

Актуальні питання навчання

УДК 681.324:623.618(06)

В.Г. Малюга, О.М. Місюра, В.В. Калачова, О.А. Трублін

Харківський університет Повітряних Сил ім. І. Кожедуба, Харків

АНАЛІЗ ТА ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ ПРОЦЕДУРИ ПРОВЕДЕННЯ УСПІШНОГО ТЕСТУВАННЯ ПРИ РОЗРОБЦІ КОМПЛЕКСУ ПРОГРАМ АВТОМАТИЗОВАНОЇ СИСТЕМИ КОНСТРУЮВАННЯ РОЗКЛАДУ ЗАНЬ

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 зі шляхів вирішення її є якісне планування навчального процесу. Розклад занять є складовою частиною планування, що пов'язує в єдину систему різноманітні ланки й елементи навчального процесу і регламентує навчальну роботу її учасників. Автоматизація процесу конструювання розкладу дозволяє уникнути багатьох помилок, суб'єктивізму та значно скоротити час, відведений на цю процедуру.

В поточний період в ХУПС здійснюється розробка комплексу програм автоматизованої системи конструювання розкладу занять. Одним із заключних та найважливіших етапів розробки комплексу має стати тестовий контроль якості програмного продукту. Успішність проведення тестувальних робіт багато в чому залежить від дотримання тестувальниками основних принципів тестування; правильної організації цієї процедури; урахування вимог замовника і таке інше. Результатом вдало проведеного тестування стає високоякісний програмний продукт, який відповідає або перевершує цільовий рівень вимог замовника.

Ключові слова: розклад навчальних занять, якість програмного забезпечення, димове та регресійне тестування, багтрекер, тест кейс, дослідницьке тестування.

Вступ

Постановка проблеми. Чітка організація навчального процесу у вищому військовому навчальному закладі є гарантом успіху підготовки високоякісних військових фахівців та може бути досягнута лише при якісному його плануванні. Одним із найбільш відповідальних, трудомістких і складних завдань планування навчального процесу є складання семестрових розкладів навчальних занять (екзамнів). 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'єктивізм – є тим не повним переліком недоліків, що вимушує звертатися до наукоємних інформаційних технологій, які в змозі звести до мінімуму існуючі негаразди шляхом автоматизації процесу конструювання розкладу занять та створення відповідного комплексу програм [1 – 4].

Заключним етапом розробки комплексу програм автоматизованої системи конструювання розкладу, що проводиться зараз в ХУПС, виступає тестування його працездатності, метою якого є виявлення якомога більшої кількості помилок і недоліків у створеному програмному продукті, ще до введен-

ня його в активну експлуатацію, з ціллю подальшого їх вилучення та досягнення максимальної якості та оптимальності програмного коду розробки.

Успішність проведення тестувальних робіт багато в чому залежить від дотримання тестувальниками основних принципів тестування; правильної організації контролю якості; урахування вимог замовника і таке інше. Результатом вдало проведеного тестування стає високоякісний програмний продукт, який відповідає або перевершує цільовий рівень 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]. Основні види та методики тестування представлені [5 – 7]. Основні принципи здійснення тестового контролю якості програмних продуктів наведено в [8, 9]. Вимоги до тестів та особливості організації тестування, що сприяють досягненню максимально якості програмного продукту розглянуто у [8 – 10].

Формулювання мети статті. Проведення аналізу існуючих видів, методик, алгоритмів, технік, стратегій тестового контролю якості програмних продуктів, і дослідження основних принципів тестування та особливостей його організації, що має на меті надати тестувальникам та розробникам комплексу програм автоматизованої системи конструю-

вання розкладу занять, що зараз проводиться в Харківському університеті Повітряних Сил, теоретичний базис для того, щоб отримати високоякісний програмний продукт з оптимальним кодом, який врахує всі вимоги замовника.

Виклад основного матеріалу досліджень

Процес тестування складається з трьох етапів: проектування тестів, виконання тестів та аналіз отриманих результатів. На першому етапі вирішується питання про вибір деякої підмножини множини тестів, яка зможе знайти найбільшу кількість помилок за найменший період часу. На етапі виконання тестів проводять запуск тестів і відловлюють помилки у програмному продукті, що тестується. Отримані результати тестування дозволяють зробити висновок про надійність та якість програмного продукту. Вони служать основою його сертифікації і гарантією якості.

До основних принципів тестування можуть бути віднесені наступні з них [5 – 7]:

1) всі тести повинні проводитися відповідно до вимог замовника з ціллю розкриття будь-яких дефектів, які могли б стати причиною не дотримання програмою або системою вимог клієнта;

2) тести мають бути сплановані заздалегідь до початку тестування: зразу після того, як модель вимог завершена, тестове планування може бути почате; деталізовані тестові кейси можуть бути запущені на виконання, як тільки модель проекту буде розроблена;

3) принцип Парето, що застосовується до тестування програмного забезпечення вказує на те, що 80 відсотків всіх помилок, виявлених під час тестування, містяться у 20 відсотках всіх програмних компонентів; проблема полягає в ізоляції цих проблемних компонентів і ретельній їх перевірці;

4) тестування має починатися “з малого” і прогресувати у напрямку “до великого”: перші тести плануються і випробуються на окремих програмних компонентах, далі, під тестування потрапляють об'єднані групи компонентів, а на завершальному етапі – вся система в цілому;

5) треба пам'ятати, що всебічне вичерпне тестування неможливо здійснити: навіть для програми незначного розміру кількість шляхів перестановок дуже велика, з цієї причини неможливо виконати всі комбінації шляхів під час тестування, проте, є можливість адекватно покрити програмну логіку і гарантувати, що всі умови компонентно рівневого проекту були здійснені;

6) щоб тестування було найбільш ефективним, його слід проводити незалежній третій стороні. Розробник програмного забезпечення - не є кращим варіантом для здійснення його тестування.

Що стосується організації тестування програмного продукту, то воно проводиться одночасно у трьох напрямках: перевірка коду, тестування високого рівня, тестування низького рівня [6 – 8] (рис. 1).

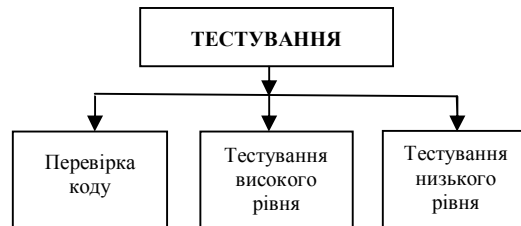


Рис. 1. Напрямки організації тестування

Під час перевірки коду тестер переглядає початковий код візуально і намагається знайти в ньому помилки, а також різні невідповідності коду і вимог до нього. Під вимогою розуміється стандарт, якого дотримуються розробники даного проекту, реакція на ті або інші дії з боку середовища дії на ПЗ, поведінку програмного продукту в різних ситуаціях.

Головною метою тестування високого рівня є з'ясування питання, чи задовольняє розробка всім вимогам замовника. Для програмного продукту пишуться емулятори, за допомогою яких тестер може спостерігати за роботою системи в ролі оператора. Він бачить, як система здійснює діалог з користувачем, які повідомлення вона видає, як реагує на різні події, зберігає інформацію і таке інше. Більшість помилок, що виявляються, на цьому етапі пов'язані з помилками взаємодії програмного продукту з користувачем – виведення помилкових повідомлень, неправильна реакція на запит від оператора і таке інше.

Під час тестування низького рівня тестер перевіряє, наскільки логічно повно початковий код покриває всі можливі варіанти роботи системи, для якої він розробляється.

В авіації США на основі стандарту тестування ISO/IEC 12207 було вироблено стандарт RTCA (Requirements and Technical Concepts for Aviation). У ньому перелічені вимоги до тестування верхнього і нижнього рівнів.

До вимог тестування верхнього рівня відносять [9, 10]:

- вимоги високого рівня повинні включати системні вимоги до ПЗ;
- вимоги високого рівня повинні формулюватися з урахуванням архітектури ПЗ;
- програмний код повинен задовольняти архітектурі ПЗ і вимогам низького рівня;
- код, що відкомпілювався і готовий до використання, повинен задовольняти вимогам до ПЗ;
- значення, що використовуються, повинні технічно відповідати поставленим цілям і виконувати їх для всіх рівнів ПЗ.

Тестування нижнього рівня вміщує:

- перевірку (verification) вимог нижнього рівня;

- перевірку архітектури програмного забезпечення (ПЗ);
- перевірку логічного покриття для всіх функцій написаних в ПЗ;
- контроль процедур тестування;
- незалежність ПЗ від тестування, тобто ПЗ не повинно перебудовуватися особливим чином під тести;
- тестування повинно кілька разів покривати початковий код, для виявлення певного класу помилок;
- робасне тестування;
- тестування на предмет непрямого виявлення помилок (наприклад, відповідність стандартам розробки ПЗ).

До основних принципів організації тестування можна віднести наступні вимоги:

- 1) необхідною частиною кожного тесту має бути опис очікуваних результатів роботи програми;
- 2) програма не повинна тестуватися її автором;
- 3) організація-розробник програмного забезпечення не повинна "самотужки" його тестувати;
- 4) необхідно підбирати тести не тільки для правильних (передбачених) вхідних даних, але і для неправильних (непередбачених);
- 5) при аналізі результатів кожного тесту необхідно перевіряти, чи не робить програма того, що вона не повинна робити;
- 6) "принцип скупчення помилок" – ймовірність наявності невиявлених помилок в деякій частині програми прямо пропорційна числу помилок, вже виявлених в цій частині.

Тестування починається не з того моменту, коли отримано робочу версію додатку, а раніше – коли лише стає відомим, що команда починає працювати над проектом.

Тестувальники залучаються в процес збору вимог до продукту. Проводиться всебічний аналіз предметної галузі. Бізнес-аналітик або власник продукту заповнює документ (беклог продукту), що містить список вимог до функціональності, які впорядковані по ступеню важливості, інакше кажучи, готує список того, що має бути реалізоване.

Перед початком тестування розробляється підхід щодо шляхів його проведення: вирішується, що є умовою для випуску програмного продукту; як буде організоване виконання кожного з видів тестування; хто відповідає за якість та інші моменти, які формують критерії такого підходу. Після отримання перших специфікацій, починається написання тест-плану (алгоритму послідовності дій при тестуванні), розробляються тест-кейси (набори тестів за відповідними напрямками дослідження), оцінюється необхідність використання автоматизації. Як тільки розробники підготують робочу версію додатку, є потреба в проведенні димового тестування, за результатами прове-

дення якого робиться висновок про можливість і доцільність подальшого тестування (рис. 2).

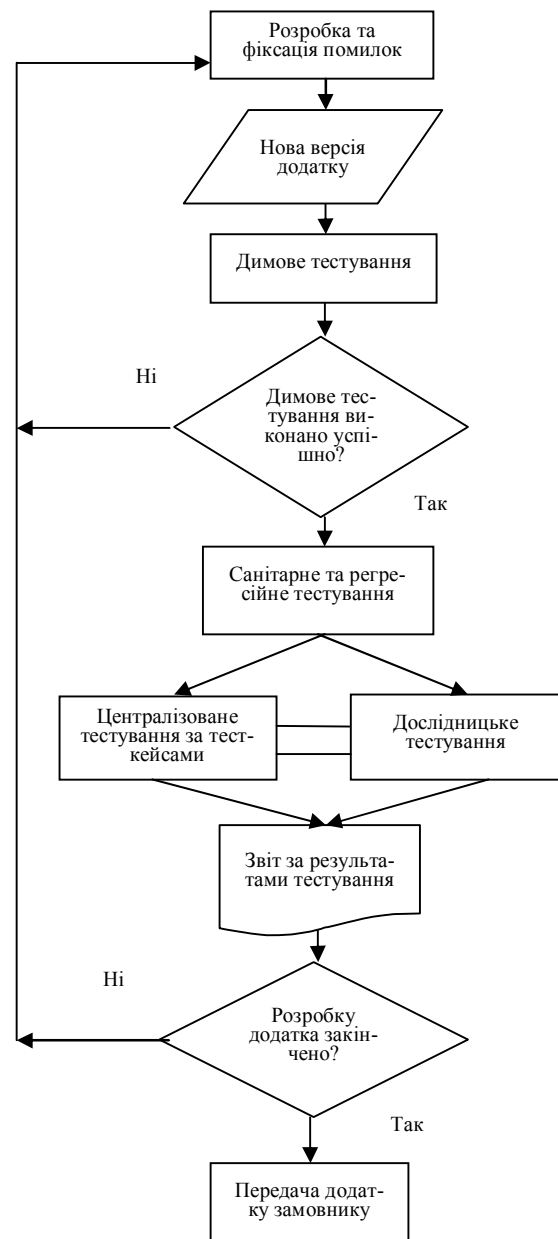


Рис. 2. Алгоритм процесу тестування додатку

«Димове» тестування є різновидом альфа-тестування, яке в свою чергу представляє собою імітацію реальної роботи з системою штатними розробниками або реальною роботу з системою потенційними користувачами (замовниками) та найчастіше проводиться на ранній стадії розробки продукту. «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. У випадку, якщо додаток не витримає димове тестування, його відправляють на доопрацювання розробникам. Якщо ж димове тестування успішно завершилося, то

відбувається перехід до регресійного та санітарного видів тестування, головною метою виконання яких є отримання гарантій того, що зміни, пов'язані з виправленням знайдених на попередньому кроці помилок, не привели до нових помилок. Із-за тісного взаємозв'язку програмних додатків навіть невеликі зміни в будь-якому програмному модулі здатні вплинути на роботу десятків додатків. Тому якщо в нову версію вкрадеться серйозна помилка, то вона може привести до збоїв і в інших додатках. Крім того, навіть у тому випадку, коли нове програмне забезпечення не містить помилок, його розгортання може привести до конфліктів за ресурси з іншими додатками і, як наслідок, до недостатньої продуктивності. Навіть короткочасні збої в їх роботі можуть привести до суттєвих фінансових збитків.

Далі, відкривши багтрекер (файл плану тестування, в якому зафіксовані статуси знайдених дефектів), перевіряються ще раз дефекти які розробники перевели у статус «Виправлено», «Відхилений», «Не можливо відтворити» і так далі. Статуси «Відхилений» і «Не можливо відтворити» є найнеприємнішими, тому що це явне свідчення того, що або тестувальник недостатньо добре локалізував дефект, не дуже зрозуміло описав кроки для відтворення, або розробник неякісно виконав відтворення ситуації.

Покінчивши із закриттям і перевідкриттям дефектів, переходять до основної роботи – централізованого тестування за тест-кейсом або дослідницького тестування додатку. Під час проведення дослідницького тестування тестувальник не має конкретних тестових сценаріїв, а намагається продумати всі можливі шляхи і ситуації, в яких можуть виникнути збої. Коли все, що було заплановане, пройдено, з'являються результати прогону тест-кейсів – баг-репорти та питання до аналітиків. Далі складається звіт про результати здійснення тестування, який відправляється проектній групі.

Програмний продукт проходить тестування за подібним алгоритмом від версії до версії. Через деякий час всі помилки, які були вказані в багтрекерці, приймають статус «Виправлено», що і є своєрідним критерієм закінчення тестувальних робіт, та водночас вказівкою щодо передачі додатку замовнику.

Щоб тестування пройшло максимально успішно, треба врахувати ряд особливостей щодо організації цього процесу:

- ще до проведення тестування команда розробників програмного забезпечення повинна провести ефективні формальні технічні огляди, які дозволять попередити появу багатьох помилок;

- тестування починається з рівня програмних компонентів і рухається у напрямі повної інтеграції комп'ютерної системи;

- у різні моменти часу і в різних точках досліджуваного програмного продукту можуть бути ви-

користані різні технології тестування, але вони повинні відповідати даному конкретному випадку;

- тестування проводять: розробники програмного забезпечення (для великих проектів) і незалежна група тестування;

- тестування і відладка – різні дії, але відладка має бути включеною в будь-яку тестуючу стратегію.

Щоб тест був вдалим, він повинен:

- бути автоматизованим і тим, що повторюється;

- бути нескладним в реалізації;

- бути актуальним;

- запускатися будь-яким бажаним шляхом одного натиснення кнопки;

- запускатися швидко;

- бути послідовним в своїх результатах (тест завжди повинен видавати один і той же результат, якщо між запусками в ньому не відбулося жодних змін);

- бути повністю контрольованим під час проведення тестування;

- бути повністю ізольованим (запускатися незалежно від інших тестів);

- у випадку, якщо тест не пройдено, має бути можливість легкого визначення очікуваного результату і визначення проблеми.

Підтримка якості програмного продукту шляхом тестування продовжуються навіть після того, як додаток поставлено замовнику. Перерва в обслуговуванні, викликана проблемами з надійністю або продуктивністю, може стати причиною невикористаних можливостей, якщо клієнти не зможуть застосувати систему або бізнес не зможе функціонувати. Моніторинг роботи, протоколювання і відстежування помилок, а також інформація від користувачів повинні передаватися назад розробникам і постійно аналізуватися ними. Періодичний аналіз і адаптація продукту є ключовим елементом гнучкості, підтримкою лояльності клієнтів та гарантом постійного покращення розробки.

Висновки

Таким чином, в результаті проведеного аналізу існуючих видів, методик, алгоритмів, технік, стратегій тестового контролю якості програмних продуктів, а також дослідження основних принципів тестування та особливостей його організації, можна зробити висновок, що все протестувати неможливо. 99,9% програмних додатків існуючих в світі не можна зробити такими, щоб вони взагалі не містили помилок. Не можна знайти всі помилки, але можна знайти більшість з них. До певного моменту застосовується закон убуючої віддачі, а після на пошук і виправлення помилок уходить так багато часу і зусиль, що це стає непрактичним. Цей факт має бути врахований при виборі стратегії тестування. Само ж тестування повинно починатися не з того моменту, коли отримано робочу версію додатку, а раніше:

коли лише стає відомим, що команда починає працювати над проектом, а тестувальники повинні бути залучені в процес збору вимог до програмного продукту. Послідовне ж застосування для кожної нової версії програмної розробки, таких видів тестування, як димове, регресійне і санітарне відкриває шлях до усунення максимальної кількості помилок та отриманні високоякісного програмний продукт з оптимальним кодом, який врахує всі вимоги замовника.

Список літератури

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

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

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

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

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

10. 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 16.09.2015

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

АНАЛИЗ И ИССЛЕДОВАНИЕ ОСОБЕННОСТЕЙ ПРОЦЕДУРЫ ПРОВЕДЕНИЯ УСПЕШНОГО ТЕСТИРОВАНИЯ ПРИ РАЗРАБОТКЕ КОМПЛЕКСА ПРОГРАММ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ КОНСТРУИРОВАНИЯ РАСПИСАНИЯ ЗАНЯТИЙ

В.Г. Малюга, О.Н. Мисюра, В.В. Калачева, А.А. Трублин

Актуальной задачей, которая стоит как перед Харьковским университетом Воздушных Сил (ХУВС), так и перед другими высшими военными учебными заведениями ВС Украины, является подготовка высококвалифицированных военных специалистов. Одним из возможных путей ее решения считается обращение к процедуре планирования. Расписание занятий является частью планирования учебного процесса и служит для связи в единую систему множества его звеньев и регламентирует работу его участников. Автоматизация процесса конструирования расписания позволяет избежать многих ошибок, субъективизма и значительно сокращает время, отведенное на эту процедуру.

В текущий период в ХУВС осуществляется разработка комплекса программ автоматизированной системы конструирования расписания занятий. Одним из заключительных и важнейших этапов разработки комплекса должен стать тестовый контроль качества программного продукта. Успех проведения тестовых работ во многом зависит от соблюдения тестировщиками основных принципов тестирования; правильной организации этой процедуры; максимального учета требований заказчика. Результатом успешно проведенного тестирования является высококачественный программный продукт, который отвечает или даже превосходит целевой уровень предъявляемых к нему требований.

Ключевые слова: расписание учебных занятий, качество программного обеспечения, димовое и регрессионное тестирование, багтрекер, тест-кейс, исследовательское тестирование.

ANALYSIS AND FEATURES RESEARCH OF LEADTHROUGH SUCCESSFUL TESTING PROCEDURE AT DEVELOPMENT OF AUTOMATED SYSTEM PROGRAMS COMPLEX OF EMPLOYMENTS CURRICULUM CONSTRUCTING

V.G. Malyuga, O.M. Misyura, V.V. Kalachova, O.A. Trublin

An actual task, which stands both before Kharkov the university of Aircrafts (KHUA) and before other higher soldiery educational establishments of Ukraine MF, is highly skilled soldiery specialists preparation. One of possible ways of its decision is consider an address to procedure of planning. A curriculum of employments is part of planning of educational process and serves for connection in the single system of great number of his links and regulates work of his participants. Automation of process of constructing of time-table allows to avoid many errors, subjectivism and considerably abbreviates time taken on this procedure.

In a current period development of complex of the programs of the automated system of constructing of curriculum of employments is carried out in KHUA. Test control of quality of software product must one of final and major design of complex times become. Success of leadthrough of test works in a great deal depends on the observance of basic principles of testing testers; to correct organization of this procedure; maximal account of requirements of customer. The result of the successfully conducted testing is a high-quality software product which answers or even excels the having a special purpose level of the requirements produced to him.

Keywords: curriculum of lessons, quality of software, smoke and regressive testing, bagtrecker, test-case, research testing