Retty

組織に良い開発文化を植え付ける
「Software Engineering Coach」という役割

2020/07/25

Retty株式会社
常松祐一

自己紹介

Retty

- 常松祐一（つねまつ ゆういち）
  - Engineering Manager
  - Software Engineering Coach
  - Agile Development
- SNSアカウント
  - tunepolo :
  - tune :
- 顧客にとって価値のあるプロダクトを、チーム一丸となって協力し、短期間にリリースする開発体制のあり方を模索しています。

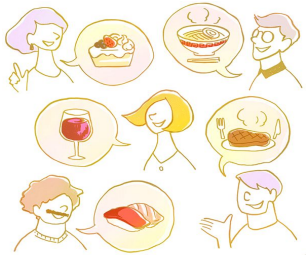
Retty

自分にとってBESTなお店が見つかる

日本最大級の"実名型"グルメサービス

批評ではなくオススメの口コミ

レビューよりもレコメンド。
Rettyは他人におすすめしたい
美味しいお店を投稿するサービス！



自分と好みが近い人から探せる

食の好みは人それぞれ。
自分と嗜好が合う人をフォローして、
BESTなお店を見つけられるSNS型！

顔が見えて信頼できる実名制

実名制の口コミだからこそ
「信頼できる」「ポジティブ」な
情報が集まっています！

今日の話

Retty

Extend Your Engineering Life!

良いソフトウェア開発を続けていくため、
自分が学んできたことを共有したい。

Photo by Razvan Chisu on Unsplash

自身の原体験 – 30歳前後で経験したデスマーチ

Retty

- プロジェクトマネージャーとして参画、1年半ほど苦しんだ。
- 当時は自分の技術に自信を持っており、技術で問題を打破できると信じていた。
- 青かった・・・

Photo by Cindie Hansen on Unsplash

今の私の肩書、社内での役割

Retty

- Engineering Manager
- Backend Software Engineer
- 社内Agile Coach

良いソフトウェア開発を
続けていきたい

良いソフトウェア開発

良いソフトウェア開発とは?

Retty

- 多くの人に使われる
- ユーザーにとって価値がある
- 社会的な意義がある
- 自分たちが作っていて楽しい
- 十分な報酬がある

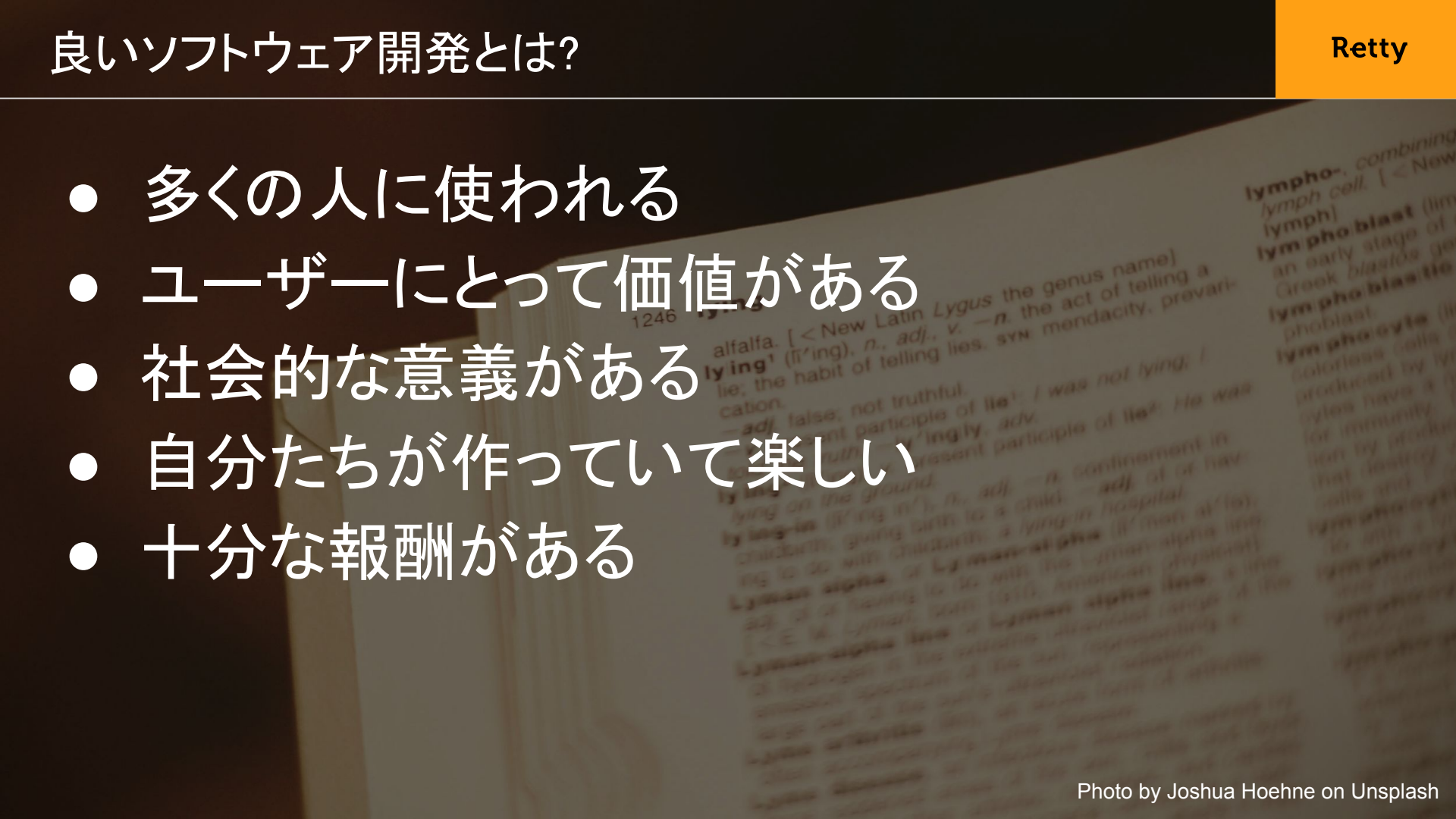
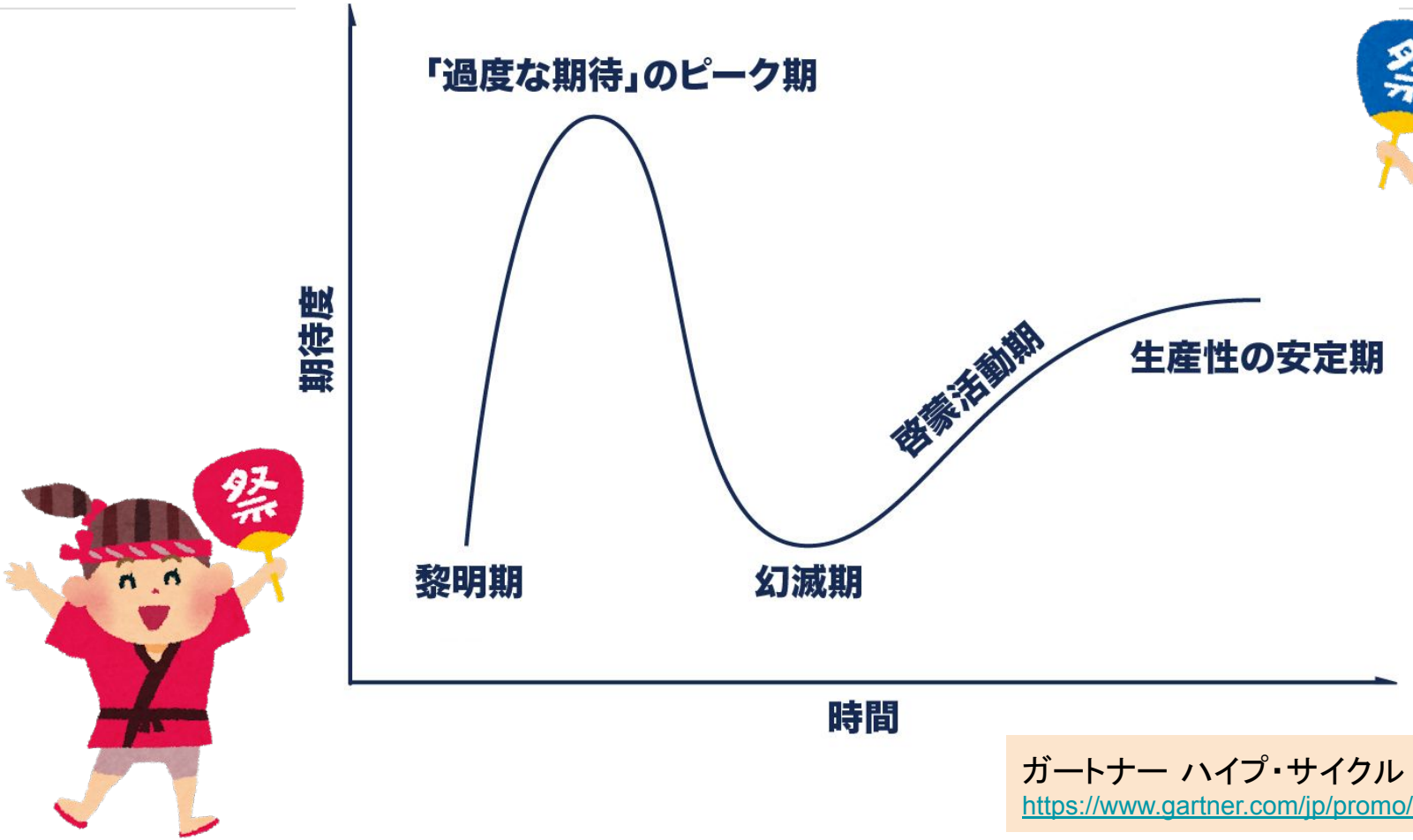


Photo by Joshua Hoehne on Unsplash

これからは「XXX」の時代だ

Retty



ガートナー ハイプ・サイクル
https://www.gartner.com/jp/promo/hype-cycle-report-list

銀の弾丸はありましたか？

Retty

- 技術
  - XML、クラウド、スマホアプリ、IoT、ブロックチェーン、AI/DeepLearning、Fintech、自動運転
- プロセス・文化
  - テスト駆動開発、アジャイル開発、ドメイン駆動開発、CI/CD、DevOps、マイクロサービス、DX
- 役割
  - CTO、VPoE、Engineering Manager、Product Manager

何が成功と失敗を分けるのか?

Retty

- 最新の技術を使っていれば・・・
- 上手くいっているやり方に倣えば・・・
- 最高のメンバーが集まれば・・・

うまくいくわけではない。

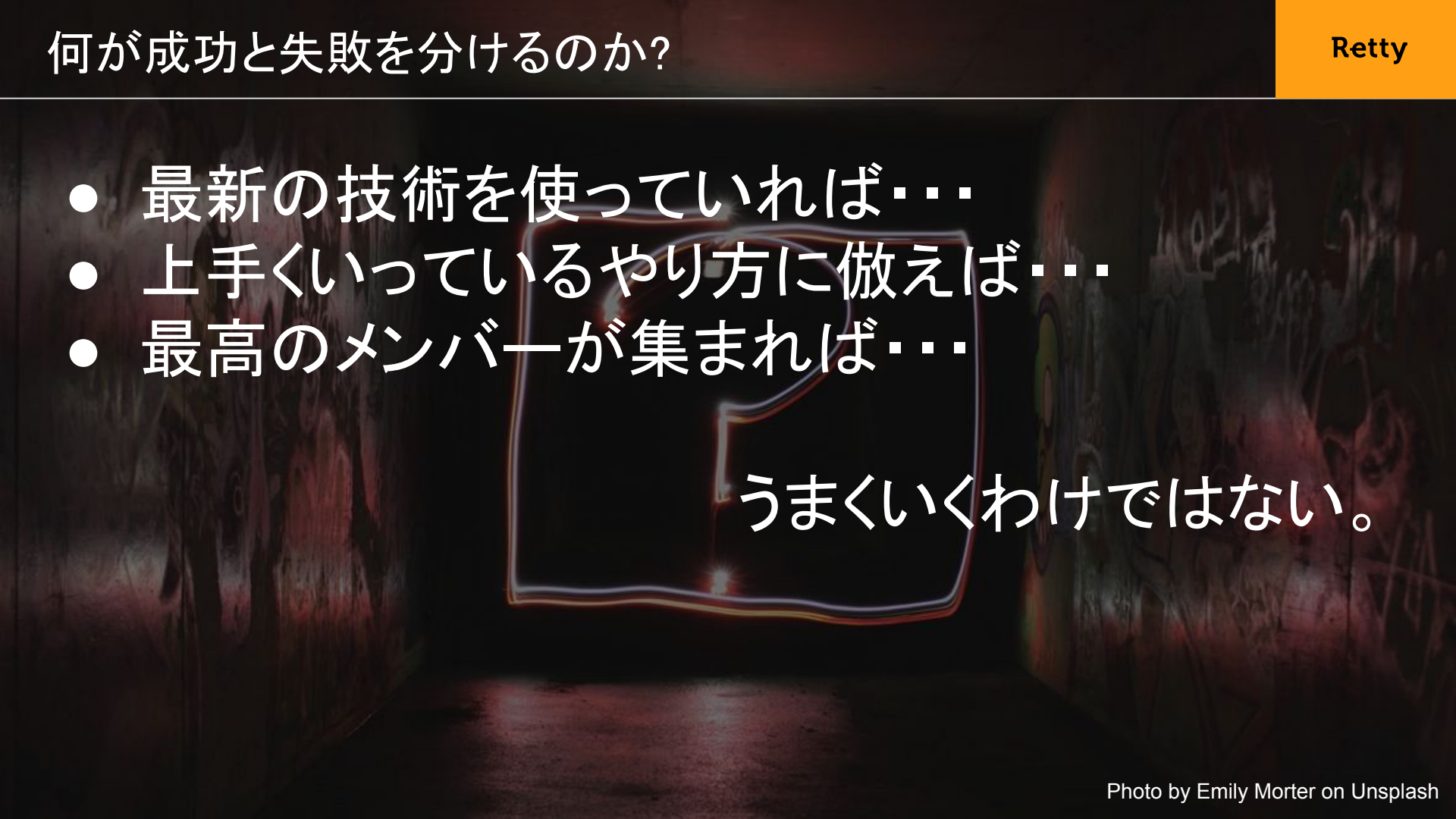
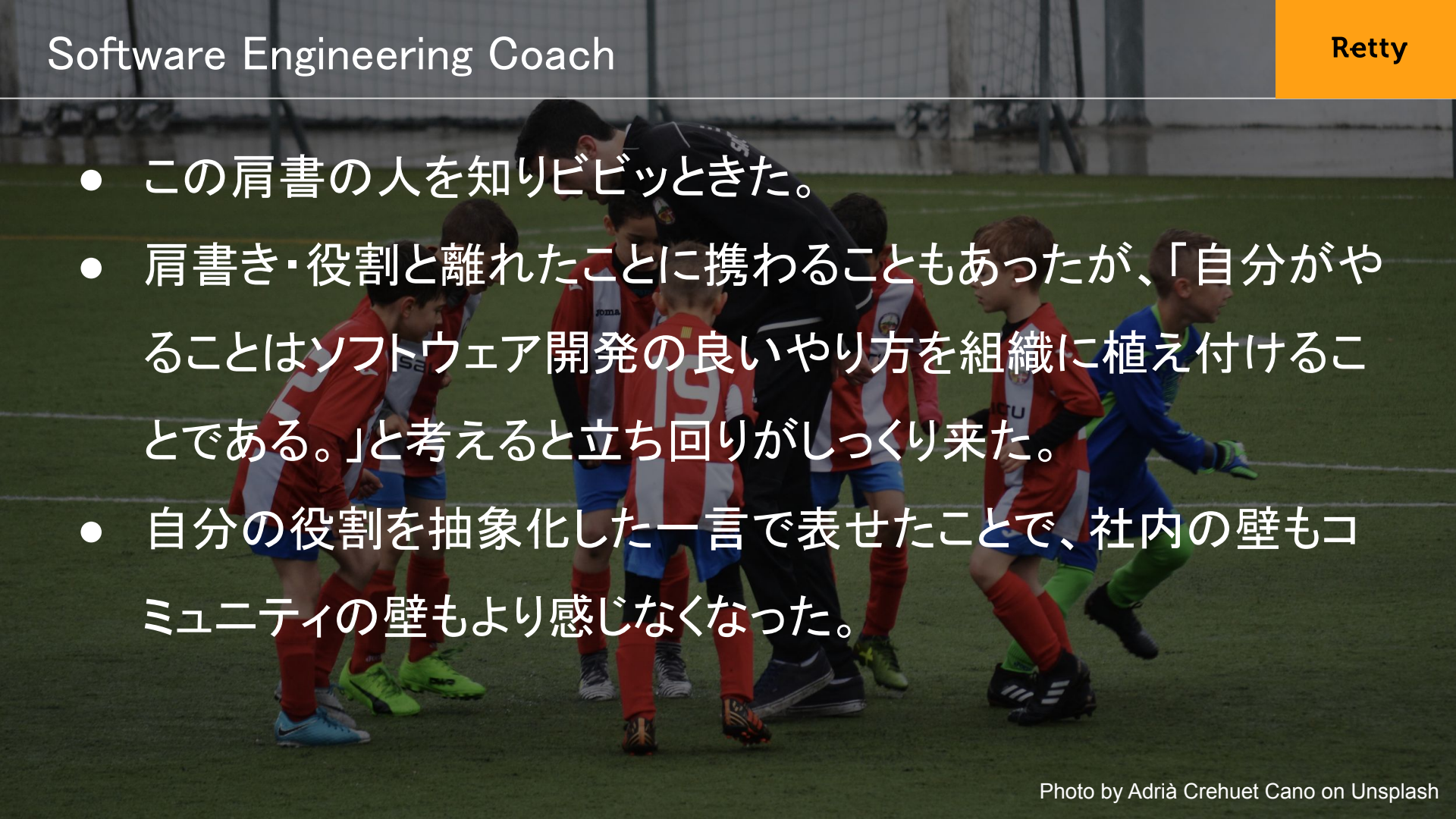


Photo by Emily Morter on Unsplash

Software Engineering Coach

Software Engineering Coach

Retty



- この肩書の人を知りビビッときた。
- 肩書き・役割と離れたことに携わることもあったが、「自分がやることはソフトウェア開発の良いやり方を組織に植え付けることである。」と考えると立ち回りがしっくり来た。
- 自分の役割を抽象化した一言で表せたことで、社内の壁もコミュニティの壁もより感じなくなった。

Photo by Adrià Crehuet Cano on Unsplash

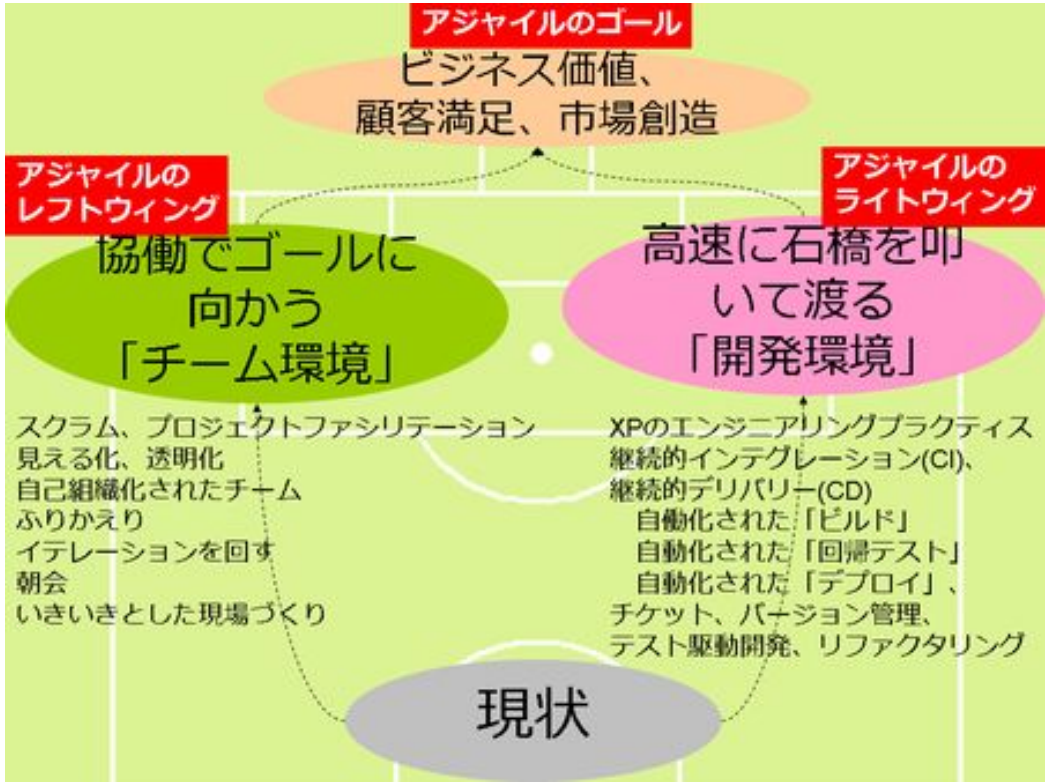
どんな素養を求められるのか

Retty

1. アジャイルの「ライトウィング」と「レフトウィング」
2. Certified Scrum Developer
3. Software Engineering at Google

アジャイルの「ライトウィング」と「レフトウィング」

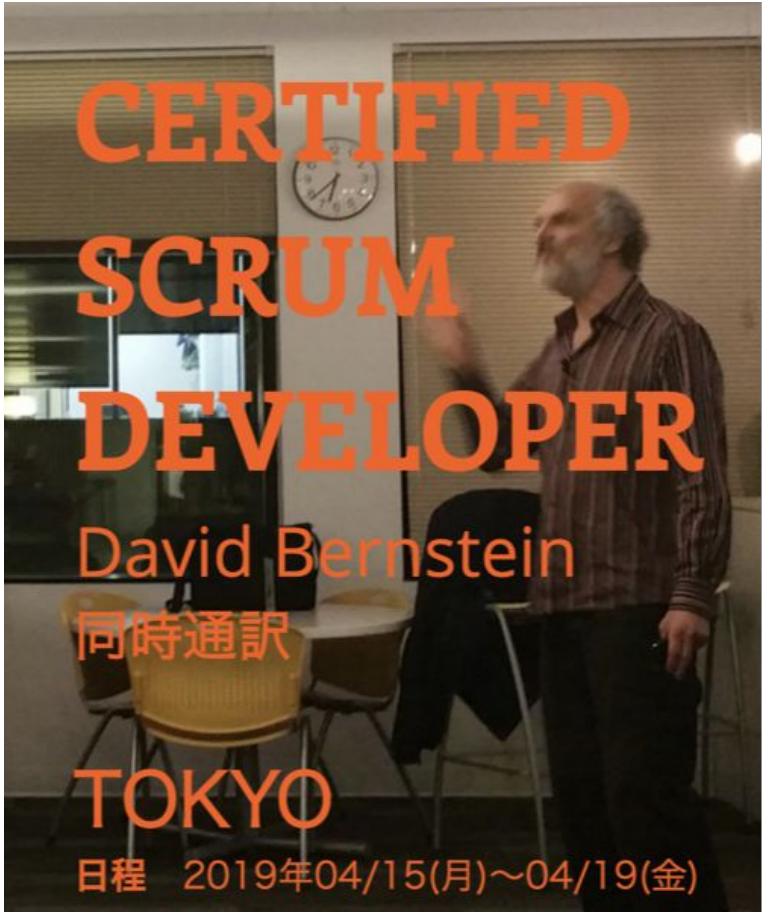
Retty



An Agile Way
https://blogs.itmedia.co.jp/hiranabe/2012/09/rightwing-and-leftwing-of-agile.html

Certified Scrum Developer

Retty

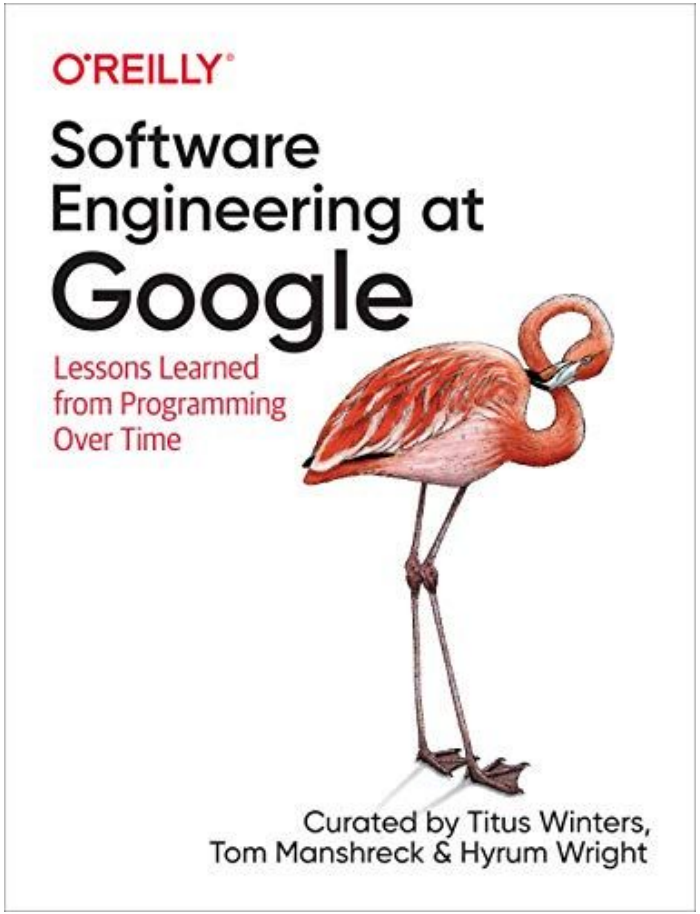


この研修で得られるもの

- スプリントでストーリーを記述し、機能を開発する
- 開発タスクをより正確に見積もる
- テストファースト開発をマスターし、設計を駆動する
- TDDの Red-Green-Refactoring のサイクルを効果的に使用する
- レガシーコードと付き合い、効果的にリファクタリングする
- 貧弱なコードの病状を診断して修正する
- テスト不能なコードをテストするテクニックを使う
- 協調的なペアプログラミングを上手に行う
- 受入テストを使って、ストーリーを定義し、ドキュメント化する
- 事前の過剰な設計を避け、ジャストインタイムな開発を行うプラクティス
- カプセル化する内容によって12のデザインパターンを使いこなす
- ソフトウェア開発の際に、ソフトウェアを継続的に統合するための戦略を定義する
- 過度の手戻りを避けながら、反復プロセスを用いてソフトウェアを作成する
- コードの共同所有を行い、共通の美的要素を取り入れる
- 反復的な開発を通じて、リファクタリングからパターンへ導き、デザインを創発する
- 設計を評価し伝達するための共通言語を共有する
- 技術的負債を認識し管理する手法を実践する
- コードの保守と拡張を容易にするソフトウェア品質を定量化する
- テスト駆動開発がデザイン決定にどのように情報を提供するかを認識する
- 共通のコーディング標準を採用することの価値を評価する
- そして他にも...

Software Engineering at Google

Retty



1. What Is Software Engineering?
2. How to Work Well on Teams
3. Knowledge Sharing
4. Engineering for Equity
5. How to Lead a Team
6. Leading at Scale
7. Measuring Engineering Productivity
8. Style Guides and Rules
9. Code Review
10. Documentation
11. Testing Overview
12. Unit Testing
13. Test Doubles
14. Larger Testing
15. Deprecation
16. Version Control and Branch Management
17. Code Search
18. Build Systems and Build Philosophy
19. Critique: Google's Code Review Tool
20. Static Analysis
21. Dependency Management
22. Large-Scale Changes
23. Continuous Integration
24. Continuous Delivery
25. Compute as a Service

初見の項目は・・・ほとんど無いですよね?

Retty

- チームでの開発
  - ドキュメント、コーディング規約、ブランチ戦略
- テスト
  - 受入テスト、テスト駆動開発
- コード品質を保つ
  - リファクタリング、コードレビュー
- 頻繁な統合・リリース
  - CI、CD

・・・とするとコーチは何をしてくれる人?

Retty

	ティーチング	コーチング
関係性	上下	対等
答え	教える	引き出す
コミュニケーション	依存する	自律する
行動	受動的	能動的
まとめると	相手に答えを教える	相手の中にある答えを引き出す

ティーティングとコーチングの違い
https://kurisu-sora.com/coaching/coaching-teaching/

コーチは必要か?

Retty

- 少年サッカーのコーチは、1流プロの戦術を教える?
- 扱う人の力量に応じて、うまく調整する役目をコーチは負っているはず。

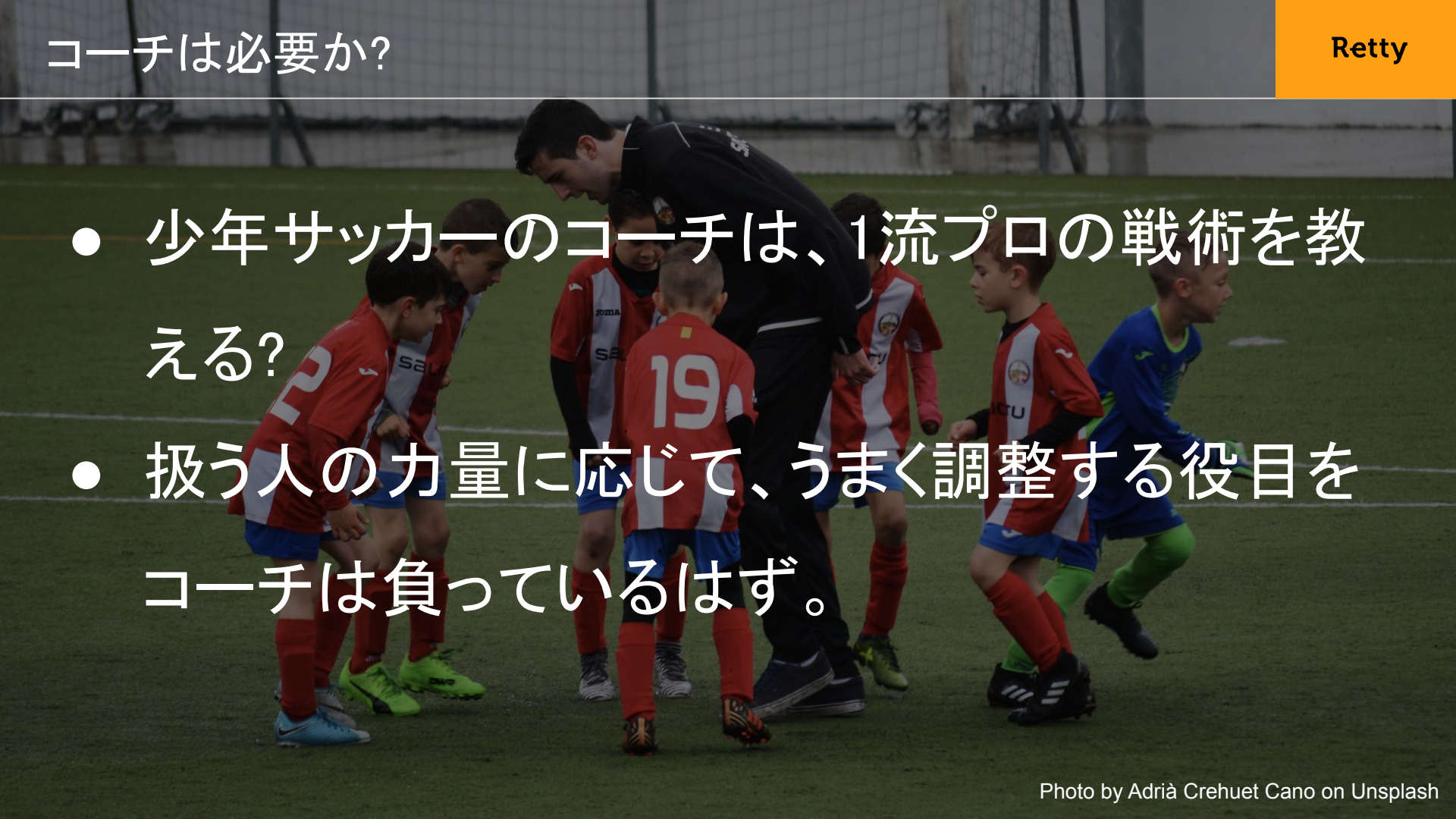


Photo by Adrià Crehuet Cano on Unsplash

コーチとしての振る舞い

状況を踏まえ、答えを引き出す

- 型にはめればうまくいくものではない。
  - テストを書くことを強制する
  - CIでゴリゴリチェック入れる・・・とかではない。
- 問題をシンプルに捉えて解決策を考える。
  - 問題の真因に迫り、適切な箇所にテコ入れする。
- 習慣化する
  - 気持ち・気合は長続きしない。
  - 「何のためにやっているのか」に立ち戻って整理する。

Rettyでの事例(1) – 型にはめる以外の解決策

Retty

ブランチ名はどうつけるべき?

Photo by NeONBRAND on Unsplash

ブランチの命名規則

Retty

- 答えは「i{Issue 番号}-{ブランチ作成者の名前}-develop」
  - 例) i12345-tunepolo-develop
- ルールが良い/悪いと議論するのではなく、なぜこのようなルールが必要になったのかを考える
- ルールが生まれた背景
  - マージされたブランチが削除されず残っている
  - 作りかけのブランチが放置される
  - ブランチの生存期間が長い（数週間〜数ヶ月）

ブランチの生存期間が短ければ名前の重要でなくなる

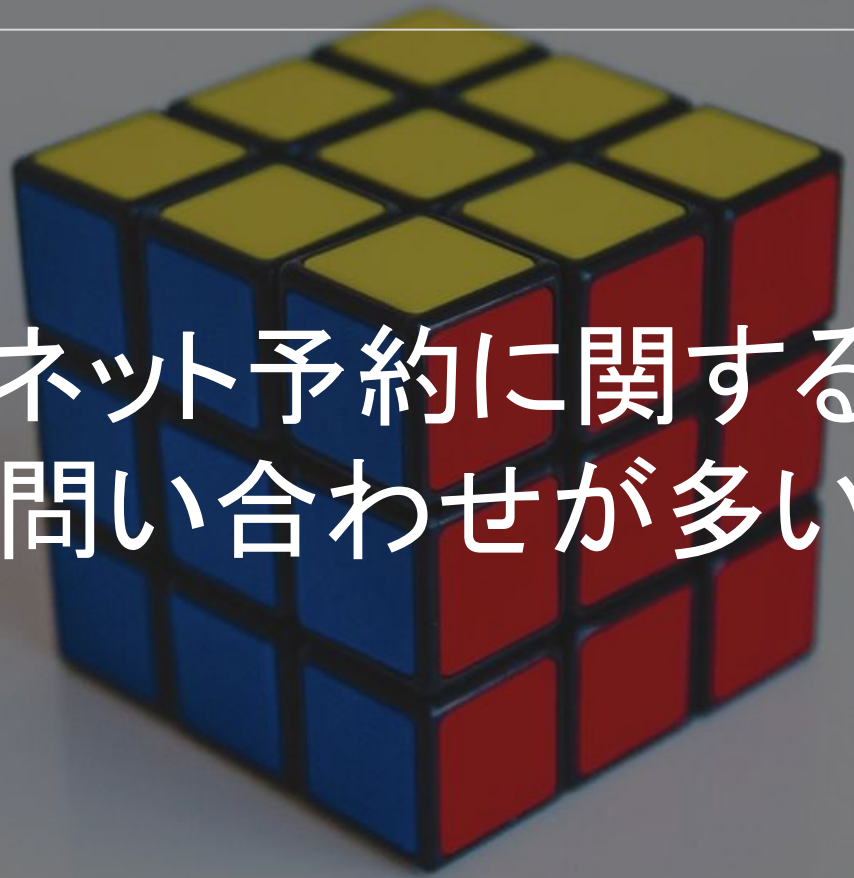
Retty

- マージされたブランチを削除
- 作りかけで不要になったブランチは削除
- マージされたブランチは自動で削除されるようGitHubを設定
- 開発単位を1週間以内で終わるまで小さく分割

→ルールは残っているが、従わなくても問題にならない

Rettyでの事例(2) – 問題をシンプルに捉える

Retty



ネット予約に関する
問い合わせが多い

Photo by NeONBRAND on Unsplash

飲食店の「ネット予約」は非常に複雑

Retty

- 基本的な概念
  - コース予約
  - 店舗の座席・テーブル
  - 在庫
- 上位の制約
  - コースの提供期間 / 受付期限
  - 臨時定休日
- 機能が複雑な上、サービス全体に影響するため、動作に関する問い合わせが非常に多かった。

コミュニケーションはシンプルにできる

Retty

- このような問題が起きた時にやりがちな解決策は・・・
  - 専任の責任者・チームを置く
  - 定例会議を開催する
- 私が実際に行ったことは・・・
  - 議論場所がたくさんあったので、誰でも参加できるオープンなSlackチャンネルを1箇所に統合し、他は全て削除(アーカイブ)
  - 責任者も定例も開催しなかったが、見つかった課題を順次潰していき問い合わせを減らすことができた。

Rettyでの事例(3) – 習慣化する

Retty

プロダクト動作に
不要なコードを削除したい

Photo by NeONBRAND on Unsplash

「不要なコード」削除を阻む壁

Retty

- 新規機能開発で手一杯で余裕がない
- 自分が書いたコードでないので影響範囲がわからない
- リファクタリングが評価されない。障害を出したらむしろマイナス評価を受ける。
- 習慣が無い。
- 削除がないため、社内でどう同意をとっていいのか誰も知らない。

実際に行った取り組み

Retty

- マネージャーとして「明確な不要コードの削除(6万行)」を指示
- 1エンジニアとして不要コードを削除するPull Requestを継続的に出す
- 不要コードの削除手順をドキュメント化
- 定期的な不要コードの削除をチーム目標に組み込む(3ヶ月で10件)
- 長期連休前後のリリース禁止期間に不要コードの削除にチームで取り組む
- 直近1年間のアクセスログを整理し、呼び出しがないAPIを確認できる資料を準備
- 1週間ごとに不要コードの削除をまとめて検証・リリースする仕組みを整備
- 不要コードの削除で障害を出しても担当者を責めない。すぐに検知・復旧できる仕組みを整備し、問題が大きくならないようにした。

コーチとして過ごすその先にあるもの

Retty

- 状況を踏まえたより良い解決策を模索していく先に、世の中に広まるアイデアの種がある
  - アジャイル開発
    - スクラム（ホンダ・キヤノンの事例をまとめた論文からJeff Sutherlandらが着想）
    - Lean / カンバン（トヨタ生産方式）
  - DevOps（Flickr）
  - モブプログラミング（Hunter Industries）

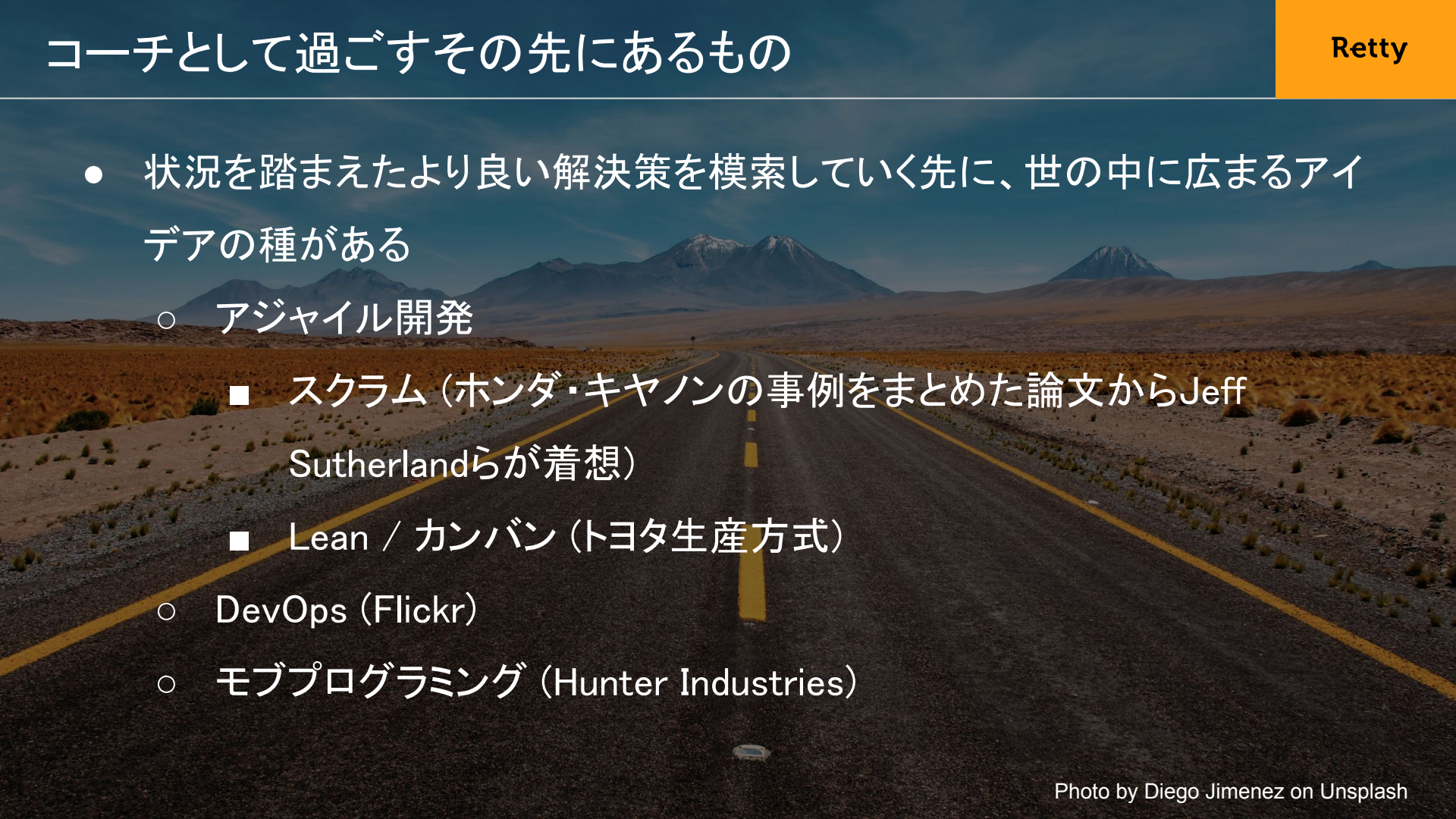


Photo by Diego Jimenez on Unsplash

まとめ

まとめ

Retty

- 状況を踏まえ、ソフトウェア開発をよくしていく答えを引き出すSoftware Engineering Coachという役回り
- 扱う人の力量に応じ、状況を踏まえて答えを引き出す役目をコーチが負う。
- 探究を積み重ねた先に優れたアイデアがある。

- 漫然と日々の開発に取り組むだけでなく、その現場ならではの悩みに正面から向き合ってみると良いのでは?

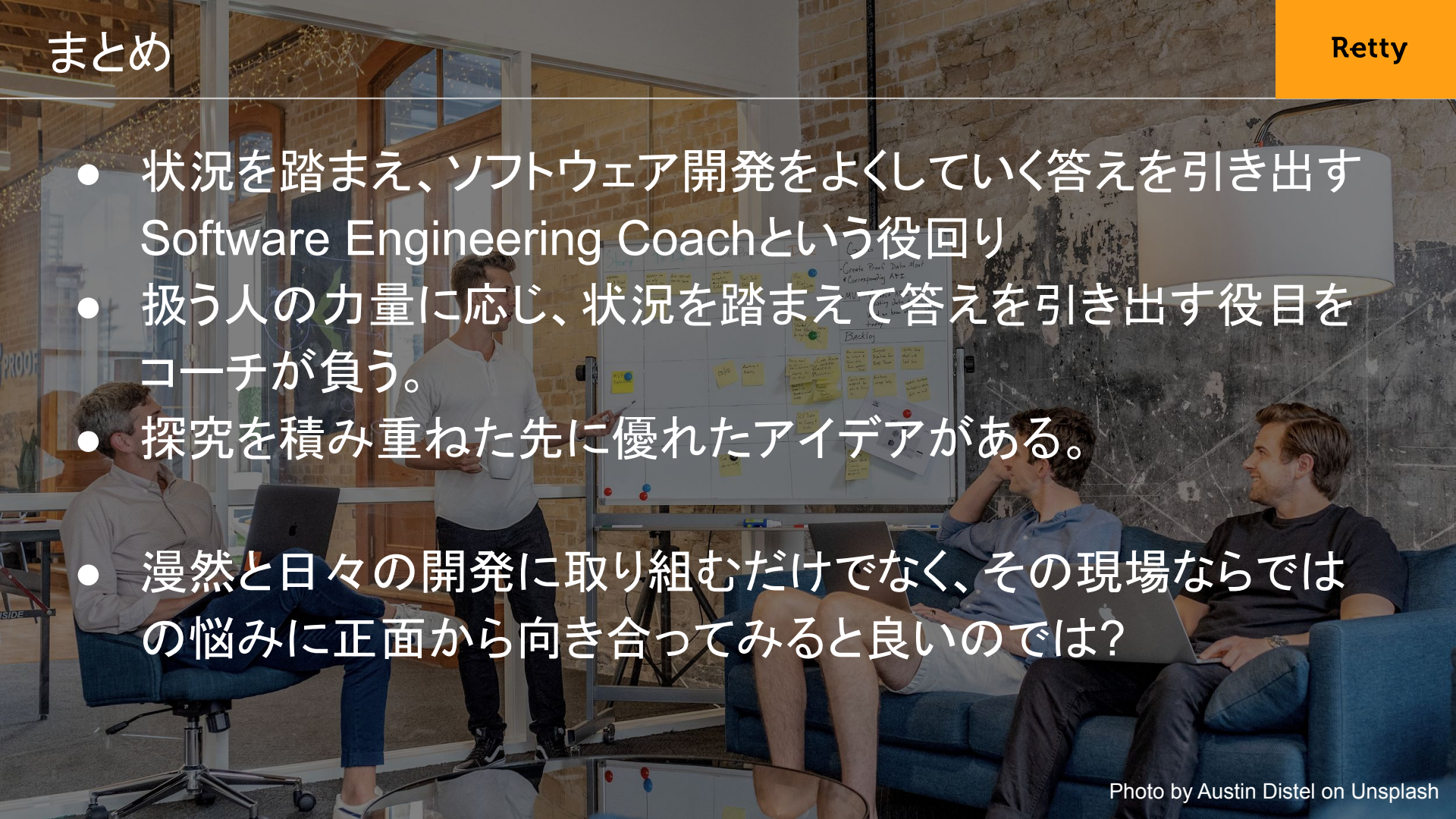


Photo by Austin Distel on Unsplash