

[InternetWeek 2019]

S12 サービス設計とリリース
～既存インフラの新陳代謝とリリースの悩み～

とある地域事業者の事例

2019/11/28

株式会社グローバルネットコア

金子 康行

<yasuyuki.kaneko@global-netcore.jp>

目次

- **導入**
 - 自己紹介、自社の状況
- **本編**
 - 自社の事例、過去の失敗やアンチパターンを添えて
- **まとめ**
 - 私がお伝えしたかったこと

導入

自己紹介、自社の状況

自己紹介

- **株式会社グローバルネットコア (AS18070)**
取締役 (本業)
- **株式会社データドック (AS131914)**
技術本部 (お手伝い)
- **越後ネットワーク・オペレーターズ・グループ (ENOG)**
運営委員 (創設メンバー)
- **日本ネットワーク・オペレーターズ・グループ (JANOG)**
運営委員



自社の事業

トータルインターネットソリューション
株式会社 **グローバルネットコア**
Global Network Core Co.,Ltd.

- **株式会社グローバルネットコア**

1998年設立、新潟県新潟市に所在、社員数95名

<https://www.global-netcore.jp/>



N-plus
Total Internet Solution

- **「N-plus」ブランドでインフラサービス全般を提供**

インターネット接続、専用線接続、データセンターフルマネージドサーバホスティング、AWS運用代行ウェブシステム開発、セキュリティ関連サービス

<https://www.nplus-net.jp/>

自社の事業



• サービスの特徴

基本的に法人向けサービスに特化

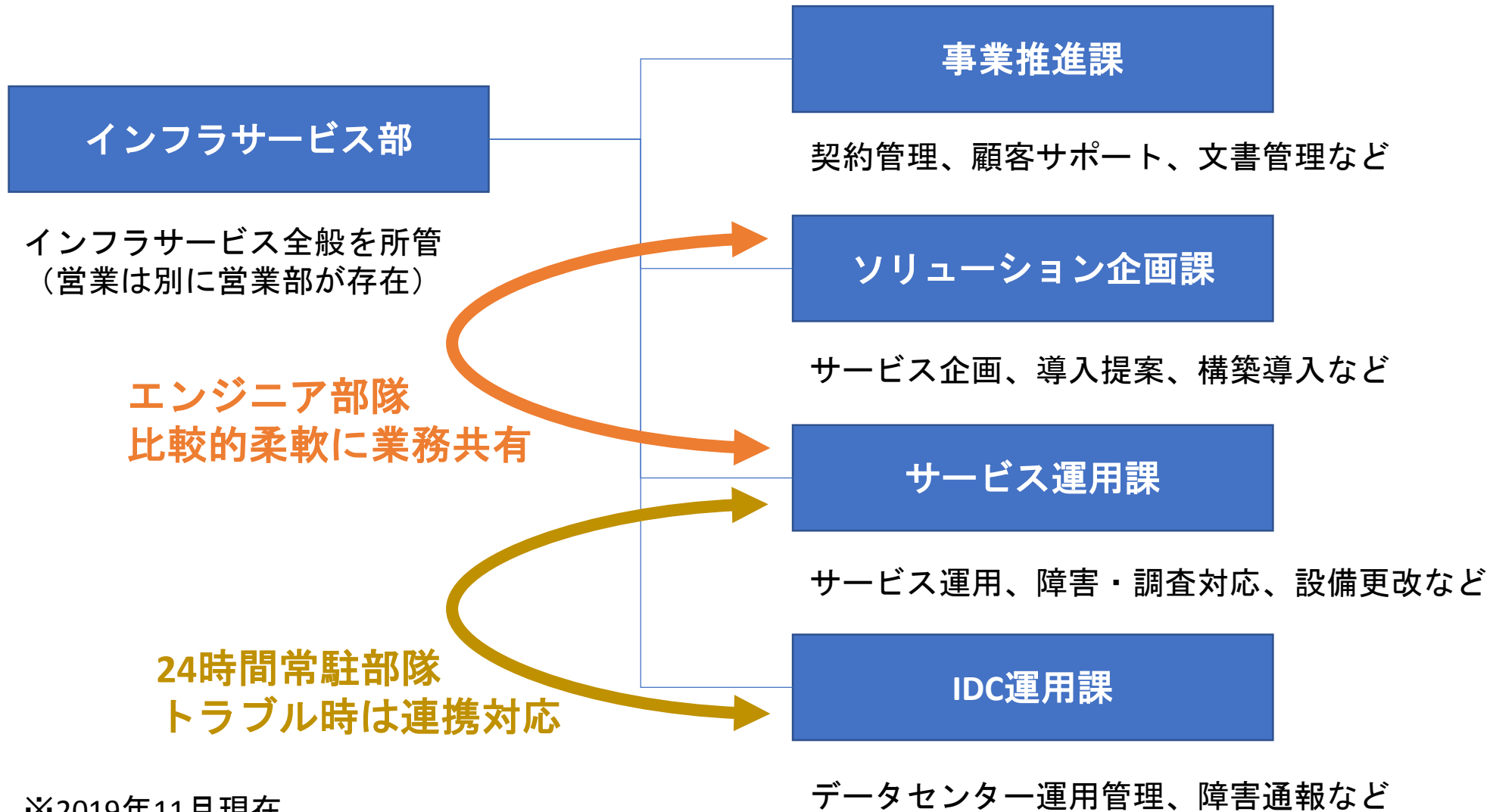
主要な顧客は新潟県内の企業、自治体、各種団体

インフラも、システムも、ウェブデザインも、一貫した体制

導入後から運用保守まで、顔が見えるきめ細かい対応

マス（一律多量）ではなく、ニッチ（少量多品種）なサービス
フルマネージドスタイルが主体、その運用は大層手間がかかる

自社の運用体制



※2019年11月現在

本編

自社の事例、過去の失敗やアンチパターンを添えて

これから話す内容

- **どのようにサービス設計・リリースをしているのか？**
設計→調達→検証→作業計画→ドキュメント
考え方ややり方は、事業者やサービスによってきっと違う
- **一つの事例として、当社の場合を話せる範囲で**
留意しているポイントなどを紹介
失敗やアンチパターンも織り交ぜながら

サービス設計

- **いきなりゼロから大規模にサービス設計をすることは稀**
 - 一品もののサービスを徐々に汎用化する
 - 既存サービスのオプションとしてリリースする
 - 既存サービスの課題を乗り越えるために別仕様として再設計

小さく試しながら作り込み、広げていく
リリース後でも、仕様の微修正はありうる

→ドキュメント化と関係先への共有を怠るべからず

サービスシステムの設計

- **基本的に自社で設計・検証・構築を実施**

小規模事業ゆえに、他社に委託することはほとんどない
必然的に、技術選定・機器選定はコンサバティブに

「大きなことはできませんが、小さなことからこつこつと」

© 西川きよし師匠

**外注を避けた結果、身の丈にあった技術と設計になった
ブラックボックス化せず、システムの新陳代謝もやりやすく
→自ら手を動かすことで、技術者の教育・育成に繋げる**

調達

- 候補機材は自社で選定、その上でベンダーに提案を依頼
必要要件と課題を明確にし、妥当性をチェックする
転ばぬ先の杖、思わぬトラブルに欠かせぬベンダーサポート
価格よりも技術対応力と営業対応力の評価を心がける

ベンダーのサポート力が自社構築のリスクヘッジになる
信頼できるベンダーを見つけ、良好な関係を構築する
→適度な余裕と緊張感、謙虚・尊敬・信頼を大切に

検証

- **基本的な検証は行うが、自社で実施できるレベルにとどまる**
少量の実機を使った機能試験や手順試験が中心
実環境と同等の検証環境、高価な試験機材の用意は困難
機材の単体性能はメーカーのデータシートを信じる

検証はもちろん必要だが、費用と時間に制約される
自社の実情に合致したサービスレベルと検証深度を見定める
→不安に感じている部分があるなら、手を抜かず検証を

作業計画

- **作業計画書を作成、レビューと実施可否判定を実施**
作業内容のリスクが高いものは承認プロセスが必要
目的、概要、構成、影響、手順、確認項目などをチェック
ノウハウを持つメンバーをレビューに巻き込む

**重要なのは投入コマンドの完璧性ではなくリスクの発見
過去の類似作業や障害の履歴も事前に洗い直しておく**

→最も恐れるのは確認漏れによる障害の見過ごしや対応遅れ

ドキュメント

- チームで運用保守することを念頭においた情報共有
概要説明資料を作成、部内説明会を開催して理解を促す
作業マニュアルwikiと別に、ノウハウを記入するwikiを作成
仕様ブレを無くすため、詳細なサービス仕様書を作成

ドキュメント作成・保守は苦手な人が多い（当社も道半ば）
受け入れ時のチェックとともに、定期的な更新が必要

→ドキュメントは他者への思いやり、未来の自分を救うもの

まとめ

私がお伝えしたかったこと

大事なことは

- 身の丈にあった技術と設計を心がける
- 自ら設計し、自ら構築する、基本的に外注しない
- 信頼できるベンダーと付き合う、値段で決めてはいけない
- 人材確保・育成は永遠の課題

おしまい

ありがとうございました