

УДК 004.7

І.О. Кутенський, А.С. Чуприна

Харківський національний університет радіоелектроніки, Харків

ДОСЛІДЖЕННЯ МЕТОДІВ ТА ЗАСОБІВ РОЗРОБКИ INTERNET ДОДАТКІВ ДЛЯ ВИЗНАЧЕННЯ НАЙБІЛЬШ ЕФЕКТИВНОГО ПІДХОДУ

В статті розглядаються існуючі методи розробки веб-сайтів та проводиться їх порівняльний аналіз, з метою обрати кращу комбінацію технологій для застосування у розробці.

Ключові слова: веб, asp.net, php, knockout, angularJs, javascript.

Вступ

Швидкість, з якою поширюється інформація сьогодні, є вражаючою. Інтернет став каталізатором до поширення інформації. В Інтернеті знаходяться мільйони веб-сайтів різного спрямування. Створення та підтримка веб-сайту є важкою працею.

Саме тому все більше поширюються технології розробки, що конкурують один з одним у простоті та можливостях створення веб-сайтів. Проте кожна технологія має свою особливість, плюси та мінуси.

Мета даної статті – розглянути декілька технологій розробки веб-сайтів та провести їх оцінку. Треба оцінити які технології є найбільш простими та дозволяють створити веб-сайт швидше за все, а які надають сайту найбільше можливостей й гнучкості. Треба спробувати знайти лідера серед обраних технологій, або виявити в яких ситуація слід надавати перевагу певній технології.

1. Критерії порівняння технологій

Для порівняння обраних технологій слід обрати такі критерії порівняння, що будуть найкраще демонструвати переваги та недоліки кожної. Критерії порівняння мають продемонструвати які технології краще використовувати в яких ситуаціях і які в них є можливості. Беручи до уваги те, що порівнюватися будуть технології з окремих груп (серверні та клієнтські технології), необхідно обрати різні критерії порівняння для різних груп.

Приведемо перелік обраних критеріїв порівняння серверних технологій:

- перелік СКБД, що підтримуються технологією. Наявність підтримки тої чи іншої СКБД впливає на можливість використання технології, коли проект, під який розроблюється веб-сайт, вже має налагоджену і заповнену базу даних. Також різноманітність підтримуваних СКБД демонструє можливість до масштабування веб-сайту;

- важкість вивчення. Важкість вивчення – важливий критерій, що визначає поріг входження но-

вих розробників в технологію. Проте низький поріг входження також може значати більшу кількість проектів, що написані новачками й що мають низьку якість коду;

- швидкодія. Цей критерій має дуже велику цінність, бо часто саме він є домінуючим при виборі технологій. Часто низька швидкодія є фактором, що відштовхує користувачів при користуванні веб-сайтом;

- масштабованість. Цей критерій також має велику цінність, бо визначає можливість витримувати велику кількість користувачів, що одночасно використовують веб-сайт.

Приведемо перелік обраних критеріїв порівняння клієнтських технологій:

- важкість вивчення. Аналогічним до серверних технологій чином впливає на поріг входження. Крім того визначає можливість міграції між різними фреймворками;

- швидкодія. Якщо у випадку з серверними технологіями, швидкодія визначає навантаження на сервер та швидкість відповіді на велику кількість запитів, що також впливає на масштабованість, то для клієнтських технологій швидкодія – це відгук веб-сторінки на дії користувача. Висока швидкодія надає веб-сайту інтерактивності й відчуття користувача, що інтерфейс миттєво реагує на його дії і перебудовується відповідно до його потреб.

2. Порівняння серверних технологій

У порівнянні серверних технологій будуть розглянуті ASP.NET та PHP.

2.1. Перелік СКБД

ASP.Net: Microsoft BI, MySQL, Oracle, SQLite, PostgreSQL, IBM DB2, MongoDB, Redis, Microsoft SQL Server;

PHP: Microsoft Access, PostgreSQL, MongoDB, CUBRID, Microsoft BI, InterBase, Informix, Oracle, IBM DB2, SQLite, MySQL, MariaDB, PHP/PDO, firebird, CouchDB, BangDB.

Отже, PHP має найбільшу кількість підтримуваних СКБД, що логічно, бо ця технологія найбільш стара і за довгі роки отримала дуже багато модулів, що надали їй такого різноманіття підтримуваних СКБД.

2.2. Важкість вивчення

ASP.NET застосування зазвичай розробляються на C# [1]. C# заснований на мові C, яка може бути важкою для вивчення для більшості новачків. Його синтаксис є складним і важким для читання, навіть для досвідчених програмістів. Люди, незнайомі з програмуванням, знайдуть ASP.NET не кращим вибором.

З іншого ж боку PHP може бути кращим вибором для вивчення (через це навіть серед досвідчених програмістів популярна думка, що усі PHP розробники - новачки).

Завдяки цьому досить вивчити декілька ключових моментів PHP щоб почати розроблювати сайти на WordPress майже одразу.

2.3. Швидкодія

Швидкодія для більшості веб-додатків є взаємодією між кодом, базою даних і сервером. Більшість веб-додатків, що написані на PHP, використовують LAMP стек - Linux (ОС), Apache (сервер), MySQL (база даних) і PHP (скриптова мова) [2]. LAMP стек є надзвичайно популярним для веб-розробки, тому оптимізований для підвищення продуктивності.

Базою даних, що найбільш часто використовується з ASP.NET, є MSSQL (Microsoft SQL Server), хоча можна також використовувати MySQL та багато інших баз даних. Різниця швидкодії ASP.NET + MSSQL і PHP + MySQL дуже мала, але все ж таки PHP + MySQL трохи обходить ASP.NET + MSSQL.

Іншим фактором, який впливає на швидкодію є ОС і файлова система, що використовується на сервері. Більшість тестів показують, що Linux з файловою системою ext4 має більш високу швидкість читання та запису у порівнянні з Windows з її файловою системою NTFS.

Таким чином, більш ймовірно, що додаток PHP працюватиме на веб-хостингу Linux трохи швидше аналогічної програми на ASP.NET на ОС Windows.

Проте слід пам'ятати, що не аби який вплив має якість написаного коду та архітектура бази даних.

2.4. Масштабованість

І ASP.NET і PHP мають високу масштабованість. В якості прикладу можна згадати, що Facebook, другий найбільш інтенсивно зростаючий сайт у світі, побудований на PHP, а MySpace, що був витіснений Фейсбуком, побудований на ASP.NET. Це ілюструє, що як ASP.Net так і PHP високо масштабовані, за умови, що програмісти знають, як масштабувати їх продукти

3. Порівняння клієнтських технологій

У порівнянні клієнтських технологій будуть розглянуті два JavaScript фреймворки: Knockout та AngularJS

3.1. Важкість вивчення

В більшості випадках, knockout [3] легший для вивчення. Зв'язування даних у Angular [4] є більш потужним, ніж у knockout, але те як воно виконується може бути дуже важким для розуміння. Angular спирається більше на "магію" - він просто працює, хай навіть ця робота не є прозорою для розробників.

Проте Angular не страждає від проблем, що має knockout з більш складними моделями: Knockout накладає значні обмеження на архітектуру застосування. Прив'язка даних почне працювати лише при правильній конструкції моделей і логіки взагалі. Тому архітектура таких застосувань буде дуже специфічною, порівняно з іншими веб-застосуваннями, тому багато яких додаткових бібліотек не будуть працювати при використанні knockout. Angular, з іншого боку, не накладає таких жорстких рамок на архітектуру моделей і може бути використаний у більшій різноманітності задач.

Angular також має додаткову перевагу, що не вимагає від розробник використовувати observable змінні у моделі. Для того, щоб knockout був у змозі зв'язати дані з DOM, вони мають бути обгорнуті в observable. Angular же використовує більше магії для вирішення цієї задачі. Angular використовує легшу логіку зв'язування даних, хоча й не таку прозору й зрозумілу як observable змінні в knockout.

Отже knockout дійсно кращий для вивчення, бо має більш прозору й зрозумілу логіку роботи. Навпаки ж, Angular важчий для вивчення і розуміння.

3.2. Швидкодія

Для зв'язування даних у Knockout була використана система відстеження змін [5]. Ви працюєте з даними не безпосередньо, а через спеціальний шар (observables), мета якого вчасно міняти подання даних при їх зміні. Будь-яка змінна перетворюється на об'єкт фреймворку, який стежить за своїм станом. Спрацьовування системи відстеження змін в основному відбувається при кожній зміні, адже у своїй основі це просто коллбек функції. Якщо ми змінюємо дані безперервно або великими порціями відразу, то це викликає дуже багато непотрібних спрацьовувань. Так як кінцева мета - всього лише змінити відображення, яке так чи інакше буде приведено до бажаного кінцевого виду, проміжні стани тут не потрібні.

З іншого боку, Angular, що використовує принцип Dirtychecking. Dirtychecking – це перевірка на оновлено даних. Раніше змінна була 1, тепер вона 2 – значить, дані змінилися і треба їх у шаблоні пе-

ремалювати. Це найпростіший підхід, який позбавляє нас як від необхідності працювати з даними через спеціальний «observable» шар, так і від усіх проблем, пов'язаних із залежностями даних. Проте такий підхід призводить до іншої проблеми – надмірної кількості порівнянь даних. Тому в Angular модифікували підхід до Dirtychecking. Перевірка виконується тоді і тільки тоді, коли дані могли бути змінені (наприклад, при дії користувача), і ніколи не виконується в інших випадках. Якщо ви очікуєте зміни даних у якийсь інший момент часу (наприклад, при вступі події від сервера або завершенні якого-небудь процесу), необхідно явно вказати Angular, що варто виконати перевірку, викликавши відповідну функцію.

Очевидний мінус цього підходу – швидкодія. Хоча й тут є невеликий виняток: наприклад, при пакетному оновленні відразу великої кількості даних перевірка виконується всього один раз в кінці, а не при кожній зміні кожного з відстежуваних об'єктів, як це відбувається у knockout.

Але в цілому, це тим не менше мінус, так як при зміні всього однієї змінної виконується dirtycheck всіх даних.

Потрібно лише зрозуміти наскільки велике падіння продуктивності. Тут варто відзначити, що Angular під час виконання dirtycheck ніколи не працює з DOM. Всі дані – це об'єкти js, з якими всі сучасні движки браузерів блискавично виконують більшість основних операцій. Хоча ви й можете самі вставляти процедури перевірки в процес dirtycheck, документація Angular наполегливо рекомендує не працювати з DOM всередині них, так як це може сильно уповільнити весь процес.

4. Підсумки порівняння

Серверні технології:

	ASP.Net	PHP
Кількість СКБД	9	16
Простота	-	+
Швидкодія	-	+
Масштабованість	+	+

Беручи до уваги обрані критерії, серед серверних технологій найбільш ефективною можна вважати PHP, тому що вона має більше підтримуваних СКБД, легша для вивчення та має трохи кращу швидкодію.

Клієнтські технології:

	Knockout	AngularJS
Простота	+	-
Швидкодія	-	+

Серед клієнтських технологій обрати найбільш ефективну технологію важче. З одного боку, knockout легша для вивчення і застосування, та має свої переваги, проте Angular має кращу архітектуру веб-застосувань, вищу швидкодію і підходить як технологія «все в одному». Тому можна зробити висновок, що клієнтську технологію слід обирати виходячи з додаткових вимог до проекту, а також від досвідченості розробників. Для новачків краще підійде Knockout, в той час як досвідчені розробники можуть взятися за AngularJS.

Список літератури

1. Сандерсон С. ASP.NET MVC Framework с примерами на C# для профессионалов [Текст] / С. Сандерсон. – М.: ООО «И.Д. Вильямс», 2010. – 560 с.
2. Суэринг С. PHP и MySQL. Библия программиста, 2-е издание [Текст] / С. Суэринг, Т. Конверс, Дж. Парк; пер. с англ. К. Птицын. – М.: Диалектика, 2010. – 912 с.
3. Документація KnockoutJS [Електронний ресурс] / портал knockoutjs.com / KnockoutDocumentation. – Режим доступу до ресурсу: [www/URL:http://knockoutjs.com/documentation/introduction.html](http://knockoutjs.com/documentation/introduction.html).
4. Документація AngularJS [Електронний ресурс] / портал docs.angularjs.org. AngularJS: DeveloperGuide. – Режим доступу до ресурсу: [www/URL:https://docs.angularjs.org/guide](https://docs.angularjs.org/guide).
5. Ключевое отличие AngularJS от Knockout [Електронний ресурс] / портал habrahabr.ru: Колективний блог. – Режим доступу до ресурсу: [www/URL:http://habrahabr.ru/post/165275/](http://habrahabr.ru/post/165275/).

Надійшла до редколегії 21.07.2015

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

ИССЛЕДОВАНИЕ МЕТОДОВ И СРЕДСТВ РАЗРАБОТКИ INTERNET ПРИЛОЖЕНИЙ ДЛЯ ОПРЕДЕЛЕНИЯ НАИБОЛЕЕ ЭФФЕКТИВНОГО ПОДХОДА

И.О. Кутенский, А.С. Чуприна

В статье рассматриваются и сравниваются существующие методы разработки веб-сайтов с целью выбрать лучшую комбинацию технологий для применения в разработке.

Ключевые слова: веб, asp.net, php, knockout, angularJs, javascript.

RESEARCHING OF METHODS AND WEB APPLICATIONS DEVELOPMENT TOOLS TO FIND THE MOST EFFICIENT APPROACH

I.O. Kutenskiy, A.S. Chupryna

In this article exploring and comparing the existing methods of developing web sites in order to choose the best combination of technologies to be used in development.

Keywords: web, asp.net, php, knockout, angularJs, javascript.