

ОЦІНКА ХАРАКТЕРИСТИК CASE-ЗАСОБІВ НА ОСНОВІ ОДНОПАРАМЕТРИЧНОЇ МОДЕЛІ РАША**Г. С. Погромська**Миколаївський національний університет імені В.О. Сухомлинського
вул. Нікольська, 24, м. Миколаїв, 54003, Україна. E-mail: pas012@ukr.net

Вибір CASE-засобу при створенні програмної системи є важливим з точки зору ефективності та надійності роботи групи проектувальників та програмістів. Аналізуючи сучасний стан організації бізнес-процесів виробництва програмних продуктів, зроблено висновок, що об'єктно-орієнтовані CASE-засоби дозволяють підвищити ефективність виробничих процесів на всіх етапах життєвого циклу програмних систем. Існує велике розмаїття CASE-засобів, які відрізняються функціональними можливостями, характеристиками якості, а також фінансовими показниками. Це обумовило проведення дослідження щодо можливостей оцінювання та вибору CASE-засобу. Розглянуто характерні особливості CASE-засобів та їх можливостей для проектування програмних систем. Виділено чотири основні компоненти, за якими можна оцінювати CASE-засоби: аналіз, проектування, розробка та інфраструктура. Визначено основні критерії, за якими можливо проводити оцінювання та вибір CASE-засобів: функціональність, надійність, супровід, здатність до перенесення. У якості критеріїв варто враховувати також вартість, витрати і ефект впровадження. Велика кількість критеріїв оцінки, якісний характер та складність збору необхідних даних, обумовлюють необхідність розробки методологічного підходу до вирішення задачі оцінювання та вибору CASE-засобів з метою підвищення ефективності виробничих процесів підтримки життєвого циклу програмних систем. Враховуючи складність використання експертних методів на практиці, було запропоновано використання статистичної процедури оцінки CASE-засобів з метою подальшого вибору певного інструменту. Запропоновано підхід до визначення оцінок CASE-засобів для проектування та розробки програмних систем на основі застосування однопараметричної моделі сучасної теорії вимірювань IRT. Для практичного застосування окресленого підходу необхідно зібрати первісні дані, за якими можна оцінити ознаки. Це можливо реалізувати або шляхом створення анкети та проведення анкетування, або використовуючи автоматизовані засоби збору даних, зокрема в мережі Інтернет. Подальше дослідження присвячено створенню механізмів автоматизованого збору даних для визначення оцінок CASE-засобів.

Ключові слова: CASE-засоби, життєвий цикл, програмна система, оцінювання, модель Раша.

ОЦЕНКА ХАРАКТЕРИСТИК CASE-СРЕДСТВ НА ОСНОВЕ ОДНОПАРАМЕТРИЧЕСКОЙ МОДЕЛИ РАША**А. С. Погромская**Николаевский национальный университет имени В.А. Сухомлинского
ул. Никольская, 24, г. Николаев, 54003, Украина. E-mail: pas012@ukr.net

Выбор CASE-средства при создании программной системы является важным с точки зрения эффективности и надежности работы группы проектировщиков и программистов. Анализируя современное состояние организации бизнес-процессов производства программных продуктов, сделан вывод, что объектно-ориентированные CASE-средства позволяют повысить эффективность производственных процессов на всех этапах жизненного цикла программных систем. Существует большое разнообразие CASE-средств, которые отличаются функциональными возможностями, характеристиками качества, а также финансовыми показателями. Это обусловило проведение исследования о возможностях оценки и выбора CASE-средства. Рассмотрены характерные особенности CASE-средств и их возможностей для проектирования программных систем. Выделено четыре основных компонента, по которым можно оценивать CASE-средства: анализ, проектирование, разработка и инфраструктура. Определены основные критерии, по которым можно проводить оценку и выбор CASE-средств: функциональность, надежность, сопровождение, способность к перенесению. В качестве критериев следует учитывать также стоимость, затраты и эффект внедрения. Большое количество критериев оценки, качественный характер и сложности сбора необходимых данных, обуславливают необходимость разработки методологического подхода к решению задачи оценки и выбора CASE-средств с целью повышения эффективности производственных процессов поддержки жизненного цикла программных систем. Учитывая сложность использования экспертных методов на практике, было предложено использование статистической процедуры оценки CASE-средств с целью последующего выбора определенного инструмента. Предложен подход к определению оценок CASE-средств для проектирования и разработки программных систем на основе применения однопараметрической модели современной теории измерений IRT. Для практического применения предлагаемого подхода необходимо собрать первоначальные данные, по которым можно оценить признаки. Это возможно реализовать путем создания анкеты и проведение анкетирования или используя автоматизированные средства сбора данных, в частности в сети Интернет. Дальнейшее исследование посвящено созданию механизмов автоматизированного сбора данных для определения оценок CASE-средств.

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

АКТУАЛЬНІСТЬ РОБОТИ. 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 отрима-

ла форму віртуального відображення. Багато ручних операцій автоматизувалися, а сам процес вирішення професійних завдань перейшов в діалог «людина – комп'ютер».

В даний час 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 методів раціоналізації в систему організаційних перетворень. Тому не дивно, що ці питання постали й з точки зору процесів розробки програмних систем.

В процесі розробки програмного забезпечення зазвичай використовуються моделі, а також інструментальні засоби для підтримки та уніфікації моделювання. Але створення програмної системи – це не тільки побудова моделі. Чим складніше і ширше 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 є інструментарієм раціоналізації процесів розробки програмних систем. Зазвичай до CASE-засобів відносять будь-яке програмний засіб, що автоматизує один процес або сукупність процесів життєвого циклу програмного забезпечення [1]. Використання CASE-засобів дозволяє 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 при створенні програмної системи є важливим з точки зору ефективності та надійності роботи групи проектувальників та програмістів. Це обумовлює актуальність дослідження питань вибору CASE-засобів та раціоналізації організаційно-виробничих процесів розробки програмних систем.

Теоретичний доробок у сфері програмної інженерії достатньо значний. Багато літератури присвячено питанням життєвого циклу програмних продуктів [2–6]. Авторами української теоретичної школи програмування (П. І. Андон, Д. Б. Буй, В. М. Глушков, С. С. Гороховський, К. М. Лавріщева, О. Цейлін та ін.) [2–4] розглядаються різноманітні підходи розв'язання проблем програмування. Зокрема, можна підкреслити особливу увагу до CASE-засобів підтримки процесів життєвого циклу програмної системи [2, 7].

Слід зауважити, що більшість робіт, які розглядають питання щодо CASE-засобів, присвячено опису переваг та можливостей певних інструментів [7–9]. Ряд робіт, наприклад [10, 12], розглядають особливості CASE-засобів як особливих типів програмних систем. В цих роботах 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 [7]. Крім того, деякі автори розглядають інтегровано-

вані CASE-засоби, які забезпечують підтримку усіх 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 обумовило напрям досліджень даної роботи.

Метою роботи є підвищення ефективності процесів розробки програмних систем шляхом розроблення методологічної основи для вирішення задачі вибору CASE-засобів як напрямку раціоналізації процесів підтримки життєвого циклу програмних систем.

МАТЕРІАЛ І РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ. CASE-технології утворюють певне середовище розробки інформаційної або програмної системи та забезпечують процес розробки 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 [8].

Сучасні CASE-засоби мають наступні характерні складові:

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

На даний час в технології розробки програмних систем існує два основних підходи, які відрізняються критеріями декомпозиції: функціонально-модульний (або структурний) та об'єктно-орієнтований [7–9].

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 необхідності 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 використовуються різні моделі, які описують [7, 8]: 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 являє собою сукупність правил і процедур, призначених для побудови функціональної моделі об'єкта будь-якої предметної області. Технологія SADT (перейменована в IDEF0) традиційно використовуються для моделювання організаційних систем (бізнес-процесів). Достоїнства 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. У даний час до сімейства IDEF прийнято відносити наступні методології [7]:

– IDEF0 – методологія функціонального моделювання, забезпечена наочною графічною мовою. Дозволяє 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 дозволяє відображати і аналізувати їх структуру та взаємозв'язки;

– IDEF1X (IDEF1 Extended) – методологія моделювання, яка застосовується для побудови інформаційної моделі, що становить структуру інформації, необхідної для підтримки функцій виробничої системи або середовища. IDEF1X часто використовується для моделювання реляційних баз даних, що мають відношення до даної 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 являють собою ієрархію 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 [7, 9].

Більшість сучасних методів ООАП базуються на використанні мови UML [9]. Уніфікована мова моделювання UML (Unified Modeling Language) являє собою мову 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.

Методи структурного проектування допомагають спростити процес розробки 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 якості блоків класи та об'єкти. Об'єктно-орієнтоване проектування та об'єктно-орієнтоване програмування покращує можливості структурного проектування, концентруючи більш уваги на даних системи, а не на тому, що система робить. Цей підхід дозволяє створювати системи, які краще супроводжувати, вони більш гнучкі, більш стійкі та більш пристосовані до багатократного використання.

Розповсюдженими 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 дозволяє 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 консорціума Object Management Group (OMG), членами якого є більш 50 провідних комп'ютерних компаній (Sun, DEC, IBM, HP, Motorola та ін.) [7]. Основним напрямком діяльності консорціума є розробка специфікацій та стандартів для створення розподілених об'єктних систем в різноманітних середовищах.

Існує декілька 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 стислу 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 технічних 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, дозволяє моделювати бізнес-логіку, дані, інформаційні системи тощо. Цей CASE-засіб потребує спеціальних практичних навичок для засвоєння та експлуатації. В якості засобів конфігураційного управління Oracle використовує Rational ClearCase.

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

Класична постановка задачі розробки програмної системи являє собою спіральний цикл інтерактивної зміни етапів об'єктно-орієнтованого аналізу, проектування та реалізації. В 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 випадку засновано на реінжинірингу програмних кодів, при якому 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 можна оцінювати CASE-засоби: аналіз, проектування, розробка та інфраструктура. Характеристики цих компонентів подані в табл. 1 та докладно описані в [10].

Не зважаючи на безліч переваг, можна підкреслити 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.

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 оцінки та вибору CASE-засобів:

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

Крім цього, у якості критеріїв варто враховувати також вартість, витрати і ефект впровадження.

Велика кількість критеріїв оцінки, якісний характер та складності збору необхідних даних, обумовлюють необхідність розробки методологічного підходу до вирішення задачі оцінювання та вибору CASE-засобів з метою підвищення ефективності виробничих процесів підтримки життєвого циклу програмних систем.

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

Таблиця 1 – 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 іншої	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 реінжинингу програмного коду, 4GL, клієнт-серверних систем в діаграми моделей

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 збору даних дозволить організувати збір даних щодо використання різноманітних CASE-засобів з урахуванням різних точок зору. В цьому випадку постає питання подальшої обробки статистичної інформації. З цієї метою запропоновано використання результатів, які отримано в рамках сучасної теорії вимірювань [14, 15]. Під сучасною теорією вимірювань розуміється теорія IRT (Item Response Theory), призначена для оцінки 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 відносять наступні.

1. IRT перетворює вимірювання, виконані в дихотомічних і порядкових шкалах, у лінійні вимірювання, в результаті якісні дані аналізуються за допомогою кількісних методів.

2. Міра вимірювання параметрів моделей IRT є лінійною, що дозволяє 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 використаного набору ознак.

5. Неповнота даних (пропуск деяких комбінацій об'єкт-ознака) не є критичним.

Найбільш відомою моделлю вимірювання, яку розроблено в IRT є модель Раша [14]. В моделі Раша встановлюється зв'язок між двома множинами значень латентних змінних. Першу множину складають значення латентної змінної, що визначає оцінку показника θ_i , де i – номер об'єкту, що змінюється в інтервалі від 1 до N (наприклад, N – кількість осіб, які приймали участь в опитуванні). Другу множину складають значення латентної змінної, що характеризує складність j -ї 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: $f(x_j, \theta, \beta_j)$.

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

де x_j – значення j -ї ознаки, яке дорівнює 1, якщо за цієї 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 же 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 можливо реалізувати або шляхом створення анкети та проведення анкетування, або використовуючи автоматизовані засоби збору 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 дозволяють 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 обумовило проведення дослідження щодо можливостей оцінювання та вибору CASE-засобу.

В результаті 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 носить якісний характер. Подальше дослідження буде присвячено створенню механізмів автоматизованого збору даних для визначення оцінок CASE-засобів.

ЛІТЕРАТУРА

1. Назарова Д. Б. Разработка реляционных баз данных с использованием CASE-средства AllFusion Data Modeler. Москва. Флинта, 2013. 74 с.
2. Лаврищева К. М. Базовые основы индустрии программ, вычислений и данных. *Проблемы программирования*. 2012. 22-24 мая. С. 57–68.
3. Основы инженерии качества программных систем / Ф. И. Андон и др.; Київ. Академперіодика, 2002. 504 с.

4. Асельдеров З. М., Лялецкий А. В. Дедуктивные, индуктивные и аналитические методы представления и обработки компьютерных знаний в интеллектуальных системах. *Математические машины и системы*. 2003. № 3-4. С. 51–74.

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

6. Харченко С. Л. Генерация компонент программного обеспечения системы управления по формальному описанию их моделей поведения и ограничений. *Восточно-Европейский журнал передовых технологий*. 2010. №3 (44). С. 33–36.

7. Вендров А. М. CASE-технологии. Современные методы и средства проектирования информационных систем. Москва. Финансы и статистика, 1998. 176 с.

8. Похилько А. Ф., Горбачев И. В. CASE-технология моделирования процессов с использованием средств BPWin и ERWin учебное пособие. Ульяновск: УлГТУ, 2008. 120 с.

9. Rational Rose Modeler. IBM. Программное обеспечение IBM. URL: <http://www-03.ibm.com/software/products/ru/rosemod>.

10. Погромська Г. С. Раціоналізація визначення об'єктно-орієнтованого CASE-засобу в життєвому циклі програмної системи. *Науковий вісник НЛТУ України: збірник науково-технічних праць*. Львів: РВВ НЛТУ України. 2015. Вип. 25.2. С. 356–365.

11. Бориско С. Н., Васильев И. Н. Оптимизация выбора объектно-ориентированного CASE-средства в жизненном цикле программного продукта. *Известия ВолГТУ*. 2011. № 10. URL: <http://cyberleninka.ru/article/n/optimizatsiya-vybora-obektno-orientirovannogo-case-sredstva-v-zhiznennom-tsikle-programmnogo-produkta> (дата звернення 15.12.2018).

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

13. Лаврищева Е. М., Слабоспицкая О.А. Подход к экспертного оценивания в программной инженерии. *Кибернетика и системный анализ*. 2009. № 4.

14. Чельшкова М. Б. Теория и практика конструирования педагогических тестов. Москва: Логос, 2002. 432 с.

15. Ким В. С. Тестирование учебных достижений. Монография. Усурийск: УГПИ, 2007. 169 с.

ESTIMATION OF THE CASE-TOOLS CHARACTERISTICS BASED ON THE ONE-PARAMETER RASCH'S MODEL

Hanna S. Pohromska

Mykolaiv National University named after V.O. Sukhomlinsky
vul. Nikolska, 24, Mykolaiv, 54003, Ukraine. E-mail: pas012@ukr.net

Purpose. Improving the efficiency of software development processes with the help of a methodological framework leads to solving the CASE-tools choice as a way to streamline processes, which support the software systems' life cycle. **Methodology.** Taking into account the complexity of expert methods in practice, it was proposed to use a statistical procedure for assessing CASE funds in order to select a specific instrument. Traditional methods of data collection are focused on questionnaires, surveys or official reports processing. **Findings.** The features of CASE-tools and their possi-

bilities for designing software systems are considered. There are four main components that can be evaluated by CASE tools: analysis, design, development and infrastructure. The main criteria for evaluating and choosing CASE-tools are the following: functionality, reliability, support, transferability. In addition, the criteria include costs, implementation and its effect. On the other hand, the automation of data collection processes will enable the data collection on the use of various CASE-tools, taking into account different points of view. In this case, this question arises further statistical information processing. Thus, we propose to use the results obtained within the modern measurement theory. **Originality.** The methodological approach in solving the task of evaluation and the choice of CASE-tools are developed in order to increase the efficiency of software systems production processes life cycle. **Practical value.** We propose the approach to the CASE-tools estimation for designing and developing software systems based on one-parameter model application according the modern theory of IRT measurements. For its practical application, it is necessary to collect the initial data which signs can be evaluated. It is possible to implement it either by creating questionnaires and conducting questionnaires, or by using automated means of data collection, in particular on the Internet. Further research will be devoted to the automated data mechanisms creation and collection to determine CASE-tools estimation. References 15, tables 1.

Key words: CASE-tools, life cycle of the program, software system, estimation, measurement theory, Rasch's model.

REFERENCES

1. Nazarova, D. B. (2013), *Razrabotka relyatsionnykh baz dannykh s ispolzovaniyem CASE-sredstva AllFusion Data Modeler* [Development of relational databases using the CASE-tool AllFusion Data Modeler], Moskva. Flinta. Russia, 74 p.
2. Lavrishcheva, K. M. (2012), "*Bazovyie osnovy industrii programm. vychisleniy i dannykh*" [Basic fundamentals of the software, computing and data industry], *Problemy programmirovaniya*, pp. 57-68.
3. *Osnovy inzhenerii kachestva programnykh sistem* (2002), [Fundamentals of software quality engineering], F.I. Andon i dr.; Kiiv. Akademperiodika. 2002. 504 p.
4. Aselderov, Z. M., Lyaletskiy, A. V. (2003), "*Deduktivnyye. induktivnyye i analiticheskiye metody predstavleniya i obrabotki kompyuternykh znaniy v intellektualnykh sistemakh*" [Deductive, inductive and analytical methods for the representation and processing of computer knowledge in intelligent systems], *Matematicheskiye mashiny i system*, vol. 3-4, pp. 51-74.
5. Lukin, M. A., Shalyto, A. A. (2013), "*Razrabotka i avtomaticheskaya verifikatsiya parallelnykh avtomatnykh program*" [Development and automatic verification of parallel automaton programs], *Informatsionno-upravlyayushchiye sistemy*, vol. 5(66). pp. 43-50.
6. Kharchenko, S. L. (2010), "*Generatsiya komponent programnogo obespecheniya sistemy upravleniya po formalnomu opisaniyu ikh modeley povedeniya i ogranicheniy*" [Generation of software components of the control system by the formal description of their models of behavior and limitations], *Vostochno-Evropeyskiy zhurnal peredovykh tekhnologiy*, vol. 3(44), pp. 33-36.
7. Vendrov, A. M. (1998), *CASE-tehnologii. Sovremennyye metody i sredstva proyektirovaniya informatsionnykh sistem* [CASE-technology. Modern methods and means of designing information systems], Moskva. Finansy i statistika. 176 p.
8. Pokhilko, A. F., Gorbachev, I. V. (2008), "*CASE-tehnologiya modelirovaniya protsessov s ispolzovaniyem sredstv BpWin i ERWin*" [CASE-technology process modeling using BPWin and ERWin], Ulianovsk: UIGTU, 120 p.
9. Rational Rose Modeler. IBM. Programmnoye obespecheniye IBM [Rational Rose Modeler. IBM. Software IBM], URL: <http://www-03.ibm.com/software/products/ru/rosemod> (Last accessed 15.11.2018).
10. Pohromska, H. S. (2015), "*Ratsionalizatsiia vyznachennia ob'ektno-orientovanoho CASE-zasobu v zhyttievomu tsykli prohramnoi systemy*" [Rationalization of the Definition of Object-Oriented CASE Tools in the Life Cycle of a Software System], *Naukovyi visnyk NLTU Ukrainy: zbirnyk naukovykh tekhnichnykh prats [Scientific Bulletin of UNFU]*, Lviv: RVV NLTU Ukrainy, vol. 25.2. pp. 356-365.
11. Borisko, S. N., Vasilyev, I. N. (2011), "*Optimizatsiya vybora ob'ektno-orientirovannogo CASE-sredstva v zhiznennom tsikle programnogo produkta*" [Optimizing the choice of object-oriented CASE-tools in the life cycle of a software product], *Izvestiya VolgGTU*, vol. 10. URL: <http://cyberleninka.ru/article/n/optimizatsiya-vybora-obektno-orientirovannogo-case-sredstva-v-zhiznennom-tsikle-programnogo-produkta> (Last accessed 15.12.2018).
12. Kanzhelev, S. Yu., Shalyto, A. A. (2006), "*Avtomaticheskaya generatsiya avtomatnogo koda*" [Automatic generation of automaton code], *Informatsionno-upravlyayushchiye sistemy*, vol. 6, pp. 35-42.
13. Lavrishcheva, E. M., Slabospitskaya, O. A. (2009), "*Podkhod k ekspertnogo otsenivaniya v programnoy inzhenerii*" [An approach to expert evaluation in software engineering], *Kibernetika i sistemnyy analiz*, vol. 4.
14. Chelyshkova, M. B. (2002), *Teoriya i praktika konstruirovaniya pedagogicheskikh testov* [Theory and practice of constructing pedagogical tests], Moskva: Logos, 432 p.
15. Kim, V. S. (2007), *Testirovaniye uchebnykh dostizheniy* [Testing educational achievements.], Monografiya. Ussuriysk: UGPI. 169 p.

Стаття надійшла 18.09.2019.