация на подхода.....	84
4.2.2 Определяне на система от показатели за оценка на качеството и ефективността на системата	85
4.2.3 Определяне на тегловни коефициенти на показателите	86
4.2.4 Изчисляване на тегловните коефициенти на показателите	88
4.2.5 Графично представяне на резултатите	90
4.3 Приложение на предложения подход за количествена оценка на качеството и ефективността на системата	90
4.3.1 Подготовка и организация на изследването.....	90
4.3.2 Цел на изследването	91
4.4 Резултати от оценяването на системата.....	91
4.4.1 Резултати от приложение на предложения подход за количествена оценка на качеството на системата на база показатели по стандарт ISO 9126	91

4.4.2 Дефиниране на задачата за оптимизация на процесите на база софтуерни показатели по стандарт ISO 9126.....	98
4.4.3 Резултати от приложение на предложения подход за оценка на ефективността на системата на база дидактически показатели.....	104
4.4.4 Мнения и коментари	112
4.5 Изводи.....	113
Заклучение	115
Приноси на дисертационния труд	117
Бъдещи насоки за работа	118
Публикации, свързани с дисертацията.....	119
Научноизследователски проекти	120
Декларация за оригиналност на резултатите	121
Литература.....	122
Използвани съкращения.....	130
Приложение 1: Разпределение Z при ниво на значимост α.....	131
Приложение 2: Стойности на χ^2 при брой на степени на свобода ν и ниво на значимост α.....	132
Приложение 3: Анкетна карта (ISO 9126)	133
Приложение 4: Анкетна карта (дидактически показатели).....	138
Приложение 5: Степен на компетентност на експертите в областта на софтуерните технологии	140
Приложение 6: Степен на компетентност на преподаватели от институцията, използваща информационната система.....	141

Въведение

Цел на дисертационния труд е като се използват постиженията на софтуерните и образователни технологии да се проектира и реализира система за Web-базирано обучение; да се приложи подход за количествена оценка на качеството ѝ чрез провеждане на изследване, базирано на мнения на експерти и преподаватели в областта на информационните технологии и образованието с оглед определяне на потенциала за подобряване на нейните характеристики.

Актуалност на проблема

Информационните системи придобиват все по-голямо значение за съвременния начин на живот. Те се използват в най-различни професионални области като производството, енергетиката, компютърните технологии, финанси, телекомуникации, продажби, болници и фармацевтика, услуги, висше образование.

Информационна система (ИС) е интегриран набор от компоненти за събиране, съхранение, обработка и предаване на информация [30]. Проектирането и изграждането на ИС следва да удовлетворява наложени стандарти в съответната сфера. Качеството на софтуера е ключова характеристика, която обобщава резултата от взаимодействието на почти всички фактори, възникващи в процеса на разработване и създаване на програмите. Съставянето на количествена оценка на качеството на софтуерния продукт е важна не само за оценката и за сравнението на отделните програмни приложения, но и за ефективното управление и усъвършенстване на процесите в информационна система като цяло [100].

Тенденциите за развитие на електронното обучение и по-голямата достъпност на образованието като цяло засилват необходимостта от внедряване на ИС на всички нива на образователния процес и подтикват институциите да работят в тази посока [73]. Разработването на информационни системи, интегриращи в себе си компоненти за административно обслужване, комуникация, електронно обучение, кариерно развитие и др. могат значително да повишат качеството на предлаганите образователни услуги на всички нива [102].

Едни от най-често използваните системи в образованието са системите за управление на съдържанието. Тези системи са многофункционални и непрекъснато се развиват и усложняват. Проблемът как да се структурира Web базиран курс и да се представи съобразно избран модел, как да се съхранява в правилно структурирана Web-интегрирана база данни е изключително актуален.

Въпреки че почти всички автори на системи изтъкват универсалния характер на представените от тях системи, на практика съществуват специфични разлики. Практиката показва, че отделните Висши училища (ВУ) използват информационни системи изцяло съобразени със спецификата им, което оправдава и налага необходимостта от разработване на собствени системи, съобразно избран педагогически модел. Докато в техническите специалности, разпространена форма на

обучение и изпитване е разработването на проекти, то в медицинските дисциплини тази форма все още не се прилага, а е важно решаването на ситуационни задачи (клинични казуси) [38, 39]. Това определи в обхвата на дисертационния труд да се проектира и разработи система за управление на съдържанието и да се приложи подход за оценка на качеството и ефективността ѝ, като въз основа на получените резултати се оптимизират възможностите на самата система.

Основната работна задача, която си поставя дисертационния труд е да проектира и разработи система от модули за организиране и управление на информационни процеси, съставлящи Web-базирано обучение. С цел оптимизиране възможностите на работните процеси на системата, но без да се усложнява работата с нея, да се приложи подход за количествена оценка на качеството и ефективността на софтуерни продукти, за да се удовлетворят специфичните изисквания на потребителите на системата.

Целите и задачите на дисертацията са изложени в края на глава първа.

Структура на дисертацията

Дисертационният труд е разделен на увод, четири глави и заключение. Дисертацията съдържа 141 страници, 37 фигури и 45 таблици, 152 цитирани литературни източника и 6 приложения, списък на използваните съкращения. По дисертационния труд са публикувани 7 публикации, като 4 от тях са доклади от международни конференции. Изследванията са част от получените резултати на 5 научноизследователски проекта.

В първа глава са дадени основни понятия за Информационни системи и принципите на софтуерните технологии като средство за създаване на прототип на компютърна информационна система. Разгледани са стандартите и моделите за оценка на качество на софтуерните продукти и методи за многокритериална оптимизация. Представени са модели на жизнения цикъл на софтуерния процес и на някои съществуващи платформи и среди за електронно обучение. Направени са основни изводи на базата на проучените литературни източници и са формулирани целите и задачите на дисертацията.

Във втора глава е разгледана и мотивирана методология за проектиране и разработване на система за организиране и управление на Web-базирано обучение по медицински дисциплини. Определена е последователността на етапите, през които преминава разработването на системата. Обоснован е изборът на използваните техники и средства. Анализирани са изискванията към системата. Посочени са сценарии на действието и на работата на отделните модули и е посочен вариант за проектиране на моделите на процесите им. Разгледана е възможност за прилагане на методи за многокритериална оптимизация към подход за оценка на качеството и ефективността на система за Web-базирано обучение. Показано е, че с малки промени подходът може да се прилага за различни софтуерни продукти в областта на web-базираните приложения, като се отчетат особеностите на всяка от тях.

В трета глава е представена последователността от дейности, изпълнени при разработването на системата за web-базирано обучение. Описани са структурата и

функциите на основните модули на системата. Направено е сравнение на разработената система с други известни софтуерни платформи, като сравнението е представено чрез радарни диаграми.

В четвърта глава е описан и анализиран извършения експеримент по прилагане на предложения подход за количествена оценка на качеството и ефективността на Web-базирана система за обучение. В изследването взеха участие специалисти от четири фирми в областта на софтуерните и информационни технологии и два български университета, занимаващи се с електронно обучение. Обобщени са резултатите от изследването и са направени анализи по тях.

В заключението са описани различните дейности, извършени при изпълнението на отделните етапи. Включени са приносите на дисертационния труд, бъдещите насоки на работа, научните публикации по дисертацията, както и научно-изследователските проекти, по които е работено във връзка с изследванията по дисертационния труд.

Глава първа. Информационни системи и процеси.

Електронно обучение

1.1 Увод

В първа глава са дадени дефиниция и основни понятия за това какво представляват Информационните системи (ИС). Синтезирани са принципите на софтуерните технологии като средство за създаване на прототип на информационна система. Разгледани са стандартите и моделите за оценка на качеството на софтуерните продукти и методи за многокритериална оптимизация. Анализирани са моделите на жизнения цикъл на софтуерния процес, както и на съществуващи платформи и среди за електронно обучение.

1.2 Същност и основни понятия на Информационни системи (ИС)

В съвременното общество компютрите се утвърдиха като незаменим помощник във всички сфери на човешката дейност. Неимоверно много нарасна и значението на *информацията, която вече се третира като бизнес ресурс*. Това стана възможно, благодарение на компютърните информационни системи, които предоставят и основната среда за приложение на информационните технологии в практиката.

Чрез добро познаване на възможностите и тенденциите в развитието на компютърните информационни системи може да се повиши ефективността от работата на всички нива и във всички сфери на действие.

Според [89] създаването на една компютърна информационна система е свързано с приложното програмиране, както и с разнообразни други средства на съвременните информационни технологии. Приложното програмиране се прилага за решаване на определени задачи от различни сфери на съвременния начин на живот. Разработването на информационна система се извършва на определени етапи и по определени методи.

Най-общо казано, под информационна система се разбира система, предназначена за съхраняване, предаване и обработка на данни, с цел получаване на необходима за потребителя информация.

Дейностите, свързани с получаване, съхраняване, обработка и разпространяване на информация се наричат информационни дейности, а всеки процес, който обхваща такива дейности – информационен процес.

За създаване на прототип на една информационна система се използват принципите на софтуерното инженерство.

1.3 Основни понятия от софтуерните технологии

Терминът “Software Engineering” се е появил за пръв път на конференция на НАТО през януари 1969 г. У нас се използват термините: Софтуерни технологии (СТ) и Софтуерно инженерство (СИ).

1.3.1 Определения

Една от първите дефиниции за СТ е предложена от Наур още през 1969 г. на първата конференция по тези проблеми [97]:

Установяване и използване на здрави принципи за разработване с цел по икономичен начин да се произведе софтуер, който е надежден и функционира ефективно на реална машина.

Според дефиницията на IEEE софтуерните технологии представляват *систематичен подход към разработването, експлоатирането, съпровождането и изваждането от експлоатация на софтуера.*

Определението на Феърли от 1984 година гласи: *технологична и мениджърска дисциплина, занимаваща се систематично с производството и съпровождането на софтуерните продукти, които се разработват на време и на основата на точно определени разходи.*

По-нататък ще се придържаме към определението, дадено в [97]:

Софтуерните технологии представляват систематичен подход към разработването на качествен софтуер, както и към неговото предлагане на пазара, експлоатация и съпровождане.

1.3.2 Цели

Условието за качество на разработвания софтуер налага да се идентифицират основните му атрибути [43]:

- Всеки програмен продукт (ПП) следва да може да бъде лесно съпровождан. Това означава, че в него трябва да могат относително лесно да се внасят изменения и подобрения, както и лесно да се отстраняват грешки;
- Софтуерът трябва да бъде надежден. Това означава, че грешките в него трябва да са сведени до възможния минимум, а продуктът да изпълнява очакваните от потребителя функции;
- ПП трябва да е ефективен;
- Да предоставя удобен и лесен за усвояване потребителски интерфейс;
- Да има минимални разходи по разработката и особено при съпровождането му (минимална цена).

Софтуерната технология (СТ) е изградена от множество стъпки, които включват различни методи, средства и подходи.

1) Методите на СТ определят методически “КАК” се изгражда софтуер. Те включват: планиране и оценяване на проекта, анализ на изискванията за системата и софтуера, проектиране на структурата от данни, подходи за програмна архитектура и алгоритми, кодиране, тестване и поддържане. Те често въвеждат специални езиково-ориентирани или графични означения и множество от критерии за софтуерно качество [43].

2) Средствата на СТ осигуряват автоматизирана или полуавтоматизирана поддръжка на методите. Съществуват средства за поддръжка на всеки един метод. Когато средствата са интегрирани, така че информацията, създадена чрез едно средство може да се използва от друго, е налице система за разработване на софтуер, наречена CASE (Computer-Aided Software Engineering) [43].

3) Подходите на СТ са свързващото звено, което съединява методите и средствата, като гарантират разумна и навременна разработка на компютърния софтуер. Подходите определят последователността, в която ще се приложат методите, продукцията (документи, отчети, форми и т.н.), която се изисква, управлението, което помага за осигуряване на качеството и координиране на промените, и ключовите моменти, които позволяват на софтуерния менажер да следи и оцени развитието на проекта [43].

1.3.3 Критерии за качество на софтуерни продукти

Качеството на софтуера е комплексна характеристика, която обобщава ефекта от влиянието на множество разнообразни фактори върху процеса на разработване и реализиране на софтуерния продукт. Съставянето на количествена оценка за качеството на софтуерния продукт е важна не само за оценката и за сравнението на отделните програмни приложения, но и за ефективното управление и усъвършенстването на процесите в софтуерното инженерство [43, 50, 55, 99].

1.3.3.1 Характеристики

Качеството на софтуера се определя от няколко основни характеристики, като някои от тях са:

- Коректност – степен на съответствие между изискванията на потребителя и разработеното приложение.
- Надеждност – способността на софтуерното приложение да изпълнява предвидените функционални изисквания при евентуални проблеми, възникващи по време на използването му.
- Стабилност – една програмна система е стабилна, ако негативните последствия от обработката, от входните данни или от хардуера са обратно пропорционални на вероятността за появяване на грешки в системата.
- Машабируемост – възможността за разширяване или модифициране на един програмен продукт по отношение на определени параметри като например повече данни, други платформи, увеличение броя на потребителите и др.
- Преносимост – степен на независимост към хардуер или операционна среда.
- Възможност за поддръжка – възможност за отстраняването на грешки на едно софтуерно приложение по време на експлоатацията му и поддържането му в актуално състояние.

- Удобен за потребителя – възможност за усвояване и експлоатация на едно софтуерно приложение с минимум усилия, т.е. софтуер, който е лесен за изучаване, стабилен и адекватен за нуждите на потребителите.

1.3.3.2 Класификация на критериите за качество

Според [108] критериите за качество се разделят на две групи – външни и вътрешни. Към външните критерии за качество се включват характеристиките, видими за потребителя като коректност, надеждност, мащабируемост, стабилност, продуктивност, удобство за потребителя и съвместимост. Към вътрешните критерии за качество спадат характеристиките, видими за разработчика като възможност за поддръжка, читаемост, преносимост, възможност за повторно използване, модулиране, възможност за тестване.

Съществена роля при оценка на качеството има и класификацията на критериите от гледна точка на участниците – купувач, потребител, разработчик. Класификация е дадена в [108], като съществено за потребителите е коректен, бърз, лесен за използване софтуер, който да изпълнява и предоставя всички необходими функции. Желателно е софтуерът да е лесен за изучаване и ръководствата за потребителя да бъдат разбираеми. От друга страна, разработчиците предпочитат софтуерът да е добре документиран, с малко грешки, да е надежден, лесен за промяна и разширение. Купувачът като трети участник се интересува от евтин софтуер с ниска цена за поддръжка, възможности за многократно използване, висока продуктивност и коректно изпълнение на функционалността.

1.3.3.3 Модели и стандарти за оценка на качеството

Моделите описват в структуриран вид смисъла на понятието „качество на СП”. Всеки модел на качеството на програмното осигуряване принципно се състои от три основни компонента: (1) структура на характеристиките на качеството; (2) метрики; (3) критерии на качеството. Моделите за оценка на качеството в софтуерните проекти имат две основни приложения: (1) като основа за определяне на изискванията към качеството на продукта и (2) по отношение на техниките за осигуряване на качеството на СП и като средство за измерване на изискванията към качеството.

Модел на МакКол (McCall – 1977 г.) [5, 33] – това е един от най-значимите предшественици на съвременните модели. Чрез него се прави опит да се запълни празнината между схващанията на потребителите и разработчиците за качеството на СП чрез акцентирание на онези фактори на качеството на софтуера, които засягат и двете групи – потребители и разработчици. Моделът на МакКол детайлизира три типа от характеристики (наречени главни перспективи) в йерархия от 11 фактора на качеството, 23 критерия и метрики. Главните перспективи в модела са: (1) възможност за внасяне на изменения и за адаптиране на продукта (product revision), (2) възможност за пренасяне на продукта в друга среда (product transition), (3) операционни способности на продукта (product operations). Идеята на модела на Мак Кол е да представи пълна картина на качеството на продукта чрез синтез на факторите на

качеството. Действителните стойности на метриките се определят чрез анкета, в която съотношението на положителните и отрицателните отговори показва степента на удовлетворяване на критерия.

Модел на Б.Боем (B.Boehm - 1978 г.) [5, 33, 97] По същество този модел прави опит да остойности качеството на СП чрез система от атрибути и метрики. Моделът на Боем обхваща разширен кръг от характеристики, съчетаващи 16 критерия. Той е близък до модела на МакКол с това, че също представя качеството като йерархична структура, съставена от характеристики от високо ниво (обща полезност на СП), характеристики от междинно ниво (7 фактора на качеството, които представят очакваните свойства на СП) и примитивни характеристики (основа за определяне на метриките на качеството) – всяка от които допринася за общото ниво на качеството.

Въпреки, че наглед моделите на МакКол и Боем си приличат, то главната разлика между тях е, че моделът на МакКол се фокусира около прецизното измерване на характеристиките от високо ниво, докато моделът на Боем се основава на широк диапазон от характеристики, ориентирани главно към възможностите за поддържане на СП.

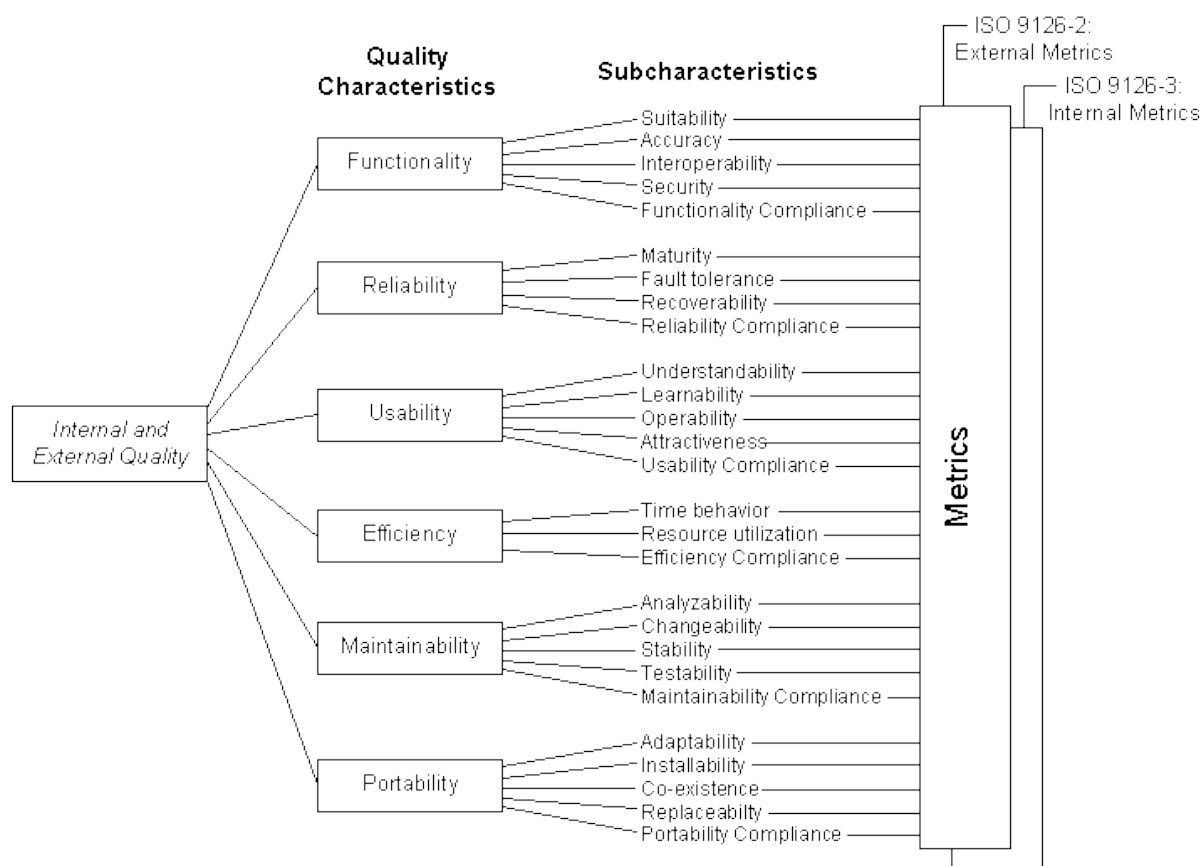
Модел на Дорми (Dormey) [5, 33] – това е един от по-новите модели, подобен на тези на МакКол и Боем. Дорми предлага модел на качеството, основаващ се на продукта, който приема, че оценката за качеството се различава за всеки продукт и, че е необходима една по-динамична идея за моделиране на процеса, така че той да бъде приложим към един по-широк кръг от различни софтуерни системи. Моделът на Дорми се фокусира върху връзката между атрибути и субатрибути на качеството и свързването на свойствата на продукта с атрибутите на качеството на СП. Общото за всички модели на качеството на СП е стремежът към организиране, детайлизиране и прецизиране на елементите на качеството (фактори, атрибути, критерии) в свързана йерархична структура от непресичащи се и некорелирани помежду си категории.

В специализираните източници могат да се намерят и предложения за използване на други модели на качеството, представляващи предимно разновидности на посочените като например йерархичен модел и класификационен модел [37, 97]. Типичен йерархичен модел на качеството на софтуера е предложен в [9], където качеството се разглежда като йерархична структура, намираща се на най-високо ниво, а на следващо ниво се намират факторите. Факторът се определя като потребителски ориентирано свойство, представящо даден аспект на качеството на софтуера от гледна точка на потребителя. От друга страна в [21] е предложен класификационен модел на качеството на ПП. Целта на този модел е да преодолее някои от недостатъците на йерархичните модели като субективността в значителна степен както и трудоемкостта.

Поради многообразието на модели за оценка на качеството, различието на отчитаните фактори и аспекти на разглеждане, се налага извода за необходимост от стандартизация [97, 104].

Основният стандарт, представящ качеството на софтуерния продукт като синтез от структурно свързани характеристики за оценяване на качеството на софтуерните продукти е ISO 9126 [5, 33, 28].

Стандартът се основава на моделите на качеството на МакКол и Боем. Състои се от четири части, които адресират следните области: модел на качеството (ISO 9126-1), външни метрики, вътрешни метрики, качество при употреба. Моделът на качеството на ISO 9126 е структуриран по начин, сходен с посочените два модела, като определя шест групи характеристики (набори от свойства) на софтуера, наречени „фактори на качеството“. Това са функционалност (functionality), надеждност (reliability), ефективност (efficiency), удобство при използването (usability), възможност за поддържане (maintainability) и преносимост (portability) [124].



Фигура 1.1: Модели на качеството на софтуера съгласно ISO 9126 [124]

Всеки от факторите се свързва с множество от субфактори (**фигура 1.1**), които от своя страна се подразделят на атрибути. Атрибутът е характеристика, която за създавания или съществуващия софтуерен продукт може да бъде проверена или измерена. Атрибутите не са строго дефинирани в стандарта, тъй като се приема, че те могат да бъдат различни за различните продукти [124].

Някои съществени проблеми, възникващи при прилагането на ISO 9126 са [124]: липсват конкретни указания за това, как да бъде представена общата оценка на качеството; липсва индикация за това как да бъдат изпълнени измерванията на

софтуерните характеристики; абстрактност на атрибутите; характеристиките на модела представят предимно гледната точка на разработчика, а не на потребителя на продукта.

В областта на осигуряване на качеството в технологиите на софтуерното инженерство съществуват и множество други стандарти като например ISO 9000 – за оценка на качеството на процеса за разработка на софтуер; ISO/IEC 15504 - оценяване на софтуерен процес; или групата стандарти IEEE Std. 1220-1998, Std. 730-1998, Std. 828-1998, Std. 828-1998 и други [27], но те нямат пряко отношение към съставянето на оценка на качеството на софтуерен продукт и затова няма да бъдат подробно разгледани в дисертационния труд.

От друга страна ръстът на системите за електронно обучение се е увеличил значително през последните години, благодарение на търсенето от страна на студентите на по-гъвкави възможности за учене и икономически натиск върху образователните институции, които виждат технологиите като мярка за намаляване на разходите. Въпреки това е налице значителна критика на качеството на използваните понастоящем системи. Проблемите включват ниска производителност, лоша използваемост и адаптивност, което прави трудно обслужването на специфичните нужди на различните учаци.

Въпреки широко разпространеното използване на системи за електронно обучение и значителните инвестиции в закупуване или разработването им, няма консенсус относно изготвянето на стандартна рамка за оценка на качеството на система в тази област. Липсата на одобрен, в електронното обучение, качествен модел на система е в рязък контраст с обширната работа по осигуряване на качеството на софтуера като цяло.

В [13] се разглежда ISO 9126 Quality Модел (ISO 1991) и се демонстрира как моделът може да се използва за оценка и подобряване на система за електронно обучение.

1.4 Разработване на информационна система

Различните дейности, които се извършват по време на разработката на един програмен продукт оформят жизнения цикъл на разработването му. Той започва с идентифицирането на изискванията към софтуера и приключва с верификация на разработения продукт спрямо спецификацията му.

1.4.1 Същност и основни понятия

Жизненият цикъл на една информационна система обхваща всички етапи на разработката, внедряването и експлоатацията ѝ. По-конкретно, при разработването и развитието на една информационна система трябва да се имат в предвид следните фази: анализ и дефиниция, проектиране, разработване, реализиране, тестване и внедряване. Постоянни компоненти са изследване на системата, анализ на реализуемостта и изграждане на базата данни, а променливи компоненти са процесите, които се променят и усъвършенстват непрекъснато.

В началото на 70-те години на миналия век възниква първият модел наречен класически модел [108], включващ следните основни фази, на които се основават в последствие повечето модели.

- **Анализ и дефиниция** - На този етап се анализират проблемите за решаване и се дефинират изискванията към програмния продукт (ПП). Т.е определят се възможните решения и изисквания към новата система. Възможни са следните алтернативи на етапа: да не се променя съществуващата система; да се модифицира или усъвършенства съществуващата система; да се реализира нова информационна система. Определя се стойността, предимствата и недостатъците на предложените решения. На тази база потребителите трябва да изберат някоя от предложените алтернативи. Ако се вземе решение за изграждане на нова информационна система, е необходимо да се определят информационните изисквания, които трябва да се удовлетворят от избраното решение. Информационните изисквания съдържат описание на потребителите на информация и техните конкретни подробности за формата на системата, мястото и срока, в която трябва да се завърши, за да се получи желаната информация от системата. На базата на анализа на изискванията се определят функциите на новата система. Основните резултати от този етап са: спецификация на изискванията и модел на продукта.

- **Проектиране** (моделиране на процесите) – Разработва се цялостен модел на новата информационна система. Определят се нейните основни компоненти, връзките между тях и използването им от потребителите. Определят се също така методите и средствата, които ще се използват за реализиране на новата система. Детайлизират се изискванията към хардуера, софтуера, базата от данни, комуникациите, взаимодействие с потребителите при новата система. Избират се информационните технологии, необходими за удовлетворяване на изискванията, определени при анализа. Анализира се бързодействието на различния възможен хардуер и софтуер, различни мрежови конфигурации, различни начини за организация на базите от данни и възможности за комуникация с потребителите, правят се тестове за сигурността и защитата на информацията. Основните резултати от този етап са: софтуерна архитектура и детайлен проект.

- **Разработка** - Създава се цялостна структурна схема на програмния продукт, описва се потокът на данните и потокът на управлението, създава се базата данни, написват се програмите на модулите на системата на избрания език.

- **Тестване и оценяване** - През този етап се извършва т.нар. модулно тестване, при което всяка самостоятелна единица се тества поотделно и се отстраняват грешките. След завършване на модулното тестване на всички създадени части се извършва обединяване в единна система, която в следващия етап се тества от независими специалисти.

При цялостното тестване се използват два подхода: интегрално и инкрементално тестване. При първия се сглобява цялата система и се тества като цяло. При втория се тестват първо два модула, после към тях се добавят един по един останалите модули.

Като последна проверка се извършват т. нар. приемни тестове: алфа и бета тестове. Алфа тестовите се извършват в средата на разработване с участието и на потребителя. Бета тестовите се провеждат в реална потребителска среда, като потребителите регистрират проблемите и ги съобщават на разработчиците, които внасят съответните промени.

- **Внедряване** – Осъществява се преход от старата към новата информационна система. Внедряването може да се реализира по няколко начина.

Паралелно – старата и новата система се използват паралелно за известно време, при което се сравняват резултатите, получени при функционирането им. Така се намалява риска за загуба на информация при сриване на новата система.

Директно внедряване – на определена дата новата система замества старата. Този начин е твърде рискован. При отказ на новата система може да се загуби информация, да се прекъсне работата на организацията. Този начин се използва при малки системи.

Пилотно внедряване – новата система се внедрява само в част от организацията. След като се установи, че всичко е нормално и след установяване на известни грешки се преминава към цялостно внедряване на новата система.

Поетапно внедряване – става постепенно.

Тук трябва да се спомене, че съществуват различни мнения относно броя и наименованието на фазите, но в общия случай те съответстват на основните дейности на процеса за разработване на един програмния продукт.

Според [104] за разработването на една компютъризирана информационна система, е необходимо процесът на разработка да премине през следните етапи:

- **Определяне на обхвата и целите** – предназначението на този етап е да се установи естеството на проблема, да се оцени неговия обхват и да се планира изследването на реализуемост.

- **Изследване на системата и анализ на реализуемостта** - предназначението на този етап е да се подготви доклад за мениджърите относно реализуемостта на съответното техническо решение.

- **Анализ на системата** - предназначението на този етап е да се състави логически модел на данните и процесите на системата.

- **Проектиране на системата** - предназначението на този етап е да се представят принципни решения на проблемите.

- **Детайлно проектиране** - предназначението на този етап е да се съставят подробни спецификации на системата, на базата на които тя ще бъде реализирана.

- **Реализация на системата** - предназначението на този етап е да се изгради системата и да се тества за съответствие със спецификациите.

- **Преход от старата към новата система** - предназначението на този етап е да се внедри работоспособна система, която адекватно да замести старата система.

- **Оценка и поддръжка** - предназначението на този етап е да се оцени до каква степен системата отговаря на критериите. Да се осигури непрекъсната поддръжка.

Тези етапи се реализират последователно. Проектът не може да премине от един етап към друг, докато не се завършат всички необходими дейности на предхождания етап.

Разработката преминава от етапа на изследване на физическите аспекти на съществуващата система (изследване на системата) през анализа на логиката на системата (анализ на системата) и логическото проектиране (проектиране на системата) до етапите, разглеждащи физическите аспекти на новата система (детайлно проектиране, реализиране и оценяване).

От съществено значение за ефективното функциониране на една система се явява и правилният подбор на методологията за изграждане на информационната система. От това какви етапи, техники и средства ще се използват и в каква последователност ще се прилагат, зависят крайните резултати. Изискванията към методологията според [115] включват:

- Получаването на един добър краен продукт – Потребителите да приемат ИС и тя да удовлетворява информационните им потребности; Подсистемите да са в синхрон с интегрираната ИС; ИС да е отворена за вграждане на нови подсистеми и за изменение на съществуващите; Ползите от ИС да надхвърлят разходите по създаването и; ИС да предлага опростен и лесен за усвояване интерфейс от потребителите.
- Добър процес на разработка - Съкращаване на времето за разработка; Намаляване на изискванията за квалификация от страна на разработчици и потребители; Контрол на процеса на разработка и идентифициране на резултатите.
- Стандартизиран процес на разработка - По-лесно може да се постигне интеграция на ИС; Потребителите по-лесно могат да се пренастроят от един проект на друг; Може да се достигне до една обща база от опит и познания.

Изискванията, породени от развитието на информационните технологии (компютри, компютърни мрежи, операционни системи, бази данни, програмни езици), като например развитието на програмното осигуряване от четвърто поколение даващо възможност за бързо, лесно и ефективно създаване на приложения, базирани на бази данни и мрежова работна среда, дават своето отражение върху методологията. Например: Да осигурява адекватна реакция на бързо променящата се действителност; Да осигурява поэтапно нарастване функциите на системата с възможност за постоянното им усъвършенстване; Да гарантира активното участие на потребителите и др.

1.4.2 Модели на жизнения цикъл

Основната цел на използването на модели на жизнения цикъл на системите е справяне със сложността и промяната. Моделът на жизнения цикъл на системата е съвкупност от дейности и връзките между тях с цел подпомагане разработката на една система (софтуерна, хардуерна, производствена). Методологията за разработка на една система представлява съвкупност от методи за изграждане на модели, прилагани по отношение на жизнения цикъл на системата. Процесът на разработка на една система включва различни фази, съответстващи на една или повече задачи, свързани с разработката на системата, като например анализ и дефиниция на изискванията към системата, проектиране, разработване, реализиране, тестване и внедряване [90]. Моделът на жизнения цикъл дефинира също и принципите и насоките, заложи в отделните фази.

Съществуват много и различни модели на жизнения цикъл и това значително многообразие се определя от твърде многото модификации на добре известни основни модели, а именно: последователни, инкрементални, базирани на създаването на прототипи, еволюционни и „пъргави” (Agile) модели.

Последователните модели се основават на серия от последователни дейности, като в тяхната първоначална редакция всяка дейност трябва да приключи преди следващите. Тези модели са подходящи за проекти, които са ясно дефинирани и разбираеми, и изискват опитни разработчици. Те са едни от най-известните модели използвани досега, а някои от техните основни и най-често използвани представители са: класически модел на жизнения цикъл [51], водопаден модел [53], V-модел или Фау-модел [20] и Cleanroom-Engineering модел [125, 126]. Известни и много популярни са и някои нови разширения на базата на итерации и обратни връзки [4, 26].

Моделите, базирани на прототипи, се опитват да подобрят последователните модели по отношение на дефиниране на изискванията чрез постигане на бързо проектиране на прототип, насочени към онези аспекти на системата, които са видими за клиента. Прототипът се оценява от клиентите и се използва като основа за усъвършенстване на изискванията [23].

Основната характеристика на инкременталните модели според [3] е, че те позволяват разделяне на проектите на отделни етапи на разработка, като всеки етап добавя нови функционалности и се базира на последователния модел на жизнения цикъл. Съществуват две основни разновидности на модела: с паралелна и последователна доставка на функционалност.

Основната характеристика на итеративните модели според [127] е възможността за връщане от една фаза в предишна. Тези модели избягват някои от основните недостатъци на последователните модели. При итеративните модели съществуват също така определени проблеми като например в стандартния итеративен подход връщането е възможно само в предходната фаза, а в определени случаи е необходимо връщане в по-предишни фази. За избягването на такива недостатъци са разработени

различни модификации на стандартния итеративен модел, като някои от основните са: Спирален модел [127], Водовъртежен модел и Фонтанен модел [122].

Моделът, основаващ се на бърза разработка на приложения (RAD модел) [44], илюстрира потенциала за създаване на разнообразни модели, чрез модификация и комбинация на основните видове. Той се основава на последователния подход като се включват и елементи на инкременталните методи и методите, базирани на създаването на прототипи с цел изключително съкращаване на времето за разработка.

Основният недостатък на всички горепосочени методи е, че те не отговарят напълно на еволюционния характер на системите, независимо от какво естество са. До известна степен, този проблем е решен с еволюционните модели [77]. При тези модели в началото на разработката не са известни всички изисквания към финалната система. Изискванията са частично определени и впоследствие усъвършенствани и разширявани чрез последователни стъпки до постигане на приемливо решение. Някои от най-приетите методи от този вид са модел на спиралния процес [7] и моделът, базиран на компоненти [14]. Основният недостатък на еволюционни модели е, че те могат да се справят с промените между фазите, но в рамките на една фаза, промени не се допускат.

В момента едни от най-обещаващите и използваните модели са „пъргавите” модели [2] на жизнения цикъл които се опитват да намалят риска и сложността на процеса на разработка в условията на бързо променящите се изисквания. „Пъргавите” методи са събирателен термин за различни нови модели при разработката на системи. Преглед и някои сравнения на тези модели са представени в [1] и [8]. Някои от най-популярните „пъргави” модели на жизнения цикъл са: Adaptive Software Development (ASD), Feature Driven Development (FDD), Crystal Clear (Crystal), Dynamic Software Development Method (DSDM), Scrum, Extreme Programming (XP), AMDD, Essential Unified Process (EssUP), Consolidated Agile Framework (CAF) и др.

1.5 Процеси в информационни системи

Информационната система е система за изпълнение на основните информационни дейности, които се състоят в събиране, съхраняване, обработване и разпространяване на информации. Съвкупността от тези дейности се нарича информационен процес. Т.е намирането на решение за дадена информационна дейност е информационен процес. Всеки информационен процес се представя с помощта на четири информационни дейности: събиране на информация от външни източници; преработване (обработване, преобразуване) на информация; съхраняване на информация; разпространяване на информация към външни потребители.

Според [116] процесът стои в основата на жизнеността на всяка система. За да се специфицира, се:

- прави дърво на целите в системата;
- определят процесите, които реализират тези цели;
- определят функциите и дейностите, които ги реализират;

- определят ресурсите, необходими за тяхната реализация.

В последните години голяма популярност като понятие придобиват бизнес процесите и тяхното управление.

Бизнес процес - понятие

Бизнес процесът (БП) представлява специфична подредба на работни дейности по време и място, които имат ясни начално и крайно състояние и ясни входни и изходни условия. Един бизнес процес, за да бъде определен като такъв, трябва да притежава определени характеристики като [41]: Определеност, Последователност, Клиент, Добавяне на стойност, Интегрираност, Функционална обхватност, Отговорник на процеса. В зависимост от различни критерии съществуват и различни видове бизнес процеси. Като цяло могат да се дефинират следните:

1. *Управленски процеси* – описват функционирането и управлението на цялостната система на организацията, напр. Корпоративно управление и Стратегическо управление.
2. *Основни* (оперативни) процеси – изграждат същността на дейността на организацията и изграждат основната верига на стойността..
3. *Спомагателни процеси* – съпровождат основните процеси, напр. Счетоводство, Човешки ресурси, Техническа поддръжка.

Управление на бизнес процеси

Управлението на бизнес процесите е както организационна стратегия, така и част от софтуерната индустрия. Набляга се върху моделирането, автоматизирането, управлението и оптимизирането на бизнес процесите с цел подобряване на резултатите на организацията. Разглеждайки и анализирайки завършени от край до край процеси, методологията премахва границите между звена, информационни системи и потребители [120].

Широко приета е дефиниция за управление на бизнес процеси, идваща от Workflow Management Coalition [134]: *Управлението на работните процеси включва концепции, методи и техники за поддържане на дизайна, администрирането, конфигурацията, изпълнението и анализа на работния процес.*

Системите за управление на бизнес процесите включват 3 основни компонента [120]:

- средства за моделиране на процеси (графично представяне на процес, който става част от приложението и задава начина, по който бизнес процесът действа, когато стартирате програмата);
- система за изпълнение на процеси - технологии за web и системна интеграция, осигуряващи извличането на данни през web браузър, които дават възможност да се включат нужните хора и приложения в процеси;

- средства за мониторинг - компонент за мониторинг на бизнес дейностите, извеждащ отчети за това как (и колко добре) се изпълняват бизнес процесите и работните потоци.

Списъкът на наличните продукти за моделиране, включително спецификация, симулация и функционално изпълнение на системите за управление на работния процес е доста дълъг [31, 54, 64, 67]. Продуктите са с различна степен на развитие и са представени едновременно като продукти налични на пазарните и софтуерни инструменти с отворен код. Класификациите на тези продукти може да се извършва, като се прилагат различни критерии [58]:

- Класификация на софтуерни инструменти за бизнес и научни работни процеси
- Класификация на софтуерни инструменти в зависимост от техния софтуерен език за моделиране
- Класификация на софтуерни инструменти в зависимост от стандарта, който поддържат
- Инструменти за работни процеси, базирани на отворен код и комерсиални такива.

Напоследък се създадоха голям брой езици за моделиране. Пример на графични езици за моделиране в областта на компютърните науки, управление на проекти и инженерни системи са представени в [54, 67, 75, 76]. Като един от най-използваните езици и графични нотации при моделиране на работни процеси в системите за управление на работния процес е BPMN.

BPMN - Целта на Нотацията за моделиране на работни процеси (BPMN) е много подобна на целта на Универсалният език за моделиране (UML) [88] при обектно-ориентирания дизайн и анализ. Тя е да се идентифицират най-добрите практики към момента и да ги обедини в нов всеобщо признат стандартизиран език. Множеството от предшественици на BPMN включва не само базирани на графи и на мрежи на Петри езици за процесно моделиране, но също така и UML диаграми и набор от процеси управлявани от събития. Докато тези езици за моделиране се фокусират на различни нива на абстракция, вариращи от бизнес ниво до по-техническо ниво, BPMN цели да поддържа целия диапазон от нива на абстракция - бизнес и технически: *“Основната цел на BPMN е да предостави нотация, която е разбираема без предварителна подготовка от всички бизнес потребители: от бизнес анализатора, който ще състави първоначалните чернови на процесите, до разработчиците отговорни за внедряването на технологията, която ще изпълнява тези процеси и накрая на служителите, които ще управляват и следят работния процес”* [10, 11, 75].

Предимства:

- Единствен стандарт за графично представяне за моделиране на бизнес процеси. Той позволява едновременно създаването на бизнес процеси на високо ниво с възможност за моделиране на сложни бизнес процеси, а така също и регистрирането им в изпълним език.

- Позволява моделиране на обработката на изключения, на транзакции и компенсации;

- Обмяна на съобщения между два обекта;
- Позволява обектите да бъдат представяни и може да бъде показано колко данни, документи или обекти биват използвани или актуализирани в процесния поток;
- Спецификацията му дава възможност за представяне в BPEL4WS. Съществуват средства за осъществяване на такова представяне.

Недостатъци:

- Спецификацията не осигурява метамодел или механизъм за обмен на диаграми. Предвижда се такъв механизъм в бъдещи версии на спецификацията;
- BPMN не предвижда схема за представяне на процесни архитектури на организационно ниво. Не съществува XML схема асоциирана с BPMN. Въпросът за XML ниво над BPM е предмет на бъдещи проучвания и разработки.

Хореография (choreography) – използва се за осъществяване на взаимодействие между процесите и приложенията на бизнес партньори. Взаимодействието се постига само с изпращане и получаване на съобщения. За да е коректно взаимодействието, процесите участващи в него трябва да имат обща хореография преди да започнат самото взаимодействие [19, 24, 86].

Оркестрация (orchestration) – представлява композиция от услуги, при която едната услуга изпълнява някакъв бизнес процес и в зависимост от стъпките в този процес извиква съответна услуга от композицията [31, 67, 87].

Технологиите, свързани с моделирането и изпълнението на бизнес процеси, могат да бъдат успешно използвани и при разработването на процеси в информационни системи свързани с електронното обучение. Всички дейности в една система, например за Web-базирано обучение, могат да бъдат също моделирани като процеси.

1.6 Методи за многокритериална оптимизация

Съвременните изследователски и приложни оптимизационни задачи са многокритериални и принципно конфликтни. Изборът на оценъчна система от критерии и тяхното ранжиране по степен на важност е трудно формализуем проблем, който няма еднозначно тълкуване и неизбежно поражда субективни решения. Подбирането на ограничен брой най-съществени критерии от множеството на реално определените възможни характеристики на оптимизируемия обект винаги представлява трудност [110].

За решаването на една многоцелева оптимизационна задача има три подхода според [110]:

- Обединяват се всички критерии и изисквания в една обща целева функция, която глобално да оценява ефективността от функционирането на обекта или системата. Многото и различни по характер критерии невинаги позволяват такова решение.
- Един показател се избира за основен. Той се приема за целева функция, а останалите се разглеждат като областни ограничения.

- Оптимизационната задача се разглежда като многоцелева или векторен критерий. Тази задача е известна като задача за многокритериално вземане на решение, многоцелева оптимизация или векторна оптимизация.

Множеството целеви параметри

$$y(x) = [y_1(x), y_2(x), \dots, y_m(x)] \quad (1)$$

носи названието векторен критерий.

Оптимизационната задача при векторен критерий изисква намирането на такова множество на управляващите фактори x^* , при които целевите параметри $y_j(x^*)$, $j = 1, 2, \dots, m$ в допустимо пространство Γ_x ще удовлетворят комплекс от компромисни изисквания.

Оптимизационната задача за многокритериално вземане на оптимално решение може да се формулира по следния начин:

$$\max_{x \in \Gamma_x} \{y_j(x)\} \equiv \max_{x \in \Gamma_x} \{y_1(x), y_2(x), \dots, y_m(x)\}, \quad (2)$$

при наложени функционални ограничения от тип равенство или неравенство (ако има такива):

$$\begin{aligned} g_l(x) &\geq 0, \quad l=1, 2, \dots, s \\ h_p(x) &= 0, \quad p=1, 2, \dots, k. \end{aligned} \quad (3)$$

Основната характеристика на задачата за многоцелева оптимизация е, че задачата е некоректна, тъй като няма едно единствено решение, а има безброй решения. Регуляризацията на тази некоректна задача обикновено става чрез скаларизация на векторния критерий.

Съществуват различни концепции за решаване на оптимизационните задачи с няколко критерия за оптималност. Въпреки това необходимостта от решаването на множество практически задачи с векторен критерий е довела до развитието на много методи за компромисни оптимални решения [15, 17, 25, 78, 81, 84, 85, 93, 109].

Въпреки многото класификации в литературата на методите за компромисна оптимизация, предложените методи могат да се разделят на две групи:

1. Методи за намиране на множество Парето-оптимални решения.

2. Методи на скаларизацията и компромисните решения.

1.6.1 Парето-оптимални решения

Липсата на единствено оптимално решение е друга характерна особеност в задачите за векторна оптимизация. Резултатът от тяхното решаване обикновено е множество от Парето-оптимални решения, получени с помощта на предложения от Вилфредо Парето [48] принцип за съгласувана оптималност.

Парето-оптималното решение (управление) $x^* \in \Gamma_x$ е онова решение, при което не съществува друго решение x^0 в допустимата област Γ_x , така че

$$y_j(x^0) \geq y_j(x^*), \forall j = 1, 2, \dots, m, \quad (4)$$

като неравенството (4) се спазва строго поне за един от критериите. Посоката на неравенството (4) е съобразена с изискването за търсене на максимум на всеки от критериите.

За намиране на неподбръваща се точка от допустимото пространство на управляващите параметри, която да удовлетворява условие (4) за Парето-оптималност, са предложени различни алгоритми [105, 112, 113], които прилагат основните методи на нелинейното програмиране.

Парето-оптимално решение може да бъде намерено по следните методи [113]:

- Случайно търсене;
- Симплексен метод;
- Сумарен вектор градиент.

1.6.2 Методи на скаларизацията и компромисните решения

Методи на препоръчаното решение за оптимизация при много критерии

Най-голямо приложение за вземане на оптимално решение при много критерии са получили различните модификации на така наречените методи на препоръчаните решения (Reference point approach) [25, 78, 81, 84, 85, 93, 109]. Тези методи се отнасят към широкия клас на методите на скаларизацията.

За всеки целеви показател се задава препоръчана стойност (препоръчана точка) - (reference point) - y_{jr} . В различни варианти на метода идеята е да се максимизират надхвърлянията над препоръчаната стойност y_{jr} , т.е. $\max \Delta_j^+$ и/или да се минимизира недостигането до препоръчаната стойност y_{jr} , т.е. $\min \Delta_j^-$, $j=1, 2, \dots, m$. Задачата е да се намери решение (управление) x^* , при което се получава $\max \Delta_j^+$ и/или $\min \Delta_j^-$, т.е.

$$\max_{x \in X_0} Fr(x) = \max_{x \in X_0} \{ \sum_{i=1}^m \Delta_j^+ - \sum_{i=1}^m \Delta_j^- \}, \quad (5)$$

където

$$\Delta_j = y_j(x) - y_{jr}, j=1, 2, \dots, m. \quad (6)$$

Δ_j се нормира, когато $y_j(x)$ имат различни дименсии, по следния начин

$$\Delta_j(x) = \frac{y_j(x) - y_{jr}}{y_{jr}}, j=1, 2, \dots, m \quad (7)$$

Когато y_{jr} клони към 0, или е много малко число се използва следният начин на нормиране

$$\Delta_j(x) = \frac{y_j(x) - y_{jr}}{y_{jmax} - y_{jmin}}, j=1,2,\dots,m \quad (8)$$

където y_{jmin} и y_{jmax} са избрани съответно минимална и максимална стойност за формиране на обобщена целевата функция, която замества векторния критерий (множеството от целеви параметри).

Във всички скаларизирани (обобщени) функции са включени тегловни коефициенти w_j за отделните показатели y_j . Тегловните коефициенти отразяват степента на важност на всеки отделен показател или удовлетвореността от показателя. Ако такива тегловни коефициенти не са зададени, те се приемат за равни на единица ($w_j = 1$).

Общ алгоритъм на методите на препоръчаните решения:

- Задават се целевите функции $y_j(x)$, $j = 1, 2, \dots, m$, изискванията към тях, ограниченията ($x_{min i} \leq x_i \leq x_{max i}$, $i = 1, 2, \dots, n$ или $x \in \Gamma_x$) и (3), тегловните коефициенти w_j и др.;
- Избира се стратегия и се специфицират препоръчаните стойности y_{jr} , $j = 1, 2, \dots, m$;
- В зависимост от избраната стратегия се формира функцията $F_r(x)$, така че да се максимизират надхвърлянията над препоръчаната стойност ($\max \Delta_j^+$) или да се минимизират не достиганията до препоръчаната стойност ($\min \Delta_j^-$), или едновременно $\max \Delta_j^+$ и $\min \Delta_j^-$;
- Максимизира се (или се минимизира) функцията $F_r(x)$ и се намира x^* ;
- Променят се препоръчаните стойности y_{jr} , $j = 1, 2, \dots, m$, и (или) начина на формиране на функцията $F_r(x)$;
- Търси се ново решение (с тегловни коефициенти w_j или без w_j);
- Прави се анализ на решенията и се избира едно от тях x^* .

Методите на препоръчаните решения имат няколко модификации [25, 78, 81, 93, 109]:

Метод на компромисна оптимизация по функция на загубите (оптимистичен подход)

Този подход се нарича "оптимистичен", тъй като се задават "оптимистични" (високи) изисквания за препоръчаните стойности на целевите параметри (качествените показатели) и се минимизират загубите, т.е. недостига до тях.

Изборът на работна точка (оптимално управление) x^* от областта на компромисите винаги е свързан с известни загуби от безкомпромисните оптимални решения. Тези загуби са теоретически равни на нула, когато екстремумите на m -те целеви функции съвпадат. Във всички други случаи се получават загуби [109].

Метод на многоцелева оптимизация по функция на полезност (песимистичен подход)

Този подход се нарича "песимистичен", тъй като се задават "песимистични" (минимално допустими) изисквания за препоръчаните стойности y_{jr}^p на целевите параметри $y_j(x)$ и се максимизират надхвърлянията над тях.

Идеята за максимизиране на надхвърлянията над песимистично зададената препоръчана стойност y_{jr}^p за j – тия целеви показател $y_j(x)$ и намиране на компромисно решение y_j^{kp} е илюстрирана в [109].

Когато целевите функции (качествените показатели) имат най-различни физически измерения, целевите параметри трябва да се приведат към еднаква скала на измерение и след това се съставя обобщен критерий. За такава скала се използва полезността на получените резултати за целевите параметри при различните стойности на x . Според теорията на полезността на всяко явление, събитие или целеви параметър може да се постави в съответствие положително число от 0 до +1, което да изразява полезността на това явление, събитие или целеви параметър.

Рамков подход (обединен оптимистичен и песимистичен подход)

Рамковият подход (Bracketing approach) [109, 110] обединява метода на функцията на загуби и функцията на полезност, т.е. на оптимистичния и песимистичния подход.

Оптималното компромисно решение y_j^{kp} се търси чрез едновременно минимизиране на недостига до желаните (идеалните) стойности y_j^* и максимизиране на надхвърлянията над необходимите (исканите) стойности y_{jr}^p :

$$\max_x F_r^{br}(x) = \max_x \left\{ \sum_{j=1}^m \frac{y_j(x) - y_{jr}^p}{y_j^* - y_{jr}^p} - \sum_{j=1}^m \frac{y_j^* - y_j(x)}{y_j^* - y_{jr}^p} \right\} \quad (9)$$

Първият член на обобщения критерий в (9) представлява функцията на полезност, (която трябва да се максимизира) и вторият член е функцията на загуби с обратен знак, (която трябва да се минимизира).

Идеята за максимизиране на надхвърлянията над песимистично зададената препоръчана стойност y_{jr}^p и за минимизиране на загубите от оптимистично зададената препоръчана стойност y_j^* за j – тия целеви показател $y_j(x)$ и намиране на компромисно решение по рамковия подход y_j^{kp} е илюстрирана в [109].

Метод на многоцелева оптимизация по функция на желателност

Въз основа на теорията на субективната желателност Харингтон [25] е формулирал обобщен критерий, който обединява всички целеви параметри към едностранна скала на желателност, използвайки експоненциална крива на изменение на желанието:

$$d_j(x) = e^{-(e^{-Q_j'(x)})}, j = 1, 2, \dots, m, \quad (10)$$

където

$$Q_j'(x) = b_{0j} + b_{1j} Q_j(x) \quad (11)$$

$d_j(x)$ е частна функция на желателност за $Q_j(x)$, която може да се изменя от 0 до +1;

b_{0j} и b_{1j} са коефициенти за j – тия целеви параметър, които зависят от желаните стойности на $Q_j(x)$.

Обобщената функция на желателност е представена подробно в [109, 110].

Както показват разгледаните модели и стандарти, качеството на софтуерния продукт е сложна категория, обобщаваща много отделни характеристики, свързани със специфични отношения. Множеството от наличните в моделите и различни по своята същност и по начин на оценяване фактори, критерии и метрики определя задачата като многокритериална.

1.7 Web-базирани образователни информационни системи

Години наред информационните технологии оказват съществено влияние върху развитието на образователната система. Това е така поради факта, че учебният процес е съвкупност от участници, средства и методи за предаване на необходими знания, така че усъвършенстването му в съответствие с новите технологични възможности е абсолютно задължително.

Електронното обучение придобива широка популярност през последното десетилетие и само по себе си представлява интерактивно обучение, което използва различни технологии за комуникация като средство за преподаване. То предоставя на обучаващия се лесен достъп до учебни материали в удобно за него време и място, като по този начин обучаваният получава бърза квалификация с минимални разходи. Електронното обучение представлява едновременно лесна за възприемане среда, която е гъвкава и висококачествена при използване в комбинация с традиционните начини на преподаване за повишаване на ефективността на образователния процес [91].

Наред с електронното обучение широко използвани понятия в наши дни са Web-базираното обучение, обучението онлайн и дистанционното обучение.

Web-базираното обучение, може да се дефинира като съвкупност от специализирани ресурси, достъпни чрез Интернет или Интранет, които позволяват на обучаващите се да получават знания и умения чрез самообучение. То не само предоставя необходимия учебен материал, но на практика изисква от обучаващите се да демонстрират нивото на овладените знания.

Факт е, че интересът към софтуерните продукти, даващи възможност да се реализира електронното обучение расте с бързи темпове. Съществува широк спектър от софтуерни продукти за осигуряване на дистанционно обучение [18, 61, 68, 92]. Може да се класифицират в следните основни групи:

- **Авторски инструменти (Authoring Packages)** – Тези инструменти дават възможност на преподавателите да разработват и публикуват самостоятелно учебно съдържание. По принцип продуктите от този клас не предоставят

възможности, свързани с контрола върху процеса на обучение и успеваемостта, нито осигуряват комуникация в реално време.

- **Системи за управление на обучението (*Learning Management Systems - LMS*)** - предназначени са за организацията на учебен процес в който участват голям брой обучаеми. LMS системите осигуряват възможности за управление, администриране, документация, проследяване и докладване на обучителните програми, класове и онлайн събития, програми по е-обучение и съдържание по обучението.
- **Системи за управление на съдържанието (*Content Management Systems - CMS*)** – Чрез тези системи се организира за публикуване голям обем от всякакво съдържание - текстове, картинки и мултимедия. CMS системите са особено ефективни, в случаите когато по създаването на учебното съдържание работят голям брой преподаватели, които трябва да ползват едни и същи фрагменти от учебни материали в различни курсове.
- **Системи за управление на учебно съдържание (*Learning Content Management Systems - LCMS*)** - Система, която създава, съхранява, окомплектова и предоставя персонализирано учебно съдържание във формата на учебни обекти. LCMS предоставят средства за създаване на учебни материали, за проследяване на обучението на голям брой хора, за съхранение и търсене на отделни елементи от учебното съдържание.

Комерсиални системи за е-обучение

Microsoft Class Server [144] - Системата е продукт на компанията Microsoft. Microsoft Class Server е мощен образователен инструмент, позволяващ на преподавателите да планират, съхраняват и предоставят учебни ресурси посредством интернет мрежата. Тази система, позволява на обучаемите използвайки Web браузер да достигнат до необходимите ресурси, да планират деня си и да завършат своите задачи. На високо ниво е комуникацията между преподавателите и обучаемите. Тя осигурява навременна обратна връзка и улеснява както оценяването, така и нормалния ход на учебния процес.

По-важни **характеристики** са: осигуряването на достъп в реално време до учебни планове, данни за успеха на студентите, разпространение на критерии за оценка, учебни материали, средства за генериране на отчети за целите на преподаватели и администратори.

Class Server системата включва приложението Class Server Teacher, което работи на локалния компютър на преподавателя и осигурява синхронизиране на информацията между компютъра на преподавателя и сървъра. Това приложение предоставя и календар на заданията, с който да се следи статусът на преподавателите, краен срок, брой предадени и неподадени задания, а също инструмент за създаване и управление на учебни ресурси, които се съхраняват в цифрово хранилище.

Lotus Learning Management System (Lotus LMS) [145] - Системата е продукт на компанията IBM. Lotus LMS дава възможност за пълно управление на процеса на

обучение - от създаване на учебното съдържание до организиране и провеждане на курсове. Студентите могат да избират курсове от онлайн каталог и да се записват самостоятелно за тези, които представляват интерес за тях или удовлетворяват техните нужди за сертификация. Lotus LMS предоставя четири модела на обучение – виртуални класни стаи, самообучение на място, сътрудничество (collaboration), и традиционно обучение.

Някои основни **характеристики** на системата са: Управление на ресурсите; LDAP поддръжка; Клиентското приложение може да работи без връзка със сървър; Модул за създаване на авторски учебен материал; Потребителски интерфейс, пригоден към нуждите на всеки един потребител; Възможност за генериране на доклади.

WebCT [146] – Системата е разработена от факултета по компютърни науки в Университета в Британска Колумбия, Канада. WebCT предоставя базисна среда, в която се създават лесно управляеми Web-страници. По-важни **характеристики** на системата са: Възможност за гъвкава организация на съдържанието, която позволява разграничаване между отделните факултети, преподаватели и тематични области; Възможност за разделяне на управлението на електронното съдържание от управлението на курса, т.е. системата позволява да се създаде учебно съдържание, което освен да се използва само в отделен курс, да може да се адаптира и включва в други модули вътре в учебната институция; Възможност за йерархичен модел на правата за достъп.

Основните **предимства** на WebCT са управлението на курсове на институционално и междуинституционално ниво и възможността обучаемите да контролират собственото си обучение.

Това, което отличава платформата от системите с отворен код е високата степен на надеждност и поддръжка.

Blackboard [147] - Blackboard е академичен софтуерен пакет, състоящ се от три части: среда за обучение, портал и система за управление на електронно съдържание. Обучаваният има свои настройки, дисково пространство, лични записи за извършваните дейности и др., които да са достъпни от всякъде. Чрез портала се постига висока степен на персонализация, която е нещо отделно от модулите за преподаване и съхранение на учебното съдържание. Платформата е достатъчно гъвкава, за да поддържа широк спектър от работни потоци. Курсовете могат да бъдат дискуссионно ориентирани или базирани на учебни ресурси.

Системата включва множество **функции** и модули като: Лесен достъп и управление на учебни материали за различни курсове; Табло за дискусии; Възможност за провеждане на обучение и дискусии в реално време чрез интегрирана система за текстови съобщения или виртуална класна стая; Възможност за създаване на тестове, анкети, домашни работи във формати, които да могат да бъдат правени онлайн и оценявани автоматично; Възможност за автоматично записване на оценките на всеки студент от проведени и оценени автоматично курсови работи; Възможност за автоматични справки

за резултатите на студентите; Възможност за оценяване на самия преподавател до каква степен е успял да обучи студентите на зададените като цели на курса теми.

Предимство на системата е предоставянето на индивидуално обучение за всеки един студент, без значение от това къде той се намира.

Системи с отворен код за е-обучение

Moodle (Modular Object-Oriented Dynamic Learning Environment) [148] – Moodle е проект на Martin Dougiamas, Curtin University of Technology. Системата се разпространява свободно като софтуер с отворен код. Той е подходящ за всички видове онлайн курсове, както и като допълнение на обучението в реална учебна среда. Системата включва множество **функции** и модули като: администриране на курсисти; публикуване на всякакви текстови и илюстративни материали; търсене на обратна връзка чрез анкетни проучвания; изготвяне на изпитни материали; възможност за даване на заданията на курсови проекти; предоставяне на средства за оценяване; форуми; възможност за обмен на съобщения; собствена електронна поща (e-mail).

Интерфейсът е прост и олекотен откъм технически детайли. Инсталира се лесно на която и да е платформа, поддържаща PHP и SQL база данни.

Основно **предимство** на Moodle е, че фокусът ѝ е насочен по-скоро върху опита, който ще придобият обучаемите, а не върху ресурсите, които трябва да бъдат прочетени. Наборът от инструменти и структурата на системата предлагат не само презентирание и предаване на знания, но и ангажиране на обучаемите посредством извършване на дейности.

В Moodle всеки един от участниците в процеса на обучение е в състояние да следи дейностите, извършвани от други участници в групата или индивидуално. Освен това, обучаемият може да наблюдава реакциите на колегите си и да разбере начина, по който са постигнали резултати и по този начин да се направят изводи за процеса на обучение.

ATutor [149] - ATutor е Web-базирана система за управление на учебното съдържание, която се предоставя под формата на отворен код. Създадена е по проект от Университета в Торонто, Канада. Системата предоставя възможност за бързо инсталиране и обновяване от страна на администраторите. Преподавателите могат лесно да създават, събират, пакетират и разпределят обучаващо съдържание, както и да предоставят създадените курсове онлайн. Основната идея на системата е да се реализира адаптивна учебна среда, която да може да бъде използвана от всеки, независимо от подхода на учене и технологичните възможности, с които разполага. Системата включва следните **функции**: Заложени са шест модела, описващи разбирането за учене и структурата на знанието: визуално възприемане, словесно възприемане, възприемане чрез дейности и движение, глобално възприемане, йерархично възприемане, последователно възприемане. Инструментите, които системата предоставя за използване от авторите са: редактор на SCORM съдържание; споделяне на съдържанието; шаблони; инструмент за създаване на тестове; форум; вътрешна поща; PHP базиран чат. **Предимства**: Категоризиране на създадените въпроси и съхраняването им в база данни; Генериране на тестове от произволни задачи

от базата данни, при възможен избор от 4 типа. Предоставя се възможност за избор на автоматично оценяване на тестовете на обучаемите от страна на системата или създателят на курса сам да направи своя оценка; Темите и уроците в курса могат да следват определена последователност или да бъдат нелинейни. Многократна използваемост на учебните обекти, създадени от автора. **Недостатъци** на системата са, че не могат да се задават учебни пътища и да се дефинират учебни дейности [80, 123].

Claroline [150] - Claroline е Web-базирана система за управление на обучението. Предоставя възможност за създаване и администриране на курсове. Проект е на Университета в Льовен, Белгия. Системата дава възможност за реализиране на широко разнообразие на педагогически решения: от традиционната класна стая до онлайн работата в сътрудничество. Системата включва следните **функции**: Създаване на тестове за самооценка; широк спектър от шаблони, които могат да се използват в различните модули; форум; Чат; Възможно е задаване на учебен път чрез подреждане на учебни ресурси в определена последователност. **Предимства**: Основно предимство на системата е, че е базирана на конструктивизма. Т.е. информацията се трансформира в знания чрез извършване на дейности от обучаемите и тези нови знания са в основата на следващите. Системата е преведена на 32 езика, което позволява използването и от голям брой институции от различни страни.

Dokeos [151] - Dokeos е web базирана система за дистанционно обучение с отворен код. Кодът на системата е достъпен за всички и може да бъде променян и приспособяван за постигане на специфичните нужди на потребителите. Също така системата предлага широк кръг от инструменти за дистанционно обучение и осигурява добри възможности за създаване и организиране на интерактивно мултимедийно съдържание за обучение, както и възможности за оценяване (тестове) на постигнатите резултати. Системата предлага ефективна потребителска среда с online съдържание и създаване на инструменти за оценка, инструменти за съвместна работа, както и инструменти за проследяване, отчитане и мониторинг на дейността на потребителите и техният напредък [74].

Функции: Във всеки един курс преподавателя е в състояние да организира студентите в групи. Платформата осигурява условия за самостоятелна работа на всяка група посредством виртуалното пространство, достъпно само за потребителите от групата. Всяка група разполага с място за споделяне на различни файлове или връзки към Web сайтове, с форум, който позволява активна асинхронна комуникация, с wiki, което позволява групово редактиране на текст, с чат за синхронна комуникация.

Преподавателя е в състояние да поставя задачи към групата, да организира синхронна или асинхронна комуникация посредством форум и чат. На студентите се предоставя възможност да добавят документи, както и да работят с такива, добавени от други студенти, да дават становище по дадени документи и да получават такова от свои колеги. Посредством системата за тестване на знанията, преподавателя е в състояние да следи ежедневната подготовка на студентите.

Допълнителните модули, които системата включва са: учебна програма, възможност за публикуване на съобщения от преподавателя, въпросници за студентите, изпращане на лични съобщения между студентите, RSS четец, терминологичен речник.

Предимства: Възможност за стартиране на различни операционни системи; отговаря на всички функционални изисквания, предоставя съдържание съгласно e-learning стандартите SCORM и IMS; възможност за развитие и разгръщане на собствени e-образователни компоненти.

Платформи - собствени разработки

eLSe (e-Learning Shell) – системата е разработена в Катедрата по информационни технологии на Русенския университет. Системата е базирана на продукти с отворен код. e-Learning Shell е платформа, която позволява на преподаватели да публикуват свой собствени Web-базирани курсове без да е необходимо да притежават компютърни познания и умения. Някои по-важни **характеристики** са: Три нива на достъп до системата; Системата притежава вградени средства за промяна на съдържанието и въвеждане на нова информация; Възможност за генериране на тестове; Средства за асинхронна комуникация като форум и електронно табло за съобщения; Средства за статистически контрол; SCORM съвместима и многоезична. eLSe платформата е описана подробно в [57].

DeLC (Distributed eLearning Center) – В отговор на съвременните изисквания и тенденции за подпомагане на обучението чрез използване на най-новите информационни и комуникационни технологии, във Факултета по математика и информатика на Пловдивския университет се разработва инфраструктура, наречена Разпределен център за електронно обучение (DeLC). Този център е предназначен да доставя по един контекстно-зависим, адаптивен и персонализиран начин електронни образователни услуги и електронно учебно съдържание, разположени върху сървъри, които физически могат да бъдат разделени. DeLC представлява динамична мрежова структура, която се състои от Възли (върху възлите могат да бъдат разполагани електронни образователни услуги и хранилища за електронно съдържание) и Релации (релациите специфицират определени зависимости и взаимоотношения между възлите). Един от възлите на DeLC е образователен портал, предназначен за обучение както на студенти във висшето образование, така и да бъде адаптиран и за обучение на ученици в средното образование. Архитектурата на образователния портал е сървисно-ориентирана и многослойна и се състои от три логически нива: потребителски интерфейс (user interface), електронни услуги (e-services) и електронни библиотеки (digital libraries) [69, 70, 71, 72].

В последните години в областта на образованието и на електронното обучение навлиза терминът „Сериозни игри”. Използването на игри като форма на обучение се счита като потенциал за подобряване на качеството на процеса на обучение. Този тип игри се използва за представяне на учебния материал в игрова форма, като по този начин се улеснява получаването на нови и затвърждаване на старите знания, навици и умения. Обучението, базирано на игри позволява на учащите да предприемат задачи и опитни

ситуации, които в друг случай биха били невъзможни и/или нежелателни като цена, време, логистика или съображения за сигурност [49, 106].

Като тенденция в развитието на платформите за е-обучение се възприема подход базиран на използване на услуги [62]. Вместо големи приложни програми с многобройни и все по-трудно обозрими възможности, се преминава към множество от малки, ефективни и лесно свързвани помежду си приложения, реализиращи ясни еднозначни функции. Не се набляга вече върху технологичните основи, а върху въпроса как най-пълно да се удовлетворят нуждите на потребителите, как да им се предоставят най-естествените средства за работа. Преминава се към втората вълна в електронното обучение. Първата бе свързана с т.нар. Виртуални Образователни Среда [63], които имаха голям успех, но не дават възможност за лесен обмен на информация помежду си, както и за многократно използване в различен образователен контекст. Приложенията за е-обучение от втората вълна имат за цел да решат именно тези проблеми. Те са базирани на масово използване на електронни хранилища от учебна информация, групирана във вида на “Учебни обекти”. В основата на тази втора вълна са и идеите за стандартизация на е-обучението.

1.8 Изводи

В съвременното общество значението на информацията нарасна изключително много, третирайки се като бизнес ресурс благодарение на компютърните информационни системи. Създаването им е свързано с приложното програмиране, както и с разнообразни други средства на съвременните информационни технологии. Приложното програмиране се прилага за решаването на определени задачи, свързани с различни сфери и области на съвременния начин на живот. Създаването на една информационна система се извършва на определени етапи и по определени методи.

Описани и аргументирани са принципите на софтуерните технологии като средство за създаване на прототип на информационна система. Софтуерната технология е изградена от множество стъпки, които включват методи, средства и подходи. Методите определят как се изгражда софтуер, средствата от своя страна осигуряват автоматизирана или полуавтоматизирана поддръжка на методите, а подходите са свързващото звено между методите и средствата, които гарантират разработка на компютърния софтуер.

Разгледани са стандартите и моделите за оценка на качество на софтуерните продукти. Важна характеристика, която обобщава ефекта от влиянието на множество разнообразни фактори върху процеса на разработване и създаване на софтуерния продукт е неговото качество. Съставянето на количествена оценка за качеството на софтуерния продукт е важна, не само за оценката и за сравнението на отделните програмни приложения, но и за ефективното управление и усъвършенстването на процесите в софтуерното инженерство като цяло. В областта на осигуряване на качеството съществуват множество стандарти, но основният, представящ качеството на софтуерния продукт като синтез от структурно свързани характеристики е ISO 9126. От друга страна, моделите на качеството описват в структуриран вид смисъла на

понятието „качество на софтуерен продукт”, базирайки се на основните си компоненти и приложения.

Различните дейности, които се извършват по време на разработката на един програмен продукт оформят жизнения му цикъл. Той обхваща всички етапи на разработката, внедряването и експлоатацията му. От значение е правилният избор на модел на жизнения цикъл на системата. Той е съвкупност от дейности и връзките между тях с цел подпомагане разработката на една система.

Електронното обучение, което може да се разглежда като вид информационен процес, придобива широка популярност през последното десетилетие и само по себе си представлява интерактивно обучение, което използва различни технологии за комуникация като средство за преподаване.

Съществуват редица причини, които оправдават необходимостта от създаването на собствена система за е-обучение, като например високите цени на чужди системи за организиране и управление на обучението, които са несъпоставими с нашите условия; Езиковата бариера, поради това че системите са ориентирани основно към английски говорещите страни; Липса на възможност за гъвкаво обновяване на ползваната система с развитието на нови информационни технологии. Закупените или получени чрез лиценз системи не могат да се променят; Въпреки универсалния характер на повечето системи, на практика съществуват характерни разлики между обучението по инженерни, хуманитарни, медицински и др. дисциплини, което оправдава и налага разработването на собствени системи, съобразно спецификата на дисциплините. Например в медицинските дисциплини е важно решаването на ситуационни задачи (клинични казуси).

Резултати от изследванията по тази глава са публикувани в [7] от публикациите по дисертацията.

1.9 Цел и задачи на дисертационния труд

Дисертационното изследване разработва *проблема* по количествената оценка на показатели на функциониране на информационни системи и процеси.

Обект на изследване е разработване на информационна система за изпълнение на електронно обучение по дисциплини от областта на медицината.

Предмет на изследването е проектиране и разработване на информационна система за електронно обучение от областта на медицината; количествена оценка на параметри на системата и процесите в нея; намиране на потенциал за подобряване на процесите в системата чрез дефиниране и решаване на оптимизационна задача.

Цел: Целта на дисертационния труд е проектиране и разработване на информационна система, количествено оценяване на характеристики на нейни процеси и определяне на потенциала за подобряване на тези характеристики.

Задачи:

1. Да се анализират технологиите за разработване на информационни системи. Да се анализират стандарти и модели за оценка на характеристиките им;
2. Да се проектира и разработи специализирана информационна система като:
 - се анализират и формулират изискванията, на които трябва да отговаря системата;
 - се разработи модел на системата и нейните процеси;
 - се разработят програмните модули и структурата на базата данни;
 - се анализира и сравни функционалността на разработената Web-базирана информационна система с известни софтуерни платформи в областта на електронното обучение.
3. Да се състави процедура за оценка на характеристики на информационната система като се ползват критерии и стандарти за качество на софтуерните продукти и дидактически показатели;
4. Да се оценят количествено характеристиките на системата;
5. Да се определи потенциала за подобряване на съществуващите характеристики на информационната система чрез дефиниране на съответна оптимизационна задача.

Глава втора: Синтез на процеси в информационни системи

2.1 Увод

Във втора глава се разглежда методология за проектиране и разработване на информационна система, включваща етапите, техниките и средствата, които ще се използват. Анализират се изискванията към системата. Синтезират се сценарии на действието и на работата на отделните модули и се проектират моделите на процесите им. Разгледана е възможност за прилагане на метод за многокритериална оптимизация към подход за оценка на качеството и ефективността на система за Web-базирано обучение.

2.2 Методология за проектиране и разработване на информационна система

От съществено значение за ефективното функциониране на една информационна системата е правилният подбор на методологията за изграждането ѝ. От това какви етапи, техники и средства ще се използват и в каква последователност ще се прилагат, зависят крайните резултати.

2.2.1 Етапи

За да се разработи една информационна система, е необходимо процесът на разработка да премине през определен брой етапи. Жизненият цикъл на една информационна система обхваща всички етапи на разработката, внедряването и експлоатацията ѝ. Съществуват различни мнения относно броя и наименованието на етапите (фазите), но в общия случай те съответстват на основните дейности на процеса за разработване на един програмен продукт.

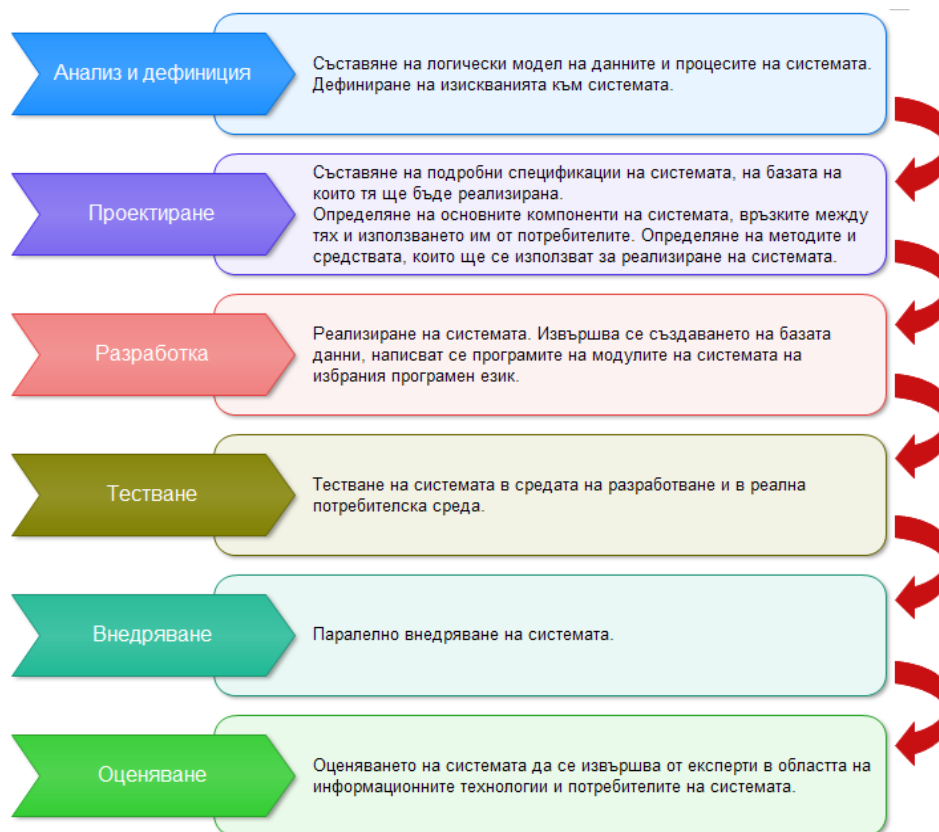
На база изложените в първа глава фази от класическия модел, за целите на дисертацията се приемат и описват шест основни фази, през които минава разработването на информационната система за организиране и управление на Web-базирано обучение по медицински дисциплини:

- Анализ и дефиниция на системата. Анализират се литературни източници и се проучват аналогични програмни системи, като по този начин се определят възможните решения и изисквания към новата система. Определянето на изискванията се разглежда като итеративен процес, като в него вземат активно участие както софтуерни специалисти, така и специалисти в съответната приложна област. Определят се целите и функциите на системата и се съставят логически модели на процесите ѝ.
- Проектиране на системата. Тази фаза се разделя на два подетапа - предварително или общо проектиране и детайлно проектиране. Като резултат от предварителното проектиране се създава архитектурата на системата, представя се цялостната ѝ структура, логическата организация на данните на системата и нейния интерфейс. При детайлното проектиране системата се представя като

съвкупност от отделни програмни модули. Чрез логически блок-схеми се описват процесите на модулите и връзките между тях. Определят се хардуерните и софтуерните средства, които ще се използват при реализирането на системата.

- Разработка на системата. Системата се изгражда в съответствие с нейните спецификации. Създава се базата от данни и се написват програмните модули на избрания програмен език.
- Тестване на системата. Тестването на системата преминава през няколко стъпки. На първо място се извършва т.нар. модулно тестване, при което всеки един програмен модул се тества поотделно и се отстраняват грешките. Следващата стъпка е цялостно тестване на системата като се използва подхода за интегрално тестване. Има се в предвид сглобяване на цялата система и тестването ѝ като цяло. Като последна проверка се извършват т. нар. приемни тестове: алфа и бета тестове. Алфа тестовите се извършват в средата на разработване с участието и на потребителя. Бета тестовите се провеждат в реална потребителска среда.
- Внедряване на системата. Осъществява се преход от старата към новата, като внедряването се реализира паралелно. Т.е старата и новата система се използват паралелно за известно време, след което се преминава окончателно към новата система.
- Оценяване на системата. Тази фаза включва количествена оценка на качеството и ефективността на системата.

Тези фази се реализират последователно и са показани на **фигура 2.1**.



Фигура 2.1. Фази на жизнен цикъл на информационната система

Към всяка една фаза е представено кратко описание на задачите, които трябва да се изпълнят.

2.2.1.1 Анализ и дефиниция на изискванията към системата

През тази фаза се извършва анализ на процесите и проблемите, определят се възможните решения и се дефинират изискванията към системата за Web-базирано обучение. На базата на анализа на изискванията се определят и функциите на системата.

- **Анализ на процесите на системата**

Целта на анализа на системата е да се изясни какво трябва да се направи, за да се реализират нейните функции. Това се осъществява чрез декомпозиция на функциите на системата на техните съставни елементи и създаване на логически модел на процесите и потоците от данни, необходими за тяхното изпълнение. Логическия модел се илюстрира чрез блок-схема на потоците от данни, а алгоритмите на действието се илюстрират чрез логически блок-схеми на информационните потоци.

Блок-схеми на потоци от данни подпомагат изграждането на логическия модел на системата. Те показват различните потоци от данни между процесите, които преобразуват данните. Също така блок-схемите показват съхраняването на данните и точките, в които данните се въвеждат и извеждат от системата [117].

Логически блок-схеми на информационни потоци са специфичен набор от добре дефинирани прости логически процедури, които трябва да се следват за решаване на проблем в краен брой стъпки. В литературата [104, 117, 128] логическите блок-схеми най-често се срещат под названието алгоритми. Те са се доказали в практиката като успешен начин за решаване на трудни проблеми чрез разделянето им на прости и лесни за разбиране стъпки. Представят нагледно серия от действия и са лесни за разбиране и проследяване. Логическите блок-схеми могат също така да се използват на етапа на проектиране на системата като средство за описание на алгоритмите на действие на отделните програмни модули.

Основните предимства и недостатъци при използването на този вид схеми са [104]:

Предимства:

- Логическите блок-схеми представят нагледно серия от действия и затова са лесни за разбиране и проследяване.
- Използват се за представяне на последователността на съответните действия при изпълнение на дадена задача.
- Подходящи са за представяне на случаи, когато има разнообразни операции и оценка на условия.
- Могат да се използват за специфициране на програми.

Недостатъци:

- Веднъж създадени се променят трудно и това изисква време.
- Не са пригодени за пряко възприемане и изпълнение от компютърни системи.

Целта на анализа на системата е да се получи представа за съответната система, която да бъде достатъчна, за да се проектира компютърен заместител на системата.

- **Дефиниране на изискванията на системата**

Дефинирането и формулирането на изискванията ще бъде съобразено с резултатите от направените анализи и изследвания и вземането в предвид мнение на независими експерти в областта на информационните технологии и електронното обучение и експерти от институцията, която предлага системата за Web-базирано обучение. Изискванията ще бъдат групирани в две категории: изисквания, изпълняващи функции характерни за универсална система за управление на обучението [46], и изисквания, включващи характеристики на качеството на софтуерните продукти [129, 130].

2.2.1.2 Проектиране на системата

През тази фаза детайлно се анализират изискванията от предходния етап и се разработва цялостен модел на информационната система. Определят се нейните основни компоненти, връзките между тях и използването им от потребителите. Определят се също така методите и средствата, които ще се използват за реализиране на системата. Детайлизират се изискванията към хардуера, софтуера, базата от данни, комуникациите, взаимодействието с потребителите на системата.

- **Проектиране на базата данни**

Информационните системи се дефинират чрез функциите, които предоставят на своите потребители. Главната им цел е събиране, съхраняване, обработване и разпространяване на информация, а основен техен елемент е базата от данни.

Проектирането на базата данни е процес, от който в голяма степен зависят крайните резултати при разработка на всяко приложение.

От това доколко правилно са подбрани обектите, техните атрибути и връзките помежду им зависи бързодействието, функционалността и възможностите за развитие на информационната система.

Процесът на проектиране на БД е свързан с проектирането на следните три модела [96] [143]:

- **Концептуален модел** – включва представяне на всички обекти и взаимовръзките между тях с помощта на моделите семантичен, фреймов и модел “същност-връзка”;

Избор на концептуален модел:

Модел “същност-връзка” (ER (Entity-Relationship) модел): Това е един от най-популярните концептуални модели на данни [12]. Моделът е предложен от Питър Чен (Chen) през 1976 г. Моделирането на предметната област се базира на използване на графични диаграми, в които обектите и връзките между тях се представят с графични символи.

Този модел се базира на основни понятия [12], като:

- обект (entity - същност) - нещо, което съществува реално, представлява интерес за предметната област и за което ще се съхранява информация в БД.
- тип обекти (entity set - клас обекти) - множество от всички подобни обекти в предметната област, т.е обекти притежаващи общи свойства.
- свойство (attribute - свойство) - обектите се характеризират със свойства - свойството е някакъв факт за обекта.
- връзка (relationship) - свързва или асоциира обекти (2, 3 или повече) от различни класове или от един и същи клас. Най-често всяка връзка свързва два обекта.
- тип връзки (relationship set) - множество от всички подобни връзки, които съществуват между два или повече класа обекти и имат еднаква семантика.
- видове връзки - съществуват няколко вида, в зависимост от това, обект от единия клас с колко обекта от другия може да бъде свързан в определен момент и обратно. Връзките се разделят на: 1:1 (едно към едно), 1:M (едно към много), M:M (много към много).

Моделът ER е доразвит от множество разработчици и днес съществуват редица негови разновидности: модел на ORACLE [131]; модел на Vantage Team Builder; модел на Баркер; метод IDEF1 [132], разработен от Т. Ремей (T.Ramey), и неговата доразвита версия IDEF1X [133] (използва се в ERwin, Design/IDEF и др.); метод IE (Information Engineering) и т. н. Разликите между отделните модели се състоят в графичното изобразяване на обектите, атрибутите и връзките, както и в степента на отразяването на особеностите на обектите, атрибутите и връзките.

Тъй като ER моделът е по-близък по принципи на организацията до релационния модел, и реализацията на последния на базата на първия е най-удобна, то в качеството на концептуален модел се избира модел “същност- връзка”.

- **Логически модел** – включва представяне на обектите и взаимовръзките с помощта на моделите на данни: мрежов, йерархичен или релационен; представяне на графически модели, изхождайки от модела на данни [118, 119, 143];

Избор на логически модел:

Релационен модел: Релационният модел на данните е предложен от Е. Код през 1970 година.

Този модел има две важни характеристики: прост и естествен начин за представяне на данните - таблици; строга математическа основа на модела - теория на множествата и предикатната логика;

Всяка таблица се състои от редове и колони, като значенията във всяка колона са сравними- принадлежат на едно множество от допустими значения. Всяка колона има наименование, което показва какъв е смисъла на значенията в колоната.

Предимства [143]:

- *Вградена цялост на много нива:* Целостта на данните е вградена в модела на ниво поле, за да се осигури точността на данните; на ниво таблица, за да се осигури, че записите не са дублирани и за да открие липсващи стойности на първичен ключ; на ниво релация, за да осигури, че релациите между две таблици са действителни; и на бизнес ниво, за да осигури, че данните са точни от гледна точка на самия бизнес.
- *Логическа и физическа независимост на данните от приложенията за бази данни:* Нито промените, които даден потребител прави с логическия строеж на базата данни, нито промените, които компанията, продаваща софтуер прави с физическата реализация на базата данни ще повлияе неблагоприятно на приложения, основаващи се на тях.
- *Съгласуваност и точност на данните:* Данните са съгласувани и точни, благодарение на различните нива на цялост вътре в базата данни.
- *Лесно извличане на данни:* По подадена от потребителя команда могат да бъдат извлечени данни от определена таблица или от няколко свързани таблици в базата данни. Това позволява на потребителя да разглежда информация по почти неограничен брой начини.

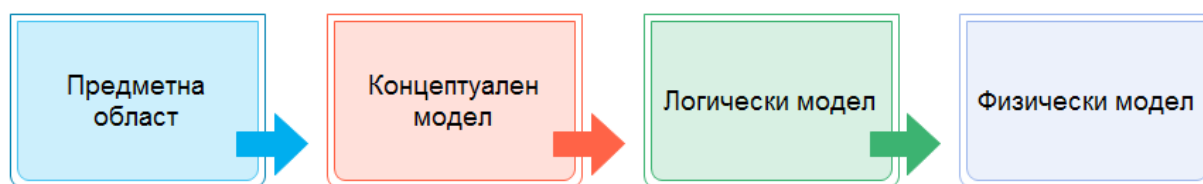
Недостатъци: Релационните модели, които не са съгласувани с нормализационни правила могат да дублират данни в различни таблици. За да се направят данните по-ефективни много такива модели не са съгласувани напълно с тези правила. [143]:

Йерархичният и мрежовият модел са по-ефективни при процеси с голям обем данни, а релационният - при по-сложни и разнообразни данни. Релационните модели не изискват поддръжка на указатели, както при останалите два модела.

Поради това, че от изброените логически модели на данните релационният притежава предимства като например лесно извличане на данни, съгласуваност и точност на данните, независимост на данните от приложенията на БД, вградена цялост на данните на много нива, за да бъдат защитени от грешки и др., релационният модел беше избран за разработване на БД на системата.

- **физически модел** – включва методите за съхраняване и достъп до данните, а също и методите за достъп до ОС;

На **фигура 2.2** е представен модел на процес на проектиране на БД.



Фигура 2.2. Процес на проектиране на БД

Подходи при проектирането на релационни бази от данни

Съществуват два основни подхода при проектирането на релационни бази от данни [52, 107] :

- Подход “отгоре –надолу” (top-down) Използва се при проектирането на приложения от тип “бази от данни”. Състои се от следните стъпки:

- проектиране на концептуална схема на базата от данни, като се използва модел от високо ниво;
- изобразяване на получената схема в множество от отношения;
- анализ на отношенията с помощта на функционалните зависимости, като се прилагат неформалните мерки за качество;
- провеждане на нормализация до трета нормална форма, за да се избегнат транзитивните и частичните функционални зависимости.

- Подход “отдолу-нагоре” (bottom-up), или релационен синтез. При релационния синтез схемата на конкретна релационна база от данни се разглежда като съвкупност от функционални и други типове зависимости, задавани върху атрибутите от проектанта на базата. След като всички зависимости се дефинират, с помощта на алгоритъм за нормализация се синтезират релационни схеми в трета нормална форма (3NF) чрез групиране на подходящи атрибути. Всяка релационна схема трябва да съдържа логически смислено множество от атрибути.

Етапи при проектиране на релационни БД:

Проектирането на релационната БД на системата включва следните етапи [52, 103, 107]:

- *Определяне на целите, на които ще служи БД.* Изграждането на една БД зависи от поставените цели към системата. Трябва да се осигурят надеждна защита на данните и различни права на достъп за различните категории потребители, бърз достъп до данните и др.

- *Определяне на необходимите таблици.* Обикновено в една таблица се записват данните, които се отнасят за един клас от обекти. Колко таблици ще има в една РБД, зависи от броя на описваните класове от обекти.

- *Определяне на полетата във всяка таблица.* При определяне на полетата се задава име на полето, тип на данните, размер на полето – посочва се минимално необходимият, формат на данните, стойност по подразбиране (Default Value), условие за допустима стойност (validation Rule).

- *Определяне на първичните и външните ключове.* За да бъде осигурен достъпът до точно определен запис, всяка таблица трябва да разполага със специална колона (поле), наречена първичен ключ. Първичният ключ се използва за идентифициране на записите в таблицата. Стойностите в тази колона (или група колони) са уникални (не могат да се повтарят).

Една релация може да има повече от един ключ. Тогава единият се избира за първичен, а останалите са вторични ключове.

Релационната база данни съхранява информация в много таблици, свързани с ключови полета. За установяване на връзка между две релационни таблици, освен първичен, се използва и външен ключ, който съответства на първичен ключ на друга релация.

- *Определяне на връзките между таблиците.* Между две релационни таблици е възможно да съществуват връзки от типа **един към един (1:1)**, **един към много (1:M)** и **много към много (N:M)**.

- **Проектиране на процесите**

Проектирането на процесите е предпоставка за специфицирането на хардуера и проектирането и специфицирането на софтуера на системата. При разработването на софтуера, програмите се разглеждат като съставени от различни модули. Тези модули трябва да бъдат конструирани така, че всеки от тях да изпълнява отделна независима функция, да има възможност да се тества по отделно, да води до програмен код, който да бъде лесен за разбиране и за промяна в по-късен етап от проектирането ако това е необходимо. Също така, модулното проектиране трябва да улеснява добавянето на допълнителни модули, което ще допринесе за гъвкавостта на системата. Връзките между модулите могат да бъдат илюстрирани графично чрез структурна диаграма или използвайки графични нотации и езици за моделиране на процеси [31, 60, 75]. Структурната диаграма показва входът и изходът на всеки модул и кои други модули го извикват или се извикват от него. Структурната диаграма дава лесно разбираема картина на системата. В повечето случаи се изпускат потоците от данни между модулите.

Според [104] един програмен модул съдържа множество от изпълними инструкции за осъществяване на процес. Тези инструкции, на етап проектиране, може да бъдат специфицирани и впоследствие тази спецификация се трансформира в програмни инструкции на избрания език за програмиране.

В литературата са известни редица методи и средства за моделиране на процеси в информационни системи [97, 104]. Съществуват методи и средства за моделиране, структуриране, обработка и съхранение на информация. Също така има и методи за

планиране, моделиране, изпълнение, вграждане, тестване, документиране, настройка и т.н. на системи [19, 24, 67, 76].

При моделирането на процесите широко приложение в практиката намират различни модификации на **структурния метод**. Структурният метод възниква в края на 70-те години и се определя като метод за йерархично функционално декомпозиране [97]. Методът използва понятия като:

- Процес - предназначен е за преобразуване на данните. Може да бъде реализиран от човек, компютърна програма, компютърно устройство.
- Поток - описва движението на информацията между процеси, процес и хранилище, източник и процес, процес и получател.
- Хранилище - обект за съхраняване на информацията. Може да бъде компютърно (база данни, файл) или некомпютърно (папка) средство.
- Източник/получател на данни - хора, организации, системи, външни за изследваната система, които взаимодействат с нея. От значение е само информацията, която се обменя с тях.
- Речник на данните - съдържа информация (наименование, тип, структура и др.) за процесите, потоците, хранилищата, източниците и получателите на информация. Неговото предназначение е да систематизира елементите на модела, да не се допуска дублиране в наименованията им.

Структурният метод описва процесите чрез три вида диаграми [97]:

- Контекстна диаграма, чието предназначение е да определя обхвата на областта, която се анализира.

- Йерархична диаграма, чието предназначение е да представи структурата на процесите в системата [29, 34].

- Диаграма на потоците от данни, чието предназначение е да показва движението на информацията.

В последните години голяма популярност придобиват и **Workflow технологиите** като средство за моделиране, изпълнение и мониторинг на процеси в различни бизнес и ИТ области.

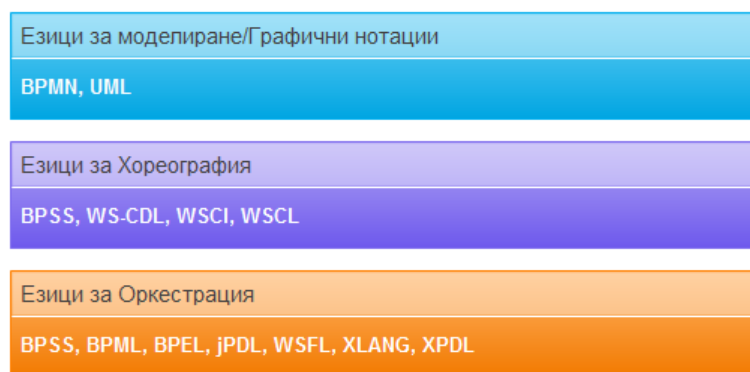
Workflow технологията с течение на годините става важна софтуерна област. Работният поток (Workflow) може да се дефинира като "последователност или стъпки, използвани в бизнес процесите" [42].

Широко приета е дефиниция за работен поток (Workflow), идваща от Workflow Management Coalition [134]: „Работен поток (Workflow) е компютъризирано улесняване или автоматизация на бизнес процесите, изцяло или частично” [82]. По-подробна дефиниция, е дадена в [83] „Workflow е система, която дефинира, създава и управлява изпълнението на работните потоци чрез използването на софтуер, вървящ върху един или повече машини за работни потоци. Машините са способни да

интерпретират процесните дефиниции, да взаимодействат с различни участници и където е необходимо да извикват външни програмни средства.”

Тези технологии могат да бъдат успешно използвани при моделирането и изпълнението на процеси в информационни системи, свързани с електронното обучение, както е посочено в [32]. Всички дейности в една система, например за управление на съдържанието, могат да бъдат също моделирани като процеси. С други думи електронна система за обучение може да се разглежда като система за управление на работния процес, изпълняваща процеси тясно свързани с дейности на електронното обучение.

Базирайки се на [66], може да се наблегне на следните функционални елементи: езици за моделиране и/или графични нотации, езици за хореография, езици за оркестрация. **Фигура 2.3** представлява функционален блок на гореспоменатите функционални елементи, както и достъпните технологии, обхващащи тези аспекти.



Фигура 2.3: Workflow технологии – функционални блокове

- **Проектиране на архитектурата на информационната система**

Архитектурата на информационната система е абстрактен модел, който обхваща отделните елементи на системата и техните взаимодействия.

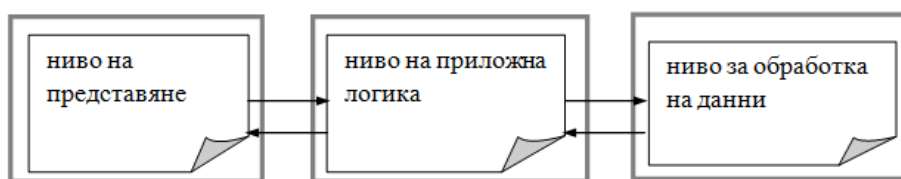
Архитектурата „Клиент-сървър” стои в основата на почти всички съвременни решения, използвани при създаване на *информационни системи*. Тази архитектура става популярна още през 80-те години на XX век. Тя включва клиентски процеси, обръщащи се към сървър процеси. Обработката на информацията представлява логическо взаимодействие на програмни модули, които са различно разположени и не е необходимо да се изпълняват в едно пространство на паметта [6].

Моделите, които предлагат клиент-сървър технологията се различават основно по начина на разпределение на изпълняваните алгоритмични функции между клиента и сървъра. Разделянето на алгоритмичната обработка на различни нива от програмната система довежда до създаването на клиент-сървър модели на две, три и четири йерархични нива [59, 65, 101].

Избор на клиент-сървър модел:

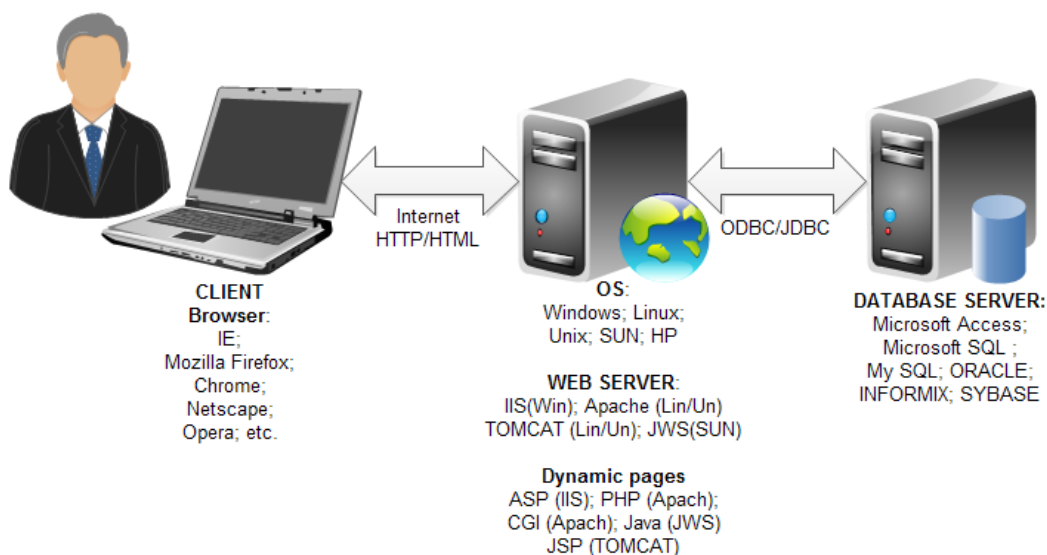
Клиент-сървър модел с три йерархични слоя

Този клас информационни системи реализират модела на клиент-сървър с три йерархични нива: ниво на представяне, приложно ниво и ниво за обработка на данни (**фигура 2.4**). Акцент се дава на динамичното генериране и предоставяне на информация. При отправено потребителско запитване сървърът стартира обработваща програма (ASP, CGI, JSP, PHP), която генерира HTML страница в реално време и я изпраща на клиента. Обработващата програма може да осъществява връзка с една или няколко бази от данни. Това позволява да се реализира интерактивна комуникация между потребителя и приложението. Събират се данни от клиента, автоматично се обновява информацията в приложението, изпълнява се търсене на информация в базата от данни.



Фигура 2.4. Клиент-сървър модел с три йерархични слоя

Главната особеност тук е въвеждането на нов системно-алгоритмичен модул. Това е базата от данни и системата за нейното управление. Базата от данни изпълнява функции, процедури и други дейности, свързани с поддържането и предоставянето на актуална информация от и за потребителите на информационната система. Системно-алгоритмичните и програмни средства използвани за реализирането на този модел са представени на **фигура 2.5**.



Фигура 2.5. Системно-алгоритмичните и програмни средства

- **Проектиране на потребителския интерфейс**

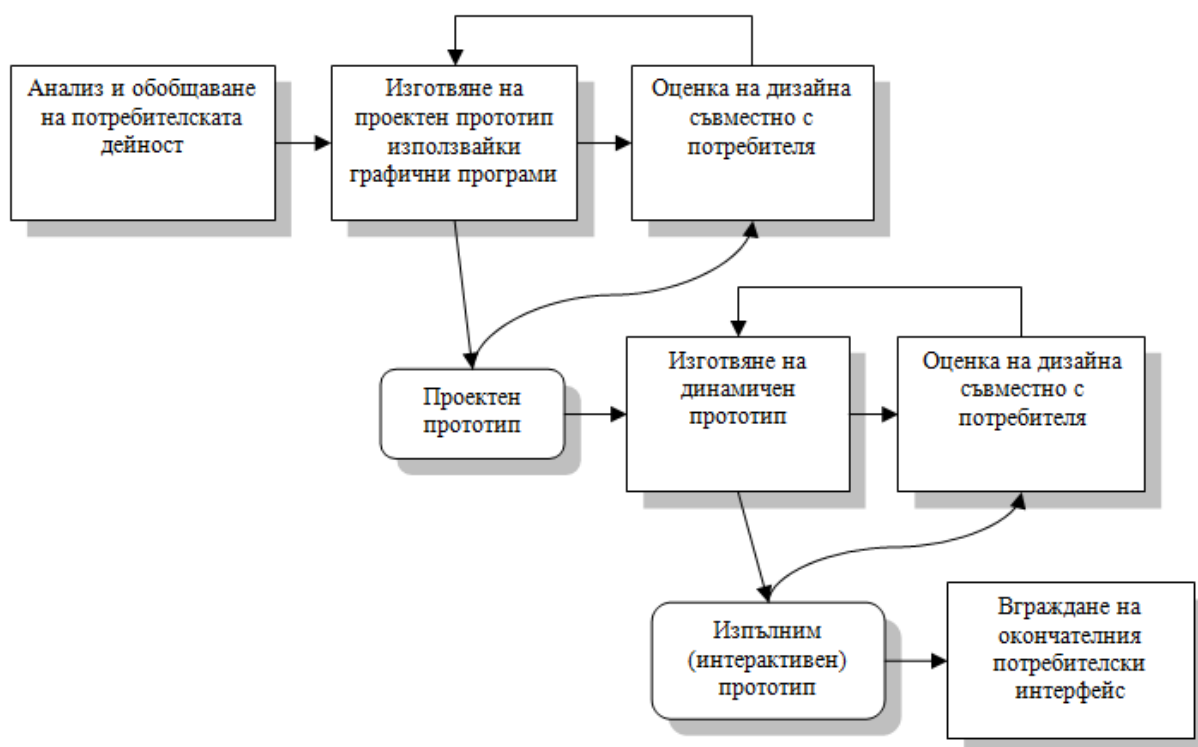
Добре разработеният потребителски интерфейс е решаващ за успеха на една система. Интерфейс, който е труден за използване, в най-добрия случай ще донесе много

потребителски грешки. В най-лошия случай, потребителите ще откажат да използват системата без значение на нейната функционалност. Ако информацията е представена по объркващ или подвеждащ начин, потребителите може да не разберат нейния смисъл. Те може да предприемат поредица от действия, които да изопачат данните или да доведат до грешки.

Като средство за комуникация между потребителя и компютъра се използва графичен интерфейс (Graphic User Interface, GUI). Предимствата на GUI са [152]:

- Относително лесен за изучаване и използване. Дори потребители с базови компютърни умения могат да се научат да използват интерфейса след кратко обучение.
- Потребителят има много прозорци за взаимодействие. Превключването към нова задача е възможно без да се губи от погледа информацията, генерирана от предходната.
- Бързина – работа върху целия екран позволява незабавен достъп до коя да е област или елемент.

На **фигура 2.6** се илюстрира итеративният процес за разработване на потребителски интерфейс.



Фигура 2.6. Итеративен процес за разработване на потребителски интерфейс

Изследователският подход е най-ефективният начин за разработване на потребителски интерфейс. Изготвянето на предварителни прототипи може да започне с моделиране на лист хартия или използване на графични програми преди да се премине към разработване на екранно-базиран интерфейс, който да симулира взаимодействие с потребителя.

В **таблица 2.1** са представени основни принципи, които са приложими за всички потребителски интерфейси [45, 95, 114].

Таблица 2.1 Основни принципи

Принцип	Описание
Пригодност (User familiarity)	Интерфейсът трябва да използва термини и концепции, извлечени от опита на хората, които най-много използват системата.
Цялостност (Consistency)	Интерфейсът би следвало да бъде цялостен по отношение на това, че подобните операции трябва да се активират по аналогичен начин.
Минимална изненада (Minimal surprise)	Потребителите не трябва никога да останат изненадани от поведението на системата.
Възстановимост (Recoverability)	Интерфейсът да включва механизми, които да позволяват на потребителите лесно да възстановят състоянието в нормален режим, след грешки.
Направляване на потребителя (User guidance)	Интерфейсът би следвало да предоставя смислена обратна връзка при възникване на грешки и да предлага контекстно-зависими средства за помощ.
Разнородност на потребителите (User diversity)	Желателно е да има подходящ набор от възможности за комуникация с различни категории потребители.

Всички интерактивни системи трябва да предоставят по някакъв начин информацията на потребителя. Това може да бъде директно извеждане на данните (както в текстообработваща програма) или чрез графично изобразяване. Добра практика е разделянето на софтуера, който се изисква за представяне на данните от самата информация.

Друг важен фактор при проектирането на потребителските интерфейси е това, че те би трябвало винаги да предоставят система за помощ (help) като съобщения, издавани от системата в отговор на потребителски действия; своевременна система за помощ (help); документация, предоставена заедно със системата.

Добра практика е и оценяването на интерфейса. Оценката му е процесът на преценка на полезността му и проверка дали покрива потребителските изисквания. Следователно той би трябвало да е част от нормалната проверка и процес на валидиране на софтуерните системи.

2.2.1.3 Разработване на системата

Съобразно с дефинираните изисквания към системата от етап "анализ и дефиниране" и разработения цялостен модел на информационната система от етап "проектиране" се пристъпва към реализирането (написването) ѝ в програмен вид. Създава се базата данни и прилежащите ѝ таблици, използвайки инструмент за администрация на базата данни. Написват се програмите на модулите на системата на избрания език.

2.2.1.4 Тестване на системата

Тестването се провежда в два етапа: в средата на разработване и в реална потребителска среда.

- Тестване в средата на разработване. През този етап се извършва и т.нар. модулно тестване. Всеки един от програмните модули се тества поотделно и се отстраняват грешките. След завършване на модулното тестване всички програмни модули се обединяват в една система, която в следващия етап се тества от потребителите в реални условия.

- Тестване в реална потребителска среда. Използва се интегралният подход за цялостно тестване на системата, а именно сглобява се цялата система и се тества като цяло. Като последна проверка се извършва тестване на системата в реална потребителска среда, като потребителите регистрират проблемите и ги съобщават на разработчиците, които внасят съответните промени.

2.2.1.5 Внедряване на системата

Внедряването се осъществява паралелно. Старата и новата система се използват паралелно за известно време, при което се сравняват получени резултатите при функционирането им. След като се установи, че всичко функционира нормално, старата система се спира и се преминава изцяло към използване на новата система.

2.2.1.6 Оценяване на системата

Оценяването на системата за Web-базирано обучение се извършва от експерти в областта на информационните технологии и потребителите на системата. Набляга се на оценяването на качеството и ефективността на системата по критерии, базирани на стандарти за качество на софтуерни продукти и дидактически показатели.

2.2.2 Описание на избора на технологии и средства за реализация

Системата за Web-базирано обучение, разработена за целите на настоящия дисертационен труд е реализирана с модерни, широко достъпни и доказали себе си технологии и средства.

- Избор на платформа

Система е базирана изцяло на софтуерни продукти с отворен код (Linux, Apache Web сървър, MySQL база данни и PHP програмен език)

- Избор на програмен език

Скрипт-езиците дават възможност на клиентите да изискват определена информация от техните сървъри и съответно, техните сървъри да получават важни входни данни от потребителя с цел обработка и визуализиране на желаните данни. PHP [56, 135] е скрипт-език с отворен код от страната на сървъра. Той работи в рамките на HTML документ, добавяйки към него възможността за генериране на съдържание по желание. PHP е един от най-популярните езици за създаване на интерактивни и динамични web сайтове. Причините са много - гъвкав, компактен, лесен за изучаване и използване, предлагащ всички възможности за създаване на динамичен, съвременен сайт. PHP предлага лесни възможности за изпълняване на сложни математически изчисления,

изпълняване на мрежови функции, възможности за обработка на електронна поща, работа с обикновени изрази и множество други. Най-голямата сила на езика е във възможностите му за работа с бази данни. Осигурена е поддръжка на най-разпространените бази данни, като MySQL, PostgreSQL, Oracle, Sybase, mSQL и други. Връзката с базата данни и работата с нея са облекчени и позволяват лесното създаване на ефективни динамични сайтове.

- Избор на СУБД

Проектирането на БД на основа на релационния модел има редица преимущества пред останалите модели като: независимост на логическата структура от физическото й потребителското представяне; гъвкавост на структурата на БД, т.е. конструктивните решения не ограничават възможностите на разработчика на БД да изпълнява в бъдеще най-разнообразни заявки.

Взимайки под внимание фактори като производителност, поддръжка, възможности, ограничения, цена и най-вече преимуществата (бързодействие, простота на използване, поддръжка на език за заявки, взаимодействие и безопасност, малък размер, работоспособност и цена, свободно разпространение), които предлага, за физическата реализация на базата от данни на информационната система е избрана релационната система за управление на бази данни (СУБД) MySQL [94, 136]. Системата е с отворен код и използването ѝ се подчинява на лиценза GPL. Може да се използва с голяма група програмни езици - C, C++, Eiffel, Java, Perl, PHP, Python и Tcl и има версии за Linux, UNIX и Windows.

MySQL използва Structured Query Language (SQL) - стандартизиран език за добавяне, прочитане и обработка на информация в базите данни. SQL е многофункционален език за работа с бази данни с управляващи конструкции за: създаване, промяна и изтриване на данни; дефиниране на данни (таблицы, колони); защита на достъпа до елементи от бази данни чрез работа с групи и индивидуални потребители; операции за управление на данните като създаване на архивни копия, блоково копиране и актуализация; и, най-важното, обработка на транзакции.

Едно от най-важните предимства на релационните БД е лекотата, с която се създават, четат и изтриват записите, както и лесната разширяемост.

MySQL е система за управление на база данни. Системата може да изпълнява заявките на неограничен брой потребители, издържа до 50 милиона записа, предлага лесна за използване система за управление правата на различните потребители, отлична скорост на обработка на заявките.

HTML

Информационната система World Wide Web е най-популярната услуга, достъпна чрез Internet. Тя позволява да се комбинират текст, аудио, видео, графика и анимация в мултимедийни документи. Хипервръзки в тези документи правят възможен достъп до

други документи, намиращи се в други сървъри по мрежата, без да имат директна връзка помежду си.

HTML (Hypertext Markup Language) [40, 137] е един от основните стандарти за преносимо структурно представяне на мултимедийни документи (и най-общодостъпен). Той представлява система от прости команди, които се включват в стандартните текстови документи с цел да се осигури интегрирана визуализация и е независим от платформи език за маркиране на текст.

- Избор на Web сървър

Web сървърът (Web server) е приложна програма, която позволява на даден компютър да предоставя информация на други компютри, под формата на страници с хипертекст. Предоставя информация на клиент под формата на HTML документ и за пренасянето на информацията се използва протокола HTTP. Обикновено Web сървърът позволява свързване с външни приложения, които да обработват информацията, преди тя да бъде изпратена до клиента, с помощта на т. нар. CGI скриптове. Тези скриптове могат да бъдат написани на един от много езици за програмиране. Най-популярните Web сървъри са Apache [138], Internet Information Server (IIS) [139], nginx [140], lighttpd [141].

След направен анализ на характеристиките на изброените Web сървъри и взетата в предвид направена от компанията Netcraft статистика за 2013 г. на най-добрите Web сървъри в интернет [142], за целите на дисертацията е избран Apache Web сървър.

Apache [79, 138] Web сървър представлява мощна и гъвкава реализация на Web – сървър, поддържаща всички съвременни стандарти в тази област. Допуска разширение на функционалността с помощта на модули, създадени от странични разработчици, с минимални загуби при настройване. Apache сървър е мултиплатформен, т.е. работи под управлението на операционни системи (ОС) Unix, GNU, FreeBSD, Linux, Solaris, Mac OS X, Microsoft Windows, OS/2, Novell NetWare и други платформи. Чрез него работят над 70% от сайтовете. Той се разпространява свободно с отворен код.

2.3 Възможност за прилагане на методи за многокритериална оптимизация към подход за оценка на качеството на система за Web-базирано обучение

Качеството на един софтуерен продукт е многомерна характеристика, в която съгласно приетия модел участват групи величини, разнородни по своята същност, по начина на получаване и по формата на изразяването си. Във всички случаи формалният модел за остойностяване на качеството трябва да взема под внимание тази многокритериалност и коректно да отчита взаимодействието и приоритетите на параметрите, обуславящи критериите.

Основните идеи и гледни точки относно концепцията на качеството на софтуерен продукт се групират в две по-главни тенденции [98]:

- Качеството се определя като съответствие с изискванията, специфицирани в заданието на проекта;
- Качеството се определя като съответствие с нуждите на потребителя – то се идентифицира независимо от каквито и да е измерими характеристики. Това означава, че качеството на продукта или услугата се определя като способност да отговори на очакванията на клиентите.

2.3.1 Определяне на тегловни коефициенти чрез ранжиране на целевите параметри

Компромисното оптимално решение при многоцелевата оптимизация силно зависи от приетите тегловни коефициенти W_j на различните критерии. Тези тегловни коефициенти характеризират степента на удовлетвореност от съответния показател.

За определяне важността на критериите и съответните им тегловни коефициенти засега няма строго формализирани методи. Най-често тези коефициенти се определят субективно. Сравнително надеждни тегловни коефициенти могат да се определят при допитване до множество специалисти и прилагане на обективни методи за оценка на субективните им мнения.

Един от методите за определяне на тегловни коефициенти по мнения на експерти е следният [35, 93, 109, 111]:

Таблица 2.2 Тегловна матрица [109]

Целеви параметър → Клиент ↓	y_1	y_2	...	y_j	...	y_m
1	a_{11}	a_{12}	...	a_{1j}	...	a_{1m}
2	a_{21}	a_{22}	...	a_{2j}	...	a_{2m}
3	a_{31}	a_{32}	...	a_{3j}	...	a_{3m}
⋮	⋮	⋮		⋮		⋮
λ	$a_{\lambda 1}$	$a_{\lambda 2}$	...	$a_{\lambda j}$	...	$a_{\lambda m}$
⋮	⋮	⋮		⋮		⋮
R	a_{R1}	a_{R2}	...	a_{Rj}	...	a_{Rm}
$\sum_{\lambda=1}^R a_{\lambda j}$						
Δ_j						
V_j						
W_j						

На R специалисти се предлага да дадат своето мнение за m целеви параметъра чрез анкетен лист, като ги подредят по важност от 1 до m . Препоръчва се в анкетния лист целевите параметри да се разположат в случайна последователност и без номерация пред тях. Получените резултати се записват в тегловна матрица (**табл. 2.2**). Всяко число в тегловната матрица $a_{\lambda j}$ определя теглото (ранга), което специалистът λ приписва на целевия параметър j .

При наличие на "свързани рангове" (съвпадащи рангове), т. е. класиране на няколко целеви параметъра с общ ранг (на едно и също място), ранговата матрица се нормализира така, че сумата $a_{\lambda j}$ за всеки ред да бъде:

$$\sum_{j=1}^m a_{\lambda j} = S_p \quad (12)$$

където

$$S_p = \frac{m(m+1)}{2} \quad (13)$$

Пресмятането на тегловни коефициенти се прави, ако има съгласуваност в мненията на специалистите. Съгласуваността се проверява, като се изчисли коефициента на конкордация (съгласие) w_k по методите на рангова корелация [35, 36]:

Изчисляване на коефициента на конкордация

- при отсъствие на съвпадащи се рангове

$$w_k = \frac{12 \sum_{j=1}^m \Delta_j^2}{R^2(m^3 - m)} \quad (14)$$

където

$$\Delta_j = \sum_{\lambda=1}^R a_{\lambda j} - S_{cp}, \quad (15)$$

$$S_{cp} = \frac{R(m+1)}{2} \quad (16)$$

S_{cp} е средната сума от всички тегла;

Δ_j - отклонението на сумата от теглата за всеки целеви параметър от средната сума.

- при наличие на съвпадащи се рангове

$$w_k = \frac{12 \sum_{j=1}^m \Delta_j^2}{R^2(m^3 - m) - 12R \sum_{\lambda} T_{\lambda}}, \quad (17)$$

където

$$T_{\lambda} = \frac{1}{12} \sum_{t_{\lambda}} (t_{\lambda}^3 - t_{\lambda}) \quad (18)$$

t_{λ} е брой на поотделно повтарящите се рангове за всеки специалист.

Очевидно е, че при отсъствие на повтарящи се (съвпадащи) рангове вторият член в знаменателя на (17) отсъства и формула (17) става еднаква с (14).

Проверка на коефициента на конкордация на значимост

Коефициентът на конкордация може да се изменя от 0 при пълна несъгласуваност в мненията до +1 при пълна съгласуваност. Оценката на значимостта на изчисления

коэффициент w_k може да стане по χ^2 -критерий при $m \geq 7$ или по Z - критерия [16] при $m < 7$.

- При $m < 7$ се изчислява величината:

$$Z_{\text{изч}} = \frac{1}{2} \ln \frac{(R-1)w_k}{1-w_k} \quad (19)$$

Ако $Z_{\text{изч}} > Z_{\text{табл}}(\alpha, v_1, v_2)$, w_k е значим. $Z_{\text{табл}}$ се взема от таблица при степени на свобода

$$v_1 = m - 1 - \frac{2}{R}, \quad (20)$$

$$v_2 = (R - 1)v_1, \quad (21)$$

- При $m \geq 7$ се изчислява:

$$\chi_{\text{изч}}^2 = R(m - 1)w_k, \quad (22)$$

Коефициентът на конкордация w_k е значим, ако

$$\chi_{\text{изч}}^2 > \chi_{\text{табл}}^2(\alpha, v) \quad (23)$$

$\chi_{\text{табл}}^2$ се взема от таблица при степени на свобода $v = m - 1$ и ниво на значимост α

Изчисляване на тегловни коефициенти

При условие, че има съгласуваност в субективните мнения на специалистите, могат да се изчислят тегловните коефициенти [93, 109].

$$W_j = \frac{v_j}{\sum_{j=1}^m v_j}; j = 1, 2, \dots, m, \quad (24)$$

където

$$V_j = \frac{(Rm - \sum_{\lambda=1}^R a_{\lambda j})}{Rm - R} \quad (25)$$

За така получените тегловни коефициенти е изпълнено условието

$$\sum_{j=1}^m W_j = 1. \quad (26)$$

Въпреки, че необходимостта от решаването на множество практически задачи с векторен критерий е довела до развитието на много методи за намирането на оптимални решения, като най-близко до целите на дисертацията, включващи изчисляването на качеството и ефективността на информационната система на база определени критерии се приема да се използва като основа метода за определяне на тегловни коефициенти по мнения на експерти.

2.4 Изводи

За разработването на една информационна система е необходимо процесът на разработка да премине през определен брой отделни етапи (фази). Информационната система за организиране и управление на Web-базирано обучение е реализирана следвайки определен брой етапи, съответстващи на основните дейности на процеса за разработване на един програмен продукт. Етапите, през които преминава разработването на системата са шест. Те включват дейности по анализиране, проектиране, разработване, тестване, внедряване и оценяване на системата за обучение. Към всеки един етап е представено кратко описание на задачите, които се изпълняват последователно.

На етап *анализ* на изискванията към системата целта е изясняване на това какво е необходимо за реализирането на нейните функции. Осъществява се декомпозиция на функциите на системата на техните съставни елементи и се създават логически модели на процесите и потоците от данни, необходими за изпълнението им.

На етап *проектиране* се разработва цялостен модел на системата за Web-базирано обучение. Определят се методите и средствата, използвани при реализирането на системата. Мотивиран е изборът на хардуер, софтуер, база от данни, архитектура, комуникациите, взаимодействието с потребителите при системата.

На етап *разработване* реализирането на системата е съобразено с предходните два етапа. Изградена е практически в съответствие с техните изисквания, методи и средства. Извършва се създаването на базата от данни и се написват програмните модули на избрания програмен език.

На етап *тестване* системата преминава през няколко стъпки. Първо всеки един програмен модул се тества поотделно и се отстраняват грешките, след което системата се сглобява и се извършва цялостно тестване. Накрая като последна проверка се извършва тестване на системата в потребителска среда.

На етап *внедряване* се осъществява преход от старата към новата, като внедряването се реализира паралелно.

На етап *оценяване* се предлага възможност за прилагане на методи за многокритериална оптимизация към подход за оценка на качеството на система за Web-базирано обучение. Направен е анализ на тези методи, и на основата на този анализ е предложен подход, чрез който се прави количествена оценка на качеството и ефективността на системата. **На базата на тази оценка се определят параметрите които трябва да се подобрят с цел оптимизиране на процесите в системата с оглед удовлетворяване на изискванията на потребителите.** Подходът може да се прилага за различни софтуерни продукти в областта на Web-базираните приложения, като се отчетат особеностите на всяка от тях.

Резултати от изследванията по тази глава са публикувани в [6] от публикациите по дисертацията.

Глава трета: Разработване на информационна система за Web-базирано обучение

3.1 Увод

Системите за организиране и управление на Web-базирано обучение, или за по-кратко, ползващи названието “система за Web-базирано обучение”, са многофункционални. Тези системи осигуряват комуникация между участниците в учебния процес, както и възможност за управление на учебния процес. Проблемът как да се структурира Web-базираният курс и да се представи съобразно избран модел, как да се съхранява в правилно структурирана Web-интегрирана база данни, е изключително актуален. Като резултат от научните изследвания в голям брой университети и компании са разработени много такива системи. Една от тези системи е и реализираният сайт, който е основа за създаване и представяне на курсове по различни дисциплини в Медицински университет – София. Необходимостта от създаването на собствена система е основано на няколко причини: високите цени на чужди системи за организиране и управление на обучението (като Blackboard и Lotus например), които са несъпоставими с нашите условия; липса на възможност за гъвкаво обновяване на ползваната система с развитието на нови информационни технологии. Закупените или получени чрез лиценз системи не могат да се променят. Въпреки универсалния характер на повечето системи, на практика съществуват характерни разлики между обучението по инженерни, хуманитарни, медицински и др. дисциплини, което оправдава и налага разработването на собствени системи, съобразно спецификата на дисциплините. Например в медицинските дисциплини е важно решаването на ситуационни задачи (клинични казуси). И не на последно място езиковата бариера. Системите са ориентирани основно към английски говорещите страни.

В настоящата глава от дисертационния труд се описва реализирането на софтуерно приложение като Web-базирана система за обучение. Системата е насочена към преподавателите от различни медицински дисциплини в Медицински университет - София (МУС), за да ги улесни в публикуването на традиционни и интерактивни средства за обучение. Системата има два интерфейса: административен, предназначен за използване от преподавателите и потребителски, предназначен за използване от студентите. Интерфейсът за студенти включва три основни модула (Лекции, Тестове и Виртуални пациенти), също така модул Виртуални модели и модули за регистриране на потребители и търсене по ключова дума или изрази и редица други допълнителни секции и под-секции. Административният интерфейс включва множество опции и форми, които позволяват дори и на преподаватели с ограничени компютърни познания с лекота да създават, редактират и публикуват лекции, тестове, симулации на казуси както и да манипулират съдържанието и на останалите модули в системата.

3.2 Реализация на системата

Реализирането на системата следва предложените етапи на изпълнение от втора глава на дисертационния труд. Системата спада към категорията управление на учебно

съдържание поради спецификата на изискванията от страна на институцията, която я използва.

3.2.1 Дефиниране на изискванията към системата

Като резултат от направения анализ на литературни източници и проучването и разглеждането на аналогични програмни системи, както и взетото в предвид мнение на независими експерти в областта на информационните технологии и експерти от институцията, която предлага системата за Web-базирано обучение, се формулират следните изисквания, включващи характеристики на качеството на софтуерните продукти и изпълняващи функции, характерни за универсална система за управление на обучението:

- Степен на съответствие между изискванията на потребителя и разработената система;
- Способност на системата да изпълнява предвидените функционални изисквания при евентуални проблеми, възникващи по време на използването ѝ;
- Възможност за разширяване или модифициране на системата по отношение на определени параметри като например повече данни, други платформи, увеличение броя на потребителите и др.;
- Степен на независимост към хардуер или операционна среда;
- Възможност за отстраняването на грешки на системата по време на експлоатацията ѝ и поддържането ѝ в актуално състояние;
- Възможност за усвояване и експлоатация на системата с минимум усилия, т.е. софтуер, който е лесен за изучаване, стабилен и адекватен за нуждите на потребителите;
- Възможност за регистрация на студенти и ползване на курсовете в системата като източник на знания. Възможност за регистрация на преподаватели и ползване на системата за създаване, редактиране и публикуване на учебни материали;
- Възможност за комбиниране и многократно използване на учебните материали. Възможност за различни режими на работа в различните модули;
- Възможност за разнообразни типове въпроси и автоматизирано оценяване на отговорите. Ползване на модули за създаване и проиграване на тестове;
- Възможност за публикуване на всички документи, необходими за организиране на учебния процес;
- Възможност за комуникация между потребителите;
- Възможност за различен достъп до модулите за използване на учебни материали в системата. Възможност за защита на базата данни и авторските материали в системата от неоторизиран достъп;
- Възможност за развитие на системата с развитие на технологиите. Системата да позволява осъвременяване, адаптиране и развитие от специалисти;
- Да е с добро външно оформление, с приятелски и интуитивен интерфейс;
- Да е защитена от неоторизиран достъп;
- Да бъде разработена, като се ползва свободен софтуер с отворен код;
- Да има възможност за търсене по различни категории;

- Поддържане на повече от един език;
- Да е независима от специфичното съдържание на информацията или дисциплината.

3.2.2 Създаване на базата от данни

За физическата реализация на базата от данни на системата за Web-базирано обучение е избрана релационната система за управление на бази данни (СУБД) MySQL.

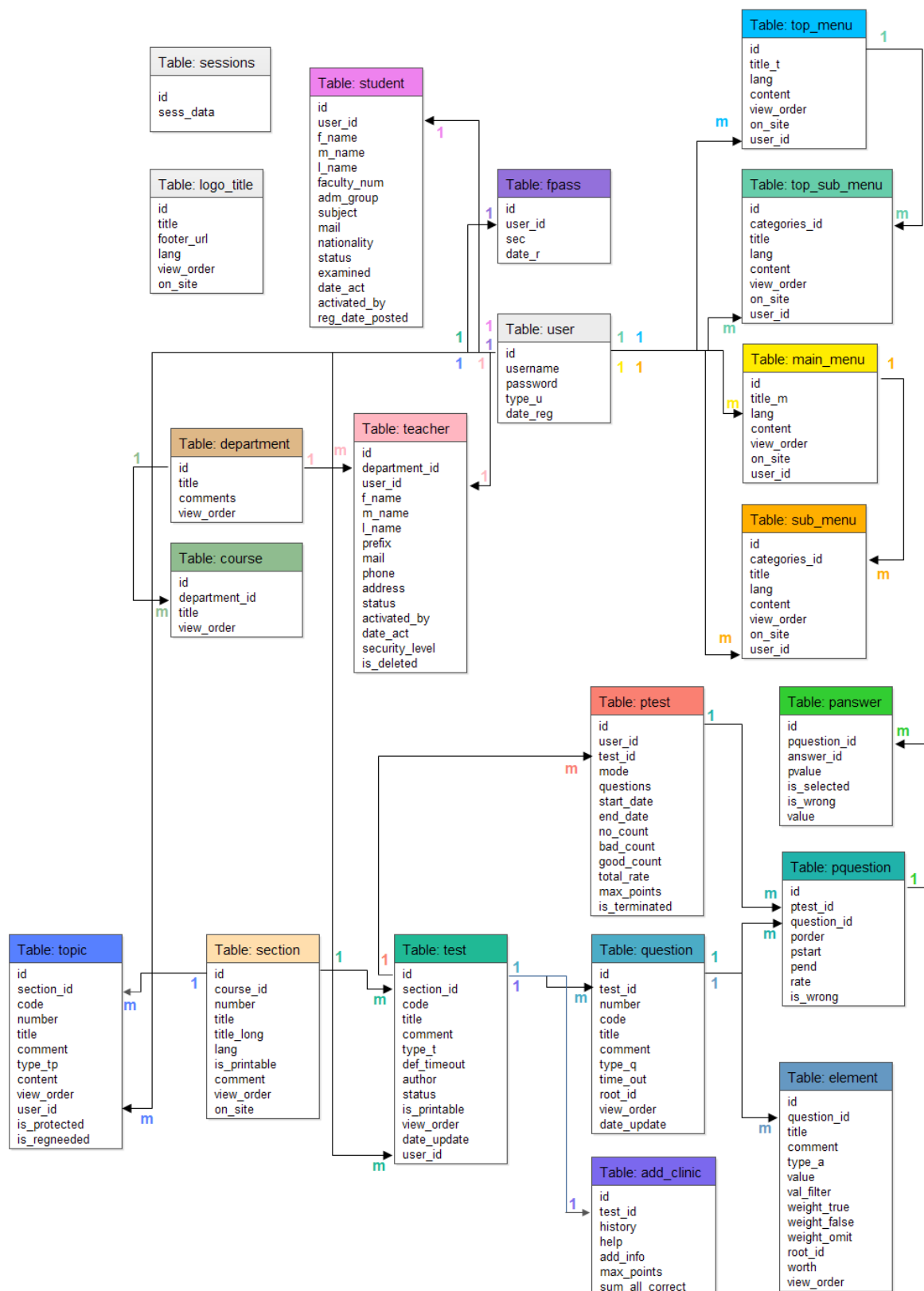
Цялото съдържание, управлявано и публикувано от системата се съхранява в база данни „templatetests”. Релационният ѝ модел е представен на **фигура 3.1**. Визуализирани са таблиците и връзките между тях.

Всяка таблица от базата данни има уникален идентификатор на ред, който представлява колона, използвана за разграничаване всеки отделен ред от останалите редове в таблицата. Нарича се *първичен ключ (primary key)*. Първичният ключ може да бъде прост или съставен. В случая се използва прост ключ. Този ключ е създаден от една колона, съдържаща уникални стойности за всеки ред от таблицата. Всяка таблица има само един първичен ключ, дори когато няколко колони или комбинации от колони съдържат уникални стойности (наречени *алтернативни ключове*).

Въпреки че първичните ключове са предназначени за отделните таблици, те са особено необходими, когато се създават отношенията между различните таблици в базата данни. Някои от таблиците в базата данни съдържат *външен ключ. Външният ключ (foreign key)* е колона или група от колони в дадена таблица, представляваща връзка с друга таблица посредством нейния първичен ключ. Чрез дефинирането на външните ключове в някои от таблиците в БД на системата се цели да се моделират връзките между обектите. *Релацията (relationship)* или връзката между таблиците, е базирана на първичен ключ от едната таблица и външен ключ от другата таблица.

Съществуват три типа релации между таблиците: релации 1:1 “едно към едно”, релации 1:М “едно към много”, релации М:М “много към много”. В случая при проектирането на базата данни на системата, моделът е нормализиран и не съдържа връзки М:М (много към много). Те са разрешени посредством свързващите таблици. Използвани са релации 1:1 и 1:М.

В **Таблица 3.1** се съдържа информация за базата данни на системата и нейните 21 на брой таблици. В базата данни има информация за регистрираните потребители в системата, за катедрите и курсовете, за разделите на всеки курс, за съставните части на разделите: Лекции, Тестове и Виртуални пациенти, за въпросите в тестовете и симулациите, както и за техните отговори и други помощни елементи. Системата позволява използването на различни типове въпроси в тестовете с различна комбинация от съставни елементи.



Фигура 3.1: Релационен модел на базата данни на системата

Таблица 3.1: Кратко описание на съдържанието на БД на системата

Потребители на системата	<u>Table user</u> – таблицата съхранява потребителските данни. Има четири типа потребители: гости, студенти, преподаватели без административни права и преподаватели с административни права. <u>Table student</u> and <u>Table teacher</u> – таблиците включват данни за имена и пароли на потребителите, лична информация като име, факултетен номер на студента, позиция и катедра за преподавателя, e-mail, телефон, адрес, и др. <u>Table fpass</u> – таблицата съдържа информация в случай на забравена парола.
Раздели на курсовете	<u>Table section</u> – в тази таблица се въвеждат данни за различните раздели, които се създават за определен курс в системата. Всеки раздел притежава пълно и съкратено наименование, номер на раздел, код, коментар и опция за избор дали разделът да може да бъде принтиран или не, също дали е видим или не.
Лекции	<u>Table Topic</u> – в тази таблица се въвеждат данни за лекционните точки, които се създават за всеки раздел от курса. Има четири нива на йерархия при процеса на манипулиране на данни: заглавие на раздел, заглавие на лекция, заглавие на тема от лекцията и заглавие на подтема от лекция. Всяка една тема принадлежи на един раздел и включва следните полета: заглавие, код, номер, коментар, тип (със или без съдържание) и текст на съдържанието.
Тестове/ Виртуални пациенти	<u>Table test</u> , <u>Table add_clinic</u> – в тези таблици се въвежда обща информация за въпросите от модули „Тестове и Виртуални пациенти”. Всеки тест или случай принадлежи на един раздел и включва следните полета: заглавие, код, автор, статус (в процес на разработка, демонстрационен, изпитен, реален, архивиран), време за решаване на един тест, опция за избор дали разделът да може да бъде принтиран или не, също дали е видим или не и коментар.
Въпроси	<u>Table question</u> , <u>Table ptest</u> и <u>Table rquestion</u> – в тези таблици се въвеждат въпросите от модули „Тестове и Виртуални пациенти” и техните компоненти. Всеки въпрос включва следните полета: номер, текст на съдържанието, код, тип, време за решаване на един тест и коментар.
Елементи на въпросите	<u>Table panswers</u> , <u>table element</u> – Всеки въпрос се състои от елементи и отговори. В тези таблици се съхранява информация за елементите и отговорите на въпросите от модули „Тестове и Виртуални пациенти”.
Катедри	<u>Table department</u> - включва информация за катедрите в институцията, ползващи системата.
Курсове	<u>Table course</u> – включва информация за курсовете, които всяка катедра предоставя на студентите.
Менюта на графичния интерфейс	<u>Table main_menu</u> , <u>Table top_menu</u> , <u>Table sub-menu</u> , <u>Table top_sub_menu</u> , <u>Table logo_title</u> – включва информация за заглавията на главните и второстепенните менюта на графичния интерфейс и името на дисциплината, която използва системата.

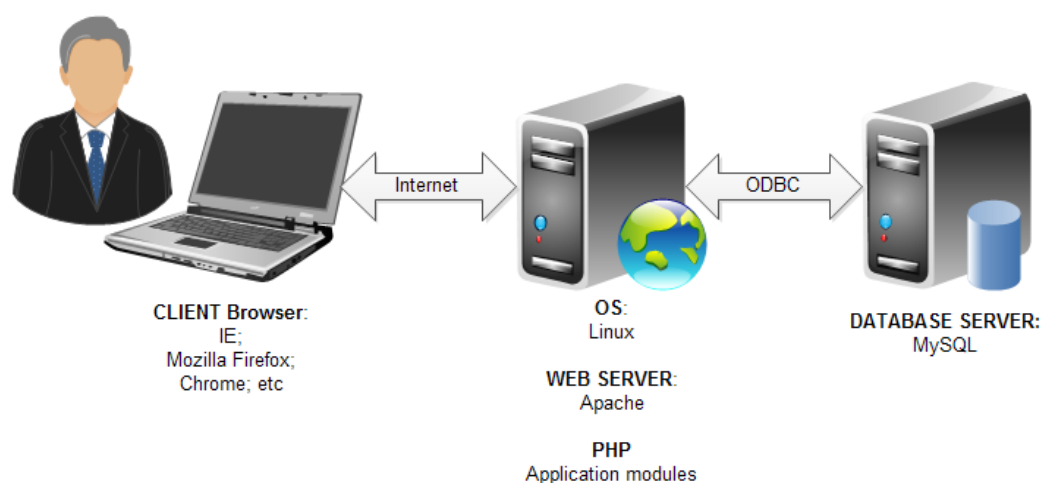
3.2.3 Проектиране на Web архитектурата на системата

Трислойният модел се счита за един от шаблоните на софтуерна архитектура. Тя представлява клиент-сървър концепция, при която потребителският интерфейс, бизнес логиката на приложението и работата с базата данни се извършват от независими модули. Освен другите предимства на модулния софтуер се счита, че трислойният модел позволява промяна на технологиите или друго обновление на всеки от модулите,

без това да повлиява на останалите слоеве, като е необходимо да се спазят само утвърдените интерфейси между модулите.

Принципът на действие на системата е базиран на архитектурния модел „клиент-сървър” с три взаимосвързани йерархични нива: ниво на представяне (клиент, който функционира чрез общодостъпните браузери), приложно ниво (Web сървър и прилежащите програми) и ниво за обработка на данни (реализирано е от системата за управление на бази от данни).

Фигура 3.2 представя графично връзката между отделните нива.



Фигура 3.2: Web софтуерна архитектура на системата

За реализацията на първото функционално ниво – потребител – се използва обикновен браузер (например Internet Explorer, Mozilla Firefox, Chrome и др.), който се свързва с главната сървърска част на приложението. В тази част от информационната система комуникацията се извършва по HTTP протокол.

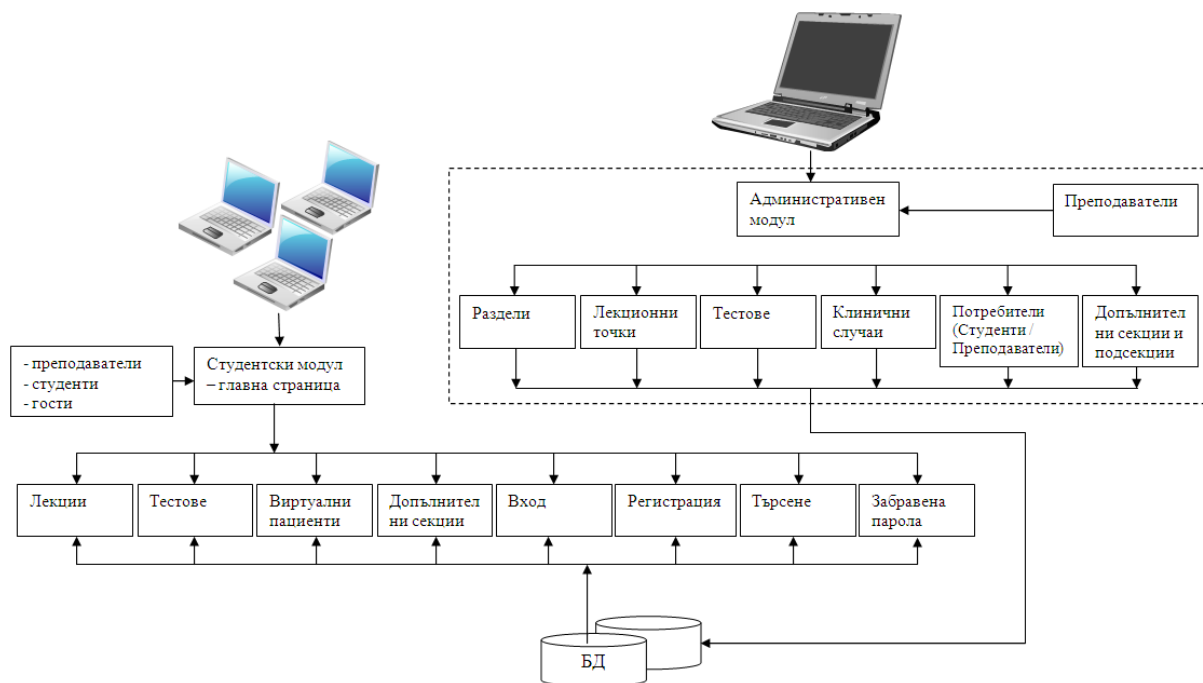
Второто ниво е изградено под Linux операционна система и се реализира чрез Web сървъра, който обслужва информационната система. Набляга се на динамичното генериране и предоставяне на информация. При отправено потребителско запитване сървърът стартира обработваща програма (ASP, CGI, Servlet, PHP), която генерира HTML страница в реално време. Избран е Web сървърът Apache, а като обработваща програма се използва PHP интерпретатор за програмните модули от страна на сървъра.

Третото ниво е нивото на базата данни. Използва се MySQL като база от данни за съхранение на информацията. За създаването и администрирането на базата данни се използва приложението phpMyAdmin.

3.2.4 Проектиране на системата като съвкупност от модули

Системата за Web базирано обучение се състои от два интерфейса – административен и потребителски, представени на **Фигура 3.3**. Всеки един от тях е удобен, лесен за използване и интерактивен за всички потребители - гости, студенти, преподаватели,

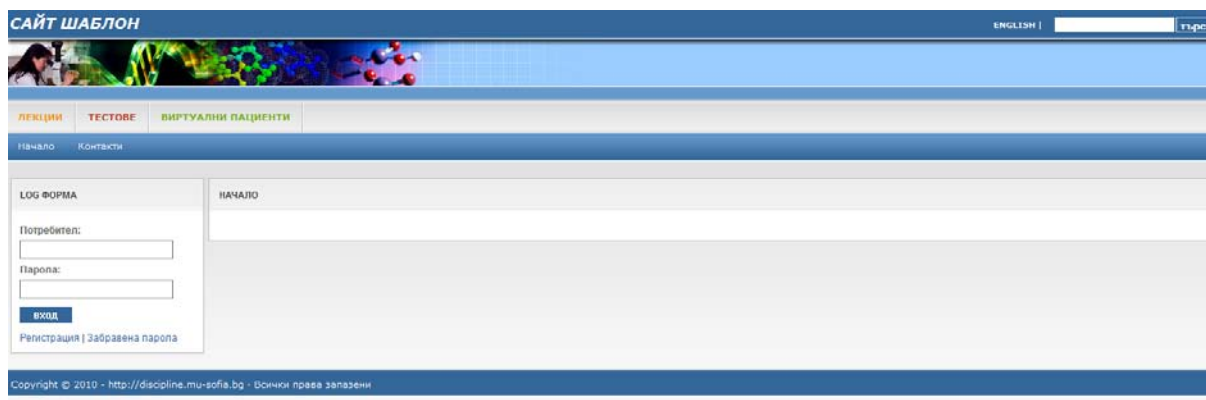
администратори. Тя е гъвкава и отворена - независима от операционна система, съвместима с широко използваните браузъри, съвместима с други Web приложения, плъгини и др. Системата е независима от конкретното съдържание на дисциплината и осигурява бърз и лесен достъп до всяка част от нея.



Фигура 3.3: Структурна схема на модулите на системата и връзката между тях

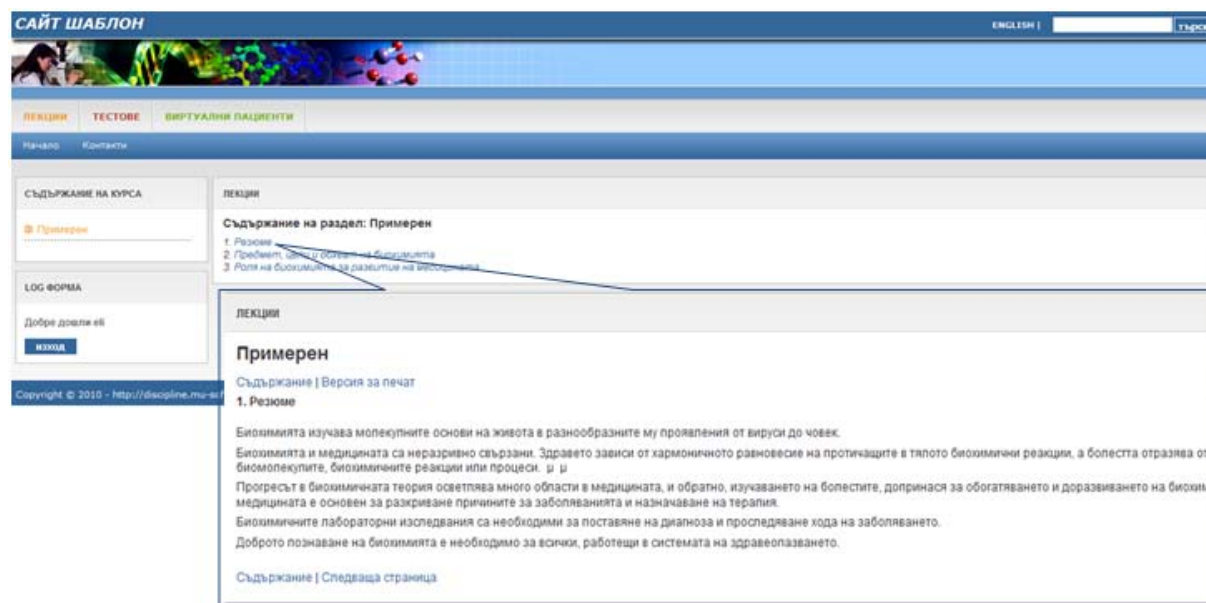
3.2.4.1 Описание на потребителския интерфейс на системата

Интерфейсът, предназначен за използване от студентите се състои от три основни модула - "Лекции", "Тестове" и "Виртуални пациенти". Всички страници в тези модули са динамично оформени. Няма ограничения за броя при създаването на нови модули и в сайта на медицинска биохимия е добавен модулът „Виртуални модели”. Освен това е предоставена възможност за търсене, регистрация на потребители с различни права и контролиран достъп до системата. Също така е възможно да се добавят допълнителни секции и под-секции, съдържащи полезна информация, като речник по дисциплината и помощни материали: конспекти, разписания, литература, списък на преподаватели и др. Този интерфейс предоставя на студентите възможност за достъп до лекциите като източник на знания, достъп до тестове като (само)-оценка на теоретичните знания и достъп до решаване на клинични казуси като възможност за добиване и подобряване на професионалните си умения. На **фигура 3.4** е представена главната страница на потребителски интерфейс.



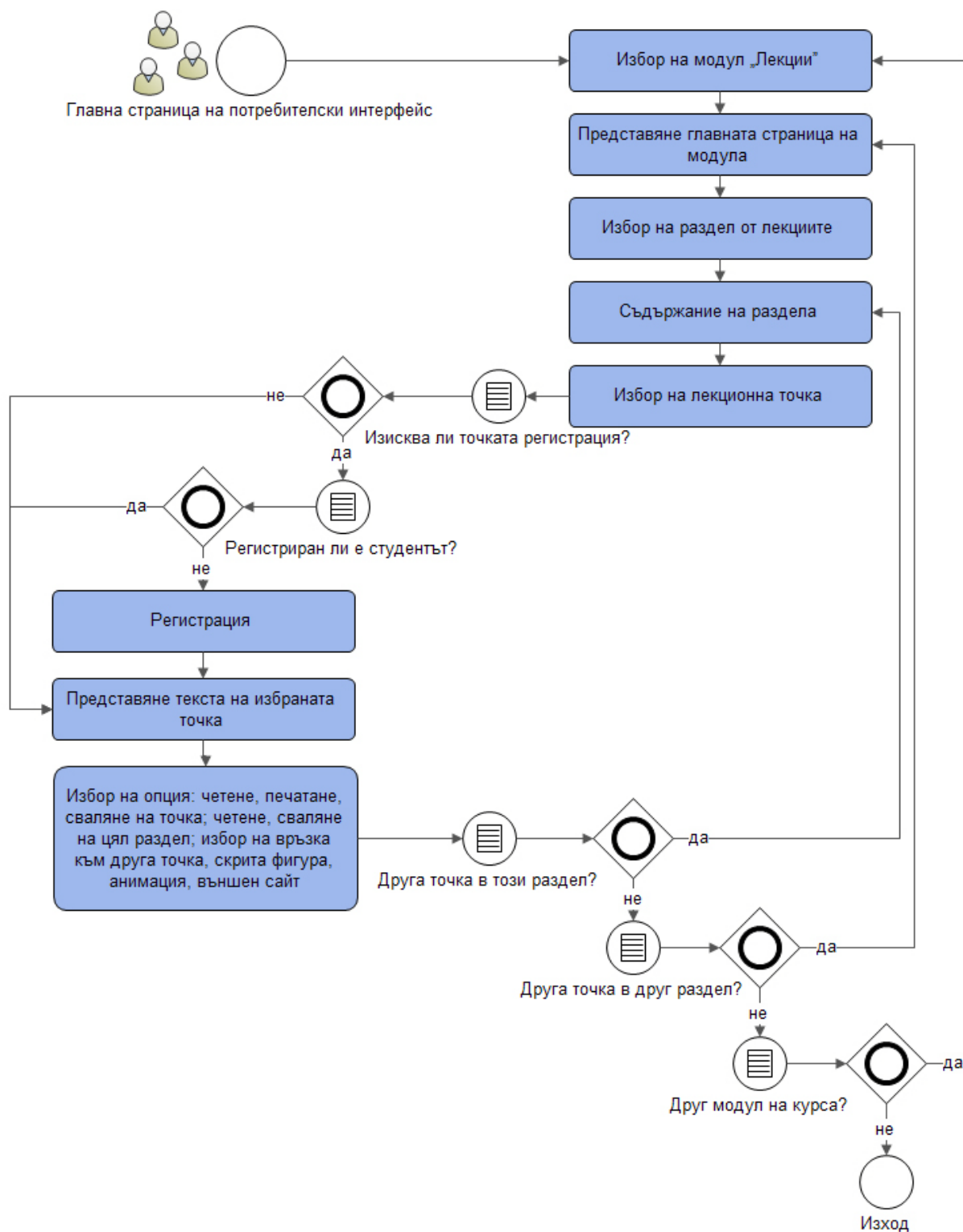
Фигура 3.4 Начална страница на потребителски интерфейс

Модул „Лекции” – Този модул служи като източник на основни знания. При избор на модул „Лекции” (фигура 3.5), в лявата част на екрана, потребителят може да види списък със заглавията на разделите (съдържание на курса). Всяко заглавие на раздел е връзка към съдържанието му, което се визуализира в дясната част на екрана. Съдържанието на всеки раздел включва лекции. Всяка лекция се състои от теми и подтеми. Всяко заглавие на тема е връзка към съответната лекционна точка на раздела. Студентите могат лесно да намерят необходимата тема и да я използват, за да обогатят своите знания. Също така има възможност за разпечатване на целия раздел.



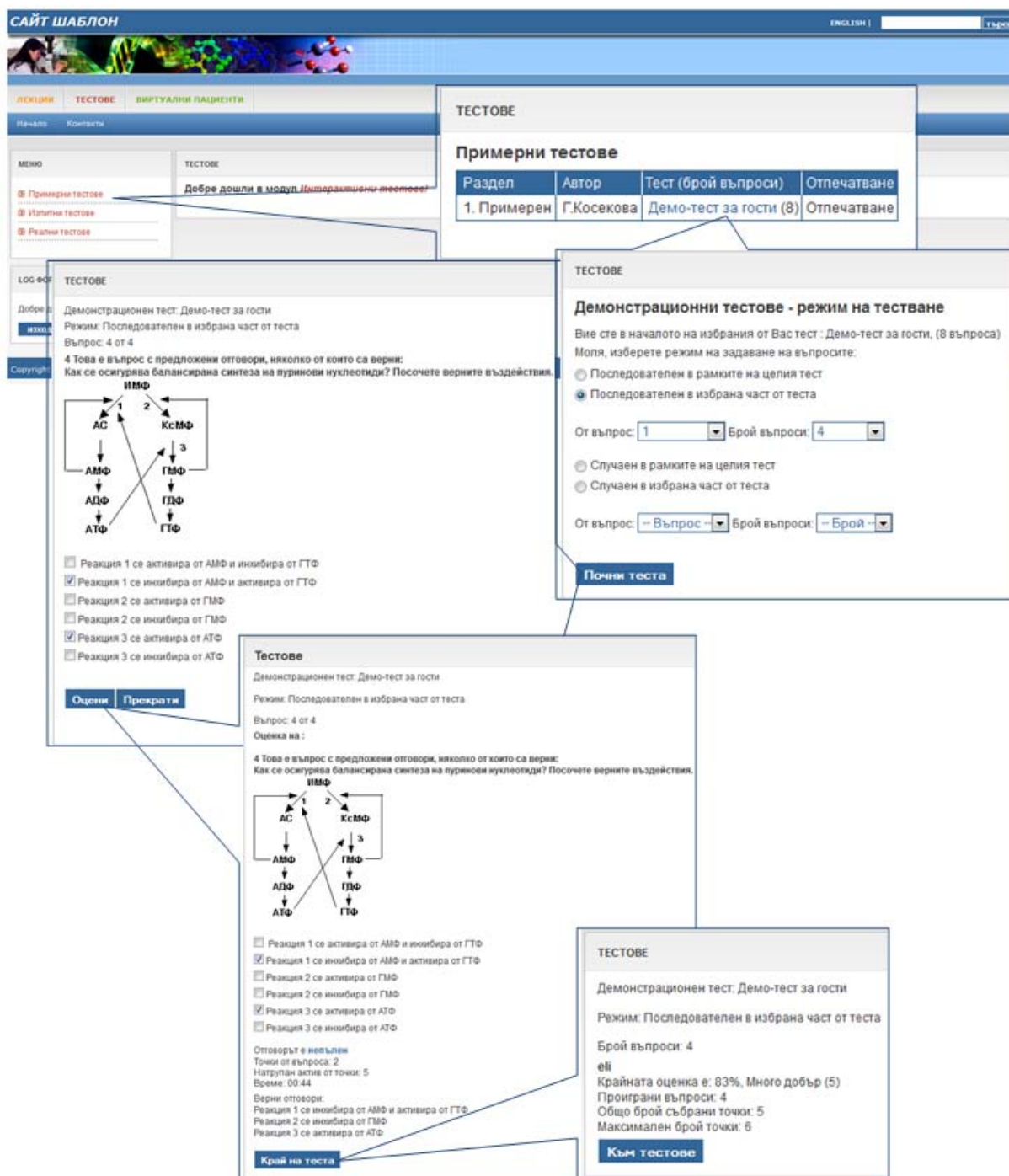
Фигура 3.5 Модул „Лекции”

На **Фигура 3.6** е представен *сценарий на действието на модул „Лекции”*. Този сценарий описва стъпките на действие, които потребителят на системата трябва да следва когато желае да получи информация за желаната лекционна тема. Отделните стъпки се обединяват в един общ процес, който впоследствие е следван при написването на програмния код, отнасящ се за конкретния модул. За моделирането и графичното визуализиране на процеса е използвана BPMN диаграма за логическо проследяване на отделните му стъпки.



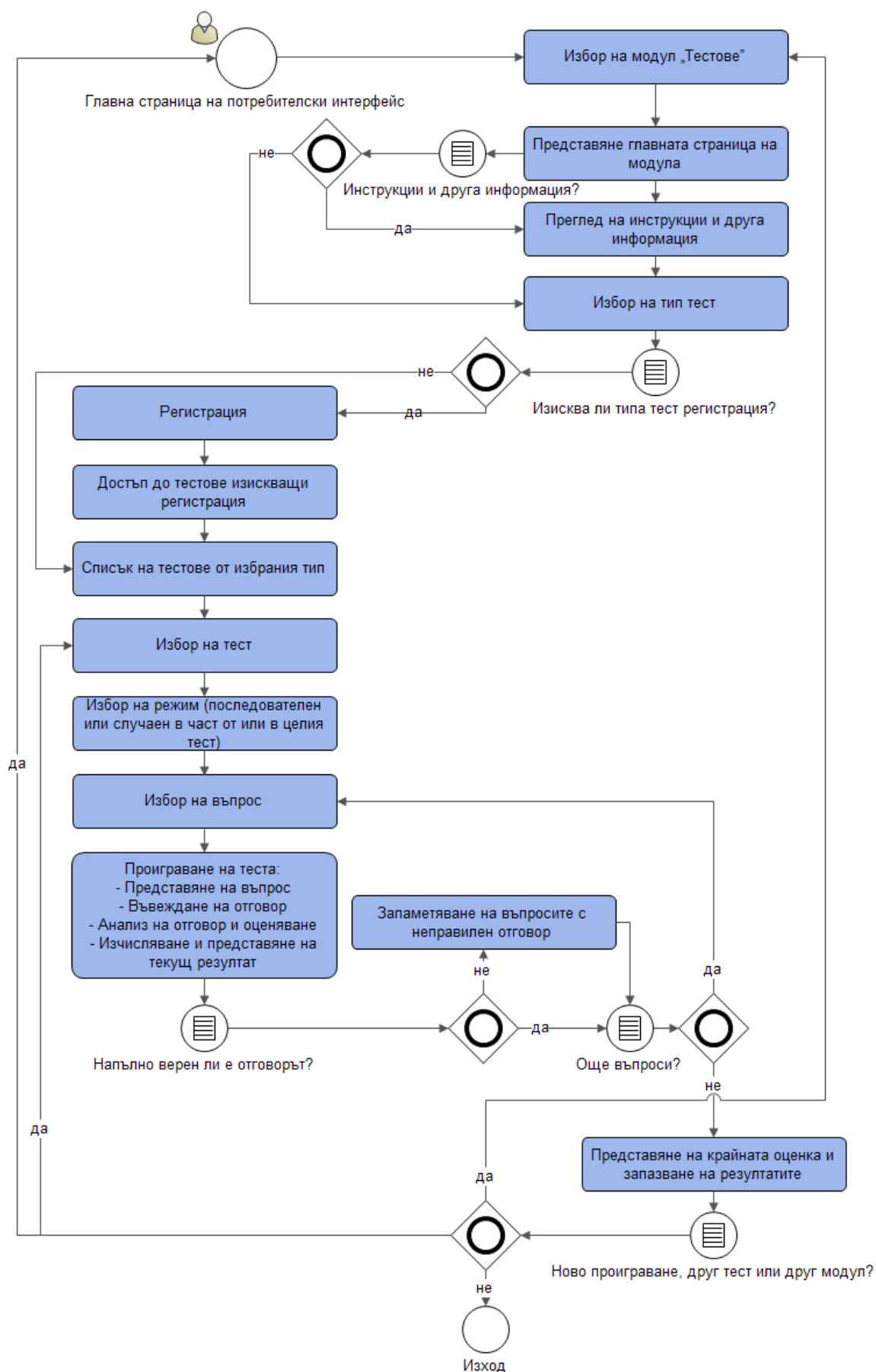
Фигура 3.6: Сценарий на действие на модул „Лекции”

Модул „Тестове” – Този модул дава на студентите възможност за (само) оценка на теоретичните им знания (**фигура 3.7**). Главната страница на този модул дава възможност за избор между няколко типа тестове - примерни, изпитни и реални. При избор на реален или изпитен тест, програмата изисква регистрация. За да се стартира и проиграе изпитен тест, освен регистрация, е нужно на студента да му бъде дадени специално разрешение от преподавател. При условие, че студентът е регистриран и неговият акаунт е активиран от преподавател, в дясната част на екрана се визуализира списък с тестове от избрания тип и студентът може да направи своя избор. Заглавието на тестовете е дефинирано като въпрос с предложени отговори. Тестовете позволяват различни режими на представяне на въпросите: последователно или случайно в рамките на целия тест и последователно или случайно в избрана част от теста. Разнообразните типове въпроси (избор от предложени отговори, запълване на празнини, ранкиране, намиране на съответствие между елементи на две групи, изискващи свободен отговор) се оценяват автоматично от системата. Различават се четири типа отговори: верен, непълен, неточен, погрешен. Теглата на верните отговори са положителни числа, а на погрешните – отрицателни. Това осигурява по-добра диференциация на отговорите. Оценката на въпроса се формулира като аритметична сума от теглата на избраните отговори. След като студентът прави своя избор на отговор след всеки въпрос се представя оценка, коментар, верните отговори и резултата от всички проиграни въпроси. Това осигурява постоянна обратна връзка със студента. Накрая на теста, ако студентът не е получил отличен резултат му се предоставя възможност за повторно проиграване на въпросите със сгрешен отговор. Крайната оценка се формира като сума от оценките на всички проиграни въпроси от избрания тест. Накрая, на финалния екран се представят името на потребителя, крайната индивидуална оценка от проиграния тест, максималната възможна оценка, процентното съотношение на оценката и друга детайлна информация. След това студентът има възможност да се върне и да избере друг тест за проиграване или друг модул от системата. Във всички тестове, с изключение на изпитните тестове, има възможност тестът да се прекъсне във всеки един момент т.е. след всеки въпрос, не само накрая на теста. Съществува възможност за регулиране на времето за отговор индивидуално при всеки въпрос в избран тест.



Фигура 3.7 Модул „Тестове”

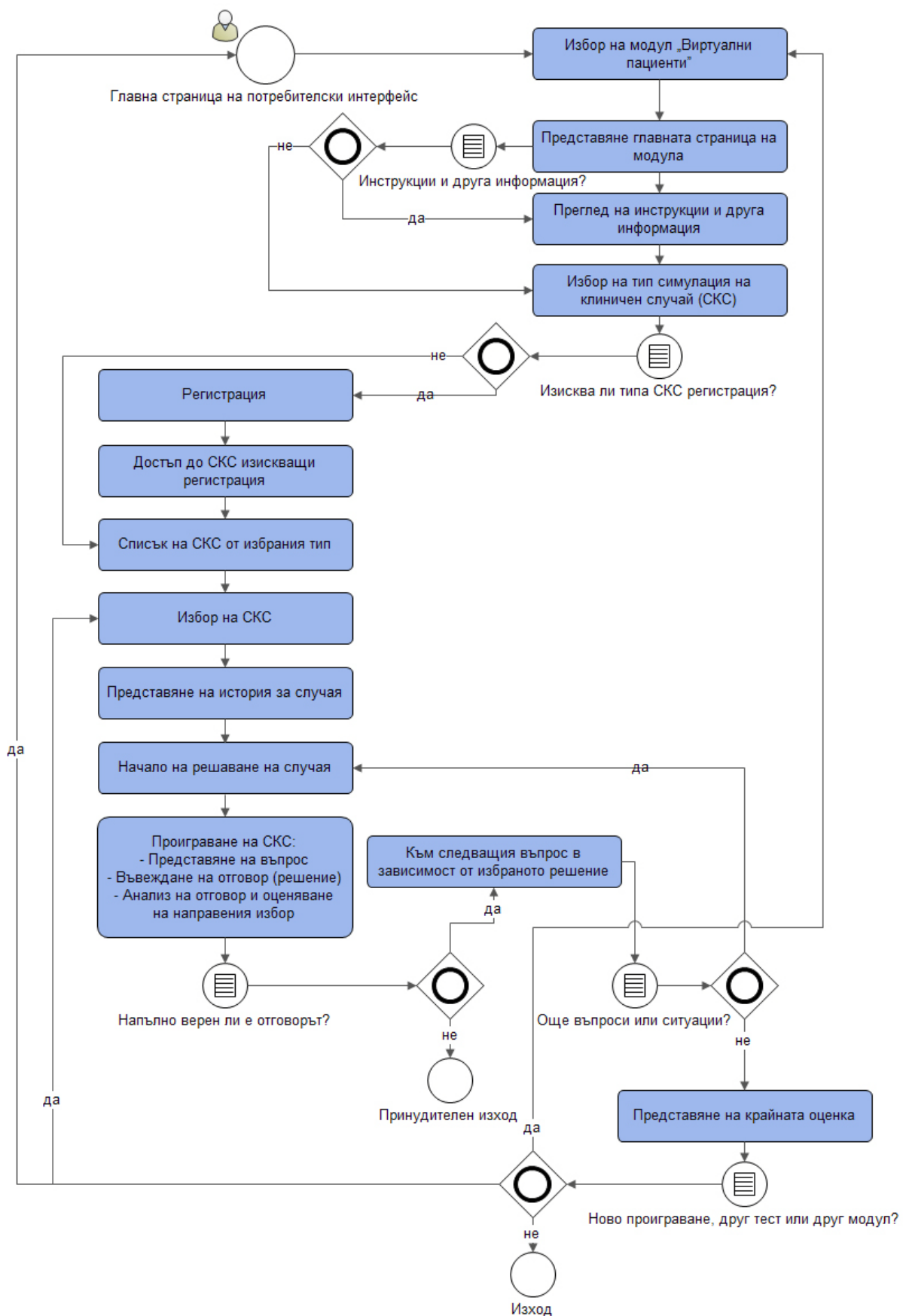
На **Фигура 3.8** е представен *сценарий на действието* на модул „Тестове”. Този сценарий описва стъпките на действие, които потребителят на системата трябва да следва, когато желае да оцени своите теоретични познания по желаната дисциплина, използвайки предложените тестове. Отделните стъпки се обединяват в един общ процес, който впоследствие е следван при написването на програмния код, отнасящ се за конкретния модул. За моделирането и графичното визуализиране на процеса е използвана BPMN диаграма за логическо проследяване на отделните му стъпки.



Фигура 3.8: Сценарий на действие на модул „Тестове”

Модул „Виртуални пациенти” - Този модул улеснява добиването и подобряването на професионалните умения на студентите за решаване на клинични случаи (**фигура 3.9**). Симулациите на клинични случаи са специализирани тестове, чиято цел е решаване на клиничен проблем. Въпросите следват съобразно логическия ред за решаване на случая. Поради това системата дава възможност връщане към текущия или предходен въпрос. Това позволява различни развръзки в зависимост от плана на преподавателя и уменията на студента. Решавайки клиничен случай, на студента се предоставя възможност да избере от предложени решения или да даде свое собствено решение на текущия въпрос. Всеки отговор е придружен едновременно с коментар и се оценява, но самата оценка се показва само на финалния екран, а не след всеки отговор, както е при теоретичните тестове. За всеки един случай има и специфична страница в помощ на студента, както и страница с история на заболяването на пациента. Тези страници са постоянно активни в модула. Прекратяването на симулацията на клиничен случай и избор на друг модул от системата е възможно по всяко време на проиграване на казус.

На **Фигура 3.10** е представен *сценарий на действието* на модул „Виртуални пациенти”. Този сценарий описва стъпките на действие, които потребителят на системата трябва да следва, когато желае да подобри своите професионални умения при решаване на клиничен казус по желаната дисциплина, използвайки предложените тестове. Отделните стъпки се обединяват в един общ процес, който впоследствие е следван при написването на програмния код, отнасящ се за конкретния модул. За моделирането и графичното визуализиране на процеса е използвана BPMN диаграма за логическо проследяване на отделните му стъпки.



Фигура 3.10: Сценарий на действие на модул „Виртуални пациенти”

Модули „Регистрация” и „Вход/Изход” - Чрез тези модули се контролира достъпа на потребителите в системата. Системата позволява четири нива на достъп: гости, студенти, преподаватели без административни права и преподаватели с административни права. Гостите в системата имат свободен достъп до съдържанието на курс, съдържание на раздели, до примерни лекционни точки, примерни тестове, демонстрационни симулации на клинични случаи и до всички допълнителни секции и подсекции, съдържащи обща за курса информация. Потребителите, които желаят да се възползват изцяло от възможностите на системата, могат да подадат заявка за регистрация, използвайки регистрационния модул (**фигура 3.11**). Този модул включва две регистрационни форми - за студенти и за преподаватели. Регистрацията на студентите и активирането им се одобрява от техния преподавател, а регистрацията на преподавателя се одобрява от преподавател с административни права. Регистрираните студенти получават достъп да използват всички учебни материали. Регистрираният преподавател освен че получава достъп до всички материали в потребителския интерфейс, но и получава достъп до административния интерфейс. В този случай, той вижда всичко създадено до момента от други преподаватели, но може да въвежда, изтрива и редактира само собствени данни. Преподавателят с административни права има пълен достъп до системата. Той може да активира, деактивира и изтрива потребители.

Фигура 3.11 Модули „Регистрация” и „Вход/Изход”

Влизането на регистрирани потребители в системата на потребителския интерфейс се осъществява чрез модул „Вход/Изход”. Този модул е постоянно достъпен в лявата част на екрана (**фигура 3.11**). За целта потребителят въвежда своето потребителско име и парола, след което се изписва неговото име и се появява връзка „Изход”. Изходът от системата е възможен във всеки един момент.

Модул „Забравена парола” - Този модул позволява максимално бързо потребителят да получи възможност за възстановяване на потребителските си права (**фигура 3.12**). Необходимо е да въведе e-mail адреса, който е използвал при регистрацията. След което ще получи e-mail с инструкции как да промени забравената си парола.

The screenshot shows the 'ЗАБРАВЕНА ПАРОЛА' (Forgot Password) module. It features a 'LOG ФОРМА' (Login Form) on the left with fields for 'Потребител:' (User) and 'Парола:' (Password), and buttons for 'Вход' (Login) and 'Регистрация | Забравена парола' (Registration | Forgot Password). The main area contains instructions: 'Въведете e-mail адреса, който сте използвали при регистрацията. Ще получите e-mail с инструкции как да промените забравената си парола.' (Enter the e-mail address you used during registration. You will receive an e-mail with instructions on how to change your forgotten password.) Below this are fields for 'Студент/Преподавател:' (Student/Teacher) with a dropdown menu, and 'Email:' with a text input field. A blue 'Изпрати' (Send) button is at the bottom.

Фигура 3.12 Модул „Забравена парола”

Модул „Търсене” - Този модул дава възможност за търсене по ключова дума или израз (**фигура 3.13**). Визуализирането на резултатите е подредено по начин, който позволява бърз, удобен и систематизиран достъп до търсената информация за „Лекции” и „Тестове”. Търсенето е възможно не само в текуща страница, но и глобално в цялата система.

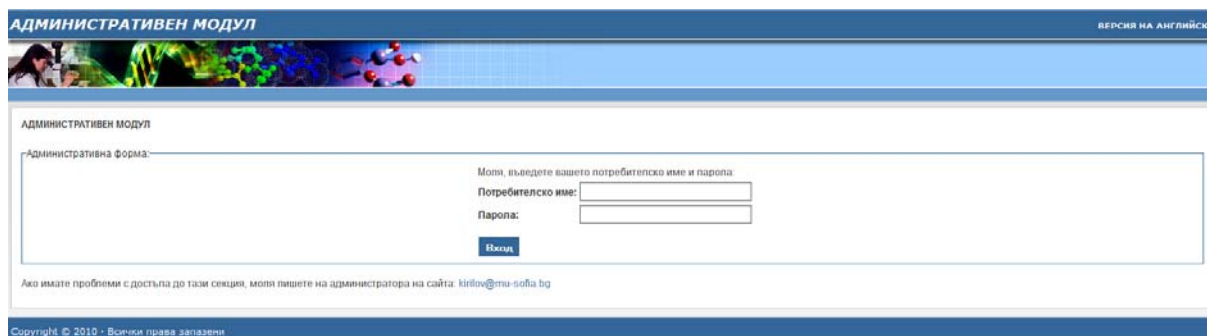
The screenshot shows the 'РЕЗУЛТАТИ ОТ ТЪРСЕНЕ' (Search Results) module. It features a 'LOG ФОРМА' (Login Form) on the left, identical to the one in Figure 3.12. The main area displays search results for the keyword 'ензими' (enzymes). It shows 'Резултати от търсене в тестове и лекции по зададена ключова дума: ензими' (Search results in tests and lectures for the given keyword: enzymes) and '1 намерен резултат в лекции' (1 found result in lectures). Below this, it lists 'Раздел: Примерен' (Section: Example) and 'Предмет, цели и обхват на биохимията' (Subject, objectives and scope of biochemistry).

Фигура 3.13 Модул „Търсене”

3.2.4.2 Описание на административен интерфейс на системата

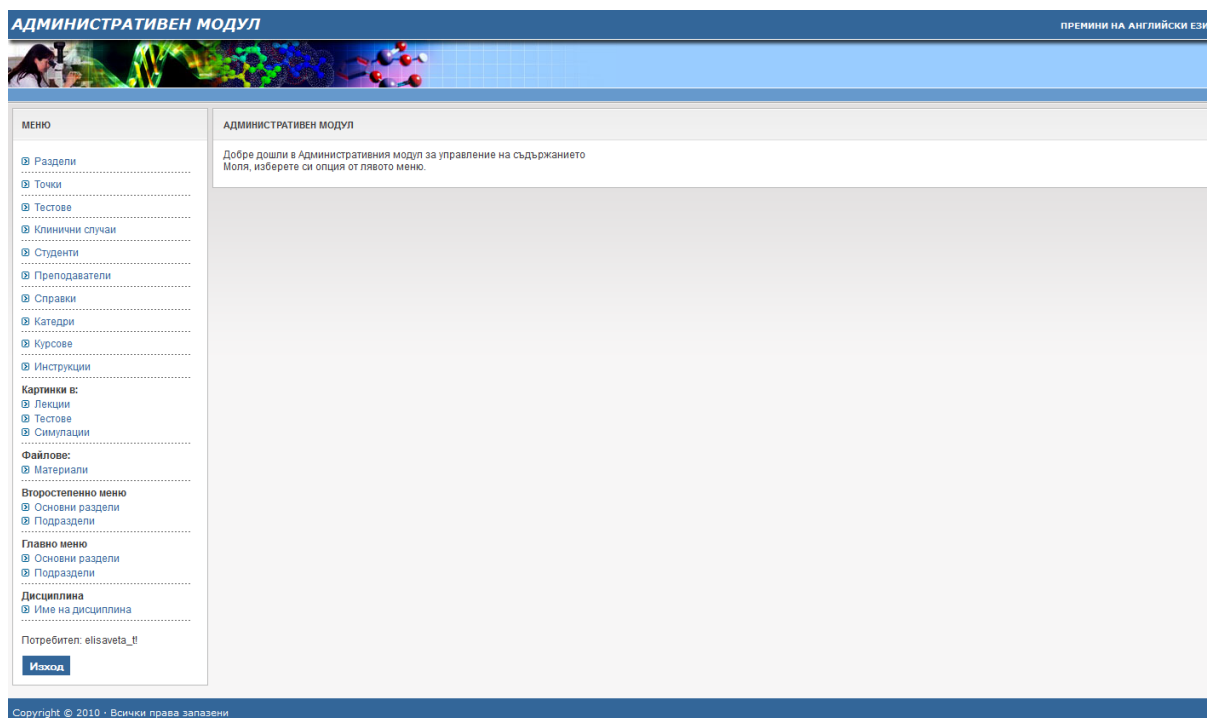
Използвайки административния интерфейс (**фигура 3.14**), преподаватели без умения в програмирането, могат лесно да публикуват лекции като източник на знания, интерактивни тестове за (само) оценка на теоретични знания и интерактивни виртуални

пациенти за решаване на клинични казуси. Само регистрирани преподаватели или администратори (преподаватели с административни права) могат да използват този модул. Неоторизираните потребители като гости, студенти или нерегистрирани преподаватели нямат достъп до този модул. Системата дава възможност за вмъкване на мултимедийни файлове, флаш файлове, различни изображения и файлове с документи, създаване на връзки в рамките на и между отделните секции на курсовете или външни източници, на тестови въпроси, управление на менюта и под-менюта. Изпълнението на много от тези операции е възможно благодарение на текстов редактор, който е интегриран в системата.



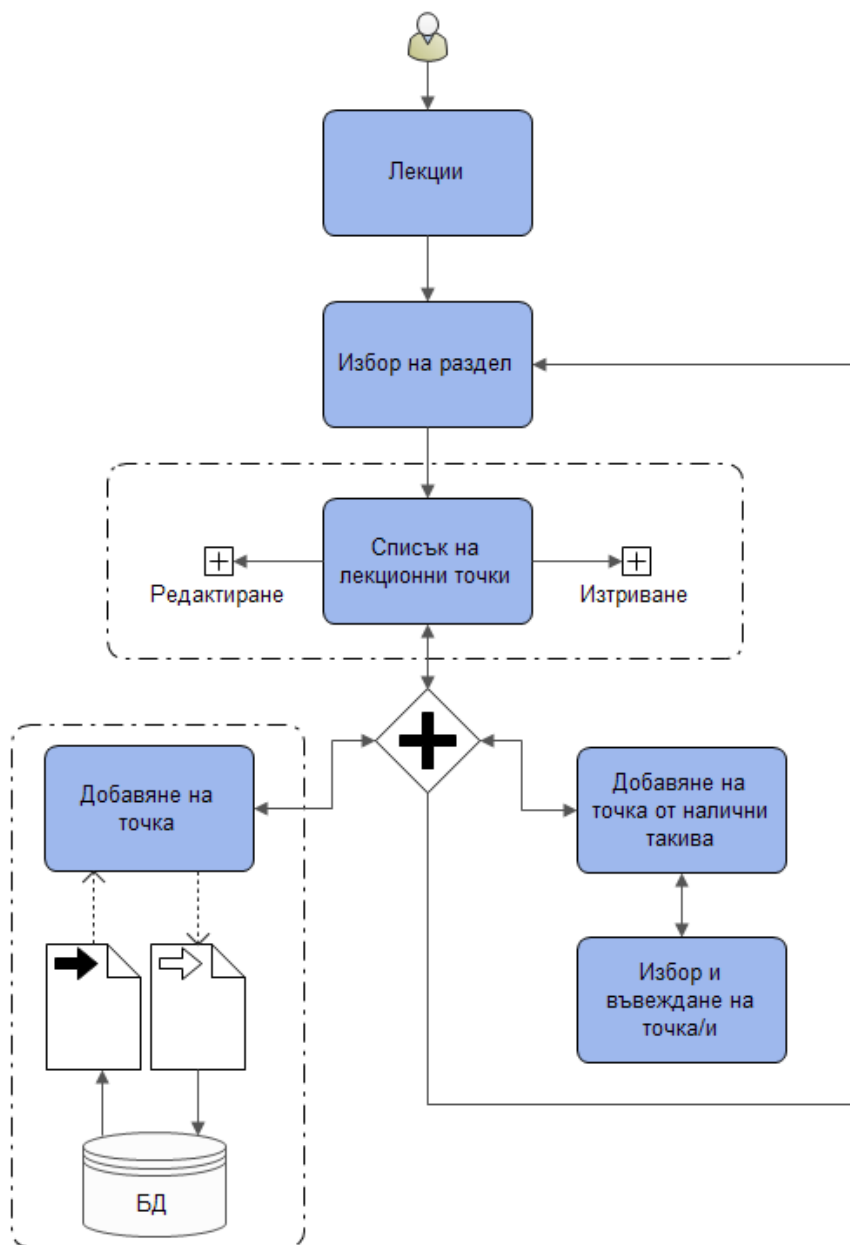
Фигура 3.14 Начална страница на административен интерфейс

Главното меню на административния интерфейс (**фигура 3.15**) се състои от следните постоянно налични връзки: раздели, теми, тестове, виртуални пациенти, студенти, преподаватели, справки, катедри, курсове, инструкции, картинки (в темите, в тестовете, в симулациите), материали, създаването на основни и второстепенни менюта (раздели и подраздели), възможност за промяна името на дисциплината, изход.



Фигура 3.15 Екран от административния интерфейс

На **фигура 3.16** е представен *сценарий на процеса на работа при създаване на лекционна тема в модул „Лекции“*. Показани са стъпките при създаване на лекционни теми в тоя модул. Отделните стъпки се обединяват в един общ процес, който впоследствие е следван при написването на програмния код, отнасящ се за конкретния модул. За моделирането и графичното визуализиране на процеса е използвана BPMN диаграма за логическо проследяване на отделните му стъпки.



Фигура 3.16: Схема на сценарий на процес за работа в модул „Лекции”

Избира се разделът, в който ще се съдържат темите на лекциите. Всеки раздел включва полета с данни, необходими за действие на модула. След избора на раздел има две възможности да се продължи – да се избере опция "Нова тема" или опция "Добави тема от налични". Първата опция позволява да се добави нова тема, попълвайки дадената форма и следвайки необходимите изисквания. Има четири нива на йерархия (раздели,

лекции, теми и подтеми) за процеса на манипулиране на данни. Всяка тема принадлежи само към един раздел и се попълват изискваните полета с данни. Втората опция позволява да се добави тема(и) от налична, ако има такива. Всяка една от вече създадените теми може да се редактира или изтрива. Няма ограничения в броя на темите в раздел.

Фигура 3.17 показва как преподавателя може да създаде лекционна тема в модул „Лекции” следвайки стъпките от **фигура 3.16**.

АДМИНИСТРАТИВЕН МОДУЛ

МЕНЮ

- Раздели
- Точки
- Тестове
- Клинични случаи
- Студенти
- Преподаватели
- Справки
- Календари
- Курсове
- Инструкции
- Картички
- Лекции
- Тестове
- Симулации
- Файлове
- Материали
- Второстепенно меню
- Основни раздели
- Подраздели
- Главно меню
- Основни раздели
- Подраздели
- Дисциплина
- Име на дисциплина
- Потребител: elizaveta_b
- Изход

ТОЧКИ

Изберете раздела, с който ще работите | Версия на английски

Номер ▼ Заглавие

Номер	Заглавие
1	Примерен

ТОЧКИ

Точки = Примерен

Добавете точка | Добавете точка от наличните

Ред ▼	Номер	Код	Заглавие	Редактирай	Изтрий
▼ ▲	1.		Резюме	Редактирай	Изтрий
▼ ▲	2.		Предмет, цели и обхват на биохимията	Редактирай	Изтрий
▼ ▲	3.		Роля на биохимията за развитие на медицината	Редактирай	Изтрий

ТОЧКИ

Точка = Примерен = Редактирайте точка

Редактирайте точка

* Задължителни полета

Раздел: Примерен

Заглавие: *

Код:

Номер: 1

Коментар:

Тип: *

Точка със съдържание

Съдържание:

Биохимията изучава молекулните основи на живота в разнообразните му проявления от вируси до човек. Биохимията и медицината са неразривно свързани. Здравото зависи от хармоничното равновесие на протичащите в тялото биохимични реакции, а болестта отразява отклонения или нарушения в биомолекулите, биохимичните реакции или процеси. μ μ

Прогресът в биохимичната теория осветлява много области в медицината, и обратно, изучаването на

Връзка към дадената точка: lecture.php?l_id=107&sec_id=26

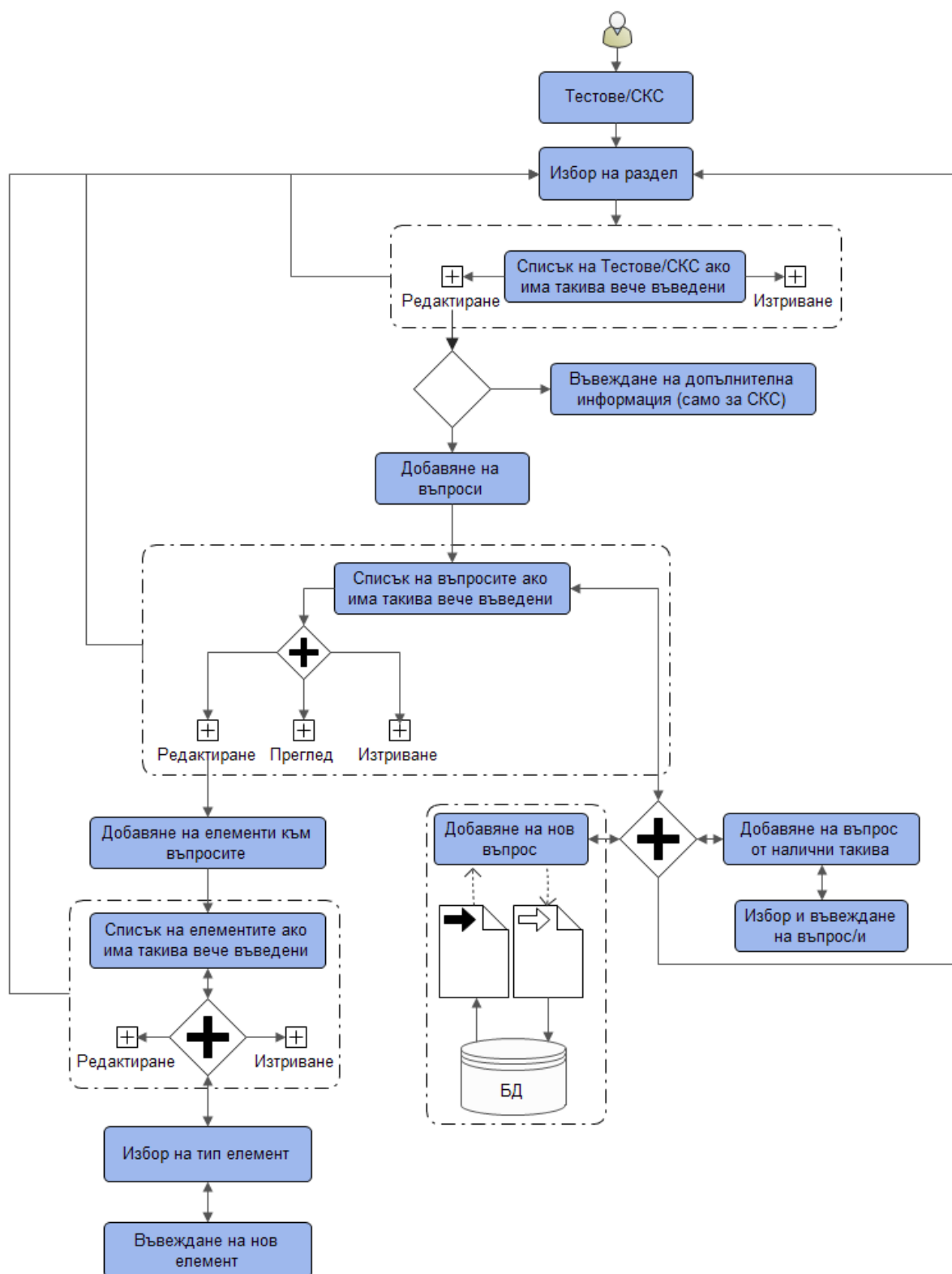
Точка, изискваща регистрация: ☒

Защитена точка: ☒

Запази Откаж

Фигура 3.17 Създаване на лекционна тема чрез административния модул

На **фигура 3.18** е представен *сценарий на процеса на работа при създаване на тестове в модул „Тестове” и „Виртуални пациенти”*. Показани са стъпките при създаване на тестове или клинични казуси в модули „Тестове” и „Виртуални пациенти” (СКС). Отделните стъпки се обединяват в един общ процес, който впоследствие е следван при написването на програмния код, отнасящ се за конкретния модул. За моделирането и графичното визуализиране на процеса е използвана BPMN диаграма, за логическо проследяване на отделните му стъпки.



Фигура 3.18: Схема на сценарий на процес за работа в модули „Тестове” и „Виртуални пациенти”

След като е избран разделът, в който ще се създава или редактира тест или СКС, системата предоставя списък от заглавия на наличните тестове или СКС заедно с техния статус, дата на промяна и др. На това ниво тестът може да бъде изтриван,

преномериран или редактиран. Статусът на теста може да бъде примерен, реален, изпитен, в процес на разработка, архивиран. Статусът на СКС може да бъде примерна или реална.

При работа с въпроси в теста наред със стандартните възможности особено полезни са опциите за добавяне на нов въпрос и за добавяне на въпрос от наличните както в тестовите, така и в СКС. Това позволява по-лесно да се подготвят комбинирани тестове или симулации. Възможността за индивидуално време за решаване на въпроса в теста допълнително улеснява диференцирането на оценката.

Добавянето на нов въпрос може да става на произволно място в списъка от въпроси. Няма ограничения за броя на тестовите и СКС, за броя на въпросите в теста и в СКС и за броя на компонентите (отговорите и елементите) във въпроса, нито за стойностите на теглата на отговорите. Добавянето на нов компонент може да става на произволно място в списъка с компоненти.

Системата позволява коментари към въпроса като цяло в тестовите или коментар към всеки отговор в СКС.

Използването на административния интерфейс позволява:

- Създаване, редактиране и изтриване на раздели;
- Създаване, редактиране и изтриване на лекционни теми;
- Създаване, редактиране и изтриване на тестове;
- Създаване, редактиране и изтриване на симулации на клинични случаи;
- Създаване, активиране, деактивиране и изтриване на потребители (студенти и преподаватели) на системата;
- Справка, по отношение на представяне на студентите или качество на въпросите. Справката е възможна само за тестове от тип изпитни;
- Добавяне, редактиране и изтриване на катедри и курсове;
- Добавяне на инструкции за работа със системата;
- Изтриване на картинки от лекции, тестове и виртуални пациенти;
- Изтриване на файлове;
- Добавяне, редактиране и изтриване на секции и подсекции от главно и второстепенно меню в системата;
- Редактиране и изтриване на името на дисциплината, за която се създава конкретния курс за Web-базирано обучение.

3.3 Сравнение на разработената система с други български и чужди системи в областта на електронното обучение

Представено е сравнение на възможностите между някои от разгледаните в глава първа системи за електронно обучение и разработената система за Web-базирано обучение в дисертационния труд (таблица 3.2). Целта на направеното сравнение е не да покаже, че системата е по-добра от посочените универсални системи за електронно обучение, а да посочи, че тя е съизмерима и се доближава до голяма степен по

основните си качества на едни от най-добрите решения в областта на електронното обучение. Сравнението е направено по критерии, базирани на независими експертни изследвания [57, 121]. Критериите са следните:

- Системи за комуникация:
 - o Асинхронни
 - Форум
 - Табло за съобщения
 - Вътрешен адрес за електронна поща
 - Бележки върху учебното съдържание
 - Персонално дисково пространство
 - o Синхронни
 - Чат
 - Бяла дъска
 - Видео конференция
- Инструменти, подпомагащи работата на обучаемите
 - o Контекстно ориентирана помощ
 - o Ръководство за работа с платформата
 - o Работа в група
 - o Самооценка
 - o Търсене в съдържанието на курсовете
 - o Създаване на собствена WEB-страница
- Авторска система и система за доставяне на учебното съдържание
 - o Поддръжка на шаблони
 - o WEB-базирана разработка на ресурси
 - o WEB-базиран текстов редактор
 - o Многократно използване на ресурси
 - o Разработка на тестове
- Инструменти, подпомагащи работата на администратора
 - o Поддръжка на потребителски роли
 - o Управление на потребителските права
 - o Пренос на потребителска информация
 - o Външна автентикация
- Поддържани стандарти и спецификации за електронно обучение
 - o AICC
 - o SCORM
 - o IMS спецификация на съдържанието
 - o IMS спецификация на мета- данните
 - o IMS спецификация на тестовете
- Хардуерни и софтуерни изисквания
 - o Изисквания за технологии от страна на сървъра
 - Intel архитектура
 - Друга архитектура
 - Windows

- Unix
- MySQL
- Microsoft SQL Server
- Oracle
- Apache WEB Server
- IIS WEB Server
- Изисквания за технологии от страна на клиента
 - Internet Explorer
 - Други WEB браузъри
 - Java RTE
 - Допълнително програмно осигуряване
- Език на интерфейса
 - Поддръжка на Български
 - Поддръжка на повече от един език
 - Възможност за добавяне на език
- Ценова политика
 - Базирана на лицензи
 - Закупуване
 - Абонамент за използване
 - Безплатна
 - Създадена за специфични изисквания

Критериите са оценени по следната скала: Да = 1; Не = 0. По данните от таблицата са съставени радарни диаграми.

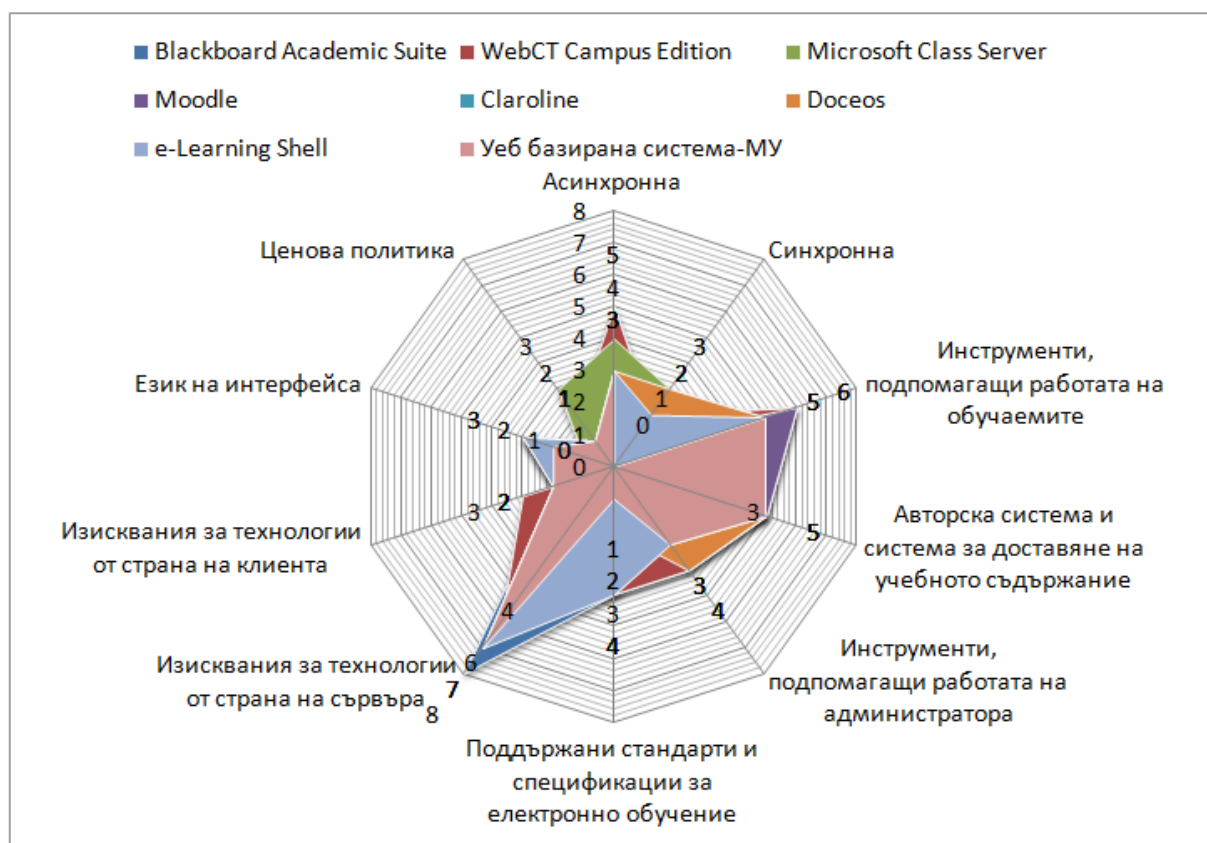
Таблица 3.2 Сравнение на системи за електронно обучение и разработената система

	Blackboard Academic Suite	WebCT Campus Edition	Microsoft Class Server	Moodle	Claroline	Doccos	e-Learning Shell	Web-базирана система на МУ
Системи за комуникация								
Асинхронна								
Форум	1	1	0	1	1	0	1	1
Табло за съобщения	1	1	1	1	1	1	1	1
Вътрешен адрес за електронна поща	1	1	1	0	0	1	0	1
Бележки върху учебното съдържание	1	1	1	0	0	0	0	0
Персонално дисково пространство	1	1	1	1	1	1	1	0
Средна оценка - 3,71	5	5	4	3	3	3	3	3
Синхронна								
Чат	1	1	1	1	1	1	1	0
Бяла дъска	1	1	1	0	0	1	1	0
Видео конференция	0	0	1	0	1	1	0	0
Средна оценка - 2,14	2	2	3	1	2	3	2	0

Инструменти, подпомагащи работата на обучаемите								
Контекстно ориентирана помощ	1	1	1	1	1	1	1	1
Ръководство за работа с платформата	1	1	1	1	1	1	1	1
Работа в група	1	1	1	1	1	1	1	1
Самооценка	1	1	1	1	1	1	1	1
Търсене в съдържанието на курсовете	1	1	1	1	1	1	1	1
Създаване на собствена WEB-страница	1	1	0	1	0	0	0	0
Средна оценка - 5,43	6	6	5	6	5	5	5	5
Авторска система и система за доставяне на учебното съдържание								
Поддръжка на шаблони	1	1	1	1	1	1	1	1
WEB-базирана разработка на ресурси	1	1	0	1	1	1	1	1
WEB-базиран текстов редактор	1	1	0	1	1	1	1	1
Многократно използване на ресурси	1	1	1	1	1	1	1	1
Разработка на тестове	1	1	1	1	1	1	1	1
Средна оценка - 4,71	5	5	3	5	5	5	5	5
Инструменти, подпомагащи работата на администратора								
Поддръжка на потребителски роли	1	1	1	1	1	1	1	1
Управление на потребителските права	1	1	1	1	1	1	1	1
Пренос на потребителска информация	1	1	0	0	0	1	0	0
Външна автентикация	1	1	1	1	1	1	1	1
Средна оценка - 3,28	4	4	3	3	3	3	3	3
Поддържани стандарти и спецификации за електронно обучение								
AICC	0	0	0	1	0	0	0	0
SCORM	1	1	1	1	1	1	1	1
IMS спецификация на съдържанието	1	1	1	0	0	0	1	0
IMS спецификация на мета- данните	1	1	1	0	0	0	1	0
IMS спецификация на тестовете	1	1	1	1	1	1	1	0
Средна оценка - 3,28	4	4	4	3	2	2	4	1
Хардуерни и софтуерни изисквания								
Изисквания за технологии от страна на сървъра								
Intel архитектура	1	1	1	1	1	1	1	1
Друга архитектура	1	1	0	1	1	1	1	1
Windows	1	1	1	1	1	1	1	1
Unix	1	1	0	1	1	1	1	1
MySQL	0	0	0	1	1	1	1	1
Microsoft SQL Server	1	1	1	0	0	0	0	0
Oracle	1	1	0	0	0	0	0	0
Apache WEB Server	1	0	0	1	1	1	1	1
IIS WEB Server	1	0	1	1	1	1	1	1
Средна оценка - 6,57	8	6	4	7	7	7	7	7
Изисквания за технологии от страна на клиента								
Internet Explorer	1	1	1	1	1	1	1	1
Други WEB браузъри	1	1	0	1	1	1	1	1
Java RTE	0	1	0	0	0	0	0	0
Допълнително програмно осигуряване	0	0	1	0	0	0	0	0
Средна оценка - 2,14	2	3	2	2	2	2	2	2
Език на интерфейса								
Поддръжка на Български	0	0	1	1	1	1	1	1
Поддръжка на повече от един език	0	0	0	0	1	1	1	1
Възможност за добавяне на език	0	0	0	1	1	1	1	0
Средна оценка - 1,71	0	0	1	2	3	3	3	2

Ценова политика								
Базирана на лицензи	1	1	1	0	0	0	0	0
Закупуване	0	0	1	0	0	0	1	0
Абонамент за използване	1	1	1	0	0	0	0	0
Безплатна	0	0	0	1	1	1	0	0
Създадена за специфични изисквания	0	0	0	0	0	0	0	1
Средна оценка - 1,57	2	2	3	1	1	1	1	1

От представените резултати на базата на направеното сравнение в **таблица 3.2** се вижда, че разработената система не отстъпва по основните си качества на най-добрите решения в тази област и притежава необходимите средства за провеждане на ефективно електронно обучение. Резултатите са визуализирани чрез радарна диаграма на **фигури 3.19 и 3.20**. Критериите, върху които трябва да се наблегне при бъдещо развитие на системата са синхронните системи за комуникация и поддържаните стандарти и спецификации за електронно обучение.

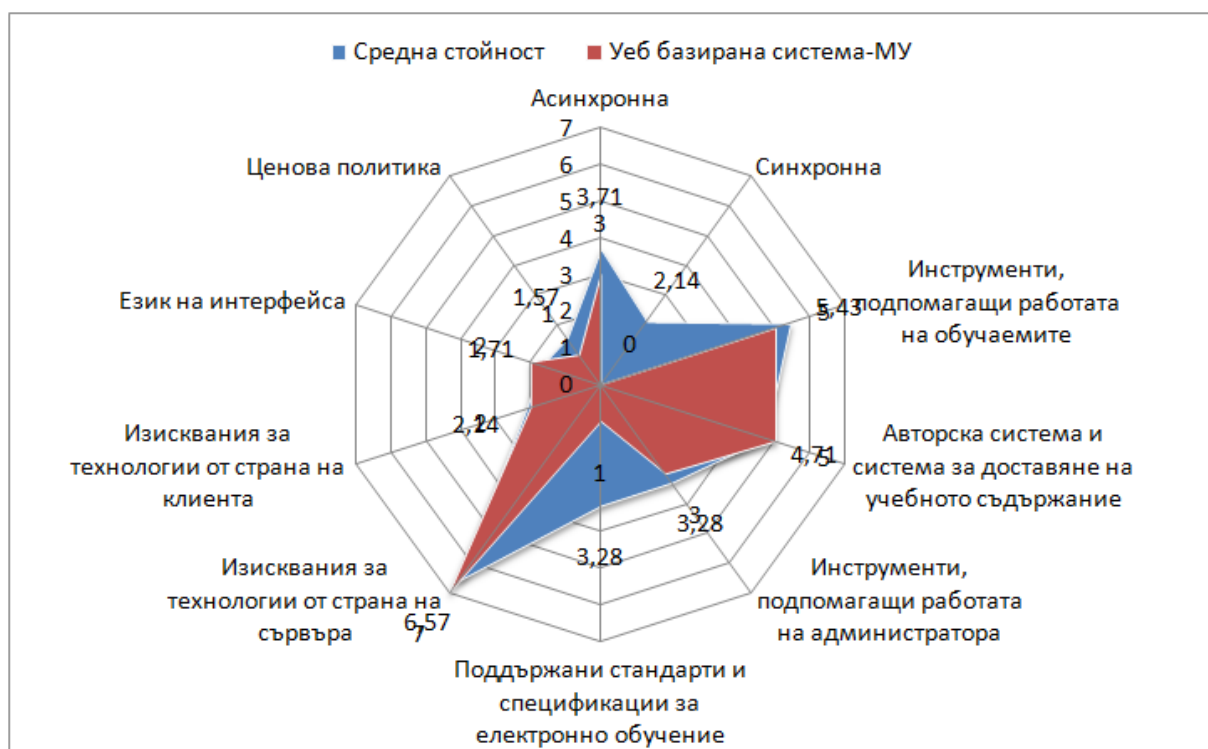


Фигура 3.19 Сравнение на възможностите на системата с други системи за електронно обучение

На радарната диаграма от **фигура 3.19** е илюстрирано сравнението на разработената система за Web-базирано обучение с всички останали системи посочени в **таблица 3.2**. Площта, която заема разработената системата в сравнение с останалите системи показва, че тя удовлетворява в голяма степен посочените критерии.

За радарната диаграма от **фигура 3.20** бе направено същото сравнение както при резултатите от **фигура 3.19**, с тази разлика, че бяха изчислени средните стойности за

всеки от критериите от **таблица 3.2**. От диаграмата се вижда, че разработената система изпълнява до голяма степен критериите, характерни за универсална Web - базирана система за управление на обучението и има голям потенциал за бъдещо развитие.



Фигура 3.20 Сравнение на възможностите на системата с други системи за електронно обучение

3.4 Изводи

В трета глава е представена последователността от дейности, изпълнени при разработването на система за Web-базирано обучение по медицински дисциплини.

Дефинирани са изискванията към системата, като на базата на тези изисквания се осъществява нейното проектиране и реализиране. Представен е релационният модел на базата данни. Следвайки този модел е направена физическата реализация на базата от данни. Принципът на действие на системата е базиран на архитектурния модел „клиент-сървър” с три взаимосвързани йерархични нива. Информационните процеси, на които е базирано разработването на програмните модули, са описани чрез сценарии на действието и на работата им.

Системата се състои от два интерфейса: административен и потребителски (студентски). Всеки от тях е удобен, лесен за използване и интерактивен за всички потребители - гости, студенти, преподаватели, администратори. Системата е гъвкава и отворена - независима от операционна система, съвместима с широко използваните браузъри, съвместима с други Web приложения и др. Също така е независима от конкретното съдържание на дисциплината. Системата осигурява бърз и лесен достъп до всяка част от нея.

Съобразена е с основни принципи на препоръчителни международни стандарти обобщени като многократна използваемост, управляемост на информацията за потребителите и учебното съдържание, достъпност, дълготрайност – възможност за развитие на системата с развитие на технологиите и възможност за взаимодействие между различни системи.

Притежава възможност за планиране и организиране на обучението чрез публикуване на всички необходими за студентите документи като инструкции, анотация, конспект, програма, разписание, информация за преподавателите, секция за съобщения и възможност за комуникация между преподавателя и студентите.

По отношение на студентите и преподавателите системата има модули за регистрация, отдалечен контролиран достъп, модул за ползване и създаване на лекции, за проиграване и създаване на тестове. Има и модул, позволяващ преглеждане и създаване на всички документи, необходими за учебния процес, както и възможност за комуникации между студентите и преподавателя. Дава възможност за търсене по ключова дума или израз.

Системата изпълнява и важни функции, **специфични** за Медицински Университет - София, като например възможността за публикуване, редактиране и ползване на виртуални пациенти и виртуални модели.

При направен сравнителен анализ на възможностите на разглежданата в дисертационния труд система за Web-базирано обучение с тези на известните софтуерни платформи (**таблица 3.2**) се вижда, че системата е сравнима по основните си качества с най-добрите решения в тази област и притежава необходимите средства за ефективно провеждане на електронно обучение. Удовлетворява посочените критерии и има голям потенциал за бъдещо развитие.

Резултати от изследванията по тази глава са публикувани в [2, 3, 4, 5] от публикациите по дисертацията.

Глава четвърта: Оптимизиране на информационни процеси

4.1 Увод

В глава четвърта от дисертационния труд се представят резултатите от направеното експериментално приложение на подхода за количествена оценка на качеството и ефективността на Web-базирана система за обучение по медицински дисциплини. Описва се начинът, по който е проведено анкетирането на участниците в оценяването. Посочват се целите за провеждането му. Анализират се и се описват подробно резултатите от направените изследвания, като се въвеждат в таблици и се визуализират чрез диаграми. **Въз основа на направената оценка на системата и решаването на оптимизационна задача, чрез която се оценява потенциала на експертните мнения, се определят по кои показатели ще се извърши оптимизация на процесите, свързани с работата на отделните модули с цел подобряване и увеличаване на възможностите на системата, но без да се усложнява работата с нея.**

4.2 Подход за оценка качеството и ефективността на системата за Web-базирано обучение

В основата на разглеждания подход е един от методите за определяне на тегловни коефициенти по мнения на експерти. Методът на експертните оценки [109, 110], се основава на анкетиране на специалисти в дадената проблемна област и обработване на получените резултати. В настоящето изследване се предлага подход, базиран на метода на експертните оценки посредством изпълнение на следната последователност от стъпки:

- Етапи на реализация на подхода;
- Определяне на система от показателите за оценка на качеството и ефективността на системата;
- Определяне на тегловни коефициенти на показателите;
- Изчисляване на тегловните коефициенти на показателите;
- Дефиниране и решаване на задачата за оптимизация на процесите;
- Графично представяне на резултатите.

4.2.1 Етапи на реализация на подхода

Оценката на качеството и ефективността на една система има съществено значение както при цялостния ѝ анализ, така и за нейното подобряване. Качеството може да се изрази посредством определена целева функция, в която независимо от нейния аналитичен вид участва набор от показатели, чиято важност може да се отрази чрез подходящи, обективно получени тегловни коефициенти [98].

Първи етап: Оценката се извършва от експерти, специалисти в областта на софтуерните технологии. Определя се списък от показатели за количествена оценка на качеството на системата на базата на стандарта ISO 9126 за качество на софтуерни продукти. След това се изчислява оценката на всеки анкетиран експерт и се определят тегловните коефициенти. Дефинира се и се решава оптимизационна задача за

оценяване на потенциала на експертните мнения и на база получените резултати се вземат решения за извършване на технологично и структурно подобряване на системата.

Втори етап: Оценката се извършва от преподаватели от институцията, която ползва системата. Определя се списък от дидактически показатели, изчислява се оценката на анкетирания и се определят тегловните коефициенти. На базата на този списък от показатели се прави оценка на ефективността на системата. Под ефективност се визира възможността системата да изпълнява поставените ѝ цели и изисквания. Дали е достатъчно лесна за изучаване, усвояване и експлоатация. Да се установи дали тя е достатъчно универсална, че да е възможно използването ѝ от други дисциплини и курсове в МУ-София и други университети в тази област. Резултатите се обработват, правят се заключения относно получените резултати и се вземат решения за подобряване на състоянието на системата.

4.2.2 Определяне на система от показателите за оценка на качеството и ефективността на системата

Съставянето на количествена оценка за качеството на една система е важна не само за оценката и за сравнението на отделните ѝ програмни модули, но и за ефективното управление и усъвършенстването на процесите, съставляващи тези модули. Правилното разработване и прилагане на моделите на информационните процеси е от изключително значение за коректното функциониране и оценяване на самата система.

Определени са две групи от показатели (**таблица 4.1**):

- Показатели за оценка на качеството на системата на базата на стандарт ISO 9126 за оценка на качеството на софтуерни продукти;
- Дидактически показатели за оценка на ефективността на системата, базирани на мнения и препоръки на преподаватели от институцията, използваща информационната система.

Таблица 4.1 Показатели

<i>Софтуерни показатели по стандарт ISO 9126</i>	<i>Дидактически показатели</i>
- Функционалност (удачност, точност, взаимоработоспособност, сигурност, съгласуваемост) - Безотказност (отработеност, отказоустойчивост, възстановяемост, съгласуваемост) - Използваемост (разбираемост, изучаемост, манипулативност, привлекателност, съгласуваемост) - Продуктивност (продуктивност по време, продуктивност по ресурси, съгласуваемост) - Обслужваемост (анализируемост, изменяемост, стабилност, изпитваемост, съгласуваемост) - Преносимост (приспособимост, инсталируемост, съвместно съществуване, заменимост, съгласуваемост)	- Степен на логическа последователност на учебния материал; - Свързаност на компонентите на учебния материал; - Насърчаване за критично мислене и креативност; - Връзка с други ресурси за допълнителна информация за изучавания проблем; - Използване на различни мултимедийни компоненти; - Наличие на речник на използваните термини в лекционния материал; - Наличие на търсеца машина в основните модули.

Всяка от характеристиките (критерии, атрибути) на качеството се измерва и остойността с различни средства и има свой принос в образуването на крайния резултат на оценката.

Предложената система от показатели за оценка е отворена. Може да се прибавят нови показатели, както и да се изключат някои съществуващи, в зависимост от конкретните цели.

4.2.3 Определяне на тегловни коефициенти на показателите

Определянето на тегловните коефициенти на показателите за качество, както и изчисляването на коефициента на съгласие между анкетираните осигурява достатъчна степен на обективност на целия процес на оценяване на информационната система.

Етапите, през които преминава оценяването на системата са следните:

- Изучаване на процеса и избор на показателите за качество на системата;
- Определяне на степента на компетентност на анкетираните;
- Съставяне на две анкетни карти, които включват:
 - Списък с показатели (критериите) за оценка на качеството и за оценка на ефективността на системата, като всеки анкетиран може да отбележи и даде своята оценка по предварително зададена скала;
 - Информация за компетентността и източника на аргументация на отделните експерти, участващи в анкетирането.
- Определяне на кръга от специалисти и провеждане на анкетирането;
- Изчисляване на коефициента на съгласие между анкетираните и проверка на значимостта му;
- Изчисляване на тегловните коефициенти на показателите;
- Анализ на резултатите и графичното им представяне чрез диаграми.

За всеки един показател от анкетната карта се изчисляват съответните точки, образуващи оценката на анкетирания експерт, на базата на следното процентно съотношение $\frac{100\%}{x}$, където x е съответния брой на атрибутите на отделните показатели.

Вземайки предвид предварително зададената скала (Скала: Слаб = 2; Добър = 4; Силен = 6; Не мога да преценя = 0), резултатите от направените изчисления се въвеждат в **таблица 4.2** и се визуализират чрез Pie-диаграма за всеки един показател.

Таблица 4.2 Резултати от оценката на анкетиран експерт

Показатели	Скала	<i>Слаб</i>	<i>Добър</i>	<i>Силен</i>	<i>Не мога да преценя</i>	<i>точки</i>
y_1						
y_2						
y_3						
....						
....						
y_m						

Степента на компетентност на всеки един от анкетираните се определя чрез израза:

$$C = \frac{a_1 + a_2 + a_3}{R}, \quad (27)$$

където $0 < a_1 \leq 12$; $0 < a_2 \leq 12$; $0 < a_3 \leq 12$ и R е броят на експертите.

Съгласно **таблицы 4.3, 4.4, 4.5, 4.6, 4.7, 4.8** коефициентите a_1 , a_2 , a_3 се определят съответно от заеманата длъжност на всеки експерт, опит в годините от работа в определената област и източника на неговата аргументация.

Определяне на степен на компетентност на експерти в областта на софтуерните технологии:

Таблица 4.3 (a_1) Стойности на a_1

Заемана длъжност	Младши експерт	Старши експерт
Разработчик на софтуер (Software developer)	6	12
Front-end Web разработчик (Front-end web developer)	4	10
Web разработчик на приложения (Web application developer)	2	8

Таблица 4.4 (a_2) Стойности на a_2

Опит в годините от работа в областта на софтуерните технологии	0	1	2	3	4	5	>5
	0	2	4	6	8	10	12

Таблица 4.5 (a_3) Стойности на a_3

Източник на аргументация (Доколко сте запознат със системата?)	
Съвсем повърхностно	2
Повърхностно	6
В детайли	12

Определяне на степен на компетентност на преподаватели от институцията, използваща информационната система:

Таблица 4.6 (a_1) Стойности на a_1

Заемана длъжност	Без степен	Доктор	Доктор на науките
Професор	-	8	12
Доцент	-	6	10
Асистент	2	4	-

Таблица 4.7 (a_2) Стойности на a_2

Опит в годините от работа в областта на информационните технологии и електронното обучение	0	1	2	3	4	5	>5
	0	2	4	6	8	10	12

Таблица 4.8 (a_3) Стойности на a_3

Източник на аргументация (Доколко сте запознат със системата?)	
Съвсем повърхностно	2
Повърхностно	6
В детайли	12

4.2.4 Изчисляване на тегловните коефициенти на показателите

На R на брой експерти се предлага да дадат своето мнение и оценка на качеството и ефективността на информационна система, като се вземат под внимание m показатели (целеви параметъра) чрез анкетен лист. Получените резултати се записват в тегловна матрица (табл. 4.9). Всяко число в тегловната матрица a_{ij} определя оценката (теглото), която експертът i приписва на показателя (целевия параметър) j .

Таблица 4.9 Тегловна матрица

Експерти	E_{x_1}	E_{x_2}	E_{x_3}		E_{x_i}		E_{x_R}	S_j	S_{jav}	δ_j	V_j	W_j
Показатели												
y_1	a_{11}	a_{21}	a_{31}		a_{i1}		a_{R1}	S_{i1}	S_{iav1}	δ_1	V_1	W_1
y_2	a_{12}	a_{22}	a_{32}		a_{i2}		a_{R2}	S_{i2}	S_{iav2}	δ_2	V_2	W_2
....												
y_j	a_{1j}	a_{2j}	a_{3j}		a_{ij}		a_{Rj}	S_{ij}	S_{iavj}	δ_j	V_j	W_j
....												
y_m	y_{1m}	y_{2m}	y_{3m}		y_{im}		y_{Rm}	S_{im}	S_{iavm}	δ_m	V_m	W_m

- Изчисляване на сумата от оценките на всички експерти S_j за всеки един показател по формулата:

$$S_j = \sum_{j=1}^R a_{ij} \quad (28)$$

Където a_{ij} е оценката на i -ия експерт за j -ия показател, R -броят на експертите.

- Изчисляване на средната стойност на групата от експерти S_{jav} за всеки показател по формулата:

$$S_{jav} = \frac{1}{R} \sum_{j=1}^R a_{ij} \quad (29)$$

- Изчисляване на средната сума от всички точки (оценки) S_{av} [109]

$$S_{av} = \frac{R(m+1)}{2} \quad (30)$$

- За определяне степента на съгласуване на мненията на експертите се определя отклонението δ_j по формулата:

$$\delta_j = S_j - S_{av} \quad (31)$$

- Изчислява се коефициентът на съгласие w_k , по формула, предложена от Морис Кендал [109]:

$$w_k = \frac{12 \sum_{j=1}^m \delta_j^2}{R^2(m^3 - m)} \quad (32)$$

- Изчислява се коефициентът на вариация V_j , характеризиращ съгласуваността на оценките на експертите, участващи в анкетата, за отделните показатели:

$$V_j = \frac{(Rm - S_j)}{Rm - R} \quad (33)$$

Колкото е по-голяма стойността на V_j , толкова по-ниска е степента на съгласуваност на мненията на експертите и обратно.

- Проверка на значимостта на коефициента на съгласие:

Коефициентът на съгласие w_k може да се изменя от 0 при пълна несъгласуваност в мненията до +1 при пълна съгласуваност [109]. Оценка на значимостта на изчисления коефициент w_k може да стане по χ^2 -критерий при $m > 7$ [22] или по Z - критерия на Фишер при $m < 7$ [22].

При $m < 7$ се изчислява величината:

$$Z = \frac{1}{2} \ln \frac{(R-1)w_k}{1-w_k} \quad (34)$$

Ако е изпълнено условието $Z > Z_{\text{табл}}(\alpha, \nu_1, \nu_2)$, коефициента на съгласие w_k е значим. $Z_{\text{табл}}(\alpha, \nu_1, \nu_2)$ се взема от таблица (Приложение 1) при степени на свобода

$$\nu_1 = m - 1 - \frac{2}{R} \quad (35)$$

$$\nu_2 = (R-1)\nu_1 \quad (36)$$

α е ниво на значимост, което отговаря на доверителна вероятност $\beta=1-\alpha$.

При $m > 7$ се изчислява величината:

$$\chi_{\text{изч}}^2 = R(m-1)w_k \quad (37)$$

Коефициентът на съгласуваност w_k е значим, ако $\chi_{\text{изч}}^2 > \chi_{\text{табл.}}^2(\alpha, \nu)$

$\chi_{\text{табл.}}^2$ се взема от таблица (Приложение 2) при степени на свобода $\nu = m - 1$ и ниво на значимост α (доверителна вероятност $\beta=1-\alpha$).

- Изчисляване на тегловните коефициенти за всеки показател:

$$W_j = \frac{V_j}{\sum_{j=1}^m V_j} \quad (38)$$

За така получените тегловни коефициенти е изпълнено условието

$$\sum_{j=1}^m W_j = 1 \quad (39)$$

- Получените резултати се изобразяват графично чрез диаграми.

Прилагайки изложения подход за оценка на качеството се постигат следните предимства:

1. Предложеният подход е ориентиран към случаите, когато се прави количествена оценка на качеството на програмен продукт, включващ работните процеси на

отделните модули, за да се вземат оптимални решения при реализирането на системата с оглед удовлетворяване на изискванията на потребителите във възможно най-кратък срок.

2. Като основни критерии за качеството са използвани характеристики, базирани на основния стандарт ISO 9126 за оценяване на качеството на софтуерните продукти, но подходът не ограничава използването и на други критерии.

3. Повишава предсказуемостта на софтуерния продукт и понижава разходите на финанси и ресурси.

4. Данните се получават по експертен път, като подходът дава възможност за минимизиране на субективния фактор от мненията на специалистите, участващи в оценката, като се определя степента им на компетентност посредством **таблици 4.3, 4.4, 4.5, 4.6, 4.7, 4.8**.

5. Възможно е прилагането на подхода за различни софтуерни продукти в областта на Web-базираните приложения, като се отчетат особеностите на всяка от тях.

4.2.5 Графично представяне на резултатите

Получените резултати се визуализират графично чрез два типа диаграми. Резултатите от оценките на всеки един експерт поотделно се визуализират чрез Pie-диаграма. Крайните резултати, базирани на данните от тегловната матрица (**таблица 4.9**), се визуализират чрез Column-диаграма.

4.3 Приложение на предложения подход за количествена оценка на качеството и ефективността на системата

4.3.1 Подготовка и организация на изследването

На базата на описания подход в глава втора от дисертационния труд за количествена оценка на качеството и ефективността на система за управление на съдържанието, е направена оценка на система за Web-базирано обучение. Системата е интегрирана на сървър в Медицински Университет-София. Анкетирани са специалисти от четири фирми в областта на софтуерните и информационни технологии – MentorMate, Perceval България ЕООД, Kostal България ЕООД, Информационно обслужване – Варна, и два български университета, занимаващи се с електронно обучение – Медицински университет – София и Икономически университет – Варна. Броят на участниците е съответно 15 експерти от областта на софтуерните технологии и 13 експерти в областта на електронното обучение. За проведения експеримент, на всеки един участник в анкетирането му бе предоставен пълен достъп до оценяваната система, както и съответните анкетни карти, чрез които всеки един от тях да даде своята оценка и мнение за системата (приложения 3 и 4). Анкетните карти са две. Първата е съставена на база показатели включени в стандарт ISO 9126 за качество на софтуерни продукти. Показателите са шест на брой като всеки един от тях включва определен брой въпроси с предложени отговори. Тази анкетна карта е насочена към специалисти в разработването на софтуерни приложения. Втората анкетна карта е съставена на база дидактически показатели, съгласувани с експерти в областта на електронното обучение.

Състои се от седем въпроса с предложени отговори свързани с оценяване ефективността на системата от гледна точка на потребителското мнение.

4.3.2 Цел на изследването

Провеждането на изследването в дисертационния труд има няколко цели. Първата цел е да се получат мнения от страна на експерти, разработващи софтуерни приложения, които биха имали важна роля при по-нататъшно технологично и структурно подобряване и развитие на системата.

Втората цел е да се получат мнения от страна на потребителите на системата (студенти и преподаватели), които биха имали важна роля при определяне на ефективността на системата. Мненията на преподавателите имат съществено значение за по-нататъшно разширяване на системата с цел насърчаване на критичното мислене и креативност на студентите.

Третата цел е да се установи дали системата е достатъчно универсална, че да е възможно използването ѝ от други дисциплини и курсове в МУ-София и други университети.

4.4 Резултати от оценяването на системата

В тази част от дисертацията са показани обработените резултати от отговорите на анкетата.

4.4.1 Резултати от приложение на предложения подход за количествена оценка на качеството на системата на база показатели по стандарт ISO 9126

На $R=15$ на брой експерти се предлага да дадат своето мнение и оценка на качеството на система за Web-базирано обучение, като се вземат под внимание $m=6$ показатели (целеви параметъра) чрез анкетна карта (**Приложение 3**). За всеки един показател от анкетната карта се изчислява съответната оценка на базата на следното процентно съотношение $\frac{100\%}{x}$, където x е съответния брой на атрибутите на отделните показатели.

Вземайки в предвид предварително зададената скала (Скала: Слаб = 2 ; До бър = 4; Силен = 6; Не мога да преценя = 0), резултатите от направените изчисления са въведени в **таблици 4.10, 4.11, 4.12, 4.13, 4.14, 4.15, 4.16, 4.17, 4.18, 4.19, 4.20, 4.21, 4.22, 4.23, 4.24** и се визуализират чрез Pie-диаграма за всеки един показател. Така получените резултати се записват в тегловна матрица (**таблица 4.25**) и се визуализират графично чрез column-диаграма. На базата на получените резултати се оптимизира системата с оглед удовлетворяване на изискванията на потребителите.

Степента на компетентност на анкетираните експерти, определена чрез израза:

$$C = \frac{a_1+a_2+a_3}{R}, \text{ където } R=15 \text{ е броят на експертите. (Приложение 5).}$$

Таблица 4.10 *Експерт 1- резултати*

Показатели	Скала	Слаб	Добър	Силен	Не мога да преценя	точки
Функционалност			20%	60%	20%	4,4
Безотказност			25%	75%		5,5
Използваемост			40%	60%		5,2
Продуктивност			33,33%	66,66%		5,333
Обслужваемост			20%	80%		5,6
Преносимост			40%	40%	20%	4

Таблица 4.11 *Експерт 2- резултати*

Показатели	Скала	Слаб	Добър	Силен	Не мога да преценя	точки
Функционалност			20%	80%		5,6
Безотказност			25%	75%		5,5
Използваемост			20%	80%		5,6
Продуктивност			33,33%	66,66%		5,333
Обслужваемост			20%	80%		5,6
Преносимост			20%	80%		5,6

Таблица 4.12 *Експерт 3- резултати*

Показатели	Скала	Слаб	Добър	Силен	Не мога да преценя	точки
Функционалност				100%		6
Безотказност				50%	50%	3
Използваемост				100%		6
Продуктивност			66,66%	33,33%		4,666
Обслужваемост				60%	40%	3,6
Преносимост				40%	60%	2,4

Таблица 4.13 *Експерт 4- резултати*

Показатели	Скала	Слаб	Добър	Силен	Не мога да преценя	точки
Функционалност				100%		6
Безотказност			25%	50%	25%	4
Използваемост			20%	80%		5,6
Продуктивност				66,66%	33,33%	4
Обслужваемост			40%	60%		5,2
Преносимост			20%	80%		5,6

Таблица 4.14 *Експерт 5- резултати*

Показатели	Скала	Слаб	Добър	Силен	Не мога да преценя	точки
Функционалност			20%	80%		5,6
Безотказност			20%	75%		5,5
Използваемост			20%	80%		5,6
Продуктивност				99,99%		5,999
Обслужваемост			20%	80%		5,6
Преносимост				100%		6

Таблица 4.15 *Експерт 6- резултати*

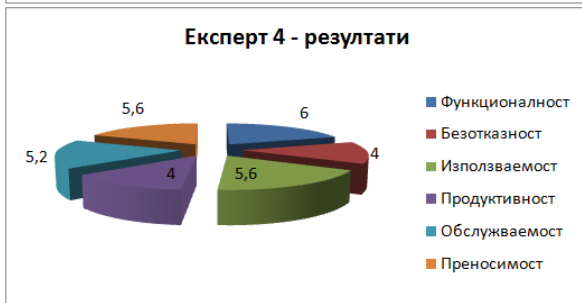
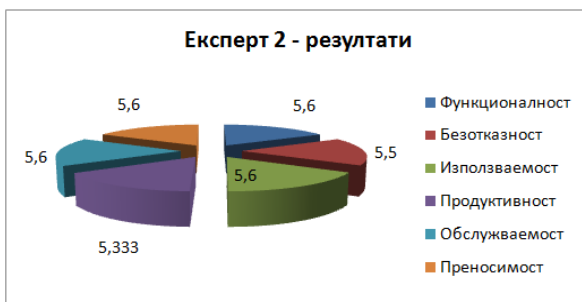
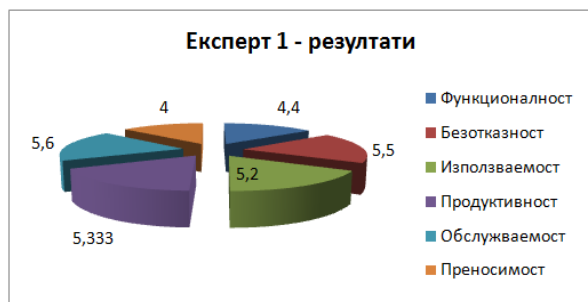
Показатели	Скала	Слаб	Добър	Силен	Не мога да преценя	точки
Функционалност				100%		6
Безотказност				100%		6
Използваемост			20%	80%		5,6
Продуктивност			33,33%	66,66%		5,333
Обслужваемост			20%	80%		5,6
Преносимост			20%	80%		5,6

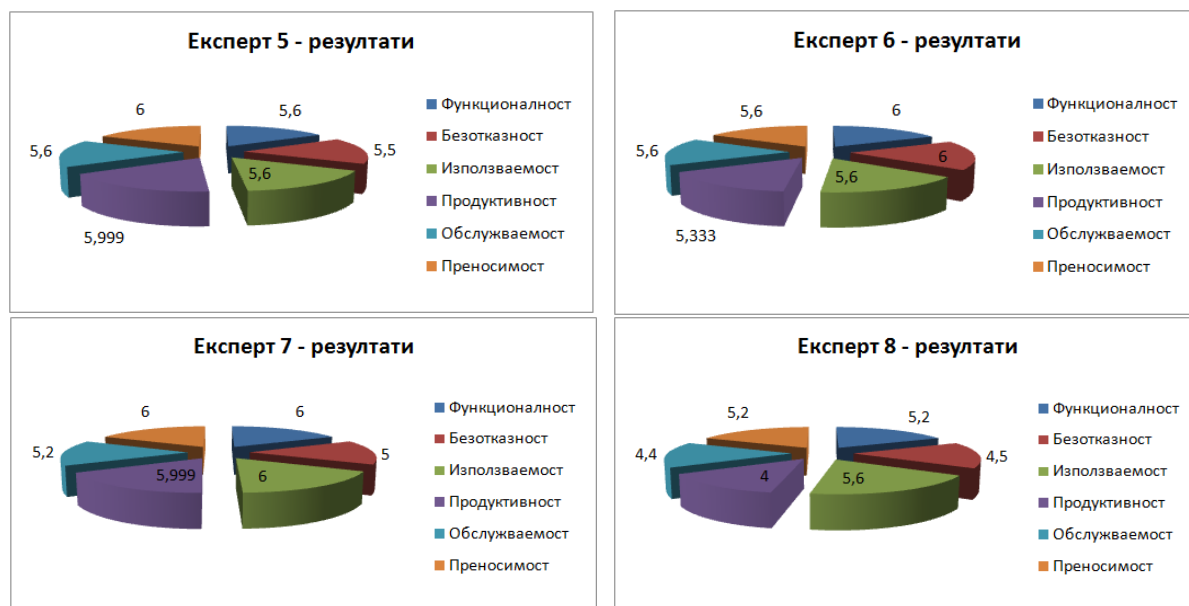
Таблица 4.16 *Експерт 7- резултати*

Показатели	Скала	Слаб	Добър	Силен	Не мога да преценя	точки
Функционалност				100%		6
Безотказност			50%	50%		5
Използваемост				100%		6
Продуктивност				99,99%		5,999
Обслужваемост			40%	60%		5,2
Преносимост				100%		6

Таблица 4.17 *Експерт 8- резултати*

Показатели	Скала	Слаб	Добър	Силен	Не мога да преценя	точки
Функционалност			40%	60%		5,2
Безотказност			75%	25%		4,5
Използваемост			20%	80%		5,6
Продуктивност			99,99%			4
Обслужваемост			80%	20%		4,4
Преносимост			40%	60%		5,2





Фигура 4.1 Графично представяне на отговорите на експерти от 1 до 8

Таблица 4.18 Експерт 9- резултати

Показатели	Скала	Слаб	Добър	Силен	Не мога да преценя	точки
Функционалност			60%	40%		4,8
Безотказност			75%	25%		4,5
Използваемост			40%	60%		5,2
Продуктивност			66,66%		33,33%	2,666
Обслужваемост			80%		20%	3,2
Преносимост			60%	40%		4,8

Таблица 4.19 Експерт 10- резултати

Показатели	Скала	Слаб	Добър	Силен	Не мога да преценя	точки
Функционалност			20%	40%	40%	3,2
Безотказност			50%		50%	2
Използваемост			40%	60%		5,2
Продуктивност			66,66%		33,33%	2,666
Обслужваемост			60%		40%	2,4
Преносимост			40%	60%		5,2

Таблица 4.20 Експерт 11- резултати

Показатели	Скала	Слаб	Добър	Силен	Не мога да преценя	точки
Функционалност			20%	40%	40%	3,2
Безотказност			50%	25%	25%	3,5
Използваемост			20%	60%	20%	4,4
Продуктивност			66,66%		33,33%	2,666
Обслужваемост			60%		40%	2,4
Преносимост			40%	40%	20%	4

Таблица 4.21 Експерт 12- резултати

Показатели	Скала	Слаб	Добър	Силен	Не мога да преценя	точки
Функционалност			20%	40%	40%	3,2
Безотказност				50%	50%	3
Използваемост			100%			4
Продуктивност			66,66%	33,33%		4,666
Обслужваемост			40%	20%	40%	2,8
Преносимост			40%	20%	40%	2,8

Таблица 4.22 Експерт 13- резултати

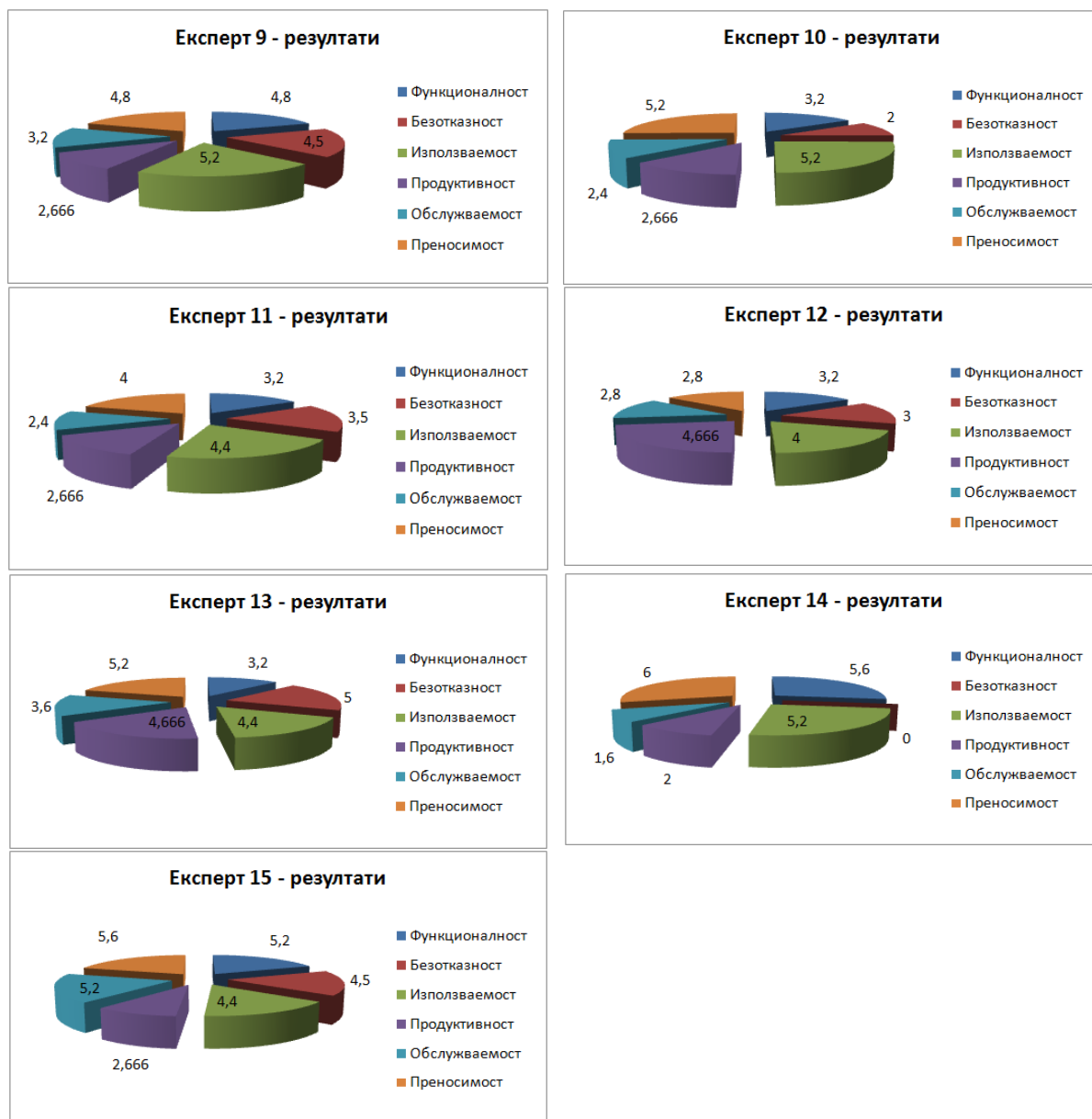
Показатели	Скала	Слаб	Добър	Силен	Не мога да преценя	точки
Функционалност			80%		20%	3,2
Безотказност			50%	50%		5
Използваемост			80%	20%		4,4
Продуктивност			66,66%	33,33%		4,666
Обслужваемост			60%	20%	20%	3,6
Преносимост			40%	60%		5,2

Таблица 4.23 Експерт 14- резултати

Показатели	Скала	Слаб	Добър	Силен	Не мога да преценя	точки
Функционалност			20%	80%		5,6
Безотказност					100%	0
Използваемост			40%	60%		5,2
Продуктивност	33,33		33,33%		33,33%	2
Обслужваемост			40%		60%	1,6
Преносимост				100%		6

Таблица 4.24 Експерт 15- резултати

Показатели	Скала	Слаб	Добър	Силен	Не мога да преценя	точки
Функционалност			40%	60%		5,2
Безотказност			75%	25%		4,5
Използваемост			20%	60%	20%	4,4
Продуктивност			66,66%		33,33%	2,666
Обслужваемост			40%	60%		5,2
Преносимост			20%	80%		5,6



Фигура 4.2 Графично представяне на отговорите на експерти от 9 до 15

Точките в последната колона от таблицата се изчисляват по следния израз:

Приемаме, че:

$x_1 \Rightarrow$ слаб = 2;

$x_2 \Rightarrow$ добър = 4;

$x_3 \Rightarrow$ силен = 6;

$x_4 \Rightarrow$ не мога да преценя = 0;

u_1, u_2, u_3, u_4 - стойности за всеки показател в проценти

$P_{\text{точки}}$ – стойност на крайната оценка за всеки един показател на анкетирания експерт

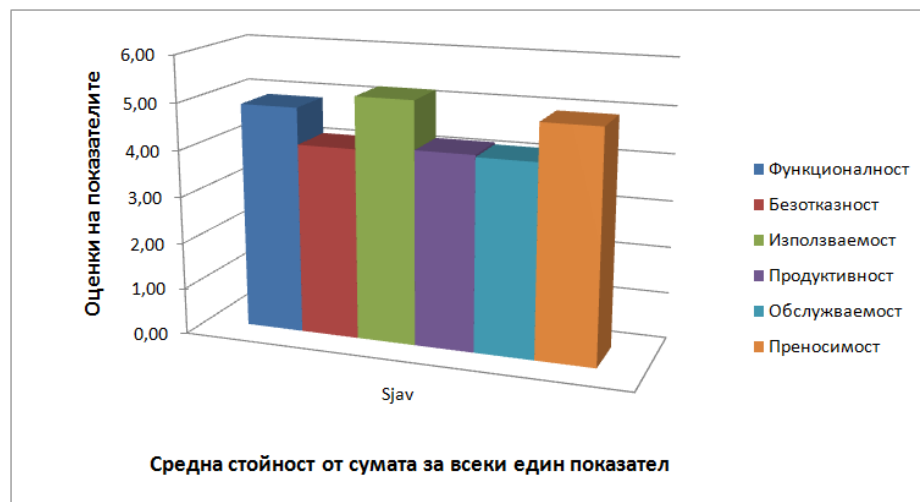
Следователно: $P_{\text{точки}} = x_1 u_1 + x_2 u_2 + x_3 u_3 + x_4 u_4$

Резултатите от анкетирането са обобщени в **таблица 4.25**. Средната сума от всички точки (оценки), определена по формула (30) е:

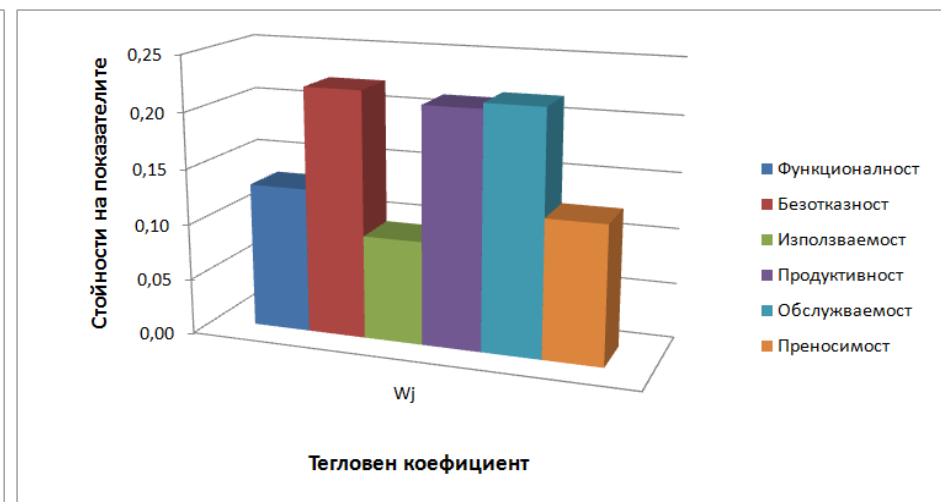
$$S_{av} = \frac{R(m+1)}{2} = \frac{15(6+1)}{2} = 52,5$$

Таблица 4.25 Тегловна матрица

Експерти Показатели	E_{x_1}	E_{x_2}	E_{x_3}	E_{x_4}	E_{x_5}	E_{x_6}	E_{x_7}	E_{x_8}	E_{x_9}	$E_{x_{10}}$	$E_{x_{11}}$	$E_{x_{12}}$	$E_{x_{13}}$	$E_{x_{14}}$	$E_{x_{15}}$	S_j	S_{jav}	δ_j	V_j	W_j
Функционалност	4,4	5,6	6	6	5,6	6	6	5,2	4,8	3,2	3,2	3,2	3,2	5,6	5,2	73,20	4,88	20,70	0,22	0,13
Безотказност	5,5	5,5	3	4	5,5	6	5	4,5	4,5	2	3,5	3	5	0	4,5	61,50	4,10	9,00	0,38	0,22
Използваемост	5,2	5,6	6	5,6	5,6	5,6	6	5,6	5,2	5,2	4,4	4	4,4	5,2	4,4	78,00	5,20	25,50	0,16	0,09
Продуктивност	5,333	5,333	4,666	4	5,999	5,333	5,999	4	2,666	2,666	2,666	4,666	4,666	2	2,666	62,66	4,18	10,16	0,36	0,21
Обслужваемост	5,6	5,6	3,6	5,2	5,6	5,6	5,2	4,4	3,2	2,4	2,4	2,8	3,6	1,6	5,2	62,00	4,13	9,50	0,37	0,22
Преносимост	4	5,6	2,4	5,6	6	5,6	6	5,2	4,8	5,2	4	2,8	5,2	6	5,6	74,00	4,93	21,50	0,21	0,12



Фигура 4.3 Графично представяне на резултатите за средната стойност на групата от експерти за всеки показател



Фигура 4.4 Графично представяне на резултатите на тегловния коефициент за всеки показател

Въз основа на данните от **таблица 4.25** са изчислени сумата от оценките на всички експерти S_j за всеки един показател по формула (28), средната стойност на групата от експерти S_{jav} за всеки показател по формула (29) и отклонението δ_j за степента на съгласуване на мненията на експертите по формула (31) и резултатите са дадени в същата таблица. Коефициентът на съгласие по формула (32) се получава:

$$w_k = \frac{12 \sum_{j=1}^m \delta_j^2}{R^2(m^3 - m)} = \frac{12 \cdot 96,36^2}{15^2(6^3 - 6)} = 2,36$$

Значимостта на w_k се проверява по Z - критерия на Фишер по формула (34).

$$Z = \frac{1}{2} \ln \frac{(R-1)w_k}{1-w_k} = \frac{1}{2} \ln \frac{(15-1)2,36}{1-2,36} = 1,595$$

Тегловните коефициенти W_j са пресметнати по формула (38), където V_j е коефициентът на вариация, характеризиращ съгласуваността на оценките на експертите и се изчислява по формула (33). Резултатите са дадени в **таблица 4.25**. От **фигура 4.3** става ясно, че системата е оценена най-високо по критерии Използваемост, Функционалност и Преносимост. Изчислените тегловни коефициенти, представени графично на **фигура 4.4**, показват, че системата е оценена най-ниско по критерии Безотказност, Обслужваемост и Продуктивност.

4.4.2 Дефиниране на задачата за оптимизация на процесите на база софтуерни показатели по стандарт ISO 9126

В дисертационния труд е представена експертната оценка за разработената информационна система за Медицински университет - София. Критериите за оценка на системата са дефинирани и представени в **таблица 4.1**. Анализът на оценките на експертите е обобщен в **таблица 4.25**.

Изследванията в дисертационния труд целят да се оцени потенциала на експертните мнения и да се определи в каква посока може да се подобряват характеристиките на информационната система. Оценяването на този потенциал е извършено чрез дефиниране и решаване на съответна оптимизационна задача. Като параметри на тази задача са използвани изчислените стойности на експертните мнения като теглови коефициенти, корелационни коефициенти.

Разработени са две оптимизационни задачи.

Задача 1:

Оценява се доколко може да се подобрят характеристиките на информационната система като се приеме отклонение от средните стойности, определени от експертните мнения.

Целевата функция на Задача 1 е дефинирана в следния вид:

$$\max_x \left\{ \sum_{j=1}^6 (x_j - \bar{x}_j)^2 W_j \right\} \quad (40)$$

Аргументът x има съдържание на промяна на j -ти критерий за оценка спрямо оценената средна стойност $\bar{x}_j$, така че сумарният ефект от промяната да има максимална стойност. Използват се коефициентите W_j , които са теглови коефициенти за важност на отделните показатели, определени от експертните оценки.

За ограничения на оптимизационната задача са дефинирани връзки между отделните критерии, които си влияят при съвместната оценка на информационната система. Например, критериите $j=1, 5$ съвместно са включени в общата оценка, но те си влияят съгласно определените коефициенти на вариация V_j . Така се дефинира ограничението (40а). Дясната част на ограничението определя, че оценките могат да се съгласуват в системата, формализирано чрез относителна единица.

$$V_1x_1 + V_2x_2 + V_3x_3 + V_4x_4 + V_5x_5 + V_6x_6 \geq 1 \quad (40a)$$

Допълнително са включени връзки между критериите $j=1, 3, 4$ (функционалност, използваемост и продуктивност). Физическото съдържание на това ограничение е, че промяната на критерий $j=1$ (функционалност) ще повлияе на критерии $j=3, 4$ (използваемост и продуктивност). Влиянието между критериите количествено прилага тегловите коефициенти W_j , получени от оценките на експертите. Дясната част на ограничението определя, че оценките се съгласуват в системата, формализирано чрез относителна единица. Ограничението има аналитичен вид (40б),

$$W_1x_1 + W_3x_3 + W_4x_4 \geq 1 \quad (40б)$$

Аналогични формални записи са приложени за критериите $j=2, 3, 5$ (безотказност, използваемост и обслужваемост), което е представено със зависимост (40в),

$$W_2x_2 + W_3x_3 + W_5x_5 \geq 1 \quad (40в)$$

а за критериите $j=3, 6$ (използваемост и преносимост) се дефинира ограничението (40г),

$$W_3x_3 + W_6x_6 \geq 1 \quad (40г)$$

Общият аналитичен вид на дефинираната задача за оптимизация е:

$$\max_x \left\{ \sum_{j=1}^6 (x_j - \bar{x}_j)^2 W_j \right\} \quad (41)$$

$$V_1x_1 + V_2x_2 + V_3x_3 + V_4x_4 + V_5x_5 + V_6x_6 \geq 1$$

$$W_1x_1 + W_3x_3 + W_4x_4 \geq 1$$

$$W_2x_2 + W_3x_3 + W_5x_5 \geq 1$$

$$W_3x_3 + W_6x_6 \geq 1$$

$$x_j \geq 0$$

където

x_j определя стойността на критерии j , където $j=1..6$, който може да се изменя спрямо определените средни стойности, така че общата положителна промяна за системата да е максимална.

$\bar{x}_j$ е средна стойност на критериите за оценка, определени от експертното мнение, отговаряща на S_{jav} (таблица 4.25)

W_j - тегловни коефициенти, отразяващи тежестта на критериите, определени от експертното мнение

V_j - коефициент на вариация, характеризиращ връзката между критериите, определени от оценките на експертите

Ограниченията на задачата отчитат свързаността между отделните критерии съгласно оценките на експертите.

Решенията на оптимизационната задача определят потенциала за подобряване на информационната система по критерии $j=1..6$.

Оптимизационната задача е решена с програмния продукт MATLAB чрез неговата функция *quadprog*. Функцията *quadprog* има следния вид:

$$\max_x \left[\frac{1}{2} x_j^T Q x_j + R^T x_j \right] \begin{cases} A \cdot x_j \leq b \\ Aeq \cdot x_j = beq \\ lb \leq x_j \leq ub \\ x_j \text{ е векторно число} \end{cases} \quad (42)$$

където Q, A, Aeq, R, b, lb, ub , са матрици, beq и x_j са вектори.

Задача (42) съгласно формата на MATLAB се представя във вида

$$x_j = \text{quadprog}(Q, R, A, b) \quad (43)$$

и съответно за разглеждания случай задача (42) добива вида

$$\max_x \left[\frac{1}{2} x_j^T Q x_j + R^T x_j \right], \text{ като } A \cdot x_j \geq b \quad (44)$$

Матриците Q, A, R, b са от вида,

$$Q = \begin{bmatrix} W_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & W_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & W_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & W_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & W_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & W_6 \end{bmatrix} \quad (45)$$

$$R = \begin{bmatrix} -\bar{x}_1 & W_1 \\ -\bar{x}_2 & W_2 \\ -\bar{x}_3 & W_3 \\ -\bar{x}_4 & W_4 \\ -\bar{x}_5 & W_5 \\ -\bar{x}_6 & W_6 \end{bmatrix} \quad (46)$$

$$A = \begin{bmatrix} V_1 & V_2 & V_3 & V_4 & V_5 & V_6 \\ W_1 & 0 & W_3 & W_4 & 0 & 0 \\ 0 & W_2 & W_3 & 0 & W_5 & 0 \\ 0 & 0 & W_3 & 0 & 0 & W_6 \end{bmatrix} \quad (47)$$

$$b = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad (48)$$

Решенията на оптимизационната задача са представени в колони $\bar{x}_j$ и x_j на **таблица 4.26**, където

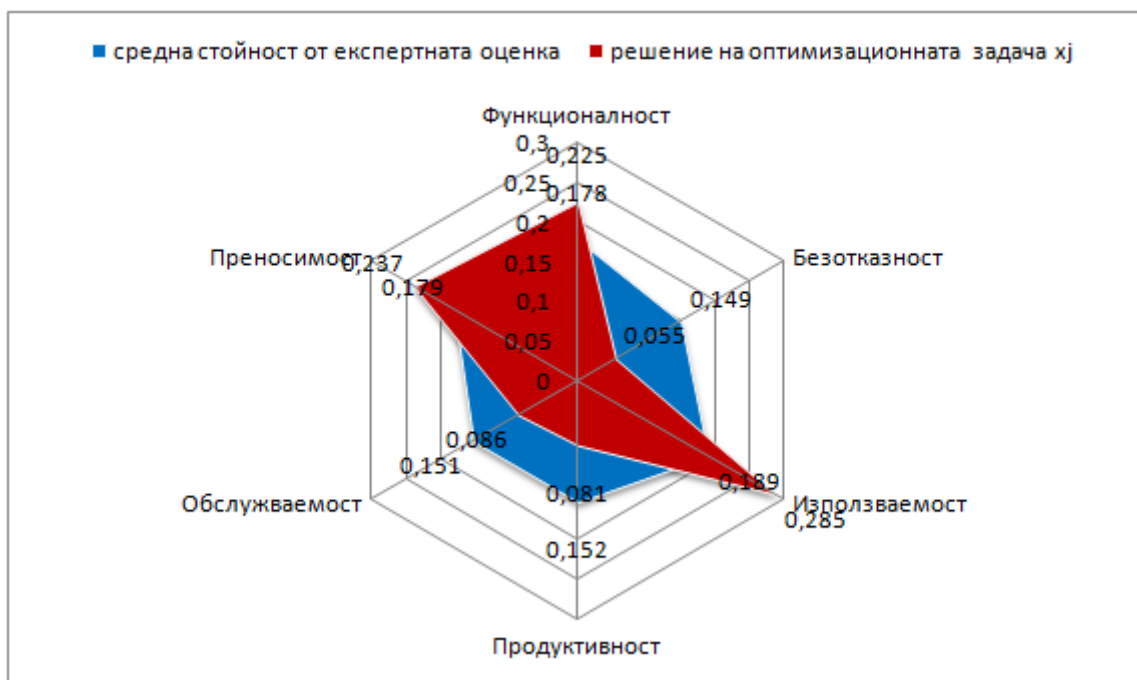
$\bar{x}_j$ е средна стойност от експертната оценка, отговаряща на S_{jav}

x_j е решението на оптимизационната задача и определя доколко може да се повиши стойността на критерия спрямо средните оценки, определени от експертите (потенциал на експертната оценка)

Таблица 4.26 Решения на задача 1

Експерти		V_j	W_j	$\bar{x}_j$	x_j
Показатели					
Функционалност	x_1	0,22	0,13	0,178	0,225
Безотказност	x_2	0,38	0,22	0,149	0,055
Използваемост	x_3	0,16	0,09	0,189	0,285
Продуктивност	x_4	0,36	0,21	0,152	0,081
Обслужваемост	x_5	0,37	0,22	0,151	0,086
Преносимост	x_6	0,21	0,12	0,179	0,237

За нагледност резултатите от решената оптимизационна задача са представени посредством радарна диаграма, **фигура 4.5**:



Фигура 4.5: Сравнение между средна стойност от експертната оценка $\bar{x}_j$ и решението на оптимизационната задача x_j

На радарната диаграма е илюстрирано сравнение на получените резултати от мненията на експертите между средната стойност от експертната оценка $\bar{x}_j$ на критериите, базирана на стойностите на S_{jav} (със син цвят на диаграмата) и стойността на решението на оптимизационната задача x_j (с червен цвят на диаграмата). От резултатите на оптимизационната задача и съответно диаграмата следва, че по критерии функционалност, използваемост и преносимост съществува по-голям

потенциал за подобряване на процесите и бъдещо развитие на системата в сравнение с другите критерии: безотказност, продуктивност и обслужваемост. Решението на оптимизационната задача показва, че за разработената информационна система за web-базирано обучение експертите считат, че някои от показателите са с по-голям приоритет отколкото други показатели. Количествено това се оценява чрез изчислените теглови коефициенти W_j . Получените решения на оптимизационната задача дават количествена оценка кои критерии могат да бъдат подобрявани при модифициране на информационната система. Това подобряване е препоръчително по критерии преносимост, функционалност и използваемост.

Задача 2:

Оптимизацията на информационната система е правена и при оценка доколко може да се подобри съществуващата система без да се намалят стойностите на критериите, които са оценени от експертите. Формално, в тази задача се поставя изискването стойността на критерия x_j да не се намали под стойността на средната стойност $\bar{x}_j$. Дефинираната задача (41) е модифицирана, за да се определи потенциала на експертните оценки за повишаване на средната стойност на оценките. Аналитичният запис на това ограничение е: $\bar{x}_j \leq x_j$

$$\max_x \left\{ \sum_{j=1}^6 (x_j - \bar{x}_j)^2 W_j \right\} \quad (49)$$

$$V_1 x_1 + V_2 x_2 + V_3 x_3 + V_4 x_4 + V_5 x_5 + V_6 x_6 \geq 1$$

$$W_1 x_1 + W_3 x_3 + W_4 x_4 \geq 1$$

$$W_2 x_2 + W_3 x_3 + W_5 x_5 \geq 1$$

$$W_3 x_3 + W_6 x_6 \geq 1$$

$$\bar{x}_j \leq x_j$$

Задача (42) съгласно формата на MATLAB се представя във вида

$$x_j = quadprog(Q, R, A, b, A_{eq}, b_{eq}, l_b, u_b) \quad (50)$$

където, $A_{eq} = 0$ и $b_{eq} = 0$, тъй като не съществува равенство

и съответно за разглеждания случай задача (42) добива вида

$$\max_x \left[\frac{1}{2} x_j^T Q x_j + R^T x_j \right], \text{ като } l_b \leq x \leq u_b \quad (51)$$

Където l_b и u_b са матрици като u_b е произволно избрана горна граница

$$l_b = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \\ \bar{x}_4 \\ \bar{x}_5 \\ \bar{x}_6 \end{bmatrix} \quad (52)$$

Решенията на оптимизационната задача са представени в колони $\bar{x}_j$ и x_j на **таблица 4.27**, където

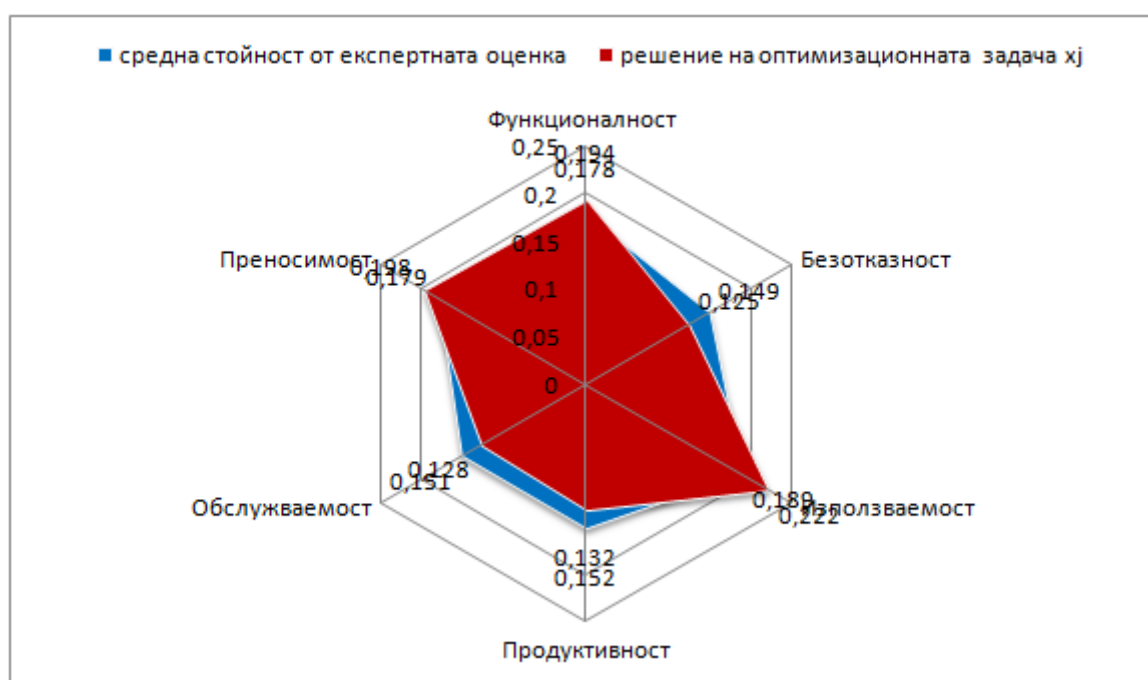
$\bar{x}_j$ е средните стойности на критериите определени от експертните оценки, отговаряща на S_{jav}

x_j е решението на оптимизационната задача и определя доколко може да се повиши стойността на критерия спрямо средните оценки, определени от експертите (потенциал на експертната оценка)

Таблица 4.27 Решения на задача 2

Експерти		V_j	W_j	$\bar{x}_j$	x_j
Показатели					
Функционалност	x_1	0,22	0,13	0,178	0,194
Безотказност	x_2	0,38	0,22	0,149	0,125
Използваемост	x_3	0,16	0,09	0,189	0,222
Продуктивност	x_4	0,36	0,21	0,152	0,132
Обслужваемост	x_5	0,37	0,22	0,151	0,128
Преносимост	x_6	0,21	0,12	0,179	0,198

За нагледност резултатите от решената оптимизационна задача са представени посредством радарна диаграма, **фигура 4.6**:



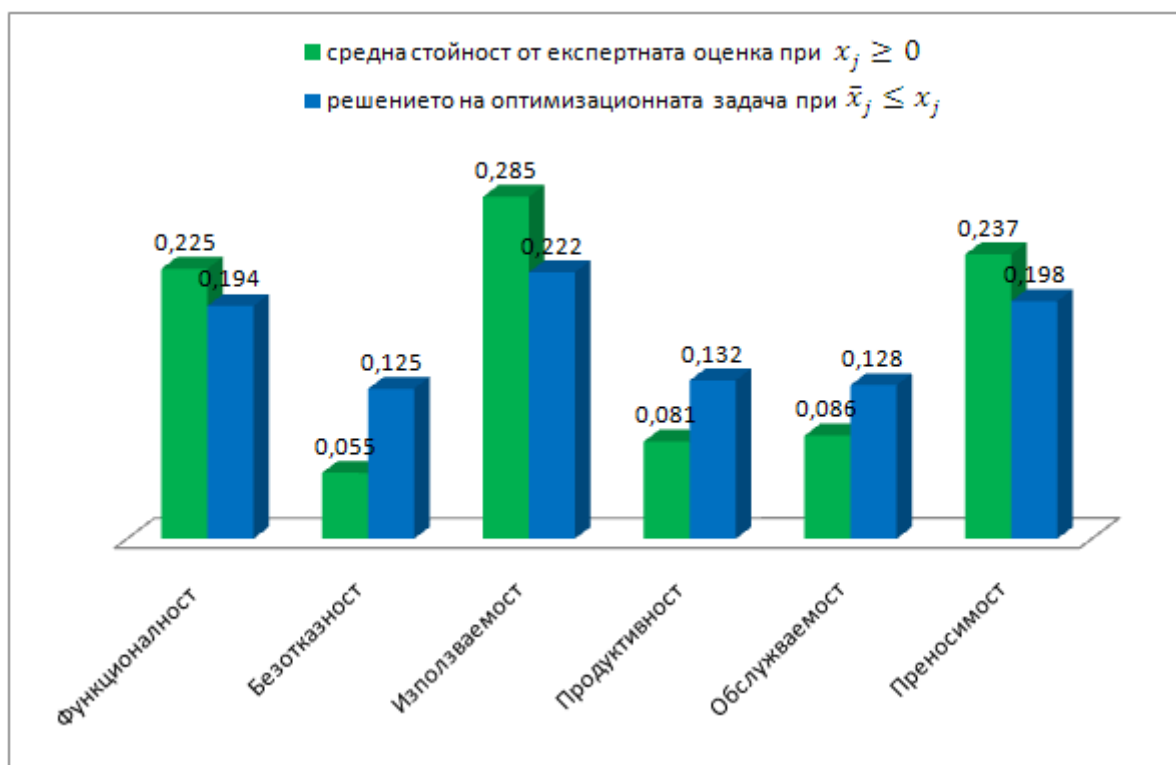
Фигура 4.6: Сравнение между средна стойност от експертната оценка $\bar{x}_j$ и решението на оптимизационната задача x_j

На радарната диаграма е илюстрирано сравнение на получените резултати от решението на оптимизационната задача x_j и изчислените средни стойности $\bar{x}_j$, определени от експертните мнения. Резултатите показват, че експертните мнения имат потенциал за увеличаване на стойностите на критериите използваемост,

функционалност и преносимост. Това решение съвпада с полученото в Задача 1. Количествените изменения в Задача 2 са по-малки защото е заложено изискване за запазване на критериите в малък диапазон около средните стойности $\bar{x}_j$.

Вземайки предвид получените резултати от радарните диаграми на **фигури 4.5 и 4.6** е направено сравнение между решенията на двете оптимизационни задачи (**фигура 4.7**). Със зелен цвят са решенията на задача (1), а със син цвят са решенията на задача (2). Вижда се, че критериите функционалност, използваемост и преносимост имат потенциал за повишение и при двете задачи. Но при Задача 2 критериите безотказност, продуктивност и обслужваемост са в околността на изчислените средни стойности $\bar{x}_j$, определени от експертната оценката за разработената информационна система.

Резултатът от дефинирането и решаването на оптимизационните задачи показва, че сегашната информационна система може да бъде подобрявана по 3 от по-важните критерии за оценка. Експертното мнение съдържа потенциал за увеличаването на стойностите на тези критерии.



Фигура 4.7: Сравнение между решенията на двете оптимизационни задачи

4.4.3 Резултати от приложение на предложения подход за оценка на ефективността на системата на база дидактически показатели

На **R=13** на брой експерти (преподаватели) се предлага да дадат своето мнение и оценка на ефективността на система за Web-базирано обучение, като се вземат под внимание **m=7** показатели (целови параметъра) чрез анкетна карта (**Приложение 4**). За всеки един показател от анкетната карта се изчислява съответната оценка на базата на

следното процентно съотношение $\frac{100\%}{x}$, където x е съответния брой на атрибутите на отделните показатели. Вземайки в предвид предварително зададената скала (Скала: Слаб = 2; Добър = 4; Силен = 6; Не мога да преценя = 0), резултатите от направените изчисления са въведени в **таблици 4.28, 4.29, 4.30, 4.31, 4.32, 4.33, 4.34, 4.35, 4.36, 4.37, 4.38, 4.39, 4.40** и се визуализират чрез Pie-диаграма за всеки един показател. Така получените резултати се записват в тегловна матрица (**таблица 4.41**) и се визуализират графично чрез column-диаграма. На базата на получените резултати се оптимизира системата с оглед удовлетворяване на изискванията на потребителите.

Степента на компетентност на анкетирания преподаватели се определя чрез израза:

$$C = \frac{a_1 + a_2 + a_3}{R}, \text{ където } R=13 \text{ е броят на експертите. (Приложение 6)}$$

Резултатите от анкетирания са дадени в **таблица 4.41**. Средната сума от всички точки (оценки), определена по формула (30) е:

$$S_{av} = \frac{R(m+1)}{2} = \frac{13(7+1)}{2} = 52$$

Таблица 4.28 Преподавател 1- резултати

Показатели	Скала	Слаб	Добър	Силен	Не мога да преценя	точки
Степен на логическа последователност на учебния материал				100%		6
Свързаност на компонентите на учебния материал				100%		6
Насърчаване за критично мислене и креативност				100%		6
Връзка с други ресурси за допълнителна информация за изучавания проблем				100%		6
Използване на различни мултимедийни компоненти				100%		6
Наличие на речник на използваните термини в лекционния материал				100%		6
Наличие на търсеща машина в основните модули				100%		6

Таблица 4.29 Преподавател 2- резултати

Показатели	Скала	Слаб	Добър	Силен	Не мога да преценя	точки
Степен на логическа последователност на учебния материал				100%		6
Свързаност на компонентите на учебния материал				100%		6
Насърчаване за критично мислене и креативност				100%		6
Връзка с други ресурси за допълнителна информация за изучавания проблем				100%		6
Използване на различни мултимедийни компоненти				100%		6
Наличие на речник на използваните термини в лекционния материал				100%		6
Наличие на търсеща машина в основните модули				100%		6

Таблица 4.30 Преподавател 3- резултати

Показатели	Скала	Слаб	Добър	Силен	Не мога да преценя	точки
Степен на логическа последователност на учебния материал				100%		6
Свързаност на компонентите на учебния материал				100%		6
Насърчаване за критично мислене и креативност				100%		6
Връзка с други ресурси за допълнителна информация за изучавания проблем				100%		6
Използване на различни мултимедийни компоненти				100%		6
Наличие на речник на използваните термини в лекционния материал				100%		6
Наличие на търсеща машина в основните модули				100%		6

Таблица 4.31 Преподавател 4- резултати

Показатели	Скала	Слаб	Добър	Силен	Не мога да преценя	точки
Степен на логическа последователност на учебния материал				100%		6
Свързаност на компонентите на учебния материал				100%		6
Насърчаване за критично мислене и креативност				100%		6
Връзка с други ресурси за допълнителна информация за изучавания проблем				100%		6
Използване на различни мултимедийни компоненти				100%		6
Наличие на речник на използваните термини в лекционния материал				100%		6
Наличие на търсеща машина в основните модули				100%		6

Таблица 4.32 Преподавател 5- резултати

Показатели	Скала	Слаб	Добър	Силен	Не мога да преценя	точки
Степен на логическа последователност на учебния материал				100%		6
Свързаност на компонентите на учебния материал				100%		6
Насърчаване за критично мислене и креативност				100%		6
Връзка с други ресурси за допълнителна информация за изучавания проблем				100%		6
Използване на различни мултимедийни компоненти				100%		6
Наличие на речник на използваните термини в лекционния материал				100%		6
Наличие на търсеща машина в основните модули				100%		6

Таблица 4.33 Преподавател 6- резултати

Показатели	Скала	Слаб	Добър	Силен	Не мога да преценя	точки
Степен на логическа последователност на учебния материал				100%		6
Свързаност на компонентите на учебния материал				100%		6
Насърчаване за критично мислене и креативност				100%		6
Връзка с други ресурси за допълнителна информация за изучавания проблем				100%		6
Използване на различни мултимедийни компоненти				100%		6
Наличие на речник на използваните термини в лекционния материал				100%		6
Наличие на търсеща машина в основните модули				100%		6

Таблица 4.34 Преподавател 7- резултати

Показатели	Скала	Слаб	Добър	Силен	Не мога да преценя	точки
Степен на логическа последователност на учебния материал				100%		6
Свързаност на компонентите на учебния материал				100%		6
Насърчаване за критично мислене и креативност			100%			4
Връзка с други ресурси за допълнителна информация за изучавания проблем				100%		6
Използване на различни мултимедийни компоненти			100%			4
Наличие на речник на използваните термини в лекционния материал				100%		6
Наличие на търсеща машина в основните модули				100%		6

Таблица 4.35 Преподавател 8- резултати

Показатели	Скала	Слаб	Добър	Силен	Не мога да преценя	точки
Степен на логическа последователност на учебния материал				100%		6
Свързаност на компонентите на учебния материал				100%		6
Насърчаване за критично мислене и креативност				100%		6
Връзка с други ресурси за допълнителна информация за изучавания проблем			100%			4
Използване на различни мултимедийни компоненти			100%			4
Наличие на речник на използваните термини в лекционния материал				100%		6
Наличие на търсеща машина в основните модули				100%		6

Таблица 4.36 Преподавател 9- резултати

Показатели	Скала	Слаб	Добър	Силен	Не мога да преценя	точки
Степен на логическа последователност на учебния материал				100%		6
Свързаност на компонентите на учебния материал				100%		6
Насърчаване за критично мислене и креативност				100%		6
Връзка с други ресурси за допълнителна информация за изучавания проблем			100%			4
Използване на различни мултимедийни компоненти				100%		6
Наличие на речник на използваните термини в лекционния материал				100%		6
Наличие на търсеща машина в основните модули				100%		6

Таблица 4.37 Преподавател 10- резултати

Показатели	Скала	Слаб	Добър	Силен	Не мога да преценя	точки
Степен на логическа последователност на учебния материал				100%		6
Свързаност на компонентите на учебния материал				100%		6
Насърчаване за критично мислене и креативност			100%			4
Връзка с други ресурси за допълнителна информация за изучавания проблем			100%			4
Използване на различни мултимедийни компоненти				100%		6
Наличие на речник на използваните термини в лекционния материал				100%		6
Наличие на търсеща машина в основните модули				100%		6

Таблица 4.38 Преподавател 11- резултати

Показатели	Скала	Слаб	Добър	Силен	Не мога да преценя	точки
Степен на логическа последователност на учебния материал				100%		6
Свързаност на компонентите на учебния материал			100%			4
Насърчаване за критично мислене и креативност				100%		6
Връзка с други ресурси за допълнителна информация за изучавания проблем			100%			4
Използване на различни мултимедийни компоненти				100%		6
Наличие на речник на използваните термини в лекционния материал				100%		6
Наличие на търсеща машина в основните модули				100%		6

Таблица 4.39 Преподавател 12- резултати

Показатели	Скала	Слаб	Добър	Силен	Не мога да преценя	точки
Степен на логическа последователност на учебния материал			100%			4
Свързаност на компонентите на учебния материал				100%		6
Насърчаване за критично мислене и креативност				100%		6
Връзка с други ресурси за допълнителна информация за изучавания проблем			100%			4
Използване на различни мултимедийни компоненти			100%			4
Наличие на речник на използваните термини в лекционния материал				100%		6
Наличие на търсеща машина в основните модули				100%		6

Таблица 4.40 Преподавател 13- резултати

Показатели	Скала	Слаб	Добър	Силен	Не мога да преценя	точки
Степен на логическа последователност на учебния материал				100%		6
Свързаност на компонентите на учебния материал				100%		6
Насърчаване за критично мислене и креативност				100%		6
Връзка с други ресурси за допълнителна информация за изучавания проблем				100%		6
Използване на различни мултимедийни компоненти				100%		6
Наличие на речник на използваните термини в лекционния материал				100%		6
Наличие на търсеща машина в основните модули				100%		6

Преподавател 1- резултати



Преподавател 2- резултати



Преподавател 3- резултати

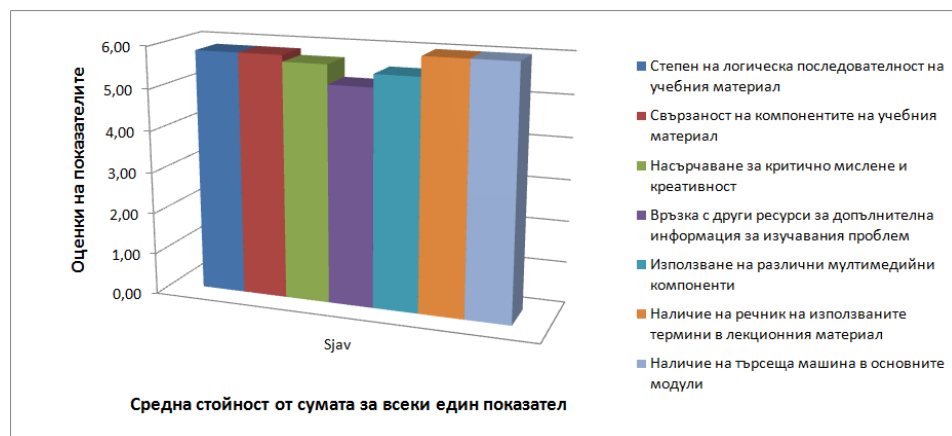


Преподавател 4- резултати

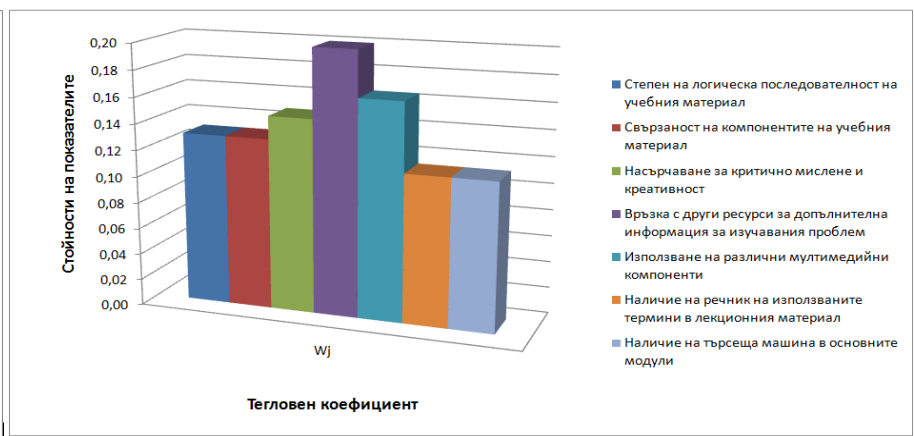


Таблица 4.41 Тегловна матрица

Показатели	Експерти	Ех ₁	Ех ₂	Ех ₃	Ех ₄	Ех ₅	Ех ₆	Ех ₇	Ех ₈	Ех ₉	Ех ₁₀	Ех ₁₁	Ех ₁₂	Ех ₁₃	S _j	S _{jav}	δ _j	V _j	W _j
Степен на логическа последователност на учебния материал		6	6	6	6	6	6	6	6	6	6	6	4	6	76,00	5,85	24,00	0,19	0,13
Свързаност на компонентите на учебния материал		6	6	6	6	6	6	6	6	6	6	4	6	6	76,00	5,85	24,00	0,19	0,13
Насърчаване за критично мислене и креативност		6	6	6	6	6	6	4	6	6	4	6	6	6	74,00	5,69	22,00	0,22	0,15
Връзка с други ресурси за допълнителна информация за изучавания проблем		6	6	6	6	6	6	6	4	4	4	4	4	6	68,00	5,23	16,00	0,29	0,20
Използване на различни мултимедийни компоненти		6	6	6	6	6	6	4	4	6	6	6	4	6	72,00	5,54	20,00	0,24	0,17
Наличие на речник на използваните термини в лекционния материал		6	6	6	6	6	6	6	6	6	6	6	6	6	78,00	6	26,00	0,17	0,11
Наличие на търсеща машина в основните модули		6	6	6	6	6	6	6	6	6	6	6	6	6	78,00	6	26,00	0,17	0,11



Фигура 4.9 Графично представяне на резултатите за средната стойност на групата от преподаватели за всеки показател



Фигура 4.10 Графично представяне на резултатите на тегловния коефициент за всеки показател

Въз основа на данните от **таблица 4.41** са изчислени сумата от оценките на всички преподаватели S_j за всеки един показател по формула (28), средната стойност на групата от преподаватели S_{jav} за всеки показател по формула (29) и отклонението δ_j за степента на съгласуване на мненията на преподавателите по формула (31) и резултатите са дадени в същата таблица. Коефициентът на съгласие по формула (32) се получава:

$$w_k = \frac{12 \sum_{j=1}^m \delta_j^2}{R^2(m^3 - m)} = \frac{12 \cdot 158^2}{13^2(7^3 - 7)} = 5,27$$

Значимостта на w_k се проверява по $\chi_{изч}^2$ по формула (37).

$$\chi_{изч}^2 = R(m - 1)w_k = 13(7 - 1)5,27 = 441,06$$

При степени на свобода $v = m - 1 = 7 - 1 = 6$ и ниво на значимост $\alpha = 0,01$ табличната стойност на $\chi_{табл.}^2 = 16,812$ (Приложение 2)

$$\chi_{изч}^2 = 441,06 > \chi_{табл.}^2 = 16,812$$

Тегловните коефициенти W_j са пресметнати по формула (38), където V_j е коефициентът на вариация, характеризиращ съгласуваността на оценките на преподавателите и се изчислява по формула (33). Резултатите са дадени в **таблица 4.41**. От **фигура 4.9** става ясно, че системата е оценена най-високо по критерии Степен на логическа последователност на учебния материал, Свързаност на компонентите на учебния материал, Наличие на речник на използваните термини в лекционния материал и Наличие на търсеща машина в основните модули. Изчислените тегловни коефициенти, представени графично на **фигура 4.10**, показват, че системата трябва да се оптимизира наблягайки на Връзката с други ресурси за допълнителна информация за изучавания проблем, Използването на различни мултимедийни компоненти и Насърчаването за критично мислене и креативност.

4.4.4 Мнения и коментари

При провеждането на анкетата, някои от участниците изказаха следните мнения и коментари относно качеството и ефективността на оценяваната Web-базирана система. Коментарите са взети от анкетните карти, попълнени от оценяващите системата експерти и преподаватели.

На въпрос относно преносимостта на системата, експерт е дал следния коментар:

„Web базираната система осигурява независимост от клиентската среда, инсталиран софтуер и хардуер.“

На въпроса Осигурява ли системата възможност за въвеждане на учебния материал с логическа последователност?, коментарът е:

„не само това , но на потребителя е предоставена възможност да използва свои собствени логически пътища чрез подходящи вътрешни връзки.“

На въпроса Осигурява ли системата използване на различни мултимедийни компоненти?, преподаватели дават следните коментари:

- „по този начин се постига висока степен на интерактивност. Студентът учи по същия начин, по който се забавлява в интернет.”

- „Чудесно направен видеоматериал с виртуални модели. Много полезна и добре синтезирана информация.”

На въпроса Осигурява ли системата възможност за връзки между отделните структурни единици в различни части?, е даден следния коментар:

„Би било добре, в съдържанието на курса да има и бързи линкове към съответната връзка. Чудесно и подробно описание на съдържанието и лекциите на курса.”

На въпроса Осигурява ли системата възможност за насърчаване за критично мислене и креативност?, коментарът е:

„Отлична база за теоретично тестване на практическите умения и съобразителността на студентите посредством изградената система от демонстрационни и реални клинични случаи.”

Един от преподавателите изказва следното мнение, относно системата като цяло:

„Системата е много удобна за работа и много функционална по отношение на възможности за създаване на учебни модули по различни курсове в рамките на медицинското образование, Особено полезен е фактът, че има възможност за "разиграване" на клинични случаи. Прекрасно средство за самопроверка на знанията и самообучение на студентите!”

4.5 Изводи

Направено е приложение на подхода за количествена оценка на качеството и ефективността на Web-базирана система за обучение по медицински дисциплини. Системата е интегрирана на сървър в Медицински Университет-София. Анкетирани бяха специалисти от четири фирми в областта на софтуерните и информационни технологии и два български университета, занимаващи се с електронно обучение. Броят на участниците в изследването е съответно 15 експерти от областта на софтуерните технологии и 13 експерти в областта на електронното обучение. За проведения експеримент, на всеки един участник в анкетирането му бе предоставен пълен достъп до оценяваната система и анкетни карти, чрез които всеки един от тях даде своята оценка и мнение за системата. В изследването са взели участие общо 28 експерти и преподаватели, от които според заеманата длъжност: разработчик на софтуер - шестима; Front-end Web разработчик - петима; Web разработчик на приложения - четирима; Професори - двама; Доценти - трима; Асистенти - осем. Според опита в годините от работа в областта на софтуерните и информационни технологии деветнадесет са с опит над 5 години, трима са с опит до 5 години, един е с опит до 4

години, един с опит до 3 години, един с опит до 2 години и трима с опит до 1 година. Според източника на аргументация в детайли са разгледали системата тринадесет от участниците, повърхностно са разгледали системата дванадесет от участниците, съвсем повърхностно са разгледали системата трима от участниците.

От отговорите на анкетата става ясно, че според критериите, базирани на стандарта ISO 9126 за качество на софтуерни продукти, системата е оценена най-високо по критерии Използваемост, Функционалност и Преносимост и най-ниско по критерии Безотказност, Обслужваемост и Продуктивност. Системата е удобна, лесна за изучаване и за използване и интерактивна за всички потребители. Тя е гъвкава и отворена, осигурява независимост от клиентската среда, инсталиран софтуер и хардуер.

Резултатът от дефинирането и решаването на оптимизационните задачи показва, че сегашната информационна система може да бъде подобрявана по три от по-важните критерии за оценка - Използваемост, Функционалност и Преносимост. Експертното мнение съдържа потенциал за увеличаването на стойностите на тези критерии.

Според критериите, базирани на дидактическите показатели става ясно, че е оценена най-високо по критерии Степен на логическа последователност на учебния материал, Свързаност на компонентите на учебния материал, Наличие на речник на използваните термини в лекционния материал и Наличие на търсеща машина в основните модули. Това показва, че системата е удобна за работа и функционална по отношение на възможности за създаване на учебни модули по различни курсове. Позволява комбиниране и многократно използване на учебните материали.

От друга страна, изчислените тепловни коефициенти на показателите показват, че системата трябва да се оптимизира наблягайки на Връзката с други ресурси за допълнителна информация за изучавания проблем, Използването на различни мултимедийни компоненти и Насърчаването за критично мислене и креативност.

Като обобщение може да се каже, че системата за Web-базирано обучение е съобразена с основни принципи на препоръчителни международни стандарти като многократна използваемост, управляемост на информацията за потребителите и учебното съдържание, достъпност, дълготрайност – възможност за развитие на системата с развитие на технологиите и възможност за взаимодействие между различни системи.

Системата изпълнява и всички функции, характерни за универсална Web-базирана система за управление на обучението като управление на студентите, управление на учебните материали, управление на материалите за оценяване, управление на учебния процес, управление на достъпа до системата.

Резултати от изследванията по тази глава са публикувани в [1] от публикациите по дисертацията.

Заклучение

За нуждите на дисертацията са анализирани 152 литературни източника, свързани с методи, етапи, техники и средства нужни за изграждане на информационната система за организиране и управление на Web-базирано обучение и са разгледани съществуващите подобни платформи и среди за електронно обучение. Дефинирани са 6 етапа, през които минава подготовката и реализирането на системата. Анализирани и описани са основните компоненти и изисквания към нея. Мотивиран е изборът на технологични средства за реализиране на програмните модули. Проектирана и разработена е система за организиране и управление на Web-базирано обучение. Направено е сравнение на системата с други подобни вече съществуващи софтуерни платформи. Резултатите са анализирани и визуализирани чрез радарни диаграми.

На базата на специализирана литература по този въпрос е приложен подход за количествена оценка на качеството и ефективността на системата, който се основава на метод на експертните оценки. Проведено е експериментално приложение на подхода с участието на експерти и преподаватели от няколко фирми и университети с цел оптимизиране работата на системата и са анализирани получените резултати. В резултат на прилагането на оптимизационния подход, с определените количествени оценки е дефинирана и решена оптимизационна задача за оценяване на потенциала на разработената информационна система и подобряване на характеристиките ѝ. Резултатите от изследването показват, че според критериите, базирани на стандарта ISO 9126 за качество на софтуерни продукти, системата има потенциал за подобряване на характеристиките ѝ по три от по-важните критерии за оценка. Според дидактическите критерии оценката на експертите е положителна по отношение на възможностите за създаване на учебни модули по различни курсове. Като цяло са получени насоки за подобряване на системата по двете направления за оценка на качеството ѝ – като програмен и образователен продукт.

Съставянето на количествена оценка за качеството на софтуерния продукт е важна не само за оценката и за сравнението на отделните програмни приложения, но и за ефективното управление и усъвършенстването на процесите им като цяло. От значение е и правилният избор на модел на жизнения цикъл на софтуерния продукт, който обхваща всички етапи на разработката, внедряването и експлоатацията му.

Тенденциите за развитие на електронното обучение и по-голямата достъпност на образованието, като цяло засилват необходимостта от внедряване на информационните системи на всички нива на образователния процес. Едни от най-често използваните системи в образованието са системите за организиране и управление на съдържанието. Тези системи са многофункционални и непрекъснато се развиват и усложняват. Проблемът как да се структурира Web-базиран курс за обучение и да се представи съобразно избран модел, как да се съхранява в правилно структурирана Web-интегрирана база данни е доста актуален в наши дни. Въпреки универсалния характер на съществуващите системи за електронно обучение, практиката показва, че отделните висши училища използват информационни системи, изцяло съобразени със

спецификата им, което оправдава и налага необходимостта от разработването на собствени системи, съобразно избран педагогически модел.

Реализираната система за организиране и управление на Web-базирано обучение е съобразена с основни принципи на препоръчителни международни стандарти обобщени като многократна използваемост, управляемост на информацията за потребителите и учебното съдържание, достъпност, дълготрайност – възможност за развитие на системата с развитие на технологиите и възможност за взаимодействие между различни системи.

Осигурената прозрачност на учебния процес увеличава възможностите за контрол и коригираща намеса. Системата не измества преподавателите, тя спестява време на всяка група потребители и улеснява взаимодействието между физически отдалечени екипи при управлението на учебния процес.

Чрез системата са осигурени възможности за провеждане на Web-базирано обучение по различни медицински дисциплини. Чрез изцяло браузер-базиран интерфейс преподавателите, без да имат познания по програмиране и дизайн, могат лесно и бързо да създават, редактират, публикуват и регрупираат Web-базирани курсове, съдържащи лекции, интерактивни тестове за оценка на теоретичните познания, интерактивни симулации на клинични случаи за придобиване на практически умения.

Осигурената гъвкавост и динамичност на системата, богатите комуникационни възможности, бързата актуализация на данните, неограничената възможност за въвеждане на нови страници, подстраници, възможността за генериране на разнообразни справки правят системата полезна за студентите, преподавателите и гостите. Създадени са възможности за (само)оценяване и подобряване качеството на обучение на всички нива, повишаване квалификацията на преподавателите и привличане на повече наши и чужди студенти и специализанти.

Курсовете, създадени чрез Web-базираната система за обучение са добър пример за интеграция между най-новите постижения в областта на избрана дисциплина (модерно съдържание), ИКТ (Web среда), педагогика (гъвкаво обучение). Разгледаната система е универсална и се използва и от други дисциплини в Медицински университет - София, като освен Биохимия (<http://biochemistry.mu-sofia.bg>), сайтове развиват Физиология (<http://physiology.mu-sofia.bg>), Физика и биофизика (<http://physics-biophysics.mu-sofia.bg>), Фармакология и токсикология (<http://pharmacology-toxicology.mu-sofia.bg>).

При направен сравнителен анализ на възможностите на разглежданата система за Web-базирано обучение с тези на известните софтуерни платформи се вижда, че тя е съизмерима по основните си качества на най-добрите решения в тази област и притежава необходимите средства за ефективно провеждане на електронно обучение. Удовлетворява до голяма степен посочените критерии и има голям потенциал за бъдещо развитие.

Приноси на дисертационния труд

Основните резултати имат научноприложен и приложен характер.

1. Анализирани са стандарти и модели за оценка на качество на софтуерните продукти.
2. Проектирана и реализирана е системата за Web-базирано обучение по медицински дисциплини. Разработени са основните компоненти на системата и са синтезирани специфични функции „Виртуален пациент” и „Виртуални модели”. Проектирани са необходимите база от данни и информационни процеси за програмните модули. Оценена е функционалността на разработената информационна система като е сравнена с известни софтуерни платформи. Определени са предимствата на системата за изпълнение на специфични функции за медицинското обучение.
3. Разработена е процедура за количествена оценка на характеристиките на информационната система. Приложен е метод на експертните оценки. Направен е избор на показателите за качество на системата в две направления: 1) по утвърден стандарт за оценяване на програмни продукти - ISO 9126; 2) по дидактически показатели, базирани на мнения на експерти от образованието.
4. Количествено са оценени характеристики на процеси в информационната система. Оценката е направена с участието на експерти от няколко фирми от областта на информационните технологии и преподаватели в университети.
5. С определените количествени оценки е дефинирана и решена оптимизационна задача за оценяване на потенциала на разработената информационна система и подобряване на характеристиките ѝ. Резултатите от изследването показват, че според критериите, базирани на стандарта ISO 9126 за качество на софтуерни продукти, системата има потенциал за подобряване на характеристиките ѝ по критерии Използваемост, Функционалност и Преносимост.

Според дидактическите критерии оценката на експертите е положителна по отношение на възможностите за създаване на учебни модули по различни курсове.

Бъдещи насоки за работа

На основание получените резултати от приложението на подхода за количествена оценка на качеството и ефективността, системата за Web-базирано обучение ще се развива в няколко насоки:

Спрямо критериите, базирани на стандарта ISO 9126 за качество на софтуерни продукти:

- Повишаване надеждността на системата в случай на софтуерни грешки и способността за възстановяване на данни.
- Повишаване обслужваемостта на системата, т.е способността на системата да бъде модифицирана. Модификациите включват корекции, подобрения или адаптиране на системата към промени в средата (включва се промяната на кода, дизайна и данните), както и в изискванията и функционалните спецификации.
- Повишаване на продуктивността на системата като първо се намалят времената за реагиране и протичане, както и скоростите на изпълнение функциите на системата и второ като се намали продължителността на използване на ресурсите при изпълнение функциите на системата.

Спрямо критериите, базирани на дидактическите показатели:

- Подобряване на връзката с други ресурси за допълнителна информация за изучаван проблем;
- Подобряване използването на различни мултимедийни компоненти (включване на инструменти за дискусии в реално време, интегриране на форум за по-динамични връзки между студентите и между тях и преподавателите);
- Насърчаване на критично мислене и креативност на студентите чрез добавяне на нови приложения и методи на обучение за придобиване на дълготрайни знания и умения.

Публикации, свързани с дисертацията

1. Trichkova, E., K. Stoilova, “*An Approach for Quality Assessment and Effectiveness of a Web-Based System for Distance Learning*”, Cybernetics and Information Technologies, Volume 13, No 4, Sofia, 2013, BAS, pp.63-73.
2. Тричкова, Е., „*Web-базирана система за електронно обучение по медицински дисциплини – административен модул*”, Първа национална тематична школа и борса за научни идеи в областта на информационните и комуникационни технологии, Русе 2013, 27-28 Юни 2013, стр. 107-112, ISSN 1314-9024.
3. Trichkova, E., “*Flexible web-based e-learning system*”, International Conference “Automatics and Informatics’12”, 3-5 October 2012, Sofia, Bulgaria, pp.361-364, ISSN 1313-1869.
4. Trichkova, E., “*PHP based site-template for distance learning in medical disciplines*”, Proceedings of E-Learning’11 conference - E-Learning and the Knowledge Society, 25-26 August 2011, Bucharest, Romania, pp. 87-92, ISBN 978-606-505-459-2.
5. Тричкова, Е., G. Kossekova, “*Interactive biochemistry – Sofia*” (ibs) – a flexible web-based e-learning system”, Proceedings of IADIS Multi Conference on Computer Science and Information Systems 2011, E-learning, 20 - 26 July 2011, Rome, Italy, pp. 3-10, ISBN 978-989-8533-00-5.
6. Тричкова, Е., *Моделиране, изпълнение и управление на бизнес процеси базирано на Workflow технологии*”, списание Автоматика и Информатика, бр.3/2009, стр. 16-19. ISSN 0861-7562.
7. Trichkova, E., E. Ivanova, K. Trichkov. “*Workflow Orchestration Software*”. International Conference ETAI, Ohrid, FYROM, 12-21 September, 2007 p. I4-5:1- I4-5:5. Proceedings on CD, ISBN 978998921753X.

Научноизследователски проекти

Изследванията в дисертационния труд са част от получените резултати на следните научноизследователски проекти:

1. Международен проект по 6-та Рамкова програма на Европейската комисия, Информационно общество. Project EC FP6, program IST No. 027178 “Virtual Internet Service Provider VISP”
2. Договор за отпускане на финансова подкрепа за проект за мобилност по секторна програма „Леонардо да Винчи”. Програма „Учене през целия живот”. 2008-1-BG1-LEO03-00374. Тема „Обмяна на опит за иновационни приложения на решения за електронно обучение”.
3. Международен проект по програма „Обучение през целия живот” - ERASMUS. Договор с Европейската комисия: 142399-LLP-1-2008-1-BG-ERASMUS-ENW на тема: „ETN Teaching, Research, Innovation in Computing Education”.
4. The thematic Network Teaching, Research and Innovation in Computing Education - TRICE is established in accordance with the ERASMUS Programme of the EUROPEAN COMMISSION.
5. Европейски Социален Фонд 2007-2013 Г. Програма "Развитие на Човешките Ресурси" . Подкрепа на творческото развитие на докторанти, пост-докторанти и млади учени в областта на компютърните науки, BG 051PO001-3.3.04/13.

Декларация за оригиналност на резултатите

Декларирам, че настоящата дисертация съдържа оригинални резултати, получени при проведени от мен научни изследвания, с подкрепата и съдействието на научния ми ръководител. Резултатите, които са получени, описани и/или публикувани от други учени, са надлежно и подробно цитирани в библиографията.

Настоящата дисертация не е прилагана за придобиване на научна степен в друго висше училище, университет или научен институт.

Е. Тричкова:

Литература

1. Abrahamsson P., Salo O., Ronkainen J., Warsta J. (2002), Agile Software Development Methods, Technical report, VTT Publications 478, VTT, Finland.
2. Agile (2010), Manifesto for Agile Software Development, <http://agilemanifesto.org> , Last access on march 2010.
3. Basili V. R., Turner A. J (1975), Iterative Enhancement: A Practical Technique for Software Development, IEEE Transactions on Software Engineering, SE-1-4, Dec.
4. Benington H. (1983), Production of Large Computer Programs. In Proceedings ONR Symposium (1956), reprinted in Annals of the History of Computing 5(4), pp 350-361.
5. Berander P. , Software quality attributes and trade-offs, Editors: Lars Lundberg, Michael Mattsson, Claes Wohlin, Blekinge Institute of Technology , June 2005.
6. Boar, Bernard H., Implementing Client/Server Computing: a strategic perspective, McGraw-Hill, 1993, p. 1947, <http://www.faqs.org/faqs/client-server-faq/>
7. Boehm B. W. (1988), A spiral model of software development and enhancement. IEEE Computer, 21 (5): 61-72.
8. Boehm B., Turner R. (2003), Balancing Agility and Discipline. Addison-Wesley.
9. Bowen T.P., G.B. Wigle, J.T. Tsai, Specification of Software Quality Attributes, Software Quality Evaluation Guidebook, RADC-TR-85-37, vol. III, 1985.
10. BPMN, 2006. Business Process Modeling Notation (BPMN) Specification, OMG Document dtc/06-02-01, Final Adopted Specification.
11. BPMN, 2008. Business Process Modeling Notation (BPMN) Specification 2.0, OMG Document bmi/2008-02-06, Initial submission.
12. Chen P. (1976), The Entry-Relationship Model-Toward a Unified View of Data, In ACM Transactions on Database Systems 1/1/1976 ACM-Press ISSN 0362-5915, pp.9-36.
13. Chua, B.B., L. E. Dyson. Applying the ISO 9126 model to the evaluation of an elearning system. Proceedings of the 21st ASCILITE Conference (pp. 184-190). Perth, 5-8 December 2004. <http://www.ascilite.org.au/conferences/perth04/procs/chua.html>
14. Davis A., Sitaram P. (1994), A Concurrent Process Model for Software Development, Software Engineering Notes, ACM Press, Vol. 19, No. 2, pp.38-51, Apr. 1994.
15. Deb, K. An ideal evolutionary multi-objective optimization procedure. IPSJ Transactions on Mathematical Modeling and Its Applications, 45(SIG 2), p. 1-11, 2004.
16. Deb, K. Genetic algorithms for optimization. In D. Kundu and A. Basu (eds.) Statistical Computing: Existing Methods and Recent Developments. New Delhi, India: Narosa Publishing House, p. 85-123, 2004.
17. Deb, K. Multi-objective optimization using evolutionary algorithms, John Wiley & Sons, Ltd, New York, 2003.
18. Dietinger, T. (2003). Aspects of E-Learning Environments. Ph.D. Dissertation, Graz University of Technology.
19. Eckert, K-P., Glickman, Y., Hall, J., et al., 2006. "Workflow Technologies for a Virtual ISP". In Exploiting the Knowledge Economy: Issues, Applications, Case Studies, P. Cunningham and M.Cunningham, ed. IOS Press, Amsterdam, pp.1631-1638.
20. Entwicklungsstandard für IT Systeme des Bundes, Vorgehens-model, Teil 1: Regelungsteil, <http://www.V-modell.iabg.de>.

21. Eskenasi A., Evaluation of Software Quality by Means of Classification Methods, J. of Systems and Software, vol. 10, No 3, October 1989, pp. 213-216.
22. Fisher R. A., Statistical Methods for Research Workers, Oliver and Boyd, Edinburgh, 1934.
23. Gordon V., Bieman J. (1994), Rapid Prototyping: Lessons Learned. IEEE Software, 12(1). pp. 85-95.
24. Hall J., Klaus-Peter Eckert, Business and Technical Workflows for E-Business in a Virtual Cluster of ISPs ICE-B 2008, International Conference on e-Business, Porto, Portugal 26-29 July 2008.
25. Harington, E. C., The desirability function. Ind. Quality Control, No 4, p. 494-498, 1965.
26. Hosier W. (1961), Pitfalls and Safeguards in Real-Time Digital Systems with Emphasis on Programming. IRE Transactions on Engineering Management, pp. 99-115.
27. Humphrey, W. S., A Discipline for Software Engineering, Reading, MA: Addison-Wesley 2002.
28. ISO, International Organization for Standardization, "ISO 9126-1:2001, Software engineering – Product quality, Part 1: Quality model", 2001.
29. IBM Corporation (1974).HIPO—A Design Aid and Documentation Technique, Publication Number GC20-1851, IBM Corporation, White Plains, NY, 1974.
30. Information System - The Encyclopedia Britannica, <http://www.britannica.com/EBchecked/topic/287895/information-system>
31. Ivanova, E., Orchestrating Web Services - Standards and Solutions, National Scientific Conference "Mathematics, Informatics and Computer Sciences"- St. Cyril and St. Methodius University of Veliko Tarnovo, 12-13 May, 2006, Veliko Tarnovo, Bulgaria, 137-142.
32. Ivanova, E., T.Stoilov. Workflow technologies in e-learning. Proceedings of “E-learning conference’06: Computer science education”. Coimbra, Portugal, 7-8 September, 2006, ISBN 978-989-20-0350-4, p.1.7-1, 1.7-4.
33. Kan H. S., Metrics and Models in Software Quality Engineering, 2nd Edition, Addison-Wesley Professional., 2002.
34. Katzan, Harry, Jr. (1976). Systems Design and Documentation: An introduction to the HIPO Method. Van Nostrand Reinhold. ISBN 0-442-24267-0.
35. Kendall M. G., Rank Correlation Methods, London, Griffin, 1957.
36. Kendall M., J. D. Gibbons. Rank Correlation Methods, Oxford University Press, New York, 1990.
37. Khaddaj S., Gerard Horgan, A Proposed Adaptable Quality Model for Software Quality Assurance, Journal of Computer Sciences 1 (4): 482-487, 2005.
38. Kossekova, G.: Course Generator – On-line Web-based system with dynamic database for problem-solving oriented distance learning in Biochemistry. In: Proceedings of the International Conference "Computer Systems and Technologies" (CompSysTech), pp. IV. 14, 1--6, ACMBUL, Sofia (2002).
39. Kossekova, G.: Course Generator – On-line Web-based system for problem-solving oriented distance learning in Biochemistry. 2. Administrative module. In: Proceedings of the International Conference” Computer Systems and Technologies” (CompSysTech), pp. IV. 15, 1--6, ACMBUL, Sofia (2002).

40. Ladd Eric, O'Donnell Jim, "Using HTML 4, XML and JAVA 1.2", 1999.
41. Lambertus, Johannes Hommes, Bart-Jan Hommes (2004). The Evaluation of Business Process Modeling Techniques. Doctoral thesis. Technische Universiteit Delft.
42. Marshak R. T., Perspectives on Workflow, In T. E. White, L. Fischer (ed.): New Tools for New Times - The Workflow Paradigm, 1994. Future Strategies, Inc., Alameda, CA, USA, pp 165-176.
43. Marsic, I., Software Engineering, Rutgers University 2008.
44. Martin J. (1991), Rapid Application Development. MacMillan.
45. Martinez, W. L. (2011), Graphical user interfaces. WIREs Comp Stat, 3: 119–133. doi: 10.1002/wics.150.
46. Nenova, S.: An Universal Web-Based Learning Management System. In: Proceedings of the International Conference on Computer Systems and Technologies (CompSysTech), Section E-Learning, pp. IV.23, 1---6, ACMBUL, Sofia (2002).
47. OMG Unified Modelling Language Specification, available at: <http://www.omg.org>
48. Pareto, V., *Cours d'economie politique*, Lausanne, Rouge, 1896.
49. Paunova, E. Approach for Creating Educational Games as Online Information Services. International Doctoral Symposium, 27-28 June 2011, Bucharest, Romania, ISSN 2247-6040, pp 22-26.
50. Pressman, R., Software Engineering: A Practitioner's Approach, 2004.
51. Ragunath P. K., Velmourougan S., Davachelvan P., Kayalvizhi S., Ravimohan R. (2010), A New Model (SDLC Model-2010) for Software Development Life Cycle (SDLC), IJCSNS International Journal of Computer Science and Network Security, Vol.10 No.1, pp.112-119, January.
52. Rochkind, M., Expert PHP and MySQL - Application Design and Development, Apress, 2013, ISBN: 978-1-4302-6007-3.
53. Royce W.W. (1987), Managing the development of large software systems, Proceedings of 9th IEEE International Conference on Software Engineering, pp: 328-338.
54. Ryan K. L. Ko, Stephen S. G. Lee, Eng Wah Lee (2009) Business Process Management (BPM) Standards: A Survey. In: Business Process Management Journal, Emerald Group Publishing Limited. Volume 15 Issue 5. ISSN 1463-7154.
55. Sametinger Johannes, Software Engineering with Reusable Components, Springer 2001.
56. Scollo, Ch., Jesus Castagnetto, Harish Rawat, Deepak Veliath, Sascha Schumann., Professional PHP Programming, Wrox, Desember 1999.
57. Smrikarov, A., 2009. National Program for creating a virtual learning environment. Main results and tasks ahead. Youth participation and contribution, Electronic Bulgaria (2009) (in Bulgarian). "Avangard Print" OOD, Sofia, Bulgaria.
58. Stoilov T., K. Stoilova, N. Lyutov. Workflow technology as a tool for automation of business systems. Proceedings of the Fourth International Bulgarian-Greek Conference "Computer Science'2008", September 2008, Kavala, Greece, vol.2., ISBN: 978-954-580-255-3, p.668-674.
59. Stoilov T., K. Stoilova, Zl. Ivanova. Application of Investment optimization as Web service. International conference Automatics and Informatics'10, Sofia, Bulgaria, 2010, ISSN 1313-1850, pp. II-407-410.

60. Stoilov, T., K. Stoilova, Computer Automation of Business Processes, National Scientific Conference "Mathematics, Informatics and Computer Sciences"- St. Cyril and St. Methodius University of Veliko Tarnovo, Veliko Tarnovo, Bulgaria, 12-13 May, 2006, 167-172.
61. Stoilov T., K. Stoilova, N. Lyutov. Technologies for integration of e-learning content. Proceedings: E-Learning Solutions – on the Way to Ubiquitous Applications, Eds. D.Dochev, R.Pavlov. Joint KNOSOS-CHIRON Open Workshop, Leonardo da Vinci Programme, Sandanski, Bulgaria, 26-27 May, 2005, p.51-58.
62. Stoilov T., K. Stoilova. *Integration of Web Services in Internet*. 18th International Conference on Systems for Automation of Engineering and Research "SAER-2004", 24-26 September, 2004, St. Konstantin resort, Varna, Bulgaria, ISBN 954-438-428-6, p.15-23.
63. Stoilov T., K.Stoilova. Virtual learning platforms and good practice. Comenius 3 Network EcoMedia – Europe International workshop “Virtual learning and school education" June 8-10, 2006, Sofia, Bulgaria, p.4-27.
64. Stoilova. K., T.Stoilov, G.Spasov. Technological Stack for development of automated info services in business information systems. Journal of the Technical University Sofia, Plovdiv branch, Bulgaria. “Fundamental sciences and applications”, vol.14 (1), 2009,. Proceedings of the Int.Conf. “ Engineering, Technologies and Systems” TECHSYS’2009, ISSN 1310-8271, p 341-346.
65. Stoilova K., Z.Ivanova, T.Stoilov. E-intelligence in Portfolio Investment Optimization. In *E-Service Intelligence: Methodologies, Technologies and Applications*, J.Lu, D.Ruan, G.Zhang (Eds). Springer – Verlag, Berlin, Heidelberg, 2007, ISBN-10: 3-540-37015-3, ISBN-13: 978-3-540-37015-4, p.457-476.
66. Stoilova K., T.Stoilov, Z.Ivanova. Automation of Information Processes. Proceeding of International Conference on e-Learning and the Knowledge Society-e-Learning’10, 26-27 August, 2010, Riga Technical University, JUMI, Riga, Latvia, ISBN 978-9984-30-181-5, pp.154-159.
67. Stoilova, K., Todor Stoilov, Comparison of Workflow Software Products, International Conference on Computer Systems and Technologies - CompSysTech'2006, 15-16 June, 2006, Veliko Tarnovo, Bulgaria, IIIA.21-1 – IIIA.21-6.
68. Stoilova, K., T.Stoilov. Technologies for integration of e-learning content. Proceedings of “E-learning conference’06: Computer science education”. Coimbra, Portugal, 7-8 September, 2006, ISBN 978-989-20-0350-4, pp.2.11-1, 2.11-6.
69. Stoyanov S., E. Doychev, V. Valkanova, G. Cholakov, Education Cluster for Intelligent Provision of eLearning Services, DBKDA 2012: The Fourth International Conference on Advances in Databases, Knowledge, and Data Applications, 29 February – 5 March 2012, Reunion, IARIA, 2012. ISBN: 978-1-61208-185-4, pp. 45-50.
70. Stoyanov S., V. Valkanova, G. Cholakov, M. Sandalski, Education Portal for Reactive and Proactive Service Provision, COGNITIVE 2011: The Third International Conference on Advanced Cognitive Technologies and Applications, 25-30 September, 2011, Rome, pp. 99-103, ISBN: 978-1-61208-155-7.
71. Stoyanov S., G. Cholakov, V. Valkanova, M. Sandalski, Personalized, Reactive and Proactive Providing of e-Learning Services, EdiLib Conference, AACE E-Learn 2011 –

- World Conference on E-Learning in Corporate, Government, Healthcare & Higher Education, Honolulu, Hawaii, USA, 17-21 October, 2011, pp. 2527-2534.
72. Stoyanov, S. et al., "Intelligent Distributed eLearning Architecture", in *Intelligent Systems*, V. M. Koleshko, Ed.: InTech, 2012, pp. 185-218.
 73. Totkov G., T. Uzunov, R. Doneva, N. Kasakliev, E-Learning Standartization – Bulgaria: Current Status, Issues and Possible Solution, *ACM International Conference Proceeding Series*; Vol. 374, Proc. of the 9th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing 2008, Gabrovo, Bulgaria, June 12-13, 2008, IV.2-1-IV.2-6.
 74. Trichkova, E., Krasimir Trichkov. *Web-based Information System for E-learning*, International Conference on e-Learning and the Knowledge Society - e-Learning'09, 31.08.09 – 01.09.09, Берлин, Германия, p.115-120. ISBN 1313-9207.
 75. VISP project deliverables: D2.2, (2009), <http://www.visp-project.org/pages/deliverables.jsp>
 76. VISP project deliverables: D2.3, (2009), <http://www.visp-project.org/pages/deliverables.jsp>
 77. Vliet H. (2007), *Software engineering: Principles and Practice*, Wiley.
 78. Voutchkov, I. I., A. J. Keane. Multiobjective optimization using surrogates. In *Proc. 7th Int. Conf. Adaptive Computing in Design and Manufacture*, Bristol, p. 167-175, 2006.
 79. Wainwright, P., *Professional Apache*, Wrox, 1999.
 80. Wehling J., W. Haupt. Multimedia in Technology Teacher Education. Learning In Technology Education Challenges For The 21st Century, *Proceedings of the 2nd Biennial International Conference on Technology Education Research held at the Parkroyal Gold Coast*, Australia, 5 – 7 December 2002. Vol. 2, pp. 254-259, 2002.
 81. Weistroffer, H. R., A combined over- and under- achievement programming approach to multiple objective decision making, *Large Scale Systems*, 7, p. 47-58, 1984.
 82. WfMC, Glossary, Terminology and Glossary, 3rd Edition. Document No WfMC-TC-1011. Workflow Management Coalition. Winchester, 1999.
 83. WfMC, Glossary, Terminology and Glossary, 3rd Edition. Document No WfMC-TC-1011. Workflow Management Coalition. Winchester, 1999.
 84. Wierzbicki, A. P., The use of reference objectives in multiobjective optimization - Theoretical implications and practical experiences, WP-79-66, IIASA, Laxenberg, Austria, 1979.
 85. Wierzbicki, A. P., A methodological guide to multiobjective optimization, WP-79-122, IIASA, Laxenburg, Austria, 1979.
 86. W3C, *Web Services Glossary*, W3C Working Group Note, 11 February 2004, available at <http://www.w3.org/TR/ws-gloss/>.
 87. W3C, *Web Services Glossary*, W3C Working Group Note, 11 February 2004, available at <http://www.w3.org/TR/ws-gloss/>.
 88. Антонова И., И. Бачкова (2005), Сравнителен анализ на използването на UML в приложения за реално време, Материали на Научна сесия “Развитие, бъдеще, перспективи и иновации в науката 2005”, 22.06.05, София, pp.103-108.
 89. Атанасова, Т., Компютърни информационни системи, ИК «Кръг», Варна, 1999 г.

90. Бачкова, И., “Интегрирани системи за моделиране и управление”, http://www.iit.bas.bg/esf40/phd-lecture-01-10-10/I_Bachkova/I_Batchkova_txt.pdf
91. Вардева И., Л. Станева, Електронно обучение-решения, проблеми перспективи. Годишник на университет „проф. д-р Асен Златаров”-Бургас 2007, т. XXXVI(2).
92. Влахова Р., Г. Дачева. E-learning, Web базирано обучение и образователни портали. II Национална конференция с международно участие по електронно обучение във Висшето образование, Китен 2006, ст. 84-86.
93. Вучков, И., С. Стоянов, Компромисни решения при оптимизация на сложни технологични процеси, Целулоза и хартия, No 5, 1970.
94. Грийнспан, Дж., Брад Бългър, MySQL/PHP приложения за бази данни, АлексСофт, 2001.
95. Джамбазов, В., Web базирани потребителски интерфейси, Нов български университет, 2011, ISBN 9789545356650.
96. Ернандес, М., „Проектиране на бази от данни“, Софтпрес, 2004, ISBN 954-685-301-1.
97. Ескенази А., Н. Манева, Софтуерни технологии. ЕТ „КЛМН”, София, 2006.
98. Иванов. М., Принципи и перспективи за оценка на качеството на софтуерните продукти, XX-ти национален научен симпозиум с международно участие "Метрология и метрологично осигуряване 2010", 9-13 септември, 2010, Созопол, България.
99. Иванов М., И.Момчев, Принципи и проблеми на многокритериалната оценка на качеството на софтуерния продукт, сп.”Автоматика и информатика”, 2006.
100. Иванов М., И.Момчев, Принципи и проблеми на многокритериалната оценка на качеството на софтуерния продукт, сп.”Автоматика и информатика”, 2006.
101. Иванова Зл., Кр.Стоилова, Т.Стоилов. *Портфейлна оптимизация – информационна услуга в Интернет*. Академично издателство “Марин Дринов”, София, 2005, ISBN 954-322-021-2, стр.275.
102. Касъкчиев, Николай. Сравнителен анализ на информационни системи за висши училища. Сборник доклади на Национална конференция "Образованието в информационното общество" Пловдив, АРИО, 26-27 май 2011, стр. 139-148 ISSN 1314-0752.
103. Колисниченко, Д., PHP & MySQL - практическо програмиране, Асеновци, 2011, ISBN 9789548898218.
104. Къртис, Г., Бизнес-информационни системи - Анализ, проектиране и практика, Издателска къща „Иван Вазов”, София, 1995.
105. Нонова, А. Оптимизация на технологични обекти по векторен критерий за оптималност. Дис. С., 1977.
106. Паунова, Е.: Проектиране и разработване на електронни игри за ученици като web-базирана услуга, дисертация за придобиване на образователната и научна степен “доктор” по специалност 02.21.10. „Приложение на принципите и методите на кибернетиката в различни области на науката” в професионално направление 5.3. „Компютърна и комуникационна техника” (2013).
107. Пенева, Ю., Бази от данни. Първа част и втора част, Регалия - 6, 2004, ISBN: 9547450786.

108. Стоянов, Ст., А.Стоянова-Дойчева, М.Трендафилова, Е.Дойчев. Софтуерни технологии, Университетско издателство „Паисий Хилендарски”, 2006.
109. Стоянов, С., Оптимизация на технологични процеси, Техника, София, 1993.
110. Стоянов, С., Методи и алгоритми за оптимизация, Техника, София, 1990.
111. Стоянов, С., Е. Удовлетвореност на клиентите и вземане на решения в здравните организации, ЕС принт, София, 2008, ISBN 97895491913-9-4.
112. Стоянов, С. И.Божов. Оптимизация на многоцелеви технологически процеси по симплекс метод. – Целулоза и хартия, 1972, N:1.
113. Стоянов, С. Върху задачата многоцелева оптимизация на технологични обекти. Дис. С., 1975.
114. Томс, Ж., В. Джамбазов, Основи на Web дизайна, Сиела, 2004, ISBN: 9546496944.
115. Тужаров, Х., Методология за анализ и проектиране на информационни системи, Издателство „Асеновци”, 2008.
116. Тужаров, Х., Информатика, Издателство „Асеновци”, 2007.
<http://tuj.asenevtsi.com/Informatica2/I008.htm>
117. Тужаров, Х., Анализ и проектиране на информационни системи, издателства "Асеновци" 2008.
118. Тужаров, Хр. Базис Данни, издателство Асеновци, 2007,
<http://tuj.asenevtsi.com/DB2007/BD22.htm>
119. Тужаров, Хр. Базис Данни, издателство Асеновци, 2007,
<http://tuj.asenevtsi.com/DB2007/BD21.htm>
120. Управление на бизнес процеси, СЮ 2008, Ай Си Ти медия.
http://cio.bg/download_wp?id=7
121. Христов, Ц.: Създаване и изследване на софтуерна платформа за електронно обучение, дисертация за присъждане на образователната и научна степен “доктор” по научната специалност 02.21.09 "Автоматизация на области от нематериалната сфера", Специализиран научен съвет по „Електронна и компютърна техника” (2009).
122. Цонев, С., В. Витлиев, П. Коев. Методи за оптимизация.(Библиотека за докторанта), 2. изд., Русенски университет, Русе, 2004, ISBN 954-712-229-0.
123. Шойкова, Е., М. Иванова. Сравнителен анализ на системи и авторски инструменти за електронно обучение с отворен код. *Автоматика и информатика*, Vol. XLI (4). pp. 55-61, 2007.
124. <http://iso9126may.blogspot.com/>
125. <http://www.khambatti.com/mujtaba/ArticlesAndPapers/Cleanroom%20Software%20Development.pdf>.
126. <http://www.sei.cmu.edu/pub/documents/96.reports/pdf/tr022.96.pdf>.
127. <http://weblog.erenkrantz.com/~jerenk/phase-ii/Boe88.pdf>
128. SEVOCAB: Software and Systems Engineering Vocabulary. Term: *Flow chart*. Retrieved 31 July 2008.
129. <http://www.sqa.net/iso9126.html>
130. <http://www.arisa.se/compendium/node6.html>
131. <http://www.oracle.com/technetwork/developer-tools/datamodeler/overview/index.html>

132. <http://www.ideal.com/IDEF1.htm>
133. <http://www.ideal.com/IDEF1x.htm>
134. <http://www.wfmc.org/>
135. <http://php.net>
136. <http://mysql.com>
137. <http://www.w3.org>
138. <http://httpd.apache.org/>
139. <http://www.iis.net/>
140. <http://nginx.org/>
141. <http://www.lighttpd.net/>
142. <http://news.netcraft.com/archives/category/web-server-survey/>
143. <http://fbm.uni-ruse.bg/d/uis/L-2.pdf>
144. <http://www.microsoft.com/education/en-us/Pages/index.aspx>
145. http://www.ibm.com/developerworks/lotus/library/ls-LMS_preview/
146. www.webct.com
147. www.blackboard.com
148. <http://www.moodle.org>
149. <http://www.atutor.ca>
150. <http://www.claroline.net>
151. <http://www.dokeos.com/>
152. Microsoft partners in learning, Webдизайн. Основни принципи, Просвета, 2012, ISBN 9789540126302.

Използвани съкращения

ASD - Adaptive Software Development
ASP - Application service providers
ASP - Active Server Page
BPEL4WS - Business Process Execution Language for Web Services
BPM - Business process management
BPMN - Business Process Model and Notation
CAF - Consolidated Agile Framework
CASE - Computer-Aided Software Engineering
CGI – Common Gateway Interface
CMS - Content Management Systems
Crystal - Crystal Clear
CSS - Cascading Style Sheets
DeLC - Distributed eLearning Center
DSDM - Dynamic Software Development Method
eLSe - e-Learning Shell
ER - Entity-Relationship
EssUP - Essential Unified Process
FDD - Feature Driven Development
GPL - General Public License
GUI - Graphical User Interface
HTML - Hypertext Markup Language
HTTP - Hypertext Transfer Protocol
IE - Information Engineering
IIS - Internet Information Server
JSP - JavaServer Pages
JWS - Java Web Server
LCMS - Learning Content Management Systems
LDAP - Lightweight Directory Access Protocol
LMS - Learning Management Systems
Moodle - Modular Object-Oriented Dynamic Learning Environment
MySQL - Система за управление на бази данни
ODBC - Open DataBase Connector
OS - Операционна система
PHP - Personal Home Page
PWS - Personal Web Server
RSS - Really Simple Syndication
SCORM - Sharable Content Object Reference Model
SQL - Structured Query Language
UML - Unified Modeling Language
XML - eXtended Markup Language
XP - Extreme Programming
БД – бази данни
БП - Бизнес процес
ВУ - Висши училища
ПП - Програмен продукт
РБД - Релационна база данни
СКС - Симулация на клиничен случай
СТ - Софтуерна технология
СУБД - Система за управление на бази данни

Приложение 1: Разпределение Z при ниво на значимост α

Разпределение Z при ниво на значимост $\alpha=0.05$ [22]

Стой- ности на ν_2	Стойности на ν_1									
	1	2	3	4	5	6	8	12	24	∞
1	2,54	2,65	2,69	2,71	2,72	2,73	2,74	2,75	2,76	2,77
2	1,46	1,47	1,48	1,48	1,48	1,48	1,48	1,48	1,48	1,48
3	1,16	1,13	1,11	1,10	1,10	1,10	1,09	1,08	1,08	1,07
4	1,02	0,97	0,94	0,93	0,92	0,91	0,90	0,89	0,88	0,86
5	0,94	0,88	0,84	0,82	0,81	0,80	0,79	0,77	0,75	0,74
6	0,89	0,82	0,78	0,76	0,74	0,73	0,71	0,69	0,67	0,65
7	0,86	0,78	0,73	0,71	0,69	0,68	0,66	0,64	0,61	0,59
8	0,84	0,75	0,70	0,67	0,65	0,64	0,62	0,59	0,57	0,54
9	0,82	0,72	0,68	0,64	0,62	0,61	0,59	0,56	0,53	0,50
10	0,80	0,71	0,66	0,62	0,60	0,58	0,56	0,53	0,50	0,47
11	0,79	0,69	0,64	0,60	0,58	0,56	0,54	0,51	0,48	0,44
12	0,78	0,68	0,62	0,59	0,57	0,55	0,52	0,50	0,46	0,42
13	0,77	0,67	0,61	0,58	0,55	0,54	0,51	0,48	0,44	0,40
14	0,76	0,66	0,60	0,57	0,54	0,52	0,50	0,46	0,43	0,38
15	0,76	0,65	0,60	0,56	0,53	0,51	0,48	0,45	0,41	0,36
16	0,75	0,64	0,59	0,55	0,52	0,50	0,48	0,44	0,40	0,35
17	0,75	0,64	0,58	0,54	0,52	0,50	0,47	0,43	0,39	0,34
18	0,74	0,63	0,58	0,54	0,51	0,49	0,46	0,42	0,38	0,32
19	0,74	0,63	0,57	0,53	0,50	0,48	0,45	0,42	0,37	0,32
20	0,74	0,62	0,56	0,53	0,50	0,48	0,45	0,41	0,37	0,30
21	0,73	0,62	0,56	0,52	0,49	0,47	0,44	0,40	0,36	0,30
22	0,73	0,62	0,56	0,52	0,49	0,47	0,44	0,40	0,35	0,29
23	0,73	0,62	0,55	0,51	0,48	0,46	0,43	0,40	0,35	0,28
24	0,72	0,61	0,55	0,51	0,48	0,46	0,43	0,39	0,34	0,27
25	0,72	0,61	0,55	0,51	0,48	0,46	0,42	0,39	0,34	0,27
26	0,72	0,61	0,54	0,50	0,48	0,45	0,42	0,38	0,33	0,26
27	0,72	0,60	0,54	0,50	0,47	0,45	0,42	0,38	0,33	0,26
28	0,72	0,60	0,54	0,50	0,47	0,45	0,41	0,38	0,32	0,25
29	0,72	0,60	0,54	0,50	0,47	0,44	0,41	0,37	0,32	0,25
30	0,71	0,60	0,53	0,49	0,46	0,44	0,41	0,37	0,32	0,24
60	0,69	0,57	0,51	0,46	0,43	0,41	0,37	0,32	0,26	0,16
∞	0,67	0,55	0,48	0,43	0,40	0,37	0,33	0,28	0,21	0

Приложение 2: Стойности на χ^2 при брой на степени на свобода ν и ниво на значимост α

Стойности на χ^2 при брой на степени на свобода ν и ниво на значимост α [22]

ν	$\alpha = 0.20$	$\alpha = 0.10$	$\alpha = 0.05$	$\alpha = 0.01$
1	1.642	2.706	3.841	6.635
2	3.219	4.605	5.991	9.210
3	4.642	6.251	7.815	11.345
4	5.989	7.779	9.488	13.277
5	7.289	9.236	11.070	15.086
6	8.558	10.645	12.592	16.812
7	9.803	12.017	14.067	18.475
8	11.030	13.362	15.507	20.090
9	12.242	14.684	16.919	21.666
10	13.442	15.987	18.307	23.209
11	14.631	17.275	19.675	24.725
12	15.821	18.549	21.026	26.217
13	16.983	19.812	22.362	27.688
14	18.151	21.064	23.685	29.141
15	19.311	22.307	24.996	30.578
16	20.465	23.542	26.296	32.000
17	21.615	24.769	27.587	33.409
18	22.760	25.989	28.869	34.805
19	23.900	27.204	30.144	36.191
20	25.038	28.412	31.410	37.566
21	26.171	29.615	32.671	38.932
22	27.301	30.813	33.924	40.289
23	28.429	32.007	35.172	41.638
24	29.553	33.196	36.415	42.980
25	30.675	34.382	37.652	44.314
26	31.795	35.563	38.885	45.642
27	32.912	36.741	40.113	46.963
28	34.027	37.916	41.337	48.278
29	35.139	39.087	42.557	49.588
30	36.250	40.256	43.773	50.892

Приложение 3: Анкетна карта (ISO 9126)

Анкетна карта, съставена на база показатели включени в стандарт ISO 9126 за качество на софтуерни продукти.

Анкетна карта за количествена оценка на качеството на система за уеб-базирано обучение

Заемана длъжност	Разработчик на софтуер (Software developer) ☐ Front-end уеб разработчик (Front-end web developer) ☐ Уеб разработчик на приложения (Web application developer) ☐
Опит в годините от работа в областта на софтуерните технологии	0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ >5 ☐
Източник на аргументация (Доколко сте запознат със системата?)	Съвсем повърхностно ☐ Повърхностно ☐ В детайли ☐

Функционалност (Functionality) – необходима е за оценяване дали изискваните функции за изпълняване са налице в софтуера.

Удачност (Suitability) – Осигурява ли системата подходящ набор от функции за специфични задачи и потребителски цели?

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Точност (Accuracy) - Осигурява ли системата точни или подходящи резултати или ефекти?

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Взаимоработоспособност (Interoperability) – Позволява ли системата взаимодействие с една или повече определени системи?

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Сигурност (Security) – Притежава ли системата способността да предотврати нерегламентиран достъп, независимо случаен или преднамерен, до програмите или данните?

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Съгласуваемост (Compliance) – Притежава ли системата способността да се придържа към стандарти, конвенции или регламенти, свързани с функционалността?

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Безотказност (Reliability) – необходима е за оценяване колко надеждна е работата на софтуера.

Отработеност (Maturity) – Притежава ли системата способността да се избегнат поражения (провал) в резултат на грешки в софтуера?

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Отказоустойчивост (Fault tolerance) – Притежава ли системата способността да поддържа установено ниво на действие в случай на софтуерни грешки или при нарушение на негов определен интерфейс?

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Възстановяемост (Recoverability) – Притежава ли системата способността да създаде наново нивото си на действие и да възстанови данните, пряко засегнати в случай на повреда за време и с действия необходими за това?

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Съгласуваемост (Compliance) - Притежава ли системата способността да се придържа към стандарти, конвенции или регламенти, свързани с надеждността?

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Използваемост (Usability) - необходима е за оценяване дали софтуерът е лесен за използване.

Разбираемост (Understandability) – Предоставя ли системата възможността да определя лекотата, с която функциите на софтуера, разпознаването на логическата му концепция и нейното прилагане могат да бъдат разбрани от потребителя.

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Изучаемост (Learnability) – Притежава ли системата способността да предостави възможност на потребителите да разучат възможностите на приложението?

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Манипулативност (Operability) - Притежава ли системата способността да предостави възможност на потребителите да работят и управляват приложението?

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Привлекателност (Attractiveness) - Притежава ли системата способността да е привлекателна и желана от потребителя при използването му в определени условия?

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Съгласуваемост (Compliance) - Притежава ли системата способността да се придържа към стандарти, конвенции или регламенти, свързани с използваемостта?

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Продуктивност (Efficiency) - необходима е за оценяване колко продуктивен е софтуерът.

Продуктивност по време (Time behavior) – Притежава ли системата способността да оценява времената за реагиране и протичане, както и скоростите на изпълнение функциите на софтуера? (колкото тези времена са по-малки, толкова софтуерът е по-продуктивен)

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Продуктивност по ресурси (Resource behavior) - Притежава ли системата способността да оценява количеството използвани ресурси и продължителността на това използване при изпълняване функциите ѝ? (колкото използваните ресурси и продължителността на използването им са по-малки, толкова софтуерът е по-продуктивен).

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Съгласуваемост (Compliance) - Притежава ли системата способността да се придържа към стандарти или конвенции, свързани с ефективността?

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Обслужваемост (Maintainability) - необходима е за оценяване колко лесен за модифициране е софтуерът.

Анализируемост (Analyzability) – Притежава ли системата способността да отразява усилието, нужно за диагноза на недостатъците или причините за отказите, или за идентификация на частите за модифициране на софтуера?

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Изменяемост (Changeability) – Притежава ли системата способността да отразява усилието, нужно за модифициране, отстраняване на причини за грешки или за промяна на среда? (включва се промяната на кода, дизайна и данните).

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Стабилност (Stability) – Притежава ли системата способността да минимизира неочакваните ефекти или последици при модифициране на софтуера?

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Изпитваемост (Testability) – Притежава ли системата способността да отразява усилието за потвърждаване на очакваните резултати от модифицирането на софтуера и правилността на извършването му?

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Съгласуваемост (Compliance) - Притежава ли системата способността да се придържа към стандарти или конвенции, свързани с обслужваемостта?

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Преносимост (Portability) - необходима е за оценяване колко лесен за преместване е софтуерът.

Приспособимост (Adaptability) - Притежава ли системата способността да отразява наличието на благоприятна възможност за адаптиране на софтуера към различни специфицирани среди без прилагане на други действия или средства, освен осигурените за тази цел от разглеждания софтуер?

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Инсталируемост (Installability) - Притежава ли системата способността да отразява нужното усилие, за да се инсталира софтуера в специфицираната среда?

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Съвместно съществуване (Co-existence) - Притежава ли системата способността да съществува съвместно с друг, независим софтуер в обща среда, споделяйки общи ресурси?

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Заменимост (Replaceability) – Притежава ли системата способността да отразява благоприятната възможност и усилието за използването ѝ, вместо друг установен софтуер, в средата на този друг софтуер?

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Съгласуваемост (Compliance) - Притежава ли системата способността да се придържа към стандарти или конвенции, свързани с преносимостта?

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Приложение 4: Анкетна карта (дидактически показатели)

Анкетна карта, съставена на база дидактически показатели, съгласувани с експерти в областта на електронното обучение.

Анкетна карта за оценка на ефективността по дидактически показатели на система за уеб-базирано обучение в Медицински университет-София

("Site-template" <http://discipline.mu-sofia.bg>

"Interactive Biochemistry – Sofia" <http://biochemistry.mu-sofia.bg>

"Physiology" <http://physiology.mu-sofia.bg>

"Physics and Biophysics" <http://physics-biophysics.mu-sofia.bg>

"Pharmacology and Toxicology" <http://pharmacology-toxicology.mu-sofia.bg>)

Заемана длъжност	Професор ☐ Доцент ☐ Асистент ☐
Опит в годините от работа в областта на информационните технологии и електронното обучение	0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ >5 ☐
Източник на аргументация (Доколко сте запознат със системата?)	Съвсем повърхностно ☐ Повърхностно ☐ В детайли ☐

Осигурява ли системата възможност за въвеждане на учебния материал с логическа последователност?

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Осигурява ли системата възможност за връзки между отделните структурни единици в различни части?

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Осигурява ли системата възможност за насърчаване за критично мислене и креативност?

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Осигурява ли системата връзка с други ресурси за допълнителна информация за изучавания проблем?

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Осигурява ли системата използване на различни мултимедийни компоненти?

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Осигурява ли системата наличие на речник на използваните термини в лекционния материал?

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Осигурява ли системата наличие на търсеща машина в основните модули?

Слаб: ☐	Добър: ☐	Силен: ☐	Не мога да преценя: ☐
--------------------------------	---------------------------------	---------------------------------	--

Коментар:

Приложение 5: Степен на компетентност на експертите в областта на софтуерните технологии

	Заемана длъжност			Опит в годините от работа в областта на софтуерните технологии									Източник на аргументация				
	Разработчик на софтуер	Front-end уеб разработчик	Уеб разработчик на приложения	a_1	0	1	2	3	4	5	< 5	a_2	Съвсем повърхностно	Повърхностно	В детайли	a_3	C
Експерт 1	12	0	0	12	0	0	0	0	0	0	12	12	0	6	0	6	2,00
Експерт 2	0	10	0	10	0	0	0	0	0	0	12	12	0	0	12	12	2,27
Експерт 3	0	0	2	2	0	2	0	0	0	0	0	2	0	6	0	6	0,67
Експерт 4	0	10	0	10	0	0	0	0	0	0	12	12	0	6	0	6	1,87
Експерт 5	12	0	0	12	0	0	0	0	0	0	12	12	0	0	12	12	2,40
Експерт 6	0	10	0	10	0	0	0	0	0	0	12	12	0	0	12	12	2,27
Експерт 7	0	0	8	8	0	0	0	0	0	0	12	12	0	0	12	12	2,13
Експерт 8	0	10	0	10	0	0	0	0	0	0	12	12	0	0	12	12	2,27
Експерт 9	12	0	0	12	0	0	0	0	0	0	12	12	0	0	12	12	2,40
Експерт 10	0	0	8	8	0	0	0	0	0	10	0	10	0	6	0	6	1,60
Експерт 11	0	0	8	8	0	0	0	0	0	0	12	12	0	6	0	6	1,73
Експерт 12	6	0	0	6	0	0	0	6	0	0	0	6	2	0	0	2	0,93
Експерт 13	0	10	0	10	0	0	0	0	0	0	12	12	0	6	0	6	1,87
Експерт 14	6	0	0	6	0	0	0	0	8	0	0	8	0	6	0	6	1,33
Експерт 15	12	0	0	12	0	0	0	0	0	0	12	12	0	0	12	12	2,40

Приложение 6: Степен на компетентност на преподаватели от институцията, използваща информационната система

	Заемана длъжност				Опит в годините от работа в областта на информационните технологии и електронното обучение									Източник на аргументация				
	Професор	Доцент	Асистент		a_1	0	1	2	3	4	5	< 5		a_2	Съвсем повърхностно	Повърхностно		
Преподавател 1	0	10	0	10	0	0	0	0	0	0	12	12	0	0	12	12	2,27	
Преподавател 2	12	0	0	12	0	0	0	0	0	0	12	12	0	6	0	6	2,00	
Преподавател 3	0	10	0	10	0	0	0	0	0	0	12	12	0	0	12	12	2,27	
Преподавател 4	0	0	4	4	0	0	0	0	0	0	12	12	0	0	12	12	1,87	
Преподавател 5	12	0	0	12	0	0	0	0	0	0	12	12	0	0	12	12	2,40	
Преподавател 6	0	0	2	2	0	2	0	0	0	0	0	2	0	6	0	6	0,67	
Преподавател 7	0	0	2	2	0	0	4	0	0	0	0	4	2	0	0	2	0,53	
Преподавател 8	0	10	0	10	0	0	0	0	0	0	12	12	0	6	0	6	1,87	
Преподавател 9	0	0	4	4	0	0	0	0	0	0	12	12	0	0	12	12	1,87	
Преподавател 10	0	0	4	4	0	0	0	0	0	0	12	12	0	6	0	6	1,47	
Преподавател 11	0	0	4	4	0	0	0	0	0	10	0	10	0	0	12	12	1,73	
Преподавател 12	0	0	2	2	0	2	0	0	0	0	0	2	2	0	0	2	0,40	
Преподавател 13	0	0	4	4	0	0	0	0	0	10	0	10	0	6	0	6	1,33	