

Analiza porównawcza technik i technologii informatycznych wspomagających proces projektowania i implementacji serwisów webowych

Grzegorz Ruciński

Promotor dr inż. Paweł Figat

Warszawska Wyższa Szkoła Informatyki

2011

Plan prezentacji

- Cel i hipoteza pracy
- Wprowadzenie do tematu
- Przedstawienie porównywanych rozwiązań
- Przedstawienie zalet i wad porównywanych rozwiązań
- Zakres objęty analizą
- Rezultaty prowadzonej analizy

Wprowadzenie

Celem pracy jest przedstawienie technik i technologii informatycznych wspomagających proces projektowania i implementacji aplikacji webowych w oparciu o wzorce projektowe oraz rozwiązania ORM na podstawie porównania czterech najbardziej znanych framework'ów dla PHP z zaimplementowaną obsługą wzorca architektonicznego MVC.

Przeprowadzona analiza miała na celu uzasadnić szczegółowo sytuacje w których mają zastosowanie framework'i, wzorce projektowe oraz inne narzędzia wspomagające proces implementacji aplikacji webowych.

Zagadnienie

Tworzenie prostych aplikacji samodzielnie, bez stosowania zewnętrznych rozwiązań jest łatwe i przyjemne. Można bez problemu szybko nanosić zmiany w kodzie. Z biegiem czasu, gdy tworzona aplikacja staje się na tyle duża, że wymaga zaangażowania nowych osób mogą wystąpić problemy z wdrożeniem tych osób w rozwój aplikacji, zwiększa się wtedy zagrożenie bezpieczeństwa aplikacji, czytelność kodu może znacznie się pogorszyć i w konsekwencji sprawić, że aplikacja będzie mało wydajna. Naprzeciw tym problemom powstały rozwiązania określane mianem framework'ów, w których zaimplementowano wzorce projektowe oraz inne narzędzia takie jak ORM. Większość rozwiązań posiada bogatą bibliotekę klas, które zostały przetestowane i dają gwarancję, że kod jest wysokiej jakości. Głównym powodem dla którego stosowane są framework'i jest szybkość tworzenia gotowych aplikacji. Framework'i stanowią dla programistów fundamentalną strukturę do budowania aplikacji internetowych, ułatwiają pracę programistom poprzez rozwiązania, które umożliwiają szybkie postępy w pracy, przy jednoczesnym elastycznym zachowaniu tworzonej aplikacji.

Zagadnienie

Tworzenie aplikacji webowych w oparciu o wzorce projektowe, framework'i, ogranicza lub likwiduje następujące problemy:

- Monotonia pisania kodu
- Zmniejsza ryzyko wystąpienia błędów
- Narzuca standard
- Przyspiesza tworzenie aplikacji

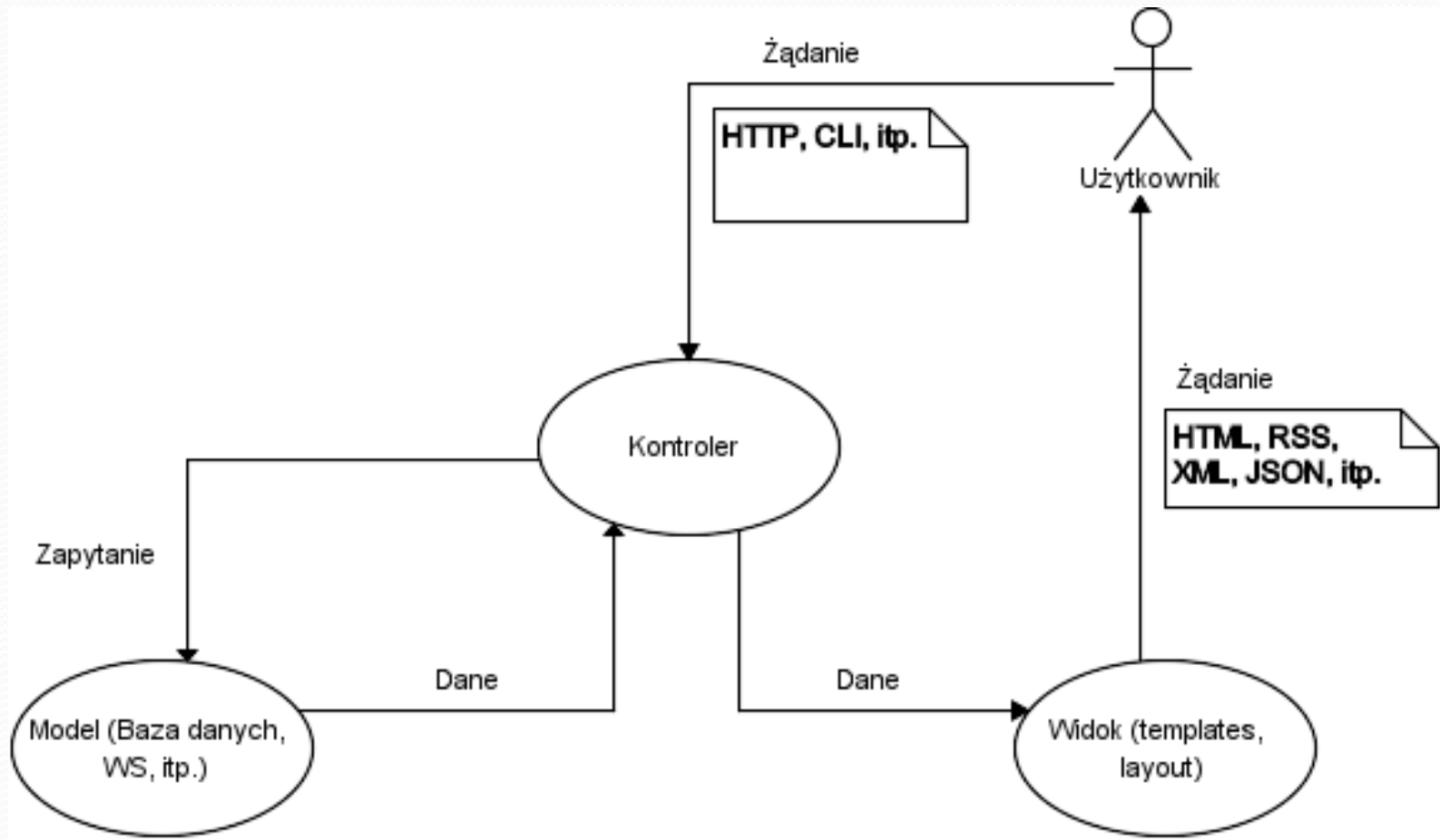
Porównywane rozwiązania



ZEND
FRAMEWORK



Wzorzec architektoniczny MVC



Zalety i wady wzorca architektonicznego MVC

Zalety

- **Łatwa modyfikacja widoku** – model jest niezależny od widoku
- **Brak zależności modelu od widoku** – modyfikacja warstwy widoku nie narzuca potrzeby modyfikowania warstwy logiki

Wady

- **Duża złożoność** – daleko posunięta abstrakcja rozwiązania przekłada się na trudniejsze rozumienie zasad działania rozwiązania i przyszłego testowania
- **Trudne testowanie widoków** – złożone widoki są trudne do testowania jeżeli w ich budowie znalazła się dodatkowa logika
- **Kosztowna zmiana modelu** – jeżeli modyfikacji ulega warstwa modelu może dojść do potrzeby modyfikacji warstwy widoku

Analiza framework'ów

- **Wymagania**
- **Instalacja i konfiguracja**
- **Dostępne funkcje, możliwości:**
 - PDO, ORM, CRUD – obsługa baz danych
 - RBAC, ACL – obsługa uwierzytelniania, nadawania praw
 - Wielojęzyczność – tworzenie aplikacji wielojęzycznych
 - Filtracja i walidacja – tworzenie filtrów oraz walidatorów
 - Routing – obsługa adresów URL
- **Rozmieszczenie katalogów, nazewnictwo plików** – sposób nazywania katalogów, plików, klas, metod oraz nazw tabel i kolumn w bazie danych
- **Wydajność utworzonej aplikacji**
- **Bezpieczeństwo** – ochrona przed atakami SQL Injection, XSS, CSRF
- **Testowanie i narzędzia deweloperskie** – narzędzia do testowania i rozbudowy aplikacji

Zalety i wady framework`ów

Zalety

- **Efektywność** – wymagana mniejsza ilość kodu do napisania przez programistę
- **Wysoka jakość kodu** – rozwiązania posiadają dobrą organizację i logikę
- **Niezawodność** – framework'i są dobrze zaprojektowane i przetestowane

Wady

- **Złożoność** – duża złożoność przekłada się na trudniejsze opanowanie narzędzia
- **Wydajność** – duża elastyczność kodu przekłada się na niższą wydajność utworzonej aplikacji

Zalety i wady narzędzi ORM

Zalety

- **Szybkość tworzenia kodu** – tworzenie kodu jest szybkie
- **Rozwiązanie niezależne od typu bazy danych** – utworzony kod jest niezależny od użytej bazy danych
- **Elastyczność przy modyfikowaniu bazy danych** – modyfikowanie nazw i typów pól w bazie danych nie wymaga modyfikacji kodu (większości przypadków)

Wady

- **Wydajność utworzonych zapytań** – utworzone zapytania są mniej wydajne od tych utworzonych w standardowy sposób
- **Problem z odwzorowania dziedziczenia i polimorfizmu**
- **Nieczytelny kod** – utworzony kod jest z biegiem czasu staje się coraz bardziej nieczytelny

Narzędzia deweloperskie

Logs

[all] [none]    Doctrine_Connection_Mysql Doctrine_Connection_Statement jobActions main sfFilterChain sfPHPView sfPartialView sfPatternRouting sfWebDebugLogger sfWebResponse		
#	type	message
1	 sfPatternRouting	Match route "localized_homepage" (/sf_culture/) for /pl/ with parameters array ('module' => 'job', 'action' => 'index', 'sf_culture' => 'pl',)
2	 sfFilterChain	Executing filter "sfRenderingFilter"
3	 sfFilterChain	Executing filter "sfExecutionFilter"
4	 jobActions	Call "jobActions->executeIndex()"
5	 Doctrine_Connection_Mysql	execo : SET NAMES 'UTF8' - 0
6	 Doctrine_Connection_Statement	execute : SELECT j.id AS j__id, j.created_at AS j__created_at, j.updated_at AS j__updated_at, j2.id AS j2__id, j2.created_at AS j2__created_at, j2.category_id AS j2__category_id, j2.type AS j2__type, j2.company AS j2__company, j2.logo AS j2__logo, j2.url AS j2__url, j2.position AS j2__position, j2.location AS j2__location, j2.description AS j2__description, j2.how_to_apply AS j2__how_to_apply, j2.token AS j2__token, j2.is_public AS j2__is_public, j2.is_activated AS j2__is_activated, j2.email AS j2__email, j2.expires_at AS j2__expires_at, j2.created_at AS j2__created_at, j2.updated_at AS j2__updated_at FROM jobeeet_category j LEFT JOIN jobeeet_job j2 ON j.id = j2.category_id WHERE (j2.expires_at > ? AND j2.is_activated = ?) - (2011-03-29 09:15:43, 1)
7	 sfPHPView	Render "sf_app_dir/modules/job/templates/indexSuccess.php"
8	 Doctrine_Connection_Statement	execute : SELECT j.id AS j__id, j.name AS j__name, j.lang AS j__lang, j.slug AS j__slug FROM jobeeet_category_translation j WHERE (j.id IN (??)) - (1)
9	 Doctrine_Connection_Statement	execute : SELECT j.id AS j__id, j.category_id AS j__category_id, j.type AS j__type, j.company AS j__company, j.logo AS j__logo, j.url AS j__url, j.position AS j__position, j.location AS j__location, j.description AS j__description, j.how_to_apply AS j__how_to_apply, j.token AS j__token, j.is_public AS j__is_public, j.is_activated AS j__is_activated, j.email AS j__email, j.expires_at AS j__expires_at, j.created_at AS j__created_at, j.updated_at AS j__updated_at FROM jobeeet_job j WHERE (j.category_id = ? AND j.expires_at > ? AND j.is_activated = ?) ORDER BY j.created_at DESC LIMIT 10 - (1, 2011-03-29 09:15:43, 1)
10	 sfPartialView	Render "sf_app_dir/modules/job/templates/_list.php"
11	 Doctrine_Connection_Statement	execute : SELECT COUNT(*) AS num_results FROM jobeeet_job j WHERE j.category_id = ? AND j.expires_at > ? AND j.is_activated = ? - (1, 2011-03-29 09:15:43, 1)
12	 Doctrine_Connection_Statement	execute : SELECT j.id AS j__id, j.name AS j__name, j.lang AS j__lang, j.slug AS j__slug FROM jobeeet_category_translation j WHERE (j.id IN (??)) - (2)
13	 Doctrine_Connection_Statement	execute : SELECT j.id AS j__id, j.category_id AS j__category_id, j.type AS j__type, j.company AS j__company, j.logo AS j__logo, j.url AS j__url, j.position AS j__position, j.location AS j__location, j.description AS j__description, j.how_to_apply AS j__how_to_apply, j.token AS j__token, j.is_public AS j__is_public, j.is_activated AS j__is_activated, j.email AS j__email, j.expires_at AS j__expires_at, j.created_at AS j__created_at, j.updated_at AS j__updated_at FROM jobeeet_job j WHERE (j.category_id = ? AND j.expires_at > ? AND j.is_activated = ?) ORDER BY j.created_at DESC LIMIT 10 - (2, 2011-03-29 09:15:43, 1)
14	 sfPartialView	Render "sf_app_dir/modules/job/templates/_list.php"
15	 Doctrine_Connection_Statement	execute : SELECT COUNT(*) AS num_results FROM jobeeet_job j WHERE j.category_id = ? AND j.expires_at > ? AND j.is_activated = ? - (2, 2011-03-29 09:15:43, 1)
16	 sfPHPView	Decorate content with "sf_app_dir/templates/layout.php"
17	 sfPHPView	Render "sf_app_dir/templates/layout.php"
18	 main	Get slot "title"
19	 Doctrine_Connection_Statement	execute : SELECT j.id AS j__id, j.category_id AS j__category_id, j.type AS j__type, j.company AS j__company, j.logo AS j__logo, j.url AS j__url, j.position AS j__position, j.location AS j__location, j.description AS j__description, j.how_to_apply AS j__how_to_apply, j.token AS j__token, j.is_public AS j__is_public, j.is_activated AS j__is_activated, j.email AS j__email, j.expires_at AS j__expires_at, j.created_at AS j__created_at, j.updated_at AS j__updated_at FROM jobeeet_job j WHERE (j.id IN (??)) - (25)
20	 main	Call "language->executeLanguage()"
21	 sfPartialView	Render "sf_app_dir/modules/language/templates/_language.php"
22	 sfWebResponse	Send status "HTTP/1.1 200 OK"
23	 sfWebResponse	Send header "Content-Type: text/html; charset=utf-8"

Testowanie kodu

```
user@debian:~/framework/symfony/doctrine$ symfony test:unit Jobeet
```

```
1..15
```

```
# Witaj Świecie
```

```
ok 1 - Wartości są równe
```

```
ok 2 - String i Integer są takie same
```

```
ok 3 - Integer nie są takie same
```

```
ok 4 - Ciąg jest zgodny z regułą
```

```
ok 5 - Ciąg nie jest zgodny z regułą
```

```
ok 6 - Równanie jest zgodne
```

```
ok 7 - Ciąg jest typu string
```

```
ok 8 - Obiekt należy do klasy
```

```
ok 9 - Obiekt ma dostęp do metody
```

```
ok 10 - Tablice są takie same
```

```
not ok 11 - Jest dostępny plik i można go załadować
```

```
# Failed test (./test/unit/JobeetTest.php at line 46)
```

```
# Tried to include 'index.php'
```

```
ok 12 - Test zakończony powodzeniem
```

```
ok 13 # SKIP Dolicza pusty test
```

```
ok 14 # TODO To nie wszystkie testy
```

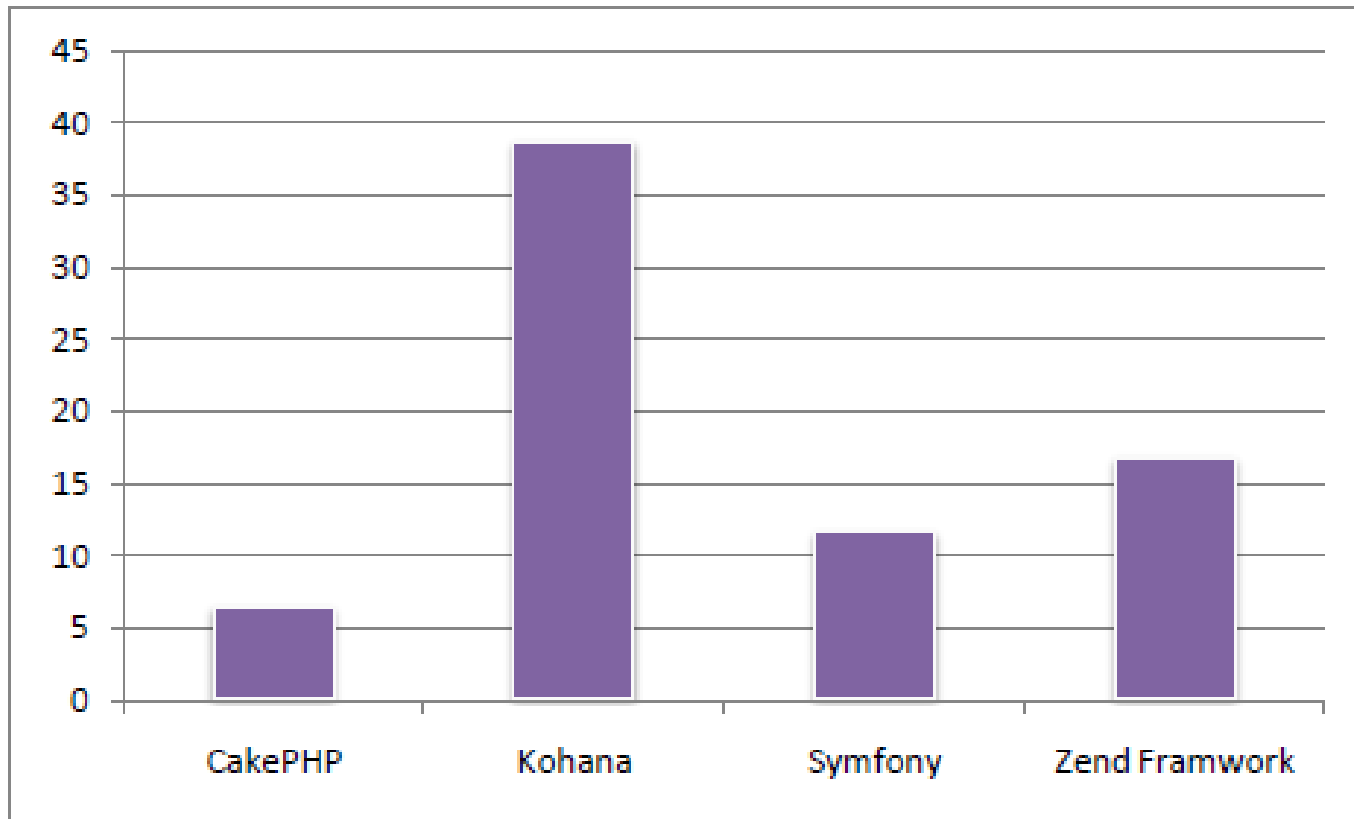
```
# Looks like you planned 15 tests but only ran 14.
```

```
# Looks like you failed 1 tests of 12.
```

```
user@debian:~/framework/symfony/doctrine$
```

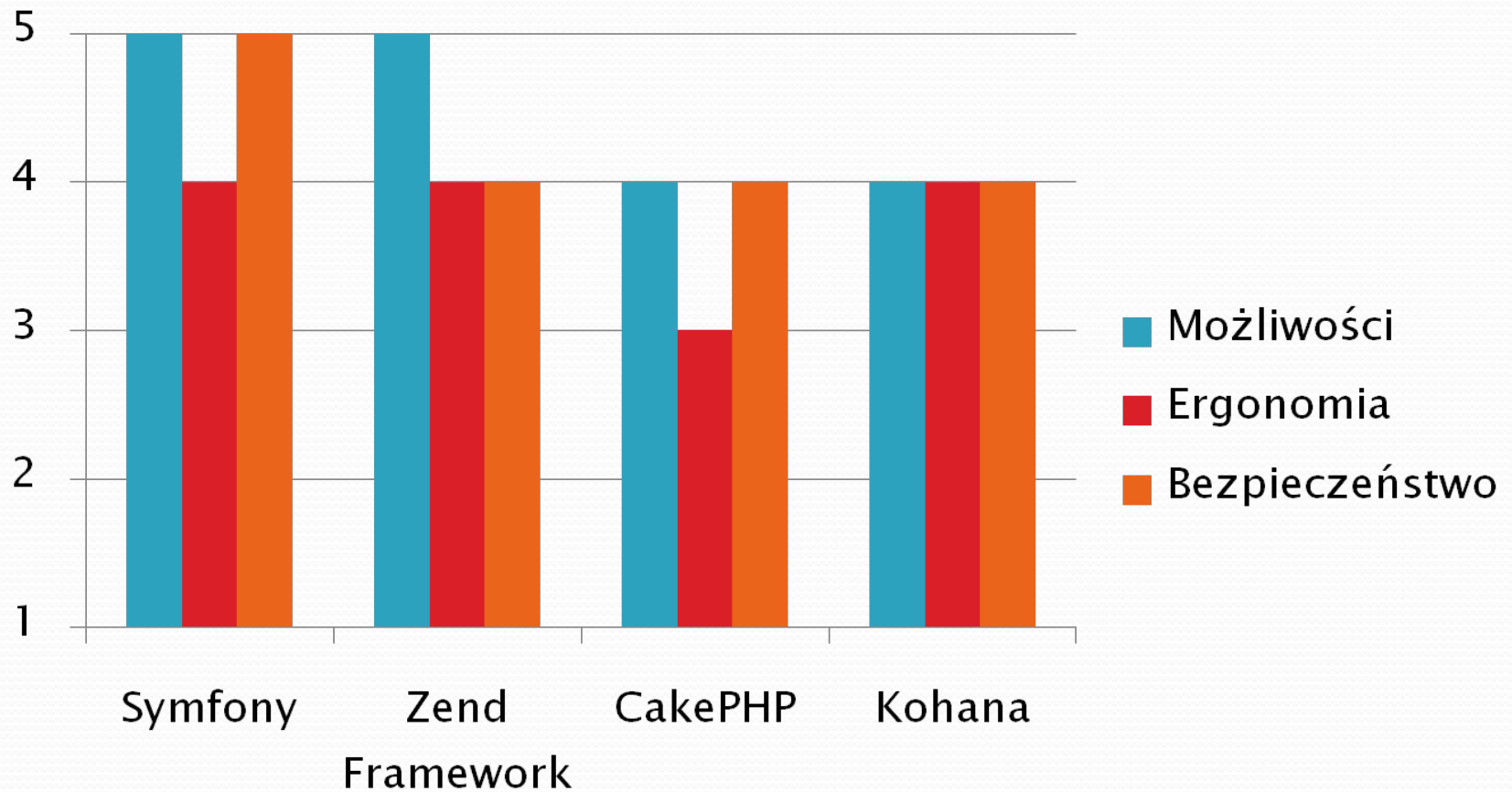
```
1 <?php
2 include dirname(__FILE__).'/../bootstrap/unit.php';
3 class myObject { public function myMethod() {}
4 }
5 function throw_an_exception() {
6     throw new Exception();
7 }
8 $t = new lime_test(15); //liczba testów
9 $t->diag('Witaj Świecie'); //komunikat
10 $t->ok(1 == '1', 'Wartości są równe'); //test 1
11 $t->is(1, '1', 'String i Integer są takie same'); //test 2
12 $t->isnt(0, 1, 'Integer nie są takie same'); //test 3
13 $t->like('abc1', '/abc\d+/', 'Ciąg jest zgodny z regułą'); //test 4
14 $t->unlike('def1', '/abc\d+/', 'Ciąg nie jest zgodny z regułą'); //test 5
15 $t->cmp_ok(1, '<', 2, 'Równanie jest zgodne'); //test 6
16 $t->isa_ok('abcd', 'string', 'Ciąg jest typu string'); //test 7
17 $t->isa_ok(new myObject(), 'myObject', 'Obiekt należy do klasy'); //test 8
18 $t->can_ok(new myObject(), 'myMethod', 'Obiekt ma dostęp do metody'); //test 9
19 $t->is_deeply(array, array('abc', 'def'), 'Tablice są takie same'); //test 10
20 $t->include_ok('index.php', 'Plik jest dostępny i można go załadować'); //test 11
21 try {
22     throw_an_exception();
23     $t->fail('Test zakończony niepowodzeniem'); //test 12
24 } catch (Exception $e) {
25     $t->pass('Test zakończony powodzeniem'); //test 12
26 }
27 if (!isset($abc)) {
28     $t->skip('Dolicza pusty test', 1); //test 13
29 } else {
30     $t->ok($abc, 'abc'); //test 13
31 }
32 $t->todo('To nie wszystkie testy'); //test 14
```

Porównanie wydajności aplikacji



Średnia liczba odpowiedzi serwera w ciągu jednej sekundy. Test wykonany w oparciu o narzędzie Httpperf.

Wyniki prowadzonej analizy



Podsumowanie

Niniejsza praca pokazała możliwości jakie niosą porównywane rozwiązania ich mocne i słabe strony, jak również szanse i zagrożenia związane z wykorzystaniem tych narzędzi w codziennej pracy. Przeprowadzona analiza potwierdziła przydatność rozwiązań framework'owy w tworzeniu aplikacji webowych, ich duży wpływ w szybkość, elastyczność, ergonomię i innowację w tworzeniu ciekawych aplikacji.

Tworzenie aplikacji w oparciu o framework'i, narzędzia ORM może stanowić duży problem dla nowych użytkowników, praktycznie każdy produkt jest dobrze opisany, posiada łatwy dostęp do materiałów, ale duża dynamika zmian niektórych produktów może prowadzić do ciągłego uczenia się ich obsługi i poświęcania na to zadanie cennego czasu, co w dzisiejszych czasach może stanowić duży problem, lecz czas ten poświęcony na naukę obsługi framework'a zwróci się w postaci szybko utworzonej aplikacji z gwarancją wysokiej jakości wykonanego kodu.