

О. Ю. Чередніченко¹
Ю. М. Гонтар¹
О. В. Іващенко¹
М. А. Вовк¹

АНАЛІЗ КОМПОНЕНТНО-ОРІЄНТОВАНИХ МЕТОДІВ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ЕЛЕКТРОННОГО БІЗНЕСУ

¹Національний технічний університет «Харківський політехнічний інститут»

Компонентно-орієнтована розробка — це шлях до розробки програмного забезпечення, де переваги повторного використання мають велике значення. Швидка адаптація до змін на ринку має вирішальне значення для електронного бізнесу. А компонентно-орієнтована розробка здатна забезпечити швидкий термін реалізації, гнучкість та масштабованість. З точки зору придатності до використання в контексті інженерії електронного бізнесу розглянуто п'ять популярних методів: CADA, Catalysis, Comet, Rational Unified Process і Select Perspective. Для кожного із зазначених методів досліджено поняття, процедури, методи та рівень інструментальної підтримки. Оцінка проводиться на основі схеми для розробки електронного бізнесу та переліку об'єктивних критеріїв: передумови походження, використання методології, функціональна підтримка процесу, повторне використання, використання сучасних технік моделювання, інструментальна підтримка, платформа впровадження.

Щодо процесу підтримки Select Perspective пропонує дуже строгий процес з чіткими результатами та діями. Catalysis обмежується прикладами того, як використовувати метод у деяких конкретних ситуаціях. CADA, COMET, і RUP пропонують щось середнє між цими крайнощами. Відносно повторного використання Select Perspective є найпривабливішим методом з точки зору наявності компонентного менеджера, механізму зберігання та пошуку компонентів, а решта досліджуваних методик покладаються на досвід розробників у пошуку компонентів. CADA, Catalysis, Comet не забезпечують спеціальну підтримку інструментів, але це стосується стандартних UML-інструментів, лише Select Perspective та Rational Unified Process, що походять від промислових методів, забезпечують велику інструментальну підтримку. Усі методи залежать від платформи. Встановлено, що основні елементи компонентно-орієнтованої розробки, компоненти та повторне використання наразі ще не зрілі, а діапазон інструментів для компонентно-орієнтованої розробки недостатній. Всі розглянуті методи акцентують увагу на окремих аспектах компонентно-орієнтованої розробки, але поки що жоден з них не пропонує повного вирішення.

Ключові слова: компонентно-орієнтована розробка, інженерія електронного бізнесу, компонент, критерії оцінки, моделювання.

Вступ

Компонентно-орієнтована розробка (КОР) — це підхід до системного аналізу та дизайну, який розвинувся з об'єктно-орієнтованої парадигми (ООП). КОР часто розглядається як рішення для розробки застосувань в 21 столітті [1], [2]. В основному, спираючись на повторне використання, прихильники КОР обіцяють швидший термін постачання на ринок, зниження витрат, поліпшення якості, гнучкість та масштабованість. Це всі якості, які дуже потрібні в сучасній економіці.

Треба зазначити, що наразі спостерігається зародження нової інженерної дисципліни — інженерія електронного бізнесу (e-business). Підприємства все більше залежать від інформаційно-комунікаційних технологій (ІКТ). ІКТ перетворилася з допоміжної ролі на основний бізнес багатьох організацій. Цей перехід до електронного бізнесу (наприклад, ведення бізнесу з використанням ІКТ) вимагає міждисциплінарного підходу, який об'єднує елементи реінжинірингу бізнес-процесів (РБП), інтеграції ланцюжків поставок, маркетингу та програмного забезпечення [3]. КОР відіграє центральну роль в інженерії електронного бізнесу.

Протягом останніх років КОР отримала велику увагу в літературі з програмного забезпечення

[4]—[6]. Замість того, щоб створювати дрібні об'єкти, КОР розміщує великі, самостійно упаковані, багаторазові компоненти, які іноді називають підсистемами [7].

КОР підкреслює повторне використання, тоді як інші методи, такі як ООП, ігнорують це питання або вводять його занадто пізно у життєвий цикл [8]. Компоненти є узгодженими частинами системи, які можна самостійно зберігати та збирати в нові програмні системи. Потенціал заощаджень часу розробки та витрат очевидний. Крім того, завдяки своїй структурі компонентно-орієнтоване застосування буде гнучкішим і масштабованим. Використання компонентів гарантує кращу якість, оскільки компоненти часто використовуються та вдосконалюються з часом.

В ідеалі, розробник програмного забезпечення може використовувати компоненти інших розробників. Це дозволяє скоротити час розробки програм, але й накладає дві важливі методологічні вимоги. По-перше, компоненти повинні бути однозначно визначені, що дозволить розробникам позиціонувати компонент: яка його функціональність, де і як її використовувати. По-друге, КОР потребує процесу, в якому вирішуються питання функціональності та використання компонентів.

Електронний бізнес потребує переваг, які пропонує КОР: швидкий термін реалізації, гнучкість та масштабованість. Швидка адаптація до змін на ринку має вирішальне значення для бізнесу. Разом з тим, виникають мережеві підприємства, бізнес-процеси яких не прив'язані до певного місця, як і їх відповідні програмні системи. Необхідний інтегральний підхід до інжинірингу електронного бізнесу, який поєднує в собі мережеве проектування з компонентним розвитком транзакційних послуг [9]. З вищевикладеного виникає питання: «Який метод дозволяє скористатися усіма перевагами КОР в контексті інженерної діяльності в галузі електронного бізнесу?»

Метою дослідження є оцінка вибору та порівняння методів компонентно-орієнтованої розробки: CADA, Catalysis, Comet, Rational Unified Process і Select Perspective щодо розробки програмного забезпечення для електронного бізнесу.

Результати дослідження

Існує багато підходів до оцінки методів розробки ПЗ. З одного боку, використовуються досить стандартні основи для оцінки [1]. Завдяки своєму загальному характеру ця структура підходить для оцінки конкретних питань в КОР. З іншого боку, в цій роботі оцінюються методи щодо їх відповідності інженерії електронного бізнесу.

Розглянемо можливі критерії оцінки (рис. 1). Для кожного методу досліджується поняття, процедури («спосіб роботи»), методи («спосіб моделювання») та рівень інструментальної підтримки («спосіб підтримки»). Кожен опис методу включає ці питання. Зазначимо ці аспекти у питаннях, специфічних для методів КОР, таких як повторне використання та застосовність методу. Щоб отримати загальніший вигляд додано загальні критерії. Спосіб мислення не деталізований, але включений в загальний опис методів.



Рис. 1. Критерії оцінки методів КОР

Таким чином розглянемо критерії оцінки методів КОР, які використовуються:

1. Загальні критерії.

1.1. Передумови. Тобто, чи було розроблено метод в промисловості або в академічному середовищі; використовується лише у поєднанні з іншими критеріями. Це показує можливі кореляції між фоновими та іншими критеріями, наприклад, критерієм підтримка інструменту.

1.2. Використання. Ступінь використання методу може вказувати його рівень зрілості. Можна дати лише грубу вказівку, оскільки у нас немає даних про фактичне використання методів. Деякі методи існували до того, як КОР було введено як нову, додаткову функцію.

2. Спосіб роботи.

2.1. Підтримка процесу. Метод не буде методом, якщо він не дає деяких вказівок для процесу розробки. Форми цієї інструкції, однак, можуть суттєво відрізнятися. Є методи, які визначають строгий порядок дій, тоді як інші лише визначають основні етапи чи аспекти, які слід розглянути.

2.2. Підтримка повторного використання. Одним з основних переваг компонентної розробки є потворне використання. Оцінюючи різні методи, розглянуто рівень підтримки, який передбачається для повторного використання. Чи можуть бути повторно використані лише компоненти програмного забезпечення (коди або виконувані файли), а також, наприклад, бізнес-моделі? Чи існує конкретна модель повторного використання? Чи дає метод загальний опис компонентів для повторного використання, включаючи контекст, в якому компонент може бути використаний, наприклад, як шаблон? Чи забезпечує цей метод процедури зберігання компонентів в сховищі та їх повторне завантаження?

3. Спосіб моделювання.

3.1. Використовуються методи моделювання. Вибір методу часто визначається методами моделювання, які підтримують або використовують цей метод. Тому будемо розглядати вказівки на види методів моделювання.

4. Спосіб підтримки.

4.1. Підтримка інструментів. На погляд авторів, метод не буде використовуватися, якщо він не підтримується інструментами. Рівень підтримки інструментів може відрізнятися від загальних інструментів моделювання або інтегрованих наборів інструментів, які допомагають розробникам.

4.2. Платформа впровадження. Деякі методи спеціально розроблені для конкретної платформи або архітектури. Це також є важливим критерієм для вибору методу.

Оскільки КОР є відносно новим підходом до розробки програмного забезпечення, кількість методів, спеціально розроблених для КОР, все ще обмежена. Деякі методи КОР є розширенням існуючих методів, зокрема методів об'єктно-орієнтованого програмування.

Через популярність інструментів підтримки моделювання, метод Rational Unified Process (RUP) [10] є очевидним кандидатом для оцінки. Вибір інструментарію Princeton Softech також добре відомий і супроводжується добре документованим методом для КОР [12]. Тому ці два методи обрано як промислові методи в нашому порівнянні.

Метод Catalysis [12], що походить із Брайтонського університету, є одним з найвідоміших і найчастіше цитованих методів з академічного середовища. Будучи іншим кандидатом у цьому класі, вивчено його розвиток в рамках проекту ЄС OBOE, розробленого в SINTEF у Норвегії, яке отримало назву COMET [14]. Нарешті, будемо описувати підхід COMPAS для аналізу та проектування (CADA) [13], розроблений в Інституті досліджень та інновацій Ордіни. CADA має академічну основу, оскільки він створений на прикладі Catalysis, але він також містить кілька додаткових ідей на основі практичних застосувань методу. Тому його можна вважати гібридом академічного та промислового методу.

Кожний метод вводиться з інформацією про його основи, спосіб мислення та спосіб роботи. Після цього метод оцінюється на підставі критеріїв, визначених раніше, і пов'язаних з ним з погляду розробки систем електронного бізнесу. Таким чином, для аналізу обрано методи компонентно-орієнтованої розробки, показані на рис. 2.

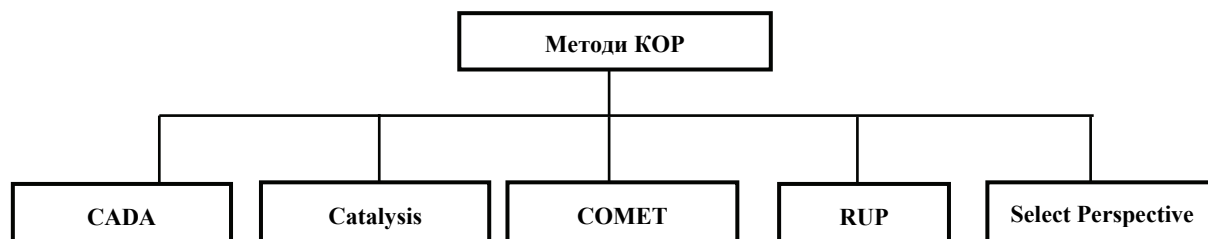


Рис. 2. Основні методи компонентно-орієнтована розробки

Надамо стислий опис кожного з методів КОР. Catalysis є дослідницькою ініціативою, розробленою у Брайтонському університеті [12]. Метод призначений для підтримки моделювання та побудови відкритих розподілених систем. Catalysis — це інкрементна побудова бізнес-моделі, що використовує пакування, виготовлення та розкладання. Catalysis наводить приклади цього процесу, але не забезпечує фіксованого набору віх, а також не забезпечує заздалегідь визначеного способу розробки програмного забезпечення. У кінцевій моделі досягається рівень деталізації, в якому компоненти моделі розпізнаються як у бізнес-процесі, так і в процесі реалізації.

CADA — це нова методологія розробки програмного забезпечення, розроблена Інститутом досліджень та інновацій Ордіни [13]. Поняття CADA в основному збігаються з поняттями Catalysis. Тим не менше, CADA включає в себе більше концепцій з діаграм, що реалізують UML, і додає низку концепцій для бізнес-моделювання. Однією з центральних ідей методології CADA є те, що метод повинен підтримувати аналітиків та дизайнерів, а не наказувати їм, що робити. Тому CADA дає вказівки для різних етапів проектування, але не встановлює точний порядок дій, які необхідно виконати.

Методологія «Component-based Methodology» (COMET), описана в [14], розроблена в рамках проекту Open Business Object Environment (OBOE), представленого в рамках системи ESPRIT IV. У проекті брали участь кілька груп користувачів і постачальників технологій. COMET характеризується розвитком незалежних від застосування бізнес-об'єктів моделей та реалізацією цих моделей у програмних компонентах. Це загальний опис, який дає рекомендації щодо аналізу, розробки та впровадження програмних систем, зокрема розподілених інформаційних систем. Для реалізації рекомендується використовувати бізнес-об'єкт OMG. Реалізація цієї платформи на базі CORBA також розробляється в рамках проекту OBOE. Значна частина методології, однак, не залежить від платформи впровадження.

Rational Unified Process (RUP) [10] — це метод розробки програмного забезпечення корпорації Rational Software. Це ітеративний, об'єктно-орієнтований, контрольований та підтримуваний інструментарій, що застосовується до всіх видів проектів розробки програмного забезпечення. RUP користується значною популярністю в індустрії програмного забезпечення, особливо серед користувачів Rational-інструментів підтримки моделювання та впровадження. RUP описує в основному «спосіб роботи», оскільки UML [15] є кращим «способом моделювання». Процес розробки поділяється на чотири етапи: створення, розробка, побудова та перехід, і довільна кількість ітерацій. Крім того, RUP складається з декількох робочих процесів, які об'єднують різні види діяльності, а саме: бізнес-моделювання, вимоги, аналіз і дизайн, впровадження, тестування та розгортання, а також з допоміжних робочих процесів, таких як управління проектами, конфігурація та управління змінами.

Select Perspective [11] — це метод розробки програмних застосувань. Підтримується кількома інструментами з Princeton Softech, з яких Select Enterprise (що забезпечує моделювання бізнес-процесів, UML і моделей дизайну даних), і Select Component Manager (це бібліотека для специфікацій компонентів та готових компонентів) є найважливішими. Select Perspective — практичний підхід до розробки застосувань. Тому основні ідеї методології — це результати, які є масштабованими та адаптованими до змін. Для досягнення цієї мети Select Perspective базується на КОР. За даними [11], КОР використовує попередні методології, такі як Structured Development, RAD, і об'єктно-орієнтована розробка, і розширює їх. Select Perspective є ієрархічним методом. Він має фази, основні види діяльності в рамках кожного етапу, а в рамках кожної діяльності — стандартний метод.

Порівнюємо основні характеристики описаних методів (табл.).

Порівняння методів КОР

Метод				
CADA	Catalysis	COMET	RUP	Select Perspective
Передумови походження				
Змішаний	Академічний	Академічний	Промисловий	Промисловий
Використання методології				
Ordina	Декілька зі списку 500 найбільших світових компаній	Тільки для партнерів проекту OBE	Прийнятий декількома великими гравцями	Регулярне використання
Функціональна підтримка процесу				
Віхи, керівні принципи	Керівні принципи	Віхи, керівні принципи	Віхи, керівні принципи, шаблони, інструментальні наставники	Фази, діаграми, діяльності
Повторне використання (одиниці)				
Патерни, компонент	Фреймворк, патерни, компоненти	Компоненти, патерни, фреймворки	Програмні компоненти	Патерни, компоненти
Використання сучасних технік моделювання				
UML; CRC картки; бізнес моделі	Декілька UML діаграм	Декілька UML діаграм	UML та інші	Моделювання бізнес процесів; Декілька UML діаграм; реляційні DBMS схеми; діаграми ERP

Метод				
CADA	Catalysis	COMET	RUP	Select Perspective
Платформа впровадження				
Не вказана	Незалежна, наприклад Java, CORBA	Business Object Framework на базі CORBA	Незалежна	Незалежна, наприклад, Java, CORBA

По-перше, методи порівнюються на основі переліку критеріїв оцінки. По-друге, обговорюється позиціонування методів у довідковій базі та загальні зауваження, що впливають з результатів оцінки.

Процес підтримки. За винятком методу Catalysis, кожний метод описує віхи, які повинні бути завершені наприкінці фази процесу розробки. Траєкторія, що проходить через етапи, не є строгою; користувач може пристосувати її до своєї ситуації. В чотирьох методах траєкторія розробки програмного забезпечення має послідовний характер. Наведені керівні принципи, які допомагають користувачеві під час процесу. Однак керівництво, яке отримує користувач в процесі, істотно відрізняється. Кроки визначені ґрунтовно; результати кожного етапу чи діяльності не наведені. П'ять методів показують дві крайності: Select Perspective пропонує дуже строгий процес з чіткими результатами та діями. Catalysis обмежується прикладами того, як використовувати метод у деяких конкретних ситуаціях. Інші — CADA, COMET, і RUP — знаходяться між цими крайнощами.

Повторне використання. Для цілей повторного використання необхідно записати інформацію про компоненти. Це слід зробити у фіксованій структурі, щоб дізнатися, на які аспекти надалі можна шукати компонент. Прикладом такого шаблону є модель. Ідея записувати знання існує у всіх досліджених методах, але реалізація цієї ідеї залишається невизначеною. Це стосується як самих патернів, так і використання шаблонів під час процесу. Тому не дивно, що деякі моделі мають доступні шаблони. Select Perspective є найпривабливішим методом з точки зору наявності компонентного менеджера, механізму зберігання та пошуку компонентів. Інші методики покладаються на досвід розробників у пошуку компонентів

Використання методології. Як академічне середовище, так і промисловість переконані в перевагах КОР. Тому КОР використовується не тільки у окремих випадках, але і в діяльності вже багатьох компаній.

Використання методів моделювання. Зрозуміло, що UML дуже добре представлений для моделювання системи під час процесу розробки. Використовуються деякі діаграми UML; на жаль, не завжди зрозуміло, які типи діаграм UML використовуються, тому важко бути точнішим. UML в основному використовується для моделювання програмних систем. Коли йдеться про бізнес-моделювання, три методи також використовують UML, але розширюють його, додаючи певну інформацію до діаграм. Інші два способи запроваджують власні, фірмові технічні прийоми для моделювання бізнес-процесу. З методологій лише Select Perspective підтримує декілька прийомів поряд із технічними прийомами, які запроваджуються самим методом.

Підтримка інструментів. Більшість методів не забезпечують спеціальну підтримку інструментів, але це стосується стандартних UML-інструментів. Як зробити власні моделі залишається не відомим. Це дивно, тому що як можна реалізувати моделі без підтримки інструментарію? Цей аспект чітко демонструє що лише Select Perspective та Rational Unified Process, що походять від промислових методів, забезпечують велику інструментальну підтримку.

Платформа впровадження. Усі методи заявляють, що вони не залежать від платформи; іноді запропоновано Java і CORBA. Як наслідок, питання перекладу моделі в код залишається відкритим.

Таким чином можна зробити висновки, що ідея КОР ще не повністю інтегрована в досліджені методи. Замість компонентів підкреслюються інші питання, такі як застосовність методу або краще керований та контрольований процес. Насправді, вивчаючи ці аргументи, здається, що роль компонентів — це лише одна з багатьох нових речей у методі. Концепція компонента погано визначена. Підходи різноманітні і на рівні «функціонального елемента» або «підсистеми». Крім CADA, характеристики компонента не визначені. Як наслідок, прозорий метод розпізнавання компонентів у процесі не може бути забезпечений. Оскільки компоненти є основою КОР, то це серйозна прогалина. Взаємодія між компонентами також не вказана. Три способи заявляють, що з'єднання здійснюється за допомогою інтерфейсів. Детальніше, наприклад, про те, як подолати розрив у функціональних можливостях, які можуть бути в обох компонентах, не надається.

Процес розвитку електронного бізнесу показує, що важливим є і бізнес, і аспекти реалізації. Також існує тісний зв'язок між бізнес-процесами та програмною системою. Проте вказівки для відображення бізнес-потреб обмежені. Три методи КОР пов'язують ці два аспекти разом за допомогою моделі, а два методи лише надають деякі керівні принципи. Всі методи повністю охоплюють основи розвитку електронного бізнесу. Бізнес-процесам та їх реалізації приділяється повна увага в CADA та Select Perspective. Catalysis підкреслює повторне використання знань у процесі розробки та спілкування між аналітиками та розробниками системи. COMET концентрується на бізнес-процесі та повторному використанні, а RUP — на моделюванні системи. Загалом, моделювання має важливе значення в процесі КОР, кожний метод використовує кілька моделей.

Висновки

Сьогодні важливість КОР визнається як у промисловості, так і в академічному світі. Розроблені вдосконалені версії існуючих методологій КОР, а також запроваджені нові методології, такі як UML Components [16] та Kobra [17].

Існує загальне розуміння тісного зв'язку між бізнесом і реалізацією. Встановлено де-факто стандарт моделювання системи в КОР, особливо за допомогою UML, як платформи розробки ООП. Однак з іншого боку можливо зробити висновок, що основні елементи КОР, компоненти та повторне використання далеко не зрілі. Діапазон інструментів для КОР недостатній. Характеристики процесу КОР не вказані. Доступні методи КОР залишають для користувача багато питань, зокрема:

1. Що точно є компонентом?
2. Як відрізнити компонент?

Аналізуючи вищезазначені проблеми існуючих методологій КОР, можна зробити висновок щодо необхідності розробки досконалішого методу КОР, який включав би в себе такі елементи:

1. Стандартний словник;
2. Характеристики компонента;
3. Стандартний опис компонентів;
4. Однозначний процес;
5. Явний спосіб моделювання;
6. Промислові інструменти розробки.

Стандартний словник покращує зв'язок між усіма сторонами, що беруть участь у процесі КОР. Він складається з визначень таких понять, як компонент, рамки та архітектура. Серед компонентів КОР є характеристики компонентів. Вони дозволяють розвивати або розрізнити компонент і судити про повноту компонента. Ця проблема недооцінена в методах. Отже, відсутній ретельний метод розробки компонентів.

Для виявлення характеристик слід дотримуватися підходу «знизу вгору»: розробляти та оцінювати компоненти для конкретної платформи. З колекцією компонентів, спроба вказати характеристики може бути виконана шляхом узагальнення їх властивостей. Як розширення характеристик можна визначити стандартний опис, такий як шаблон компонента. Опис забезпечує, що вся відповідна інформація про компонент записується стандартизованим способом. Це дозволяє розробнику використовувати компоненти невідомих розробників, їх інструменти розробки та платформи. Крім того, стандартні категорії можуть використовуватися для пошуку компонентів. Інструменти можуть підтримувати це, надаючи функціональні можливості для зберігання та вилучення компонентів. Один з методів, Select Perspective, досяг цього етапу; компонентний менеджер є частиною набору інструментів.

Після наявності передумов для доступного сховища компонентів, необхідна наявність однозначного способу для його використання. Сьогодні типовий метод КОР відрізняє діяльність як у бізнесі, так і в процесі впровадження. Інструкції під час процесу часто залишають багато свободи для користувача. Питання, які необхідно вирішити, — це, наприклад, розробка компонентів та монтаж програмного забезпечення з компонентів. Важливим аспектом монтажу є співпраця компонентів. Можливі декілька сценаріїв, наприклад, співпраця через інтерфейс, фреймворк або виклики методу. Ця проблема ледь охоплена досліджуваними методами. Співпраця через інтерфейси коротко згадується три рази; два способи взагалі не обговорюють це. Автори вважають це важливим питанням у успішному КОР. Дослідження використання компонентів на тій же платформі та на різних платформах можуть дати початок для вирішення цих проблем.

Враховуючи кількість моделей, запропонованих розглянутими методами, моделюванню приділяється значна увага в методах КОР. Тим не менше, співвідношення між моделями часто залиша-

ється невизначеним. Щоб визначити, які мови моделювання, відображення між моделями та інструментами необхідні, типовий процес КОР має отримати детальнішу інформацію. На сьогодні дефіцитом є мова моделювання та інструмент для бізнес-моделювання. Всі методи запропонували для цього фірмові технічні прийоми. Таким чином, подальше дослідження має бути направлено на розробку методології КОР, яка би поєднувала переваги існуючих методологій та включала в себе стандартний словник, характеристики компонента, стандартний опис компонентів, однозначний процес, явний спосіб моделювання та промислові інструменти розробки.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ:

- [1] N. Boertien, M. van Steen, and H. Jonkers, "Evaluation of component-based development methods," *Information Modeling Methods and Methodologies*. Idea Group, pp. 323-343, 2005.
- [2] M. Tahir, F. Khan, M. Babar, F. Arif, and S. Khan, "Framework for Better Reusability in Component Based Software Engineering," *Journal of Applied Environmental and Biological Sciences*. TextRoad Publication, pp. 77-81, 2016.
- [3] H. Bouwman, M. De Reuver, S. Solaimani, and D. Daas, "Business models: Tooling and a research agenda," in *25th Bled Conference*. 17-20 June 2012. Bled, Slovenia.
- [4] B. Bozeman, and J. Melkers, *Evaluating R&D impacts: Methods and practice*. New York: Springer Science & Business Media, 2016.
- [5] S. Khwaja, and M. Alshayeb, "Survey on software design-pattern specification languages," *ACM Computing Surveys (CSUR)*, vol. 49, no. 1, pp. 21, 2016.
- [6] De Cesare S., M. G. Lycett, and R. Macredie, *The Development of Component-based Information Systems*. Routledge, 2015.
- [7] I. Crnkovic, M. Larsson, and F. Lüders, *The Different Aspects of Component Based Software Engineering* [Electronic resource], Access mode: <https://pdfs.semanticscholar.org/9c97/79809161d4e7598bc1495ce06006ec0d13bd.pdf>.
- [8] B. Smith "Object-Oriented Design... A Revisit" *Advanced ActionScript 3*. Apress, pp. 357-371, 2015.
- [9] Глоба Л. С., та Т. М. Кот, *Розробка інформаційних ресурсів та систем* [Електронний ресурс]. Київ, Україна: КПІ, 2016.
- [10] A. Anwar, "A review of rup (rational unified process)," *International Journal of Software Engineering (IJSE)*, vol. 5, no. 2, pp. 8-24, 2014.
- [11] Princeton Softech, *Select Perspective: Princeton Softech's practical methodology for delivering next-generation applications*, 2000.
- [12] D'souza D. F., and Wills A. C. *Objects, components, and frameworks with UML: the catalysis approach*. Boston: Addison-Wesley Longman Publishing Co., Inc, 1999.
- [13] J. W. Hubbers, & D. Verhoef, *Workshop component-based development*. Ordina Institute for Research and Innovation, 2000.
- [14] A. J. Berre, B. Elvesæter, J. O. Aagedal, J. Oldevik, A. Solberg, and B. Nordmoen, "Comet (component and model based development methodology)," *COMET Methodology Handbook*, 2004.
- [15] J. Rumbaugh, I. Jacobson, and G. Booch, *The unified modeling language reference manual, Second Edition*. Addison-Wesley Professional, 721 p., 2004.
- [16] T. L. Phan "Modeling and verification techniques for incremental development of UML architectures," in *Doctoral Symposium at ECOOP'2013*, Montpellier, France, June 2013.
- [17] C. Atkinson, J. Bayer, and D. Muthig "Component-based product line development: the Kobra approach," *Software product lines*. New York: Springer Science & Business Media, pp. 289-309, 2010.

Рекомендована кафедрою захисту інформації ВНТУ

Стаття надійшла до редакції 20.12.2017

Чередніченко Ольга Юрївна — канд. техн. наук, доцент, доцент кафедри програмної інженерії та інформаційних технологій управління, e-mail: olha.cherednichenko@gmail.com ;

Гонтар Юлія Миколаївна — аспірант кафедри програмної інженерії та інформаційних технологій управління, e-mail: gontaryn@gmail.com ;

Івашченко Оксана Віталіївна — інженер кафедри програмної інженерії та інформаційних технологій управління, e-mail: Oksana_ivashchenko@ukr.net ;

Вовк Марина Анатоліївна — канд. екон. наук, доцент кафедри програмної інженерії та інформаційних технологій управління, e-mail: marihavovk@gmail.com .

Національний технічний університет «Харківський політехнічний інститут», Харків

O. Yu. Cherednichenko¹
 Yu. M. Gontar¹
 O. V. Ivashchenko¹
 M. A. Vovk¹

Analysis of Component-oriented Methods of Software Developing for E-business Engineering

¹National Technical University “Kharkiv Polytechnic Institute”

Component-based development is a way to develop software, where the benefits of reuse are of great importance. Rapid adaptation to market changes is critical to e-business. A component-oriented development can provide a quick implementation, flexibility, and scalability. Five popular methods are discussed in the paper. They are as follows CADA, Catalysis, Comet, Rational Unified Process, and Select Perspective. Listed methods were estimated in terms of suitability for use in the context of e-business engineering. The concepts, procedures, methods, and level of instrumental support are investigated for each of these methods. The evaluation is based on a scheme for the development of e-business and a list of objective criteria. The criteria are background, methodology, process support, reuse, use of modern simulation techniques, tool support, and implementation platform.

The Select Perspective support process offers a very strict process with clear results and actions. Catalysis is oriented on how to use the method in some specific situations. CADA, COMET, and RUP offer something average between these extremes. Regarding reuse, Select Perspective is the most attractive method in terms of having a component manager, a mechanism for storing and finding components. The rest of the examined techniques depend on the experience of the developers in finding the components. CADA, Catalysis, Comet do not provide special tool support, but this applies to standard UML tools. Select Perspective and Rational Unified Process methods, derived from industrial methods, provide great tool support. All methods depend on the platform. Component's characteristics are important for component-oriented development. They allow developing or distinguishing a component and judging the completeness of the component. This problem is underestimated in the researched methods. Consequently, there is no method of developing components which could satisfy listed elements. The article finds that the main components of component-oriented development, components and reuse are not yet mature, and the range of tools for component-oriented development is insufficient. All of the methods discussed focus on some aspects of component-oriented development. However, none of them offers a complete solution so far.

Keywords: component-oriented development, electronic business engineering, component, evaluation criteria, modeling.

Cherednichenko Olga Yu. — Cand. Sc. (Eng.), Associate Professor, Associate Professor of the Chair of SEMIT, e-mail: olha.cherednichenko@gmail.com ;

Gontar Yulia M. — Post-Graduate Student of the Chair of SEMIT, e-mail: gontaryn@gmail.com ;

Ivashchenko Oksana V. — Engineer of the Chair of SEMIT, e-mail: Oksana_ivashchenko@ukr.net ;

Vovk Maryna A. — Cand. Sc. (Econ.), Associate of the Chair of SEMIT, e-mail: marihavovk@gmail.com

O. Ю. Чередниченко¹
 Ю. М. Гонтарь¹
 О. В. Иващенко¹
 М. А. Вовк¹

Анализ компонентно-ориентированных методов разработки программного обеспечения для электронного бизнеса

¹Национальный технический университет «Харьковский политехнический институт»

Компонентно-ориентированная разработка — это путь к разработке программного обеспечения, в котором преимущества повторного использования имеют большое значение. Быстрая адаптация к изменениям на рынке имеет решающее значение для электронного бизнеса. А компонентно-ориентированная разработка способна обеспечить быстрый срок реализации, гибкость и масштабируемость. С точки зрения пригодности к использованию в контексте инженерии электронного бизнеса рассмотрены пять популярных методов: CADA, Catalysis, Comet, Rational Unified Process и Select Perspective. Для каждого из указанных методов исследованы понятия, процедуры, методы и уровень инструментальной поддержки. Оценка производится на основе схемы для разработки электронного бизнеса и перечня объективных критериев: предпосылки происхождения, использование методологии, функциональная поддержка процесса, повторное использование, использование современных техник моделирования, инструментальная поддержка, платформа внедрения.

Что касается процесса поддержки, Select Perspective предлагает очень строгий процесс с четкими результатами и действиями. Catalysis ограничивается примерами того, как использовать метод в некоторых конкретных ситуациях. CADA, COMET, и RUP предлагают нечто среднее между этими крайностями. Относительно повторного использования Select Perspective является привлекательным способом с точки зрения нали-

чия компонентного менеджера, механизма хранения и поиска компонентов, а остальные исследуемые методики полагаются на опыт разработчиков в поиске компонентов. CADA, Catalysis, Comet не обеспечивают специальную поддержку инструментов, это касается стандартных UML-инструментов, только Select Perspective и Rational Unified Process, которые произошли от промышленных методов, обеспечивают большую инструментальную поддержку. Все методы зависят от платформы. Установлено, что основные элементы компонентно-ориентированной разработки, компоненты и повторное использование пока еще не зрелые, а диапазон инструментов для компонентно-ориентированной разработки недостаточен. Все рассмотренные методы акцентируют внимание на отдельных аспектах компонентно-ориентированной разработки, но пока ни один из них не предлагает полного решения.

Ключевые слова: компонентно-ориентированная разработка, инженерия электронного бизнеса, компонент, критерии оценки, моделирование.

Чередниченко Ольга Юрьевна — канд. техн. наук, доцент, доцент кафедры программной инженерии и информационных технологий управления, e-mail: olha.cherednichenko@gmail.com ;

Гонтарь Юлия Николаевна — аспирант кафедры программной инженерии и информационных технологий управления, e-mail: gontaryn@gmail.com ;

Иващенко Оксана Витальевна — инженер кафедры программной инженерии и информационных технологий управления, e-mail: Oksana_ivashchenko@ukr.net ;

Вовк Марина Анатольевна — канд. экон. наук, доцент кафедры программной инженерии и информационных технологий управления, e-mail: marihavovk@gmail.com