

n 年塩漬けされていたプロジェクトを 8.2移行するときにやったこと

Migrate a PHP project that has been neglected for several years to 8.2.



pixiv Inc.
USAMI Kenta

pixiv

お前誰よ



- うさみけんた (@tadsan) / Zonu.EXE / にゃんだーすわん
- ピクシブ株式会社 pixiv事業本部 Webエンジニアリングチーム PHPer
 - 2012年末から現職でPHPを書いているWebプログラマ
- Emacs PHP Modeを開発しています (2017年-)
- プログラミング言語にちょっとこだわりのある素人 (spcamp2010)

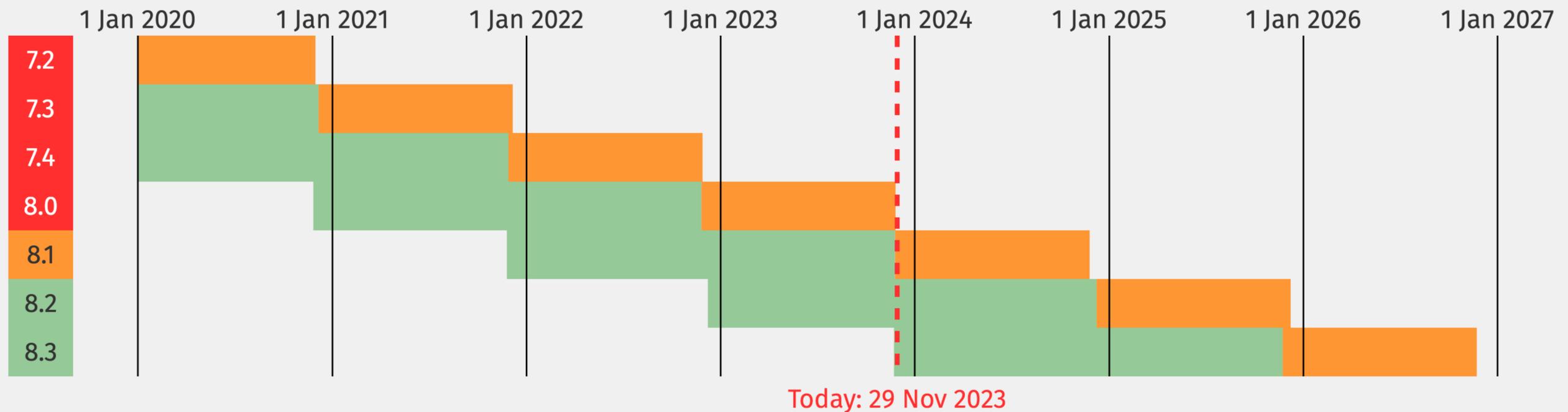
さて

PHP 8.3リリース
されましたね 🎉

Currently Supported Versions

Branch	Initial Release		Active Support Until		Security Support Until	
8.1	25 Nov 2021	<i>2 years ago</i>	25 Nov 2023	<i>4 days ago</i>	25 Nov 2024	<i>in 11 months</i>
8.2	8 Dec 2022	<i>11 months ago</i>	8 Dec 2024	<i>in 1 year</i>	8 Dec 2025	<i>in 2 years</i>
8.3	23 Nov 2023	<i>6 days ago</i>	23 Nov 2025	<i>in 1 year, 11 months</i>	23 Nov 2026	<i>in 2 years, 11 months</i>

Or, visualised as a calendar:



みなさまが愛した
PHP7.xは全てEOL

社内のプロダクトは
全て8.2移行済み

その前にPHP バージョンアップの 定石おさらい

開発を止めずにバージョン更新する

- アプリケーションコード内の非推奨記法をすべて置き換える
- 依存パッケージを現行と移行先の両方で動作するバージョンにアップデート
- ユニットテストを現行と移行先バージョンの両方で完動するように
- CI(ユニットテスト・静的解析)が両バージョンで動作するようにし、開発メンバーには互換性を維持して開発してもらおう
- 本番環境をどうにかして新しいバージョンに切り替える

というわけで無事に
PHP 8.2化 🎉

.....こいつ以外は

ContestCMS

ContestCMS

- pixivのコンテスト機能のデータを入稿するための仕組み
- 2013年に内製の独自フレームワークで書かれた
 - 入社1年目のtadsanが開発に参加、紆余曲折あって今まで運用担当
- 弊社のPHPコードの多くは2013年頃からpixiv.gitというモノレポで運用しているが、歴史的経緯により合流せずに存続している
- アプリケーション: 13,900行、テストコード3,430行の規模

塩漬けにしていたのは
ほかならぬ私です👼

10年前のプロジェクト
ここが辛い

年季の入った放置プロジェクトあるある

- E_NOTICEをハンドリングしてない
- 依存パッケージがめちゃくちゃ古い
- PHPStanで解析しようにもフレームワークが型のことをまるで配慮していないのでひたすら警告が出まくる
 - ``@return int|int[]`` のような型はついてないのといっしょ！
- そもそも既存バージョンでもPHPUnitが… 通ってないんだが…

嘆いても
仕方ないので直す

0. 通らないなりに PHPStanを入れる

PHPStanの導入

- いきなりレベルを高めにする必要はまったくない
- 現状の問題をphpstan-baseline.neonに記録してコミットする
- コードを弄ったり、依存パッケージのバージョンアップをしたときにその都度baselineを再生成して変な差分が出ていないかチェックする

1. Syntaxの非互換

文字列補間の非互換

- いままでPHPの文字列に `$var` という変数を埋め込むときにシェル風の `"${var}"` と `"{$var}"` が両方使えた
- PHP 8.2から `"${var}"` が非互換になったので `"{$var}"` に置き換える

2. インターフェイスの 非互換

PHP 8.1の非互換変更 (その1)

- インターフェイスに型宣言が追加された
 - `MySession::gc($max_lifetime) ↓`
`SessionHandlerInterface::gc(int $max_lifetime): int|false`
 - `int|false` なんてPHP 8.0にならないと書けないんだが???
 - メソッド定義に `#[ReturnTypeWillChange]` を付ければ黙ってくれる

PHP 8.1の非互換変更 (その2)

- PDO::query() の非互換

PDO::query

(PHP 5 >= 5.1.0, PHP 7, PHP 8, PECL pdo >= 0.2.0)

PDO::query – プレースホルダを指定せずに、SQL ステートメントを準備して実行する

説明

```
public PDO::query(string $query, ?int $fetchMode = null): PDOStatement|false
```

```
public PDO::query(string $query, ?int $fetchMode = PDO::FETCH_COLUMN, int $colno): PDOStatement|false
```

```
public PDO::query(  
    string $query,  
    ?int $fetchMode = PDO::FETCH_CLASS,  
    string $classname,  
    array $constructorArgs  
): PDOStatement|false
```

```
public PDO::query(string $query, ?int $fetchMode = PDO::FETCH_INT0, object $object): PDOStatement|false
```

PHP 8.1の非互換変更 (その2)

- PDO::query() の非互換
 - こんなのどうしろという感じだが、トレイト動的定義という荒技で回避可能
 - ……そもそもそんな変なことをしなくても、PDO::prepare()を使えばいい
- そもそもこの問題自体、PDOを自前拡張してるようなプロジェクト以外には刺さらない

ここで手間暇かけて
両バージョン対応を
諦める
(社内サービスなので)

3. PHPUnitの バージョンを上げる

PHPUnit 9.5

- PHPUnit 9.5 は \geq PHP 7.3 をサポートしているので、単に上げればヨシ
- PHP 7.3未満からPHP 8.xまで上げる場合はその方法が使えないので…
 - `composer require --dev phpunit/phpunit:^7.0|^8.0`

4. Smartyの アップデート

Smarty 3.x → 4.0に移行

- 社内ではSmarty 2が多く残っているが、今回のプロジェクトは3だった
 - 3.xと4.xは互換が高いのであまり気にせず上げていい
 - Smartyはmodifierという仕組みで `$var|func` と書くとプラグインが未定義だったときに `func($var)` に置き換えられたが、4.0から非互換に
 - 明示的にmodifier pluginとして定義してやればいい

5. その他パッケージミジの アップデート

依存パッケージのアップデートなど

- 依存パッケージに非互換の問題がある場合は手を入れなければいけない
 - [cweagans/composer-patches](#) プラグインで composer install時に書き換えられる
 - [symplify/vendor-patches](#) というツールを使うと composer-patches用のパッチを簡単に作れるのでおすすめ
- composer-patchesは本番運用向けに必要なので、うっかり `composer require --dev`しないように注意

5. E_WARNINGを 潰す

E_WARNING

- いままで未定義変数アクセスや未定義配列アクセスはE_NOTICEだったが、PHP 8からE_WARNINGに昇格した
- カバレッジが十分なプロジェクトならテストを通せば十分はずだが…
- レガシープロジェクトにそんなものはない
 - 地道に動かして動作チェックする…

エラーハンドラのカスタマイズ

- 未定義アクセスエラーは雨後の筍のように出てくる
- エラーハンドリングは set error handler() でカスタマイズできる
- E_WARNINGは無視しつつエラーログに流すようなコードを書けばいい
 - 流量などはいろいろな方法で調整しつつ、最終的には全量流す
- エラーログに簡単にアクセスできるようにする or 無駄なものをだらだらとエラーログに流し続けない(フィルタできるようにする)

まとめ

PHP7.xからの移行は
最後は地道なもぐら叩き

本番環境も活用しながら
完全は求めず
安全に移行しような