

Методика навчання тестування програмних продуктів студентів напрямку підготовки «Інформатика»

На сьогоднішній день існує велика кількість програмних засобів, призначених для тестування. У процесі навчання напряму підготовки «Інформатика» студенти на заняттях реалізують різного роду програми. Наприклад, веб-сайти з дисципліни «Програмування та підтримка веб-застосувань», програми з дисципліни «Об'єктно-орієнтоване програмування» та ін.

Фундаментальні концепції тестування були розглянуті в роботах Бориса Бейзера, Луїзи Тамра, Джона Макгрегора, Девіда Сайкса, Сема Канера та ін. вчених. Наприклад, у роботі доктора Бейзера «Тестування чорного ящика» [1, 34-76] розглядаються основні питання тестування програмного забезпечення, що дозволяє відшукати максимум помилок при мінімумі витрат часу. У роботі Луїзи Тамра «Тестування програмного забезпечення» [2, 105-134] розглядаються і ключові процедури, що виконуються на ранніх етапах тестування програмного забезпечення (ПЗ). У роботі Джона Макгрегора «Тестування об'єктно-орієнтованого програмного забезпечення» [3, 41-115] велика увага приділяється таким питанням як: тестування аналітичних і проектних моделей, тестування ієрархії спадкування, тестування класів, тестування взаємозв'язків між об'єктами, тестування розподілених об'єктів, ефективний вибір тестових наборів.

Тестування ПЗ – це процес технічного дослідження, який виконується на вимогу замовників, і призначений для виявлення даних про якість продукту відносно контексту, в якому він має використовуватись [1, 5].

Дуже важливо сформулювати у студентів в процесі реалізації програм навички тестування проекту. Для цього студентів необхідно навчати:

1. Планування тестування ПЗ;
2. Підготовки ПЗ для тестування;
3. Визначення правильності функціонування ПЗ;
4. Визначення помилок ПЗ з найменшою кількістю витрат;
5. Оцінювання якості вимог до ПЗ;
6. Складання тестової документації.

Для наочного уявлення проведення тестування програмних продуктів наведено якісний аналіз продуктів фірми Microsoft – **MS Test Manager** (Visual Studio 2010) та **TestComplete** фірми SmartBear Software [4], [5].

MS Test Manager – програма вдосконалення програмного забезпечення, використання якої дозволяє проводити тестування програм та які містить такі відомості [6]:

1. Конфігурація – кількість процесорів в комп'ютері користувача, кількість використовуваних мережових підключень, використовувана операційна система, дозволи екранів пристроїв відображення, потужність бездротового сигналу між комп'ютером і пристроєм відтворення медіафайлів, деякі особливості, наприклад застосування технології бездротового з'єднання Bluetooth або високошвидкісних USB-підключень.

2. Продуктивність і надійність – час відгуку програми при натисканні кнопки, кількість проблем при роботі з програмою або пристроєм, а також швидкість передавання і приймання даних через мережеве підключення.

3. Використання програми – найбільш часто використовувані функції, частота запуску програм, а також кількість створених на робочому столі папок.

Наведемо приклад використання Microsoft Test Manager в рамках дисципліни «Програмування та підтримка веб-застосувань», яка вивчається на п'ятому семестрі напряму підготовки «Інформатика». Програма MS Test Manager функціонує спільно з встановленим сервером Team Foundation Server. Студенти отримують проектне завдання. Далі команда здійснює розробку отриманого проекту, спираючись на основні етапи розробки: 1. Визначення процесу розробки ПЗ; 2. Управління проектом розробки; 3. Опис цільового програмного продукту; 4. Проектування продукту; 5. Розробка продукту; 6. Тестування модулів; 7. Інтеграція частин продукту; 8. Супровід продукту [7, 35-110], [8, 44-65]. Розглянемо детальніше процес тестування з використанням MS Test Manager. Для цього рекомендується орієнтуватися на наступні кроки:

1. Обґрунтування чітких цілей тестування та узгодження їх з усіма учасниками проекту.
2. Визначення плану тестування, виходячи з описів функціональності користувачів, проектних припущень і нефункціональних вимог у поточному спринті.

3. Визначення та створення тестових випадків, таких як приймальні тести, модульні тести, функціональні тести і тести продуктивності.

4. Групування тестових варіантів в набори тестів.

5. Виконання наборів тестів, що містять тестові варіанти.

При запуску програми MS Test Manager перший раз студент не підключений до жодного командного проекту. Спочатку необхідно підключитися через локальну мережу до сервера Team Foundation Server, а потім вибрати колекцію проектів і командний проект. При підключенні слід вибрати командний проект, який використовується іншими членами команди для створення вимог або описів функціональності користувачів, підтримки вихідного коду і побудови тестових додатків. Створення артефактів тестування та управління ними здійснюється в тому ж командному проекті.

На рис. 1 показаний процес підключення до командного проекту.

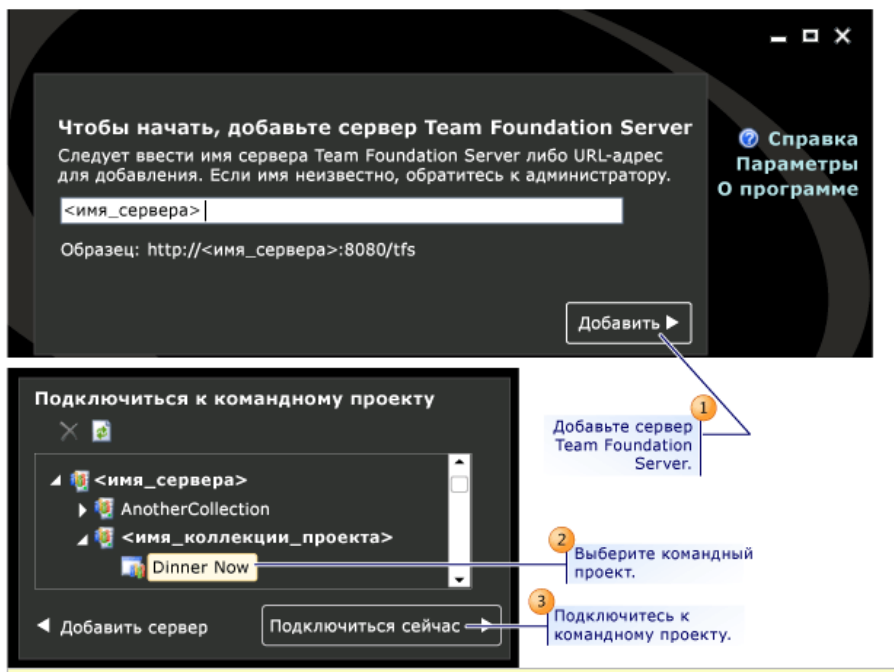


Рис. 1. Підключення командного проекту

Потім необхідно створити або вибрати план тестування (рис. 2). Плани тестування використовуються в командних проектах для планування та виконання тестів.

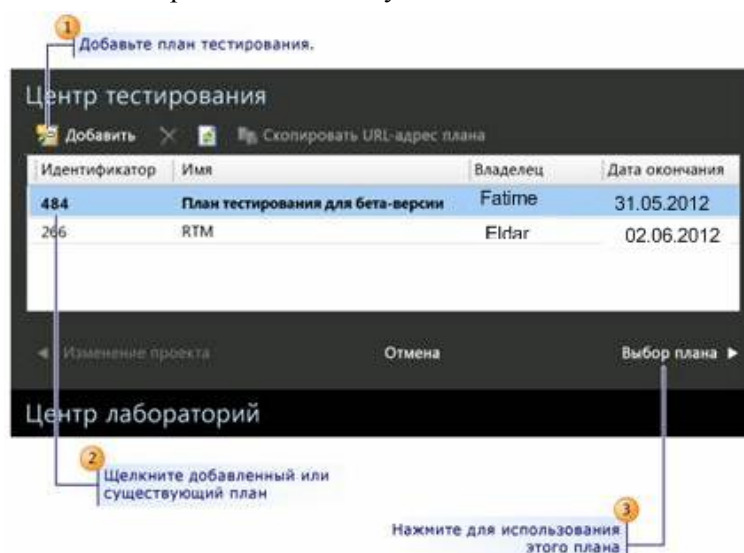


Рис. 2. План тестування

Таким чином, за допомогою програми вдосконалення програмного забезпечення створюється ідентифікатор GUID, який зберігається на комп'ютері студента і визначає його унікальний номер. Цей ідентифікатор представляє собою створений випадковим чином номер, який не містить жодних особистих даних і не використовується для встановлення особи користувача. В програмі вдосконалення програмного забезпечення використовуються ідентифікатори GUID для визначення ступеня поширеності одержуваних від користувачів відгуків та встановлення пріоритетів. Наприклад,

за допомогою такого ідентифікатора можна з'ясувати, що один клієнт зіткнувся з проблемою 100 разів, а іншим клієнтам ця проблема зустрілася тільки одного разу. Ідентифікатор GUID зберігається в пам'яті комп'ютера студента і дані про нього відправляються разом з кожним звітом програми вдосконалення програмного забезпечення. Звіти програми вдосконалення програмного забезпечення навмисно не містять контактних даних, таких як ім'я учасника, його адресу або номер телефону [6].

Test Complete – це інструмент для автоматизації тестування, створення та виконання тестів для будь-яких Windows і веб-програм різної функціональності. Автоматизовані тести дозволяють прискорити роботу, охопити більше областей тестування програми і помітно знизити витрати.

Процес проходження тестів супроводжується створенням докладних звітів для кожного тесту, що містять у собі час виконання подій, а також коментарі до кожного етапу.

Використання пакета TestComplete дозволяє управляти виконанням і синхронізувати тести, запущені на різних робочих станціях.

Функціональні характеристики [5]:

- Легке у використанні keyword тестування, яке не потребує створення скриптів;
- Створення потужних і гнучких тестових скриптів;
- Тестування Windows і веб-додатків;
- Підтримка .NET, ASP.NET, WPF, Silverlight та XBAP програм;
- Підтримка Adobe AIR, Flex і Flash програм;
- Підтримка Java і JavaFX програм;
- Підтримка Ajax, ASP, PHP та інших типів веб-програм;
- Підтримка різних Інтернет браузерів, включаючи останні версії Internet Explorer 8 і Firefox 3.6;
- Підтримка операційних систем Windows 7, Vista, XP, 2000, Windows Server 2003 і 2008;
- Підтримка телефонів і емуляторів на базі Windows Mobile;
- Підтримка як 32-бітних, так і 64-бітних додатків;
- Підтримка популярних систем відслідковування помилок (bug tracking system): Software Planner, Axosoft OnTime, Atlassian JIRA та інших;
- Відкритий API;
- Легка розширюваність.

Підтримувані види тестування [9, 56]:

- Функціональне тестування (functional testing);
- Модульне тестування (unit testing);
- Навантажувальне тестування (load testing);
- Тестування клієнт-серверних додатків або розподілене тестування (client / server or distributed testing);
- Регресійні тестування (regression testing);
- Тестування, залежне від даних (data-driven testing).

Наведемо приклад використання пакету TestComplete з дисципліни «Програмування та підтримка веб-застосувань». Студенти отримують завдання на створення веб-програм, після завершення етапу проектування і написання програми студенти переходять до етапу тестування. Для цього створюється новий проект в пакеті TestComplete, вибирається мова програмування, якою буде описаний тестуючий скрипт. Створюється сам скрипт і проводиться процес тестування програми. Далі за отриманими результатами оцінюються показники ефективності коду програми.

Перший крок при запуску TestComplete, вибір проекту (рис. 3).

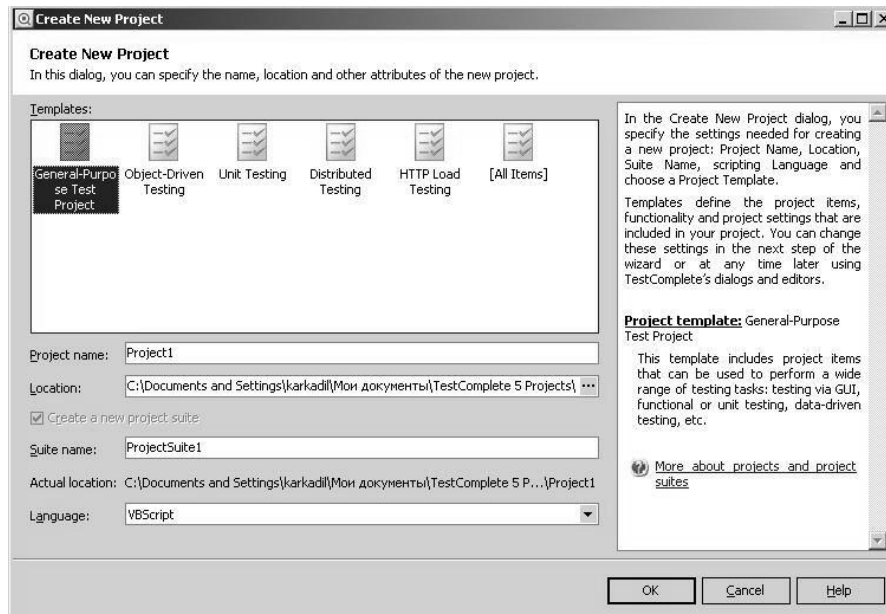


Рис. 3. Вибір проекту

В опції Language (Мова) підтримуються 5 скриптових мов: VBScript, JScript, C ++ Script, C# Script і Delphi Script. Мова проекту вибирається в момент його створення і не може бути змінена пізніше. Мова проекту може відрізнятися від мови, якою описана тестуюча програма, існує лише рекомендація щодо створення самотестованої програми, для якої мови повинні збігатися.

Вікно нового проекту виглядає наступним чином (рис. 4).

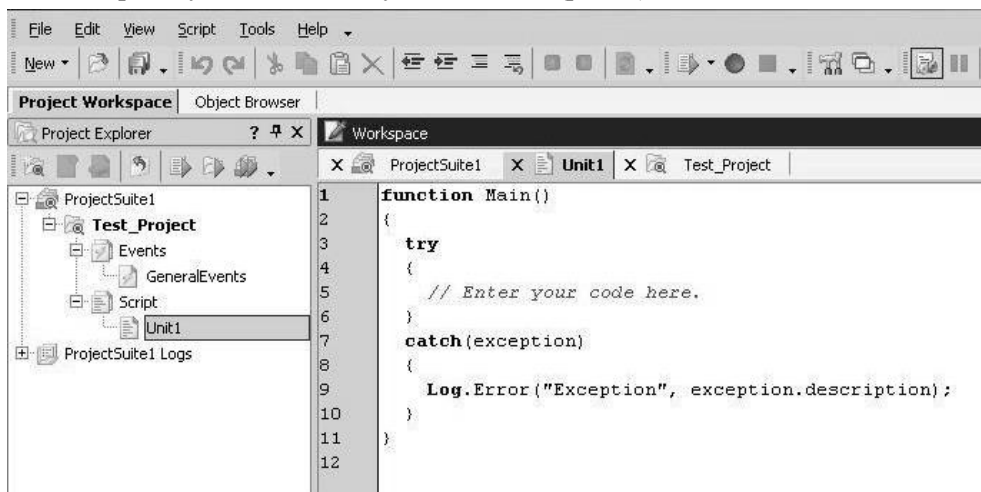


Рис. 4. Вікно створення нового проекту

Після завершення тестування створюється звіт (рис. 5), в якому будуть відображені всі виконані дії.

Test Log	
Type	Message
	The window was clicked with the left mouse button.
	Keyboard input.
	The menu item 'Edit Select All' was clicked.
	The menu item 'Edit Delete' was clicked.
	The menu item 'Edit Undo' was clicked.
	The window was clicked with the right mouse button.
	The menu item 'Copy' was clicked.

Рис. 5. Звіт TestComplete

За допомогою створених за допомогою TestComplete тестів, можна здійснювати: вимірювання продуктивності програмного забезпечення та серверів, перевірку бізнес-логіки програми, аналізувати бази даних, оцінювати якість інтерфейсу додатків з точки зору користувача.

Зведемо вищеописане в таблицю 1.

Таблиця 1

Назва продукту	MS Test Manager	Test Complete
З якими ПЗ використовується	Team Foundation Server	Не потрібно
Практична цінність	Поліпшення якості програмного забезпечення, за допомогою визначення ступеня поширеності одержуваних від користувачів відгуків та встановлення пріоритетів	Підтримка видів тестування, включаючи: Функціональне тестування (functional testing) Модульне тестування (unit testing) Навантажувальне тестування (load testing) Тестування клієнт-серверних додатків або розподілене тестування (client / server, or distributed testing) Регресійне тестування (regression testing) Тестування, залежне від даних (data-driven testing)
Звітні відомості	1. Конфігурація; 2. Продуктивність; 3. Надійність; 4. Використання.	1. Продуктивність; 2. Надійність; 3. Використання.
Навчальні дисципліни, при вивченні можуть використовуватися дані продукти	«Програмування та підтримка веб-застосувань», «Архітектура обчислювальних систем», «Хмарні технології», «Мова програмування Java».	«Програмування мовою python», «Програмування мовою C++», «Об'єктно-орієнтоване програмування», «Програмування та підтримка веб-застосувань», «Мова програмування Java».

Test Manager і TestComplete – зручні у використанні цілісні закінчені системи, які можуть бути широко застосовні як початківцями тестувальниками, так і професіоналами. Використовуючи Visual Studio Test Manager і TestComplete, команда студентів може постійно мати уявлення про стан планування та виконання тестів.

Розглянуті засоби MS Test Manager і TestComplete можна використовувати у вищих навчальних закладах на заняттях зі спеціальності «Інформатика», з дисциплін: «Програмування та підтримка веб-застосувань», «Архітектура обчислювальних систем», «Хмарні технології», «Мова програмування Java», «Програмування мовою Python» та ін. для проведення якісного аналізу розроблених систем.

Тестування програмного забезпечення є невід'ємною частиною в розробці програмного забезпечення. Виходячи з цього стає очевидною необхідність вивчення студентами спеціальності «Інформатика» всіх аспектів тестування. Для успішного тестування програмного забезпечення студенти повинні розуміти принципи роботи тестуючих пакетів і бути ознайомлені з необхідною літературою в цій галузі.

У процесі тестування програмного забезпечення у студентів формуються такі навички, як: вміння складати план тестування, виявлення помилок ПЗ, вміння виправляти помилки ПЗ, оптимізація коду програм, виявлення різних недоліків у кодї ПЗ, робота в команді, оцінювання якості ПЗ, складання документального супроводу ПЗ, застосування різних засобів тестування (наприклад, Test Manager і TestComplete). Отримані знання, вміння, навички студенти спеціальності «Інформатика» можуть використовувати в майбутньому для роботи тестувальниками.

Література

1. Бейзер Б. Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем/ Б. Бейзер. – СПб.: Питер, 2004. – 320 с.
2. Тамре Л. Тестирование программного обеспечения/ Л. Тамре. – М.:ООО «И.Д. Вильямс», 2003. – 354 с.
3. Макгрегор Дж. Тестирование объектно-ориентированного программного обеспечения/ Дж. Макгрегор, Д. Сайкс. – М.: «ДиаСофт», 2002. – 417 с.
4. Сайт фірми Microsoft Visual Studio 2010 [Електронний ресурс]. – Режим доступу: <http://www.microsoft.com/products>
5. Сайт фірми Smartbear [Електронний ресурс]. – Режим доступу: <http://smartbear.com/products/qa-tools/automated-testing/pricing/>
6. Електронна бібліотека фірми Microsoft Visual Studio 2010 [Електронний ресурс]. – Режим доступу: <http://msdn.microsoft.com/ru-ru/library>

7. Брауде Э. Технология разработки программного обеспечения /Э. Брауде. – Спб.: Питер, 2004. – 655 с.
8. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений. Пер. с англ./ Г. Буч, Р. Максимчук, М. Энгл. – 3-е изд. – М.:ООО «И.Д. Вильямс», 2008. – 720 с.
9. Блэк Р. Ключевые процессы тестирования. Планирование, подготовка, проведение, совершенствование. Пер. с англ./ Р. Блэк. – М: «Лори», 2011. – 456 с.