

Інноваційні комп'ютерні технології в освіті та медицині

УДК 004.054

Т.І. Брагіна

Запорізький національний технічний університет, Запоріжжя

РОЗРОБКА ЗАСОБІВ ІНТЕГРАЦІЇ

В статті проведено аналіз методів тестування програмних проектів, що можуть бути використані при інтеграції складних систем, були визначені переваги та недоліки даних методів. Також автором розглянута бібліотека методів тестування, що використовувалась при інтеграції даних університету та бази даних-клієнта «Єдиної державної електронної бази з питань освіти» України.

Ключові слова: керування якістю, методи тестування, інтеграція.

Актуальність

Керування програмними проектами (ПП) – одна областей діяльності в програмній інженерії, які розвиваються найбільш швидкими темпами, що пояснюється високими темпами зростання складності інформаційних систем. Однією з найбільш актуальних областей знань в контексті керування ПП є керування якістю складного, найчастіше інтегрованого проекту, що обумовлено вимогою забезпечення заданої функціональності, визначеної вимогами користувачів [1].

З метою верифікації якості як усього проекту, так і його функціональних частин, використовується тестування. Проблема полягає в тому, що рівень проведеного тестування визначається термінами реалізації продукту, що часто відсуває питання якості на другий план. До того ж процес тестування досить складний і вимагає виділення значних коштів. Особливо гостро дане питання постає при розробці інтегрованого проекту, за рахунок того, що тестувати необхідно не тільки функціональність окремих модулів, але й обмін даними, їх втрати і відповідність. Тому актуальним завданням є визначення оптимальних методів тестування для інтегрованого комплексу програмного забезпечення, тому що результатом недостатнього рівня тестування може стати виявлення безлічі непередбачені проблем на різних етапах розвитку ПП, що призводить до додаткових фінансових втрат.

Постановка задачі

У 2012 році перед кожним вищим навчальним закладом постала необхідність включення до звичного процесу вступної кампанії Єдиної державної електронної бази освіти (ЄДЕБО). Деяка частина навчальних закладів поклатася на використання

web-інтерфейсу, наданого для взаємодії з ЄДЕБО, проте більшість вже мала програмні засоби, які були розроблені під вимоги конкретного закладу. Зважаючи на це, постала мета – інтегрувати існуючі програмні продукти з ЄДЕБО та організувати тестування процесу передачі даних між ними. Для досягнення даної мети необхідно виконати аналіз методів тестування та розробити засоби тестування, які забезпечать контроль збереження даних при інтеграції.

Аналіз методів тестування

Надалі, для аналізу методів тестування, будемо спиратися на стандарт ISO 9126, який виділяє наступні характеристики ПП [1]:

1. Функціональні можливості (придатність, правильність, здатність до взаємодії, узгодженість, захищеність)/
2. Надійність (стабільність, стійкість до помилок, відновлюваність)/
3. Практичність (навченість, простота використання)/
4. Ефективність (характер зміни в часі, характер зміни ресурсів)/
5. Супроводжуваність (аналізований, змінність, стійкість, тестованість)/
6. Мобільність (адаптованість, простота впровадження, відповідність, взаємозамінність).

Для оцінки даних характеристик якості повинні бути встановлені метрики показників, що визначаються наступними методами [2]: вимірний, реєстраційний, органолептичний, розрахунковий та експертний. Тестування об'єднує засоби вимірнього, реєстраційного, органолептичного та розрахункового методів та забезпечує засоби оцінки якості на різних етапах життєвого циклу проекту, наприклад:

1. На етапі формування вимог – приймальне тестування, прототипування, статичне тестування/

2. На етапі проектування – системне і інтеграційне тестування, зворотне проектування, статичне тестування моделі, конфігураційне тестування, об'ємне тестування/

3. На етапі кодування – компонентне тестування, регресійне тестування, функціонального тестування, тестування безпеки, тестування зручності користування/

4. На етапі інтеграції – модульне тестування, інтеграційне тестування, системне тестування.

Зупинимося більш детально на інтеграційному тестуванні, як на невід'ємній частині розробки інтегрованих ПП. Метою інтеграційного тестування є верифікація вимог з функціональності, продуктивності, надійності до основних компонентів програми. Ці компоненти, тобто групи модулів, тестуються методом чорної скриньки, успішні та неуспішні тест-кейси симулюються відповідними вхідними параметрами.

В процесі інтеграційного тестування тестуються симульоване використання спільних даних та комунікація між процесами. Тест-кейси перевіряють чи коректно взаємодіють всі компоненти, наприклад: через виклик процедури або активізацію процесу. Дуже важливо те, що до інтеграційного тестування приступають тільки після модульного. Це дозволяє реалізувати стратегію «будівельних блоків», коли до верифікованої системи інтегруються верифіковані модулі. Серед різновидів інтеграційного тестування є: тестування «великого стрибка», тестування зверху-донизу та знизу-вгору та їх модифікації [3].

Метод "великого стрибка" на сьогодні є чи не найпоширенішим методом інтеграції програмних модулів. Відповідно до цього методу перед інтеграцією кожен модуль тестується автономно. Після тестування всі модулі інтегруються в систему одночасно. Метод "великого стрибка" у порівнянні з іншими методами має чимало недоліків і мало переваг: заглушки та драйвери необхідні для кожного модулю; модулі не інтегруються до останнього моменту, а це означає, що протягом тривалого часу суттєві недоліки та помилки у їх спряженні залишаються невиявленими; одночасна інтеграція всіх модулів суттєво ускладнює ладнання програмного комплексу. Метод "великого стрибка" рекомендується до використання для невеликих за об'ємом програм, за умови їх проектування досвідченими фахівцями.

Тестування за методом "знизу вгору" передбачає збирання та тестування програми "знизу вгору". Лише модулі найнижчого рівня ("термінальні" модулі, що не передають інформацію до інших модулів, а працюють виключно в режимі приймання інформації) тестуються автономно. Після того як тестування цих модулів завершено,

виклик їх має бути гарантованим. Після цього тестуванню піддаються модулі більш високого рівня, що безпосередньо викликають вказані вже перевірені модулі. Ці модулі тестуються не автономно, а разом із вже перевіреними модулями більш низького рівня. За такий спосіб процес повторюється до тих пір, поки не буде зібраним весь програмний комплекс.

Тестування за методом "зверху-вниз" передбачає збирання та тестування програми "зверху вниз". За таких умов ізольовано тестується лише головний модуль. Надалі один за другим підключають модулі, що безпосередньо викликаються головним модулем, і тестується отримана комбінація. Цей процес повторюється до тих пір, поки не будуть зібрані та перевірені всі модулі. При цьому для імітації функцій модулів, що ще не підключені до комплексу, але мають викликатися модулями, що підключаються та тестуються на даному етапі, розробляються спеціальні модулі-заглушки, котрі моделюють функції цих відсутніх модулів. Задача розроблення модулів-заглушок, зазвичай виявляється вельми складною, що є одним із суттєвих недоліком методу тестування "зверху вниз". Поруч з цим слід відмітити й суттєві переваги цього методу, який забезпечує суміщення тестування самого модулю, елементів його з'єднання (спряження) з іншими модулями програмного комплексу та часткове тестування зовнішніх функцій.

Тестування за методом сандвіча є компромісом спрямованим на використання переваг методів тестування "знизу вгору" та "зверху вниз". Під час використання цього методу одночасно починають тестування (а відповідно й збирання) програмного комплексу й "знизу вгору" й "зверху вниз". За такий спосіб інтеграції модулів "точка зустрічі" знаходиться у середині програмного комплексу і має бути визначена заздалегідь, на підставі прискіпливого аналізу структури всієї програми. Метод сандвіча, зазвичай, використовується під час проектування значних за об'ємом програм.

Проаналізувавши розглянуті методи тестування та особливості взаємодії з ЄДЕБО був зроблений наступний висновок – не використовувати методи тестування "великого стрибка" та "зверху вниз" тому, що немає можливості визначити кількість модулів та масштаби системи через поступове наповнення бази даних та триваючу розробку ЄДЕБО. З цієї ж причини неможливо заздалегідь визначити "точка зустрічі" для методу сандвіча. Зважаючи на це, було прийняте рішення – з метою автоматизування процесу підключення та коректного обміну даними з ЄДЕБО розробити бібліотеку методів тестування для інтеграції клієнта ЄДЕБО відповідно до методу інтеграційного тестування "знизу-вгору".

Бібліотека методів тестування

Комп'ютерна програма «Бібліотека методів тестування» призначена для тестування взаємодії через Simple Object Access Protocol (SOAP) – інтерфейс клієнта з web-сервером ЄДЕБО. В представленому програмному засобі реалізовані методи тестування помилок при інтеграції з ЄДЕБО, які спрямовані на підтримку бази даних у актуальному стані, перевірку кількості втрачених даних та причин втрати.

Програмний комплекс розроблений за принципами клієнт-серверної архітектури, в якій серверна частина містить три основні web-сервіси:

- для керування обліковими записами;
- для роботи з довідниками ЄДЕБО;
- для роботи з персонами (абітурієнтами, студентами).

Для роботи с web-сервісами використовується протокол SOAP, що підтримується усіма основними платформами розробки програмного забезпечення та дозволяє виконувати обмін даними в форматі XML.

Клієнтська частина дозволяє адміністратору приєднатись до діючого сервера, ідентифікувати себе серед користувачів та прийняти участь в процесі обміну даними. Фактично, клієнтська частина є базою даних MS Access з модулями, макросами та класами модулів, написаних на мові Visual Basic for Application.

Взаємодія між локальною базою даних та web-сервісом здійснюється за допомогою класів модулів `clsOf_Factory_EDBOGuides` та `clsOf_Factory_EDBOPerson`. Взаємодія між модулями здійснюється через розроблені інтерфейси взаємодії. На рис. 1 показана фізична діаграма та на рис. 2 функціональна схема програми.

Вхідні дані для серверної частини: логін та пароль доступу до бази даних ЄДЕБО. Вхідні дані для клієнтської частини: логін адміністратора, налаштування підключення та розташування серверної частини системи (адрес та порт).

Вихідними даними є отримані або переслані до ЄДЕБО дані, відомості щодо втрат даних та помилок, що виникли у процесі взаємодії з ЄДЕБО.

Робота з серверною частиною виконується лише за умови підключення захищеного з'єднання з web-сервісом ЄДЕБО-клієнт.

На головній формі (форма «EDBO») представлені основні функції роботи адміністратора з ЄДЕБО:

1. «Додати до ЄДЕБО» – при натисненні цієї кнопки відбувається відправка усіх облікових записів абітурієнтів з запиту «EDBO» до Єдиної державної бази з питань освіти. При цьому тестується підключення до ЄДЕБО, повнота передачі даних, відповідність даних та формату даних.

У разі помилки підключення жоден запис не буде передано, тому метою тестування є з'ясування причини збою.

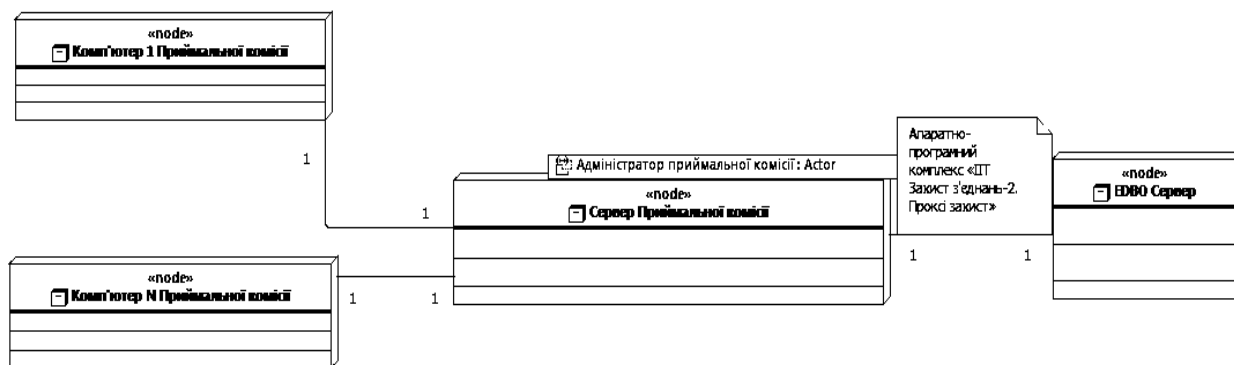


Рис. 1. Фізична діаграма програми

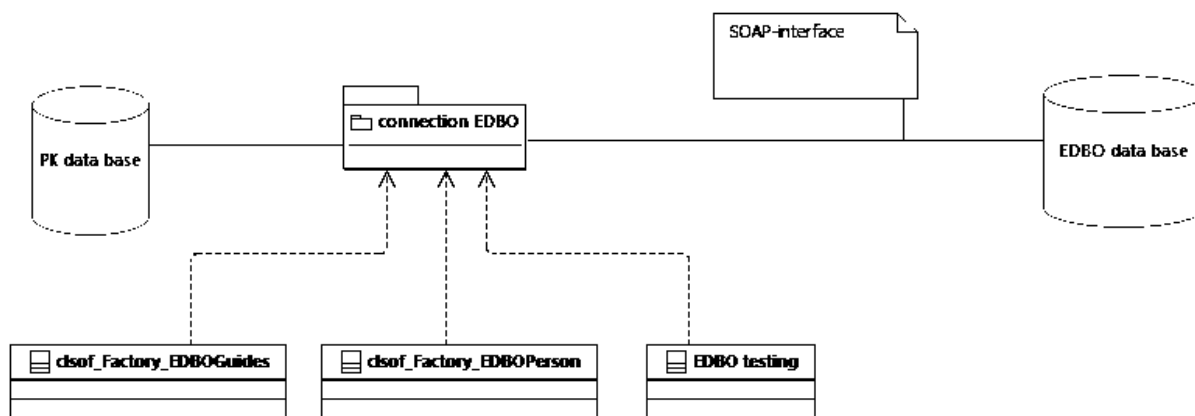


Рис. 2. Функціональна схема програми

Можливі варіанти:

- гередається невірний логін або пароль (реакція – перевірка відповідності логіну та паролю, які передаються у функції Login(string User, string Password, int ClearPreviewSession));

- уа сервері ЄДЕБО ведуться технічні роботи (реакція – очікування відновлення роботи спроможності серверу).

Повнота передачі даних тестується наступним чином – ідентифікатори записів, які не були передані, записуються до таблиці «false_EDEBO» з метою подальшої обробки. Відповідність даних та формату даних перевіряється за допомогою функцій тестування та виправлення.

2. «Внести id request заявам» – тестування передачі заяв абітурієнтів. За умови втрати контакту під час оновлення локальної бази даних, заявка зчитується повторно та у локальній таблиці заповнюється поле «Id_PersonRequest» таблиці MAIN, яка містить усі заяви абітурієнтів.

3. «Зчитати заяви електронного вступу» – тестування передачі даних у напрямку від системи Електронного вступу до локальної бази даних приймальної комісії. Тестування спрямовано на перевірку відповідності наповнення даними основних таблиць бази даних. Кожне поле даних таблиць має власний формат, тому необхідно тестувати перед внесенням кожен запис. Помилки при зчитування інформації з ЄДЕБО заносяться до відповідних таблиць.

4. «Отримати контактні дані абітурієнтів» – тестується втрата інформації про контактні дані абітурієнтів та оновлюється інформація у разі знаходження невідповідності;

5. «Додати бали курсів» – тестується втрата інформації про додаткові бали абітурієнтів та оновлюється інформація у разі знаходження невідповідності;

6. «Додати результати іспитів» – тестується втрата інформації про результати іспитів абітурієнтів та оновлюється інформація у разі знаходження невідповідності;

7. «Оновити оригінали документів про освіту» – тестується втрата інформації про оригінали докумен-

тів про освіту абітурієнтів та оновлюється інформація у разі знаходження невідповідності;

8. «Змінити статус заяв», «Змінити статус заяв – реком/не допуск», «Змінити коди вступу» – тестується втрата інформації про рекомендацію абітурієнтів та оновлюється інформація у разі знаходження невідповідності.

Висновки

У статті були розглянуті методи тестування при розробці та інтеграції складних програмних проектів, визначено їх переваги і недоліки. Наукова новизна роботи полягає у виконаному порівняльному аналізі методів тестування, в результаті якого була обрана методика інтеграційного тестування та розроблена бібліотека методів тестування ЄДЕБО. В програмі реалізовані методи тестування за принципом "знизу-вгору", які спрямовані на підтримку клієнтської бази даних при інтеграції з ЄДЕБО у актуальному стані, перевірку кількості втрачених даних та причин втрати. Планується розширити розроблену бібліотеку тестування засобами аналізу відмов для подальшої оптимізації роботи з ЄДЕБО.

Список літератури

1. Брагина, Т.И. Сравнительный анализ итеративных моделей разработки программного обеспечения / Т.И. Брагина, Г.В. Табуницкий // Радиоэлектроника. Информатика. Керування. – 2010. – № 2. – С. 130 – 139

2. ГОСТ 28195-89 Оценка качества программных средств. Общие положения: [Электрон. ресурс]. – Режим доступа: <http://www.gosthelp.ru/gost/gost11212.html>

3. Мірських, Г.О. Порівняльний аналіз методів тестування програмного забезпечення радіоелектронних засобів / Г.О. Мірських, Ю.Ю. Реутська // Вісник Національного технічного університету України "КПІ". Серія – Радіотехніка. Радіоапаратобудування. - 2011. - № 44. - С. 58-67.

Надійшла до редколегії 1.10.2012

Рецензент: д-р техн. наук, проф.. С.І. Гоменюк, Запорізький національний університет, Запоріжжя.

INTEGRATION TOOLS DEVELOPMENT

Т.И. Брагина

В статье проведен анализ методов тестирования программных проектов, которые могут быть использованы при интеграции сложных систем, были определены преимущества и недостатки данных методов. Также автором рассмотрена библиотека методов тестирования, используемой при интеграции данных университета и базы данных-клиента «Единой государственной электронной базы по вопросам образования» Украины.

Ключевые слова: управление качеством, методы тестирования, интеграция.

FUZZY RISK PROJECT ANALYSIS

Т.И. Брагина

Software projects testing methods that can be used to integrate complex systems were analyzed, advantages and disadvantages of these methods were identified. Also, the testing methods library, used in interaction between database-client and Unified electronic education database in Ukraine was examined by the author.

Keywords: quality management, testing methods, integration.