

Проанализированы статистические структуры текстов поэзии Дж. Г. Байрона и Т. Мура. Доказано наличие в текстах общих элементов, обусловленных принадлежностью к одному историческому периоду и литературному направлению. Сделана попытка определить статистическим методом действие фактора манеры авторского повествования в сравниваемых текстах на фонологическом уровне. Степень действия упомянутого фактора определена по количеству групп согласных фонем, по которым определены существенные расхождения между сравниваемыми текстами. На основании полученных результатов создана модель, которая представляет действие авторского фактора в подстиле поэзии художественного стиля английского языка.

Ключевые слова: средняя частота групп согласных фонем, нормальное распределение, критерий Стьюдента, статистическая структура.

Khomyska I.Yu., Teslyuk V.M. Statistical Analysis of English Poetical Texts

The statistical structures of the texts of poetry by G.G. Byron and T. Moore have been analyzed. The texts are proved to have common elements caused by belonging to one and the same historic period and literary trend. An attempt is made to determine by statistical method the effect of the factor of the author's style in the compared texts of poetry on the phonological level. The degree of the effect of the mentioned factor is established by a number of groups of consonant phonemes by which the essential difference is determined between the compared texts. On the basis of the obtained results a model representing the author's factor effect in the substyle of poetry of the belles-lettres style of the English language is built.

Key words: mean frequency of occurrence of groups of consonant phonemes, normal distribution, Student's t-test, statistical structure.

УДК 004;621.398;681.5

Доц. Г.С. Погромська, канд. пед. наук –
Миколайський НУ ім. В.О. Сухомлинського

РАЦІОНАЛІЗАЦІЯ ВИЗНАЧЕННЯ ОБ'ЄКТНО-ОРІЄНТОВАНОГО CASE-ЗАСОБУ В ЖИТТЄВОМУ ЦИКЛІ ПРОГРАМНОЇ СИСТЕМИ

Запропоновано концепцію визначення оцінок CASE-засобів для проектування та розроблення програмних систем на основі застосування однопараметричної моделі сучасної теорії вимірювань IRT. Розглянуто характерні особливості CASE-засобів та їх можливостей для проектування програмних систем. Визначено основні ознаки, за якими можна проводити оцінювання та вибір. Розроблено і рекомендовано застосування методу статистичного вимірювання на основі використання однопараметричної моделі Раша для визначення оцінок CASE-засобів, які характеризуються великою кількістю різноманітних ознак, більшість з яких мають якісний характер.

Ключові слова: CASE-засоби, життєвий цикл програми, програмна система, вибір CASE-засобу, модель Раша.

Актуальність дослідження. Комп'ютерні програми є основним засобом праці для великої кількості користувачів персональних комп'ютерів. Процес вирішення безлічі професійних завдань змінився, і комп'ютер став його невід'ємною частиною. Діяльність, яка раніше виконувалася в реальному середовищі шляхом переміщення та перетворення реальних об'єктів, з впровадженням комп'ютерів отримала форму віртуального відображення. Багато ручних операцій автоматизувалися, а сам процес вирішення професійних завдань перейшов в діалог "людина – комп'ютер".

На сьогодні спостерігається сплеск інтересу до проблем раціоналізації у зв'язку з розвитком теорії організації, теорії систем, теорії прийняття рішень. Формулюються принципи раціоналізації, йде пошук варіантів включення мето-

дів раціоналізації в систему організаційних перетворень. Тому не дивно, що ці питання постали й з точки зору процесів розроблення програмних систем.

У процесі розроблення програмного забезпечення зазвичай використовуються моделі, а також інструментальні засоби для підтримки та уніфікації моделювання. Але створення програмної системи – це не тільки побудова моделі. Чим складніший й ширший проект, тим більшою мірою він вимагає інструментальної підтримки всіх етапів життєвого циклу програмного забезпечення. Раціоналізація покликана забезпечити узгоджену та ефективну роботу всіх елементів і прогресивний розвиток всієї системи.

CASE-засоби (*Computer-Aided Software Engineering*) є інструментарієм раціоналізації процесів розроблення програмних систем. Зазвичай до CASE-засобів відносять будь-який програмний засіб, що автоматизує один процес або сукупність процесів життєвого циклу програмного забезпечення [1]. Використання CASE-засобів дає змогу автоматизувати моделювання та проектування програмних компонент, надає можливості документування та опису програмної системи, повторно використовувати ефективні рішення, а також автоматично генерувати програмний код.

Вибір CASE-засобу у створенні програмної системи є важливим з погляду ефективності та надійності роботи групи проектувальників та програмістів. Це зумовлює актуальність дослідження питань вибору CASE-засобів та раціоналізації організаційно-виробничих процесів розроблення програмних систем.

Аналіз останніх досліджень і публікацій. Теоретичний доробок у сфері програмної інженерії достатньо значний. Питання життєвого циклу програмних продуктів розглянуто у [2-7]. Автори української теоретичної школи програмування П.І. Андон, Д.Б. Буй, В.М. Глушков, С.С. Гороховський, К.М. Лаврішева, О. Цейтлін та ін. [2-5] розглядають різноманітні підходи до вирішення проблем програмування. Зокрема, можна підкреслити особливу увагу до CASE-засобів підтримки процесів життєвого циклу програмної системи [2, 8].

Варто зауважити, що у більшості робіт, які розглядають питання щодо CASE-засобів, описано переваги та можливості певних інструментів [8-10]. У багатьох роботах, наприклад в [11, 12], висвітлено особливості CASE-засобів як особливих типів програмних систем. У них основну увагу приділено опису узагальнюючих характеристик певного типу інструментів. Так, за типом технології, яка використовується для створення програмних систем, виділяють об'єктно-орієнтовані та структурні засоби [8]. Крім того, деякі автори розглядають інтегровані CASE-засоби, які забезпечують підтримку всіх етапів життєвого циклу.

Аналіз публікацій показав, що достатньо висвітлено питання щодо використання CASE-засобів у життєвому циклі програмної системи, натомість проблеми вибору певного CASE-засобу залишаються відкритими. Це зумовило напрям досліджень у цій роботі.

Постановка завдання. Зважаючи на викладене вище, метою роботи є розроблення методологічної основи для вирішення задач із вибору CASE-засобів як напрямку раціоналізації процесів підтримки життєвого циклу програмних систем і підвищення їх ефективності.

Виклад основного матеріалу. CASE-технології утворюють певне середовище розроблення інформаційної або програмної системи та забезпечують процес розроблення складних програмних систем загалом: аналіз, формування вимог, проектування прикладного програмного забезпечення та баз даних, генерація коду, тестування, документування, забезпечення якості, конфігураційне керування та управління проектом, а також інші процеси [9].

Сучасні CASE-засоби мають такі характерні складові:

- методологія (*Method Diagrams*) – задає єдину графічну мову та правила роботи з нею;
- графічні редактори (*Graphic Editors*) – допомагають створювати діаграми;
- генератор – дає змогу за графічним поданням моделі згенерувати вихідний код для різних платформ;
- репозиторій – своєрідна база даних для збереження результатів роботи програмістів.

На цей час у технології розробки програмних систем існує два основних підходи, які відрізняються критеріями декомпозиції: функціонально-модульний (або структурний) та об'єктно-орієнтований [8-10].

Структурним аналізом прийнято називати метод дослідження статичних характеристик системи шляхом виділення в ній підсистем і елементів різного рівня ієрархії і визначення відношень і зв'язків між ними. Сутність структурного підходу до розроблення моделі полягає в поділі аналізованої системи на частини – "чорні ящики", та ієрархічній організації цих "чорних ящиків". Перевага оперування "чорними ящиками" полягає в тому, що немає потреби знати, як вони працюють, досить мати інформацію про їх входи і виходи, а також функції, які вони виконують.

У структурному аналізі і проектуванні бізнес-процесів використовують різні моделі, які описують [8, 9]:

- функціональну структуру системи;
- послідовність виконуваних дій;
- передачу інформації між функціональними процесами;
- відносини між даними.

Найбільш поширеними моделями перших трьох груп є:

- функціональна модель SADT (Structured Analysis and Design Technique);
- модель IDEF3 (Integration Definition for Function Modeling);
- DFD (Data Flow Diagram) – діаграма потоків даних.

Метод SADT – це сукупність правил і процедур, призначених для побудови функціональної моделі об'єкта будь-якої предметної області. Технологію SADT (перейменована в IDEF0) традиційно використовують для моделювання організаційних систем (бізнес-процесів). Достоїнствами застосування моделей SADT для опису бізнес-процесів є:

- повнота опису бізнес-процесу (управління, інформаційні та матеріальні потоки, зворотні зв'язки);
- жорсткі вимоги методу, що забезпечують отримання моделей стандартного виду;
- відповідність підходу до опису процесів стандартам ISO 9000.

IDEF є досить поширеним сімейством методів моделювання організаційних систем, на основі якого в різних країнах розроблено множини різних нормативних документів. На цей час до сімейства IDEF відносять такі методології [8]:

- IDEF0 – методологія функціонального моделювання, забезпечена наочною графічною мовою. Дає змогу подати модельовану систему у вигляді набору взаємозалежних функцій. Як правило, моделювання засобами IDEF0 є першим етапом вивчення системи;
- IDEF1 – методологія моделювання інформаційних потоків усередині системи, що дає змогу відображати та аналізувати їх структуру та взаємозв'язки;
- IDEF1X (*IDEF1 Extended*) – методологія моделювання, яку застосовують для побудови інформаційної моделі, що становить структуру інформації, необхідної для підтримки функцій виробничої системи або середовища. IDEF1X часто використовують для моделювання реляційних баз даних, що мають відношення до цієї системи;
- IDEF4 – методологія об'єктно-орієнтованого проектування. IDEF4 реалізує об'єктно-орієнтований аналіз великих систем, надаючи користувачеві графічну мову для зображення класів, діаграм успадкування, таксономії методів;
- IDEF5 – методологія онтологічного дослідження складних систем. Застосовуючи методологію IDEF5, онтологію системи можна описати за допомогою певного словника термінів і правил, на підставі яких можуть бути сформовані достовірні твердження про стан аналізованої системи в певний момент часу. На базі цих тверджень формуються висновки про подальший розвиток системи та проводиться її оптимізація.

Діаграми потоків даних DFD – це ієрархія функціональних процесів, пов'язаних потоками даних. Мета такого подання – продемонструвати, як кожен процес перетворює свої вхідні дані у вихідні, а також виявити відносини між цими процесами.

Модель системи визначається як ієрархія діаграм потоків даних, що описують асинхронний процес перетворення інформації від її введення в систему до видачі споживачеві. DFD з самого початку створювалися як засіб проектування інформаційних систем (тоді як SADT – як засіб моделювання систем взагалі) і мають більш багатий набір елементів, що адекватно відображають специфіку таких систем (наприклад, сховища даних є прообразами файлів або баз даних, зовнішні сутності відображають взаємодію модельованої системи із зовнішнім світом).

Розглянуті вище методи структурного аналізу приблизно однакові з точки зору можливостей виразних засобів моделювання, тому одним з основних критеріїв вибору того чи іншого методу є ступінь володіння ним з боку консультанта або аналітика. Концептуальною основою об'єктно-орієнтованого аналізу і проектування (ООАП) є об'єктна модель. Її основні принципи (абстрагування, інкапсуляція, модульність та ієрархія) і поняття (об'єкт, клас, атрибут, операція, інтерфейс та ін.) найбільш чітко сформульовані у Г. Буча [8, 10].

Більшість сучасних методів ООАП базуються на використанні мови UML [10]. Уніфікована мова моделювання UML (*Unified Modeling Language*) – це мова для визначення, подання, проектування та документування програмних систем, організаційно-економічних систем, технічних систем та інших систем різної природи. UML містить стандартний набір діаграм і нотацій найрізноманітніших видів. Їх можна розділити на ті, які моделюють статичну структуру системи (статичну модель), і ті, які моделюють динамічну структуру системи (динамічну модель). Статична модель фіксує сутності та структурні відносини

між ними. Динамічна модель відображає те, як сутності взаємодіють для формування необхідної поведінки системи.

До діаграм структури належать:

- діаграма класів (*class diagram*);
- діаграма складової структури (*composite structure diagram*);
- діаграма компонентів (*component diagram*);
- діаграма розгортання (*deployment diagram*);
- діаграма об'єктів (*object diagram*);
- діаграма пакетів (*package diagram*);
- діаграма профілів (*profile diagram*).

До діаграм поведінки належать:

- діаграма діяльності (*activity diagram*);
- діаграма прецедентів (*use-case diagram*);
- діаграма станів (скінченних автоматів) (*state machine diagram*);
- діаграми взаємодії (*interaction diagrams*);
- діаграма послідовності (*sequence diagram*);
- діаграма комунікації (*communication diagram*);
- діаграма огляду взаємодії (*interaction overview diagram*);
- діаграма синхронізації (*timing diagram*).

Методи структурного проектування допомагають спростити процес розроблення складних систем за рахунок використання алгоритмів як готових будівельних блоків. Аналогічно, методи об'єктно-орієнтованого проектування створені, щоб допомогти розробникам застосовувати потужні засоби об'єктного та об'єктно-орієнтованого програмування, яке використовує в ролі блоків класи та об'єкти. Об'єктно-орієнтоване проектування та об'єктно-орієнтоване програмування покращує можливості структурного проектування, концентруючи більше уваги на даних системи, а не на тому, що система робить. Цей підхід дає змогу створювати системи, які краще супроводжувати, вони більш гнучкі, більш стійкі та більш пристосовані до багатократного використання.

Поширеними представниками реалізації методології структурного моделювання є програмні засоби CA ERWin Process Modeler (*BPwin*) та CA ERWin Data Modeler (*ERWin*). Структурне проектування працює добре, тому що воно дає змогу одночасно зосереджуватися на меншій кількості деталей. Ця методика передбачає зменшення ступеня інтеграції на кожній з наступних стадій проекту. Такий тип проектування підходить краще за все до проблем, які мають чітко виражений ієрархічний характер.

Концепції об'єктно-орієнтованого підходу та розподілених обчислень стали базою для створення консорціуму Object Management Group (*OMG*), членами якого є понад 50 провідних комп'ютерних компаній (Sun, DEC, IBM, HP, Motorola та ін.) [8]. Основним напрямком діяльності консорціуму є розроблення специфікацій та стандартів для створення розподілених об'єктних систем у різних середовищах.

Існує кілька комерційних CASE-засобів, спрямованих на застосування об'єктно-орієнтованого підходу у проектуванні програмних систем (з підтримкою мови UML). Найбільш відомими є IBM Rational Rose, MS Visio. Засоби, які мають версії вільного розповсюдження – Borland Together, Enterprise Architect,

Poseidon, тощо. Також створено та знаходиться у вільному розповсюдженні багато засобів об'єктно-орієнтованого проектування з відкритим кодом для некомерційного використання, наприклад Torcased, Modelio, Visual Paradigm for UML (*Community Edition*), Dia.

Дамо стилю оцінку найбільш відомих продуктів та їх можливостей: Computer Associates, Rational Software, Oracle, Arena тощо. Rational Software – це лінійка продуктів, яка підтримує весь цикл проектування та створення програмного забезпечення. Починаючи від створення моделі на UML та закінчуючи різними видами тестування готового програмного забезпечення (функціональне, навантажувальне тощо). Виконується автоматичне документування усіх етапів розробки, автоматизований контроль задоволення технічним вимогам та управління конфігураціями та версіями, як компонентів, так і готового програмного забезпечення.

Computer Associates – три CASE-засоби, які добре інтегровані між собою та з продуктами інших компаній. BPwin – моделювання та опис бізнес-процесів, ERwin – моделювання даних (моделювання та реінжиніринг структури баз даних, сховищ даних), AllFusion Component Modeler (*Paradigm Plus*) – об'єктно-орієнтований CASE-засіб, який підтримує UML. У Computer Associates є також дуже потужний інструмент управління конфігураціями та версіями AllFusion Harvest Change Manager (*CCC/Harvest*).

У Oracle є потужний інтегрований продукт Oracle Development Suite, що включає попередні Designer та Developer, а також додаткові продукти. Він реалізує UML та IDEF, дає змогу моделювати бізнес-логіку, дані, інформаційні системи тощо. Цей CASE-засіб потребує спеціальних практичних навичок для засвоєння та експлуатації. Як засоби конфігураційного управління Oracle використовує Rational ClearCase.

Rockwell Software (*Systems Modeling*) пропонує засіб імітаційного моделювання – Arena. Якщо діяльність підприємства достатньою мірою формалізована, то за допомогою Arena можна вирішувати різноманітні бізнес-задачі типу "що буде, якщо" завдяки поданню бізнес-процесів у динаміці.

Класична постановка задачі розробки програмної системи – це спіральний цикл інтерактивної зміни етапів об'єктно-орієнтованого аналізу, проектування та реалізації. У реальній практиці у більшості випадків є передісторія у вигляді сукупності розроблених та впроваджених програм, які доцільно використовувати при розробці нової програмної системи. Процес проектування в такому разі засновано на реінжинірингу програмних кодів, за якого шляхом аналізу текстів програм відновлюється вихідна модель програмного середовища. Сучасні CASE-засоби підтримують процеси інжинірингу та автоматизованого реінжинірингу.

Можна виділити чотири основні компоненти, за якими можлива оцінка CASE-засобів: аналіз, проектування, розроблення та інфраструктура. Характеристики цих компонентів подані в таблиці.

Незважаючи на безліч переваг, можна зазначити деякі фактори, які ускладнюють визначення можливого ефекту від застосування CASE-засобів:

- широке розмаїття можливостей CASE-засобів;
- відносно невеликий час використання CASE-засобу у різних організаціях та нестача досвіду їх застосування;

- широке розмаїття в практиці впровадження різних організацій;
- відсутність детальних метрик та даних для вже виконаних поточних проектів;
- великий діапазон предметних галузей проектів;
- різна ступінь інтеграції CASE-засобів в різних проектах.

Табл. Стисла характеристика компонентів об'єктно-орієнтованого CASE-засобу

Аналіз		Проектування	Реалізація
Можливість додавати пояснювальні надписи до діаграм та у документацію	Можливість створювати різні уявлення та скривати зайві в певний час шари системи	Можливість переглядати та обирати елементи та бізнес-об'єкти для використання в системі	Можливість генерувати заготовки програмного коду на декількох об'єктно-орієнтованих мовах
Середовище для створення діаграм різнорідних моделей		Можливість створення користувацького інтерфейсу (підтримка OLE, ActiveX, Open Doc, HTML)	Можливість перевірки коду на синтетичну коректність
Підтримка різних нотацій	Можливість динамічного моделювання подій в системі	Можливість визначення бізнес-моделі та бізнес-правил	Можливість генерувати код для 4GL та клієнт-серверних продуктів (Power Builder, Forre, Visual Age, Visual Works)
Можливість генерації документації для друку	Можливість динамічної корекції однієї діаграми з іншою	Можливість зв'язку з об'єктно-орієнтованими базами даних та розподіленими модулями (підтримка COBRA, DCOM, HOP, HTML)	
Інфраструктура			
Контроль версій. Блокування та узгодження частин системи при груповій розробці		Репозиторій	Можливість реінжинірингу програмного коду, 4GL, клієнт-серверних систем у діаграмах моделей

Навколо визначення ефективності використання CASE-технологій існує дві точки зору: перша, що реальна вигода від застосування деяких типів CASE-засобів може бути отримана тільки після одного-двох років досвіду їх застосування та друга, що позитивний вплив може реально проявитися у фазі експлуатації життєвого циклу програмної системи, коли технологічні покращення призведуть до зниження експлуатаційних затрат.

Таким чином, можна виділити основні критерії оцінювання та вибору CASE-засобів:

- 1) функціональність;
- 2) надійність;
- 3) супровід;
- 4) здатність до перенесення.

Крім цього, як критерії варто враховувати також вартість, витрати і ефект впровадження.

Велика кількість критеріїв оцінки, якісний характер та складності збирання необхідних даних, зумовлюють потребу в розробленні методологічного підходу до вирішення задачі оцінювання та вибору CASE-засобів з метою під-

вищення ефективності виробничих процесів підтримки життєвого циклу програмних систем.

Проведений аналіз існуючих підходів до оцінювання складних багатооб'єктових об'єктів дає змогу виділити два основних напрямки: експертні методи та статистичні моделі оцінювання [13]. Ураховуючи складність використання експертних методів на практиці, запропоновано використання статистичної процедури оцінки CASE-засобів з метою подальшого вибору певного інструменту. Традиційні методи збирання даних орієнтовано на анкетування, опитування або обробку офіційних звітів. На практиці реалізація таких методів має певні недоліки, які пов'язано з організацією процедури збирання даних.

Оцінка і накопичення відповідних даних може виконуватися такими способами:

- аналіз CASE-засобів та документації постачальника;
- опитування реальних користувачів;
- аналіз результатів проектів, які використовували дані CASE-засобу;
- перегляд демонстрацій і опитування демонстраторів;
- виконання тестових прикладів;
- застосування CASE-засобів у пілотних проектах;
- аналіз будь-яких доступних результатів попередніх оцінок.

З іншого боку, автоматизація процесів збирання даних дасть змогу організувати збір даних щодо використання різноманітних CASE-засобів з урахуванням різних точок зору. У цьому разі постає питання подальшої оброблення статистичної інформації. З цієї метою запропоновано використання результатів, які отримано в рамках сучасної теорії вимірювань [14, 15]. Під сучасною теорією вимірювань розуміють теорію IRT (Item Response Theory), що призначена для оцінювання латентних параметрів шляхом застосування математично-статистичних моделей вимірювання. На відміну від більшості існуючих підходів, для IRT характерно прагнення до фундаментального теоретичного підходу та розв'язання практичних задач. До найбільших значущих переваг IRT належать такі:

1. IRT перетворює вимірювання, виконані в дихотомічних і порядкових шкалах, у лінійні вимірювання, внаслідок якісні дані аналізуються за допомогою кількісних методів.
2. Міра вимірювання параметрів моделей IRT є лінійною, що дає змогу використовувати широкий спектр статистичних процедур для аналізу результатів вимірювань.
3. Оцінка складності ознак, за якими проводиться вимірювання, не залежить від вибірки, на якій вона була отримана.
4. Оцінка показника не залежить від використовуваного набору ознак.
5. Неповнота даних (пропуск деяких комбінацій об'єкт-ознака) не є критичним.

Найвідомішою моделлю вимірювання, яку розроблено в IRT, є модель Раша [14]. У ній встановлюється зв'язок між двома множинами значень латентних змінних. Першу множину становлять значення латентної змінної, що визначає оцінку показника θ_i , де i – номер об'єкта, що змінюється в інтервалі від 1 до N (наприклад, N – кількість осіб, які брали участь в опитуванні). Другу множину становлять значення латентної змінної, що характеризує складність j -то

ознаки β_j . Індекс j змінюється в межах від 1 до M , де M – кількість ознак, за якими проводиться оцінювання.

Для кожної ознаки j модель Раша виражає ймовірність відповіді x_j як функцію від θ і параметра β_j , що характеризує складність ознаки: $f(x_j, \theta, \beta_j)$.

$$P(x_{ij} = 1 | \theta_j, \beta_i) = f(x_j, \theta_j, \beta_i) = \psi(\theta_j - \beta_i),$$

де: x_j – значення j -то ознаки, яке дорівнює 1, якщо за цією ознакою можна охарактеризувати об'єкт (наприклад можливість додавати коментарі до діаграм у певному CASE-засобі), 0 – якщо цієї ознаки немає; θ_j – показник, за яким оцінюється об'єкт j ; β_i – показник складності ознаки i ; $\psi()$ – логістична функція.

Георг Раш припустив, що рівень θ_j і рівень β_i розміщені на одній шкалі і вимірюються в одних і тих самих одиницях – логітах [14,15]. Аргументом функції є різниця $(\theta_j - \beta_i)$. Оскільки модель Раша описує ймовірність наявності ознаки як функцію одного параметра $(\theta_j - \beta_i)$, то іноді її називають однопараметричною моделлю IRT [14].

Таким чином, якщо побудувати систему ознак, за якими можна охарактеризувати певні CASE-засоби, то можна оцінити кожне з них на основі однопараметричної моделі Раша. Для практичного застосування цього підходу необхідно зібрати первісні дані, за якими можна оцінити ознаки x_j . Це можна реалізувати або шляхом створення анкети та проведення анкетування, або використовуючи автоматизовані засоби збирання даних, зокрема в мережі Інтернет.

Висновки. Таким чином, аналізуючи сучасний стан організації бізнес-процесів виробництва програмних продуктів, можна зробити висновок, що об'єктно-орієнтовані CASE-засоби дають змогу підвищити ефективність виробничих процесів на всіх етапах життєвого циклу програмних систем. З іншого боку, існує велике розмаїття CASE-засобів, які відрізняються функціональними можливостями, характеристиками якості, а також фінансовими показниками. Це зумовило проведення дослідження щодо можливостей оцінювання та вибору CASE-засобу.

Внаслідок проведеного дослідження запропоновано застосування методу статистичного вимірювання на основі використання однопараметричної моделі Раша для визначення оцінок CASE-засобів, які характеризуються великою кількістю різноманітних ознак, більшість з яких мають якісний характер. Подальше дослідження буде присвячено створенню механізмів автоматизованого збирання даних для визначення оцінок CASE-засобів.

Література

1. Назарова Д.Б. Разработка реляционных баз данных с использованием CASE-средства AllFusion Data Modeler / Д.Б. Назарова. – М. : ФЛИНТА, 2013. – 74 с.
2. Лаврищева К.М. Базовые основы индустрии программ, вычислений и данных / К.М. Лаврищева // Спецвыпуск Восьмой межд. научн.-практ. конф. УкрПРОГ' 2012 "Проблемы программирования". – К., 22-24 мая 2012 г. – С. 57-68.
3. Сергієнко І.В. Інформатика та комп'ютерні технології / І.В. Сергієнко. – К. : Наук. думка, 2004. – 430 с.
4. Основы инженерии качества программных систем / Ф.И. Андон, Г.И. Коваль, Т.М. Коротун, Е.М. Лаврищева, В.Ю. Сулов. – К. : Академперіодика, 2002. – 504 с.

5. Дедуктивные, индуктивные и аналитические методы представления и обработки компьютерных знаний в интеллектуальных системах / З.М. Асельдеров, К.П. Вершинин, А.В. Лялецкий, А.Ю. Паскевич, В.П. Клименко, Ю.С. Фішман // Математические машины и системы. – 2003. – № 3-4. – С. 51-74.

6. Лукин М.А. Разработка и автоматическая верификация параллельных автоматных программ / М.А. Лукин, А.А. Шальто // Информационно-управляющие системы. – 2013. – № 5(66). – С. 43-50.

7. Харченко С.Л. Генерация компонент программного обеспечения системы управления по формальному описанию их моделей поведения и ограничений / С.Л. Харченко // ВЕЖИТ. – 2010. – № 3(44). – С. 33-36.

8. Вендров А.М. CASE-технологии. Современные методы и средства проектирования информационных систем / А.М. Вендров. – М. : Финансы и статистика, 1998. – 176 с.

9. Похилько А.Ф. CASE-технология моделирования процессов с использованием средств BPWin и ERWin : [учеб. пос.] / А.Ф. Похилько, И.В. Горбачев. – Ульяновск : УлГТУ, 2008. – 120 с.

10. Rational Rose Modeler // Программное обеспечение IBM. [Электронный ресурс]. – Доступный с <http://www-03.ibm.com/software/products/ru/rosemod>.

11. Бориско С.Н. Оптимизация выбора объектно-ориентированного CASE-средства в жизненном цикле программного продукта / С.Н. Бориско, И.Н. Васильев, В.И. Лобейко // Известия ВолгГТУ. – 2011. – № 10. – С. 42-45.

12. Канжелев С.Ю. Автоматическая генерация автоматного кода / С.Ю. Канжелев, А.А. Шальто // Информационно-управляющие системы. – 2006. – № 6. – С. 35-42.

13. Лаврищева Е.М. Подход к экспертному оцениванию в программной инженерии / Е.М. Лаврищева, О.А. Слабоспицкая // Кибернетика и системный анализ. – 2009. – № 4. – С. 151-168.

14. Чельшкова М.Б. Теория и практика конструирования педагогических тестов : [учебн. пособие] / М.Б. Чельшкова. – М. : Логос, 2002. – 432 с.

15. Ким В.С. Тестирование учебных достижений : монография / В.С. Ким. – Уссурийск : Изд-во УГПИ, 2007. – 169 с.

Погромская А.С. Рационализация определения объектно-ориентированного CASE-средства в жизненном цикле программной системы

Предложена концепция определения оценок CASE-средств для проектирования и разработки программных систем на основе применения однопараметрической модели современной теории измерений IRT. Рассмотрены характерные особенности CASE-средств и их возможностей для проектирования программных систем. Определены основные признаки, по которым можно проводить оценку и выбор. Разработано и рекомендовано применение метода статистического измерения на основе использования однопараметрической модели Раша для определения оценок CASE-средств, характеризующихся большим количеством признаков, большинство из которых носит качественный характер.

Ключевые слова: CASE-средства, жизненный цикл программы, программная система, модель Георга Раша.

Pogromska H.S. Rationalization of the Definition of Object-Oriented CASE Tools in the Life Cycle of a Software System

The concept of determining estimates of CASE-tools for the design and development of software systems through the application of a one-parameter model of modern measurement theory IRT. The characteristic features of CASE tools and their capabilities for designing software systems. We have examined the characteristics of CASE-tools and their capabilities for designing software systems. Developed and recommended the use of the method of statistical measurement with a one-parameter Rasch model to determine the grade of CASE-tools.

Key words: CASE tools, life cycle of the program, software system, Georg Rasch model.