

УДК 519.6, 004.7

**Проценко Михайло Михайлович**

студент

Національний технічний університет України

«Київський політехнічний інститут»

**Проценко Михаил Михайлович**

студент

Национальный технический университет Украины

«Киевский политехнический институт»

**Protsenko Mikhail M.**

Student

**АНАЛІЗ ФРЕЙМВОРКІВ ЯК ЗАСОБІВ РОЗРОБКИ WEB-ДОДАТКІВ**

**АНАЛИЗ ФРЕЙМВОРКОВ КАК СРЕДСТВ РАЗРАБОТКИ WEB-  
ПРИЛОЖЕНИЙ**

**ANALYSIS FRAMEWORKS AS TOOLS FOR DEVELOPING WEB-  
BASED APPLICATIONS**

**Анотація:** Досліджені переваги та недоліки використання фреймворків для розробки Web-додатків. Виділено критерії вибору фреймворку для розробки Web-додатку.

**Ключові слова:** MVC, фреймворк, Back-End, web-додатки, база даних.

**Аннотация:** Исследованы преимущества и недостатки использования фреймворков для разработки Web-приложений. Выделены критерии выбора фреймворка для разработки Web-приложения.

**Ключевые слова:** MVC, фреймворк, Back-End, web-приложения, база данных.

**Summary:** The advantages and disadvantages of using frameworks for developing Web-based applications. Highlight the selection criteria framework for developing Web-based application.

**Key words:** MVC, Framework, Back-End, web-applications, Database.

## ВСТУП

Швидкість, з якою поширюється інформація сьогодні, є вражаючою. Інтернет став каталізатором до поширення інформації. В Інтернеті знаходяться мільйони веб-сайтів різного спрямування. Розвиток Інтернету нерозривно пов'язано з проектуванням сайтів. Масова поява сайтів спровокувала проблему їх якості. Популярність створення веб-ресурсів сприяла розробці різних систем і програм, які спрощують процес написання сайту. Також вони допомагають підвищити ефективність роботи, а також дозволяють розробнику сфокусуватися на основною логікою програми. Такі технології, як PHP, Java, Microsoft.Net, MySQL, Oracle, Microsoft SQL Server і розроблені на їх основі фреймворки - це каркаси системи або підсистеми, що можуть включати допоміжні програми, мови сценаріїв - і все, що полегшує розробку й об'єднання різних компонентів.

Використання фреймворків стає все більше і більше популярним, і це обгрунтовується тим, що розробка з допомогою фреймворку зменшує навантаження на процес розробки web-додатків, це досягається тим, що розробка з використанням фреймворку позбавляє від проблеми використання повторюваного коду. Без використання фреймворків, стає набагато складніше створювати web-додатки, супроводжувати і модернізувати їх. Між тим, використання фреймворків робить процес створення програми набагато більш легким і функціональним. Проаналізувавши інформацію з мережі Інтернет, можна побачити що існує сотні фреймворків для створення web-додатків. Тому досить складно зробити вибір фреймворку, так як кожен з них має велику кількість привабливих функцій та доповнень. А неправильний вибір фреймворку може стати основною причиною невдачі проекту.

## **Що являють собою Web-фреймворки?**

WEB фреймворк - це каркас, призначений для створення динамічних веб-сайтів, мережевих додатків, сервісів або ресурсів. Він спрощує розробку і позбавляє від необхідності написання рутинного коду. Багато фреймворків спрощують доступ до баз даних, полегшують розробку інтерфейсу, а також зменшують дублювання коду .

Виділяють п'ять типів веб-фреймворків: Request-based, Component-based, Hybrid, Meta and RIA-based.

Request-based - фреймворки, які безпосередньо обробляють вхідні запити. Збереження стану відбувається за рахунок серверних сесій. Приклади: Django, Ruby on Rails, Struts, Grails.

Component-based - фреймворки, які абстрагують обробку запитів всередині стандартних компонентів і самостійно стежать за станом. Дані каркаси мають багато спільного зі стандартними програмними графічними інтерфейсами. Приклади: JSF, Tapestry, Wicket.

Hybrid-based - фреймворки, які комбінують Request-based та Component-based фреймворки, беручи під свій контроль всі дані і логічний потік в заснованої на запиті моделі. Розробники мають повний контроль над URL, формами, параметрами, cookies і pathinfos. Однак замість того, щоб відобразити дії і контролери безпосередньо до запиту, гібридні фреймворки забезпечують об'єктну модель компонентів, яка поводить себе тотожно в багатьох різних ситуаціях, таких як окремі сторінки, перервані запити, подібні порталу фрагменти сторінок та інтегровані віджети. Компоненти можуть розподілятися окремо і ефективно інтегруватися в інші проекти. Приклади: RIFE.

Meta-based – фреймворки, що мають ряд базових інтерфейсів для загального обслуговування і основу, яка легко розширюється з метою інтегрування компонентів і служб. Приклад: Keel.

RIA-based (фреймворки для розробки Rich Internet Applications (RIA) – фреймворки, що служать для розробки повноцінних додатків, які запускаються всередині браузера. Приклад: Flex.

Найбільш поширеними є Request-based і Component-based веб-фреймворки. Зібравши і проаналізувавши інформацію, мною було виділено такі характерні компоненти web фреймворків:

- Шаблонизатор. Відповідає за незалежність верстки від програмного коду.
- Роутер. Розпізнає URL, за яким відбулося звернення до сервера.
- Модуль доступу до бази даних.
- Модуль кешування. Прискорює завантаження сторінок.
- Модуль безпеки. Аутентифікація і авторизація користувачів.
- Файли конфігурації.

WEB фреймворки також можуть керувати сесіями, вести логи, спрощувати використання технології Ajax та ін.

### **Переваги використання фреймворків для розробки Web-додатків**

Під час вибору фреймворку можуть виникнути певні труднощі, пов'язані з визначенням завдань, які він може виконувати, та його призначення. Якщо для створення сайту потрібно знайти зручний і простий в освоєнні фреймворк, то необхідно ретельно підійти до питання його вибору та зважити всі «за» і «проти».

Розглянемо загальні переваги використання фреймворку:

- гнучкість і масштабування - завжди має гнучке рішення нестандартних завдань і можливість далі розширення функціоналу шляхом підключення сторонніх бібліотек або окремих класів; ефективно використання ресурсів сервера;
- використання підходу модель-вид-контролер (MVC) суттєво розширює функціонування та гнучкість проекту, так як використовується під час

проектування та розробки програмного забезпечення. фреймворки написані розробниками для розробників, що дозволяє мати на виході прекрасно написаний код і своєчасне виправлення помилок;

- наявність детальної документації з використання фреймворку;
- безпека - забираються всі проломи в безпеці, практично немає вузьких місць для SQL-ін'єкцій; фреймворк дозволяє сконцентруватися на вирішенні архітектурних завдань, а не базових, як при розробці без його застосування;
- якість матеріалу на виході

Головна перевага фреймворків , те що вони якнайкраще підходять для створення масштабованих і унікальних сайтів . Жоден масштабний проект не розроблений на готовій CMS – вони для цього не призначені. Майже всі унікальні web-додатки розробляються з використанням фреймворків. Web-проект, розроблений за допомогою фреймворку, розвивається динамічно. При зміні вимог змінюється і сайт, для створення нового розділу або внесення новизни в дизайн , достатньо змінити окремих модулів.

замінити окремих блок (модуль), створити новий розділ або внести новизну в дизайн.

### **Недоліки використання фреймворків для розробки Web-додатків**

Недоліки застосування фреймворку досить умовні і незначні порівняно з перевагами:

- Важко обслуговувати - якщо проект створював один розробник, а потім з якихось причин він пропадає або просто відмовляється супроводжувати створений ним проект, то його подальший розвиток і обслуговування стає більш складним питанням і часто не вигідним заняттям;
- Ціна розробки - вартість стандартного сайту зробленого на фреймворку з нуля буде дорожче, ніж на готовій CMS, тому що часу

на розробку витратиться в кілька разів більше.багато коду не використовується і лежить мертвим вантажем в проекті;

- складність в освоєнні;
- відсутність готових модулів і компонентів, які міг би встановити клієнт, в мережі Інтернет немає ані безкоштовних, ані платних. Всі доробки необхідно замовляти у розробників;

## **Шляхи та області використання**

Для того щоб скористатися всіма можливостями фреймворка потрібен чималий багаж знань в розробці додатків. фреймворки можуть допомогти усунути дуже часту помилку при програмуванні додатків, а саме повторення коду, а також систематизувати процес розробки. Фреймворк є потужним інструментом для такої мови програмування, що швидко розвивається, як РНР, який допоможе організувати ваш код.

Кожна людина має різні вподобання і потреби. Для одного розробника використання фреймворків може допомогти у прискоренні процесу програмування, а для іншого це може здатися марною тратою часу. У більшості випадків це залежить від рівня професіоналізму, але, загалом, фреймворки призначені, щоб заощадити час і абстрагуватися від рутинних завдань.

В основному, фреймворки застосовуються для розробки проектів складніше, ніж дво- або тресторінковий сайт з текстовими сторінками.

## **Критерії критерії вибору фреймворку**

Фреймворки використовуються в різних областях залежно від їх основних характеристик. Виділимо основні критерії вибору фреймворку в таблиці 1

Таблиця 1– Критерії раціональності використання фреймворку

<b>Назва</b>	<b>Характеристика</b>
<b>Підтримка баз даних</b>	Питання підтримки баз даних в фреймворк дуже важливе. Наприклад частина фреймворків мають вбудований ORM – шар, частина – ні. Залежно від використовуваної бази даних для розробки проекту доводиться вибирати той чи інший фреймворк.
<b>Підтримка спільноти</b>	При розробці додатків, часто виникають моменти коли він сам не в змозі впоратись, з вирішенням проблеми , але при наявності великої спільноти розробників готових допомогти можна вирішити більшість проблем.
<b>Документація</b>	Деякі фреймворки не мають достатньої кількості документації, або вона є не актуальною. Тому перед вибором фреймворка необхідно переконатися в тому що документація актуальна, вчасно оновлюється і доповнюється, і що інструкція із застосування проста в розумінні.
<b>Продуктивність</b>	Ключовим фактором при виборі фреймворку все ж залишається продуктивність. Так як частина фреймворків, має низьку продуктивність.
<b>Безпека</b>	Частина фреймворків немає захисту від SQL-ін'єкцій , відсутність цього функціональ може негативно вплинути на роботу сайту вцілому і привести до краді інформації.
<b>Поріг освоєння</b>	Не всі фреймворки прості в освоєнні, це дуже важливо враховувати при виборі, так як на освоєння одного фреймворка може не вистачити й року, а на освоєння іншого – вистачить всього тижня.

Таблиця 1 – Критерії раціональності використання Back-end фреймворку(Продовження)

<b>Назва</b>	<b>Характеристика</b>
<b>Швидкість розробки</b>	Доцільно виділити, що на різних фреймворках і різна швидкість розробки проектів. Приміром, розробка із застосуванням фреймворка zend триває більше ніж із застосуванням уїі .
<b>Архітектура</b>	Для зручного написання сайту фреймворк повинен використовувати MVC архітектуру.
<b>Швидкість розвитку</b>	Цей пункт так само дуже важливий, так як деякі фреймворки оновлюються раз на пару років (codeigniter), а деякі раз на пару місяців. Це дозволяє уникнути використання старого, недопрацьованого коду при розробці.
<b>Використання шаблонів при створенні інтерфейсів користувача</b>	Наявність даної функції значно спрощує створення інтерфейсу сайту , а іноді коли потрібний простий сайт то може звести написання дизайну до мінімуму.
<b>Створення і перевірка форм</b>	Наявність даного функціоналу в фреймворку полегшує створення полів для вводу тексту, логіну чи паролю до мінімуму. А також додає потрібну майже в усіх проектах функцію валідації форм.

## **Висновки**

Системи створення web-додатків або фреймворки активно використовуються розробниками при створенні web-додатків із різним функціоналом і рівнем складності. Аналіз основних характеристик і можливостей сучасних фреймворків дозволяє вибрати оптимальний варіант для конкретних web-додатків з урахуванням поставлених завдань. Вибір та використання фреймворків відіграє важливу роль при проектуванні, реалізації та супроводі як простих web-додатків, так і складних програмних комплексів.



### Література:

1. Результаты тестирования шести ведущих фреймворков на производительность [Электронный ресурс] – Режим доступа до ресурсу: <http://www.alrond.com/ru/2007/jan/25/rezultaty-testirovaniya-6-frameworks/>.
2. Використання PHP фреймворків в розробці сайту [Електронний ресурс] – Режим доступа до ресурсу: <http://ukrbukva.net/page,5,39718-Ispol-zovanie-PHP-freymvorkov-v-razrobotke-saiyta.html>.
3. Сравнение каркасов веб-приложений [Электронный ресурс] – Режим доступа до ресурсу: [https://ru.wikipedia.org/wiki/Сравнение\\_каркасов\\_веб-приложений](https://ru.wikipedia.org/wiki/Сравнение_каркасов_веб-приложений).
4. Обзоры Web-фреймворков [Электронный ресурс] – Режим доступа до ресурсу: <https://praktikatech.wordpress.com/category/обзоры-web-фреймворков/>
5. Что такое фреймворк? [Электронный ресурс] – Режим доступа до ресурсу: <http://www.dbhelp.ru/what-is-framework/page/>.
6. Десять причин избегать тяжеловесных фреймворков, а также лишних зависимостей в проекте [Электронный ресурс] – Режим доступа до ресурсу: <http://eax.me/avoid-frameworks/>.