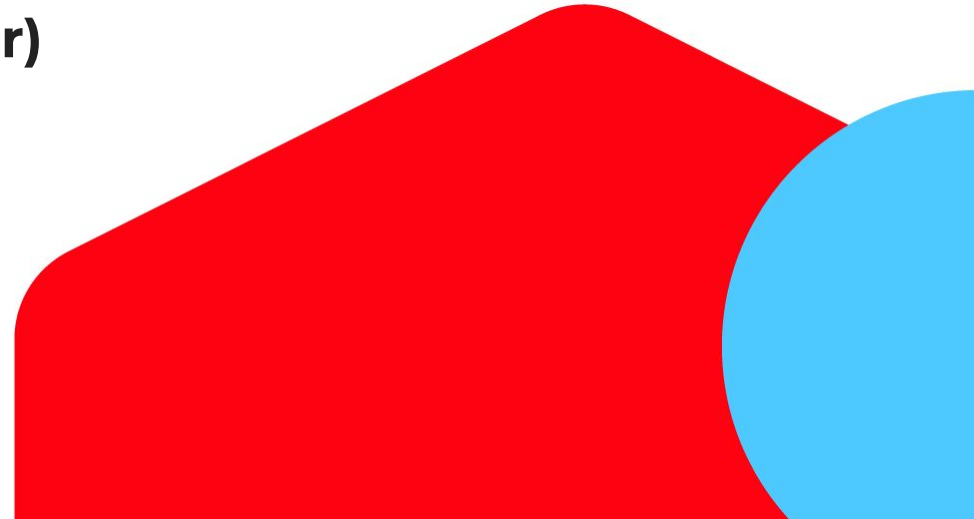
「推測するな、計測せよ」
～小さく始める生産性可視化と分析～

2023-05-30 開発生産性を高める～ソウゾウ、Voicyの挑戦と苦労～

株式会社メルカリ 佐藤直人(@naopr)



mercari

自己紹介

@naopr (なおぱー)

- Engineering Manager at Mercari
- メルカリShopsの開発を担当
- Love 🐱 ⛰ 🍺 🍶
- https://twitter.com/naopr

アジェンダ

01 メルカリShopsと開発組織

02 生産性可視化プロジェクト(PJ-Productivity)の立ち上げ

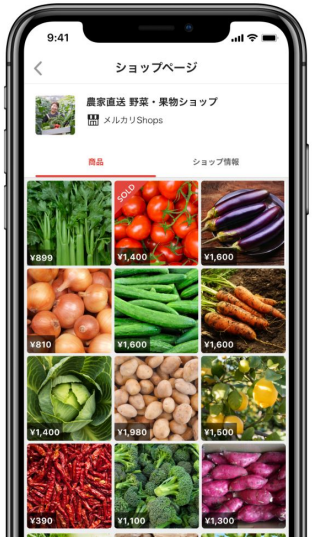
03 生産性可視化までの挑戦と苦労

04 指標の分析と改善の取り組み

メルカリShopsとは

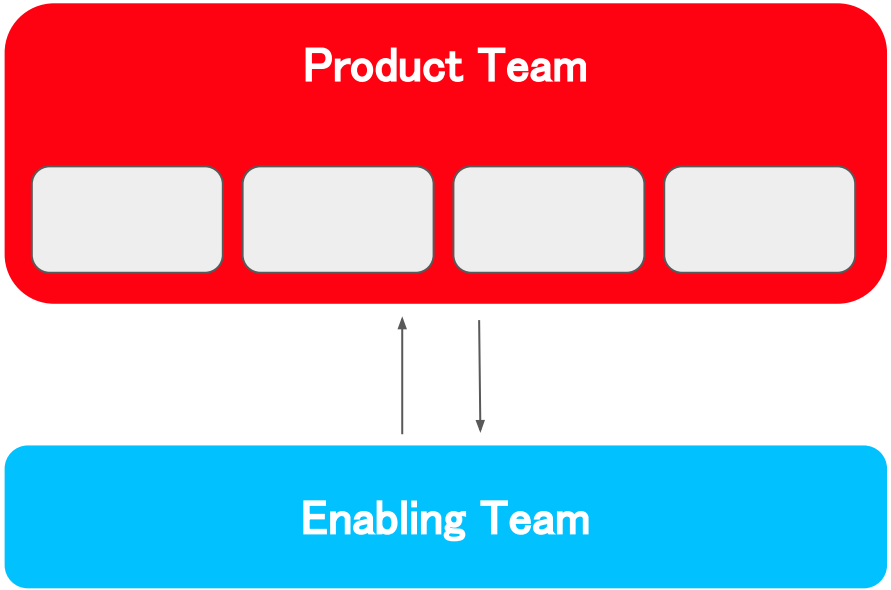
「かんたんで、売れる」スマホ1つでネットショップを開設できる
Eコマースプラットフォーム

Shops



mercari

メルカリShopsの開発組織



Product Team

- お客さまの課題を解決
- PdM/Designer/QA/SWE/EM

Enabling Team

- Product Teamの技術を強化
- SRE/ML/Architect/EM

生産性可視化プロジェクト(PJ-Productivity)の立ち上げ

～リリース2ヶ月後、メルカリOB(mさん)とnaoprの飲み会～

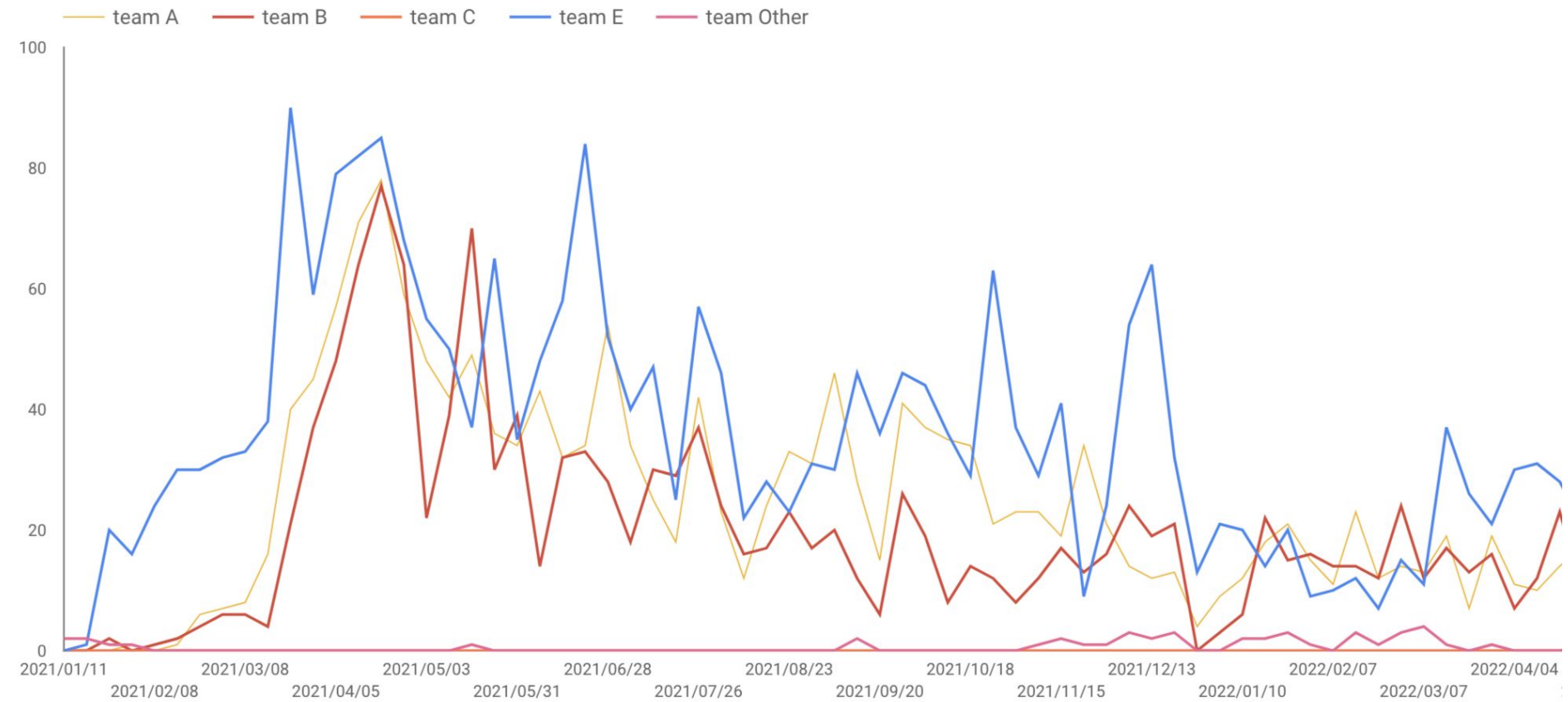
- m「開発組織の生産性可視化やってる会社が増えてきたよねー」
- n「生産性落ちたと思ったときに客観的な数値あるといいよね」
- m「『推測するな、計測せよ』は大事だよね」
- n「ソウゾウも組織が大きくなる前に可視化しておきたいなあ」
- n「mさん、ソウゾウで一緒に生産性可視化やりませんか！」

→ 1ヶ月後 mさん業務委託でソウゾウに入社
→ naoprとmさんの2人でPJ-Productivity立ち上げ

生産性可視化までの挑戦と苦労

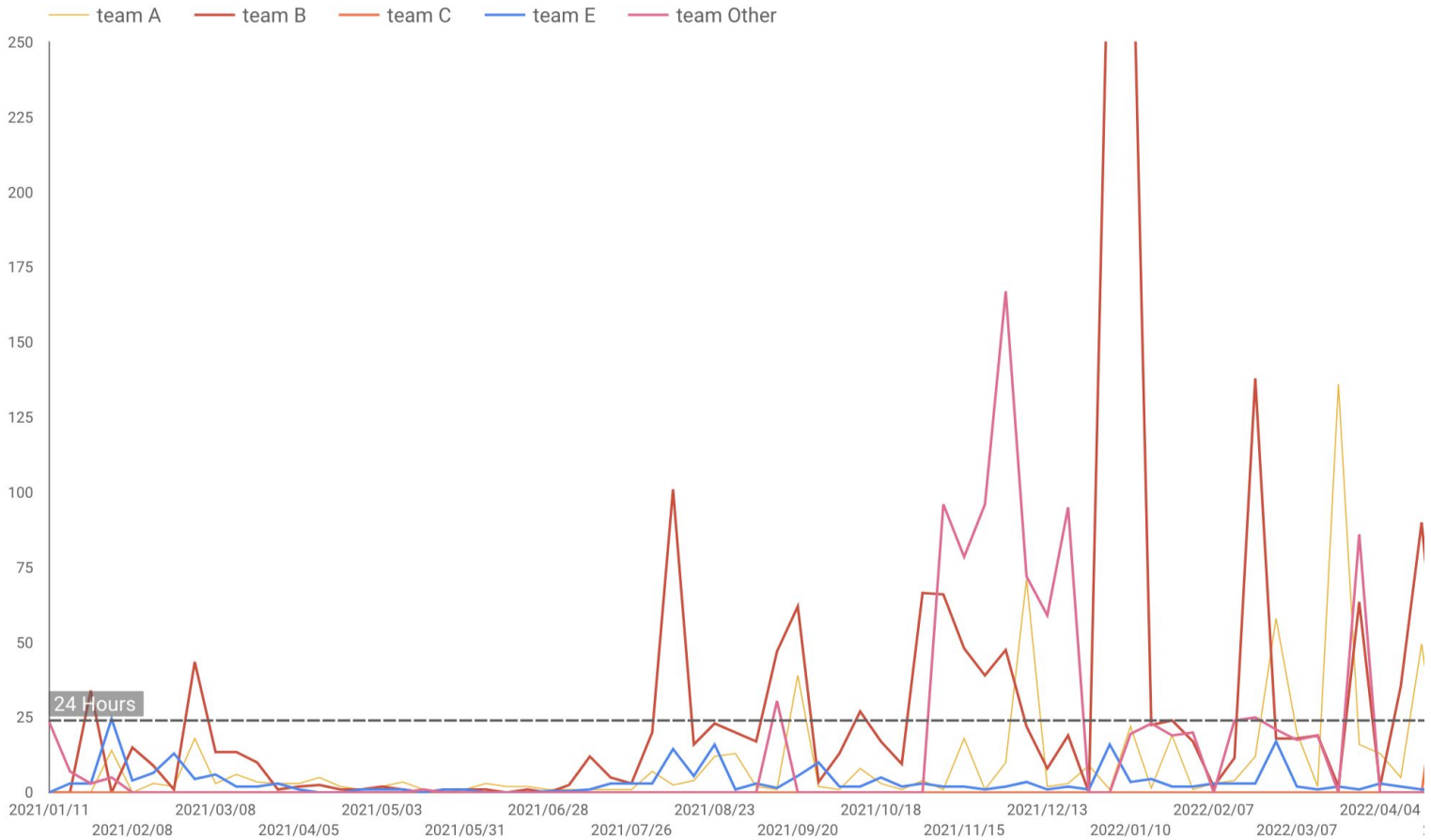
1. 外部SaaSの導入を検討
    a. 社内IT・セキュリティチームと相談
    b. 先方とメール・オンライン会議で相談
        → セキュリティ・コンプライアンス観点の条件が折り合わず見送り
2. 自前での開発を検討
    a. GitHub APIのPull Request関連のものを調査
    b. Googleの提唱しているFour Keys※について調査
        → 最小工数で取得でき、自分たちにとって意味のある指標を特定

※ 参照：エリート DevOps チームであることを Four Keys プロジェクトで確認する
https://cloud.google.com/blog/ja/products/gcp/using-the-four-keys-to-measure-your-devops-performance



デプロイ数：masterブランチにmergeされた週ごとのPull-reqの数

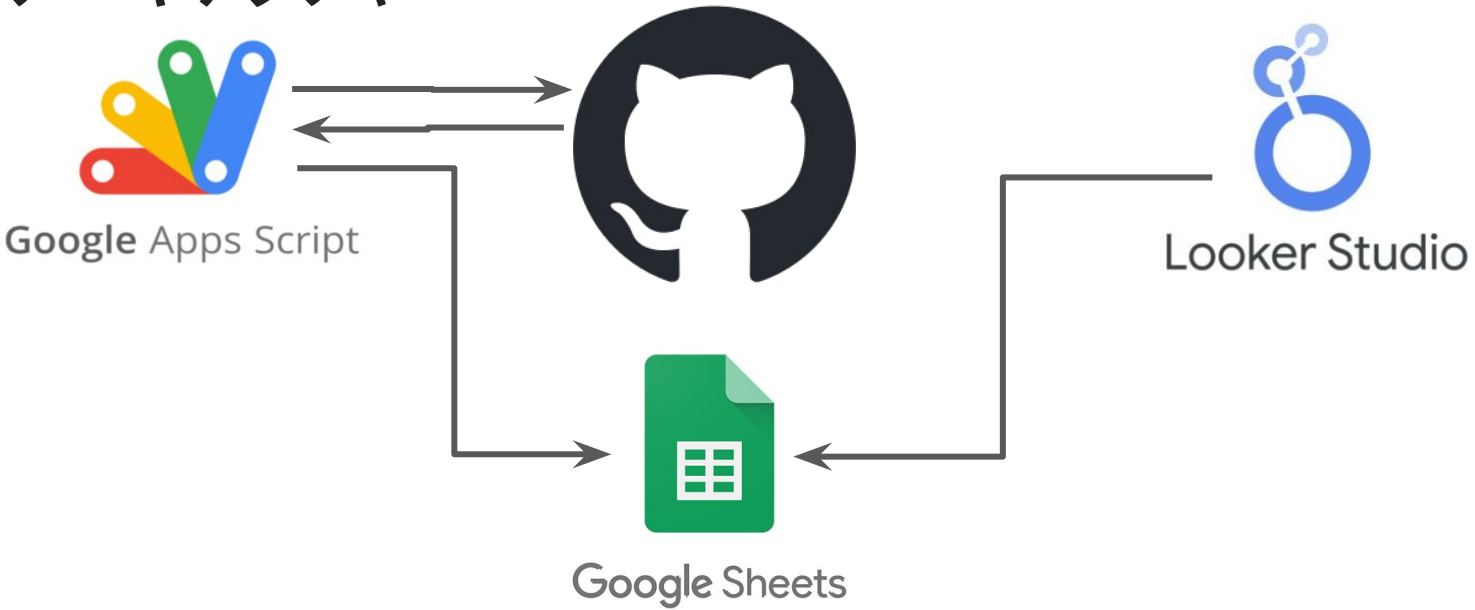
mercari



リードタイム：Pull-reqが作成されてからmasterブランチにmergeされるまでの時間の週ごとの中央値

mercari

アーキテクチャ



	A	B	C	D	E	F
1	Title	Author	Create a	Closed a	GitHub URL	Team
2	chore(go): update version	naopr	2022/01/01	2022/01/02	https://github.com/hoge/foo/pull/1	Team A
3	fix(item): fix bug	bob	2022/01/01	2022/01/02	https://github.com/hoge/foo/pull/2	Team B
4	test(auth): add test	bunta	2022/01/01	2022/01/03	https://github.com/hoge/foo/pull/3	Team C

指標の分析：相対的な自社の立ち位置

Breaking Down DevOps Performance Metrics

	Elite	High	Medium	Low
Deployment frequency	**On-demand** (multiple deploys per day)	Between **once per day** and **once per week**	Between **once per week** and **once per month**	Between **once per month** and **once every six months**
Lead time for changes	Less than **one day**	Between **one day** and **one week**	Between **one week** and **one month**	Between **one month** and **six months**
Time to restore service	Less than **one hour**	Less than **one day**	Less than **one day**	Between **one week** and **one month**
Change failure rate	0-15%	0-15%	0-15%	46-60%
2019 Distribution	20%	23%	44%	12%

Google Cloud

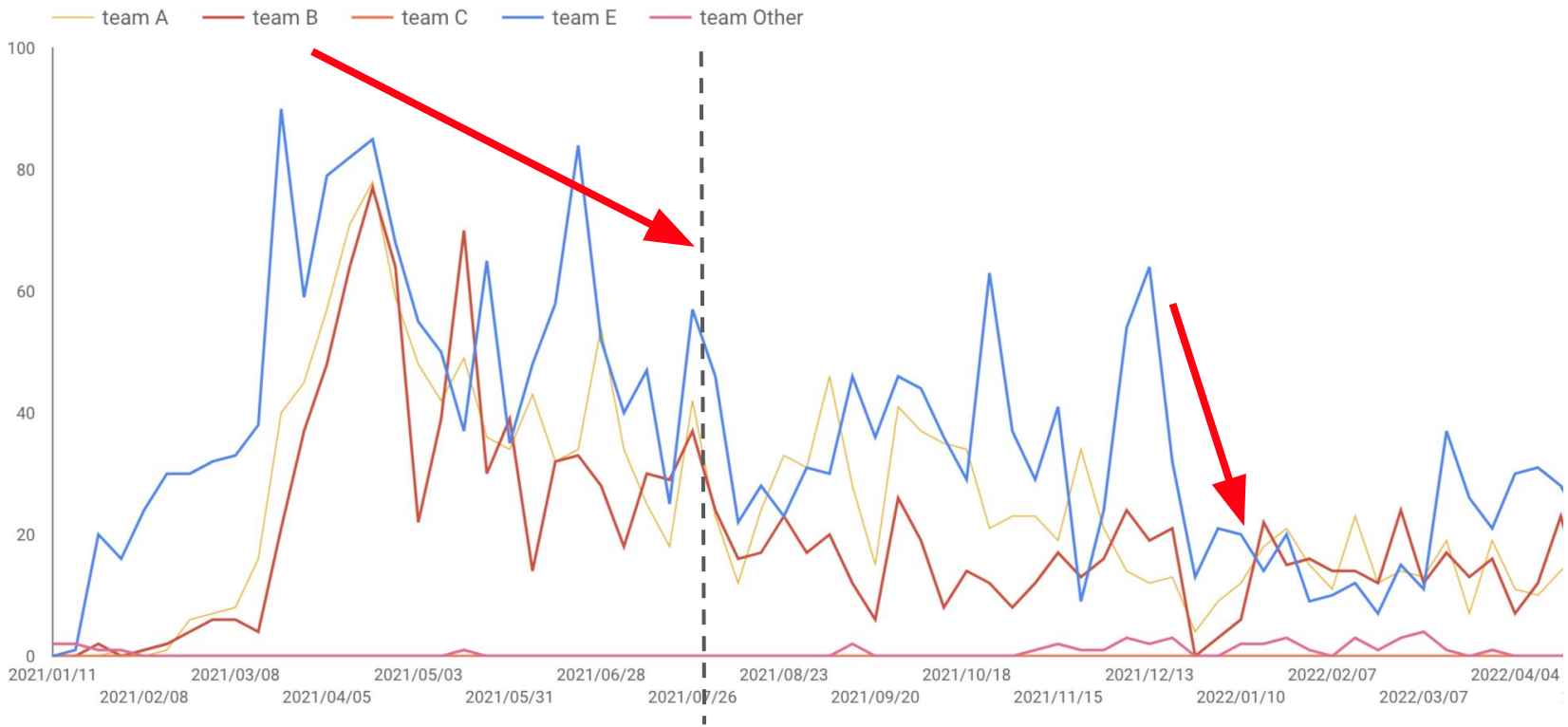
※ 参照：Google Cloud で実行されている DevOps 組織の有効性を評価する
https://cloud.google.com/blog/ja/products/gcp/another-way-to-gauge-your-devops-performance-according-to-dora

指標の分析：デプロイ数とリードタイムの大まかな傾向

- 1stリリース時に比べると
  - デプロイ数 減少
  - リードタイム 増加
- 開発チームの人数が増えてるのにデプロイ数が減っている
- リードタイムはチームによる差が大

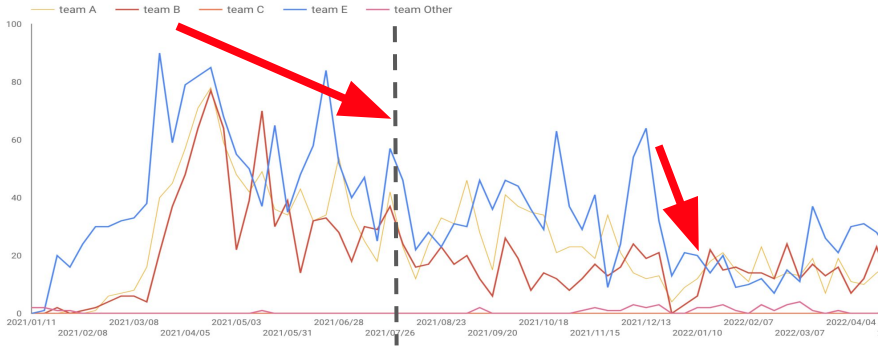
⇒ 深掘って調査する

指標の分析：デプロイ数の変遷（1/3）



指標の分析：デプロイ数の変遷（2/3）

- リリース前のデプロイ数の多さ
  - コードレビューが厳格でない
  - QAせずにデプロイしている
  - 稼働が高い

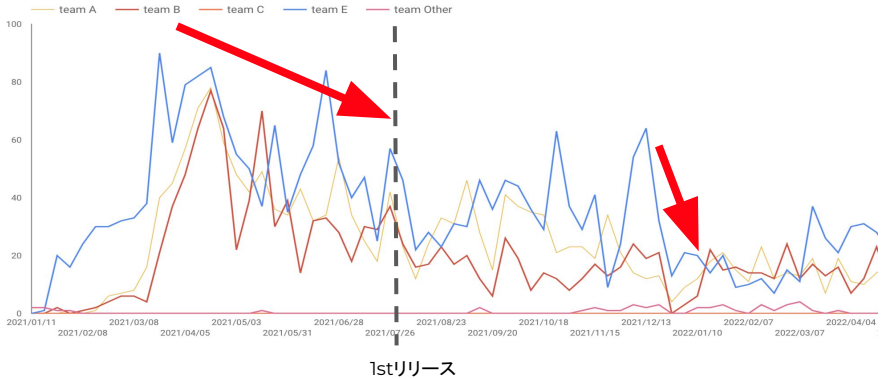


1stリリース

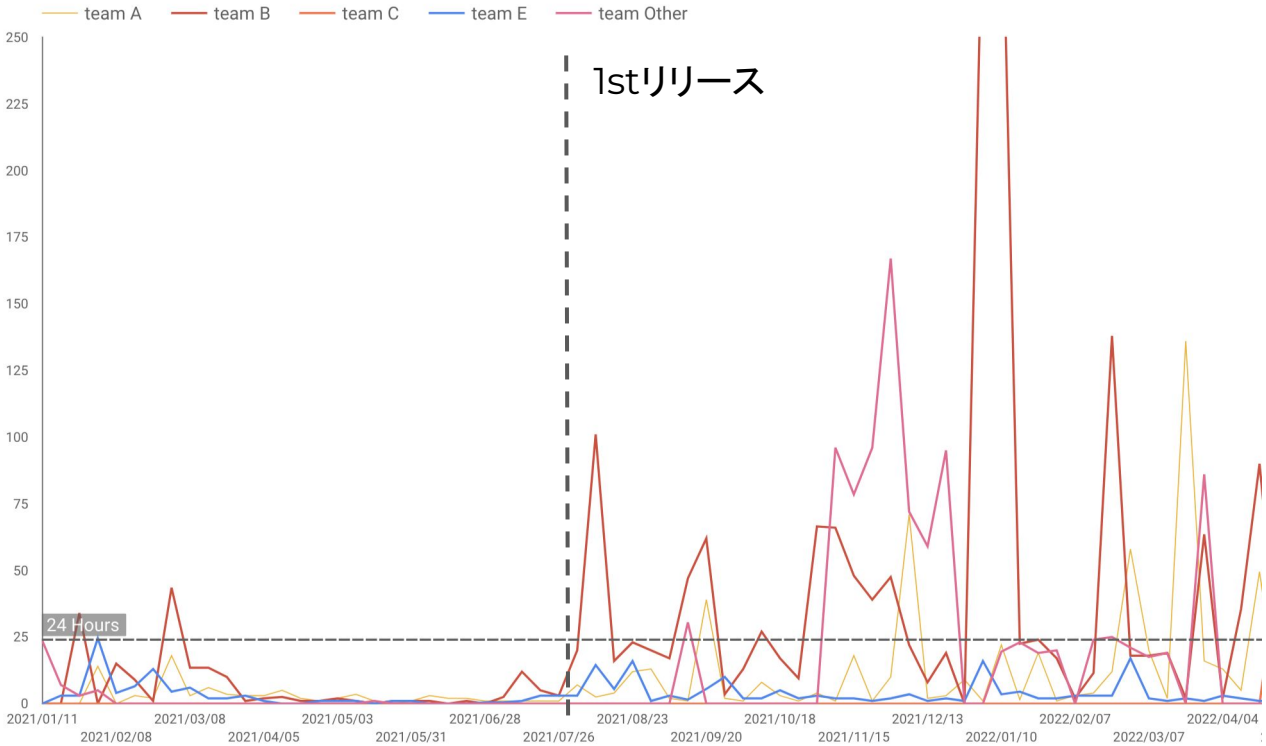
- 1stリリース後、1月にデプロイ数が減ったのはなぜか
  - 開発チームの人数は増えているのでデプロイ数も増えるはずと思っていた
  - デプロイ数の多い月と少ない月の 2ヶ月分を選んでPull-reqの内容を全部読んで分類した
  - 1stリリース後の12月までと1月からで明確にPull-reqの傾向が変わっていた

指標の分析：デプロイ数の変遷（3/3）

- 1stリリース〜12月までのPull-req
  - 機能リリース
  - バグ対応
  - インフラの設定追加
    - team Eのデプロイの大部分がこれ
  - ⇒ 見かけ上のデプロイ数が増えていた
- 結論
  - 12月までのデプロイ数は参考値として扱うべき
  - 1月以降はデプロイ数が微増しているので想定通り

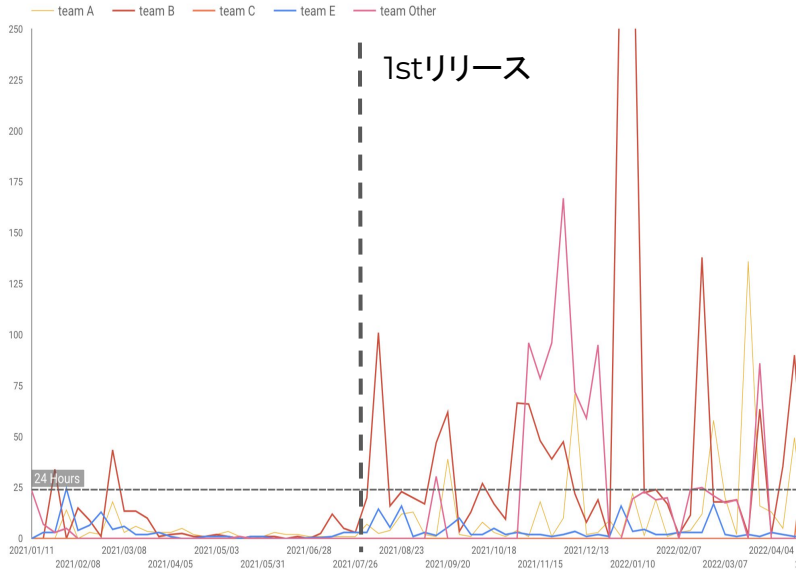


指標の分析：リードタイムの変遷（1/3）

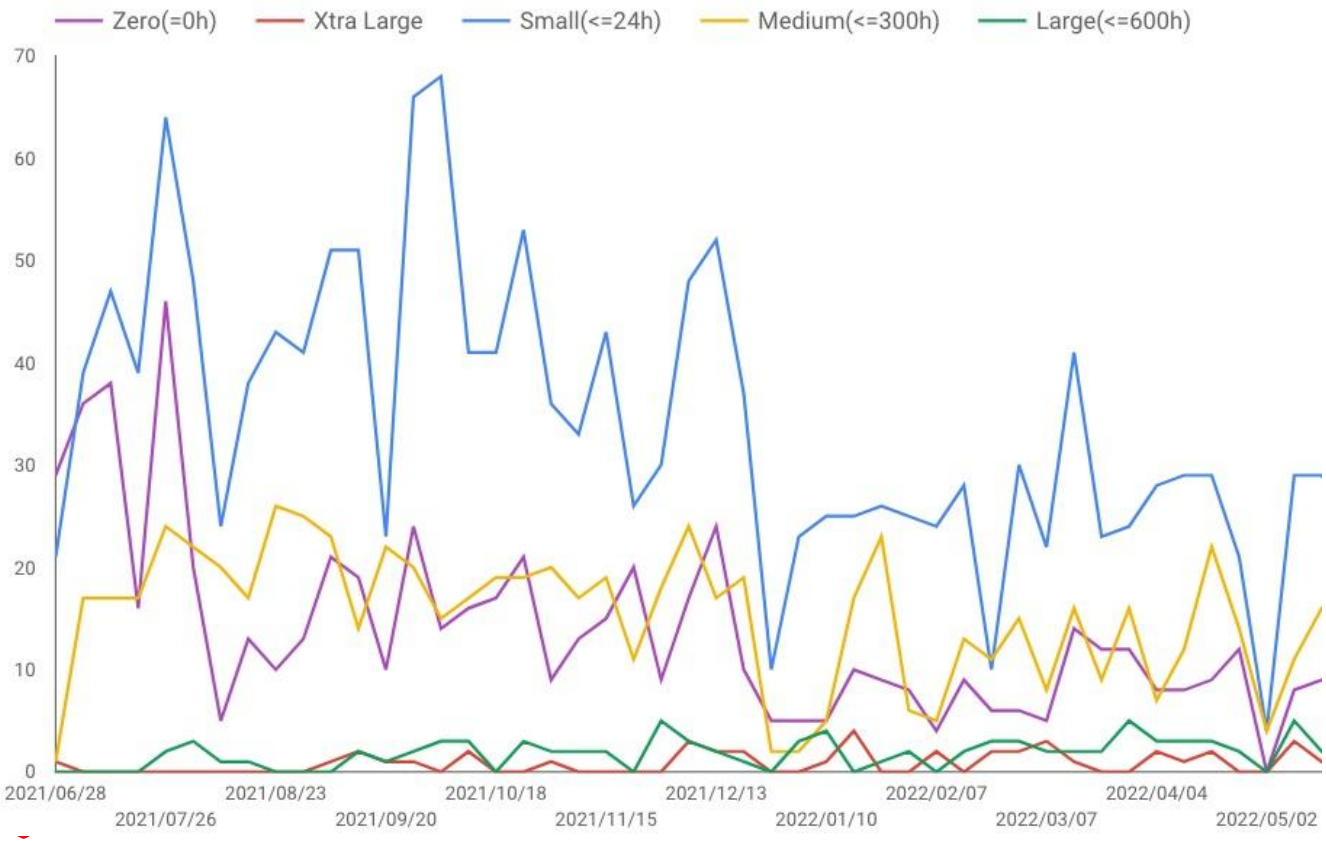


指標の分析：リードタイムの変遷（2/3）

- 1stリリース後に急増
  - QAによるもの
- チームによって大きな差がある
  - team B：大
  - team E：小
  - ⇒ 開発する機能の違い
- これだけではよくわからなかったので別の角度から分析した



指標の分析：リードタイムの変遷（3/3）



改善の取り組み：定量アプローチ

- リードタイムが極端に大きいPull-reqを減らす
  - Why
    - 長期間masterブランチにマージできない Pull-reqはQAコストが大きかったり、master ブランチを取り込む際のコンフリクトが発生しやすい
    - ビッグバンリリースになりやすいので障害のリスクが大きい
  - How
    - 新規マイクロサービスや新規 APIといった既存機能に影響のないバックエンド差分を先に リリースする取り組みを一部チームで始める
  - Result
    - 短期間に劇的にリードタイムが小さくなることはなかった
    - デプロイ作業が増えることへの負荷

改善の取り組み：定性アプローチ

- 開発チームにアンケートを実施
  - 開発に対する満足度をお答えください（ 5段階評価）
    - なぜそのように感じましたか？
  - 開発をする上で、一番課題に感じてる点は何ですか？
  - より生産性の高い組織にするために足りないものは何でしょうか？
- 寄せられた回答の中から取り組みやすく効果の高いもの3つに絞って改善活動をスタート
  - サブプロジェクト化して PJメンバー以外にリードしてもらっているものもある

まとめ

- 業務委託メンバーと2人で生産性可視化PJを立ち上げた
- 最小工数で必要最低限の数値から可視化ダッシュボードを実装した
- 可視化したデプロイ数とリードタイムをもとに少しずつ改善を開始した
- 定量面だけでなく定性面からも改善を行っている

Appendix

- 「推測するな、計測せよ」ソウゾウで始動したエンジニア組織の生産性可視化
  - https://mercan.mercari.com/articles/34283/
- 生産性可視化PJを通して実現したい理想的な開発組織とは
  - https://open.spotify.com/episode/6xJXsNynFv1bvQ8WvYhm4g
- 【続編】「推測するな、計測せよ」エンジニア組織の生産性可視化 ～定量分析で得られた知見、定性分析の開始、そしてチーム拡大へ～
  - https://engineering.mercari.com/blog/entry/20221116-souzoh-productivity/