

## К вопросу о связи объектно-ориентированных метрик дизайна и качественных характеристик программных систем

О. А. Загривый, И. Т. Зарецкая

*Харьковский национальный университет им. В.Н. Каразина, Украина*

The results of the empirical research on advisability of using object oriented metrics at the design phase of the software development process are presented in the paper. The maintainability of the object oriented system and adaptability to the requirements changes was analyzed by object oriented metrics. Different models and implementations of the learning management system were used for the study.

### 1. Общая постановка задачи и её актуальность

Вопросы влияния качества дизайна на стоимость разработки, возможность повторного использования, сопровождаемость и адаптивность программного обеспечения активно обсуждаются в литературе. Имеется ряд исследований и результатов, основанных на анализе эмпирических данных, подтверждающих связь между характеристиками дизайна и результирующими свойствами объектно-ориентированных систем [1 - 6]. В этих работах приведены данные, полученные в результате оценивания качества дизайна конкретных проектов по разработке программных систем с помощью объектно-ориентированных метрик [6 - 9]. Проанализирована связь этих данных с такими характеристиками систем, как стоимость разработки, устойчивость к сбоям и т. п. Результаты исследований допускают разные интерпретации, но свидетельствуют о целесообразности использования объектно-ориентированных метрик на этапе проектирования для адекватного предсказания указанных выше качественных характеристик системы.

В данной статье ставилась задача анализа данных аналогичного исследования, проведенного при разработке и сопровождении системы управления учебным процессом. Исследовалось свойство гибкости системы при изменении требований. В статье приводятся результаты исследования, основанные на сравнении различных вариантов дизайна такой системы. Один вариант – двухуровневый, с бизнес – логикой жестко встроенной в уровень представления; второй – трехуровневый, основанный на использовании каркаса и набора его расширений, полностью инкапсулирующих бизнес – логику системы. Для оценки дизайна использовался набор объектно-ориентированных метрик, наиболее полно, на взгляд авторов, отражающий требуемые в результате свойства системы: расширяемость и легкую адаптацию к изменению требований.

### 2. Объекты экспериментов

Для проведения исследования использовались уже существующая система управления учебным процессом DOWS (Dean's Office Work Station), реализованная как монолитный блок, жестко привязанный к специфике

определенного учебного процесса, и разработанные в процессе исследования базовый каркас системы, его расширение, также представляющее собой каркас, и две его реализации, приспособленные к определенным требованиям.

Базовый каркас системы управления учебным процессом дает абстрактное представление о структуре учебного процесса. Он легко адаптируем к специфике любого учебного процесса и учебного заведения. На основе каркаса был построен расширенный каркас, предназначенный для управления учебным процессом в высших учебных заведениях. Для исследования целесообразности применения каркаса при построении моделей систем управления учебным процессом на основе расширенного каркаса были разработаны дизайны систем, соответствующие специфике учебного процесса в университете на данный момент и специфике Болонского учебного процесса. Для сравнения полученных дизайнов был проведен обратный инжиниринг системы DOWS (Dean's Office Work Station).

Три модели были оценены с помощью следующих объектно-ориентированных метрик [8 – 13]:

- Weighted Methods per Class (WMC);
- Number of Children (NOC);
- Coupling Between Object Classes (CBO);
- Lack of Cohesion in Methods (LCOM);
- Response for a Class (RFC);
- Instability (I).

Полученные оценки были сопоставлены с вычисленными по модели, приведенной в [14], трудозатратами на разработку соответствующих систем.

Результаты показали, что выбранные объектно-ориентированные метрики достаточно адекватно отражают влияние характеристик дизайна на гибкость системы при изменении требований. «Старая» система (DOWS) практически не расширяема, и любое изменение в требованиях повлечет ее полное перепроектирование, тогда как системы, построенные на базе каркаса, сводят к минимуму количество и сложность действий, необходимых для приспособления к новым требованиям.

### **3. Базовый каркас системы управления учебным процессом**

Общая модель системы управления учебным процессом приведена на рис.1. При разработке этого каркаса естественно предполагалось, что в любом образовательном процессе присутствуют «обучаемый» и «обучающий». На модели им соответствуют классы «aTrainee» и «aTeachingStaff». Учебный процесс всегда координируется некоторым «органом управления». Например, в высших учебных заведениях таким органом является деканат. На модели «органу управления» соответствует класс «aAuthority». Обучение ведется по предметам (модулям, дисциплинам). Предмету соответствует класс «aModule». Обучение ведется в соответствии с некоторым планом. На модели ему соответствует класс «aCurriculum». Любой учебный план состоит из записей. В модели записи соответствует класс «aEntry». Процесс обучения должен контролироваться, поэтому для каждого предмета нужно иметь список отчетностей (видов контроля), которые необходимо сдать. Для моделирования

этого аспекта был добавлен класс «aTeachingControl». Все классы каркаса абстрактные. Для работы с подклассами в них включены фабричные методы [7].

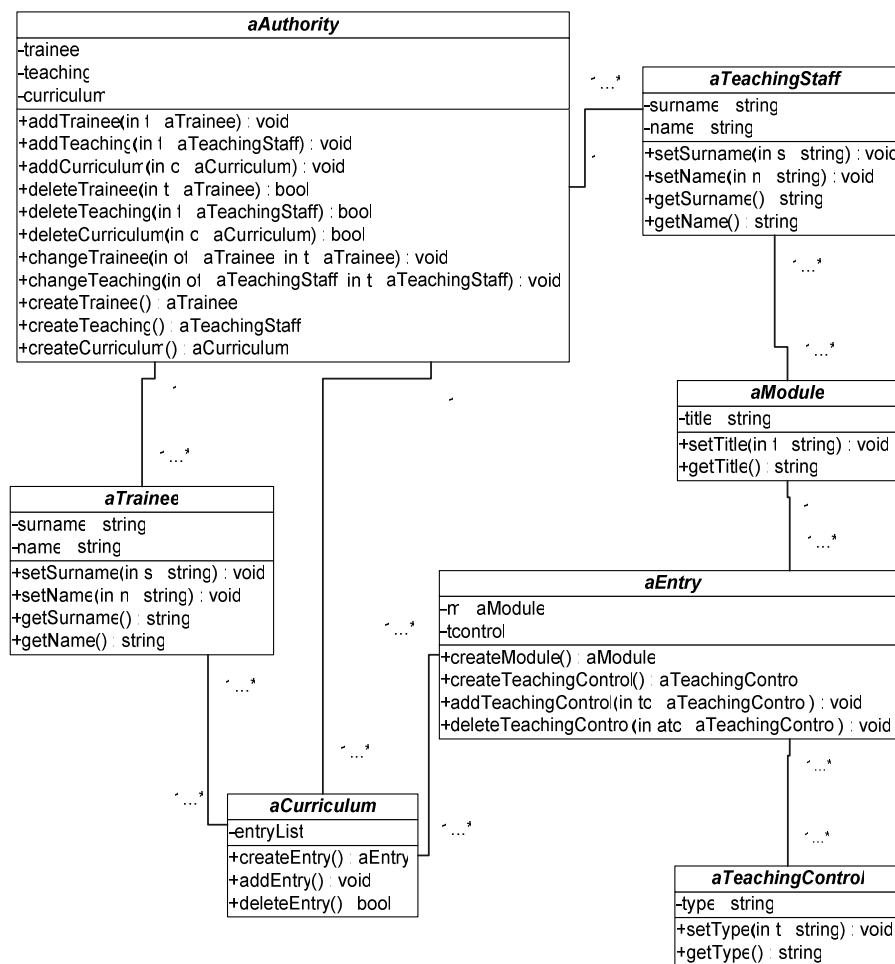


Рис. 1. Базовый каркас системы управления учебным процессом

#### 4. Расширенный каркас системы управления учебным процессом

Расширенный каркас (рис. 2) адаптирует базовый к требованиям учебного процесса в высших учебных заведениях:

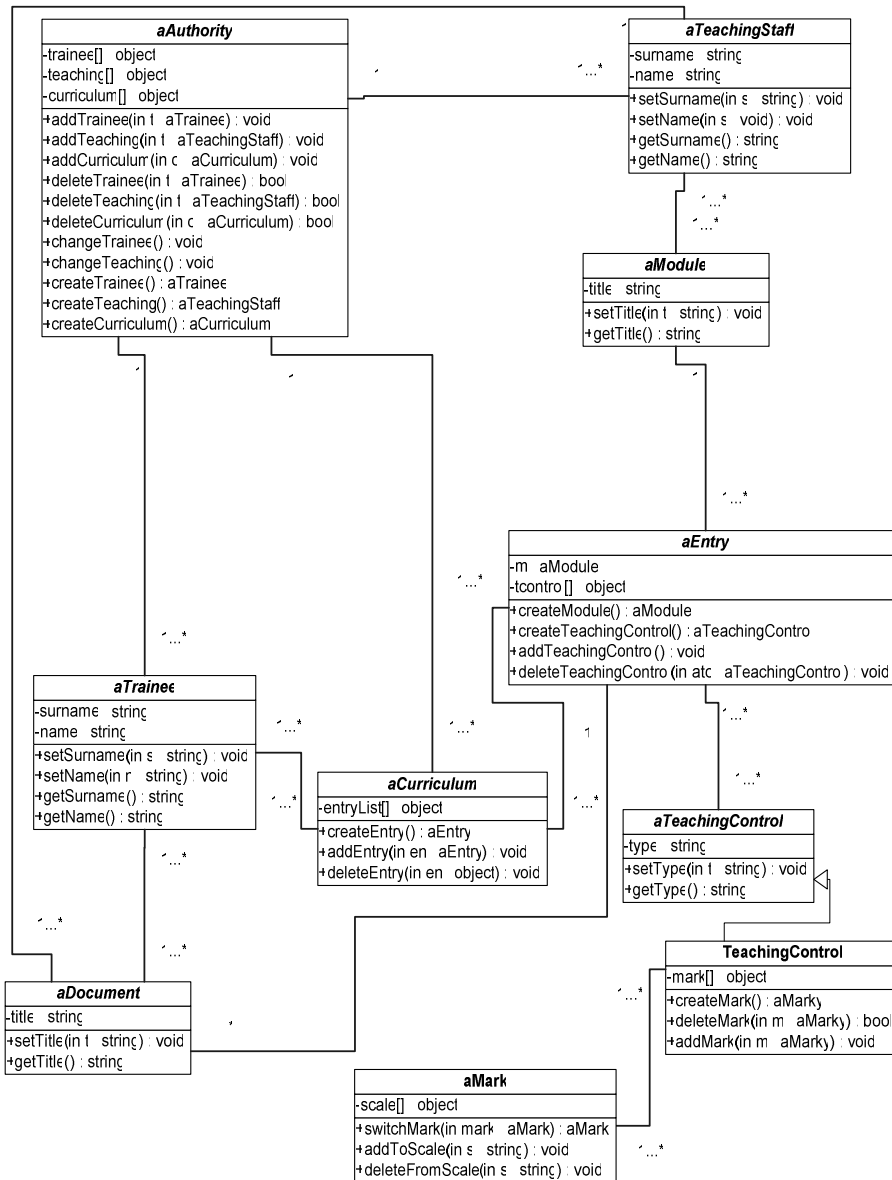


Рис. 2. Расширенный каркас системы управления учебным процессом

Основные изменения связаны с необходимостью учета успеваемости с возможностью выставления оценок в разных системах оценивания. В связи с этим были добавлены класс «TeachingControl», наследующий классу «aTeachingControl» и класс «aMark» для моделирования системы оценивания.

Для учета разного рода документации (например, ведомости, справки и т.д.) в модель был добавлен класс «aDocument».

### 5. Старая и новая модели системы

Под старой моделью системы подразумевается модель, которая была получена посредством обратного инжиниринга системы DOWS (Dean's Office Work Station).

Новая модель системы создана на основе расширенного каркаса. Рассмотрим классы, которые были добавлены. Конкретным «обучаемым» (студент) и «обучающим» (преподаватель) в модели соответствуют классы «Student» и «Lecturer» (рис.3). Класс «Student», наследует классу «aTrainee», а класс «Lecturer» – классу «aTeachingStaff».

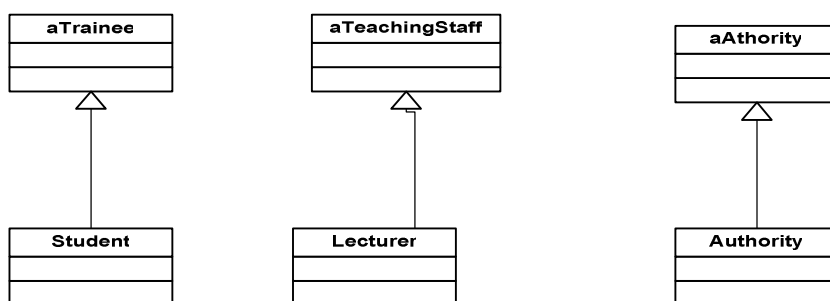


Рис. 3. Фрагмент 1 новой модели системы

Конкретный орган управления – деканат – представляется классом «Authority», наследующим классу «aAuthority» (рис. 3) .

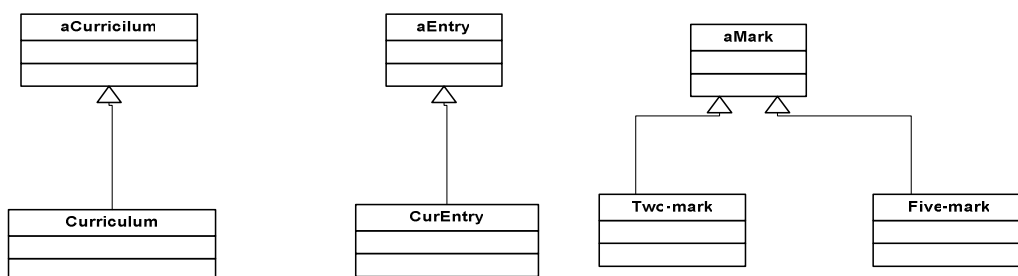


Рис. 4. Фрагмент 2 новой модели системы

Для моделирования учебных планов введен класс «Curriculum», который наследует классу «aCurriculum». Он отличается тем, что будет оперировать записями вполне определенного типа, моделируемого классом «CurEntry» (рис. 4) - наследником класса «aEntry». Введены также классы для моделирования

конкретных систем оценивания: зачет \ не зачет - класс «Two-mark» и пятибалльной системы - класс «Five-mark» . Оба класса наследуют классу «aMark» (рис. 4).

Для моделирования учебных групп добавлен класс «Group» (рис. 5). Класс «EduProgram» (рис. 5) моделирует программу образования.



Рис. 5. Фрагмент 3 новой модели системы

### 6. Изменение новой модели (Болонская модель)

При переходе к Болонской системе образования появляются два типа учебных планов: абстрактные и индивидуальные. В абстрактных учебных планах хранится перечень всех дисциплин, которые студент должен изучить в соответствии с выбранной программой обучения. Индивидуальный учебный план составляется для конкретного студента на семестр и содержит перечень обязательных дисциплин, которые необходимо пройти в этом семестре и перечень необязательных дисциплин, которые студент выбрал перед началом семестра. Поэтому вместо класса «Curriculum», который был в новой модели, добавлены два класса «AbsCurriculum» и «IndCurriculum» (рис. 6): первый - для моделирования абстрактных учебных планов, а второй - для индивидуальных. Оба класса наследуют классу «aCurriculum» и отличаются только типом записей. В случае абстрактного учебного плана используются записи типа «AbsEntry», а в случае индивидуального - типа «IndEntry» (рис. 6). Оба класса наследуют классу «aEntry».

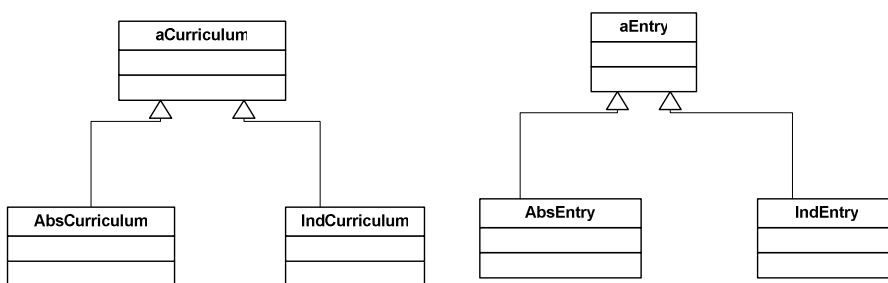


Рис. 6 Фрагмент Болонской модели системы

### 7. Оценка трудозатрат

Оценка трудозатрат на разработку соответствующих систем проводилась с помощью следующей формулы [14]:

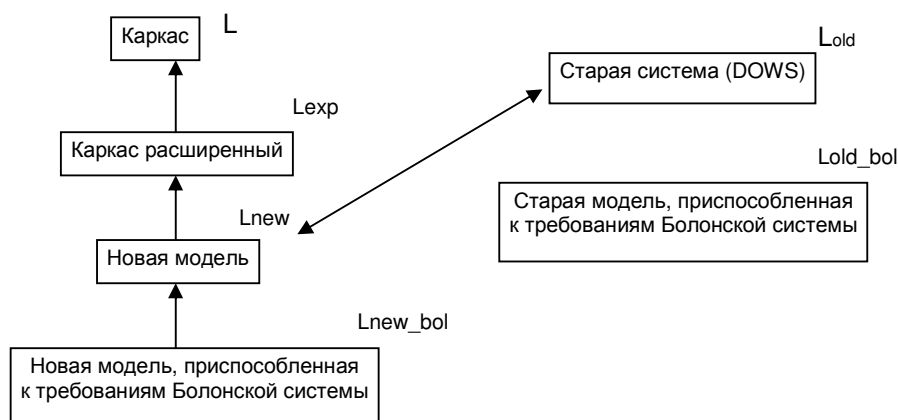
$$L = 2.4 * ((ControlClasses * 0.094) + (InterfaceClasses * 0.059) + (OtherClasses * 0.047) + (Methods * 0.024) + (Tables * 0.282))$$

Для всех моделей переменные InterfaceClasses и Tables полагались равными нулю, поскольку в моделях, построенных на базе расширенного каркаса, уровни данных и представления не учитывались. В модели старой системы (DOWS) уровень представления и бизнес-логика практически не разделены. Поэтому в старой модели классы графического интерфейса не учитывались.

Получены следующие результаты.

Базовый каркас –  $L = 2.633$ , расширенный каркас –  $L_{exp} = 0.794$ , новая модель –  $L_{new} = 9.821$ , Болонская модель –  $L_{new\_bol} = 1.025$ , старая система (DOWS) –  $L_{old} = 46.618$ , старая модель, приспособленная к требованиям Болонской системы, –  $L_{old\_bol} \geq 46.618$ .

Развитие систем можно представить в виде схемы, приведенной на рис. 7.



Количество новых классов:

$$(L_{exp})10 \Rightarrow (L_{new})16 \Rightarrow (L_{new\_bol})3$$

$$(L_{old})41 \Rightarrow (L_{old\_bol})41 + n$$

Количество новых методов:

$$(L_{exp})40 \Rightarrow (L_{new})126 \Rightarrow (L_{new\_bol})12$$

$$(L_{old\_bol})707 \Rightarrow (L_{old\_bol})707 + n$$

Рис. 7 Схема развития системы и оценки трудозатрат

## 8. Сравнительный анализ моделей

Для каждой из моделей были посчитаны приведенные выше метрики и вычислены стандартные статистические характеристики. Результаты сведены в таблицу 1.

Табл.1 Сводная таблица результатов исследований

|      |        | Lexp=<br>0.794 | Lnew=<br>9.821 | Lnew_bol=1.025 | Lold=<br>46.618 |
|------|--------|----------------|----------------|----------------|-----------------|
| WMC  | min    | 2              | 2              | 2              | 0               |
|      | max    | 11             | 43             | 43             | 292             |
|      | StdDev | 2.57           | 8.69           | 8.32           | 46.04           |
|      | Mean   | 4.2            | 7              | 6.69           | 17.24           |
| NOC  | min    | 0              | 0              | 0              | 0               |
|      | max    | 1              | 2              | 3              | 0               |
|      | StdDev | 0.32           | 0.65           | 0.83           | 0               |
|      | Mean   | 0.1            | 0.46           | 0.97           | 0               |
| CBO  | min    | 0              | 0              | 0              | 0               |
|      | max    | 3              | 4              | 5              | 29              |
|      | StdDev | 1.06           | 1.29           | 1.35           | 4.38            |
|      | Mean   | 0.7            | 0.92           | 0.96           | 2.93            |
| LCOM | min    | 0              | 0              | 0              | 0               |
|      | max    | 32             | 840            | 840            | 42486           |
|      | StdDev | 10.01          | 164.64         | 156.19         | 666597          |
|      | Mean   | 3.6            | 43.81          | 37.93          | 1186.41         |
| RFC  | min    | 2              | 3              | 3              | 0               |
|      | max    | 11             | 43             | 43             | 707             |
|      | StdDev | 2.57           | 8.69           | 8.3            | 10963           |
|      | Mean   | 4.2            | 7              | 6.13           | 27.71           |
| I    | min    | 0              | 0              | 0              | 1               |
|      | max    | 0.33           | 0.33           | 0.33           | 1               |

Анализ этих данных показывает, что каркас состоит из максимально стабильных категорий (метрика Instability для любой категории из каркаса равна нулю). В расширенном каркасе появляется реализация класса aTeachingControl и вместе с ней появляются связи между некоторыми категориями. Поэтому метрика Instability немного изменяет свое значение для данной категории. Остальные категории остаются максимально стабильными. В модельных реализациях, построенных на основе расширенного каркаса, появляется еще больше зависимостей между категориями. Поэтому значение метрики Instability для полученных категорий находится в пределах [0, 0.33]. Для старой модели метрика Instability равна единице у всех категорий, т. е.. все категории старой системы максимально нестабильны.

Проведенные исследования показывают, что старая система практически не расширяема и любое изменение в требованиях будет приводить к полному перепроектированию соответствующей части системы. Классы же систем, построенные на базе каркаса, имеют высокий уровень сцепления и низкий уровень связывания. Категории этих систем стабильны. Они приспособляются к новым требованиям с минимальным количеством усилий.



## 9. Выводы

Результаты исследования подтверждают существенное влияние дизайна на качественные характеристики системы. В применении к классу систем, аналогичных рассмотренной, результаты, несомненно, свидетельствуют в пользу трехуровневой архитектуры и решений, основанных на каркасах.

## ЛИТЕРАТУРА

1. L. Briand, W. Melo, J. Wüst. Assessing the Applicability of Fault-Proneness Models Across Object-Oriented Software Projects//IEEE Transactions on Software Engineering, 28 (7). – 2002. – P. 706-720,
2. L. Briand, J. Wüst. Empirical Studies of Quality Models in Object-Oriented Systems// Advances in Computers Vol. 59. – 2002. – P. 97-166,
3. L. Briand, J. Wüst, H. Lounis. Replicated Case Studies for Investigating Quality Factors in Object-Oriented Designs//Empirical Software Engineering: An International Journal, Vol 6, No 1. – 2001. – P. 11-58,
4. L. Briand, J. Wüst The Impact of Design Properties on Development Cost in Object-Oriented Systems// IEEE Transactions on Software Engineering, 27 (11). – 2001. – P. 963-986.
5. L. Briand, J. Wüst, J. Daly, V. Porter A Comprehensive Empirical Validation of Product Measures for Object-Oriented Systems//Journal of Systems and Software 51. – 2000. - P. 245-273.
6. Lionel C. Briand, Jirgen Wist, John W. Daly, D. Victor Porter. Exploring the Relationship between Design Measurers and Software Quality in Object-Oriented Systems// Journal of Systems and Software, Volume 51 , Issue 3. – 2000, - P. 245 – 273.
7. Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влссидес Приемы объектно-ориентированного проектирования. Паттерны проектирования. - СПб: «Питер». – 2001.- 368с.
8. Dr. Linda H. Rosenberg . Applying and Interpreting Object Oriented Metrics// Software Technology Conference, Utah. – April 1998.
9. Robert Martin. Object Oriented Design Quality Metrics. An Analysis of Dependencies.– [Электронный ресурс].– Режим доступа: <http://www.objectmentor.com/resources/articles/oodmetrc.pdf>
10. Shyam R. Chidamber, Chris F. Kemerer. A Metrics Suite for Object Oriented Design// IEEE Transactions on Software Engineering, June. – 1994. – P. 476-492.
11. Rosenberg, Linda H.. Metrics for Object Oriented Environments// EFAITP/AIE Third Annual Software Metrics Conference.–December, 1997
12. The UML design quality metrics tool. – [Электронный ресурс]. – Режим доступа: <http://www.sdmetrics.com/>
13. Bertrand Meyer. The role of object-oriented metrics// IEEE Computer Society Press , Computer, Volume 31 , Issue 11.– 1998. - P. 123 - 125
14. William Roetzheim . Estimating Effort Using Use-Case and UML Class-Method Points .– UML&Design World 2005.–Austin, Texas.–June, 2005