

# Drupalの おいしいところ

UMAMIを引き出すチーム開発

Tomotsugu Kaneko - @snize

# 金子 智嗣

Zerobase Incの自由なエンジニア

Drupalは7から

**Drupal Meetup Tokyo**の運営メンバの3人のうちの1人

本日はコミュニティ枠での参加



# Drupal Meetup Tokyo

ユーザ主体のコミュニティ(スポンサーなし)

明確な登壇者のないスタイル

知識の交換

毎回のトピックはTrelloで相談

毎月第1木曜日に新宿で開催

デモソースコードはGithubで公開

[drupal-tokyo.github.io](https://drupal-tokyo.github.io)

Twitter: #DrupalTokyo

本日の内容

# DrupalのUMAMIを引き出す チーム開発

キーワード

分離

# Drupalを導入する理由は？

# Drupalを導入する理由は？

コストの最小化と価値の最大化

少ない労力でたくさん喜ばれたい

Drupalの他は？

# 他のPHP製のソフトウェア

## CMS

Wordpress

Movable Type

concrete5

## Framework

Symfony

Laravel

CakePHP

# Drupalの特徴

他のPHP製のソフトウェア比較して

# 他のCMSと共通するところ

コンテンツマネジメント機能

GUIの管理画面

コンテンツの作成編集

画像などのファイル管理

ユーザ管理、認証

# 他のCMSと共通するところ

コードを書かなくても  
かなりのことが実現できる

# 他のFrameworkと共通する ところ

依存性の解決ツール

composer

専用CLIツール

drush, drupal console

設定のコード化

Configuration Management

# 他のFrameworkと共通する ところ

設定含むソースコード管理

開発タスクの自動化

# CMS

コードを  
書かなくても  
かなりのことが  
実現できる



Movable Type 6



concrete5



# Framework

設定含む  
ソースコード管理

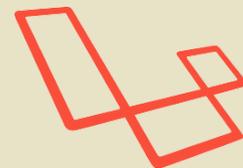
開発タスクの  
自動化



Symfony



CakePHP



真ん中の部分が特徴  
おいしいところがありそう



Movable Type 6



concrete5



Symfony



CakePHP



# 特徴を生かしたチーム開発

# Decoupled Drupal

ただし  
フロントエンドとバックエンド  
ではなく

何を分離するのか

1.コードとコンテンツ

何を分離するのか

## 2.役割

# 何を分離するのか

## 2.役割

開発チーム

開発者

デザイナー

編集チーム

編集者

何を分離するのか

## 2.役割

開発チーム

開発者

デザイナー

編集チーム

編集者

サイトビルダー

# サイトビルダーとは

コードは書かないけど、Drupal本体の機能やモジュールの機能を駆使してサイトを構築する人

## スキル

Webについての知識は必要

コード書けなくてもまあOK

Drupalについての知識は**必要**

→[Roles in Drupal development](#) | [Drupal Groups](#)

## List of roles

**Note regarding context:** we recognize on many projects a person might take on some or even all of these roles. We're using the redesign of Drupal.org as the context for the associated roles. It's a larger project, therefore with greater differentiation in the roles. Therefore, helping to guide the discussion.

**Note regarding tasks:** the descriptions in the roles mentioned here are *not exhaustive*. We're mainly focusing on the names of the roles. The descriptions are for clarification.

The next step is to undertake a more detailed *task analysis*, with the community. So no worries, we'll get to it. The first thing we need to do is ensure we're on the same page with roles.

### \* System Architect

This person sets up and maintains the system and infrastructure on which Drupal is deployed. They can manage the migration of data and content. They also install and maintain access to the site and version control. An example: <http://groups.drupal.org/node/66638>

### \* Developer

This person can code custom modules according to coding standards and best practices. They can test the quality and security of the code they write.  
An example: <http://groups.drupal.org/node/66643>

### \* Theming, Front-end developer

This person can interpret visual designs into code. They are experienced with the complementary skills of HTML CSS and JavaScript. They can use theme functions and may be able to create custom modules to implement hooks to create displays they need.

### \* Site Builder

This person installs and configures modules to create site features. They will be knowledgeable about theming and development, but will mostly use Drupal through the administrative interface.

### \* Content Editor & Manager

This person manages content and users on a Drupal site. They may not know many of the advanced functionality of the Drupal Administrative interface.

### \* Design, UX

This person may not know HTML, CSS or JavaScript. They specialize in visual design but with an understanding of the capabilities of Drupal which help them create designs which can be

サイトビルダーとは

## 注記

プロジェクトの規模や種類で役割の**分類や名前は変わる**  
またその**役割の一部や複数を担う**ことがある

# とはいえ、重要なこと

Drupalではサイトビルダーという役割が  
言語化され意識されている

Drupalエンジニアより

他のCMS経験者からコンバートしやすい

# とはいえ、重要なこと

Drupalではサイトビルダーという役割が  
言語化され意識されている

Drupalエンジニアより

他のCMS経験者からコンバートしやすい

少し脱線しました...

なぜ分離するのか

なぜ分離するのか

ライフサイクルが違う

なぜ分離するのか  
関心事が違う

つらい

# 違いを受け入れる

ライフサイクルと関心事

混ぜると...

謎のHTML

なぜか崩れてるCSS

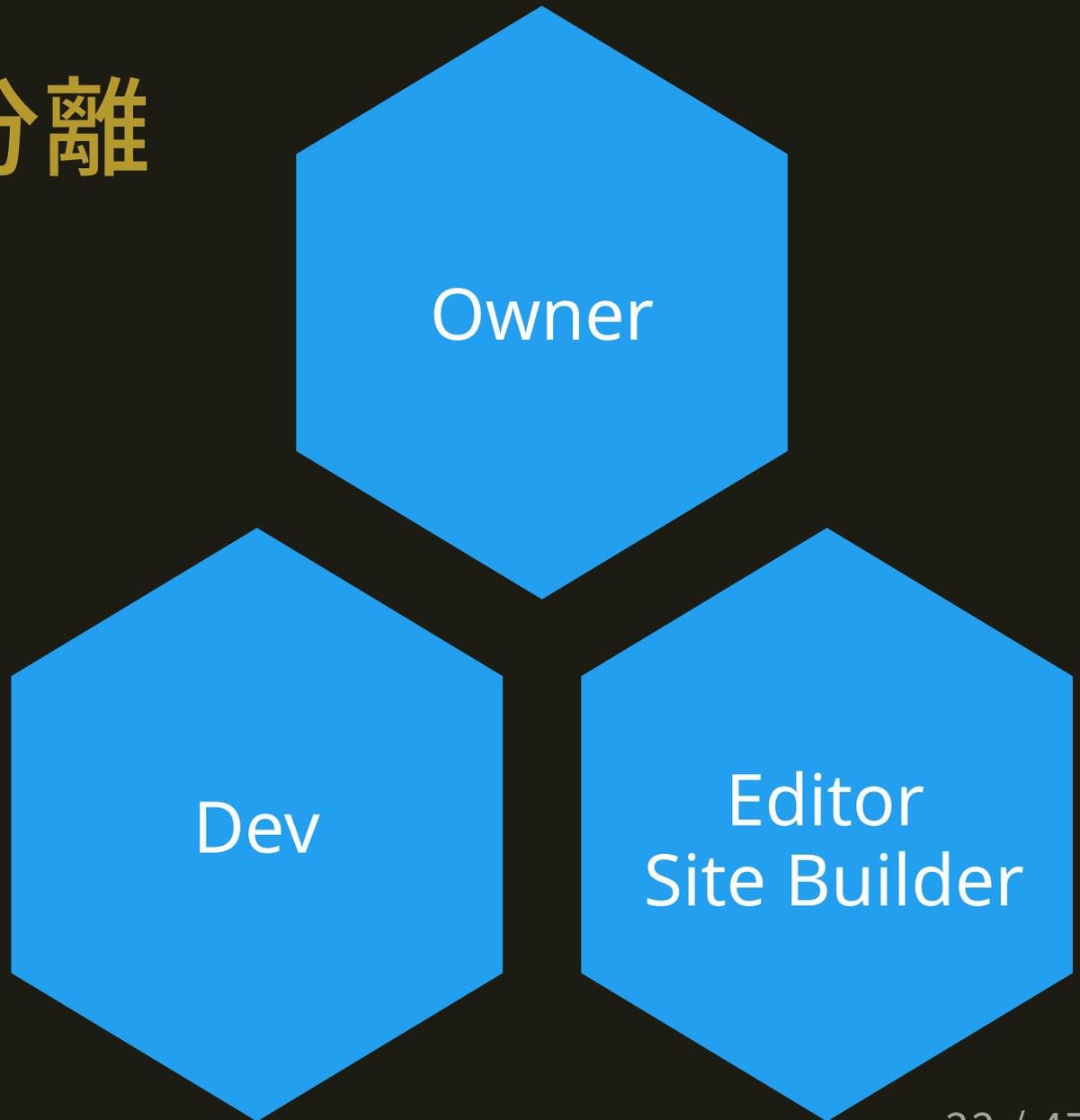
延々と公開されない超特急で書いページ

**荒れる運用**

違いを受け入れる、どうやって？

分離しよう

# 役割の分離



# 役割の分離

とあるメディアサイトの例  
週次で定例会

取材の打ち合わせ

コンバージョン率や広告の効果報告

新機能の相談

開発の進捗や問題点

Owner

Dev

Editor  
Site Builder

# 役割の分離

とあるメディアサイトの例  
週次で定例会

取材の打ち合わせ

コンバージョン率や広告の効果報告

新機能の相談

開発の進捗や問題点

分離したぶん**情報共有を密に**

Owner

Editor  
Site Builder

# オーナー

意思決定



# 開発チーム

普通のDrupalでの開発に加えて

編集チームへ**スタイルガイド**の提供

WYSIWYG

使用するモジュールに合わせてCSSを設計

サイトビルダーの**レクチャーとサポート**

コンテンツは**触らない**

Owner

Dev

Editor  
Site Builder

# 編集チーム

普通のDrupalでのコンテンツ編集に加えて

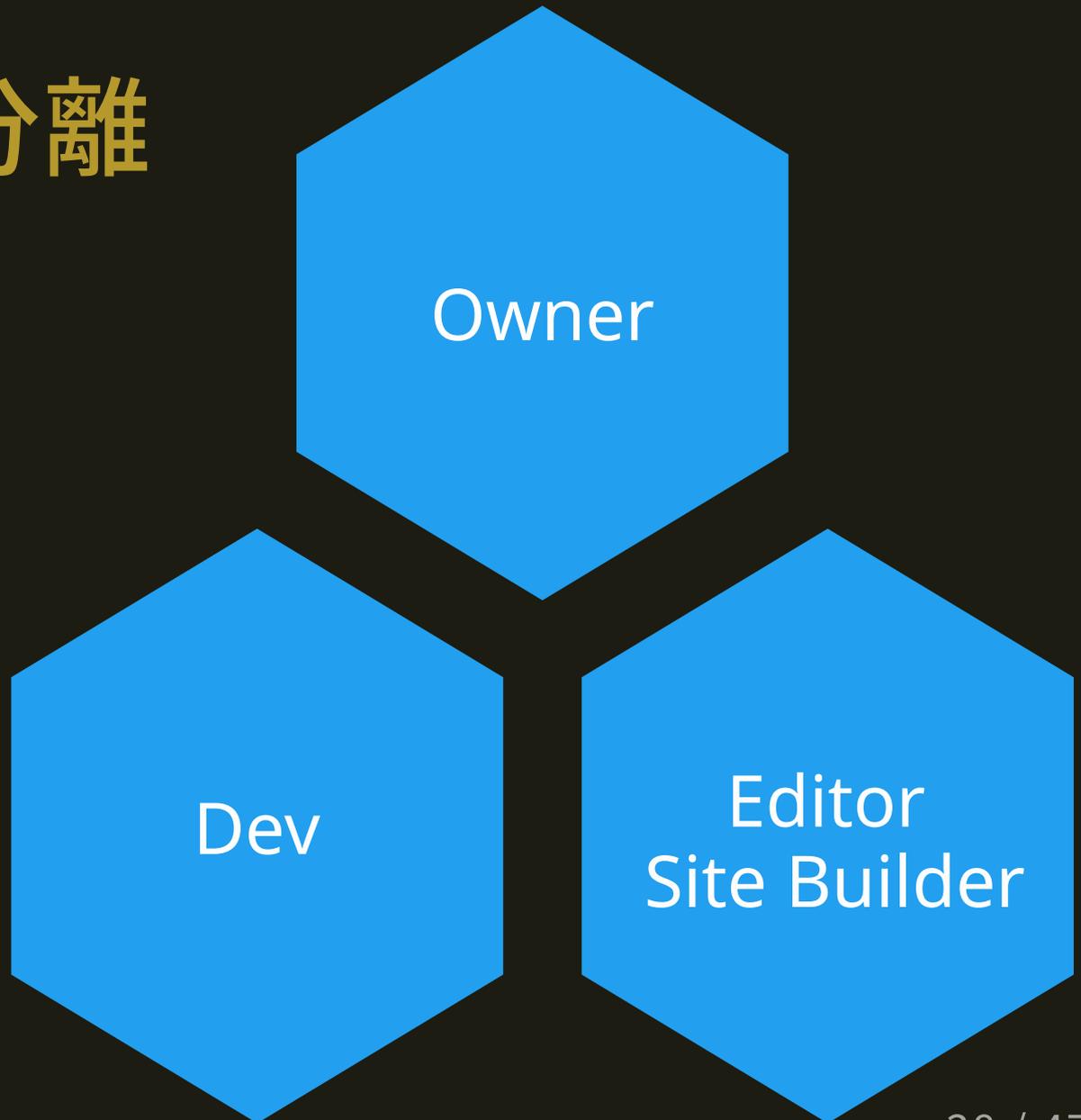
提供される機能のみでサイトを構築

機能が足りなければオーナーへ通知

Dev

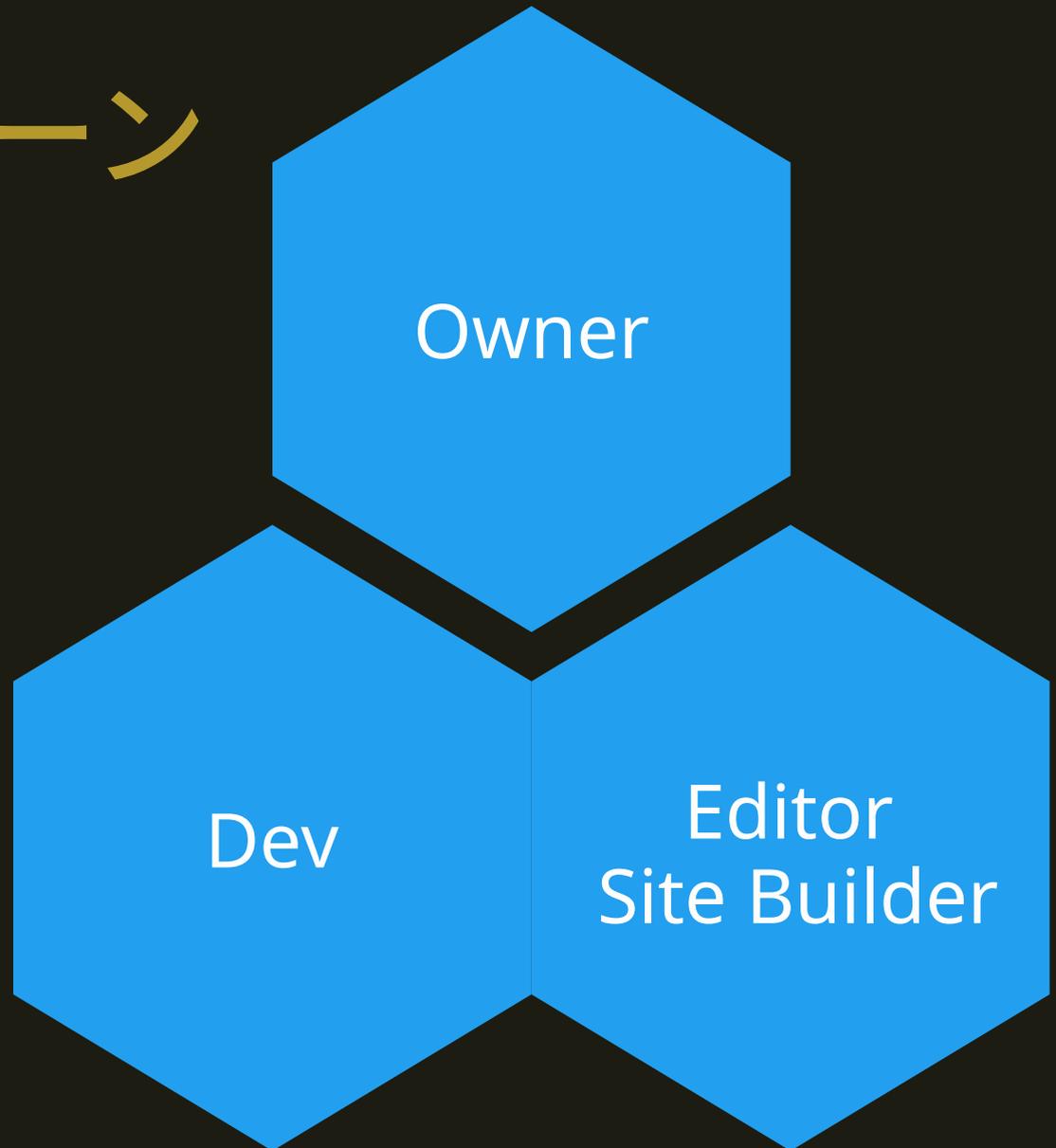
Editor  
Site Builder

# 役割の分離



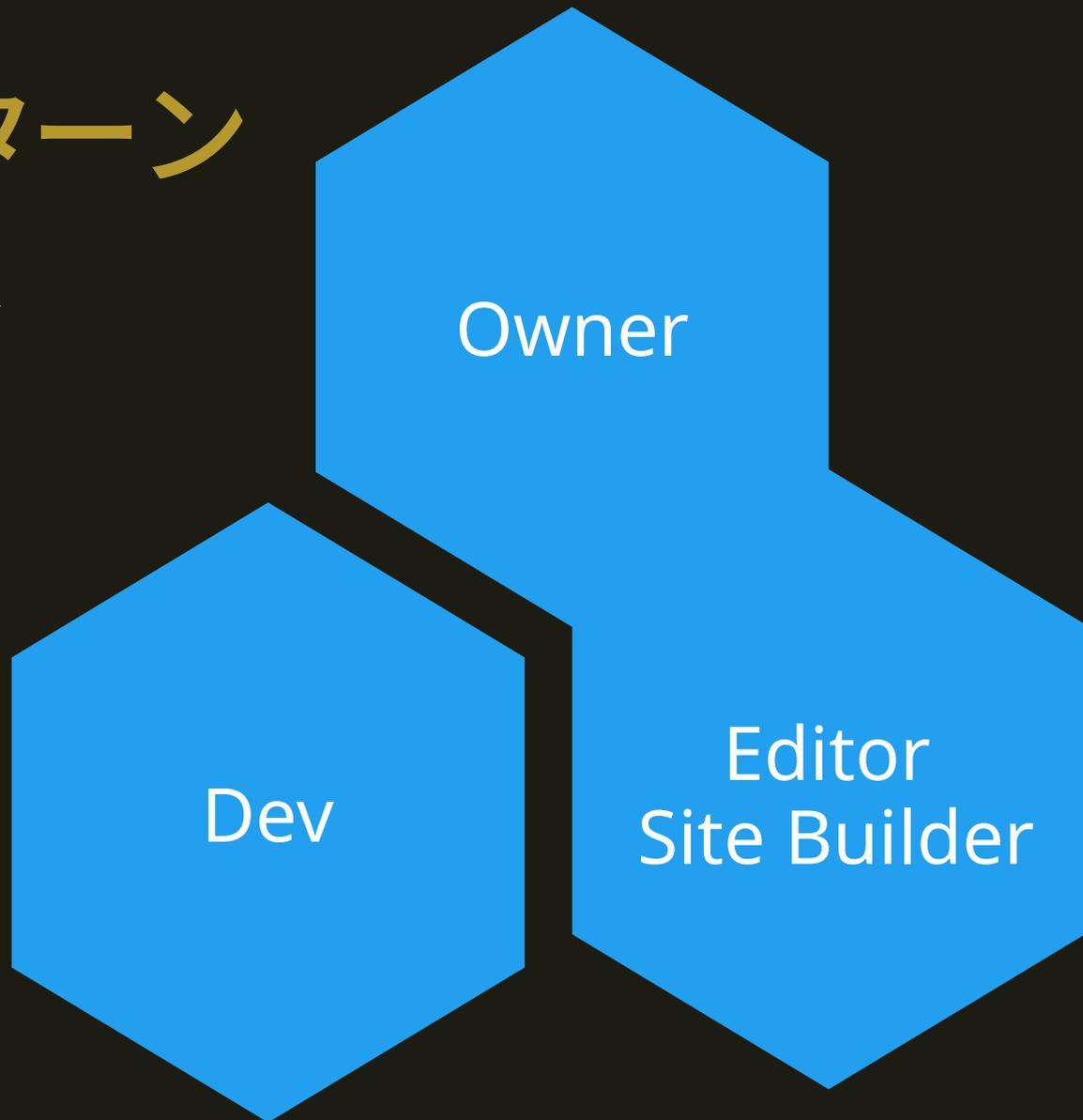
# 別のパターン

サイト制作全体



# 別のパターン

CMSの開発のみ



# Drupalを導入する理由は？



Symfony



Movable Type 6



concrete5



CakePHP



# どこから始めるか？

コードとコンテンツを分離から始める  
実際やってみると**落とし穴が結構ある**  
解決法も大抵ある

スタイルガイドの整備

役割についてドキュメントの作成と共有

分離したぶん**情報共有を密に**

**サイトビルダーの育成**

# Drupalのおいしいところ

今日話したこと

1. Drupalの特徴

2. 何を分離するのか

コードとコンテンツ / 役割

3. なぜ分離するのか

ライフサイクルが違う / 関心が違う

# Drupalのおいしいところ

今日話さなかったこと

インフラ / デプロイ

composer

環境ごとの設定管理 local,dev,stage,prod

CSSフレームワーク

スタイルガイド

ビジュアルリグレッションテスト

# Contact

具体的なテクニック等は

## Drupal Meetup Tokyo

懇親会

twitter : [@snize](#)

Slack : tomotsugu kaneko

[Drupal Japan 参加フォーム](#) | [Slack](#)

[kaneko@zerobase.jp](mailto:kaneko@zerobase.jp)

# Contact

具体的なテクニック等は

## Drupal Meetup Tokyo

懇親会

twitter : [@snize](#)

Slack : tomotsugu kaneko

[Drupal Japan 参加フォーム](#) | [Slack](#)

[kaneko@zerobase.jp](mailto:kaneko@zerobase.jp)

# 以上!

質問？

# 過去のスライド

[Configuration Management with Drupal](#)

[Why Drupal - Configuration Management](#)

[Secure your site with Drupal Paranoia](#)

[Talking with Drupal - psyshでの対話型デバッグ](#)