

Изучаване на UML в магистратурата по Информатика

Пламенка Христова

Studying UML in the "Master Degree in Informatics" education level: This paper presents an approach to studying the UML modeling language in the "Master Degree in Informatics" level of education. Emphases on the necessities and the benefits, which can be achieved by studying UML, are made. Also a comparison with the studying of the UML modeling language in similar education levels in Sofia University has been done. Conclusions are made and recommendations for studying improvements are given.

Key words: Education, Informatics, Software Engineering, Computer Systems and Technologies, UML.

ВЪВЕДЕНИЕ

През последните 20 години софтуерът се превърна не само в голяма световна индустрия, но и в основен елемент от модерната икономическа инфраструктура според проучване на Световната банка под заглавие The Role of Software in Economic Development [1].

Далеч не е преувеличено да се каже, че в началото на XXI век софтуерът е близък аналог на това, което са били въглищата и стоманата 100 години по-рано - фундаментален фактор на икономическия растеж. В съвременната глобална икономика просто не съществува организация, която може да функционира ефективно без използването на софтуер. Нещо повече, във всеки момент, когато говорим по мобилния си телефон, сърфираме в интернет, гледаме телевизия или управляваме нов автомобил, ние сте зависими от хиляди редове програмен код, написани от някого в места като София, Сан Франциско или Бангалор. През последните години софтуерният бранш е сред най-динамично развиващите се сектори на българската икономика. Годишният растеж на този сектор е трайно висок и се измерва в двуцифрени стойности (10% и повече).

Един от най-големите проблеми на софтуерната индустрия е липсата на стандарти и/или неспазването на наличните стандарти. Последните 5-10 год. се наблюдават усилия за създаване на стандарти. Трябва да се отбележи, че днес има стандарти в почти всяка една област на софтуерната индустрия. Проблемът със съществуващите стандарти е, че почти никой не е запознат с тях и съответно никой не ги спазва. Голяма част от софтуерните фирми толерират неспазването на стандартите и използването на техните собствени спецификации. По този начин се цели дългосрочно обвързване на клиента с продуктите на компанията. Създаването и спазването на норми и стандарти при разработването на програмни продукти, превръща тази дейност от изкуство в производство. Това води до изграждане на система за оценка на качеството на софтуера и съпътстващата го документация, оценка на риска при внедряване на системите и т.н.

За създаването на софтуер са необходими подготвени хора, докато за някои други отрасли освен кадри са необходими и природни дадености, рудни залежи и пр. Повечето от провалите в софтуерната индустрия се отчитат като обективни - лоша комуникация, лошо планиране, неясни задания от страна на клиента и др., но всички те водят към липсата на подготвени кадри. В липсата на подготвени кадри няма нищо изненадващо, компютрите навлязоха с шеметна скорост във всяка една сфера на нашия живот - стопанската, финансовата, политическата и пр. Търсенето на кадри изпревари подготовката им.

За обучението на един добре подготвен специалист в сферата на ИТ са необходими 5 год. в университета и после около 5 год. практика и чиракуване. Точно толкова, колкото и за подготвянето на един добър инженер или лекар.

Голямото търсене на ИТ специалисти доведе до покачване на заплатите в бранша, което от своя страна доведе до наплив на хора от други професии с или без

завършено висше образование по някаква специалност. Според някои данни над 70% от хората, заети в софтуерната индустрия са от други професии. Това от своя страна води до влошаване на качеството при разработване на софтуерни продукти и е предпоставка за допускане на провали в сферата на ИТ. В настоящия момент възниква необходимост обучението по софтуерни технологии да се придържа към съществуващите стандарти. По този начин специалистите, участници в даден проект ще притежават необходимата квалификация да възприемат технологията по еднакъв начин, и да допринасят максимално за качествената реализация на проекта в срок. На тези изисквания отговаря езика за проектиране UML (Unified Modeling Language) – Унифицирания език за моделиране, създаден през 1997 г. от отворения консорциум от компании Object Management Group (OMG), съчетаващ в себе си както спазването строгостта на стандарта, така и разнообразни възможности за формализиране и документиране на различните гледни точки за представянето на проект [8, 3, 7].

Преподаването на UML се превърна в задължителна част от обучението на студентите по информатика. Целта на настоящата работа е да разгледа мястото на UML в дисциплината Съвременни софтуерни технологии от магистърския курс по Информатика в РУ “Ангел Кънчев”.

РАЗВИТИЕТО НА ТЕХНОЛОГИЯТА НА СОФТУЕРНОТО ПРОИЗВОДСТВО И ПОСТАВЯНЕТО МУ НА ПРОМИШЛЕНИ ОСНОВИ

Основните проблеми, които съпътстват разработването на съвременни софтуерни системи са:

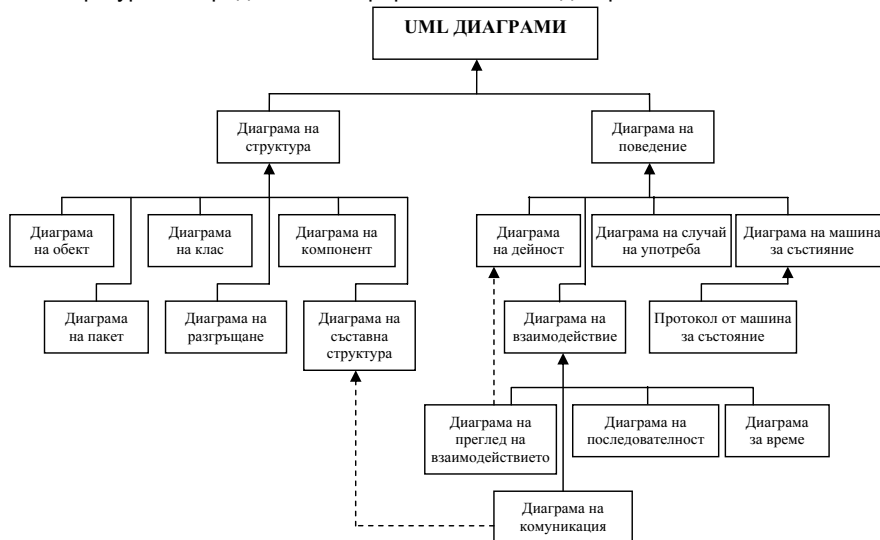
- Зле дефинирани и непрекъснато изменящи се изисквания във фазата на анализа и проектирането;
- Липса на общ език между ИТ специалистите от страна на разработчиците и бизнес тима на клиентите и възложителите;
- Оценки за срокове и приблизителна стойност на разработките, на които не може да се вярва, понеже са твърде индивидуални. В процеса на анализа и дизайна е загубена метриката на разработката;
- Неспазване на договорените срокове и надхвърляне на предвидените разходи;
- Непреодолимото предизвикателство да се поддържат създадените системи без актуална и подробна документация.

Разработката на все по-сложни системи поставя нови изисквания пред методите за разработка на софтуер. През последните години обектно-ориентираният подход се наложи като стандарт при разработката на бизнес софтуер. Необходимостта от общи стандарти в тази област доведе до появата на UML. Характерни черти на UML са [2, 3, 4, 5]:

- Съвместима форма за комуникация и представяне на идеите по време на анализ и дизайн фазите;
- Графично представяне, което се възприема еднакво добре от разработчиците и представителите на възложителя – хората от предметната област;
- Разширяем език, който може да се напаса за описване на широк клас приложения от най-разнообразни области – индустрия, машиностроене, бизнес приложения, сложни софтуерни системи, работещи в реално време;
- Програмният език на UML дава независим начин за специфициране на софтуер;
- Прецизно средство за описване на софтуерни структури и поведение и генериране на код, дори за генериране на цялостни приложения (executable UML);
- Средства за работа в UML среда се предоставят в редица съвременни среди за разработване на софтуер (Rational Rose, Net Beans, Eclipse, Visio и др.).

UML предлага набор от графични средства, позволяващи всеотраслен анализ на сложни проекти както от гледна точка на бизнес-целите, така и от гледна точка на техническата реализация. Езикът опростява проектирането, редуцира свързаните с него разходи и повишава ефективността.

На фигура 1 е представена йерархията на UML диаграмите.



Фиг. 1 Йерархия на основните видове UML диаграми

Предимствата му са: стандартна методология, интуитивно ясни означения, автоматично генериране на код.

Недостатъци: необходимост от обучение и едновременно възприемане на технологията от всички участници в проекта.

ИЗУЧАВАНЕ НА UML В МАГИСТЪРСКИЯ КУРС ПО ИНФОРМАТИКА

Изучаването на UML е включено в дисциплината “Съвременни софтуерни технологии” (ССТ), която е задължителна в магистърския курс на специалност Информатика. Дисциплината е с хорариум 30 часа лекции и 30 часа практически упражнения и се явява естествено продължение на дисциплината “Софтуерни технологии” от бакалавърския курс по същата специалност. Тежестта на обучението по ССТ пада върху обектно-ориентираното проектиране на софтуерни системи, посредством използването на съвременни средства за автоматизация.

В таблица 1 са дадени основните теми по дисциплината, които са свързани с изучаването на UML, заедно с предвидения им хорариум. От нея се вижда, че само една трета от времето в лекциите е посветено на тази тематика. В останалите лекции се разглеждат общи теми от софтуерните технологии. Това се налага, защото все още в магистратурата се записват студенти, които в бакалавърската си степен не са изучавали основите на софтуерните технологии.

В практическите упражнения се обръща внимание на работата в екип. В рамките на самостоятелна практическа работа, екипите разработват конкретни софтуерни задачи, използвайки автоматизирани среди за проектиране. След завършване на обучението по дисциплината, студентите трябва да са в състояние да проектират, разработват, и документират цялостни софтуерни системи и да работят в екип.

Таблица 1

№	Тема	часа	
		л	пу
1	Разработване на бизнес модел на система.	2	4
2	Въведение в UML.	2	2
3	Запознаване със средата на Rational Rose.	2	2
4	Създаване на Use Case диаграми.	-	2
5	Създаване на Interaction диаграми – Sequence и Collaboration диаграми.	-	4
6	Създаване на Class диаграма.	-	2
7	Създаване на State и Activity диаграми.	-	4
8	Създаване на Component диаграма.	-	2
9	Създаване на Deployment диаграма.	-	2
10	web публикуване на модела на системата.	2	2
11	Създаване на документация.	2	4

Дисциплината е изучавана от три випуска магистри по информатика. Натрупаният опит и наблюдения позволяват на автора да анализира обучението и да го сравни с това в други висши учебни заведения.

Таблица 2

№	ТЕМА	часа	
		л	пу
1	Увод в моделирането. Обектно-ориентиран подход. Визуално моделиране. UML – история и цели. Разрези на модела.	2	2
2	Процес на софтуерна разработка. Основни моменти и проблеми. Примери. Инструментални средства.	2	2
3	Use cases – типични случаи на употреба на софтуерната система. Actors – потребителски роли. Use-case диаграми.	2	2
4	Класове и обекти. Диаграми на класовете.	2	2
5	Взаимодействия между обекти. Диаграми на последователността на изпълнение и съвместната работа на обектите – Sequence и Collaboration диаграми.	2	2
6	Поведение и структура. Диаграми на активностите. Диаграми на състоянието. Диаграми на пакетите. Диаграми на внедряването. Компоненти.	2	2
7	Механизми за разширение на UML. Ограничения, свойства и стереотипи.	2	2
8	Проектиране на системната архитектура.	2	2
9	Жизнен цикъл на софтуерната разработка.	6	6
10	Унифициран процес на разработка.	2	2
11	Управление на UML проект	2	2
12	Комплементарни (допълнителни) въпроси	4	4

Проблем номер едно е, че все още в магистратурата се записват студенти, които в бакалавърската си степен не са изучавали в достатъчна степен или въобще не са изучавали основите на софтуерните технологии. Това налага да се отдели голяма част от времето за изучаване на неща, с които останалите студенти са наясно. Друг проблем се явява недобрата езикова подготовка на студентите. При изучаването на UML и бизнес моделирането, както и на всичко свързано със ССТ, задължително е студентите да следят най-новите публикации в интернет пространството, повечето от които са на сравнително сложен английски език със специфична терминология. Самостоятелната работа с развойната среда Rational Rose и нейните help-ове изисква също добър английски. Тази среда е избрана, заради реалното ѝ използване от водещи софтуерни фирми и поддържането на актуалните нотации на UML.

UML се изучава във всички ВУЗ-ове, които подготвят ИТ специалисти, но по различно време на обучението, с различен хорариум и учебни програми.

За сравнение - в Софийския университет тематиката, посветена на UML е застъпена в дисциплината „Обектно-ориентиран анализ и дизайн на езика UML” със същия хорариум, както и в РУ. Тази дисциплина се предлага като избираема в бакалавърската степен за всички информатични специалности на Факултета по математика и информатика [6]. Учебната ѝ програма е дадена в таблица 2. От нея се вижда, че лекционното време изцяло е посветено на изучаването на UML, което е голямо предимство. Тематиката на упражненията не се различава съществено. В упражненията на СУ е предвидено по-малко време за практически занимания по създаването на различните видове диаграми, за сметка на теми с номера 8, 9 и 12, които имат по-теоретичен характер. Тези теми са важни и фундаментални и по тази причина са изнесени в бакалавърския курс по СТ за специалност Информатика на РУ и в магистърското обучение се разчита на знанията и уменията, които студентите вече са придобили. От тук става ясно, че няма съществени различия в обема и начина на изучаване на UML в двете учебни заведения.

ЗАКЛЮЧЕНИЕ

Направеният преглед на изучаването на UML в дисциплината ССТ от магистърския курс по Информатика позволява да се направят следните *изводи и препоръки*:

- Предлаганата тематика е актуална и полезна.
- Изучаването на UML съответства на съвременното ниво в достатъчна степен, но времето, което се отделя в лекциите не е достатъчно.
- Предварителните познания на студентите в областта на създаването на бизнес модели все още са на ниско ниво.
- Езиковата подготовка на студентите не е на достатъчно добро ниво.
- Да се засили изучаването на създаването на бизнес модели, които са в основата на проектирането на софтуерни системи.
- Да се приучават студентите да използват on-line материали, за да следят новостите и тенденциите при софтуерното производство.

ЛИТЕРАТУРА

- [1] Стригулев Н., в-к Капитал, 29 Май 2004
- [2] Фаулър М., UML. Основи: кратко ръководство за стандартния език за обектно моделиране, СофтПрес ООД, 2004
- [3] Boch G., Runbaugh J., Jacobson I., Unified Modeling Language User Guide, The Addison - Wesley. 1998
- [4] Pender T.A., UML Weekend Crach Course, Wiley Publishing, Inc., 2002
- [5] Quatrani T., Visual Modeling with Rational Rose 2000 and UML, Addison – Wesley, 2001
- [6] <http://www.fmi.uni-sofia.bg/education> - сайт на ФМИ на СУ "Кл. Охридски"
- [7] www.rational.com – официален сайт на фирмата Rational Corporation
- [8] www.uml.com – официален сайт за UML на OMG Group

За контакти:

гл.ас. Пламенка Тодорова Христова, катедра “Информатика и информационни технологии”, Русенски университет “Ангел Кънчев”, Тел.: 082/ 888 326, e-mail: ptx@ami.ru.acad.bg.

Докладът е рецензиран.