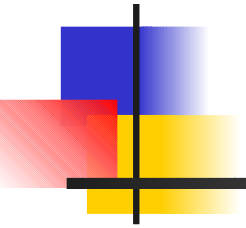
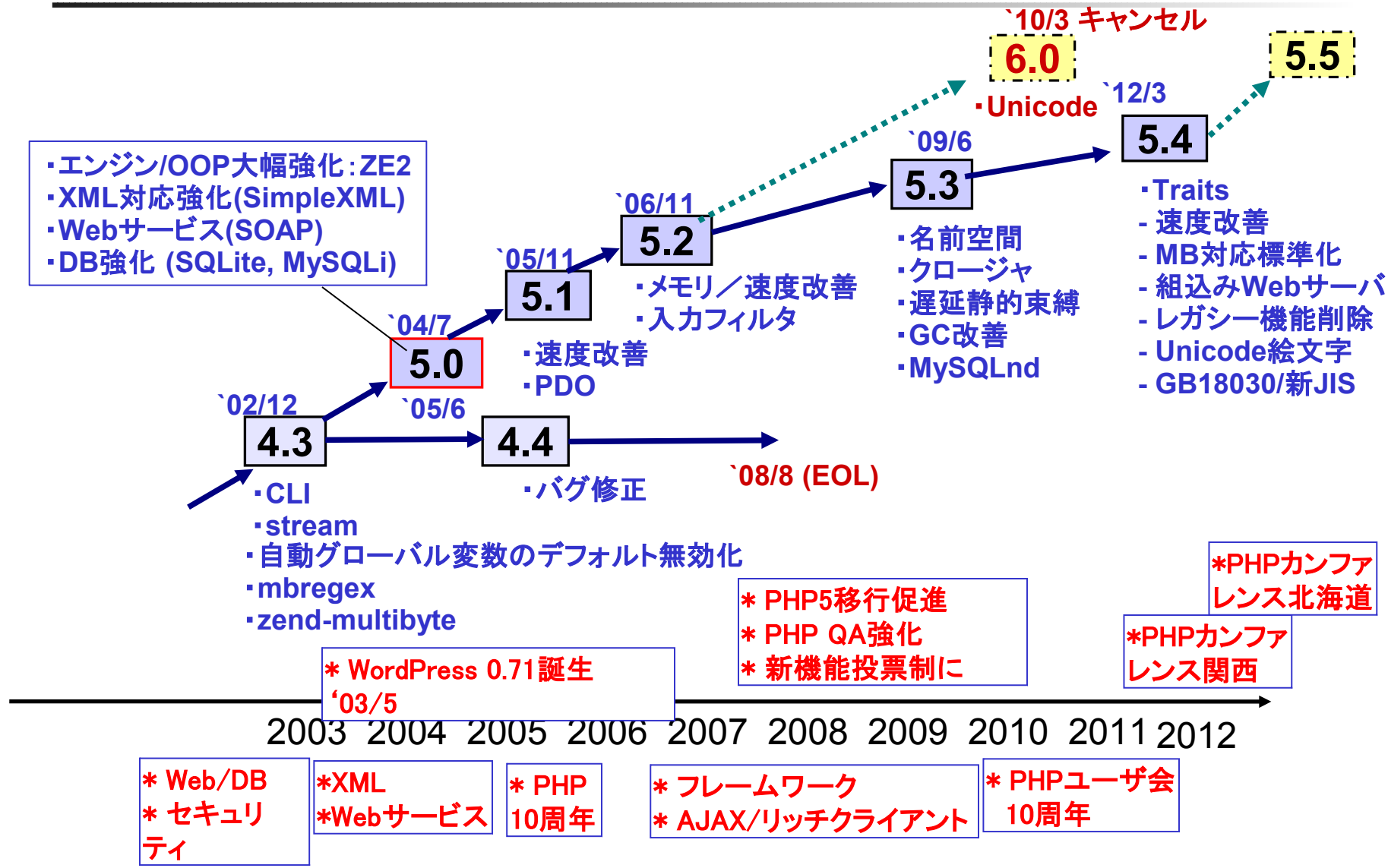


Language Update in 10 years: PHP

2012.8.4 at LL Decade

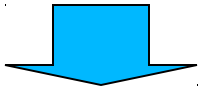
Rui Hirokawa (Japan PHP Users Group)

PHPの歩み

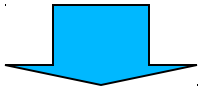

`10/3 キャンセル
6.0
・Unicode
5.5
`12/3
5.4
・Traits
- 速度改善
- MB対応標準化
- 組込みWebサーバ
- レガシー機能削除
- Unicode絵文字
- GB18030/新JIS
`09/6
5.3
・名前空間
・クロージャ
・遅延静的束縛
・GC改善
・MySQLnd
`06/11
5.2
・メモリ／速度改善
・入力フィルタ
`05/11
5.1
・速度改善
・PDO
`04/7
5.0
`05/6
`02/12
4.3
・CLI
・stream
・自動グローバル変数のデフォルト無効化
・mbregex
・zend-multibyte
4.4
・バグ修正
`08/8 (EOL)
・エンジン/OOP大幅強化：ZE2
・XML対応強化(SimpleXML)
・Webサービス(SOAP)
・DB強化 (SQLite, MySQLi)
＊ WordPress 0.71誕生
‘03/5
＊ PHP5移行促進
＊ PHP QA強化
＊ 新機能投票制に
*PHPカンファレンス関西
*PHPカンファレンス北海道
2003 2004 2005 2006 2007 2008 2009 2010 2011 2012
＊ Web/DB
＊ セキュリティ
*XML
*Webサービス
＊ PHP 10周年
＊ フレームワーク
＊ AJAX/リッチクライアント
＊ PHPユーザ会 10周年

PHP 6.0リリースキャンセル事件

- Unicode対応(ICUベース、2006年にPreview版)



- コア／エクステンション全面的に要修正
- ブランチ（PHP 5.3）に新機能が実装された（6.0は無視…）



Rasmusが実装キャンセルを宣言（2010/3）

PHP 5.5に向けて

- パスワードハッシュ化用API
- 定数配列／文字列参照
- 詳細は https://wiki.php.net/rfc まで
（あなたの提案も採用されるかも！）

```
echo array(1, 2, 3)[0]; //output 1
echo "foobar"[2]; //output o
echo "foobar"["foo"][0] // output f
echo [1,3,4][2]; //output 4
```

PHP らしさとは?

- Rasmus語録
  http://en.wikiquote.org/wiki/Rasmus_Lerdorf

PHP is about as exciting as your toothbrush. You use it every day, it does the job, it is a simple tool, so what? Who would want to read about toothbrushes?

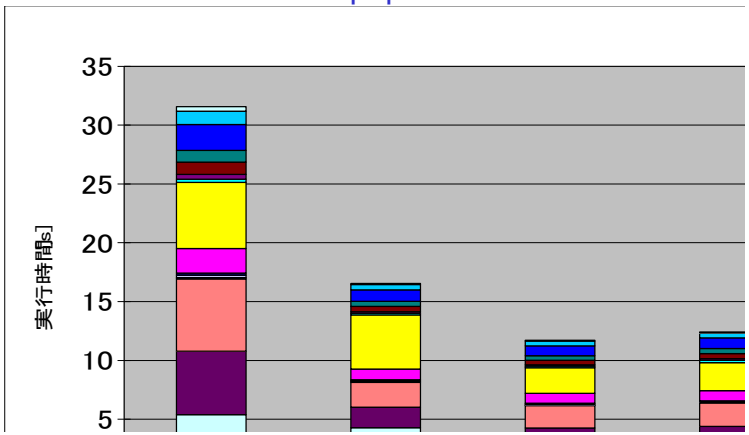
I've never thought of PHP as more than a simple tool to solve problems.

いつも身近で役に立つ自然な存在、それがPHP

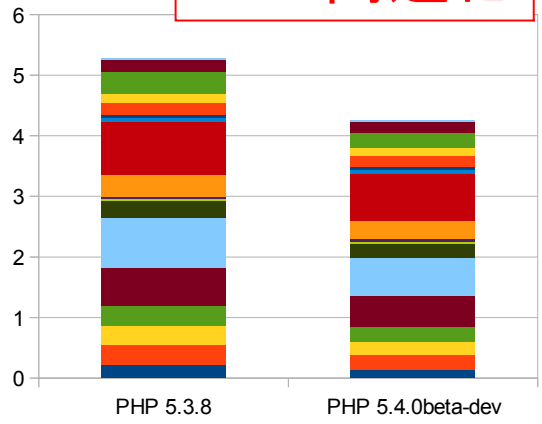

Thank you !

PHP 5.3/5.4における高速化

Zend/bench.php



19%高速化

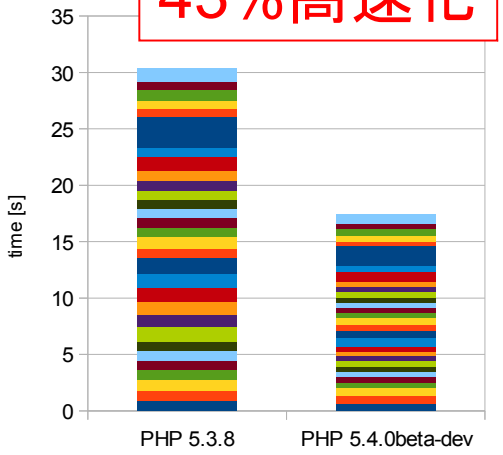


Zend/bench.php

- strcat(200000)
- ary2(50000)
- sieve(30)
- ary(50000)
- nestedloop(12)
- ackermann(7)
- matrix(20)
- mandel2
- heapsort(20000)
- mandel
- hash2(500)
- simpleudcall
- hash1(50000)
- simpleucall
- fibo(30)
- simplecall
- ary3(2000)
- simple

Ubuntu 11.04, AMD Athlon II X4 640

43%高速化



Zend/microbench.php

- $x = $str[0]
- $this->x = 0
- $x = $hash['V']
- $x = $this->x
- $x = $GLOBALS['V']
- Foo::f()
- $x = $_GET
- self::f()
- $x = TEST
- empty(Foo::$x)
- new Foo()
- isset(Foo::$x)
- $x = Foo::TEST
- Foo::$x = 0
- $this->f()
- $x = Foo::$x
- empty($this->x)
- empty(self::$x)
- isset($this->x)
- isset(self::$x)
- $this->x--
- self::$x = 0
- $this->x++
- $x = self::$x
- --$this->x
- int_func()
- ++$this->x
- undef_func()
- $this->x += 2
- func()

PHP 5.4 でコード再利用:Traits

- Trait:　単一継承の言語でコードを再利用する仕組み

「言語がサポートするコピー／ペースト」（多重継承よりシンプル）

- クラスや他のtraitsとメソッド名の衝突を解決する仕組み

```php
<?php
 trait A {
   public function show() {
          echo 'Hello,',$this->getName(),'!';  }
   abstract public function getName();
 }

class Base { /* 1 */ }

class Foo  extends Base {
        use A;
        public function getName() { echo 'PHP'; }
}
$obj = new Foo();
echo $obj->show(); // Hello,PHP!
```

```php
<?php
 trait A {
   public function show() { echo 'Hello,';} }

 trait B {
   public function show() {echo 'PHP!';} }

class Foo {
  use A, B {
     A::show insteadof B;  リネームでは
     B::show as showN; }  なくエイリアス
}

$obj = new Foo();
echo $obj->show() . $obj->showN();
```

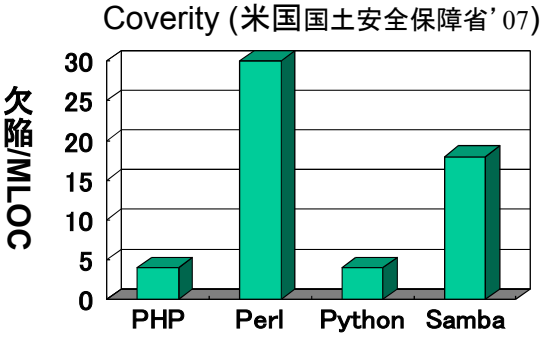
PHPとQA

- **PHPのコード品質は比較的高い**

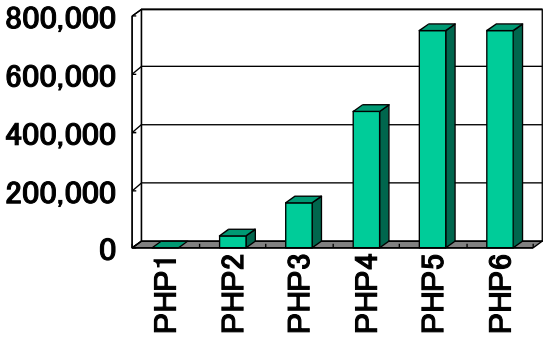
- **テストされていないコードには欠陥がある**

  カバレッジ率改善：約60%（PHP 5.2）

  → 約70% (PHP 5.3)



- PHP 5.3.7 (8/18公開)でcrypt関数（MD5)が動作しなかった　　strncat != strlcat



・RC5で静的解析ツールの警告を修正した際にエンバグ

・テストケースの失敗をノイズとして見過ごす

・修正版（PHP 5.3.8）を緊急公開(8/23)

・軽微な未修正の問題についてもテストケースがコミットされFAILする原因となっている。

→ 提案「XFAIL (experimental fail)を導入し、FAILの出現の判別を容易にする」