

Platform は好きですか？

Press Space for next page →

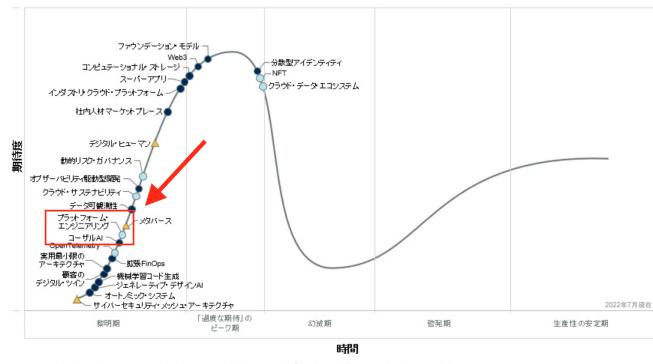
Platform Engineering?

- 最近、Platform Engineering という言葉が話題
 - Platform Engineering MeetUpの参加者500人超
 - Gartnerのハイプサイクルにも登場
 - 3 Exciting New Trends in the Gartner Emerging Technologies Hype Cycle
 - Thoughtworks TechRadarでは2017年に登場
 - Platform engineering product teams | Technology Radar | Thoughtworks
 - 2017/MAR: Assess ~ 2021/OCT: Adopt



ref: Platform Engineering Meetup - connpass

先進テクノロジのハイプ・サイクル:2022年



ref: Gartner、「先進テクノロジのハイプ・サイクル:2022年」を発表

"Platform"...? 🤔

モチベーション

- Platform Engineeringはバズワードっぽい匂いがする
 - 一般的な名詞の組み合わせで、どうとでも解釈することができ、かつ多くの人が関係していそうな気がしてくる(名前的に近しい活動をしてると尚更)
- よく分からぬままワードが降ってくる前に調べておこうというモチベーション
 - [Why] なぜPlatform Engineeringをやるのか
 - [Who] 誰がPlatform Engineeringをやるのか
 - [What] Platform Engineeringとは何をすることなのか
 - [How] どのようにPlatform Engineeringをやるのか

Why?

Platform Engineering

CNCF TAGの『Platforms White Paper v1』を読んでみる - Zenn

過去20～30年にインフラストラクチャの柔軟性が向上すると共に、ビルドやテスト、オブザーバビリティなどの開発者向けサービスが増加し、利便性が向上するというメリットが生まれました。一方で、アプリケーション/プロダクトチームはインフラストラクチャの管理に要する責任と時間が増え、それらを管理する認知負荷が向上した結果、開発者として価値を生み出す作業（つまり開発作業という意味でしょう）に要する時間が減ってきた、という問題が発生したことです。

→ DevとOpsの垣根を取り払ってきた結果、開発チームが担う役割が多くなってきた。（問題）

別の問題として、各チームごとにインフラストラクチャや開発者向けサービスを導入することで、同様の作業を各チームで重複してやっており、チームごとの品質面のバラツキが生じている場合があります。

→ DevOpsチームごとに重複する作業があるが、そこを共通化する余裕がない（問題）

これらをPlatformおよびPlatform Teamとして共通化することで、開発者の認知負荷を減少させると共に、信頼性やガバナンス、セキュリティ、コストの最適化をはかる、というのがPlatformの価値ということのようです。

→ 組織内でやり方を統一化するPlatformにより全体最適を図りたい（解決の方向性）

Platform Engineeringは
DevOpsの推進による開発チームの負荷向上や
チーム間のバラツキを解消するための取り組み

共通技術施策「やあ、また会ったね。」

Who?

Platform Engineering

Team Topology

本書は高速なデリバリーを実現することを目的とした、4つの基本的なチームタイプと3つのインタラクションパターンに基づく、組織設計とチームインタラクションのための実践的な適応モデルを紹介する。

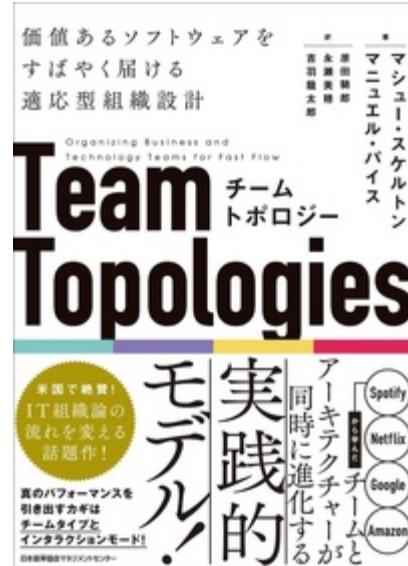
ref: チームトポロジー - この本の内容

コンウェイの法則 ref: コンウェイの法則 - Martin Fowler's bliki

システム（広義に定義）を設計するあらゆる組織は、組織のコミュニケーション構造をコピーした構造を持つ設計を生み出す。
—メルヴィン・コンウェイ

逆コンウェイの法則 ref: チームトポロジー - KEY TAKEAWAYS

チームトポロジーはチームの目的と責任を明確にし、チーム間の相互関係の効果を向上させる



Team Topology

4つのTeam

Stream-aligned team

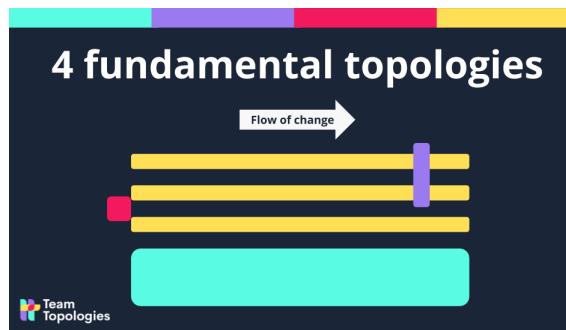
ストリームアライドチームとは、価値のある単一の仕事のストリームに沿って働くチームのことだ。(中略)さらに、なるべくそのチームだけで素早く安全に顧客やユーザーに価値を届けられるよう、チームに権限が委譲されている。(中略)ストリームアライドチームは、組織で根幹となるチームタイプで、**残りの基本的なチームタイプの目的は、ストリームアライドチームの負荷を減らすことである。**

ref: Ch.5 4つの基本的なチームタイプ | ストリームアライドチーム

Enabling team

Complicated Subsystem team

Platform team



ref: [Core ideas in Team Topologies](#)

Team Topology

4つのTeam

Stream-aligned team

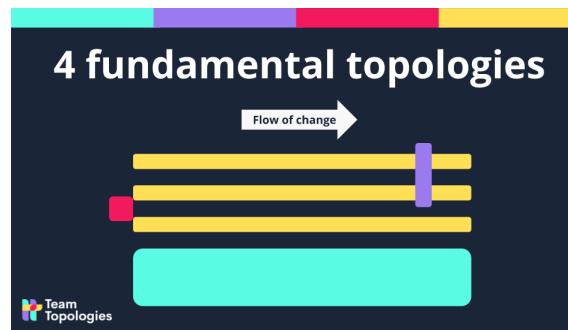
Enabling team

イネイブリングチームは、特定のテクニカル(プロダクト)ドメインのスペシャリストから構成され、(ストリームアライドチームの)**能力ギャップを埋めるの**を助ける。複数のストリームアライドチームを横断的に支援し、適切なツール、プラクティス、フレームワークなどアプリケーションスタックのエコシステムに関する調査、オプションの探索、正しい情報に基づく提案を行う。

ref: Ch.5 4つの基本的なチームタイプ | イネイブリングチーム

Complicated Subsystem team

Platform team



ref: [Core ideas in Team Topologies](#)

Team Topology

4つのTeam

Stream-aligned team

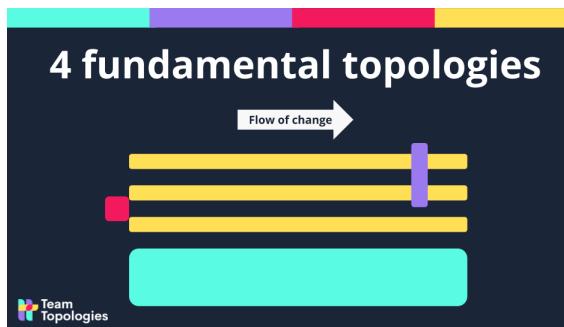
Enabling team

Complicated Subsystem team

コンプリケイティッド・サブシステムチームは、システムのなかでスペシャリストの知識が必要となるパートを開発、保守する責任を持つ。(中略)このチームの目的は、複雑なサブシステムを含んだり利用するシステムの担当となるストリームアライドチームの認知負荷を減らすことにある。

ref: Ch.5 4つの基本的なチームタイプ | コンプリケイティッドサブシステムチーム

Platform team



ref: [Core ideas in Team Topologies](#)

Team Topology

4つのTeam

Stream-aligned team

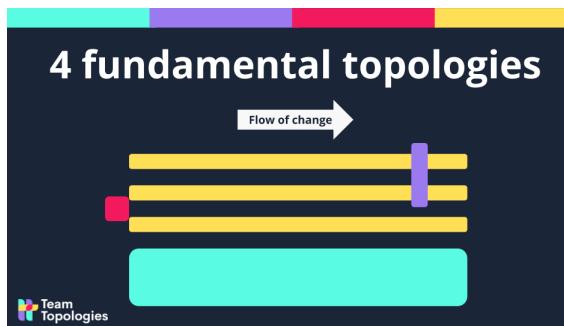
Enabling team

Complicated Subsystem team

Platform team

プラットフォームチームの目的は、ストリームアライドチームが自律的に仕事を届けられるようにすることである。(中略) プラットフォームチームは、内部サービスを提供することで、ストリームアライドチームが下位のサービスを開発する必要性をなくし、認知負荷を下げる。

ref: Ch.5 4つの基本的なチームタイプ | プラットフォームチーム



ref: [Core ideas in Team Topologies](#)

Platform Engineeringは
Stream-aligned team の自律的な取り組みを
サポートするPlatform teamの責務

What?

Platform Engineering

IDP

Internal Devloper Portal/Platform

- Platform teamの一つの形として組織内でIDPを提供することがある
- IDP (Internal Developer Portal/Platform)?
 - 開発組織がセルフサービスで利用できるツールおよびワークフローを提供する基盤

Platform engineering is the discipline of designing and building toolchains and workflows that **enable self-service capabilities for software engineering organizations** in the cloud-native era. Platform engineers provide an integrated product most often referred to as an “**Internal Developer Platform**” covering the operational necessities of the entire lifecycle of an application.

ref: What is platform engineering? | platformengineering.org

【3】IDPの必要性

Platformとして提供しておくべきもの

- サービスカタログ
 - PlatformやToolのカタログ
 - Product等のサービスAPI仕様
- テンプレート
- ドキュメント

重要な点

- Product Teamが必要なものにすぐにリーチし利用できること
- Self-serviceで利用できること

こうした機能・サービスを提供するのがBackstage (やHumanitec) などのIDP製品・サービス

【3】IDPの必要性

IDP(Internal Developer Portal/Platform)は「銀の弾丸」ではない

- Developer Portalが提供するのは入り口のみ
- Developer Platform製品はセルフサービス化する仕組みを提供

提供すべきPlatformやその抽象化は、Platform Teamが実現すべき課題

(IDP導入による認知負荷増大も考慮する必要がある)

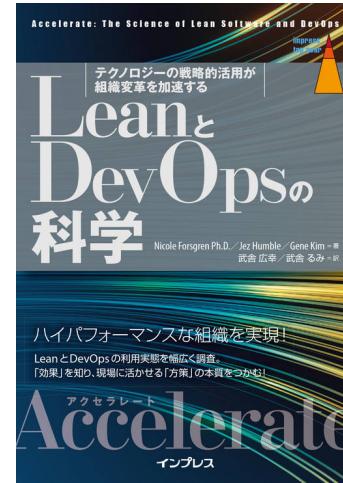
ref: PlatformCon 2023 RecapというテーマでPlatform Engineering Meetup #3
で登壇しました - APC 技術ブログ

Good/Bad IDP

開発チームが欲した機能の単なる集合がプラットフォームになるわけではない。業界の技術動向と組織ニーズを踏まえ、全体として一貫性を保ちながら精巧に開発されるものだ。良いプラットフォームは、セキュリティチーム、監査チームが開発チームと行う仕事も減らすことができる。

ref: Ch.5 4つの基本的なチームタイプ | 良いプラットフォームの大きさは「ちょうどよい大きさ」

単一の技術スタックやツールを強制し、チームの選択の自由を奪うと、仕事で適切なツールを使う能力に大きな悪影響を及ぼし、モチベーションを阻害したり、時にはモチベーションを消し去ったりしてしまう。『LeanとDevOpsの科学』の中で著者は、チームに標準化を強制することで学習や実験が減り、その結果貧弱なソリューションの選択につながるという研究結果に言及している。

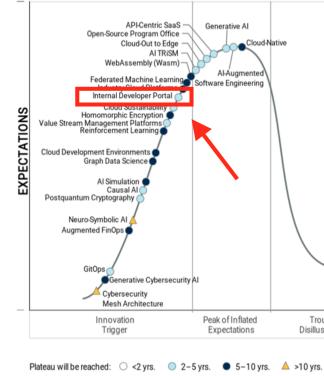


ref: LeanとDevOpsの科学 [Accelerate] テクノロジーの戦略的活用が組織変革を加速する - インプレスフックス

ref: Ch.6 チームファーストな境界を決める | 隠れモノリスと結合

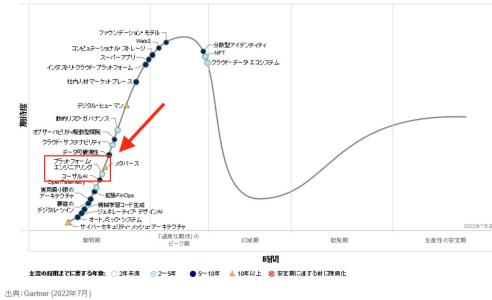
IDP

- 2023/08/16に発表されたGartner「先進テクノロジーのハイプサイクル2023年」には`Platform Engineering`に代わって`Internal Developer Portal`が登場している



ref: 米ガートナー「先進テクノロジーのハイプサイクル2023年」を発表...
Publickey

先進テクノロジーのハイプ・サイクル: 2022年



ref: Gartner、「先進テクノロジーのハイプ・サイクル: 2022年」を発表

Platform Engineerは
Stream-aligned teamが必要とする機能を
セルフサービス型で提供するために
Internal Developer Portal (IDP)を構築する

How?

Platform Engineering

Platform teamへの期待

期待される振る舞い

- ストリームアラインドチームと密接にやり取りしニーズを理解する
- ストリームアラインドチームを巻き込んでフィードバックを素早く得る
- プラットフォームをプロダクトとして扱いUXと信頼性にフォーカスする
- 提供したサービスをストリームアラインドチームと共に利用し、手本を示す
ことでリードする

ref: Ch.5 4つの基本的なチームタイプ | プラットフォームチーム

Principles of platform engineering

- Clear mission and role
- Treat your platform as a product
- Focus on common problems
- Glue is valuable
- Don't reinvent the wheel

ref: What is platform engineering? | platformengineering.org

Backstage

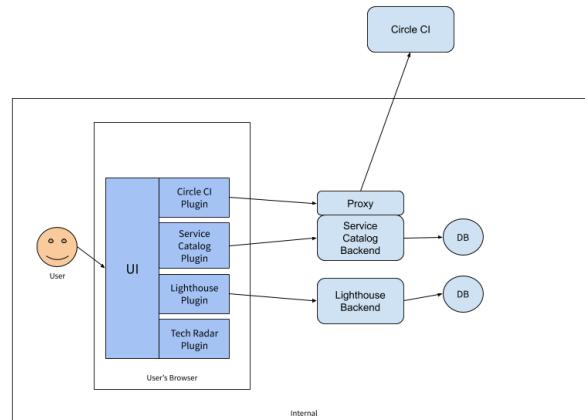
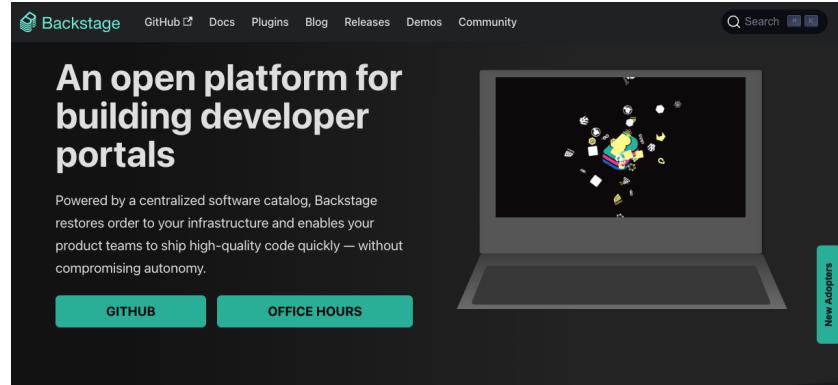
<https://backstage.io/>

- Spotifyが開発している開発者ポータルシステム
 - 2020/3月にApache License v2.0として公開
 - 2022/3月にCNCF Incubatingプロジェクトに昇格 [Backstage | Cloud Native Computing Foundation](#)
 - 2022/3月にv1.0がリリース

Backstage unifies all your infrastructure tooling, services, and documentation to create a streamlined development environment from end to end.

ref: [What is Backstage? | Backstage Software Catalog and Developer Platform](#)

→ Demo: <https://demo.backstage.io>



ref: [Architecture overview | Backstage Software Catalog and Developer Platform](#)

Platform Engineerは
IDPをプロダクトとして(Platform as a Product)提供し、
そのUX/信頼性を向上することで、
Stream-aligned teamの自律性をサポートする。
IDPの例としてBackstageなどがある。

まとめ

Wrap Up

[Why]

Platform EngineeringはDevOpsの推進による開発チームの負荷向上やチーム間のバラツキを解消するための取り組み

[Who]

Platform EngineeringはStream-aligned team の自律的な取り組みをサポートするPlatform teamの責務

[What]

Platform EngineerはStream-aligned teamが必要とする機能をセルフサービス型で提供するためにInternal Developer Portal (IDP)を構築する

[How]

Platform EngineerはIDPをプロダクトとして(Platform as a Product)提供し、そのUX/信頼性を向上することで、Stream-aligned teamの自律性をサポートする。IDPの例としてBackstageなどがある。

感想

- 基本的には今まであった共通技術基盤的な話
 - システム開発の中で開発組織が担う部分が薄くなってきており、XaaS/OSSなど共通して使われるものが多くなってきているという背景も共通技術基盤を後押し
 - DevOpsというコンテキストの流れで生まれた新たな潮流