

# Актуальні питання навчання

УДК 681.324:623.618(06)

С.В. Дуденко, В.В. Калачова, С.В. Алексєєв, М.М. Колмиков

*Харківський університет Повітряних Сил імені Івана Кожедуба, Харків*

## АНАЛІЗ ТА ДОСЛІДЖЕННЯ ІНСТРУМЕНТІВ АВТОМАТИЗАЦІЇ ТЕСТУВАННЯ ПРИ РОЗРОБЦІ КОМПЛЕКСУ ПРОГРАМ АВТОМАТИЗОВАНОЇ СИСТЕМИ КОНСТРУЮВАННЯ РОЗКЛАДУ ЗАНЯТЬ

*В поточний період в ХУПС ведеться розробка комплексу програм автоматизованої системи конструювання розкладу занять. Успішність проведення тестувальних робіт багато в чому залежить від застосування автоматизованих засобів. Аналіз та дослідження інструментів автоматизації проведення тестувальних робіт є одним з найважливіших етапів розробки комплексу, який дасть можливість обґрунтовано підійти до визначення найбільш привабливого варіанту з множини можливих та в результаті отримати високоякісний програмний продукт, який буде відповідати цільовому рівню вимог замовника.*

**Ключові слова:** розклад навчальних занять, якість програмного забезпечення, автоматизація тестування, тестовий набір, автоматизований тест-скрипт.

### Вступ

**Постановка проблеми.** Чітка організація навчального процесу у вищому військовому навчальному закладі є гарантом успіху підготовки висококваліфікованих військових фахівців та може бути досягнута лише при якісному його плануванні. Одним з найбільш відповідальних, трудомістких і складних завдань планування навчального процесу є складання семестрових розкладів навчальних занять (екзаменів). Трудомісткість завдання складання розкладу обумовлена участю багатьох представників різних ланок управління, планування і забезпечення навчального процесу при підготовці, обробці і використанні великої кількості нормативної, навчальної, методичної документації та іншої інформації. Тому великі часові витрати, помилки та суб'єктивизм – є тим не повним переліком недоліків, що змушує звертатися до наукоємних інформаційних технологій, які взмозі звести до мінімуму існуючі проблеми шляхом автоматизації процесу конструювання розкладу занять.

У поточний період в ХУПС ведеться розробка комплексу програм автоматизованої системи конструювання розкладу занять. Заключним етапом розробки комплексу має стати тестування, яке має на меті звести кількість помилок програмного продукту до мінімуму, шляхом їх виявлення та подальшого вилучення. Спроба підвищити швидкість проведення тестувальних робіт, збільшити вірогідність виявлення дефектів та посприяти зниженню собівартості процедури тестування напряму зв'язана з його автоматизацією.

Сьогодні на ринку програмного забезпечення представлена велика кількість програмних продуктів, реалізуючих процедуру автоматизованого тестування. У зв'язку з цим, актуальним стає питання

аналізу та дослідження інструментів автоматизації проведення тестувальних робіт, яке є одним з найважливіших питань при введенні в постійну експлуатацію комплексу програм автоматизованої системи конструювання розкладу занять ХУПС, а всебічний аналіз можливостей програмних продуктів для здійснення цієї процедури відкриває шлях до обґрунтованого вибору найбільш привабливого варіанту з множини можливих, що дозволить вдало провести тестування та отримати високоякісний програмний продукт, який буде відповідати цільовому рівню вимог замовника.

**Аналіз останніх досліджень і публікацій.** Відомості щодо питання впровадження новітніх інформаційних технологій у навчальний процес в вищих військових навчальних закладах ЗС України докладно розкрито в [1 – 4]. Характеристики програмних продуктів для автоматизації тестування надані в [5 – 10].

**Формулювання мети статті.** Проведення порівняльного аналізу представлених на ринку програмного забезпечення продуктів для здійснення автоматизованого тестового контролю якості додатків, та взагалі, дослідження основних принципів та особливостей застосування автоматизації в подібних процесах, має на меті надати тестувальникам та розробникам комплексу програм автоматизованої системи конструювання розкладу занять, що зараз розробляється в Харківському університеті Повітряних Сил, можливість здійснення обґрунтованого вибору програмного продукту для виконання процедури тестування розробки, що дасть можливість врахувати всі вимоги замовника та гарантує досягнення максимальної якості і оптимальності програмного коду.

## Виклад основного матеріалу

У зв'язку з тим, що більшість дефектів у програмному забезпеченні виявляється на стадії тестування, визначальною для економії коштів стає процедура її автоматизації. Автоматизація тестувальних робіт дозволяє зменшити собівартість розробки і підвищити вірогідність виявлення дефектів.

На ринку програмних засобів існує велика кількість продуктів, якісно вирішуючих задачу автоматизованого тестування різних видів (функціонального, дослідницьке, тестування навантаження та інш.), серед яких є безкоштовні та комерційні. До головних аспектів, що впливають на вибір конкретного інструменту автоматизації тестування з великої множини існуючих можна віднести наступні рекомендації:

По-перше, треба звернути увагу наскільки добре інструмент для автоматизації розпізнає елементи управління додатку, що тестується. У разі, коли елементи не розпізнаються, варто шукати плагін, або відповідний модуль. Якщо такого немає – від інструменту краще відмовитися. Чим більше елементів може розпізнати інструмент – тим більше часу заощаджується на написання і підтримку скриптів.

По-друге, треба звернути увагу на те скільки часу потрібно на підтримку скриптів написаних за допомогою вибраного інструменту. Для цього треба записати простий скрипт, який вибирає пункт меню, а потім уявити, що змінився пункт меню, який необхідно вибрати. Якщо для відновлення працездатності сценарію доведеться перезаписати скрипт цілком, то інструмент не оптимальний, оскільки реальні сценарії набагато складніші. Краще всього той інструмент, який дозволяє вам винести назву кнопки в змінну на початку скрипта і швидко замінити її значення. У ідеалі – описати меню як клас.

По-третє, потрібно звернути увагу – наскільки зручний інструмент для написання нових скриптів. Скільки вимагається на це часу, наскільки можна структурувати код, наскільки код можна читати, наскільки зручне середовище розробки для рефакторингу (переробки коду) і тому подібне.

Вибір інструменту часто також залежить від об'єкту тестування і вимог до тестових сценаріїв, оскільки інструменти тестування не можуть підтримувати абсолютно всі технології, що використовуються при розробці додатків.

Тобто, вибір інструменту зводиться до методу проб і помилок. У результаті, нерідко вибирається декілька інструментів для тестування функцій додатку. Наприклад, GUI можуть перевірятися застосовуючи засоби Mercury WinRunner, а бекенд процеси – використовуючи "Java based test tools" або інші інструменти.

Потреба в перевірці продуктивності програмного продукту зводиться до тестування навантажен-

ня, метою якого є перевірка роботи системи з максимальним значенням цього параметра. До комерційних за цим напрямком продуктів відносяться:

- HP Performance Center (включає HP LoadRunner) від Hewlett-Packard (Mercury Interactive);
- Rational Performance Tester від IBM Rational;
- Silk Performer від Borland (Segue);
- LoadComplete Web Load Testing від SmartBear;
- Neotys від NeoLoad.

До безкоштовних інструментів автоматизованого тестування навантаження можна віднести Jmeter, Grinder та інші.

Тестування програмного додатку з метою перевірки його функціональних можливостей проводиться за допомогою інтерфейсу розробки. Перевірка ведеться безвідносно до внутрішньої логічної структури програми і тільки на підставі функціональних вимог до програмного продукту та вимог до окремих його властивостей. Це дозволяє отримати відповіді на ряд ключових питань: чи вирішуються поставлені задачі якісно та у повному обсязі; чи сприймаються вхідні дані належним чином, а вихідні – вірно генеруються.

При розробці комплексу програм автоматизованої системи конструювання розкладу занять в Харківському університеті Повітряних Сил, аналіз та дослідження особливостей програмних продуктів автоматизації функціонального тестування, представлених на ринку програмного забезпечення, має визначальне значення, враховуючи всю багату палітру функцій, що реалізуються комплексом. До інструментів тестування, представлених на ринку програмних засобів та заслуговуючих на увагу у вище наведеному контексті по праву можна віднести:

- QuickTest Professional, WinRunner від Hewlett-Packard (Mercury Interactive);
- Rational Robot, Rational Functional Tester від IBM Rational;
- SilkTest від Borland (Segue);
- TestComplete від AutomatedQA Corp;
- Selenium від SeleniumHQ;
- Microsoft VS 2010/2012 від Microsoft.

Засіб автоматизації HP QuickTest Professional від компанії Hewlett-Packard розповсюджується на платній основі (8000-10000 USD). Він є основним інструментом автоматизації функціонального тестування від даного виробника та дозволяє автоматизувати функціональні та регресійні тести через запис дій користувача під час проведення тестувальних робіт над додатком, і у подальшому використовувати записані дії для перевірки працездатності програмного забезпечення. Записані дії зберігаються у вигляді скриптів. У HP QuickTest Professional скрипти можуть бути відображені як VBScript (expert view), або як візуальні послідовні кроки з діями (keyword view). Кожен крок може бути відредагований і на нього можна додати точки

перевірки (checkpoint), які порівнюють очікуваний результат з тим, що одержується.

IBM Rational Functional Tester (RFT) платний додаток, але не настільки коштовний в порівнянні з попереднім (6000 USD). Rational Functional Tester надає тестувальникам засоби автоматизованого тестування, які дозволяють виконувати функціональне і регресивне тестування, тестування інтерфейсу користувача і тестування кероване даними. Для створення функціонального тесту засобами Rational Functional Tester (використана інтеграція IBM Rational Functional Tester з середовищем розробки Microsoft Visual Studio) необхідно виконати наступні дії:

1) у середовищі розробки Microsoft Visual Studio створюється новий проект «Functional Test Project» (рис. 1);

2) виконується запис призначених для користувача дій з тестовим додатком;

3) створюється перевірна точка в процесі виконання запису;

4) зберігаються результати запису;

5) формується bat-файл, який викликає скрипт тестування на виконання і перевіряє результат. Bat-файл виконує виклик IBM RFT;

6) IBM RFT записує свої результати в звіт у форматі HTML. Для того, щоб визначити чи був провалений хоч один крок в процесі виконання скрипта тестування, знаходиться слово «failed» в звіті. Залежно від результату пошуку повертається результат 0 або 1.

7) Результат виконання тестування має вигляд наведений на рис. 2.

Selenium є безкоштовним пакетом від компанії OPENQA. В основі Selenium лежить середовище для тестування web-додатків, реалізоване на JavaScript, що виконує перевірки безпосередньо засобами браузера. В рамках проекту Selenium випускається три інструменти – Selenium Core, Selenium IDE, Selenium RC і Selenium GRID, кожен з яких має свої особливості та сферу застосування. До технологій, що підтримуються Selenium належать DHML, JavaScript, Ajax. Пакетом підтримуються такі операційні системи, як MacOS, Microsoft Windows, Linux, Solaris. Мовами ж тестів Selenium є HTML, Java, C#, Perl, PHP, Python і Ruby. Пакетом тестуються веб-сервера-додатку.

Компанія Microsoft запропонувала розробникам програмного забезпечення свій продукт Visual Studio, який надає комплексні, добре інтегровані рішення для тестування програмних продуктів. Використання його допомагає значно, в порівнянні з іншими програмними засобами за цим напрямком використання, заощадити час; скоротити витрати і підвищити якість програмних додатків упродовж всього життєвого циклу розробки; виконати першокласне функціональне тестування; швидко створити і виконати тести, записуючи при цьому результати в

log-файли і відео та багато іншого.

Функціональні можливості тестування в Visual Studio 2010/2012 забезпечують три продукти:

Перший – це Visual Studio Test Professional, який використовується переважно тестувальниками широкого профілю, для створення ручних тестів, їх запуску та перегляду результатів у складі плану тестування. Включає новий додаток Microsoft Test Manager, який можна швидко і просто встановити на тестовому комп'ютері, на якому тестувальник планує виконувати свої тести.

Другий – Visual Studio Premium, що надає користувачам можливості для створення модульних тестів і автоматичних тестів інтерфейсу користувача. Створені тести можуть запускатися з Microsoft Visual Studio 2010/2012, командного рядку, Team Foundation Build, а також запускатися з плану тестування за допомогою Microsoft Test Manager.

Третім продуктом є Visual Studio Ultimate, який містить всю функціональність тестування, що надається Visual Studio 2010/2012. В порівнянні з попереднім продуктом, до переліку тестів, які можна утворити з його допомогою можна віднести всі вище вказані для попереднього продукту, а також ще тести навантажень, тести веб-сервера продуктивності, звичайні тести і впорядковані. Також до складу Visual Studio Ultimate включено програмний додаток Microsoft Test Manager, який дозволяє забезпечити функціональні можливості планування, розробки і виконання тестів.

В Visual Studio 2010/2012 для тестувальників програмного забезпечення існує багато можливостей:

1) визначення робіт по тестуванню: можна визначити дії процедури тестування для конкретної ітерації в проекті за допомогою додатку Microsoft Test Manager (рис. 3). Для визначення необхідного об'єму тестування можна створювати плани тестування, набори тестів, конфігурації тестів і тестові випадки. Спочатку треба вказати, які набори тестів повинні виконуватися в кожній конфігурації тесту. Потім можуть бути вибрані тести, які повинні запускатися у складі плану тестування;

2) створення і виконання ручних та автоматизованих тестів: за допомогою Microsoft Test Manager (МТМ) можна створювати тестові випадки, що містять окремі кроки тесту. Кожен крок ручного тесту включає дію, яка виконується. Існує можливість вказувати для кожного із кроків очікуваний результат або помічати їх, як ті, що завершилися успіхом або невдачею в результаті тестування. А для того, щоб скоротити час, який є потрібним для створення кроків тесту, а також зменшити поточні витрати на обслуговування, можна створювати загальні кроки, які подалі можуть бути використані в декількох тестових випадках. МТМ дозволяє створювати плани тестування, додавати та поновлювати тестові випадки.

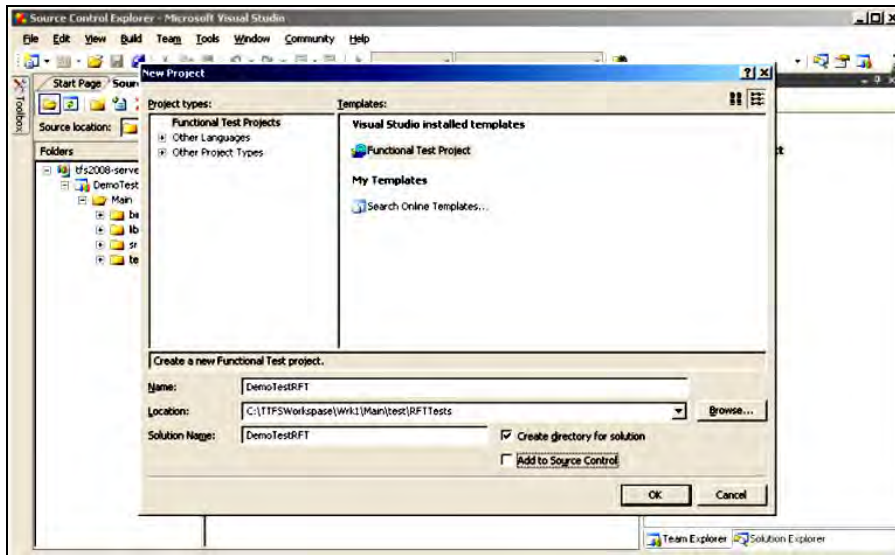


Рис. 1. Створення у середовищі розробки Microsoft Visual Studio нового проєкту «Functional Test Project»

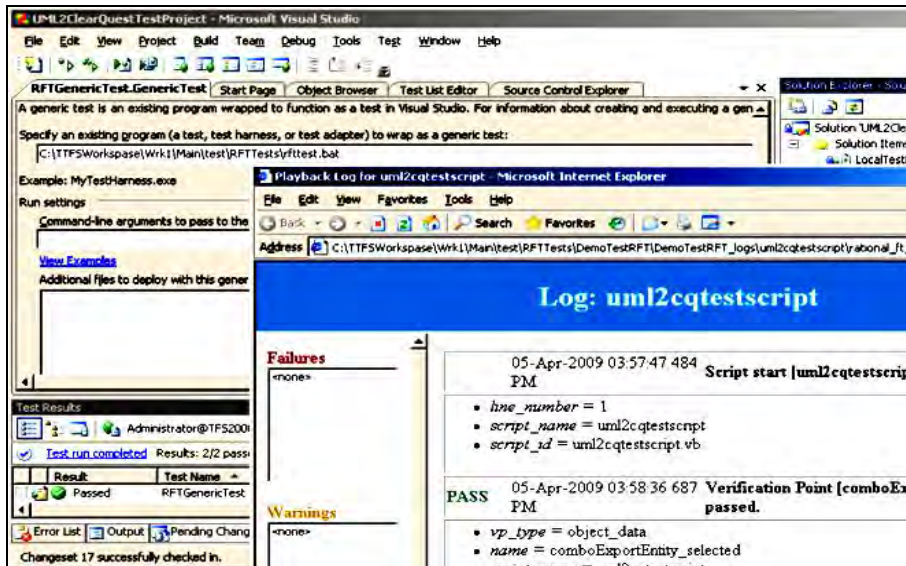


Рис. 2. Екран з результатами виконання тестування в середовищі розробки Microsoft Visual Studio засобами Rational Functional Tester

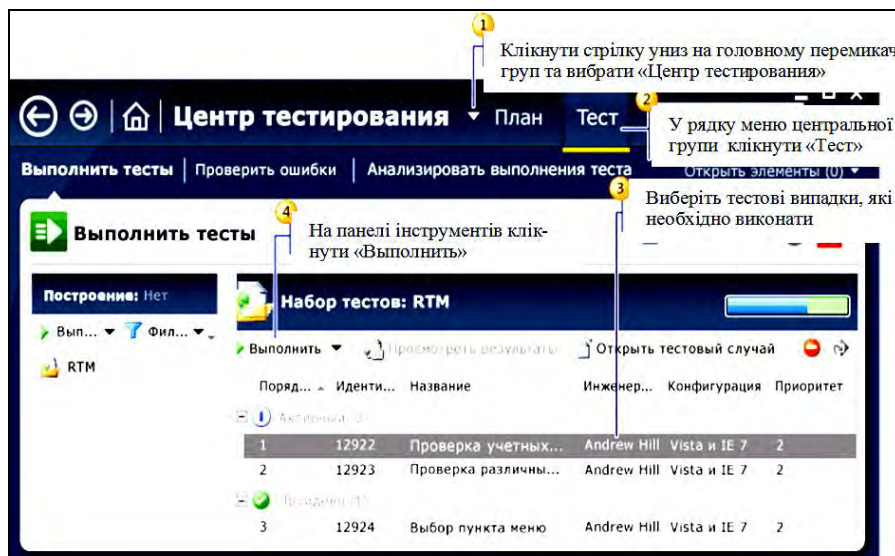


Рис. 3. Головне вікно програмного додатку Microsoft Test Manager

Він також повідомляє тестувальників про зміни між різними версіями, що виключає необхідність їх контактів з розробниками та дозволяє, сфокусувавши свої зусилля на тестуванні нових можливостей програмного продукту, працювати ефективніше. У MTM стандартна робоча процедура документування помилки при виконанні тесту має на увазі включення до звіту: знімків екрану з проблемою; журналу з відеороликами (з відмітками часу) виконання кожного кроку в тесті; журналу

IntelliTrace, що включає повну налагоджувальну інформацію (журнал всіх дій і методів, які супроводжують прогон тесту); списку всіх тестів, які були виконані до і після виявлення помилки; повної системної інформації з комп'ютера, на якому виконувався тест;

3) запис кроків ручного тесту для відтворення: можна створити запис дій, що виконуються для ручного тестового випадку за допомогою програмного додатку Test Runner (рис. 4).

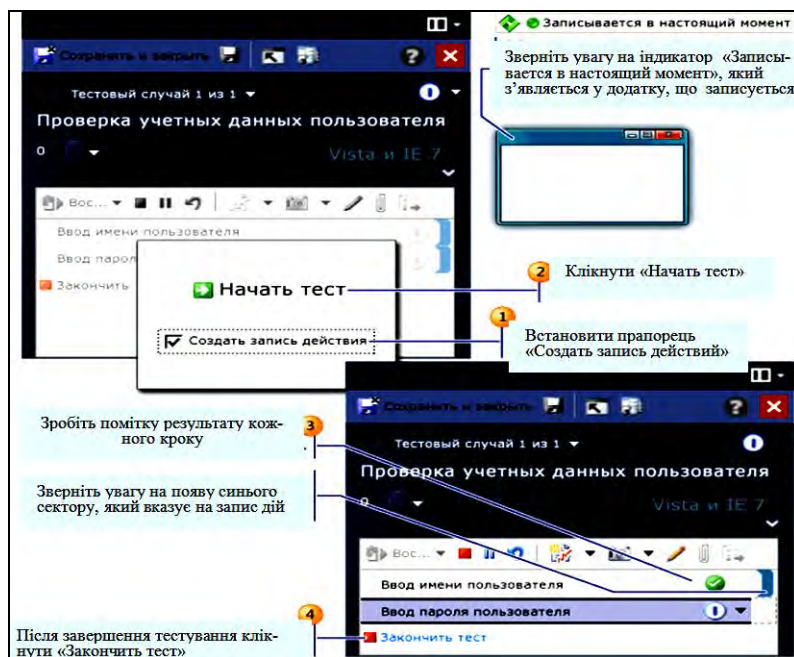


Рис. 4. Вікно програмного додатку Test Runner

Можна відтворити цей запис дій для прискореного переходу до конкретного кроку ручного тесту з метою перевірки усунення помилки, а також для перемотування з метою скорочення часу виконання тесту;

4) автоматизоване тестування інтерфейсу користувача (ІК): можна створювати автоматизовані тести інтерфейсу користувача (закодовані тести інтерфейсу користувача), використовуючи при цьому нові бібліотеки автоматизації. Можна імпортувати запис дій і створити код, який представляє елементи управління інтерфейсу користувача, або знайти необхідні елементи управління і виконати над ними дії. Потім можна додати код перевірки, щоб перевірити правильність роботи додатку, що тестується. Автоматизовані тести інтерфейсу користувача виконують функціональне тестування ІК і перевірку елементів управління ІК. Також вони дозволяють після внесення змін до коду перевірити, чи коректно функціонує ІК. Виконуються такі тести набагато швидше, ніж ручні тести;

5) створення середовищ, які планується використовувати для тестування: можна створювати як фізичні, так і віртуальні середовища, які визнача-

ють набір ролей, необхідних для виконання конкретного додатку, і комп'ютерів, що використовуються для кожної ролі. Для видаленого виконання тестування і збору даних фізичні середовища використовують контролери агентів тестування, пов'язані з Team Foundation Server, і агенти тестування. Можна створювати віртуальні середовища для розгортання і тестування додатків за допомогою Visual Studio Lab Management. Для створення віртуальних машин можна використовувати HYPER-V, а для управління віртуальними машинами і бібліотеками шаблонів машин можна використовувати Systems Center Virtual Machine Manager. При створенні плану тестування можна вибрати середовища і комп'ютери, які планується використовувати;

6) збір діагностичних даних і відомостей про вплив на систему при тестуванні та просте додавання зібраних даних до помилки: при виконанні ручного тесту можна додатково зробити відеозапис тестового випадку, а також записати дії у файл журналу або створити запис дій. В ході виконання тесту можна додавати коментарі, знімки екрану та інші файли. Також можна збирати діагностичні

дані трасування (дані IntelliTrace), дані про покриття коду або дані аналізу впливу тестів. Підтримується можливість емуляції певної мережі на комп'ютері або створення призначеного для користувача адаптера діагностики даних. Ці дані зберігаються разом з результатами тесту. Можна створити запис про помилку, яка буде автоматично заповнена зібраними даними. Це створює помилку з докладними відомостями для розробника, які можна використовувати для повторного усунення проблеми;

7) пошук тестів для повторного виконання на основі змін коду: можна порівняти побудови, щоб проглянути, які тести рекомендується виконати ще раз, на основі змін тестованого додатку;

8) перегляд звітів, які допомагають відстежувати хід тестування: можна переглядати звіти про готовність тестових випадків і хід тестування, створені для плану тестування;

9) групування автоматичних тестів за допомогою категорій тестів: можна використовувати категорії тестів для групування автоматичних тестів. Категорії тестів надають велику гнучкість, чим списки тестів, при групуванні тестів і виборі тестів для виконання;

10) використання Team Foundation Server (TFS) в якості центрального концентратора всієї інформації, пов'язаної з функцією, дозволяє розробникам і тестувальникам ефективно взаємодіяти в спільному, а не в конкурентному середовищі. Всебічний збір даних і генерація звітів - відмінна риса TFS. Можливості звітності транслюють всю доступну інформацію в показники об'єму і кількості (загальна кількість помилок; кількість помилок за етап; кількість помилок на функцію; покриття коду; регресійні помилки; першопричина дефекту; вартість дефекту і таке інше), за рахунок чого кожен керівник отримує інструмент, який миттєво показує, скільки тестових випадків є, скільки було виконано, скільки пройшло успішно і який об'єм коду вони покривають, таким чином керівники можуть ефективніше працювати з повнішою інформацією, знання ж стану якості коду, а не прості припущення, суттєві для успішного випуску. Можливості відстежування TFS є неперевершеними з тієї причини, що жоден інший інструмент тестування не інтегрується безшовно з репозиторієм початкового коду або відстежуванням робочих елементів. У інших інструментах тих, що мають ті ж можливості, це досягається головним чином через інтеграцію між різними системами. Вони не були розроблені цілком і повністю з урахуванням інтеграції і складання звітів за даними, а у випадку з TFS це саме так;

11) використання Test Impact Analysis (TIA) – аналізатора впливу на тести: по ходу наближення

до дати випуску продукту, у тестувальників накопичуються сотні (якщо не тисячі) тестових випадків. Але до коду вже внесено безліч змін і є необхідність переконатися, що в нього не потрапили помилки. Єдиним способом перевірити це є виконання регресійних тестів, але їх кількість також нараховує тисячі варіантів. Потрібно дізнатися, які саме тести необхідно перевиконати, щоб перевірити, що дана конкретна зміна не внесла регресійних помилок. Команди тестування витрачають масу часу на підтримку матриць, в яких якомога повніше записується вся ця інформація. Використання нових можливостей Visual Studio, що реалізовані в Test Impact Analysis (TIA) позбавляє тестувальників від необхідності вирішувати питання: які тести і в якій кількості слід виконувати, щоб переконатися у відсутності регресійних помилок. TIA інформує, що колишні тести були проведені успішно, але зміна зроблена в тій частині коду, яка перевірялася тестом. Таким чином, TIA підвищує ефективність роботи тестувальників.

## Висновки

Таким чином, проаналізувавши велику кількість програмних продуктів для автоматизації проведення тестувальних робіт, як безкоштовних, так і комерційних, для тестування комплексу програм автоматизованої системи конструювання розкладу занять для Харківського університету Повітряних Сил було зроблено вибір на користь Visual Studio Ultimate, який містить всю функціональність тестування, що надається Visual Studio 2010/2012. Перелік тестів, які можна утворити з його допомогою майже один з самих широких та вміщує: модульні тести, автоматичні тести інтерфейсу користувача, тести навантажень, тести веб-сервера-продуктивності, звичайні тести і впорядковані. Також до складу Visual Studio Ultimate включено програмний додаток Microsoft Test Manager, який дозволяє забезпечити функціональні можливості планування, розробки і виконання тестів. Можливості ж відстежування Team Foundation Server є неперевершеними з тієї причини, що жоден інший інструмент тестування не інтегрується безшовно з репозиторієм початкового коду або відстежуванням робочих елементів. У інших інструментах тих, що мають ті ж можливості, це досягається головним чином через інтеграцію між різними системами. Вони не були розроблені цілком і повністю з урахуванням інтеграції і складання звітів за даними, а у випадку з Team Foundation Server це саме так. Така ж складова Visual Studio Ultimate як Test Impact Analysis позбавляє тестувальників від необхідності вирішувати питання: які тести і в якій кількості слід виконувати, щоб переконатися у відсутності регресій-

них помилок. Вона інформує, що колишні тести були проведені успішно, але зміна зроблена в тій частині коду, яка перевірялася тестом. Test Impact Analysis підвищує ефективність роботи тестувальників.

### Список літератури

1. Біла книга 2012. ЗС України. – К. Видання МОУ, 2013. – 78 с.
2. Наказ Міністра освіти України від 02.06.1993 р. №161 "Про затвердження положення про організацію навчального процесу у вищих навчальних закладах".
3. «Інструкція про організацію освітньої діяльності у вищих військових навчальних закладах Збройних Сил України та військових навчальних підрозділах вищих навчальних закладів України», затверджена наказом Міністра освіти і науки України та Міністра оборони України від 13.04.2005 р. № 221/217.
4. «Інструкція з планування та обліку діяльності науково-педагогічних (педагогічних) працівників вищих військових навчальних закладів (військових навчальних підрозділів вищих навчальних закладів, навчальних центрів Збройних Сил України) введена наказом Міністра оборони України та Міністра освіти і науки України від 08.05.2002 р. №155/291 зі змінами згідно сумісного наказу МОУ та МОНУ від 21.02.2006 №99/116.

5. Майерс Г. Искусство тестирования программ / Г. Майерс, Т. Баджетт, К. Сандлер. – 3-е издание. – М.: Диалектика, 2012. – 272 с.

6. Кристин Л. Гибкое тестирование: практическое руководство для тестировщиков ПО и гибких команд / Л. Кристин, Д. Грегори. – М.: Вильямс, 2010. – 464 с.

7. Канер К. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений / К. Канер, Д. Фолк, Е.К. Нгуен. – К.: Диасофт, 2001. – 544 с.

8. Калбертсон Р. Быстрое тестирование / Р. Калбертсон, К. Браун, Г. Кобб. – М.: Вильямс, 2002. – 374 с.

9. Синецын С.В. Верификация программного обеспечения / С.В. Синецын, Н.Ю. Налютин. – М.: БИНОМ, 2008. – 368 с.

10. Бейзер Б. Тестирование чёрного ящика. Технологии функционального тестирования программного обеспечения и систем / Б. Бейзер. – СПб.: Питер, 2004. – 320 с.

Надійшла до редколегії 3.09.2015

**Рецензент:** д-р фіз.-мат. наук, проф. С.В. Смеляков, Харківський університет Повітряних Сил ім. І. Кожедуба, Харків.

### АНАЛИЗ И ИССЛЕДОВАНИЕ ИНСТРУМЕНТОВ АВТОМАТИЗАЦИИ ТЕСТИРОВАНИЯ ПРИ РАЗРАБОТКЕ КОМПЛЕКСА ПРОГРАММ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ КОНСТРУИРОВАНИЯ РАСПИСАНИЯ ЗАНЯТИЙ

С.В. Дуденко, В.В. Калачева, С.В. Алексеев, М.Н. Колмыков

Актуальной задачей, которая стоит как перед Харьковском университете Воздушных Сил (ХУВС), так и перед другими высшими военными учебными заведениями ВС Украины, является подготовка высококвалифицированных военных специалистов. Решить ее можно путем качественного планирования учебного процесса. Расписание занятий является составной частью планирования и связывает в единую систему множество звеньев и элементов учебного процесса, регламентируя работу ее участников. Автоматизация процесса конструирования расписания позволяет избежать многих ошибок, субъективизма и значительно сократить время отведенное на выполнение этой процедуры.

В текущий период в ХУВС ведется разработка комплекса программ автоматизированной системы конструирования расписания занятий. Успешность проведения работ по тестированию во многом зависит от применения автоматизированных средств. Анализ и исследование инструментов автоматизации выполнения работ по тестированию является одним из важнейших этапов разработки комплекса, который позволяет обоснованно подойти к выбору наиболее подходящего средства тестирования из множества возможных и в результате получить высококачественный программный продукт, который будет отвечать целевому уровню требований заказчика.

**Ключевые слова:** расписание учебных занятий, качество программного обеспечения, автоматизация тестирования, тестовый набор, автоматизированный тест-скрипт.

### ANALYSIS AND RESEARCH OF INSTRUMENTS OF TESTING AUTOMATION AT DEVELOPMENT OF EMPLOYMENTS CURRICULUM CONSTRUCTING OF AUTOMATED SYSTEM PROGRAMS COMPLEX

S.V. Dudenko, V.V. Kalachova, S.V. Alekseyev, M.M. Kolmykov

An actual task, which stands both before the Kharkov university of Aircrafts (KHUA) and before other higher soldiery educational establishments of Military Force of Ukraine, is preparation of highly skilled soldiery specialists. Deciding it is possible by the high-quality planning of educational process. A curriculum of employments appears component part of planning and binds the great number of links and elements of educational process in the single system, regulating work of its participants. Automation of process of constructing of time-table allows to avoid many errors, subjectivism and considerably to shorten time taken on implementation of this procedure.

In a current period development of complex of the programs of the automated system of constructing of curriculum of employments is conducted in KHUVS. Success of leadthrough of works on testing in a great deal depends on application of the automated facilities. An analysis and research of instruments of automation of implementation of works on testing appears one of major design of complex, which allows grounded to walk up to the choice of the most suitable mean of testing from a great number possible and as a result to get a high-quality software product which will answer the having a special purpose level of requirements of customer, times.

**Keywords:** curriculum of lessons, quality of software, automation of testing, test set, automated test-script.