JANOG-42 Mie

その運用自動化では行き詰まる
～「つながらない」「つたわらない」「つみあがらない」を防ぐために～

波田野 裕一　運用設計ラボ合同会社
長久 勝　　国立情報学研究所
今井 祐二　　株式会社富士通研究所
畠山 慎平　　エヌ・ティ・ティ・コミュニケーションズ株式会社

- JANOGで、自動化ネタ定着
  - JANOG39: Ansibleことはじめ, ネットワーク運用自動化BoF
  - JANOG42: Ansibleチュートリアル, ハッカソン

技術トレンド把握から
スキル獲得 & 実装へ

- つぶらな瞳で問答

Q. 運用自動化、そんなに上手くいってますか？

A. 先達曰く、

「自動化は、**1周まわした先に**、違う景色が待っている」

「自動化は、**してからが勝負**」

それJANOGで聞きたい。

運用自動化に向け、転ばぬコツを共有しよう

本日のフォーカス

自動化した運用を「つなげる、つたえる、つみあげる」方法

運用自動化に向け、~~転ばぬ~~コツを共有しよう

ちゃんと転んでおく

本日のフォーカス

自動化した運用を「つなげる、つたえる、つみあげる」方法

お話いただくかた

- 波田野 裕一(運用設計ラボ合同会社)
  - ADSL ISP, EC企業での運用実務
  - JAWS-UG CLI専門支部(支部長)、AWS Samurai 2017, MSP協会特別会員
  - 運用自動化、不都合な真実 @ ssmjp201712 はたのさん祭

AWS企業システムに強い
運用設計Samurai

- 長久 勝さん(国立情報学研究所)
  - インフラ構築・運用(ネットゲーム & アカデミッククラウド)
  - JANOG ネットワーク運用自動化BoF 主宰
  - Literate Computing for Reproducible Infrastructure の開発・公開
    (Jupyter Notebookで、自動化コードを埋め込んだ運用手順書が執筆可能。再現性の高い構築が実施可能)

アカデミックIT基盤工房の
職人頭・マイスター

- 畠山 慎平(エヌ・ティ・ティ・コミュニケーションズ株式会社)
  - Global ネットワーク運用従事
  - オペレーション自動化・高度化と全社への定着に取組み

グローバルICTプラットフォーマー
運用自動化の社内伝道師

- 事例・対応・教訓 (30 min)
  - 波田野さん、長久さん、畠山さん
- Q&A(25 min)
- まとめ(10 min)

- 第2部@懇親会

その運用自動化では行き詰まる

～「つながらない」「つたわらない」「つみあがらない」を防ぐために～

JANOG42 ミーティング

運用設計ラボ合同会社
シニアアーキテクト 波田野 裕一

2018-07-12

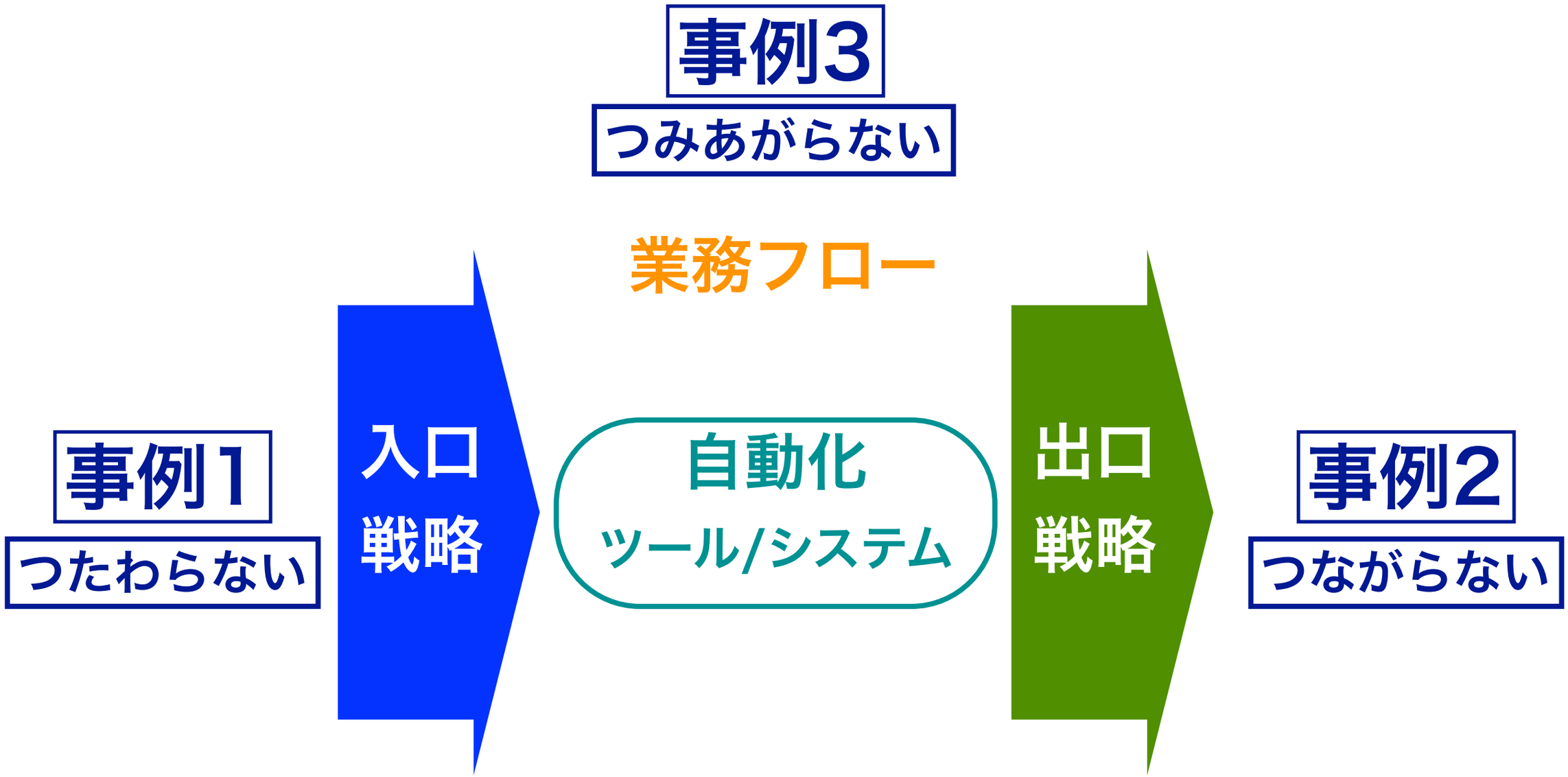
「いったん自動化してしまえば
あとは面倒を見る必要がない」

と思っていませんか?

そんなわきゃないです

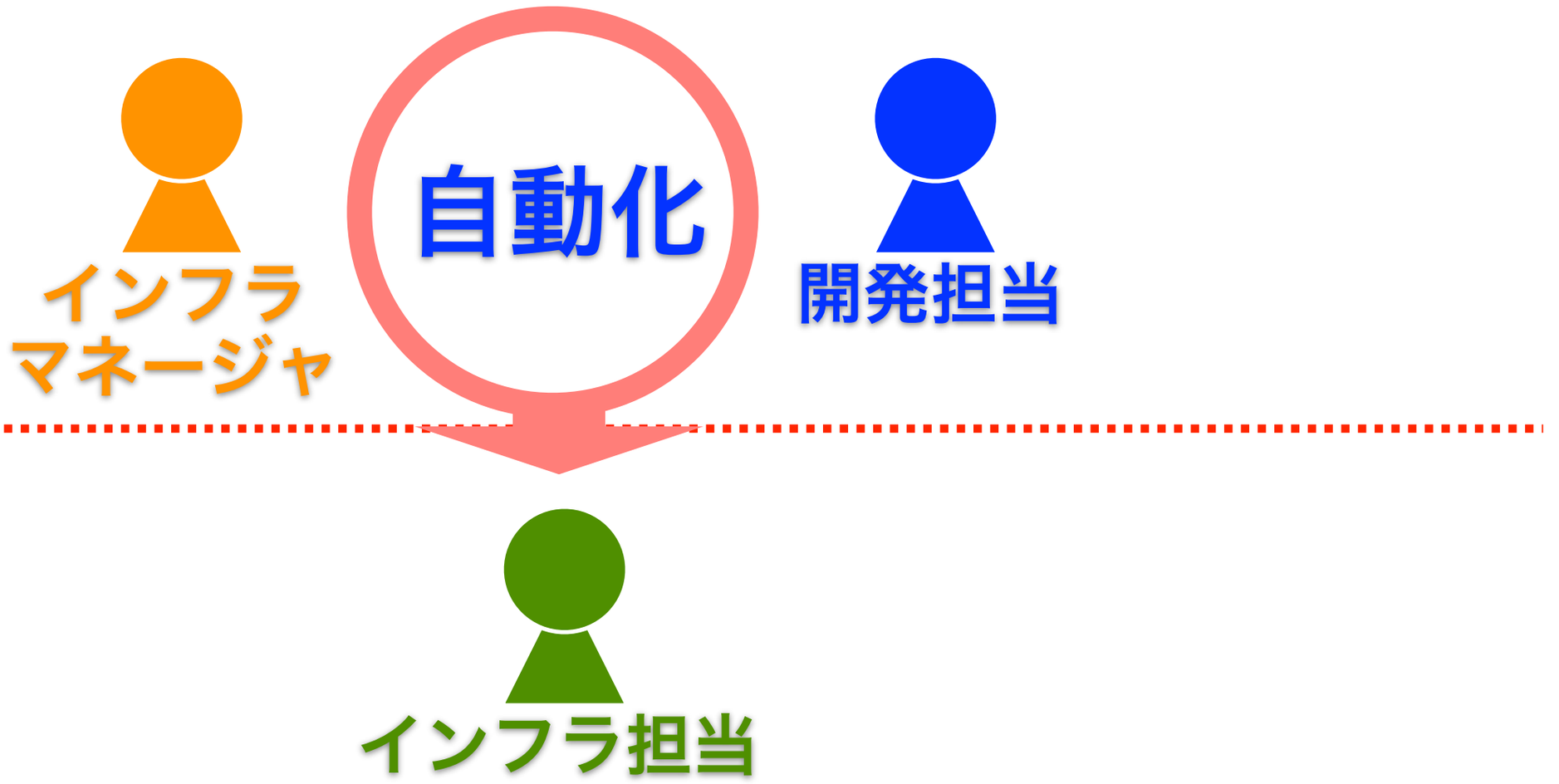
1. 事例

事例のマッピング


事例3
つみあがらない
業務フロー
事例1
つたわらない
入口
戦略
自動化
ツール/システム
出口
戦略
事例2
つながらない

つたわらない

事例1: 他人からの自動化引き継ぎ

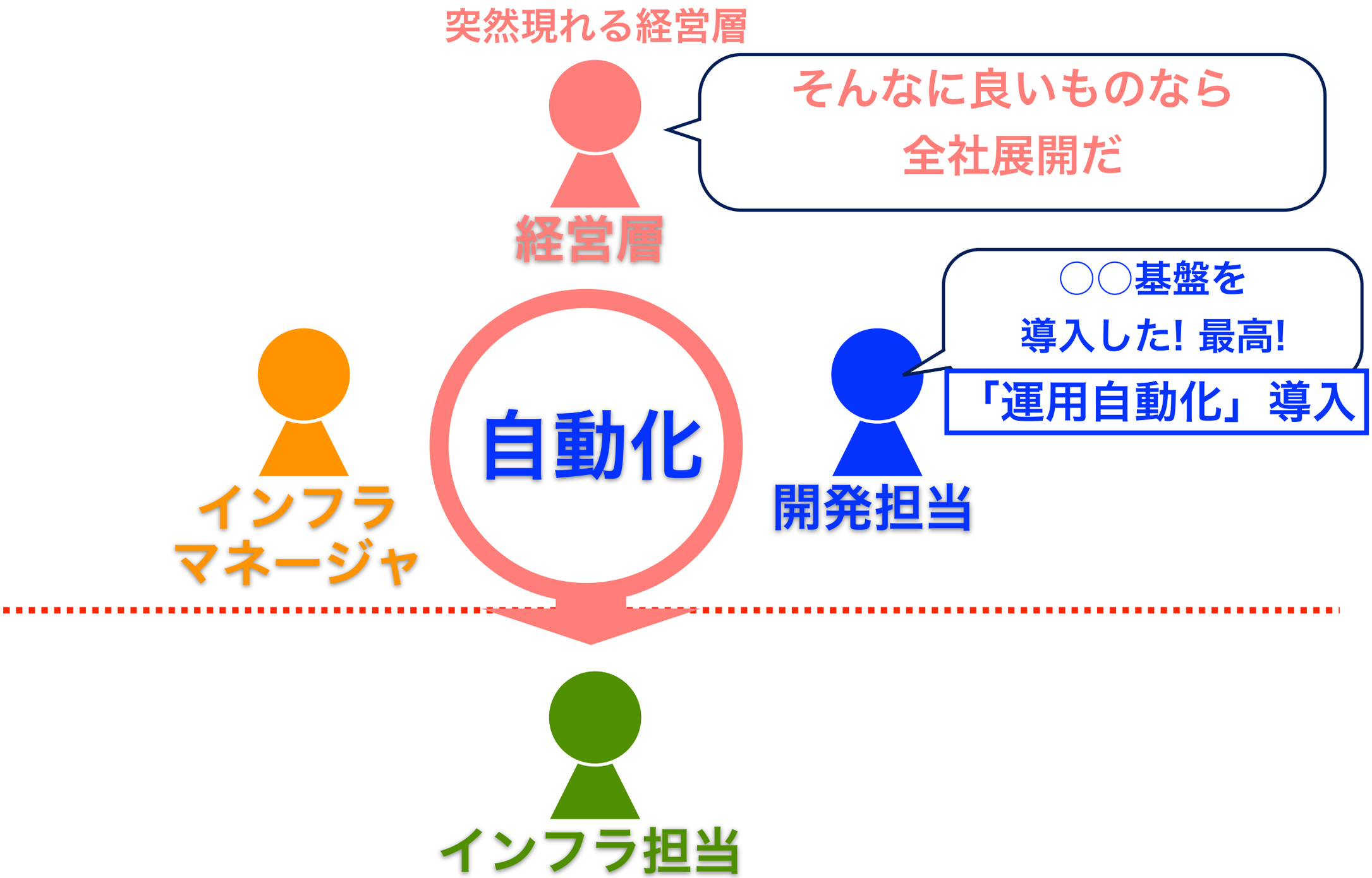


つたわらない

事例1: 他人からの自動化引き継ぎ

つたわらない

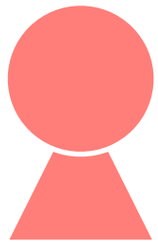
事例1: 他人からの自動化引き継ぎ



つたわらない

事例1: 他人からの自動化引き継ぎ

突然現れる経営層



そんなに良いものなら
全社展開だ

経営層

経営層に提案するマネージャ

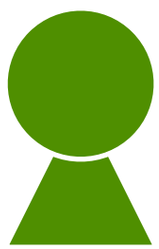
○○基盤は
うちで面倒見る?

インフラ
マネージャ

自動化

○○基盤を
導入した! 最高!
「運用自動化」導入

開発担当

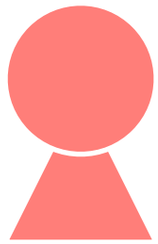


インフラ担当

つたわらない

事例1: 他人からの自動化引き継ぎ

突然現れる経営層



そんなに良いものなら
全社展開だ

経営層

経営層に提案するマネージャ

○○基盤は
うちで面倒見る?

インフラ
マネージャ

自動化

○○基盤を
導入した! 最高!
「運用自動化」導入

開発担当

動揺する現場

は?!

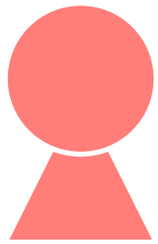
この時点で外堀は埋まってる

インフラ担当

つたわらない

事例1: 他人からの自動化引き継ぎ

突然現れる経営層



そんなに良いものなら
全社展開だ

経営層

経営層に提案するマネージャ

○○基盤は
うちで面倒見る?

インフラ
マネージャ

自動化

開発担当

○○基盤を
導入した! 最高!

「運用自動化」導入

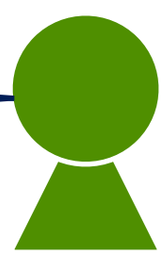
謎のカスタマイズ

つたわらない(伝承不可能)

経験なし
ベンダサポートなし
ドキュメントなし

動揺する現場

は?!



インフラ担当

この時点で外堀は埋まってる

つながらない

事例2: 他人への自動化引き継ぎ

実装した自動化の引継ぎ相手がいない

コードメンテナンス
できる人は皆無

自動化実装した人は
退職予定

半自動化済の業務

離職後に何かトラブったら終了

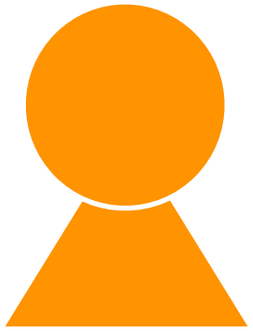
明日につながらない

つみあがらない

事例3: (人間に)高負荷な障害監視システム

自動通知とその対応に押しつぶされそうな日々

労力は消費されるが、業務成果はつみあがっていかない



監視チーム

数百のルール

数千台の物理ノード

数十万セルの
対応ルール表
(Excel)

アラート

監視システム

監視項目は数万

数百監視/秒

全員に全部通知
年10万通/人

メールはつみあがる

実例: 初JANOGスタッフ キックオフの夜

JANOG19 Meeting
- 開催概要
- プログラム
- ホスト/協賛

嬉し、はずかし、初対面の懇親会

実例: 初JANOGスタッフ キックオフの夜

JANOG19 Meeting
- 開催概要
- プログラム
- ホスト/協賛

嬉し、はずかし、初対面の懇親会

場が和みはじめた頃、突然鳴る会社携帯

実例: 初JANOGスタッフ キックオフの夜

JANOG19 Meeting
- 開催概要
- プログラム
- ホスト/協賛

嬉し、はずかし、初対面の懇親会

場が和みはじめた頃、突然鳴る会社携帯

「これはマジでヤバイかも」タクシーで向かう会社

実例: 初JANOGスタッフ キックオフの夜

JANOG19 Meeting
- 開催概要
- プログラム
- ホスト/協賛

嬉し、はずかし、初対面の懇親会

場が和みはじめた頃、突然鳴る会社携帯

「これはマジでヤバイかも」タクシーで向かう会社

復旧対応できる気がしねぇ

実例: 初JANOGスタッフ キックオフの夜

JANOG19 Meeting
- 開催概要
- プログラム
- ホスト/協賛

嬉し、はずかし、初対面の懇親会

場が和みはじめた頃、突然鳴る会社携帯

「これはマジでヤバイかも」タクシーで向かう会社

復旧対応できる気がしねぇ

結構、今でもトラウマな思い出

2. 対応と教訓

つたわらない → つたわる

事例1: 他人からの自動化引き継ぎ

対応1: 完全に作りなおし

可能な範囲で旧仕様を調査して、新仕様を起こす

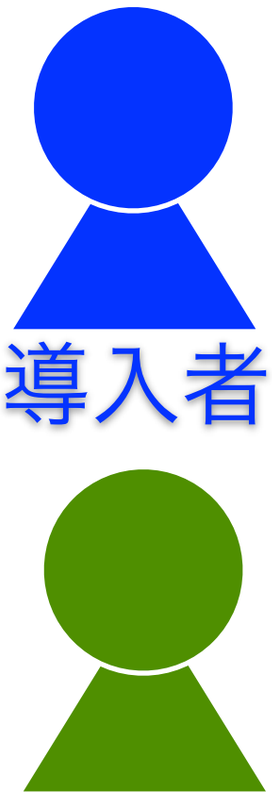
- 「自分を上手く使ってくれれば、役員の説得はきっちりやるよ」と言ってくれる新上司を最大限に活用。
- 利用者の声をとにかく拾った。(RFPは第5版まで改定)
- ベンダーのプロフェッショナルサポートを確保。

利用者や次の担当者向けのドキュメントも整備

つたわる

教訓1. 引き継ぎまでが自動化

自動化は「動作すれば完了」ではない。



導入者

動いたのであとはよろしく～

後任の人

泣きながら、徹夜で
仕様調査と運用設計

リバースエンジニアリングつらい…

もしくは
運用破綻

「後任の人にきちんとつたわる」のが大事

つながらない → つながる

事例2: 他人への自動化引き継ぎ

対応2: 引き継げる状態を作り出す

自動化された業務の全面手動化を実施

自動化前の業務がドキュメントとして残っていないため困難を極める

コードメンテナンス
できる人は皆無

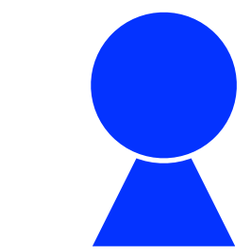
もう忘れているので
リバースエンジニアリング

顧客影響無く業務を継承してもらえる状態に

つながる

教訓2. 業務継承では手動化も選択肢

「自動化」は一方向だけではない (逆方向も想定すべき)



自動化(片方向)だけなら
(時間かければ)**誰でもできる**

手動化対応まで想定するのが
プロの自動化

自動化前の業務をドキュメントとして残しておかないと困難

引継ぎには、業務仕様ドキュメントが必要

(業務が明確なら手動化もできる)

対応3: 障害通知の構造化 &「振り分け」自動化

アラート総数が数百万件/年 -> 数万件/年に減少

年10万通/人　　年2000通/人

監視チーム

Step1

業務の構造化 (フローの整理)

ルールと通知先にID付与

ルールと通知先の棚卸・廃止

通知フローの整理

アラート

監視システム

Step2

自動化の実装

振り分け処理/データの実装

必要な人にだけ必要なアラート

想定外事象対応、EoL対応が随時発生

教訓3. 業務構造化がまず大事

「運用自動化」前の「業務の構造化」は必須

つみあがる

監視チーム

Step1

業務の構造化 (フローの整理)

- 業務構造化が自動化の起点
- 論理的に正しければ寿命が長い

自動化
システム

Step2

自動化の実装

- 完璧は無理、寿命は短かい
- たまにバグや隠れ仕様が牙を剥く

業務構造化した後の自動化は、確実に効果出る

Operation運用設計Lab

http://www.operation-lab.co.jp/

GakuNin Cloud

地獄のワンオペ自動化運用からの生還

2018年7月12日

国立情報学研究所　クラウド基盤研究開発センター
特任研究員　長久 勝

Literate Computing
for Reproducible Infrastructure

自己紹介

長久 勝（Masaru Nagaku）

国立情報学研究所（NII）
クラウド基盤研究開発センター　特任研究員

1994年から2010年まで、ゲームや映像配信などコンテンツ業界を中心に、技術職として、ソフトウェア開発（クライアント・サーバ）、システム構築・運用に従事。2010年から2015年まで、NIIにて、特任技術専門員として、トップエスイー（社会人向け教育プログラム）運営、所内クラウド基盤構築・運用に従事。2015年から2017年まで、オンラインゲームサーバソリューションの企画・開発・運用に従事。現職では、研究・教育分野でのクラウド利用促進の活動に従事。

https://researchmap.jp/mnagaku

自動化しないと死ぬ

- 概ね、現場に人は足りてないし、お金も足りてない
- コードだけは書ける

→この先生きのこるには「自動化」するしか

※選択肢はない、死ぬから

自動化しないと死ぬと思ったけど やっても死んだ（自動化黎明期編）

- 10年ぐらい前
- リアルタイムな端末挙動蓄積DB（50GBぐらい）のデータ移行
- システム開発：長久、システム運用：長久
- 平日深夜の負荷が少ない時間帯に、本番系DBを一時退避系DBに切り替え、本番系DBに旧系DBのデータコピーを一定時間実施、本番系DBに切り戻してから、始業時刻までに一時退避系DBのデータを本番系DBにコピー、とゆー作業が必要だった
- 本番系DBに負荷をかけると、端末からのデータ投入とコールセンタでのデータ参照に影響が出るので、深夜帯しか本格的に触れない
- 作業手順を策定し、一時退避系DBなどの準備を行い、夜間バッチとして自動実行できるphpスクリプトを書いた。
  システムのパフォーマンスを見ながら、随時、コピーするデータ量などを調整しつつ、夜間バッチを実施
- データ移行が終わるまで3か月ぐらいかかった
- その間、ずっと気にしながら、調整しながら

GakuNin Cloud

自動化しないと死ぬと思ったけど
やってる人は少ない

- 10年ぐらい前
-
- 平日深夜にデータコピーを一定時間
  実施、本番系に
- 本番系DB
  触
  作業手順を策定し、一時退
  システムのパフォーマンス
- データ移行が終わるまで
- その間、ずっと気にしながら、調整しながら

ワンオペ状態の時、
自動化しても、
自動化した人が
システムの奴隷になる

Literate Computing
for Reproducible Infrastructure

自動化しないと死ぬと思ったけど やっても死んだ（現代編）

- JenkinsのCI/CDスクリプト（シェルスクリプト）
- システムの開発・構築・運用を数人のチームで行う
- 出荷判定は負荷試験含めJenkinsで各種試験を回して判定
- 開発/ステージング/本番環境への配備もJenkins
- 各メンバの成果物をJenkinsでまとめて使う
- メンバが各担当部分で手一杯なので、Jenkins上のスクリプト開発がワンオペでブラックボックス化
- Wikiに技術情報を書くも、どのスクリプトの話がどのページに書かれているのか分かりにくい

GakuNin Cloud

自動化しないと死ぬと思ったけど やってこなかった理由

- Jenkinsの
- 自
- 開発/
- タ
- メンバが各担当 ワンオペでブラックボックス
- Wikiに技術情報を書くも、 プトの ページに書かれているのか分かりにくい

チームでも
普通に
ワンオペ化する

Literate Computing
for Reproducible Infrastructure

GakuNin Cloud

未来に、つながらない

みんなに、つたわらない

スペランカーかと思うぐらい

簡単に死ぬ

経験値が、つみあがらない

ワンオペ人の
孤独な末路

Literate Computing
for Reproducible Infrastructure

おお 長久！死んでしまうとは なにごとだ！

- 復活したら見慣れた前の職場だったw
- なぜか手には新しい武器が！

自動化しても死なないかも知れない（萌芽編）

- JenkinsのCI/CDスクリプト（シェルスクリプト）
- スクリプト開発をJupyterNotebookで分業して書けた
- 配備先環境のインフラ構築を行う前半部分と、コードをコンパイルして配備する後半部分を分業できた
- mdで書かれた説明文を手掛かりに前半部分を理解して、それに合わせた後半部分を書けた

GakuNin Cloud

自動化しても死なないかも知れない（萌芽編）

- Je
-

た
ドをコンパ

して、それ

これってもしかして、
俺たち・私たち、
ワンオペじゃ
なくなってるー！？

つたわる

Litera
for Reproducible Infrastructu

自動化しても死なないかも知れない（誕生編）

- LC4RIによる手順書の妥当性検証（2015年）
- 納品されたシステム構築手順書はNotebook
- 納品されたシステムを構築手順で組み上げれるか、機材を初期状態にして、Notebookを実行することで確認
- 引っかかったら修正を依頼する。検収作業がNotebookで行える

GakuNin Cloud

自動化しても死なないかも知れない（誕生編）

- LC4RIによる手順書の
-
- 機材を初
- bookで行え

動かない手順書に
絶望しなくていい世界

つたわる

Literate Computing
for Reproducible Infrastructure

自動化しても死なないかも知れない（未来へのバトン編）

- 障害調査の様な非定型な作業
- リードエンジニアがNotebookベースで調査・解決。何をやったか、それぞれの結果、参考にした情報（mdにメモ）などが残る
- 同様の障害が出た場合、前回のNotebookをメンバに渡して、復旧作業を依頼する
- リードエンジニアの作業をなぞりながら復旧作業を行うことで、技能移転が行える

自動化しても死なないかも知れない（未来へのバトン編）

- 障害調査の様な非定型な作業
- リードエンジニアがNotebookベースで調査・解決、何をやったか、それぞれの結果
- 同
- リードエンジ

安心して旅行に行ける

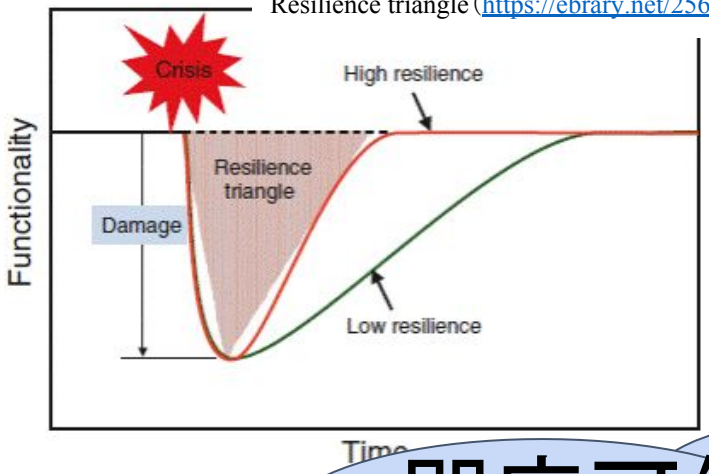
つながる

つみあがる

自動化しても死なないかも知れない（楽園編）

- Notebookベースで多能工化したチームは、対象システムの制御に自信を持てるようになる
- 対象システムの再構築すら可能なので、突発的な緊急事態が起きても、心に余裕（安心）を持つことができる

Resilience triangle (https://ebrary.net/25636/environment/resilience)



GakuNin Cloud

なないか

能工化したチー は、対象システムの制

- が

未来に
つながる

即応可能なチームの実現
⇒レジリエンスの獲得

みんなに
つたわる

経験値が
つみあがる

この先生きのこるには「機械化」だった

- 現代の情報システムは、開放系で変化していく
  - コード化しても賞味期限が分からない
  - 1回書いて終わりにならない

→機械化（Infrastructure as Code）≠ 自動化

- 機械化された作業はメンテされ続けなければならない
- 賞味期限が長く、何度も行う退屈な作業は、自動化OK
- 機械化した作業を、チームで共有し、多能工化

この先生きのこるには「機械化」だった

- 現代の情報システム
  - コード化
  - 1回書いて終わり

ワンオペは
「機械化」で
倒せる！

関連情報

- GitHub
  https://github.com/NII-cloud-operation
  https://literate-computing.github.io/index-ja.html
- Facebook
  https://www.facebook.com/groups/LiterateComputing/
- お問合せ
  mnagaku@nii.ac.jp
  @mnagaku

NII

大学共同利用機関法人 情報・システム研究機構

国立情報学研究所

National Institute of Informatics

Literate Computing

for Reproducible Infrastructure

JANOG42

NTTコミュニケーションズ株式会社
畠山　慎平

自己紹介

氏名：畠山　慎平（はたけやま　しんぺい）
所属：NTTコミュニケーションズ株式会社
経歴：Global NW運用 2年
　　オペレーション自動化・高度化 ５年
JANOG：始めて

＜キーメッセージ＞

- パンドラの箱を開ける覚悟、箱の底にある希望
- 現場主導での自動化
- 取り組みやすい自動化

■自動化の闇と希望

現状分析、あるべき姿の議論を重ねていくと
ヤバいところ、イケてないところが見えてくる
（プロセス、社内ルール・ポリシー、システム・NW等）
→パンドラの箱開け係
　→会社目線で見ると課題発掘役にも

課題を１つずつ解決していく。組織を超えて
→より良いオペレーションが具現化

■アメーバ化/運用主導の課題

ツール・システム・NWが現場に乱立
簡単なAccess等でツールを作成
当然仕様書なし
→忘れた頃にリバースエンジニアリングが
　待ってます！

もうちょっとスキルがあるRuby,Python…
→言語が書ける人が異動すると、
　メンテできないのは同じorz

■アメーバ化/運用主導の課題

「セキュリティ」って何それ？美味しいの？
→自動化するためのシステム・NWを作る基礎となる法・規定等への理解不足な場合もある

→一方、理解しようと試みても難易度高

→スピーディーな自動化の足かせ・諦めを産むハードルにもなりうるので、
ハードルを引き下げる・無くすことも重要

■組織としての自動化

組織として継続的に自動化をするために

・虎の巻
そもそもの取組姿勢、スタンスの可視化・共有
対応レベルの標準化（人が変わっても続けられる心得）

・Why（背景・理由）
ドキュメント等で経緯や目的を残すのが大事
仕様は実装を見れば理解できるが
Whyは残しておかないと後ではわからない

■組織としての自動化

・ツールの選択
各現場で個別のスクリプト作成等ではなく、
共通的な自動化ツールを採用
（プログラミングレス、ビジュアルモデリング）

→自動化へのハードルの引下げ
　引き継ぎやすい環境整備

NTT Communications
Transform. Transcend.

ご清聴ありがとうございました

最初の質問

３人とも「痛い目にあわないとわからない」とおっしゃった。

ほんとうに、痛い目にあわないとダメですか？

- 本質的に、経験しないと「つたわらない」知識か
- コミュニティに「つみあげる」ことは不可能か

まとめ： 最後にひとことづつ

まとめ

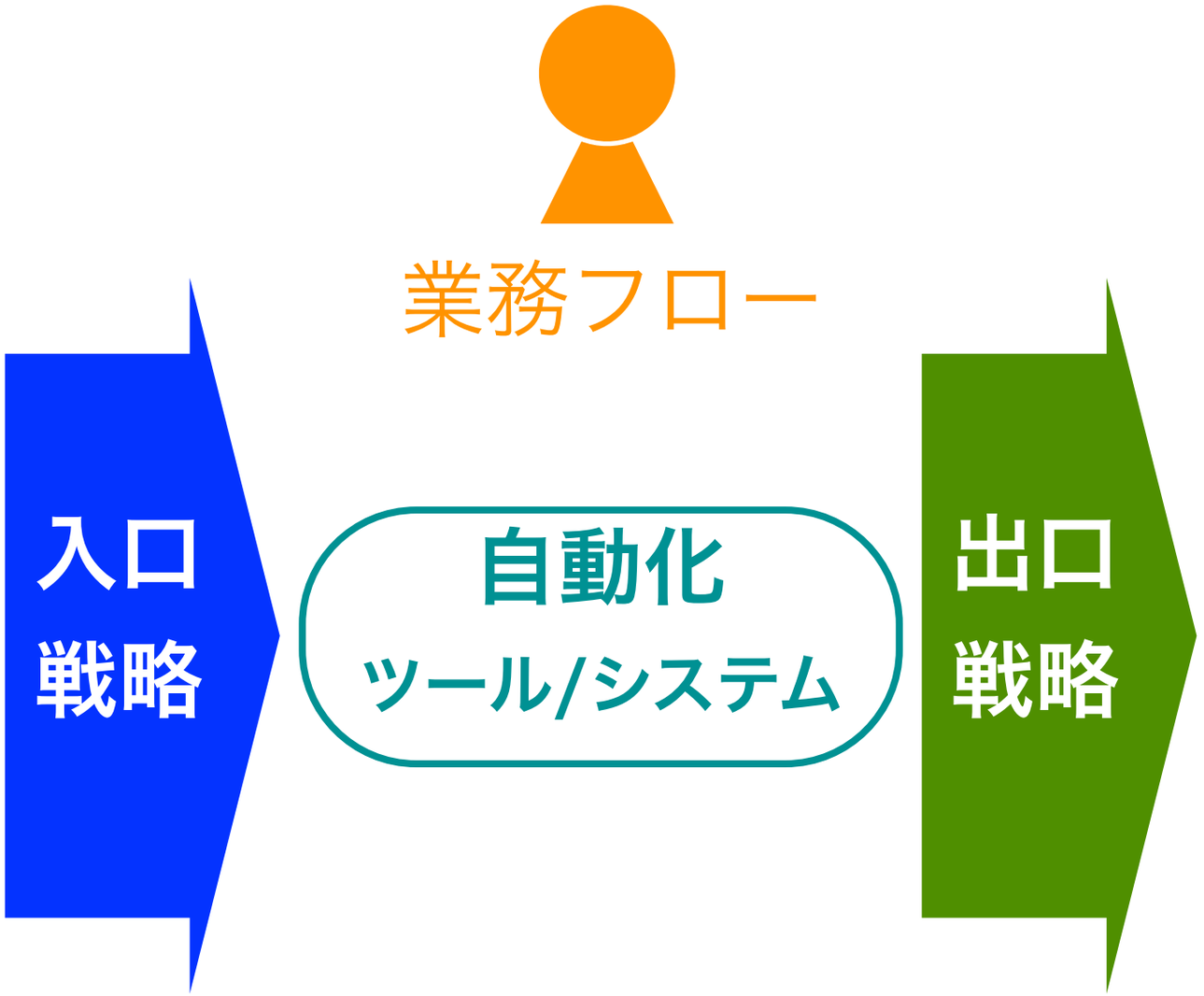
~~「いったん自動化してしまえば~~
~~あとは面倒を見る必要がない」~~

引き継ぎまで意識して

常に業務を構造化していく必要

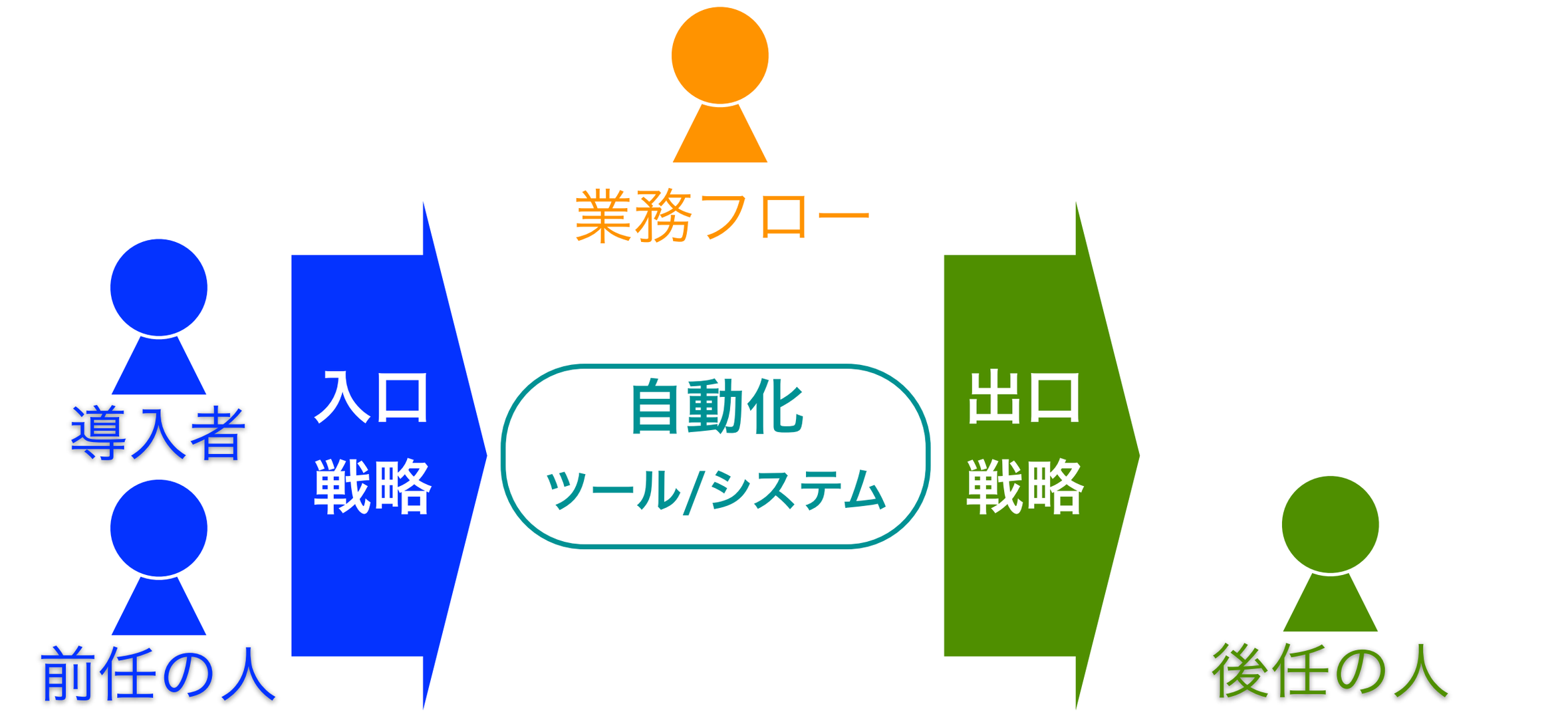
やらないと「運用自動化」は腐る

運用自動化、教訓のまとめ



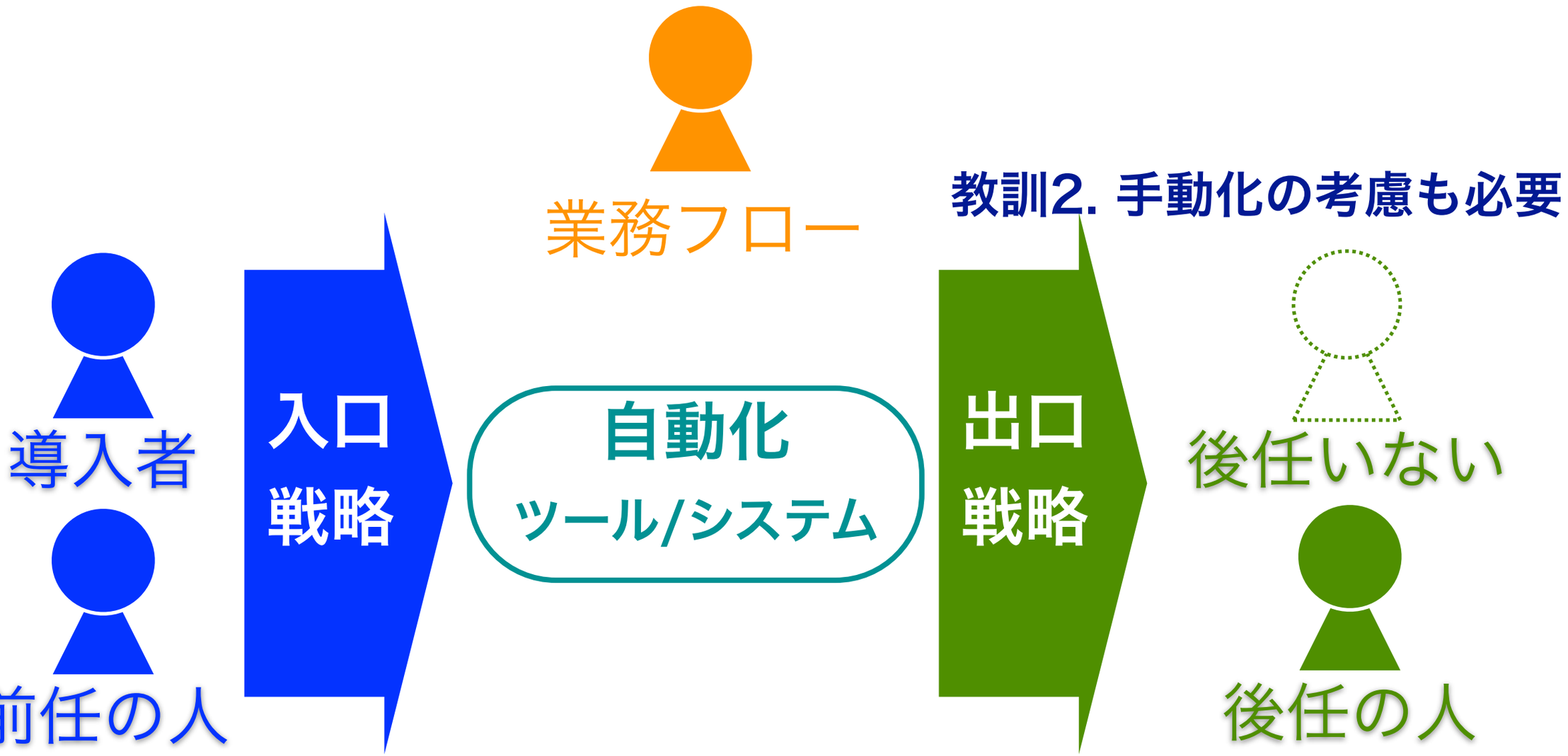
つたわる

運用自動化、教訓のまとめ



つたわる
つながる

運用自動化、教訓のまとめ

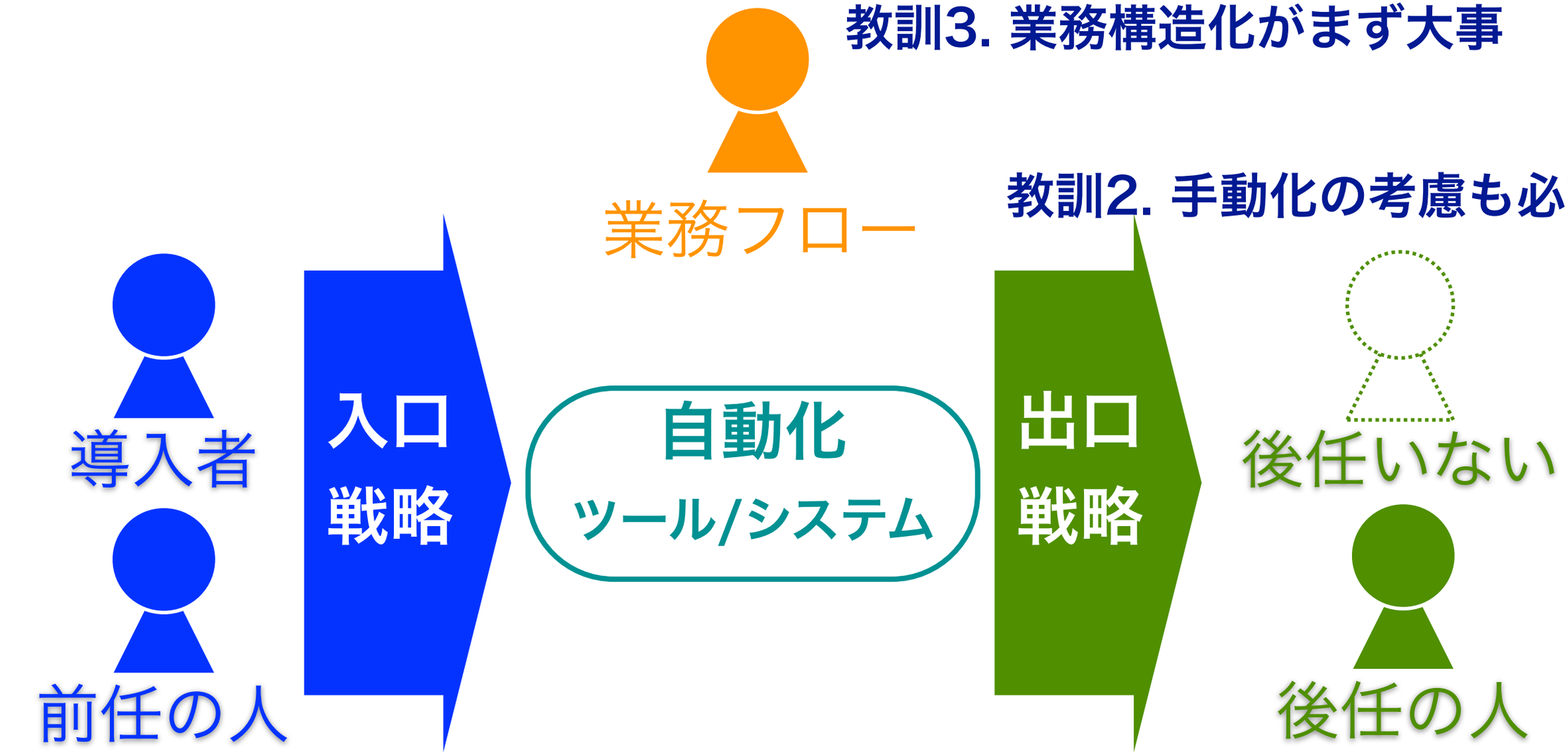


運用自動化、教訓のまとめ

つたわる
つながる
つみあがる

教訓3. 業務構造化がまず大事

教訓2. 手動化の考慮も必要



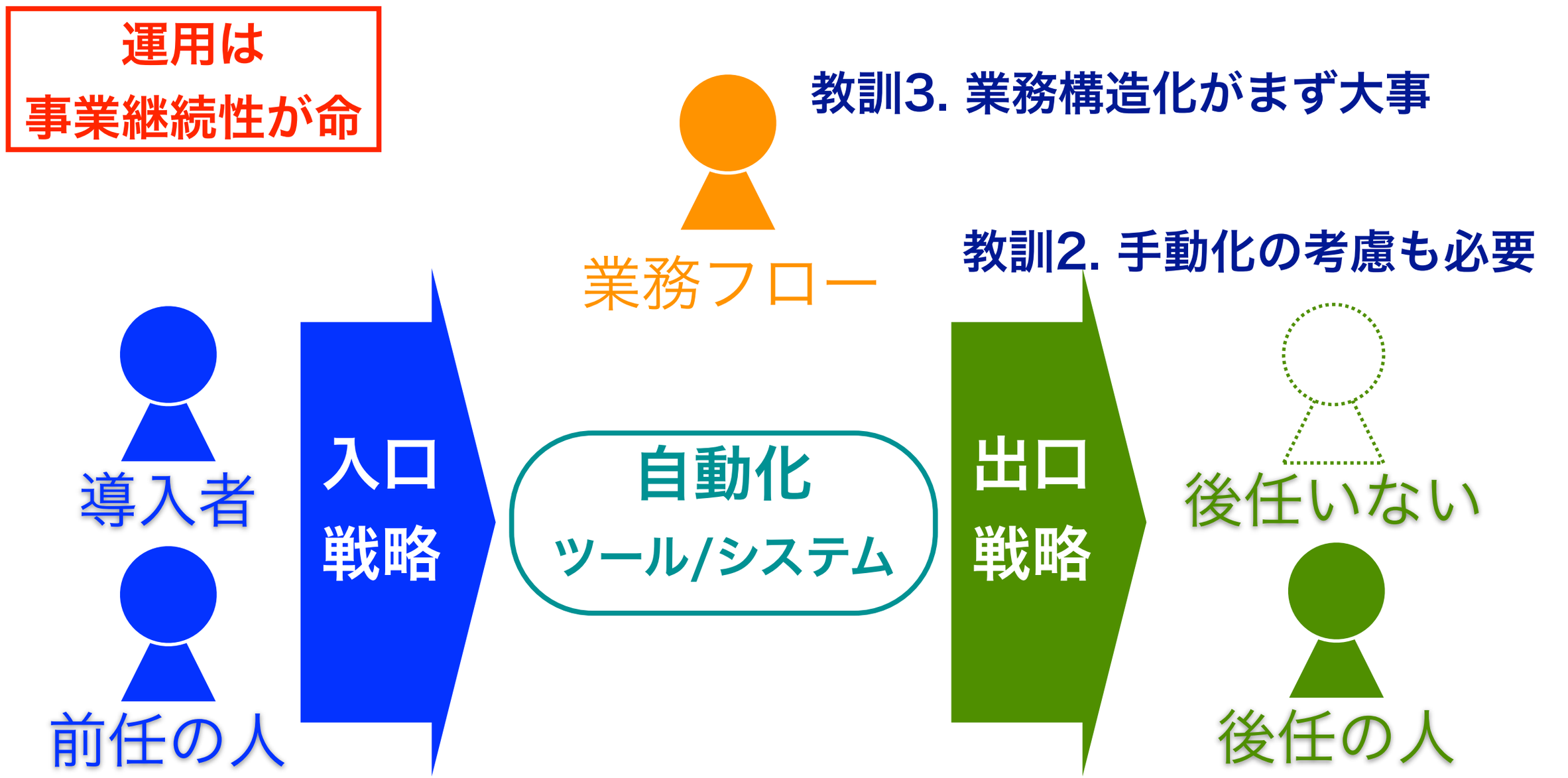
教訓1. 引き継ぎまでが自動化

教訓1. 引き継ぎまでが自動化

運用自動化、教訓のまとめ

つたわる
つながる
つみあがる

運用は
事業継続性が命



「運用自動化」で酷い目に合う人が減りますように(-人-)

自動化の先でつまづかないため、やっておくべきこと(長久より)

- 本番で事故ると致命傷になる
  - なるべく早いうちに転んで怪我しておく(通過儀礼)
  - PoC、β版の間に身体に刻み込んでおく
- ワンオペ(属人化)、自動化への依存は、人間の能力の限界(有限の注意力)から生まれるので、逃れられない
  - 低いコスト(良い道具で人間を拡張)で抗い続ける工夫
  - Someone on the member in the loop


BOGDEN
STEPIEN


BOGDEN
STEPIEN
RANGER

■自動化の先でつまづかない為に

- 早目に小規模のもので失敗をしておく
  →勘所を押さえる
- たまには手でやってみる
  →自動化のありがたみがわかる（利用者）
  　＆頑張ろうと思う（開発側）
- Whyを残すのが大事
- マインド醸成