

エンジニアリングの良い塩梅



【ORION】エンジニアLT会#6

'25/07/10

福井崇人

エンジニアリングにおける「ちょうどいい」バランスを探る旅
へようこそ

自己紹介

福井崇人

金融系SIや事業会社のエンジニアなどを経て、
現在はRetty株式会社に所属



エンジニアリングの良い塩梅(バランス)は難しい

観点



オーバーエンジニアリングかどうか

- どこまでパフォーマンスを良くするか
- どこまでセキュリティ対策するか
- ...



テストの厚さやカバレッジ



コードコメントの詳細度



ログの量



アラートの種類



Lintルールの厳格さ



...



良い塩梅はどう難しいか

(例えば)テストの厚さ

- 厚すぎると：ちょっとした変更でテストが失敗し、テスト修正コストがかかる
- 薄すぎると：品質リスクが高くなる

(例えば)ログの量

- 多すぎると：大事なログが埋もれて見つけにくく、保管コストがかかる
- 少なすぎると：調査しにくい

良い塩梅はなぜ難しいか

良い塩梅を決めるコンテキスト変数が多く、時变的

提供するものの中

- 提供者への影響
 - 人命, お金, 個人情報
- 提供するものの使い方
 - 本番で動く, 1回だけ使うスクリプト, ミッションクリティカル
- ドメインの可変性
- 実装上の手の入れやすさ
- ...

提供するものの外

- チームスキルセット
- 提供するもののフェーズ
 - PoC, 安定稼働期
- 会社の社会的信用
- ...

ケーススタディ

万博スポンサーのセキュリティ

万博スポンサーをしている会社の社内セキュリティ環境

→ 情報漏洩があったら信用を失う影響が大きいので強め

金のなる木の保守

保守できる人が少なく、手を入れる頻度が少ない"金のなる木"のサービスに手を入れる際のコードコメント

→ サービス廃止の可能性は低く、今後も保守し続ける可能性があるため、詳細に書く

社内システムの監視

社内で使われる、業務が滞りにくいWebサービスのアラート

→ アラート設定せず、不具合があった場合は利用者から直接連絡してもらう

どう良い塩梅を見つけていくか

やりすぎを経験せずに、良い塩梅向かっていくのではなく...

やりすぎ

良い塩梅

やらないすぎ



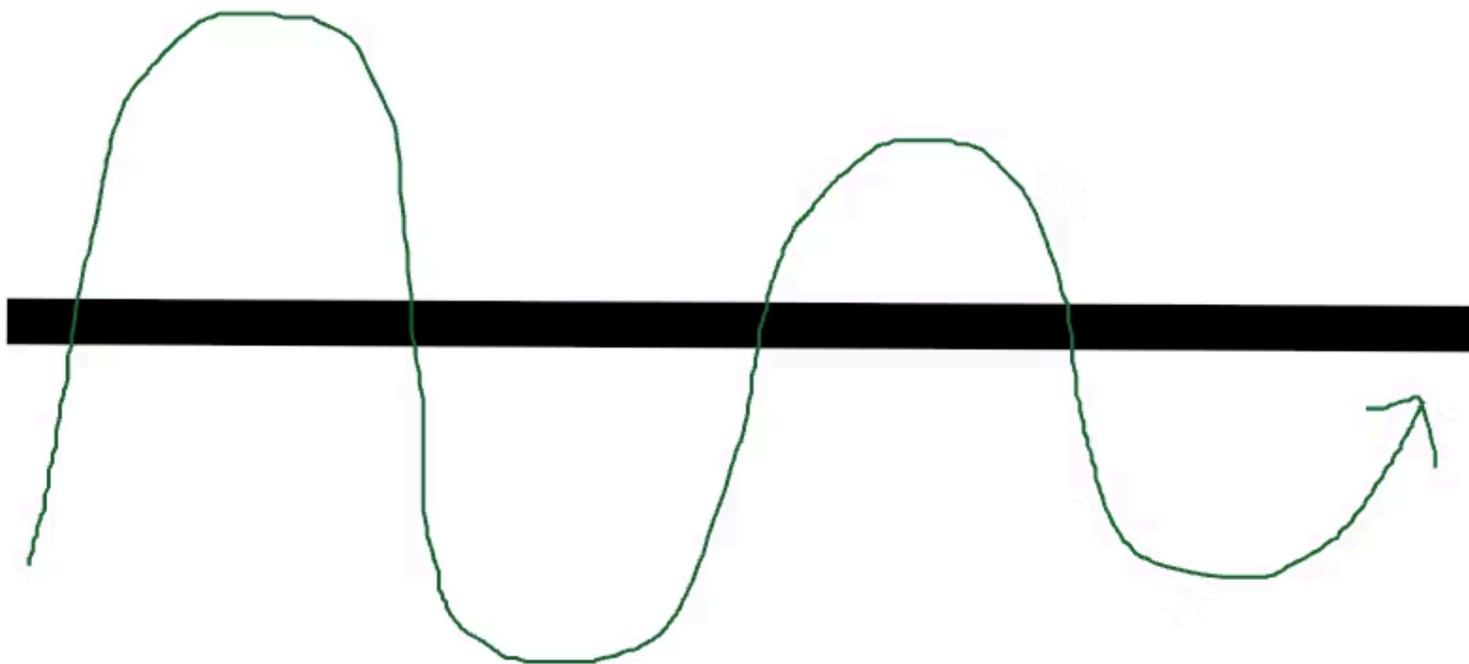
どう良い塩梅を見つけていくか

やりすぎ、やらないすぎの経験を繰り返して、良い塩梅を見つけていくイメージ

やりすぎ

良い塩梅

やらないすぎ





やりすぎた、やらなすぎたを どう気づき、振り返るか



"感想"を深掘る

「分かりづらい」「使いづらい」といった他エンジニアの感想、自分の感想を深掘りする

日常会話から気づきを得られる



開発活動を深掘る

「開発規模の割に開発遅くない？」
「アウトプットの割に手を入れる箇所が多くない？」



不具合を深掘る

不具合、インシデント原因を深掘る

「不具合が起きるのは複雑な設計だから？」

でそれは良い塩梅なの？

課題が少なくなってくれば良い塩梅になってきてる
課題が少なければ良い塩梅
と言えるのでは



エンジニアリング以外でも同じ

これら良い塩梅の考え方はエンジニアリング以外でも同じ：

- チーム内外のコミュニケーション量
- サービス/プロダクトマニュアルの詳細度
- テレワーク or 出社
- ...



まとめ

- 良い塩梅はコンテキスト変数に依存する
- 良い塩梅を見つけるには"収束する周波数"のように
- 雑に言えば、何事においても「トレードオフを考慮してPDCAを回す」と同じ話

