

Ю. О. Дажук, магістрант
В. Г. Левицький к.т.н., доц.
В. М. Толкач, магістрант

Житомирський державний технологічний університет

АВТОМАТИЗАЦІЯ ПРОЦЕСУ ТЕСТУВАННЯ ВЕБ-ПРЕЗЕНТАЦІЙ НА ОСНОВІ ВІЗУАЛЬНОГО ПОРІВНЯННЯ З ПОПЕРЕДНІМИ ВЕРСІЯМИ

В сучасних методах розробки програмного забезпечення, сервісів та веб-продуктів процес тестування займає близько 25-30% від загального часу розробки і часто потребує безпосереднє залучення людини. З метою автоматизації процесу створення веб-сайтів та веб-презентацій, в даній статті аналізується авторський інструмент порівняння, аналізу і оцінки веб-продукту на їх візуальну відповідність еталонним моделям та ескізам.

Процес розробки програмного забезпечення є одним із найприбутковіших і найскладніших процесів у сучасному світі. Люди, які працюють у сфері інформаційних технологій високо цінують час як один із основних і невідновних ресурсів.

Шлях зменшення загальної кількості годин розробки ПЗ розвивався на різних рівнях. Були вдосконалені програмні середовища розробки, які пришвидшували написання коду. Сформовано багато організаційних схем для оптимізації процесу, серед яких і парне програмування, і екстремальне програмування, і ітеративні методи розробки. Особливий акцент варто зробити на шляху автоматизації процесу тестування. Впровадження автоматичних тестів у проект дозволяє запобігти широкого діапазону помилок і зберегти систему відносно «чистою» протягом всього процесу розробки, і, відповідно, зменшити час на пошук і виправлення помилок.

Серед наявного арсеналу засобів автотестування найважче знайти інструмент для тестування візуальної частини продукту. Саме для оцінки візуальної частини, для порівняння її відповідності еталонному ескізу чи моделі і залучають тестувальника.

Метою роботи є розробка спеціалізованого програмного продукту для тестування візуальної складової веб-презентацій(створених для роботи на продуктах, що основані на браузерному движку WebKit), процес тестування якої являє собою перевірку її слайдів на ідентичність (або відповідність з певною точністю) еталонним слайдам або, якщо вже була прийнята презентація за еталонну – слайдам попередніх еталонних презентацій.

Під час створення презентації нелегко розміщувати веб-елементи за дизайнерським задумом із піксельною точністю. Ще важче, знайшовши невідповідність, виправити її, не спричинивши нові небажані зміни у розміщенні чи розміри вже існуючих елементів. Ітеративний процес тестування, в такому випадку, на кожному циклі накопичував певну надлишкову роботу – перевірку веб-елементів(розміщення, розмір, колір, шрифти), які вже були протестовані раніше. Для збереження часу тестувальника і позбавлення його від монотонної роботи, було прийнято рішення про автоматизацію частини тестування.

В період пошуку програмного інструменту було знайдено такі продукти: BadBoy і QAVOT, який, на той час, краще підходив для вирішення нашої задачі. Проте, через відсутність супроводу, повільний розвиток продукту і неповне покриття його функціональністю наших спеціалізованих задач, продукт не задовольнив наших потреб – ані BadBoy, ні QAVOT не могли порівняти дві версії одного слайду на ідентичність чи схожість. Тому було прийнято рішення про створення власного інструменту із більш розширеним для сформульованих вимог набором функцій, який отримав назву TwinPics.

Проект був створений із використанням фреймворка для розробки веб-сервісів PlayFramework на мові Scala.

За архітектурну модель програми було обрано клієнт-серверну архітектуру, серверна частина якої являє собою тривірневу систему(згідно шаблону проектування MVC, -- Рис.1).

Перший архітектурний рівень – засоби візуальної взаємодії з користувачем – є проміжним рівнем, введеним для організації зручної взаємодії між користувачем-тестувальником і системою. Графічний інтерфейс створений для виконання основного необхідного комплексу операцій із презентаціями: додання, видалення, огляд, безпосереднє тестування та оповіщення про результати тестування. Кожну презентацію та кожен її слайд можна проглянути, оцінивши зміни у зовнішньому вигляді відносно попередньої сесії тестування та еталонного зображення.

Другим рівнем архітектури є модуль керуванням презентацією(запуск, навігація по слайдам, завершення презентації), збереження візуальної складової слайду як зображення, виконання порівняння відповідних слайдів, аналіз результатів, маркування тих слайдів презентації, результати яких є незадовільними, і оповіщення користувача щодо цих результатів.

Процес порівняння зовнішнього інтерфейсу слайдів відбувається в декілька етапів:

- автоматичний запуск презентації на сервері та запуск Selenium тестів на браузері (наразі реалізована підтримка Mozilla, Chrome та iPhone Safari);
- збереження слайду як зображення (знімається скріншот області документа);
- піксельне порівняння зображень (слайди, які тестувались на попередніх ітераціях, або встановлене еталонне зображення) із встановленою точністю (перевірка на повну ідентичність або на відсоткову частину відхилення);
- оцінка результатів порівняння і маркування несхожих слайдів;
- надання презентації еталонного статусу або статусу презентації, що не пройшла тестування

На третьому рівні схеми структурно-функціональної організації підсистеми управління обробкою даних знаходяться низькорівневі засоби фізичного створення, збереження, модифікації та видалення даних.

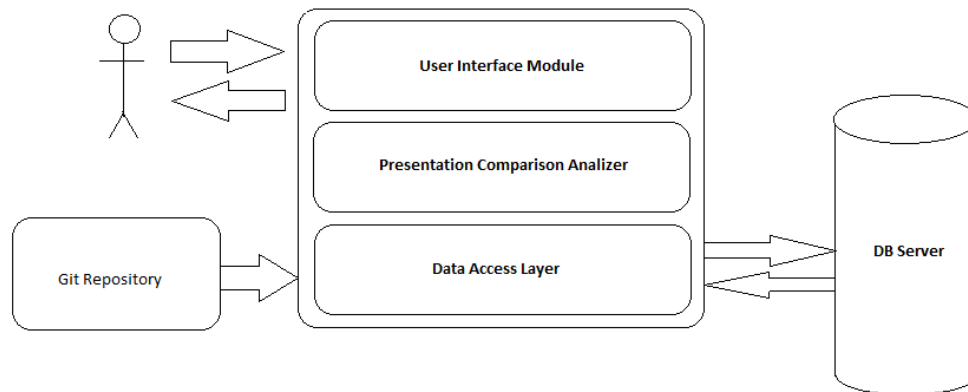


Рис.1 Схема структурно-функціональної організації підсистеми

Завдяки простій та гнучкій модульній архітектурі, в проект легко додаються нові функціональні можливості: використання Git репозиторіїв як джерела презентацій, використання як браузерних програм, так і програм на платформі iOS для прямого контролю та моніторингу процесу тестування. Зараз система широко розвивається в сторону підтримки хмарних технологій (Apple Cloud, DropBox, HiDrive), та розширення набору тестових операцій.

Даний інструмент – не панацея від усіх проблем, які виникають під час перевірки якості програмної системи. Завдяки йому було автоматизовано лише невелику частину від загального комплексу робіт по тестуванню веб-презентацій. Але той факт, що людина звільнена від рутинної перевірки збереження цілісності та дизайнерського задуму на слайдах, і те, що для виявлення помилок варто лише запустити тести та переглянути результат автоматичної перевірки – вже досягнення прогресом на шляху автоматизації перевірки якості. Практичним результатом апробації розробленої системи TwinPics стало зменшення часу тестування презентацій на 7%.

Висновок: був розроблений спеціалізований програмний продукт для тестування візуальної складової веб-презентацій, який автоматизував частину роботи тестувальника і підвищив якісні та кількісні показники у ефективності створенні презентацій, що і було основною задачею розробленої програмної системи.

ДАЖУК Юрій Олександрович – магістр групи ПІ-37м кафедри ПЗОТ, Житомирського державного технологічного університету, прикладний програміст компанії QAR Int. Наукові інтереси: архітектура програмних систем, графічні системи та візуалізація даних.

ЛЕВИЦЬКИЙ В'ячеслав Георгійович – к.т.н., доц., доц. каф. ПЗОТ Житомирського державного технологічного університету. Наукові інтереси: комп'ютерні інформаційні технології, розробка мов програмування та предметно-орієнтованих мов, побудова компіляторів, програмне забезпечення математичного моделювання технічних та екологічних систем, використання обчислювальної техніки в навчальному процесі.

ТОЛКАЧ Віталій Миколайович – магістр групи ПІ-37м кафедри ПЗОТ, Житомирського державного технологічного університету, прикладний програміст компанії QAR Int. Наукові інтереси: архітектура програмних систем, розробка та оптимізація алгоритмів, використання штучного інтелекту в іграх.