

### III. Принципи розробки адаптивних веб-сайтів

Хороша система управління контентом повинна підтримувати динамічний редизайн сайтів, а в її основі мають бути закладені принципи адаптивності.

Проаналізуємо основні принципи розробки адаптивних веб-сайтів [2]:

- 1) Використання резинового шаблону (англ. Fluid grid) - макет, в якому значення розміру ширини батьківського елемента (елементів) задається у відсотках по відношенню до роздільної здатності монітора.
- 2) Використання резинових зображення (англ. Flexible images). Всі зображення повинні масштабуватись, виходячи з гранично допустимого значення ширини. При зміні користувачем вікна оглядача до розміру меншого за розмір зображення, воно має автоматично зменшуватись, щоб вписуватися по ширині у вікно оглядача.
- 3) Використання media queries - правил CSS3, які встановлюються у якості атрибутів при виклику інших правил з таблиці стилів, які базуються на параметрах пристрою виводу, таких як тип, ширина і висота вікна оглядача, дозвіл, орієнтація в просторі.
- 4) Застосування поступового покращення (англ. Progressive enhancement). Ідея полягає в тому, що спочатку створюється найпростіша розмітка документа, яка коректно відображається навіть у найпростіших оглядачах. Потім додаються стилі, інтерактивність та інше, використовуючи каскадні таблиці стилів, JavaScript, SVG, Flash і все інше, що можна вставити в сторінку.
- 5) Проектування для мобільних пристроїв з ранніх етапів (англ. Mobile first). Це принцип, при якому проектування сайту починається з мобільної версії.
- 6) Використання модульної сітки сторінки - це система горизонтальних і вертикальних направляючих, що допомагають зорієнтувати і узгодити між собою окремі елементи сайту. Для генерації модульної сітки зазвичай використовують так звані css-фреймворки.

#### Висновок

Системи управління контентом активно використовуються для швидкої розробки веб-сайтів. У роботі проведено аналіз принципів розробки адаптивних веб-сайтів із використанням CMS, використання яких при розробці сайту дозволяє виконувати його динамічний редизайн та перегляд засобами персональних мобільних пристроїв.

#### Список використаних джерел

1. Nirav Mehta. Choosing an Open Source CMS. Beginner's Guide. – April, 2009. — 340 p.
2. Система керування вмістом. Вікіпедія — вільна енциклопедія. [Електронний ресурс]. – Режим доступу URL: <http://uk.wikipedia.org/wiki/CMS>

УДК 004.054

## МЕТОД “ПРОЗОРОЇ ЖУРНАЛІЗАЦІЇ” ДЛЯ ОРГАНІЗАЦІЇ ПРОЦЕСУ ТЕСТУВАННЯ WEB-ОРІЄНТОВАНИХ ІНФОРМАЦІЙНИХ СИСТЕМ

Мельник А.М.<sup>1)</sup>, Лабик І.Д.<sup>2)</sup>

*Тернопільський національний економічний університет*

<sup>1)</sup> к.т.н., доцент; <sup>2)</sup> магістрант

### I. Постановка проблеми

Інформаційні системи, засновані на технологіях WWW, несуть в собі нові проблеми, як для розробки, так і для тестування [1]. Тести для web-додатків повинні бути орієнтовані на передбачення поведінки вузла. Необхідно оцінювати наступні проблемні моменти: функціональні можливості; практичність; навігацію; форму, вміст сторінки. Детально принципи тестування web-додатків викладені в [2].

Тестування баз даних часто є дуже важливою частиною тестування інформаційних систем. При тестуванні баз даних потрібні всебічні знання програми, яка тестується. До ключових проблем, що

виникають при тестуванні баз даних, відносяться цілісність даних; достовірність даних (форма при введенні в бази даних); маніпуляції з даними і оновлення.

Тестування розподіленого додатку досить складне завдання. Якщо додаток використовує багаторівневу архітектуру «клієнт-сервер», то класичні підходи мають застосовуватися для кожного рівня.

Але при цьому потрібно провести інтеграційне тестування для перевірки системних інтерфейсів і системне тестування для перевірки зовнішніх інтерфейсів (як правило, інтерфейсів користувача або зовнішньої системи). Для складних додатків тестування розподіляється на інкрементне тестування окремих компонент, окремих модулів, окремих підсистем, окремих частин додатку; інтеграційне - тестування відношень усіх частин додатку, інтерфейсу користувача, інтерфейсу системи в цілому.

## II. Мета роботи

Метою дослідження є підвищення ефективності тестування web-орієнтованих інформаційних систем шляхом розробки методу «прозорі журналізації» для організації процесу тестування.

## III. Метод «прозорі журналізації» для тестування web-систем

Процес тестування web-додатку являє собою складну задачу. Для її вирішення необхідно накопичити статистику виконання тестових сценаріїв і станів, визначити структуру системи. Крім того, під час виконання тестових сценаріїв бажано мати можливість швидко змінювати структуру програми та визначати гілку структурного графа, яку необхідно перевірити. Множина значень вхідних параметрів, що задаються через форми web-інтерфейсу, також необхідно накопичувати разом з відповідними повідомленнями системи.

Для проведення тестових заходів пропонується методика «прозорі журналізації», яка нижче буде пояснена на прикладі тестування тільки невеликої частини системи-компонента авторизації на сервері конференції АСІТ.

Суть методу полягає в наступному. У вихідному XML-файлі тестер розставляє спеціальним чином мітки. Далі при виконанні цього коду мітки разом з фрагментами результуючого коду і значеннями відповідних змінних поміщаються в спеціальну таблицю - журнал активності. Формат запису в простому випадку приведений в таблиці 1.

Таблиця 1

Формат запису журналу активності

Назва модуля (компонента)	Мітка вузла	Мітка наступного вузла	Дія/предикат	Значення предиката	Значення, яке повертається	Коментар

Кожен запис являє собою елемент графа структури тестованого додатку. Набір записів повністю відображає таблицю на графі. Фрагмент графа структури наведено на рисунку 1, а відповідний йому набір записів у журналі активності - в таблиці 2.

Таблиця 2

Приклад формату записів журналу активності

Назва модуля (компонента)	Мітка вузла	Мітка наступного вузла	Дія/предикат	Значення предиката	Значення, яке повертається	Коментар
http://acit.tneu.edu.ua/administratrator/	[1]	[2]	If name = "\$Person/ID\$" value="" op="eq"	True		Значення Person/ID = null
http://acit.tneu.edu.ua/administratrator/	[2]	[3]	conf/{conferenceAliases}/reg/ins_person.htm	Значення параметрів		Передача формі управління
http://acit.tneu.edu.ua/administratrator/	[1]	[4]	##LAST_NAME## ##FIRST_NAME## ##MIDDLE_NAME## ##ORGANISATION#	Значення змінних	Звідки повернулись	Вивід в XML файл

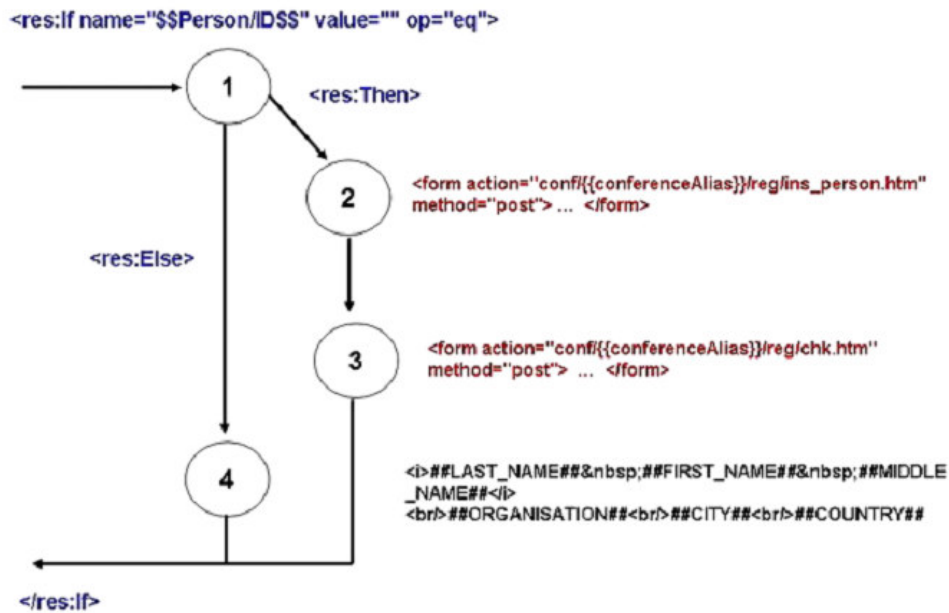


Рисунок 1 – Фрагмент графа структури додатку

У рядку таблиці з номером 1 перевіряється стан авторизації користувача (вузол з номером 1 на рисунку 1). Якщо значення змінної `$$Person/ID$$` невизначено, то користувач не пройшов реєстрацію в системі і йому необхідно заповнити свої дані за допомогою спеціальної форми (вузол з номером 4 на рисунку 1).

Якщо користувач почав процес реєстрації в системі (вузол з номером 4 на рисунку 1) за допомогою спеціальної форми, то його параметри можуть бути поміщені в стовпчик 4 таблиці 2 (змінні `##LAST_NAME##`, `## FIRST_NAME ##`, `## MIDDLE_NAME ##`, `## ORGANISATION ##`, `## CITY##`, `##COUNTRY##`) і можуть бути використані для проведення детального тестування процесу реєстрації.

Після виконання необхідних тестових сценаріїв тестер має у своєму розпорядженні структуру частин модуля (компонента) системи, набір значень переданих/прийнятих змінних, набір повідомлень системи (поміщаються в останнє поле) у разі виникнення проблемних ситуацій. На підставі цих даних можна провести аналіз покриття структури додатку. Аналіз покриття дає можливість оцінити ефективність тесту (з використанням спеціалізованих інструментів тестування) і провести спостереження за напрямками виконуваного коду. Журнал активності зберігається в спеціальній схемі бази даних, де і проходить накопичення необхідних статистичних даних.

### Висновок

Запропонований метод об'єднує на своїй основі стратегії інтеграційного тестування тестування модулів/компонент в єдиному додатку із використанням підходів, притаманних функціональному тестуванню, тобто методика «сірого ящика») і системного тестування (тестування зовнішніх інтерфейсів або інтерфейсів користувача).

### Список використаних джерел

1. Splaine S., Jaskiel S.P., Savoia A. The Web Testing Handbook // Software Quality Engineering. 2001
2. Блейзер Б. Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем. СПб.: Питер, 2004.