

Sponsor LT

「マイクロサービスはもう十分」か？

森 久太郎 (@qsona), 株式会社FiNC

Microservices Meetup vol.6

半年前の記事

マイクロサービスはもう十分 | プロダクト・サービス | POSTD

669 users [ロテクノロジー](#) 記事元:  POSTD

※現在はなぜか削除済み・・・

元の記事

Enough with the microservices

May 20, 2017

Share this: [twitter](#) // [facebook](#) // [linkedin](#) // [google+](#)

Tl;dr:

Don't even consider microservices unless you have a system that's too complex to manage as a monolith. The majority of software systems should be built as a single monolithic application. Do pay attention to good modularity within that monolith, but don't try to separate it into separate services.

– Martin Fowler

– Enough with the microservices | Adam Drake

<https://aadrake.com/posts/2017-05-20-enough-with-the-microservices.html>

記事の主張:

スタートアップ企業のほとんどは、
マイクロサービスを採用すべきではない。

記事の主張

- ・ 主張

- ・ **スタートアップ企業のほとんどでは、マイクロサービスを採用すべきではない。**

- ・ 理由

- ・ マイクロサービスを採用する理由として挙げられる課題に対しては、たいてい、先に考えるべき他の解決方法がある。
- ・ マイクロサービスを採用するには、技術的に前提となる条件が存在する。それをスタートアップのフェーズで得ることは難しい。
- ・ マイクロサービスは、本質的に複雑さを生むので、スタートアップのフェーズで対処するのは難しい。

感想

- その通りだと思います。
- マイクロサービスはモジュールの境界が不明瞭なうちにやると失敗しやすい。
- テスト、ロギング、監視などをきちんと行うのが、モノリシックに比べて大変。
- 分散システムは本質的に複雑。

銀の弾丸はない

- 糸冬 -

…もう少し詳しく。

- 記事の主張のうち、最も引用が多かった部分について、精査してみる。
- 「チーム間の依存性」という課題に対しての解決策として、いきなりマイクロサービスを導入するのはうまくいかないという話

“Don’t confuse decoupling with
distribution.”

– Enough with the microservices | Adam Drake

記事の主張

- 組織が大きくなってくると、
複数のチームが一つのコードベースで作業するようになる。
- チーム間の依存度(コミュニケーションコスト)を下げるため、
マイクロサービスにしようとする例がある。
 - しかし、マイクロサービスにしたとしてもコミュニケーションの問題が
解決するわけではない。
- 実際には、**モノリス内のモジュール分割**で十分対処できる。

丁寧に考察してみる

- 「チーム」とは何なのか？
- なぜチームが大きくなるとチームを分割するのか？
- 「チーム間の依存度」とは何なのか？
「チーム間の依存度」が高いと何が問題なのか？

「チーム」とは何なのか？

- ある目的のために協力して行動するグループ (広辞苑)
- その目的に向かって、チームは自主的・自律的に動けるべき
- 多くのコミュニケーションや決定は、チーム内で完結するべき

チームが大きくなると

- コミュニケーションのパスが多くなる
- チームの目的とメンバーの役割が遠くなり、自律性が下がる

チームを分割する

- 大きな目的を、より細かい目的に分解する
- 細かい目的に合わせて、小さなチームに分解する
- しかし、大きいチームの問題点を解消できるかは、この分解の方法による
- 悪い分解の例: 目的同士がコンフリクトする

悪い分割の例: Dev / Ops

- 開発と運用は、目的がコンフリクトする
- 本来は、開発と運用を合わせてビジネス上の目的を達成するものである

「チームの依存度」とは？

- 各チームが完全に**独立して意思決定したり行動できる**状態は、チームの依存度がゼロ
- 実際には、一定以上のコミュニケーションが必要
 - 協力、利害関係の調整など
- 依存度が少ないほど、チームが自律的に動ける

使命型組織/機能型組織

- 組織は、2つの典型的な形態に分けられる。完全な"使命中心"の形態と"機能別"編成形態である。
- 会社の大部分を使命中心形態に組織化する長所は(略)、個々の集団や単位が、絶えず自分の事業あるいは製品分野に対するニーズと接触を保ち、こうしたニーズの変化に対して迅速に対応できるという点"だけ"である。
- しかし、どんな事業でも、その本務は環境からの需要とニーズに応えることであり、この即応できるか否かがきわめて重要なカギである。

アンドリュー・S・グローブ著
小林 薫訳
ベン・ホロウィッツ 序文

人を育て、
成果を最大にする
マネジメント

HIGH
ハイアウトプット マネジメント
OUTPUT
MANAGEMENT

日経BP社

– アンドリュー・S・グローブ
HIGH OUTPUT MANAGEMENT
8章 ハイブリッド組織

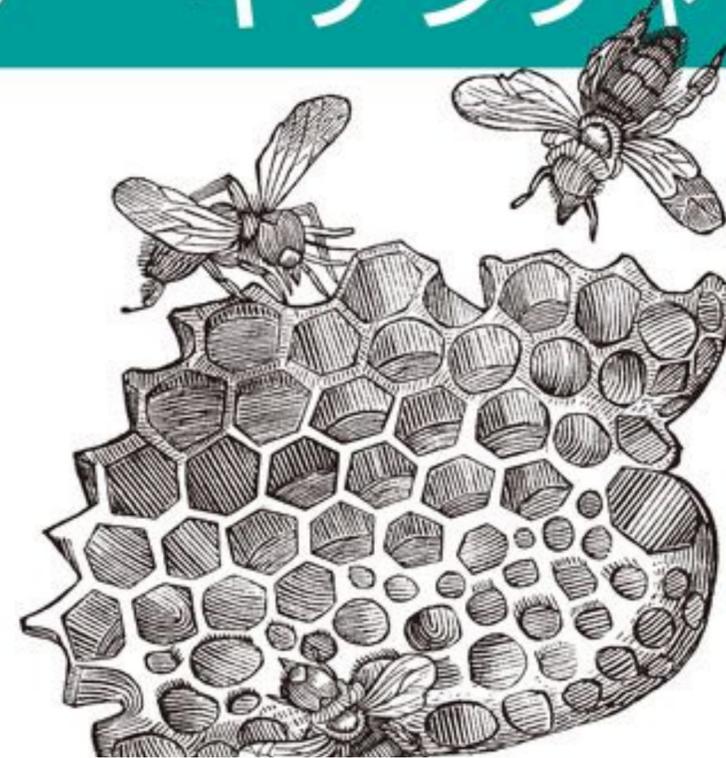
使命型組織/機能型組織

- 使命中心の組織
- 使命 \Leftrightarrow ビジネス
 - ※ 技術そのものがビジネスの組織もある
- 垂直分割
- e.g. FiNCアプリチーム
- 機能中心の組織
- 機能 \Leftrightarrow 技術
- 水平分割
- e.g. サーバーサイドチーム

マイクロサービス=使命型

"マイクロサービスは主に**ビジネスドメインに基づいてモデル化**することで、従来の階層型アーキテクチャの問題を避けています。"

マイクロサービス
アーキテクチャ

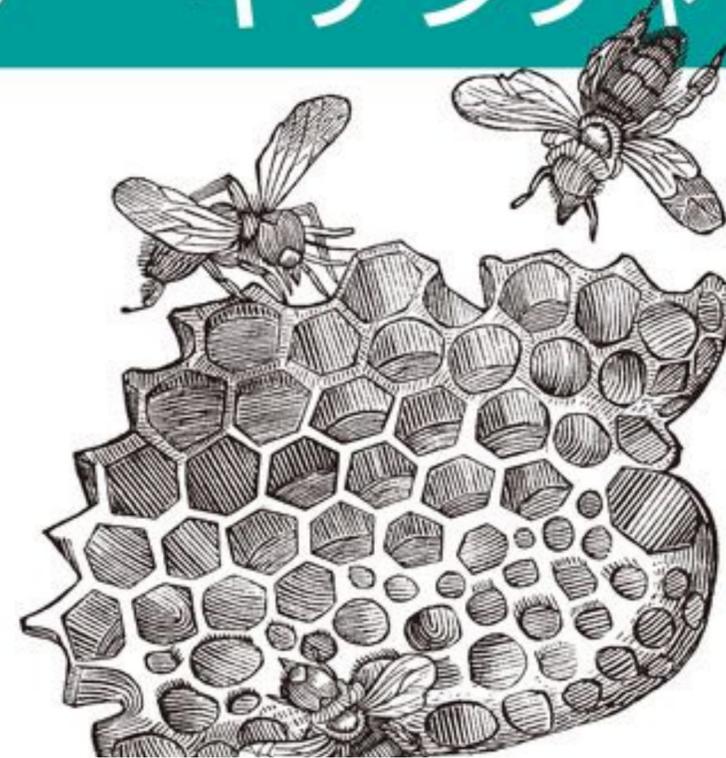


– マイクロサービスアーキテクチャ, はじめに
Sam Newman 著、佐藤 直生 監訳、木下 哲也 訳

マイクロサービス=自律的

"マイクロサービスは、協調して動作する小規模で**自律的**なサービスです。"

マイクロサービス
アーキテクチャ



- マイクロサービスアーキテクチャ, 1章
Sam Newman 著、佐藤 直生 監訳、木下 哲也 訳

マイクロサービスと組織

- 組織構造とマイクロサービスは、完全にアナロジーの関係にある
 - 組織の独立性 \Leftrightarrow デプロイの独立性
 - 組織同士の連携 \Leftrightarrow API連携 / イベント連携
- 組織同士が密結合なら、
それに沿ってマイクロサービス化してもやはり密結合になる

記事の主張(再掲)

- 組織が大きくなると、**複数のチームが一つのコードベース**で作業するようになる。
- チーム間の依存度(コミュニケーションコスト)を下げるため、マイクロサービスにしようとする例がある。
- しかし実際には、**モノリス内のモジュール分割**で十分対処できる。
- また、マイクロサービスにしたとしてもコミュニケーションの問題が解決するわけではない。

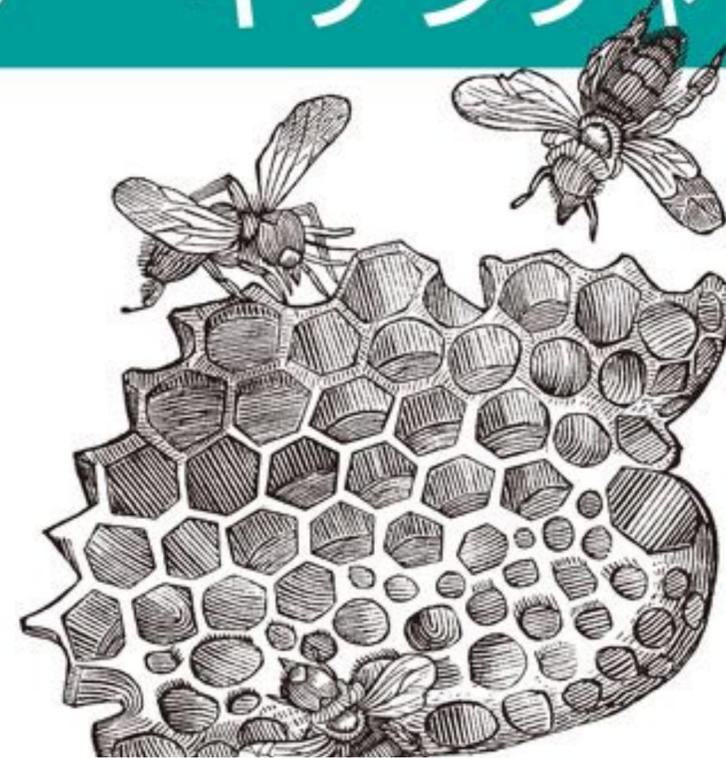
スタートアップと組織

- 多くのスタートアップでは、組織をきれいに分割することが難しい
 - 目的の分解 (戦略) が十分でない
 - ころころ変わる
- だからマイクロサービスに分割するのも難しい
- では、**最初はモノリスのままモジュール化だけすべきなのか？**

1つの反論 (引用による)

技術的には、適切に分解された独立したモジュールを1つのモノリシックプロセス内に作成することは可能なはずですが。しかし、まだほとんど登場していません。**モジュールはすぐに残りのコードと密結合になり、主な利点の1つを放棄することになります。プロセス境界で分離すると、この点で正しい状態が強制されます** (少なくとも間違ったことをしにくくなります)。もちろん、主にこのことによって、プロセス分離を推進すべきだとは言いませんが、**実世界ではプロセス境界内でのモジュール分離の約束が守られたことはほとんどありません。**

マイクロサービス
アーキテクチャ



– マイクロサービスアーキテクチャ, 1.4.2章
Sam Newman 著、佐藤 直生 監訳、木下 哲也 訳

スタートアップと組織

- 巨大なモノリスになってしまうと、やはり分割することが難しい
- マイクロサービスに分けなくても、普段からドメインを意識する
- 明らかに明確なドメインができたタイミングで、積極的にマイクロサービスに分割していく
- バランス感覚が重要

総論

- "スタートアップ企業のほとんどでは、マイクロサービスを採用すべきではない。" は、たぶん正しい。
- しかし、スタートアップにも状況やフェーズがある。
- すぐにマイクロサービスに向かわなくても、**「マイクロサービス精神」**を持つことは大事なのではないか。