Смесено обучение по софтуерни технологии

ДОЦ. АСЯ СТОЯНОВА-ДОЙЧЕВА

ПУБЛИЧНА ЛЕКЦИЯ

СЕМИНАР НА ФМИ НА ПУ – ГР. ХИСАР

10.11.2015

Смесено обучение (Blended Learning)

- Форма на формално обучение, която се провежда по предварително установена учебна програма и разписание с частична доставка на учебния материал по електронен път.
  - Основни предимства са подобрените възможности за анализ, използване на ресурсите, организация на времето, съчетаване с практикуване и персонализация на традиционния учебен процес.
- DeLC порталът подпомага провеждане на смесено обучение в магистърската програма "Софтуерни технологии" и частично в бакалавърските програми в компютърните науки на ФМИ на Пловдивския университет.

Какво е SCORM 2004 R4

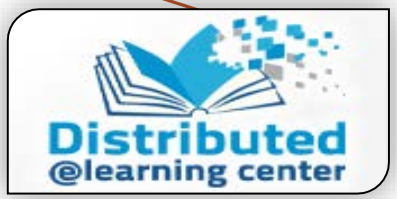
- Дефиниция - **Sharable Content Object Reference Model** (**SCORM**) е набор от стандарти и спецификации за уеб-базирано електронно обучение. Той определя комуникациите между съдържанието от страна на клиента и хост система, наречена среда за изпълнение, която обикновено се поддържа от Системата за управление на обучението. SCORM също дефинира как учебното съдържание може да бъде структурирано.

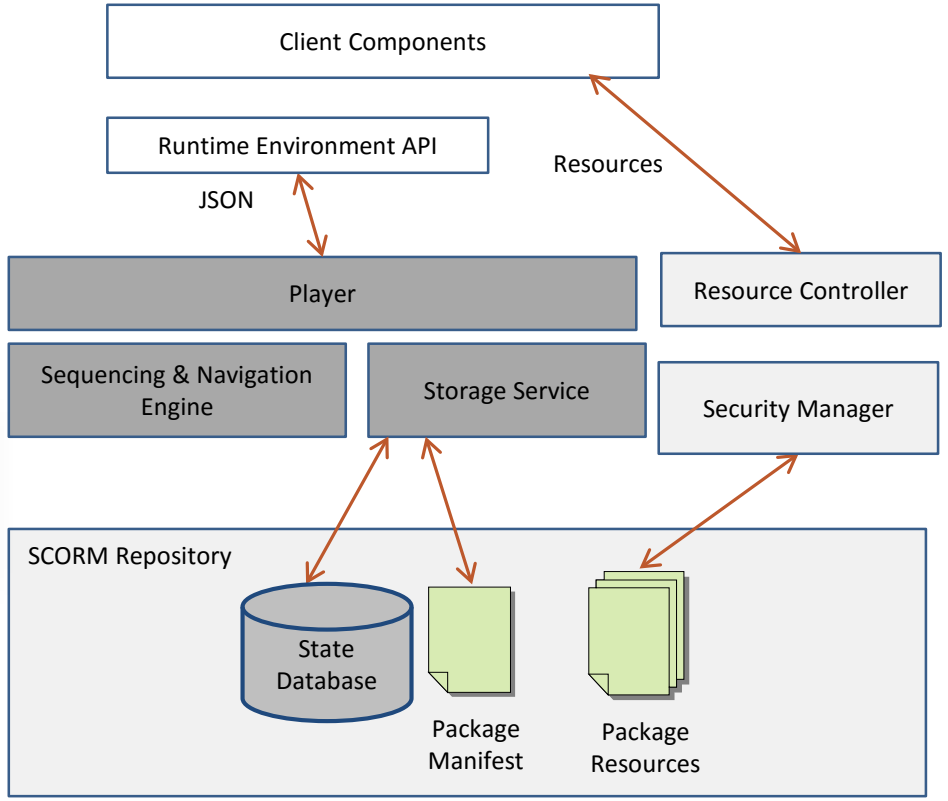
  - Създаване на малки и преизползваеми компоненти – **SCOs** – те съдържат **assets** (текст, картинки, анимации, диаграми и др.)
  - Организацията на SCOs в по-големи компоненти се наричат **Aggregations**
  - Дефиниране на **sequencing and navigation** между различните компоненти

SCORM Engine в DeLC

SCORM Engine

- SCORM 2004 R4
- DeLC SCORM Player
- Продължава ADL сертификация
- Интегриран е в DeLC2
- Имплементация:
  - Groovy, Grails (JVM)





Разработване на електронно учебно съдържание в SCORM формат

Електронен SCORM учебник по СТ

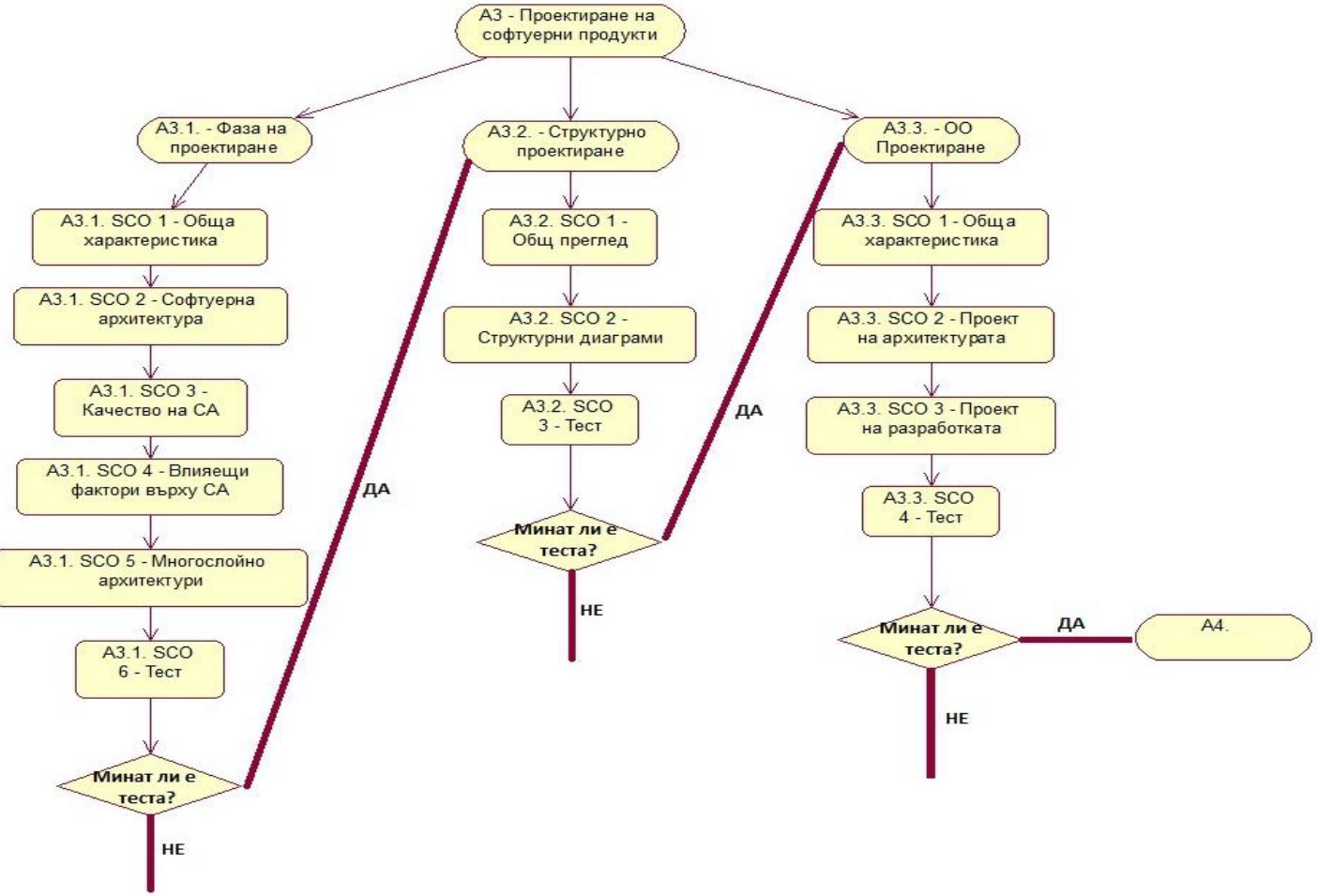
http://delc2.fmi.uni-plovdiv.net/

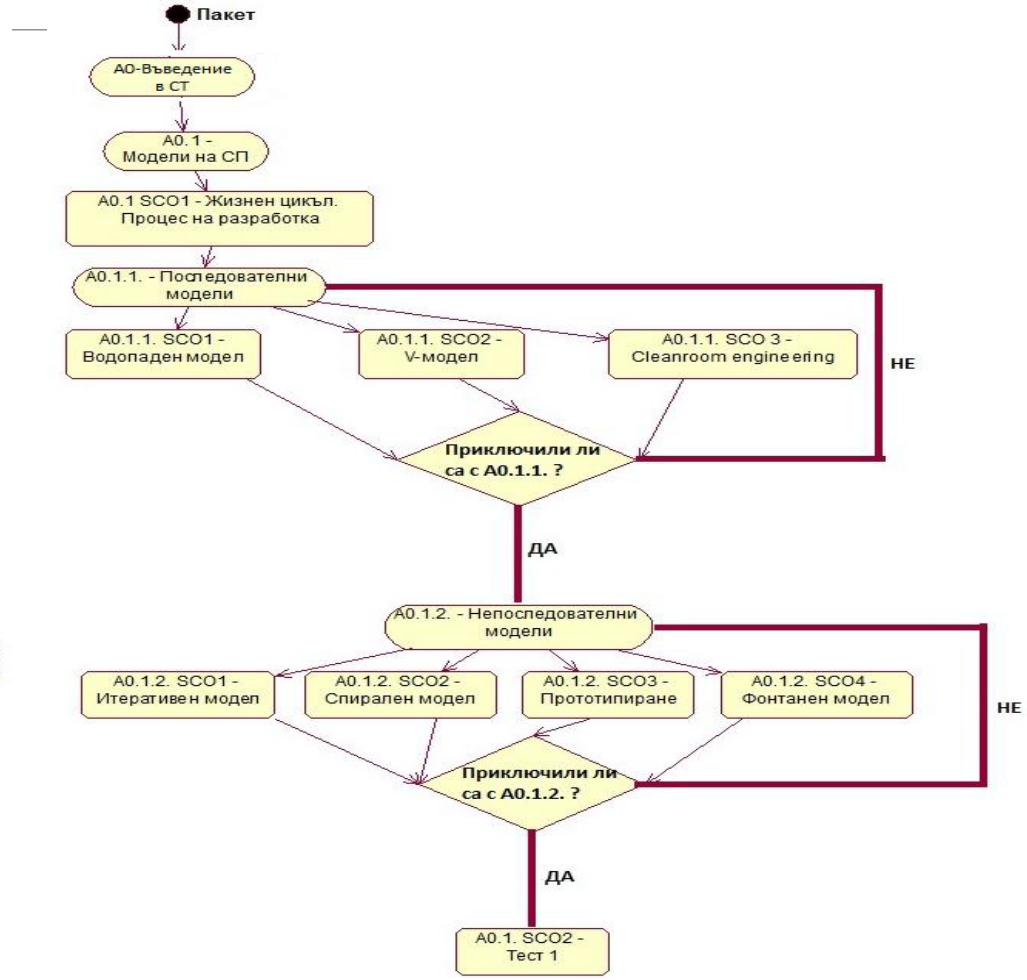
Електронно SCORM ръководство по СТ

Процес на разработване

1. Проектиране на учебника в SCORM формат;
2. Разработване на електронните ресурси за всяко SCO ;
3. Дизаин и разработване на електронния учебник в SCORM редактор;
4. Имплементация на контролните тестове;
5. Тестване на учебника в DeLC SCORM-player.

Проектиране на учебника в SCORM формат





Проектиране на контролните тестове

Тест 1

Въведение в СТ и разработка на СП

1. С какво се занимава дисциплината СТ?

- С разработка на софтуер
- с разработка на качествен софтуер
- с определяне качеството на софтуера

2. Кои са фазите на жизнения цикъл на разработката на софтуер?

- анализ, проектиране, разработка, тестване, поддръжка
- анализ на домейна, разработка, тестване, поддръжка
- анализ и дефиниция, разработка, тестване, поддръжка

3. Какви недостатъци има водопадния модел?

- Грешките в проектирането или анализа се откриват късно
- Не могат да се провеждат тестове
- Клиентите не участват в процеса

4. V-модела се нарича още:

- модел за тестване на софтуер
- модел за сигурност
- модел на функционалните точки

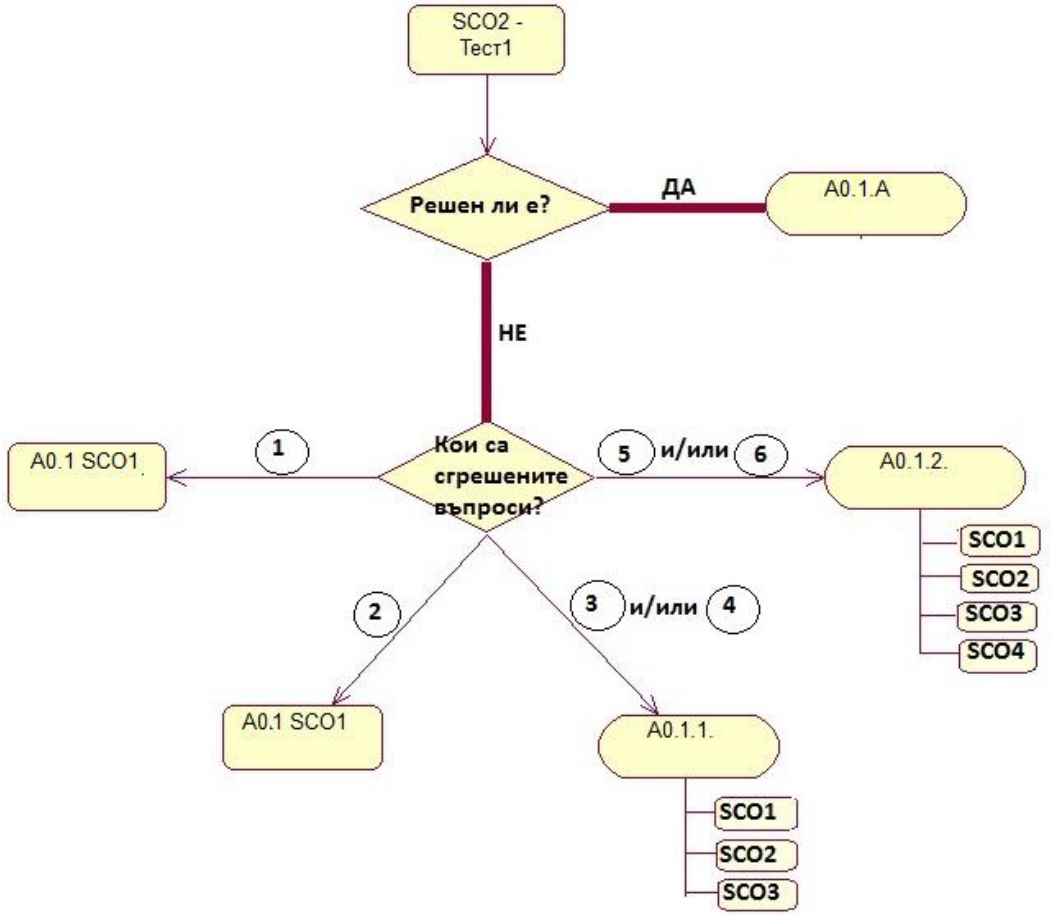
5. Колко типа прототипиране има?

- Два - хоризонтално и вертикално
- Един
- Няма верен отговор

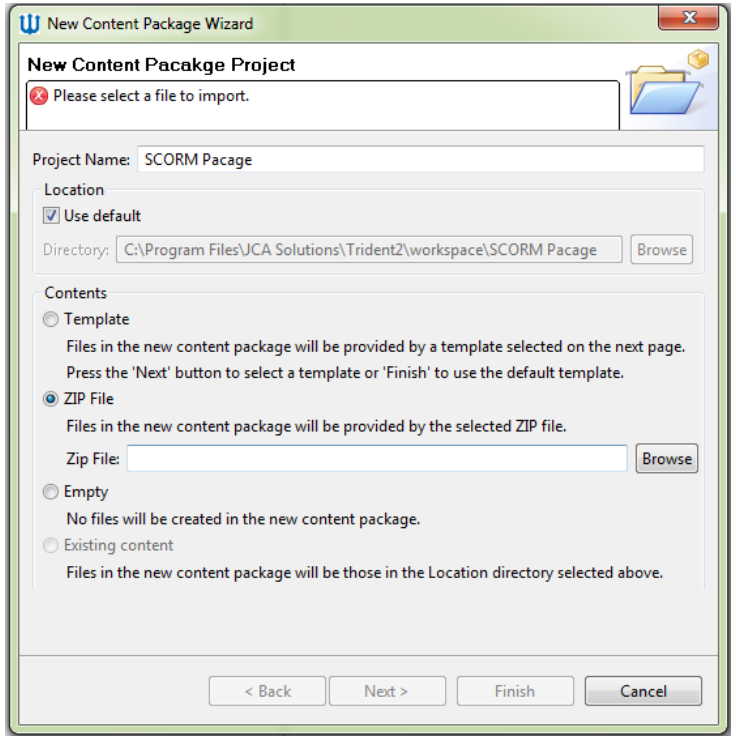
6. Итеративният модел на процеса се характеризира с:

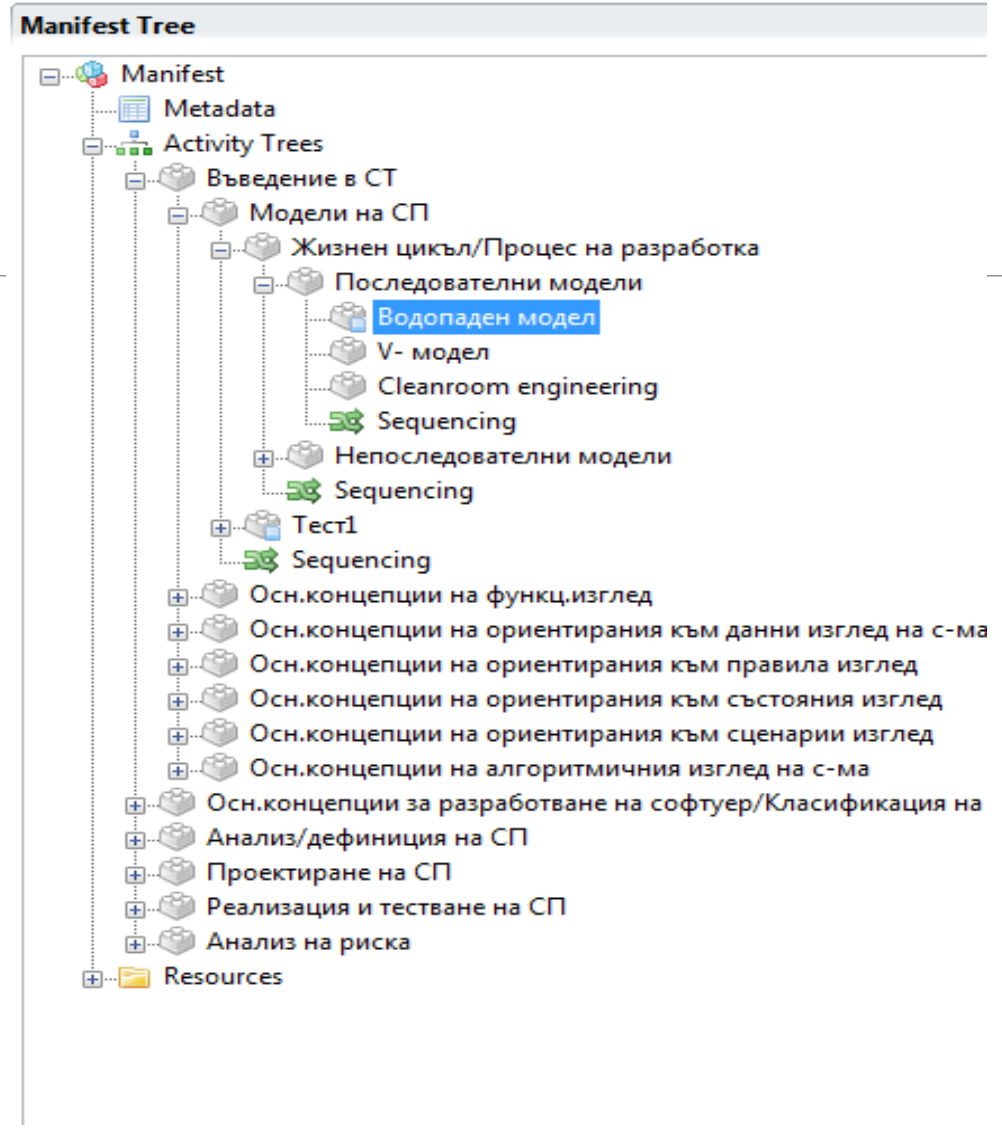
- разработване на софтуера на версии
- разработване на софтуера на малки части
- и двата отговора са верни
- няма верен отговор

Провери

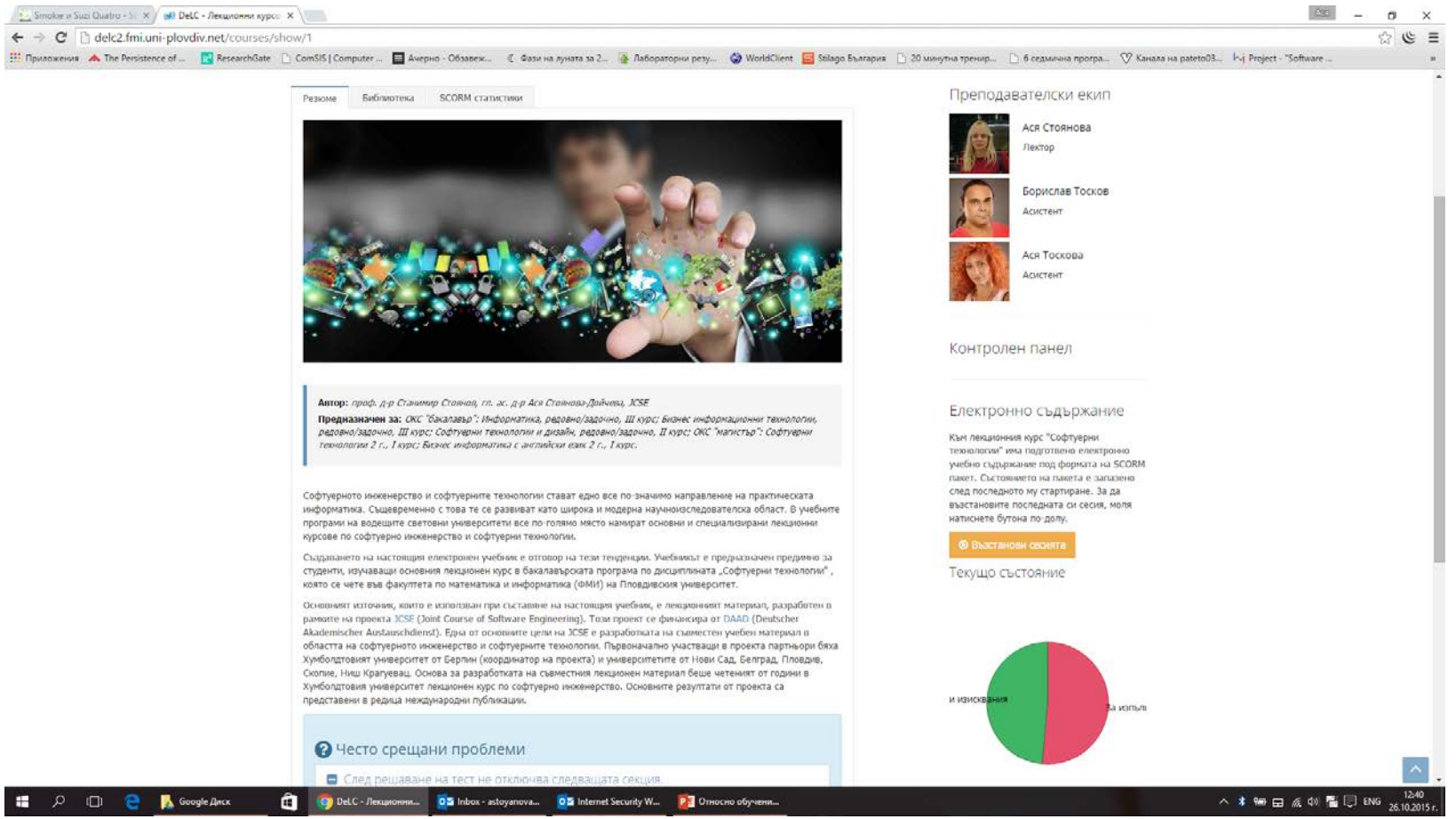


Trident 2.0
The SCORM IDE

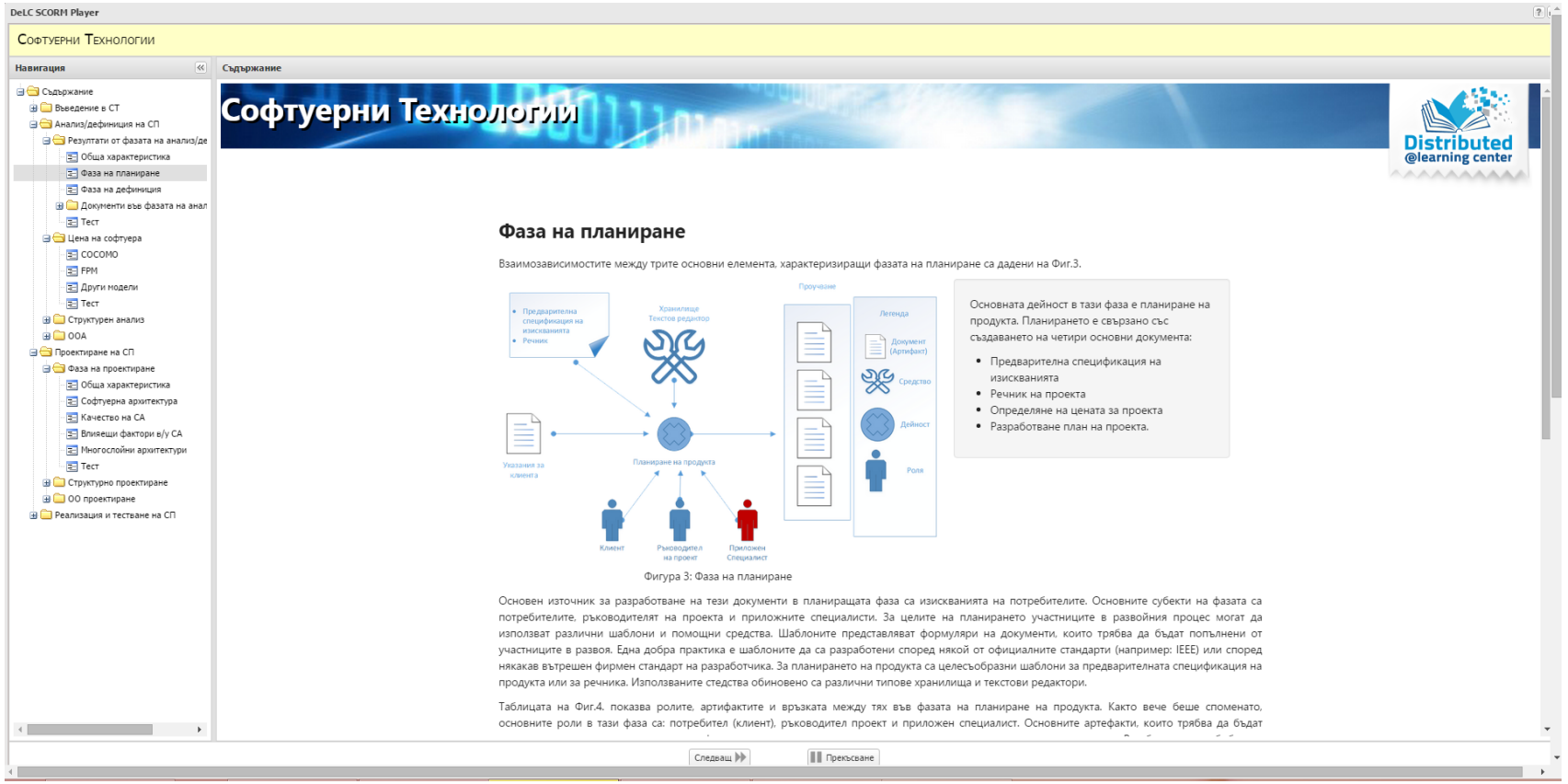




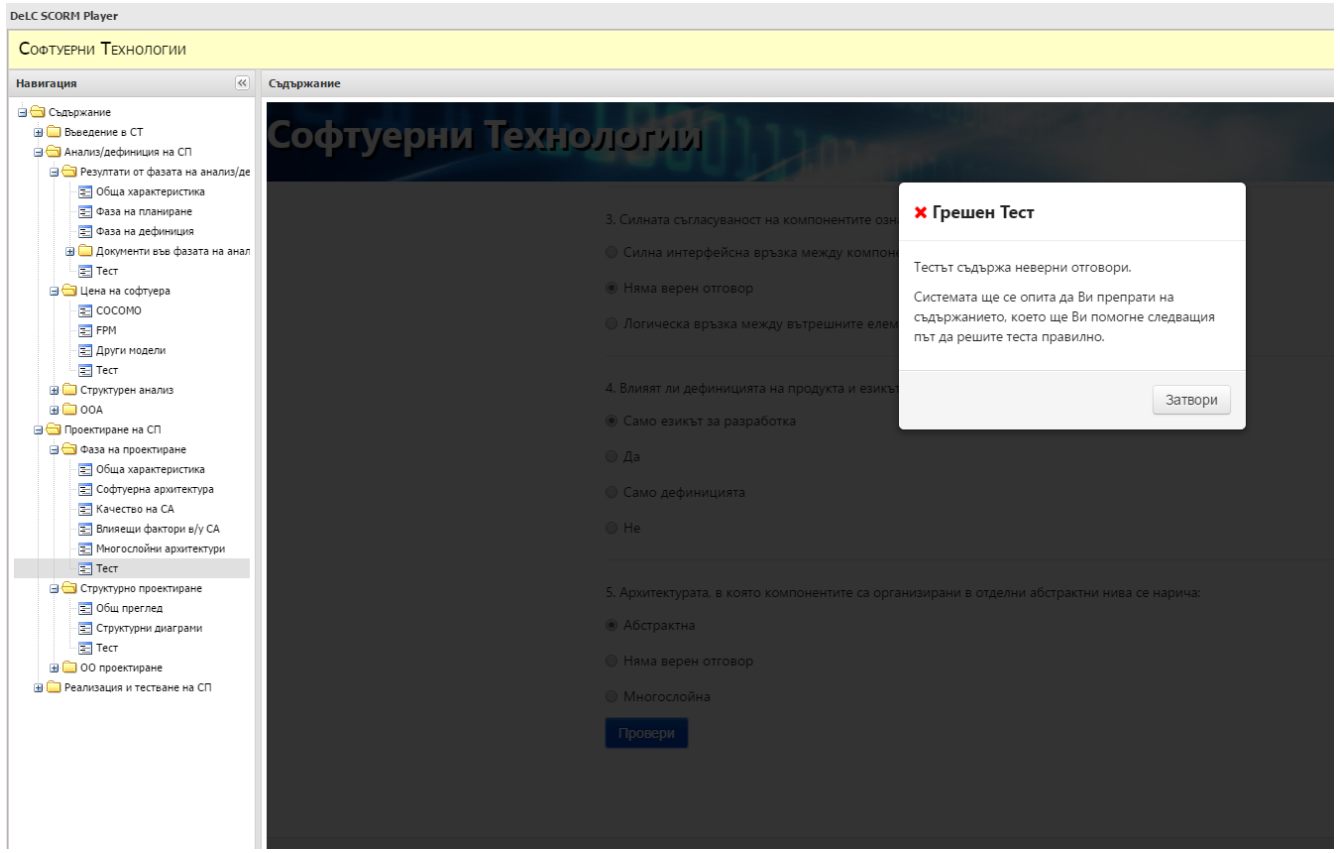
E-учебник по софтуерни технологии



Е-учебник по софтуерни технологии



Грешен тест!


DeLC SCORM Player
Софтуерни Технологии
Навигация
Съдържание
Съдържание
Въведение в СТ
Анализ/дефиниция на СП
Резултати от фазата на анализ/де
Обща характеристика
Фаза на планиране
Фаза на дефиниция
Документи във фазата на анал
Тест
Цена на софтуера
COCOMO
FPM
Други модели
Тест
Структурен анализ
OOA
Проектиране на СП
Фаза на проектиране
Обща характеристика
Софтуерна архитектура
Качество на СА
Влияещи фактори в/у СА
Многослойни архитектури
Тест
Структурно проектиране
Общ преглед
Структурни диаграми
Тест
ОО проектиране
Реализация и тестване на СП
Софтуерни Технологии
3. Силната съгласуваност на компонентите озн
Силна интерфейсна връзка между компон
Няма верен отговор
Логическа връзка между вътрешните елем
4. Влияят ли дефиницията на продукта и езикъ
Само езикът за разработка
Да
Само дефиницията
Не
5. Архитектурата, в която компонентите са организирани в отделни абстрактни нива се нарича:
Абстрактна
Няма верен отговор
Многослойна
Провери
Грешен Тест
Тестът съдържа неверни отговори.
Системата ще се опита да Ви препрати на съдържанието, което ще Ви помогне следващия път да решите теста правилно.
Затвори

Как използваме е-учебник в обучението по Софтуерни технологии

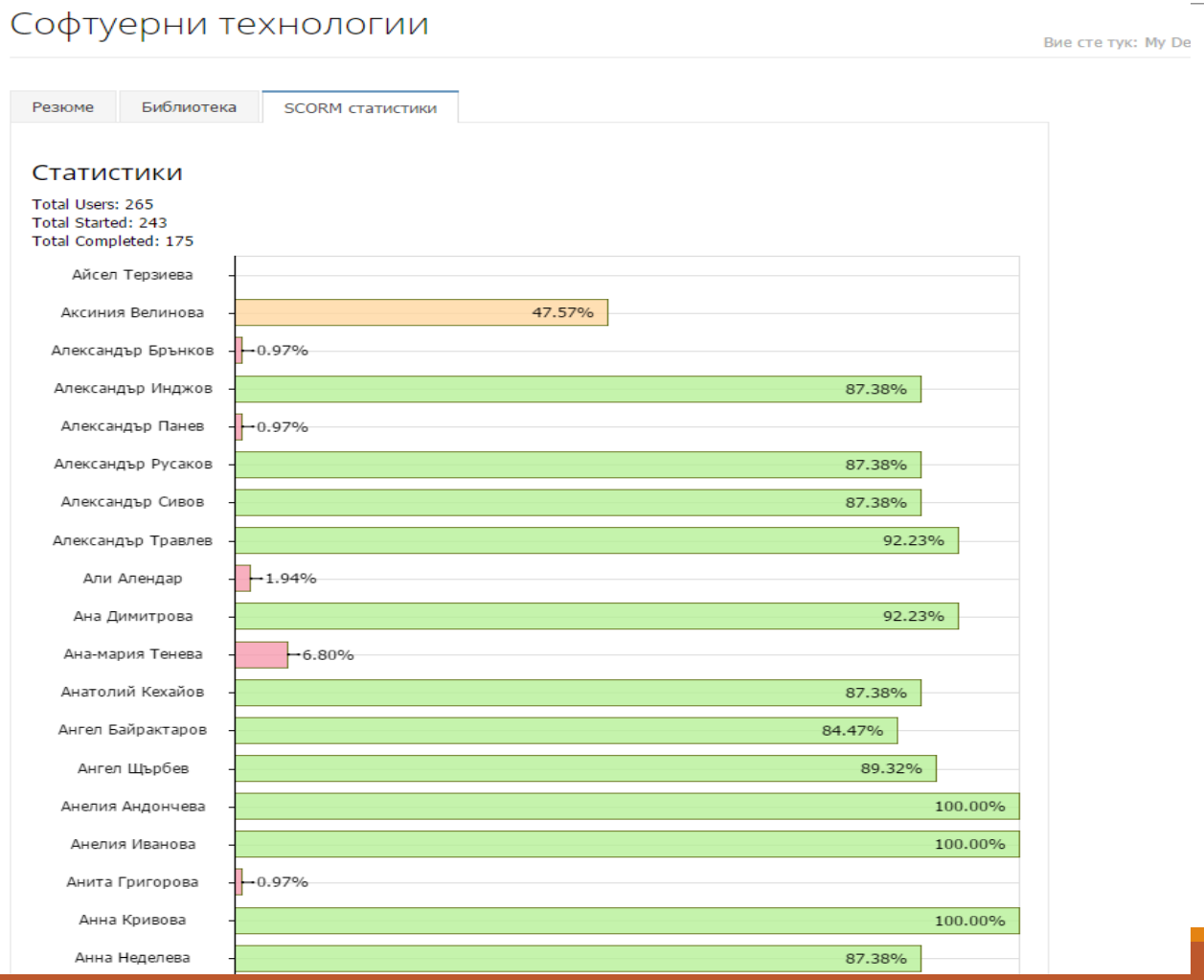
Семестриална работа според учебния план

- 30 часа лекции – face to face обучение, използваме материали от JCSE
- 20 часа упражнения – UML
- 100 часа самоподготовка – освен предложената литература, студентите трябва да премината е-учебника
- Обратна връзка : от студентите , **от SCORM**

Изпит – крайната оценка е сума от:

- UML – практически изпит;
- Краен тест върху материала на лекции;
- **Преминаването на учебника дава допълнителни точки към крайния резултат!**

SCORM статистика



SCORM статистика

Stats for user with ID 234

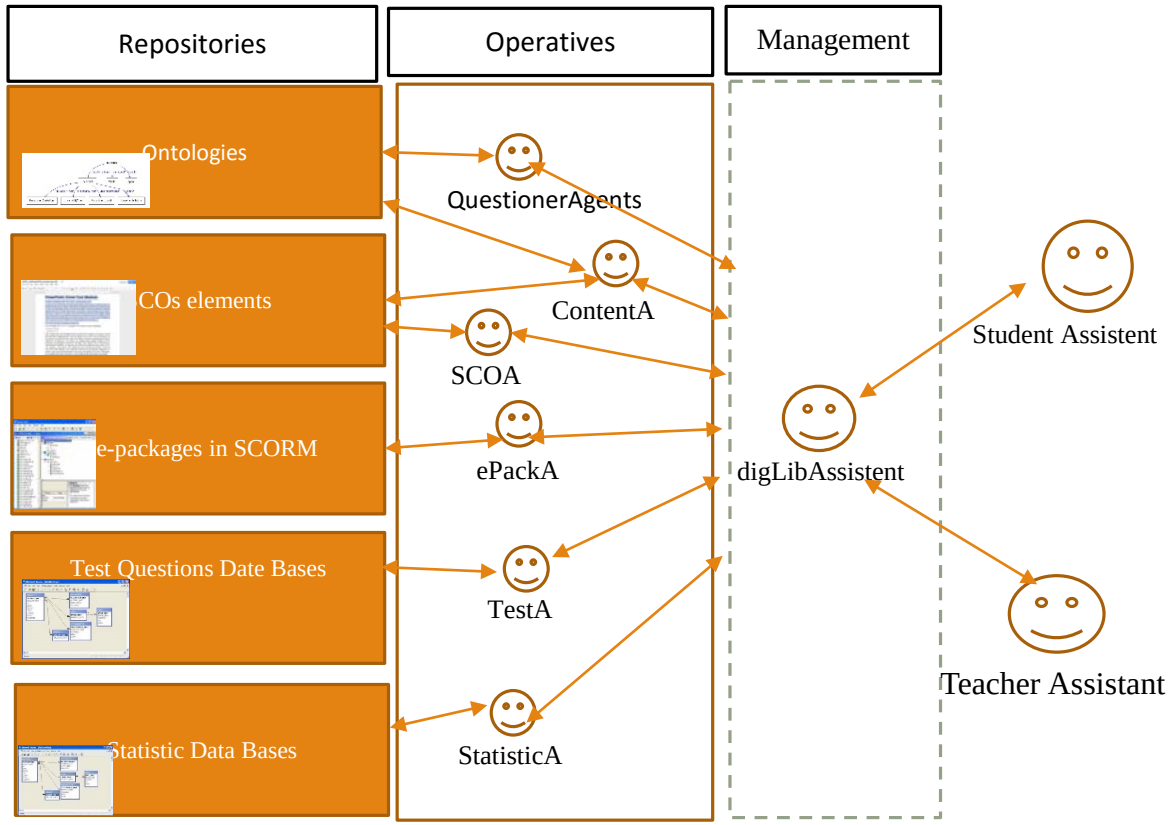
ctrivity ID	Title	Attempts
CT-813274	Прототипиране	1
CT-860078	Въведение	1
CT-971339	Тест	1
CT-259305	Въведение	1
CT-848767	Речници от данни	1
CT-687593	Въведение	1
CT-867276	Правила	1
CT-319886	Въведение	1
CT-68501	Тест	1
CT-979258	Sequence диаграми	1
CT-828976	Collaboration диаграми	1
CT-627485	Обща характеристика	1
CT-683133	Спецификация на изискванията	1
CT-701674	FPM	1
CT-256247	Други модели	1
CT-93136	Осн. принципи	1
CT-402361	Речници от данни и усъвършенстване на данните	1
CT-294616	Тест	1
CT-320917	ERD	2
CT-864274	Тест	2
CT-Introduction	Въведение	3
CT-540318	Cleanroom engineering	3
CT-155499	Спирален модел	3
CT-132900	Фонтанен модел	3
CT-149332	Тест	3
CT-132198	V- модел	4
CT-223897	Тест	4
CT-983256	Въведение	4
CT-726476	Тест	4
CT-737424	Итеративен модел	6
CT-255427	Тест	8
CT-271201	COCOMO	9
CT-998208	Модели на СП	10
CT-823647	Тест	25
CT-889116	Водопаден модел	27

activity_id	title	attempts	users	attempts_for_success	parent
ACT-405926	Многослойни архитектури	188	130	1,4462	Фаза на проектиране
ACT-319886	Въведение	292	199	1,4673	Осн.концепции на ориентирания към състояния изглед
ACT-53607	Системни тестове	176	119	1,479	Стратегия за тестване на софтуер
ACT-80387	Влияещи фактори в/у СА	199	128	1,5547	Фаза на проектиране
ACT-470479	Activity диаграми	236	150	1,5733	Осн.концепции на ориентирания към състояния изглед
ACT-307755	Интегр. тестове	188	119	1,5798	Стратегия за тестване на софтуер
ACT-509617	Валид. тестове	178	112	1,5893	Стратегия за тестване на софтуер
ACT-979258	Sequence диаграми	237	148	1,6014	Осн.концепции на ориентирания към сценарии изглед
ACT-156169	Валидационни тестове	182	112	1,625	Тестване на ОО софтуер
ACT-763070	Динамичен модел	231	132	1,75	ООА
ACT-431728	Статичен модел	256	145	1,7655	ООА
ACT-241207	Дърво на решенията	271	152	1,7829	Осн.концепции на ориентирания към правила изглед
ACT-673369	Общ преглед	330	185	1,7838	Структурно проектиране
ACT-186866	Пример за спецификация на изискванията	228	127	1,7953	Документи във фазата на анализ/дефиниция
ACT-130755	Use case	279	155	1,8	Осн.концепции на функц.изглед
ACT-487185	Таблица на решенията	270	150	1,8	Осн.концепции на ориентирания към правила изглед
ACT-687593	Въведение	377	202	1,8663	Осн.концепции на ориентирания към правила изглед
ACT-808281	Крайни автомати(State transition)	282	151	1,8675	Осн.концепции на ориентирания към състояния изглед
ACT-259305	Въведение	379	202	1,8762	Осн.концепции на ориентирания към данни изглед на с-ма
ACT-519927	Обща характеристика	342	182	1,8791	ОО проектиране
ACT-867276	Правила	312	166	1,8795	Осн.концепции на ориентирания към правила изглед
ACT-50435	Структурни диаграми	267	139	1,9209	Структурно проектиране
ACT-613241	Проект на разработката	263	135	1,9481	ОО проектиране
ACT-411436	Проект на архитектурата	267	137	1,9489	ОО проектиране
ACT-828976	Collaboration диаграми	294	150	1,96	Осн.концепции на ориентирания към сценарии изглед
ACT-90810	Обща характеристика	373	190	1,9632	ООА
ACT-291771	Модел на продукта	249	125	1,992	Документи във фазата на анализ/дефиниция
ACT-590636	Качество на СА	269	135	1,9926	Фаза на проектиране
ACT-732699	Софтуерна архитектура	285	143	1,993	Фаза на проектиране
ACT-614991	Речник	265	132	2,0076	Документи във фазата на анализ/дефиниция
ACT-701674	FPM	300	149	2,0134	Цена на софтуера
ACT-86344	Миниспецификации	292	145	2,0138	Структурен анализ
ACT-594438	Обща характеристика	380	188	2,0213	Фаза на проектиране
ACT-256247	Други модели	300	145	2,069	Цена на софтуера
ACT-398061	Предварителна спецификация на изискванията	274	131	2,0916	Документи във фазата на анализ/дефиниция
ACT-450472	Въведение	385	181	2,1271	Стратегия за тестване на софтуер
ACT-952942	Тест	404	185	2,1838	Структурно проектиране

Изводи

- Използването на учебника е полезно и за студентите и за преподавателите;
- Мотивация на студентите;
- SCORM статистиката:
  - е добър начин за проследяване на прогреса на студентите по време на семестъра;
  - Помага на преподавателя да оцени качеството на електронното съдържание.
- SCORM помага за персонализацията на учебния процес.

DigLib във ВОП

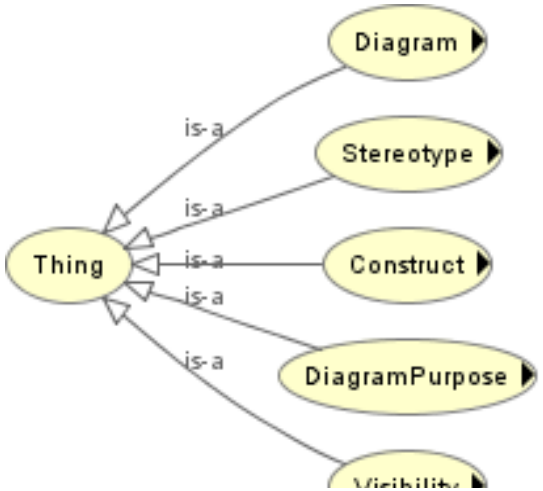


UML тестова среда – Questioner Agents

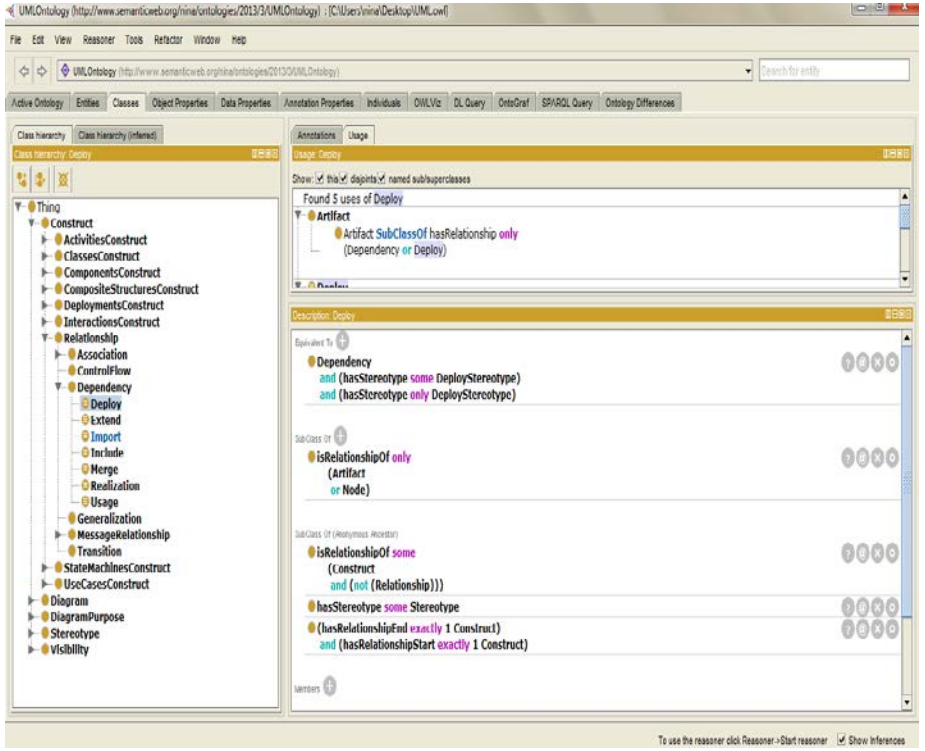
- Основната функционалност на средата е да генерира тестове за UML, да ги представя на студента в подходящ вид и да проверява отговорите.

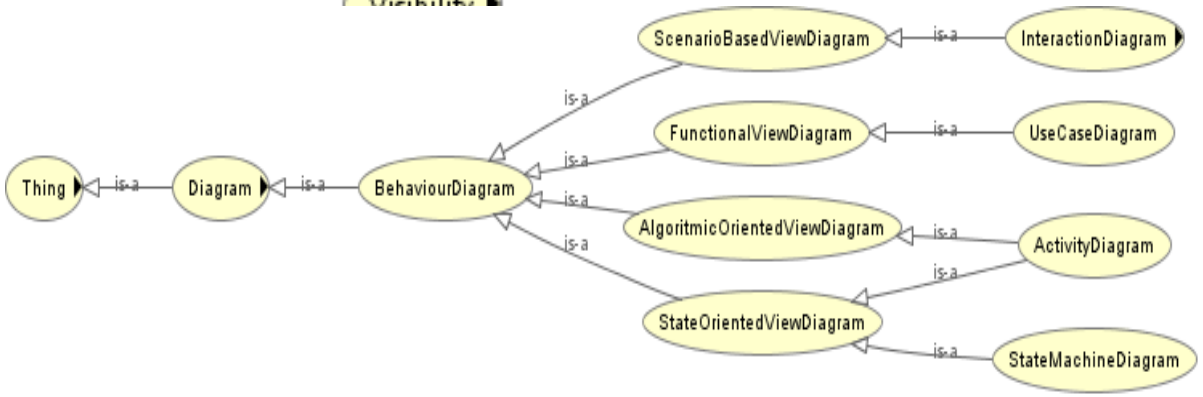
- Тестовете са проектирана да съдържат последователност от въпроси с кратък отговор, като всички въпроси в един тест са различни.

UML онтология

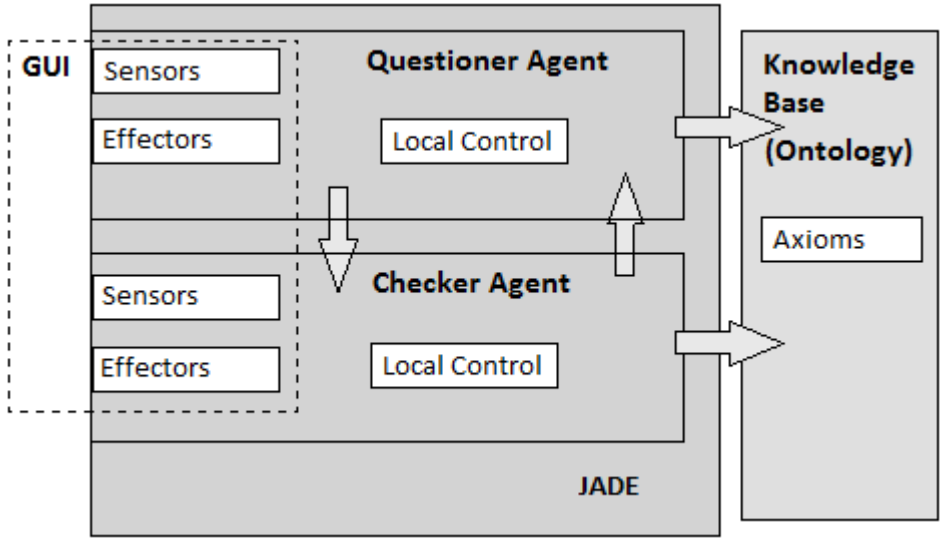


Имаме над 850 аксиоми





Архитектура на средата



Front-end component – Graphical User Interface (GUI);

Back-end component –интелигентни агенти, които генерират теста, проверяват отговорите на потребителите и анализират резултата. За тази цел използват база знания онтология.

Questioner Agent

Генерира въпроси от създадената структурата, връзките и аксиомите в онтологията.

ПРИМЕР:

Произволно избираме от онтологията класа UseCase и избрания израз от този клас казва, че той може да се свърже посредством връзката hasRelationship с класовете BidirectionalAssociation или Include или Extend.

Генерирания въпрос ще е: UseCase hasRelationship ……

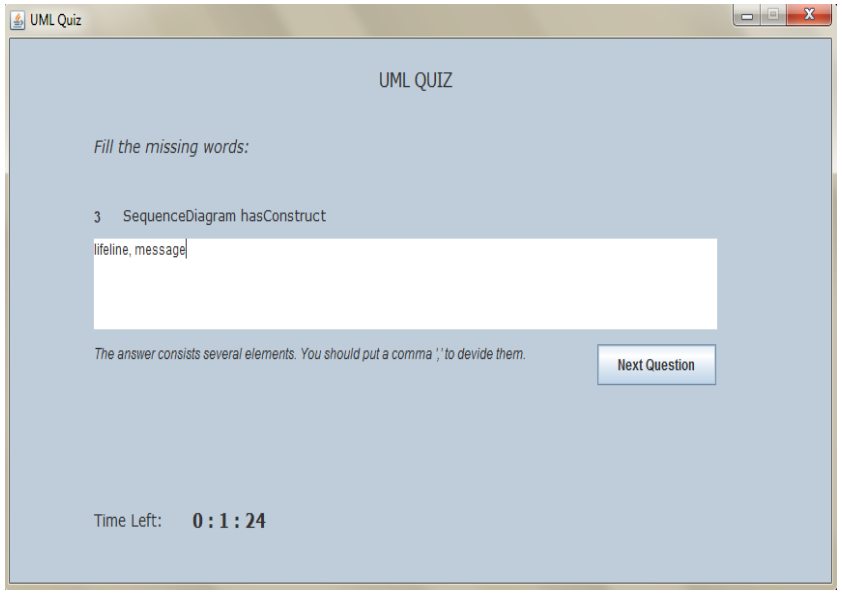
Студента трябва да изброи посочените класове.

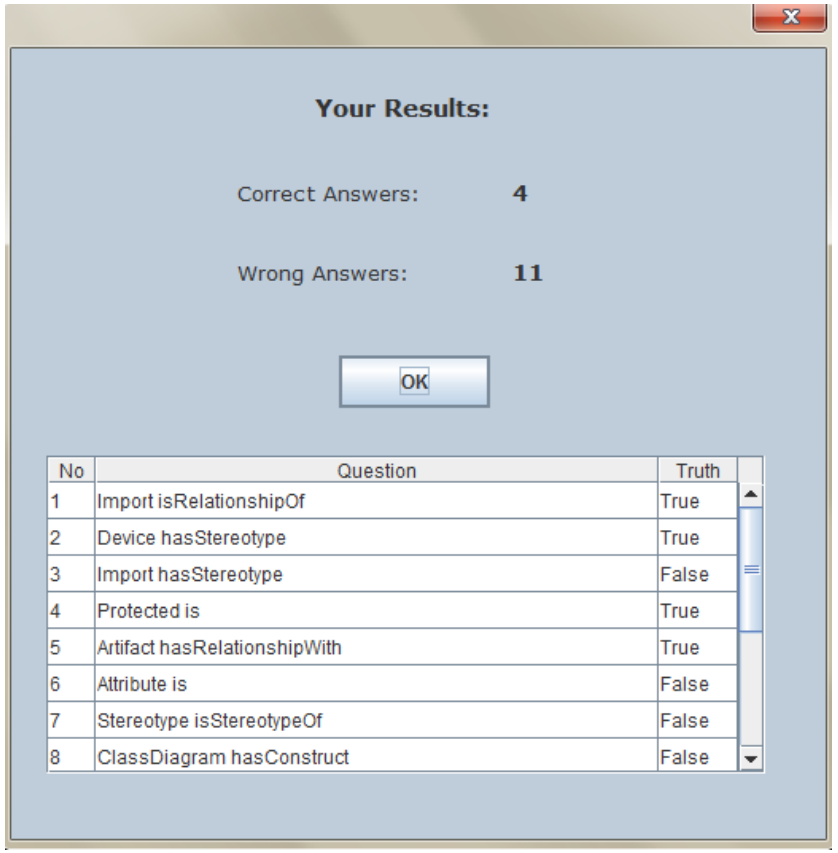
Assessment Agent

Поведение на Оценящия Агент:

- Получава името на класа и аксиомата от QA. Те се извличат в OWLClass и OWLAxiom обекти, така че да могат да се ползват по-късно при оценяването.
- Получава отговора на въпроса от GUI
- Assessment Service се активира когато въпроса и отговора са получени
- SendCorrectAnswersService – изпраща резултатите до студента след приключване на теста.

GUI


UML Quiz
UML QUIZ
Fill the missing words:
3 SequenceDiagram hasConstruct
lifeline, message
The answer consists several elements. You should put a comma ',' to devide them.
Next Question
Time Left: 0 : 1 : 24


Your Results:
Correct Answers: 4
Wrong Answers: 11
OK
No Question Truth
1 Import isRelationshipOf True
2 Device hasStereotype True
3 Import hasStereotype False
4 Protected is True
5 Artifact hasRelationshipWith True
6 Attribute is False
7 Stereotype isStereotypeOf False
8 ClassDiagram hasConstruct False

Бъдеща работа

- Превод на онтологията на български;
- Добавяне на още типове въпроси според стандарта QTI;
- Експортиране на въпросите в XML.

Благодаря за вниманието!

