

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*

## СТРУКТУРИЗАЦІЯ ВИДІВ ТЕСТУВАННЯ ТА АНАЛІЗ ЇХ ОСОБЛИВОСТЕЙ ПРИ РОЗРОБЦІ КОМПЛЕКСУ ПРОГРАМ АВТОМАТИЗОВАНОЇ СИСТЕМИ КОНСТРУЮВАННЯ РОЗКЛАДУ ЗАНЯТЬ

Актуальною задачею, що стоїть як перед Харківським університетом Повітряних Сил (ХУПС), так і перед іншими вищими військовими навчальними закладами ЗС України, є підготовка висококваліфікованих військових фахівців. Одним з можливих шляхів її вирішення є якісне планування навчального процесу. Розклад навчальних занять є важливим інструментом процесу планування, який пов'язує в єдину систему різноманітні ланки й елементи навчання та регламентує навчальну роботу її учасників. Автоматизація процесу конструювання розкладу дозволяє уникнути багатьох помилок механічного характеру, суб'єктивізму розробника розкладу та значно скоротити час, відведений на цю процедуру. У зв'язку з цим в ХУПС в поточний період ведеться розробка комплексу програм автоматизованої системи конструювання розкладу навчальних занять. Одним із заключних етапів розробки комплексу має стати тестовий контроль якості створеного програмного продукту. Застосування окремих видів тестування та їх сукупне використання під час контролю якості створеного програмного забезпечення дає можливість досягти максимального рівня відповідності готового продукту вимогам замовника.

**Ключові слова:** розклад навчальних занять, якість програмного забезпечення, техніка тестування «чорного ящика» і «білого ящика», структурне та функціональне тестування.

### 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 зумовлена участю багатьох представників різних ланок управління і забезпечення навчального процесу при підготовці, обробці і використанні великої кількості нормативної, навчальної, методичної документації та іншої інформації [1 – 3]. Тому великі часові витрати, помилки та суб'єктивізм – є тим неповним переліком недоліків, що змушує звертатися до наукоємних інформаційних технологій, які в змозі звести до мінімуму існуючі проблеми шляхом автоматизації процесу конструювання розкладу занять та створення відповідного комплексу програм.

Заключним етапом розробки комплексу програм автоматизованої системи конструювання розкладу є тестування на працездатність створеного програмного забезпечення (ПЗ). Метою тестового контролю є виявлення якомога більшої кількості помилок і недоліків ще до введення додатку в активну експлуатацію, з ціллю подальшого їх усунення та досягнення максимальної якості та оптимальності програмного коду.

Сьогодні на ринку програмного забезпечення присутня велика кількість програмних продуктів,

що реалізують процедуру тестування. Завдяки використанню в них різних видів тестування результати контролю якості теж різняться – програмний код має різний рівень оптимальності. У зв'язку з цим, питання аналізу особливостей різних видів тестування та їх структурування, відкривають шлях до обґрунтованого вибору одного з цих продуктів.

**Аналіз останніх досліджень і публікацій.** Відомості щодо впровадження новітніх інформаційних технологій у навчальний процес в вищих військових навчальних закладах ЗС України докладно розкрито в [1 – 3]. Основні види та методики тестування представлені в [4 – 6]. Основні принципи здійснення тестового контролю якості програмних продуктів наведено в [7 – 9].

**Формулювання мети статті.** Проведення аналізу особливостей різних видів тестування та їх структурування ставить на меті надання тестувальникам та розробникам комплексу програм автоматизованої системи конструювання розкладу занять можливості здійснення обґрунтованого вибору програмного засобу для проведення його тестування, який буде враховувати всі вимоги замовника та гарантує досягнення максимальної якості і оптимальності програмного коду.

### Виклад основного матеріалу досліджень

Комплекс програм автоматизованої системи конструювання розкладу занять для ХУПС розробляється як клієнт-серверний додаток, який буде функціонувати у локальній мережі ВВНЗ. Його клієнтська частина реалізує графічний інтерфейс користу-

вача для внесення й обробки даних щодо визначених функціональних задач комплексу. Серверна частина представлена безпосередньо фізичною базою даних та СУБД, які забезпечують необхідний рівень надійності та функціональності.

Основними етапами створення комплексу програм автоматизованої системи конструювання розкладу занять є аналіз предметної області; вибір методів, алгоритмів, інструментальних засобів і технологій; розробка програмних модулів; перевірка функціонування та працездатності створеного програмного продукту шляхом тестування.

Процедура тестування представляє собою перевірку відповідності між реальною і очікуваною поведінкою програми, що здійснюється на кінцевому наборі тестів з метою виявлення в ній помилок. Після виправлення розробником дефекту, знайденого тестувальником в протестованому програмному продукті, має бути проведене повторне тестування, з метою підтвердження факту виправлення помилки та перевірки працездатності продукту після внесених змін.

За даними, опублікованим Національним інститутом стандартів (NIST 2002 RTI Project 7007.011), основна кількість помилок в продукті – 70%! – закладається на стадії вироблення вимог і побудові дизайну. А виявляються переважна більшість дефектів або в процесі тестування (близько 60%), або вже при експлуатації (21%). При цьому питома вартість виправлення дефектів швидко росте по мірі просування продукту до стадії експлуатації. Так, в статті "Software Defect Reduction Top 10 List", опублікованій в IEEE Computer, доведено, що вартість виправлення дефекту після введення системи в експлуатацію удвічі перевищує аналогічну вартість на стадії тестування продукту, і більш ніж в тисячу разів – в період вироблення вимог до продукту.

Всі існуючі види тестування (Т) програмного забезпечення за цілями їх проведення можуть бути класифіковані так, як представлено на рис. 1 [4 – 6].

Функціональне тестування проводиться за допомогою самого інтерфейсу програми, щоб проказати, що функції програми виконуються, вхідні дані сприймаються належним чином, вихідні дані генеруються правильно. Все це перевіряється безвідносно до внутрішньої логічної структури програми, тільки на підставі функціональних вимог до програмного продукту і вимог до окремих його властивостей. Тобто функціональне тестування виконується для перевірки конкретних вимог. Воно підходить для постійного процесу тестування, в якому кожна функція тестується після свого створення, а також постійно повторно тестується по ходу інтеграції нових функцій для виявлення регресійних помилок.

Класифікацію видів тестування програмного забезпечення можна проводити і за різними напрямками [7 – 9] (рис. 2). Компонентне (модульне), інтеграційне та системне тестування ПЗ відносять до тесту-

вання за ступенем ізольованості компонентів, яке характеризує також рівень тестування програмного продукту. Для модульного тестування характерним є здійснення перевірки мінімально можливого для тестування компоненту (наприклад, окремого класу або функції) самим розробником ПЗ. При інтеграційному ж тестуються інтерфейси між компонентами, підсистемами або системами. За наявності резерву часу на даній стадії тестування ведеться ітераційно, з поступовим підключенням подальших підсистем.



Рис. 1. Класифікація видів тестування за цілями

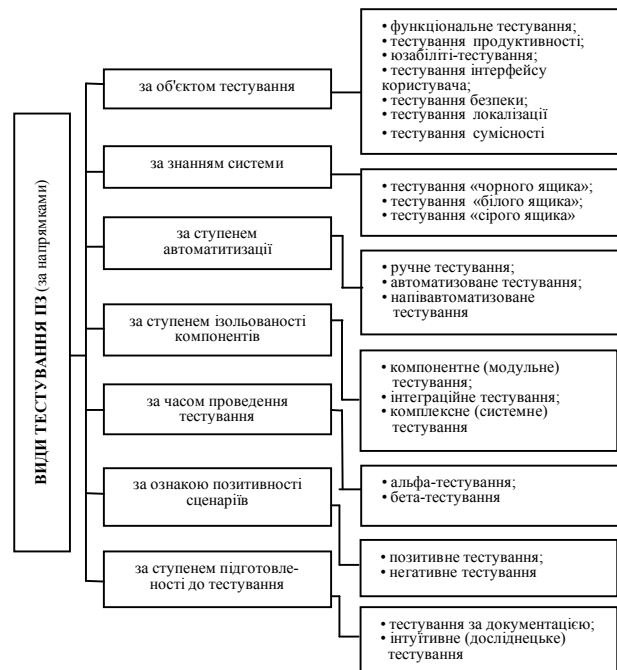


Рис. 2. Класифікація видів тестування за напрямками

Під час проведення комплексного (системного) тестування проводиться тестування всієї системи, яке дозволяє переконатися, що функціональні і нефункціональні вимоги дотримуються. Це тестування виконується всією командою, тобто в ньому беруть участь розробники, тестувальники і бізнес-аналітики. Воно може вважатися за «тестування перед прийманням замовником» і дозволяє гарантувати, що все працює

так, як треба. Бажано проводити комплексне тестування в середовищі, близькому до того, в якому система буде працювати. Альфа і бета-тестування відносяться до системного тестування. Альфа-тестування імітує реальну роботу з системою штатними розробниками або реальну роботу з системою потенційними користувачами (замовниками). Найчастіше альфа-тестування проводиться на ранній стадії розробки продукту, але в деяких випадках може застосовуватися і для закінченого продукту як внутрішнє приймальне тестування. Різновидами альфа-тестування є «димове», підтверджуюче, регресійне, прийомне тестування та тестування нової функціональності.

«Димові» тести використовуються в якості приймального тесту до запуску повноцінного тестування. Успішне проведення тестування цього виду дає деяку гарантію того, що система, що проходить тестування, не видасть катастрофічний збій. Під час здійснення регресійного тестування проводиться перевірка працездатності програмного продукту після внесених до нього після «димового» тестування змін, яка дозволяє переконатися, що, незважаючи на внесення цих змін (наприклад, виправлення помилок або додавання нової функціональності), старий код продовжує працювати - виправлення помилки, не привело до нових помилок. Набори регресійних тестів по можливості має сенс автоматизувати.

При застосуванні бета-тестування в деяких випадках виконується розповсюдження попередньої версії (у разі пропріетарного програмного забезпечення іноді з обмеженнями по функціональності або часу роботи) для групи осіб з тим, щоб переконатися, що продукт містить достатньо мало помилок. Іноді бета-тестування виконується для того, щоб отримати зворотній зв'язок про програмний продукт від його майбутніх користувачів. Часто для вільного/відкритого ПЗ стадія альфа-тестування характеризує функціональне наповнення коду, а бета-тестування стадію виправлення помилок. При цьому, як правило, на кожному етапі розробки проміжні результати роботи доступні кінцевим користувачам.

При тестуванні функціональної поведінки програми найчастіше звертаються до техніки "чорного ящика" (текст програми закритий для користування), під час застосування якої відбувається перевірка відповідності поведінки програми її зовнішній специфікації. Критерієм повноти тестування за цією технікою є перебір можливих значень вхідних даних. При структурному ж тестуванні програма розглядається як "білий ящик" (тобто її текст відкритий для користування) і тестування коду проводиться з ціллю з'ясування логіки і коректності роботи його з погляду компілятора тієї мови, на якому вона складалася.

Техніка "сірого ящика" об'єднує методологію "чорного" та "білого ящиків" в єдине ціле. Вона включає знання внутрішніх структур даних та алгоритмів, які використовуються для створення тестів, але тестування виконується на рівні користувача або

"чорного ящика". Тут тестувальник складає більш ефективні тести, тому що знає, як компоненти, які лежать в основі, працюють та взаємодіють один з одним. Типовими прикладами тестів, виконаних за цією технікою, є дослідження граничних умов та повідомлень про помилки.

Техніки тестування «білого і чорного ящиків» припускають, що код виконується, а різниця полягає лише в тій інформації, якою володіє тестувальник. У обох випадках це динамічне тестування. При статичному тестуванні програмний код не виконується – аналіз програми відбувається на основі початкового коду, який вичитується вручну, або аналізується спеціальними інструментами. В деяких випадках аналізується не початковий, а проміжний код (такий як байт-код або код на MSIL).

Якщо є потрібність в перевірці продуктивності програмного продукту, має сенс перейти до тестування навантаження, метою якого є перевірка роботи системи з максимальним його значенням. Тестування навантаження зазвичай виконується в контрольованому лабораторному середовищі, що дозволяє проводити точну оцінку в умовах, що повторюються, але також може виконуватися в «польових» умовах, щоб отримати кількісну оцінку продуктивності системи в «реальному світі».

Іноді найбільший ефект від тестування досягається шляхом застосування такого його різновиду як дослідницьке (пошукове, інтуїтивне) тестування, проводячи яке тестувальник не має конкретних тестових сценаріїв, а тестує головним чином для того, щоб досліджувати можливості програмного забезпечення і виявити невідомі помилки. Тут метою є дізнатися і зрозуміти програмне забезпечення і його можливості. В ході цього виду тестування тестувальник також намагається продумати всі можливі сценарії, в яких можуть виникнути збої. Дослідницьке тестування є тим єдиним типом тестування, яке допомагає виявити помилки, які мають високі шанси залишитися непоміченими при використанні інших стратегій тестування.

Спроба підвищити швидкість проведення тестувальних робіт, збільшити вірогідність виявлення дефектів та знизити собівартість тестування напряму зв'язана з автоматизацією цього процесу. Тому ручне та напівавтоматичне тестування має сенс застосовувати лише для перевірки роботи програмних додатків з невеликим обсягом програмного коду, а в інших випадках треба звертатися до автоматизованих тестів.

## Висновки

Проведений аналіз існуючих видів тестування та їх структурування дали можливість сформулювати перелік контрольних заходів для перевірки працездатності комплексу програм автоматизованої системи конструювання розкладу занять, що розробляється в Харківському університеті Повітряних Сил. Була доведена доцільність застосування більшості з

наведених видів тестування під час контролю якості розробленого програмного продукту. Таким чином, можна стверджувати, що застосування різних видів тестування є гарантією того, що готовий продукт буде відповідати вимогам замовника або перевершувати їх. Жоден вид тестування не дозволяє поодинокі отримати повний контроль працездатності програмного продукту. Тільки сукупне використання декількох видів тестування допомагає команді розробників і зацікавленим особам отримати високий рівень якості всього проекту. Тести можуть записуватися як вручну, так і автоматично. У будь-якому випадку тести повинні повторюватися, щоб можна було оцінювати прогрес і покращення в ході розробки. Також слід прагнути до високого ступеня автоматизації і використовувати ручне тестування лише там, де має найбільший сенс використовувати людську оцінку і інтерпретацію.

### 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 Мініс-

тра освіти і науки України та Міністра оборони України від 13.04.2005 р. № 221/217.

4. Гленфорд Майерс. Искусство тестирования программ, 3-е издание = *The Art of Software Testing, 3rd Edition* / Гленфорд Майерс, Том Баджетт, Кори Сандлер. – М.: «Диалектика», 2012. – 272 с. – ISBN 978-5-8459-1796-6.

5. Лайза Криспин. Гибкое тестирование: практическое руководство для тестировщиков ПО и гибких команд = *Agile Testing: A Practical Guide for Testers and Agile Teams* / Лайза Криспин, Джанет Грегори. – М.: «Вильямс», 2010. – 464 с. – (Addison-Wesley Signature Series). – 1000 экз. – ISBN 978-5-8459-1625-9.

6. Канер Кем. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений / Канер Кем, Фолк Джек, Нгуен Енг Кек. – К.: ДиаСофт, 2001. – 544 с. – ISBN 9667393879.

7. Калбертсон Роберт. Быстрое тестирование / Калбертсон Роберт, Браун Крис, Кобб Гэри. – М.: «Вильямс», 2002. – 374 с. – ISBN 5-8459-0336-X.

8. Синицын С.В. Верификация программного обеспечения / С.В. Синицын, Н.Ю. Налютин. – М.: БИНОМ, 2008. – 368 с. – ISBN 978-5-94774-825-3.

9. Бейзер Б. Тестирование чёрного ящика. Технологии функционального тестирования программного обеспечения и систем / Б. Бейзер. – СПб.: Питер, 2004. – 320 с. – ISBN 5-94723-698-2.

Надійшла до редколегії 20.08.2015

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

### СТРУКТУРИЗАЦИЯ ВИДОВ ТЕСТИРОВАНИЯ И АНАЛИЗ ИХ ОСОБЕННОСТЕЙ 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 Харьковским университетом Воздушных Сил (ХУВС), так и перед другими высшими военными учебными заведениями Вооруженных Сил Украины, является подготовка высококвалифицированных военных специалистов. Одним из путей ее решения является качественное планирование учебного процесса. Расписание занятий является важным инструментом процесса планирования, который связывает в единственную систему разнообразные звенья и элементы процесса обучения и регламентирует учебную работу ее участников. Автоматизация процесса конструирования расписания позволяет избежать многих ошибок механического характера, субъективизма разработчика расписания и значительно сократить время отведенное на эту процедуру. В связи с этим в ХУВС в текущий период ведется разработка комплекса программ автоматизированной системы конструирования расписания учебных занятий. Одним из заключительных этапов разработки комплекса должен стать тестовый контроль качества созданного программного продукта. Применение отдельных видов тестирования, а также совместное их использование во время проверки качества созданного программного обеспечения дает возможность достичь максимального уровня соответствия готового продукта требованиям заказчика.

**Ключевые слова:** расписание учебных занятий, качество программного обеспечения, техника тестирования «черного ящика» и «белого ящика», структурное и функциональное тестирование.

### STRUKTURIZATION OF TESTING TYPES AND THEIR FEATURES ANALYSIS FOR DEVELOPMENT OF THE EMPLOYMENTS CURRICULUM CONSTRUCTING OF AUTOMATED SYSTEM PROGRAMS COMPLEX

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

Actuality task, which stands both before the Kharkov university of Aircrafts and before other higher soldiery educational establishments of Military Powers of Ukraine, is preparation of highly skilled soldiery specialists. One of ways of its decision is the high-quality planning of educational process. A curriculum of employments is the important instrument of planning process which binds in the unique system various links and elements of teaching process and regulates educational work of its participants. Automation of process of constructing of time-table allows to avoid many errors of mechanical character, subjectivism of developer of time-table and considerably to shorten time taken on this procedure. In this connection in KHUVS development of complex of the programs of the automated system of constructing of curriculum of lessons is conducted in a current period. Test control of quality of the created software product must one of final design of complex times become. Application of separate types of testing, and also their sharing during test of quality of the created software enables to attain the maximal level of conforming to of the prepared product the requirements of customer.

**Keywords:** curriculum of lessons, quality of software, technique of testing of «black box» and «white box», structural and functional testing.