

aws **DEV DAY**

ONLINE | September 28, 2021

B-4

分散型チームによるAWSを活用した Webアプリケーション開発のナレッジ

小笹佑京 (Ozasa Yuki)

株式会社アンチパターン(Anti-Pattern Inc.)



自己紹介



小笹 佑京(Ozasa Yuki)

株式会社アンチパターン CEO兼VPoE

日本CTO協会 Contributor

Twitter: @yukiozasa



株式会社アンチパターンについて



ANTI-PATTERN INC.

Vision:

日本のソフトウェアエンジニアを憧れの職業へ

事業:

- ソフトウェアエンジニア採用サービス
- ソフトウェアエンジニア特化型コミュニティスペース
- 大学生向けプログラミング学習コミュニティ
- ソフトウェア開発支援

本セッションのゴール

- 開発者が幸せなことはビジネスに良い影響があることを知ってもらう。
- その上で、ナレッジをシェアすることによって、より良いソフトウェア開発の現場が増えるきっかけを作る。

本セッションのアジェンダ

- 開発者体験をより良くすることの重要性
- 分散型チームになった背景
- Webアプリケーション開発のナレッジ
- 終わりに

開発者体験をより良くすることの重要性

VUCAの時代

開発者体験をより良くすることの重要性

- Volatility : 変動性
- Uncertainty : 不確実性
- Complexity : 複雑性
- Ambiguity : 曖昧性

未来の予測が困難な状態

VUCAの時代における競合優位性の源とは

開発者体験をより良くすることの重要性

変化の激しい状況に対応できる能力が必要。
=ダイナミックケイパビリティを獲得する。

■ダイナミックケイパビリティの3要素



デジタルは増幅器として作用する。

経済産業省
「ものづくり白書2021年版」より。

DX (デジタルトランスフォーメーション)

開発者体験をより良くすることの重要性

「企業がビジネス環境の激しい変化に対応し、データとデジタル技術を活用して顧客や社会のニーズを基に、製品やサービス、ビジネスモデルを変革するとともに、業務そのものや、組織、プロセス、企業文化・風土を変革し、競争上の優位性を確立すること。」

※経済産業省

「デジタルトランスフォーメーションを推進するためのガイドライン（DX推進ガイドライン） Ver. 1.0」より

DXとダイナミックケイパビリティを研究した論文からの示唆

開発者体験をより良くすることの重要性

デジタルトランスフォーメーションとは、
新しいデジタル技術を組織の日常生活で活用する継続的なプロセス。

その根幹たるアジリティとは、

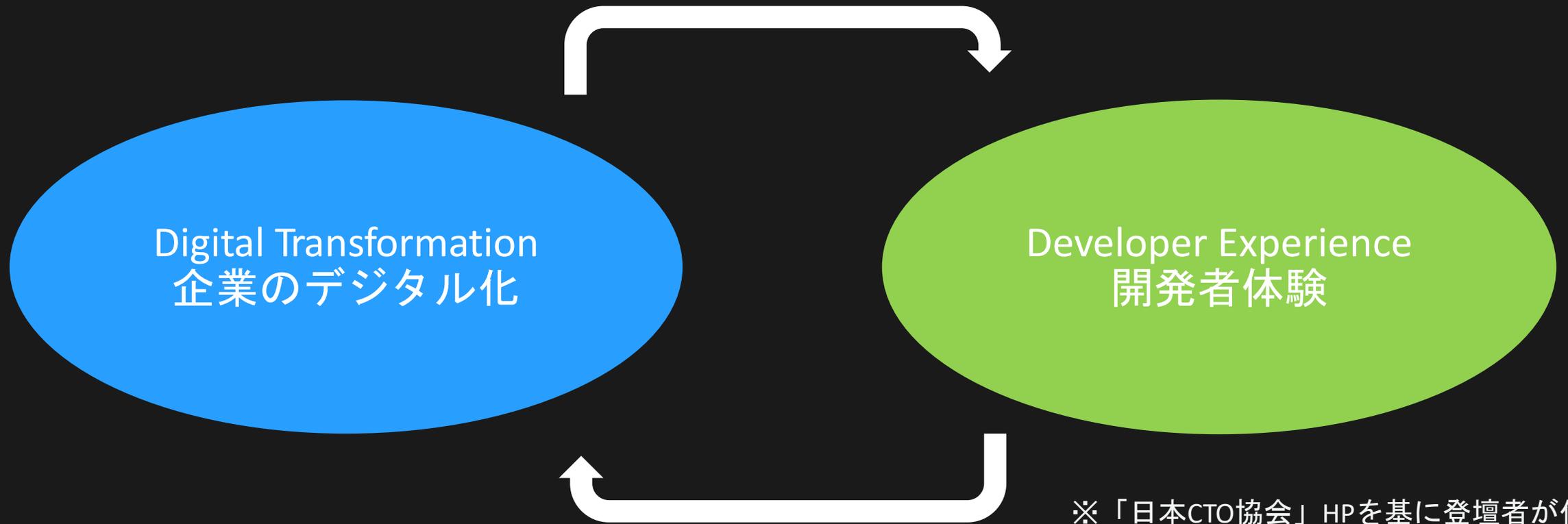
- (1) ビジネスモデル
- (2) 協調的アプローチ
- (3) 文化

を戦略的に刷新するための中核的なメカニズムである。

※エディンバラ・ネピア大学Karl Warner氏とのITコンサルタントのMaximilian Wäger氏の研究
「Building dynamic capabilities for digital transformation: An ongoing process of strategic renewal」より

2つのDXは両輪で繋がっている

開発者体験をより良くすることの重要性



より良い開発者体験を生む場作りが
ダイナミックケイパビリティの獲得に繋がる。

分散型チームになった背景 (株式会社アンチパターンのケース)

生存戦略としての分散型チーム

分散型チームなった背景

地理的制約

- 新型感染症の流行によりオフィスに集まらない。

スタートアップの制約

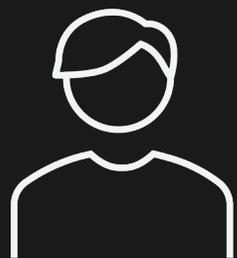
- 熾烈なエンジニア採用競争。
- 資金調達していないこともあり、知名度が低い。

優秀な副業メンバー(フリーランス含む)を
場所に依らず、採用する戦略へ

Webアプリケーション開発のナレッジ

現状の開発チーム構成

WEBアプリケーション開発のナレッジ



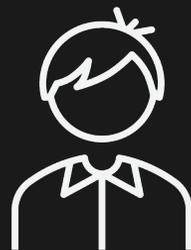
Product
Engineering
Owner

1名



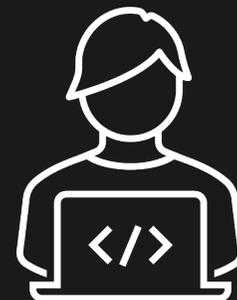
Designer

副業者：1名



Agile Coach

1名



Developers

社員：4名

副業者：4名

フリーランス：2名



対象Webサービスの概要

WEBアプリケーション開発のナレッジ

AWSエンジニアのための副業・転職スカウトサービス

engineed★

「優秀なエンジニアにより良い仕事と報酬を」

- AWS認定保持者限定で登録可能
- 実技試験を受けることが可能で、
スキルに応じたスカウトを受けられる

技術スタック

WEBアプリケーション開発のナレッジ

環境 : Docker

フロント : TypeScript / Vue.js / Nuxt.js / Vuetify

API : Go / chi (Web Framework)

API定義 : OpenAPI

DB : PostgreSQL

ソース/課題管理 : GitHub

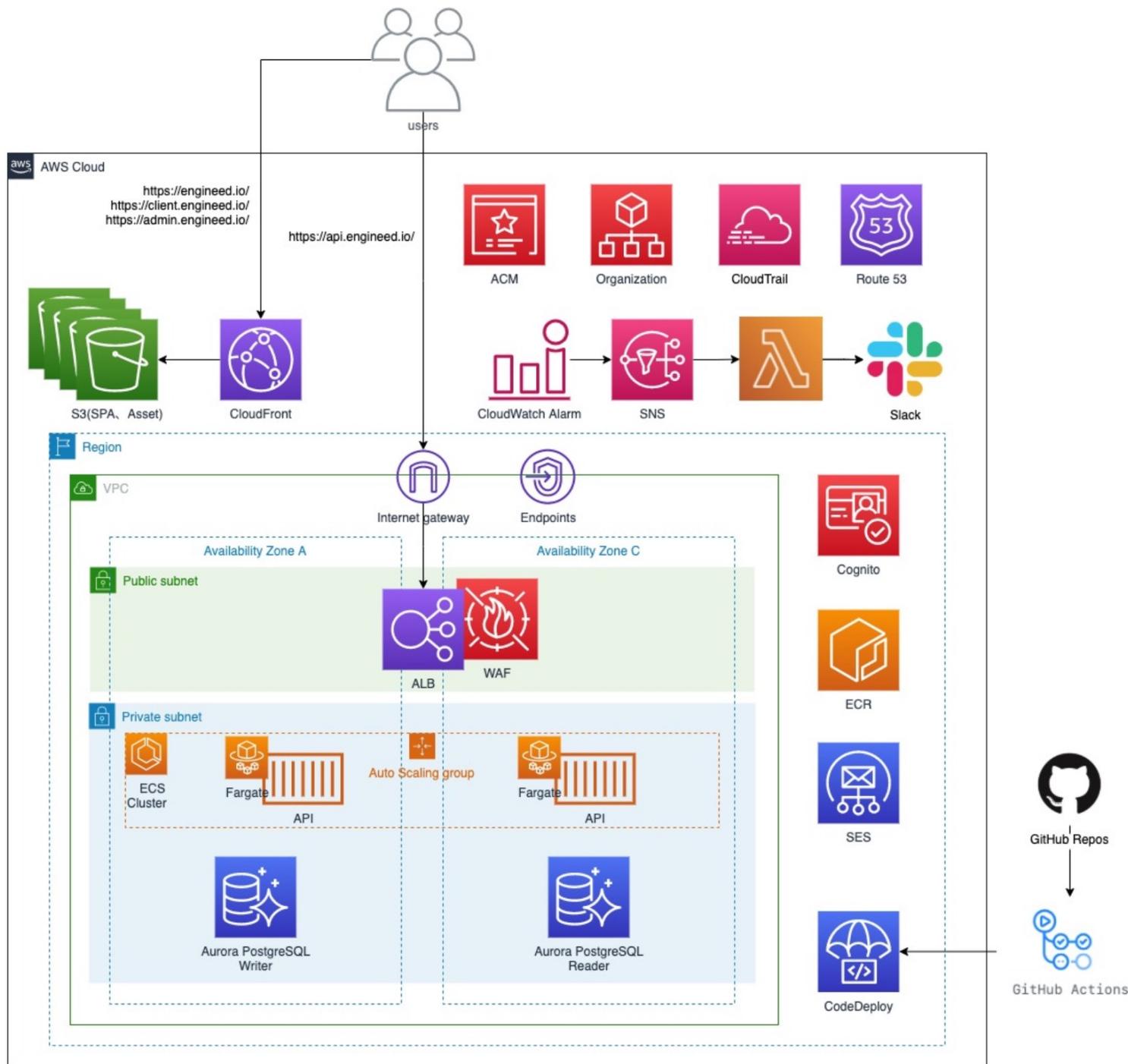
インフラ : AWS

デザインツール: Figma

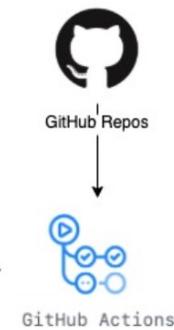
アーキテクチャ

WEBアプリケーション開発のナレッジ

あくまでシンプルに、素早く作ることを意識して設計。



- SPA+APIのシンプルな構成。
- 認証には Amazon Cognito を利用。
- JWT を用いて API にアクセス。



ナレッジを整理する

WEBアプリケーション開発のナレッジ

ツール・技術

- クラウド(AWS)を活用する
- CI/CDを整える

プロセス

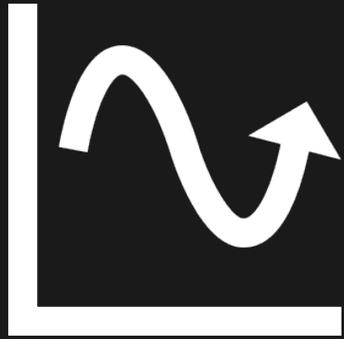
- OpenAPIを活用した型安全な開発
- トランクベースなブランチ&マージ戦略
- UIライブラリ駆動デザイン

チーム・文化

- カンバン
- アジャイル

クラウド(AWS)を活用する

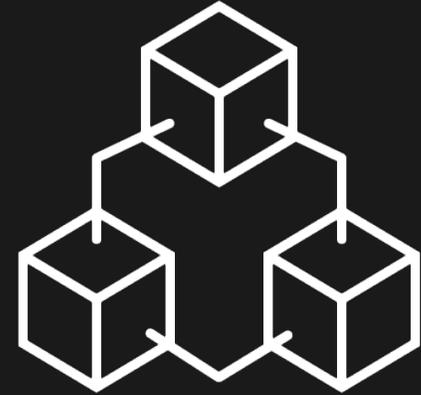
ツール・技術



スケーラビリティ



デプロイの
スピード



豊富な
マネージドサービス

=>積極的に巨人の肩に乗ることが重要。

クラウド(AWS)を活用する具体例

ツール・技術

認証基盤としてAmazon Cognitoを採用



採用の背景

- ・ 使い勝手の良さ
セキュリティ的にも安心できる。
加えて、機能が豊富なため素早くアプリケーションを実装できる。
- ・ 十分なリソースクォータ
ユーザープールあたりのユーザーの最大数が4000万。
対して、日本のIT技術者は約110万人(※)なので十分足りる。
- ・ サービス特性との整合
認証機能の1秒あたりのリクエスト数に制限があるものの、
ピークタイムが分散される想定のため、十分に活用できると判断。

※ヒューマンリソシア株式会社

第1回：世界各国のIT技術者数～アジア・オセアニア編～

https://corporate.resocia.jp/ja/info/investigation/case/global_report01

CI/CDを整える

ツール・技術

- リリースによる作業ミス(=開発者の心理的な負荷)、リリースにかかる工数などを減らすためにも、早いうちから投資しておくべき。早ければ早いだけ利益を享受できる。
- GitHub Actions + AWS CodeDeploy(ECSタスクの兼ね合い)を利用中。AWS CodePipelineに寄せようかと計画。

OpenAPIを活用した型安全な開発

プロセス

■開発のフロー

1. 要件をもとにAPIを定義。
2. OpenAPI generatorを使用し、型の定義がされたコードを自動生成。
3. StoplightのPrismを使用し、型定義に合ったランダムな値を出力するmockコンテナを生成。



フロント側の実装

API側の実装

=>型安全かつパラレルに開発を行うことが可能に。

トランクベースなブランチ&マージ戦略

プロセス

ある程度の規模がある開発だとしても、developブランチにどんどんマージしていく方式。リリース直前にmainにマージ。

フロント

- 新規画面の場合
 - .nuxtignoreを使用し、ビルドの対象外に。
- 一部コンポーネントの場合
 - v-ifを使用し表示対象外に。

API

- 新規APIの場合
 - 空文字を返すようにしておく。
- 既存APIの場合
 - /v2/などpathでバージョンングの設定をする。

トランクベースなブランチ&マージ戦略

プロセス

テストコードを書くことによって、
ソフトウェア開発のアジリティを高く保つ。

フロント

- ユニットテスト
 - 書けていない部分があるので徐々に増やしている。
- スナップショットテスト
 - UIの差分を確認可能。

API

- ユニットテスト
- インテグレーションテスト

UIライブラリ駆動デザイン

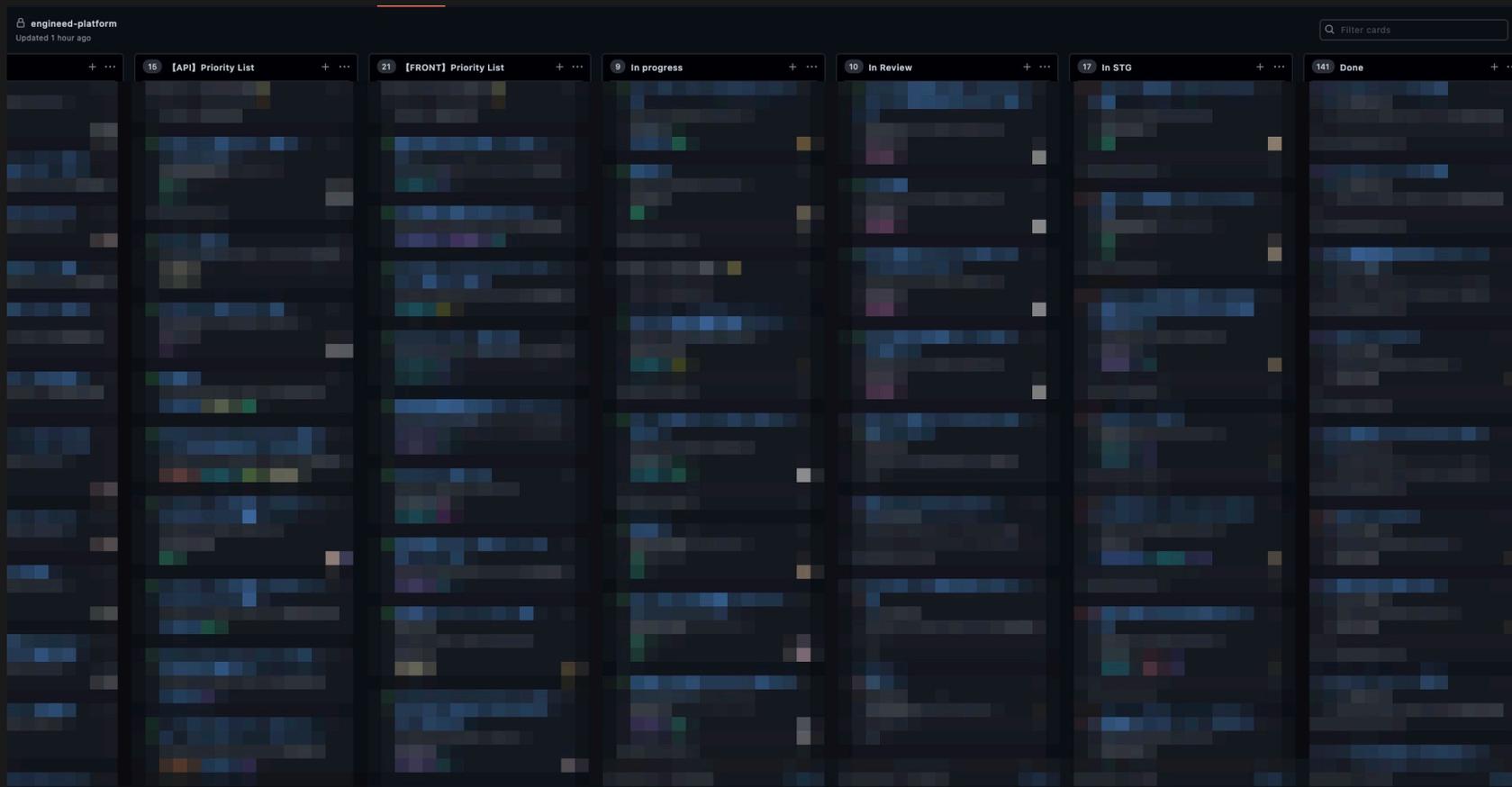
プロセス

- 機能のデザイン時には、まずVuetifyにその要求を満たすコンポーネントが存在するかを確認する
- 確認をした上でFigma上にデザインを書き起こす。
Vuetifyのコンポーネントを使う場合にはこの時点で指定しておく。

大抵の要求は満たされる。
満たされない=複雑なデザインが必要な場合は、
要求自体が複雑すぎる可能性を考慮する。

カンバン

チーム・文化



- GitHubを活用。

カンバン

チーム・文化

Priority
list

In progress

In review

In STG

Done

- 優先順に並べられている。
- フロントとAPIで2レーンあり、上からとっていけば良い状態に保たれている。

カンバン

チーム・文化

Priority
list

In progress

In review

In STG

Done

- 開発者が作業しているレーン。

カンバン

チーム・文化

Priority
list

In progress

In review

In STG

Done

- 開発者がPull Requestを出すと自動で移動。
- Pull Requestには確認事項がチェックリスト形式でテンプレート保存されており、各開発者は確認の上レビューを依頼。

カンバン

チーム・文化

Priority
list

In progress

In review

In STG

Done

- Pull Requestがマージされると自動で移動。
- その際ステージング環境に自動でデプロイされる。
- 結合テスト及び受け入れテストを実施できるISSUEが載っている状態。

カンバン

チーム・文化

Priority
list

In progress

In review

In STG

Done

- ステージングでのテストが完了後、リリースされたISSUEはこちらに移動。
- リリース頻度は1週間に1回程度。長くても2週間。

カンバン

チーム・文化

Priority
list

In progress

In review

In STG

Done



開発者が自分で管理する必要があるのはこの移動だけ。
あとは全てシステムで自動化ないしはOwnerが管理する。
Priority listは常に手入れがされている状態に保たれている。

カンバン / ISSUE

チーム・文化



ISSUE

複数の情報を管理。

Size:

TシャツサイズのS / M / L。

Assign:

誰が担当者か。

初期はanyoneラベルがついており、
誰がとってもいい。

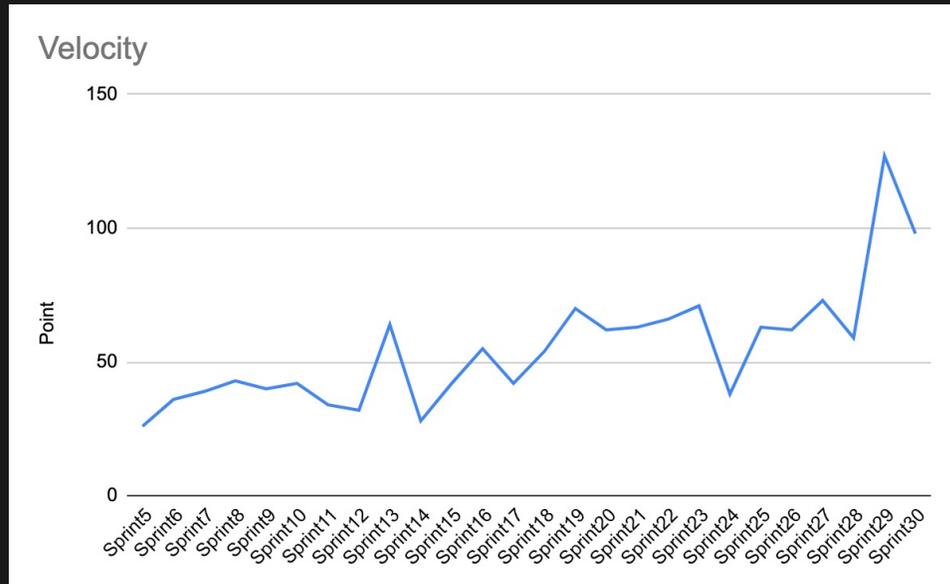
Milestone:

どのSprintか。

アジャイル

チーム・文化

- 毎週月曜日の夜に45分だけ同期活動。
- 事業の状況をシェア。
- その週に完了した(In STGに入った)ISSUEを計測。
- In STGに溜まっているチケットの数も計測し、溜まり過ぎを検知。

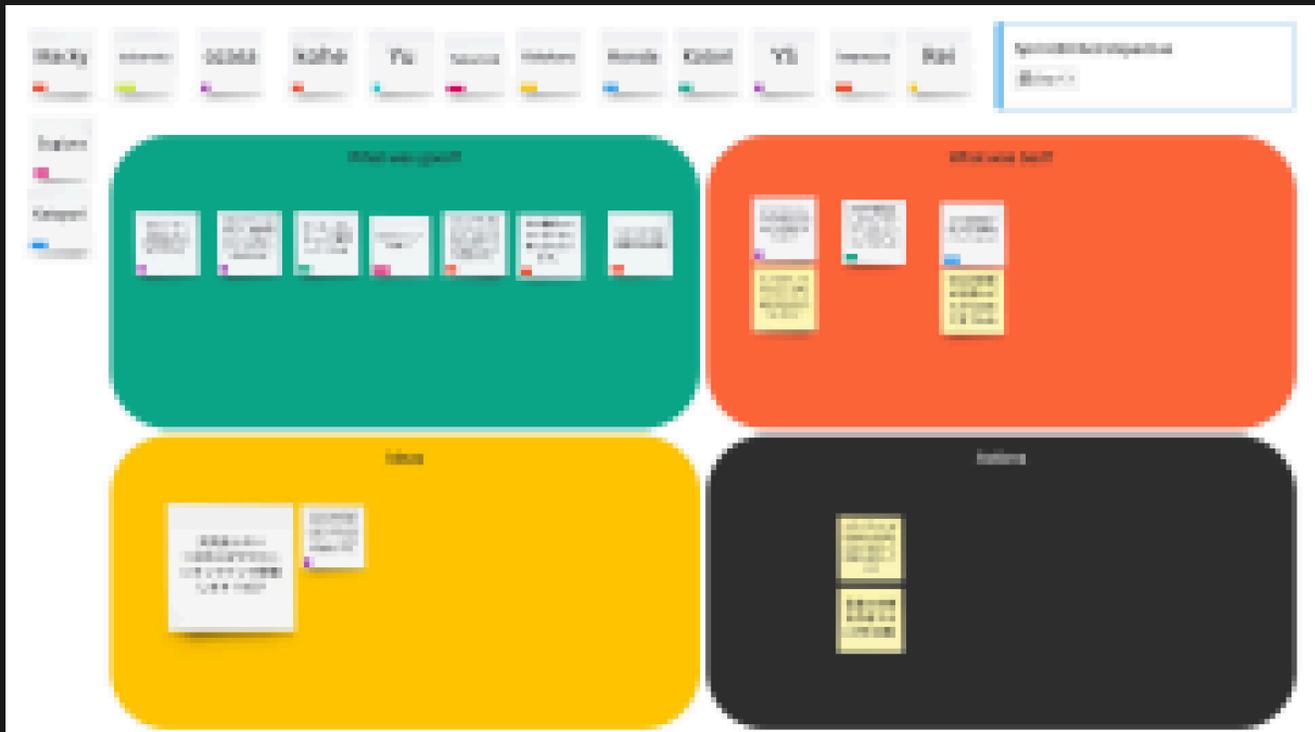


計測のため、
Sサイズは1ポイント、
Mサイズは4ポイント、
Lサイズは8ポイント、
とし、ベロシティを計測。

アジャイル

チーム・文化

- ベロシティとIN STGの数を見つつ、ふりかえりをMiro(オンラインホワイトボードツール)で実施。



- Good
- Bad
- Ideas
- Actions

アジャイル

チーム・文化

パートタイムメンバーの多い現在のチームでやめていること。

- 稼働時間を管理すること。
- スプリントゴールを設定すること。

エンジニアをインスタンスに見立てたとき、
副業者はスポットインスタンスのようなもの。

チームがスケールした時にも耐えられるシステム/仕組みに。

ナレッジを整理する

WEBアプリケーション開発のナレッジ

ツール・技術

- クラウドを活用する
- CI/CDを整える

プロセス

- OpenAPIを活用した型安全な開発
- トランクベースなブランチ&マージ戦略
- UIライブラリ駆動デザイン

チーム・文化

- カンバン
- アジャイル

より良い
Developer
Experience
開発者
体験を

成果

リリースから半年間での変化

成果

チームの人数
(=投資)

アウトプット量

アウトカム

3月

約2倍

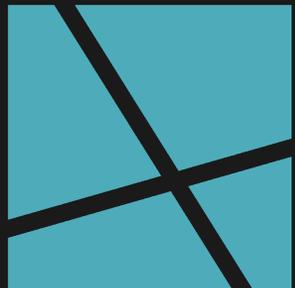
約8倍

0から1

9月

最後に

幸せなソフトウェア開発の現場を作ることが、
事業発展に繋がることを証明していきたいと思いますので、
応援いただけると嬉しいです。



ANTI-PATTERN INC.

Thank you!





Please complete the
session survey