

「効果が出る」 ネットワーク自動化の始め方

～課題設定と自動化のアプローチ～

2023-11-16

Tomoaki Nakajima

Red Hat

Introduction

こんな経験はありませんか？

自動化して効率化したい

こんな経験はありませんか？

自動化して効率化したい

作業のリストを作る

#	作業内容	回数/月	時間[h]	合計[h]
1	仮想マシンの払い出し	10	2	20
2	仮想マシンの再起動	8	0.5	4
3	XXXログ収集	6	0.5	3
4	YYYサービス再起動	6	0.5	3
5	ACL変更	1	3	3
6	ロードバランサーの閉塞・開放	1	2	2
7	FWルール変更	1	2	2
8	AAA作業	2	1	2
9	BBB作業	1	0.5	0.5
10	CCC作業	1	0.5	0.5
11	DDD作業	4	0.25	1
12	EEE作業	3	0.25	0.75
13	FFF作業	3	0.25	0.75
14	GGG作業	3	0.25	0.75
15	HHH作業	2	0.25	0.5
16	III作業	2	0.25	0.5
17	JJJ作業	2	0.25	0.5
18	KKK作業	1	0.25	0.25
19	LLL作業	1	0.25	0.25

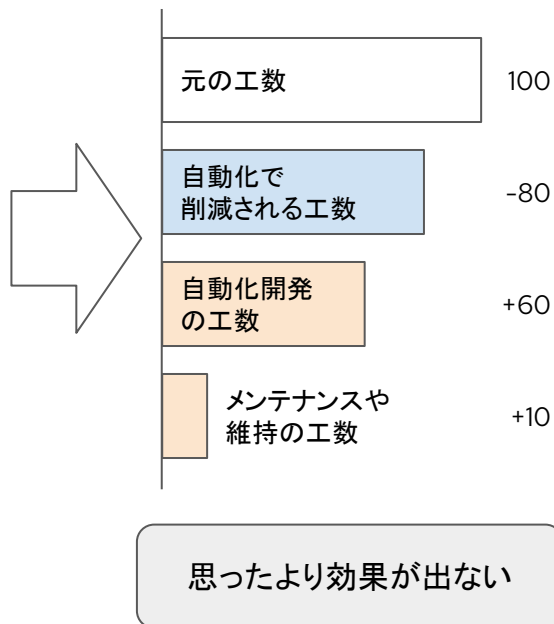
こんな経験はありませんか？

自動化して効率化したい

作業のリストを作る

#	作業内容	回数/月	時間[h]	合計[h]
1	仮想マシンの払い出し	10	2	20
2	仮想マシンの再起動	8	0.5	4
3	XXXログ収集	6	0.5	3
4	YYYサービス再起動	6	0.5	3
5	ACL変更	1	3	3
6	ロードバランサーの閉塞・開放	1	2	2
7	FWルール変更	1	2	2
8	AAA作業	2	1	2
9	BBB作業	1	0.5	0.5
10	CCC作業	1	0.5	0.5
11	DDD作業	4	0.25	1
12	EEE作業	3	0.25	0.75
13	FFF作業	3	0.25	0.75
14	GGG作業	3	0.25	0.75
15	HHH作業	2	0.25	0.5
16	III作業	2	0.25	0.5
17	JJJ作業	2	0.25	0.5
18	KKK作業	1	0.25	0.25
19	LLL作業	1	0.25	0.25

実際に試してみると



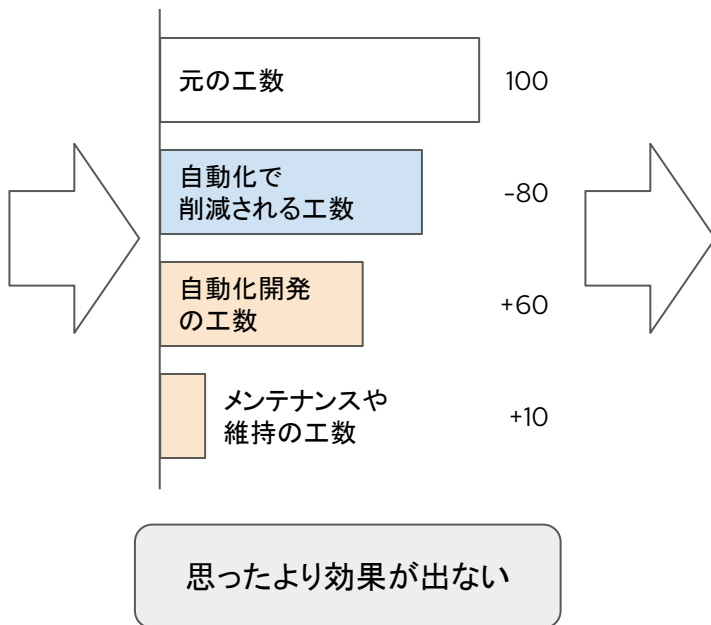
こんな経験はありませんか？

自動化して効率化したい

作業のリストを作る

#	作業内容	回数/月	時間[h]	合計[h]
1	仮想マシンの払い出し	10	2	20
2	仮想マシンの再起動	8	0.5	4
3	XXXログ収集	6	0.5	3
4	YYYサービス再起動	6	0.5	3
5	ACL変更	1	3	3
6	ロードバランサーの閉塞・開放	1	2	2
7	FWルール変更	1	2	2
8	AAA作業	2	1	2
9	BBB作業	1	0.5	0.5
10	CCC作業	1	0.5	0.5
11	DDD作業	4	0.25	1
12	EEE作業	3	0.25	0.75
13	FFF作業	3	0.25	0.75
14	GGG作業	3	0.25	0.75
15	HHH作業	2	0.25	0.5
16	III作業	2	0.25	0.5
17	JJJ作業	2	0.25	0.5
18	KKK作業	1	0.25	0.25
19	LLL作業	1	0.25	0.25

実際に試してみると

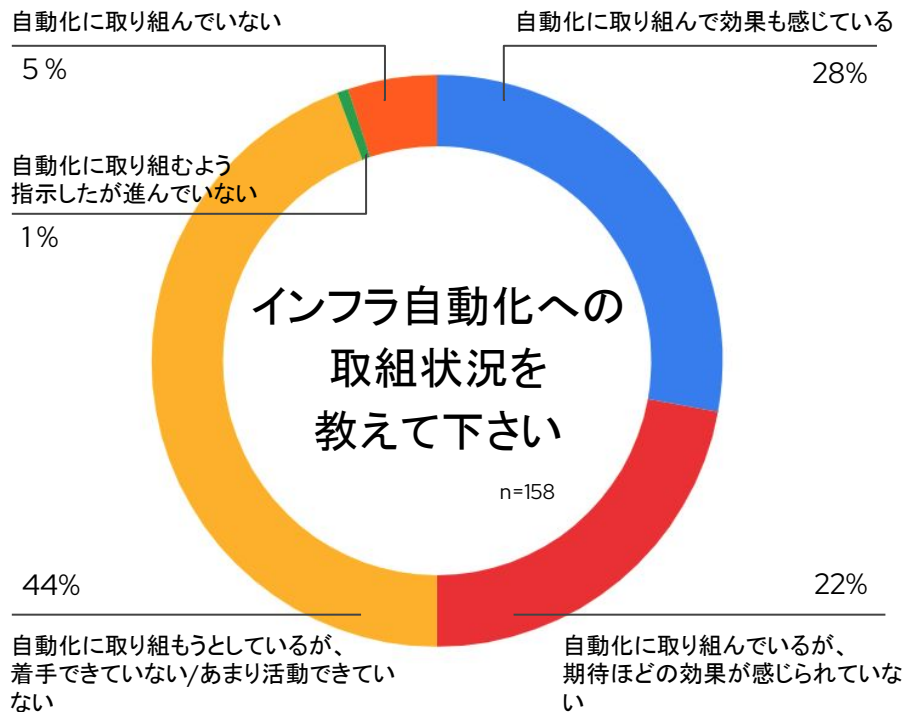


どうすればよいか？

- 同じ作業の繰り返しを増やせば効果を上げられる
- しかし手順はシステムごとにバラバラで同じ手順の繰り返しは少ない
- システム側を標準化することで効果を上げられる
- しかしシステムごとに個別最適化されており標準化することは難しい

世の中のインフラ自動化の状況

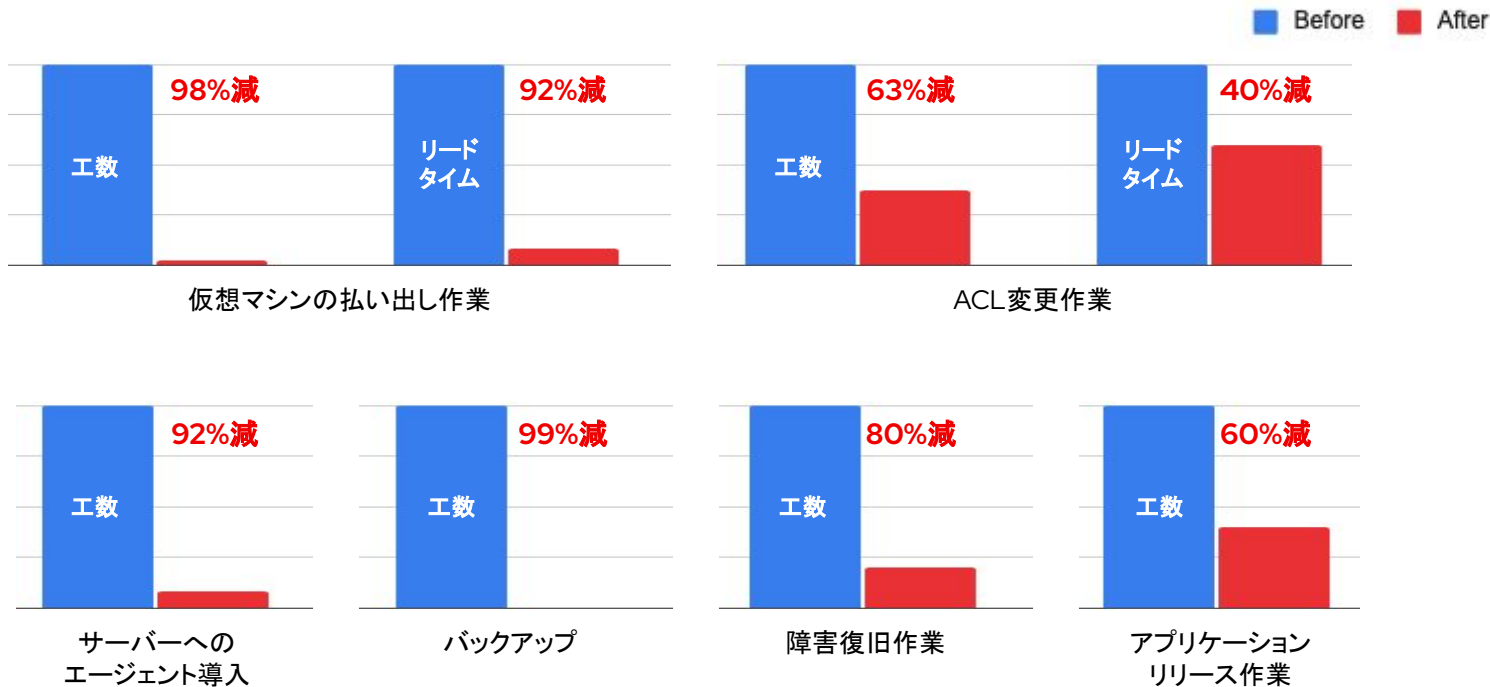
各社の自動化の取組状況



- ▶ ほぼ全ての企業が何らかの形で自動化には取り組んでいる
- ▶ しかし、効果を実感できているのは全体の3割弱に留まる
- ▶ 全体の約7割の企業は自動化の推進に課題を抱えている

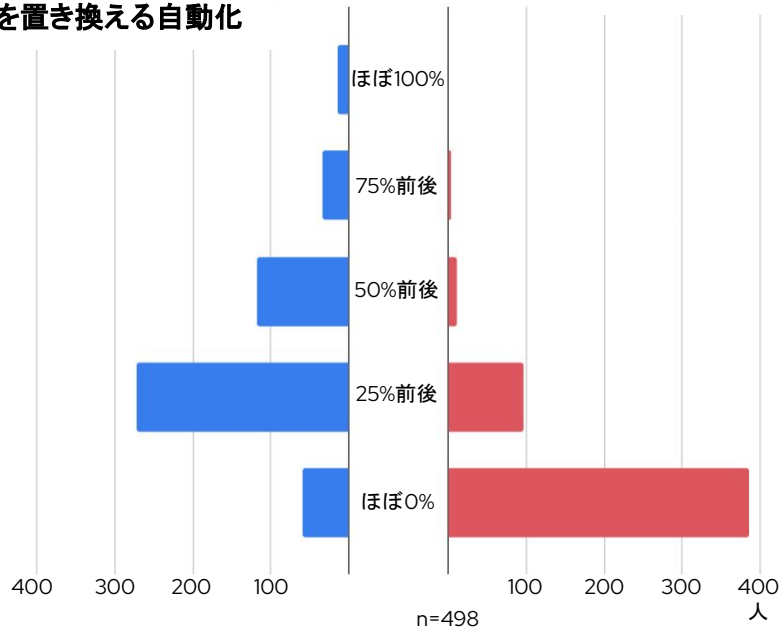
効果が出ている例

- 人の手作業だけでなく、その前に発生する調整や打合せを含めたプロセス全体での効果



各社の取り組んでいる「自動化」とは？

スクリプトやマクロで 手順を置き換える自動化



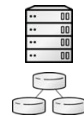
手順書



作業者



インフラ環境



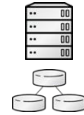
プログラム



ツール



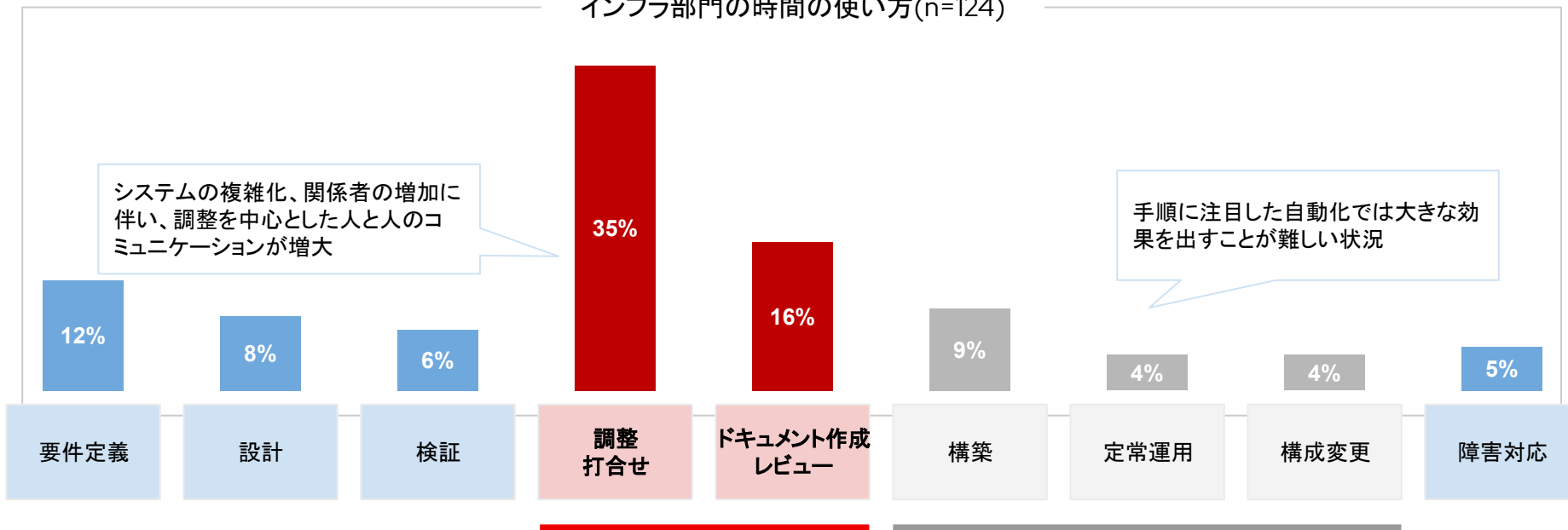
インフラ環境



工数の削減、品質の向上、リードタイム短縮を目的に、多くの(ほとんどの)企業において長年取り組まれてきた(取り組まれている)

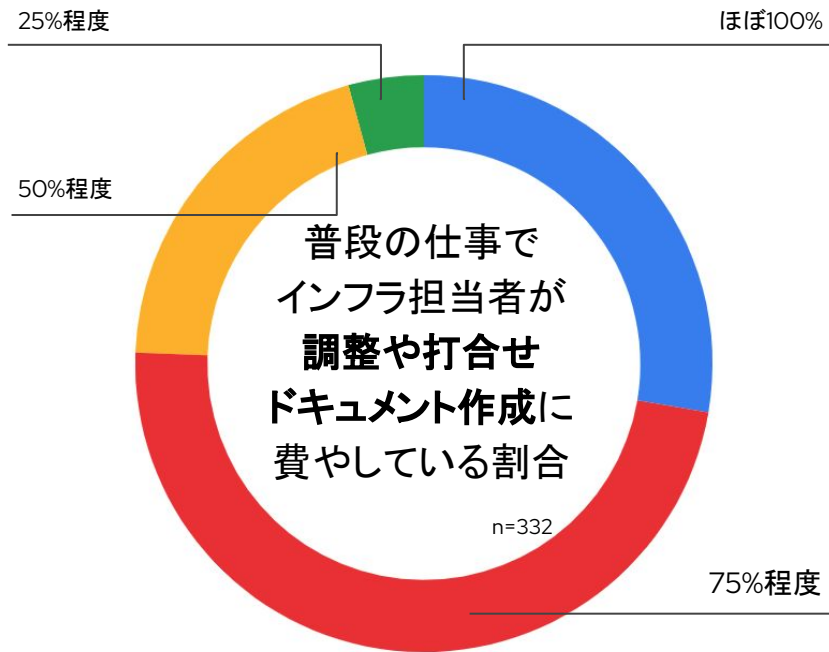
インフラ部門の働き方の実態

インフラ部門の時間の使い方(n=124)



変化したインフラ担当者の働き方

- ▶ 現在のインフラ担当者は、多くの時間を打合せ、相談、調整、ドキュメント作成に費やすようになった
- ▶ これはシステムの大規模化、複雑化に伴い、システム関係者が増えたことで、様々な連携が必要となったことに起因する
- ▶ 多くのインフラ担当者が手順の課題に取り組んでいる。しかし手順の実行の機会は減っている。

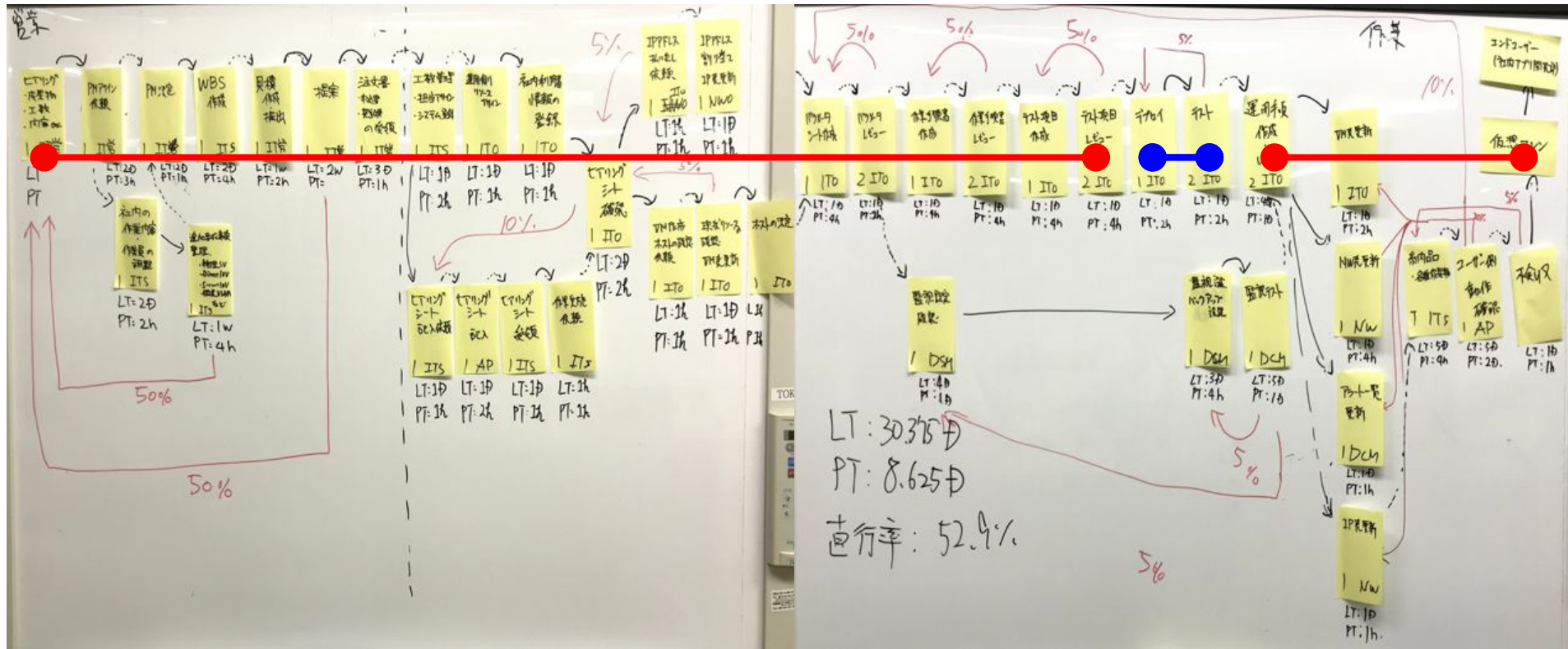


課題とゴールを明確化する手法

課題を分類する(上がった課題は、どの観点で解決したいのか?)

コスト	スピード	品質	その他	対象外
<p>NWスイチ 新製品の追加に伴う運用や設定の差が原因(メカ差異)</p> <p>設備 HW追加時のテスト等の手間</p> <p>全NW構築 設置後の確認作業に時間がかかる(1日)</p> <p>運用 VM画像構築・インストール・リソース設定など</p>	<p>共有VPMのSet-up 顧客側のConfigを顧客側で管理</p> <p>WAN PC画像移行が面倒</p> <p>SW LVM追加・変更に時間がかかる</p> <p>監視 監視の項目追加に時間がかかる</p> <p>全SW 増設時のテスト項目が多いため時間がかかる</p> <p>テスト環境 本番と差分が大きいテスト環境</p>	<p>ルーター DC毎のサービス追加等のテスト時間がかかる</p> <p>サーバ ID/OS/アプリのインストール/追加作業の時間</p> <p>監視 コア監視の追加に時間がかかる</p> <p>利用者 VM追加し申請が手間</p> <p>監視 監視項目追加に時間がかかる</p> <p>運用 リクエストによるVM追加に時間がかかる</p>	<p>プレイブック 作成時にミスが多い</p> <p>運用 手戻りの多い作業</p> <p>運用 手戻りの多い作業</p> <p>運用 手戻りの多い作業</p> <p>運用 手戻りの多い作業</p> <p>運用 手戻りの多い作業</p>	<p>情報共有 特定の個人しか運用が実施できない</p> <p>運用 運用の人数が足りない</p>

全体の構造を把握する



全体を可視化することで、初めて共通認識(課題、アプローチ)が作れる

他部署連携

作業の準備と品質確認

NW担当からIPアドレスや接続先のNWについての確認を行っている。

作業手順書やテスト項目のドキュメントを作成し、内部でのレビューを行っている。

監視チームと連携して、作成されたVMに監視設定を行っている。

払い出しの依頼を貰って、パラメータを確認したり、作業者のアサインを行っている

要件の確認と作業の調整

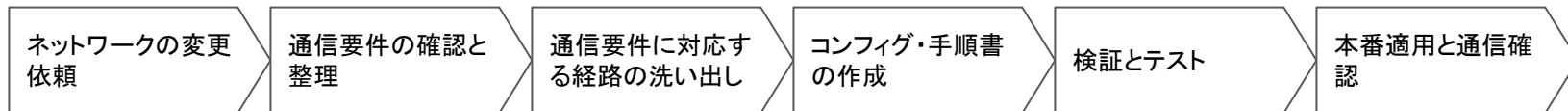
LT: 30.375日
PT: 8.625日
直行率: 52.9%

他部署連携

ネットワーク自動化へのアプローチ

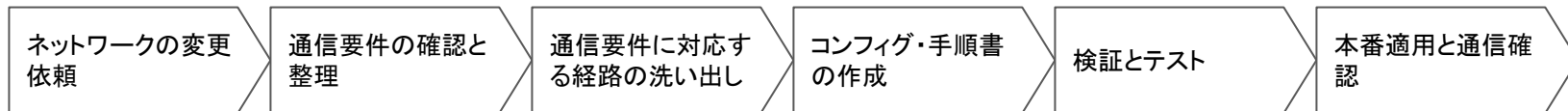
ネットワーク自動化へのアプローチ事例のまとめ

ネットワーク作業の流れ

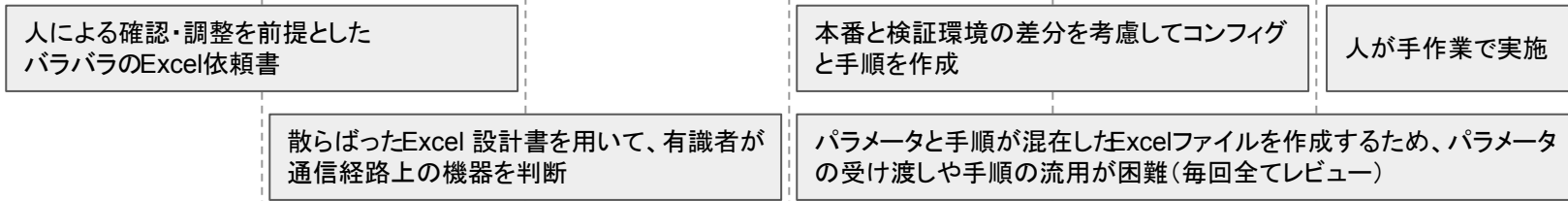


ネットワーク自動化へのアプローチ事例のまとめ

ネットワーク作業の流れ

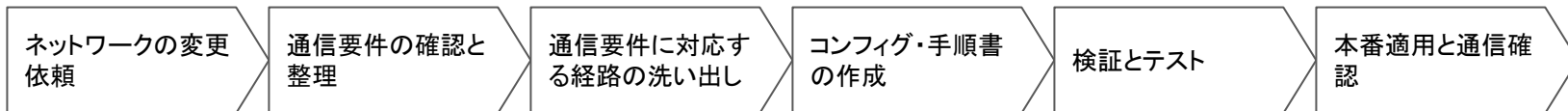


課題

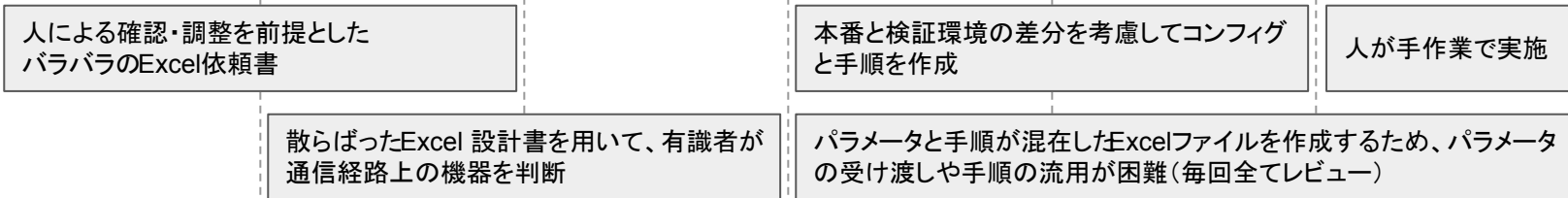


ネットワーク自動化へのアプローチ事例のまとめ

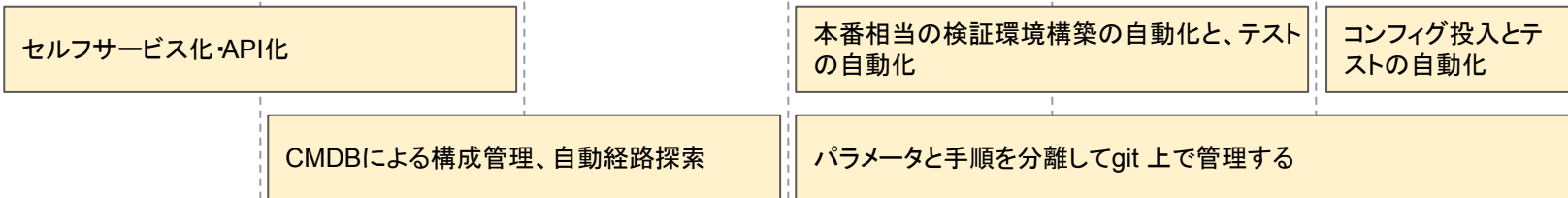
ネットワーク作業の流れ



課題

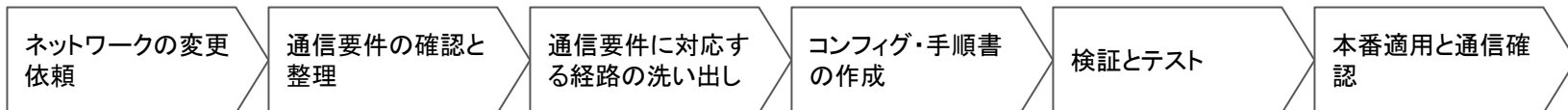


アプローチ方法

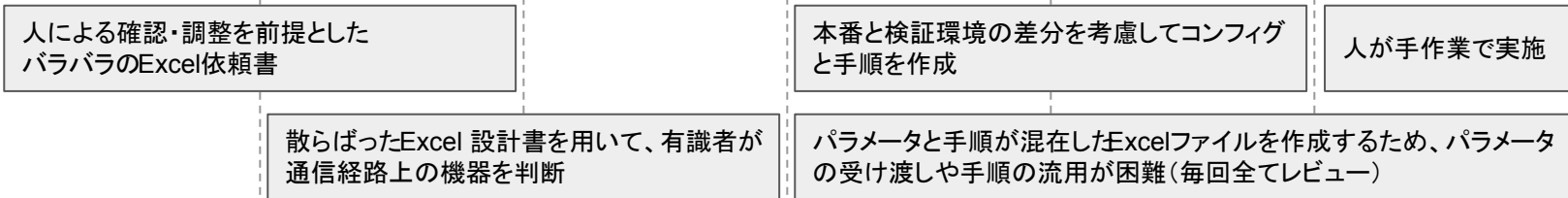


ネットワーク自動化へのアプローチ事例のまとめ

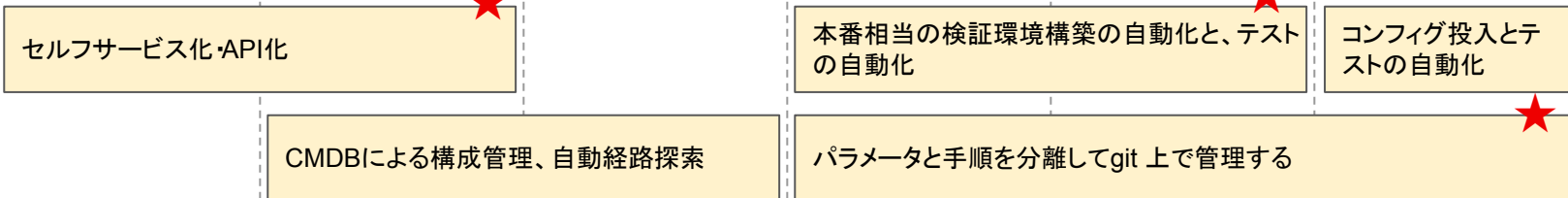
ネットワーク作業の流れ



課題

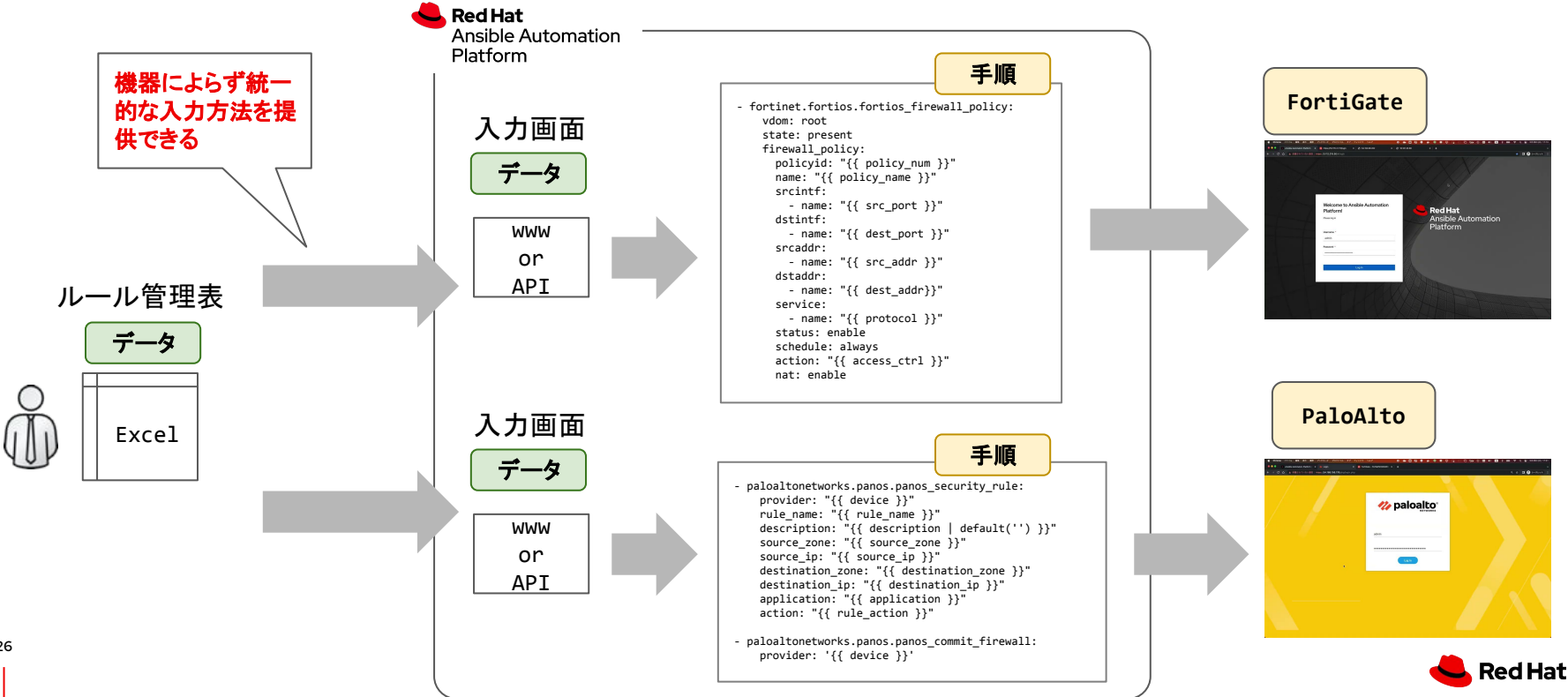


アプローチ方法



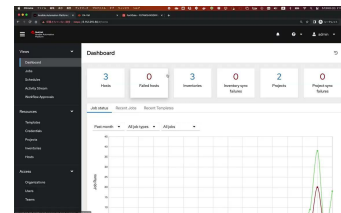
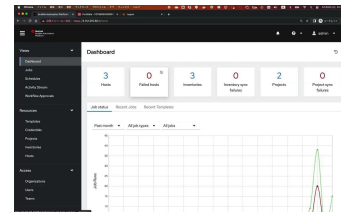
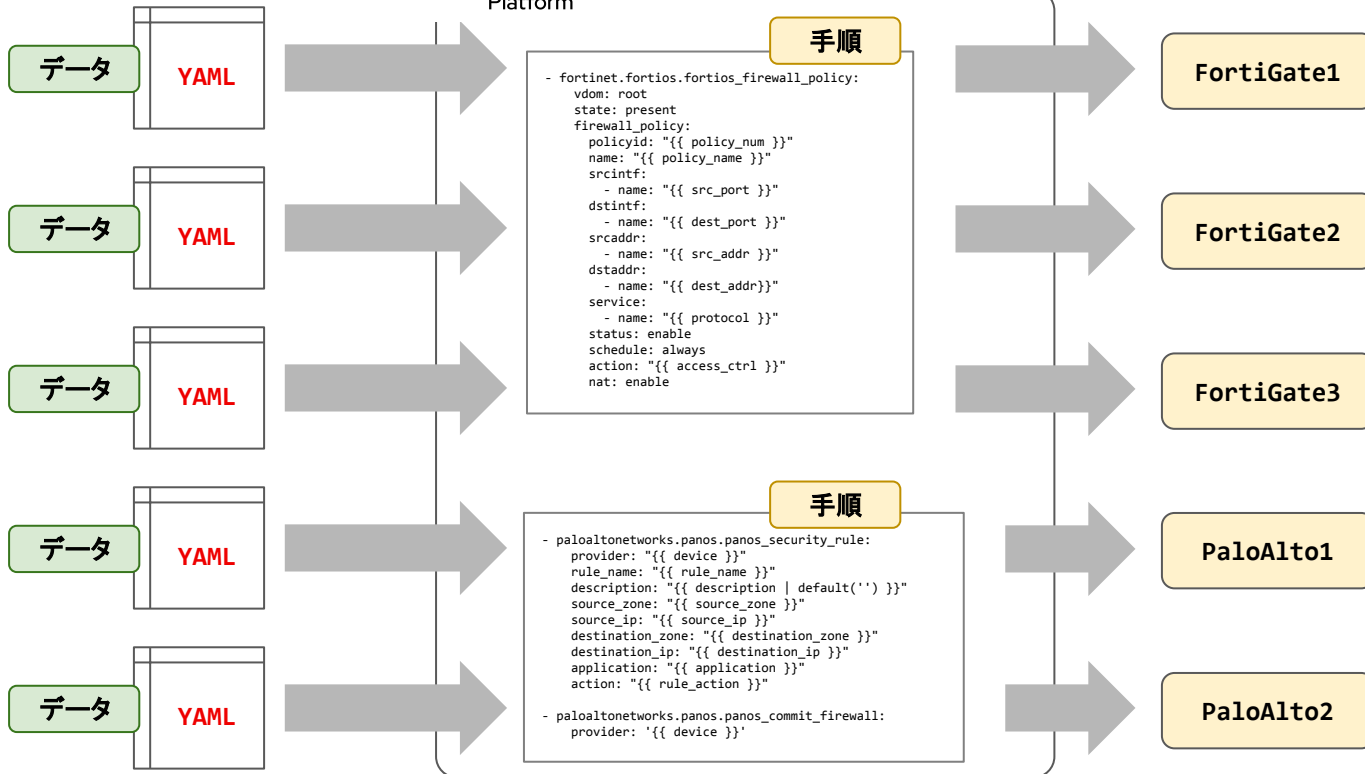
セルフサービス化・API化

- 依頼や機器の種類ごとに異なるインプットを標準化して、統一された呼び出し方法で自動化を実行できるようにする(別システムとAPI連携可能にする)



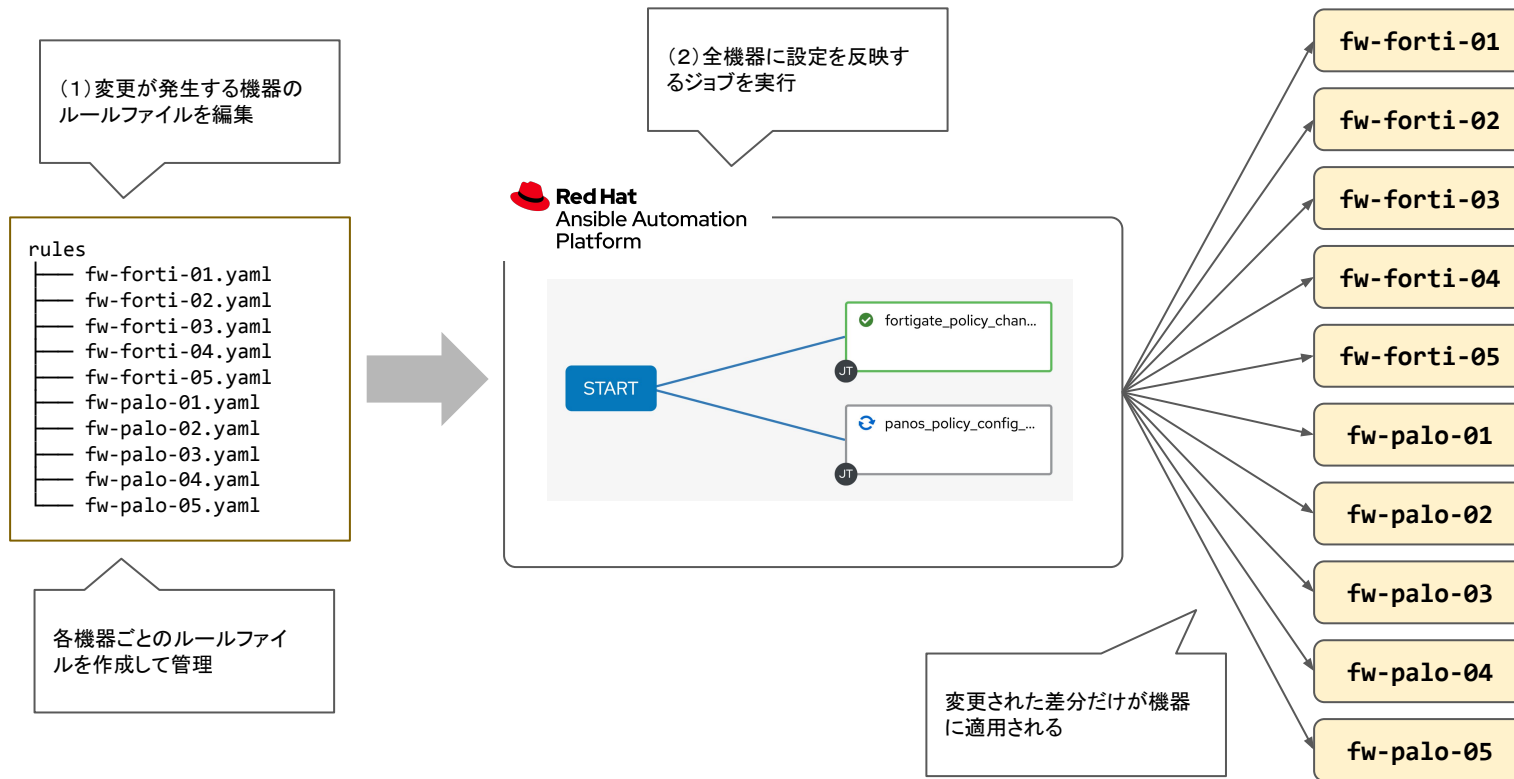
パラメータと手順を分離してgit上で管理する(1)

ルール管理表
(各機器ごとにgit上で管理)



パラメータと手順を分離してgit上で管理する(2)

FWポリシー管理の例



本番相当の検証環境構築の自動化と、テストの自動化

Comprehensive Network Automation Testing with Molecule and Cisco CML

The screenshot shows a presentation slide with a dark blue background and a white content area. On the left, there are two circular video feeds of speakers. The slide content includes:

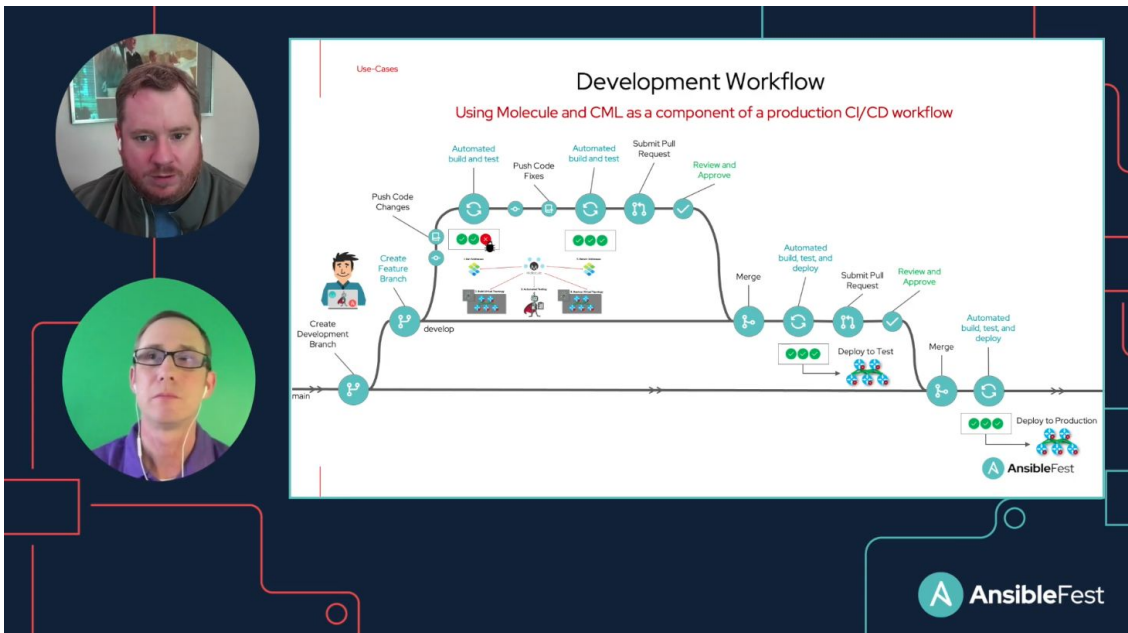
- Problem Description**
- Organizations often lack dedicated network hardware in their labs, causing resource contention for automation developers (with a red brain icon).
- Lab environments commonly have near-production networks to support application and server teams (with a red server rack icon).
- Many organizations lack automated testing for network changes (with a red clipboard icon).

The AnsibleFest logo is visible in the bottom right corner of the slide.

- ▶ 根本的なネットワーク作業の問題点
 - ・ 唯一無二な本番環境に対して、貧弱なテスト環境
 - ・ ネットワーク作業の多くの課題はこの差分を人間が解消することから発生する
- ▶ 仮想化と自動化でこの問題にアプローチしてする話

本番相当の検証環境構築の自動化と、テストの自動化

Comprehensive Network Automation Testing with Molecule and Cisco CML



▶ デモの構成要素

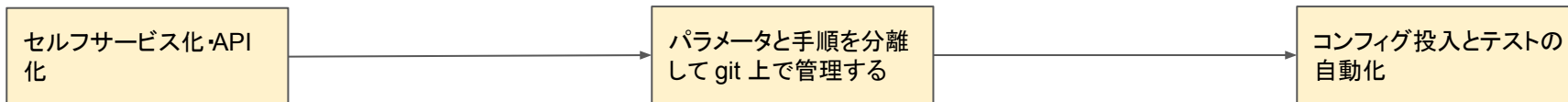
- ・ Github action
 - ・ CI/CD workflow
- ・ Molecule
 - ・ テストシナリオ
- ・ Cisco CML(旧VIRL)
 - ・ テスト環境

▶ デモの概要

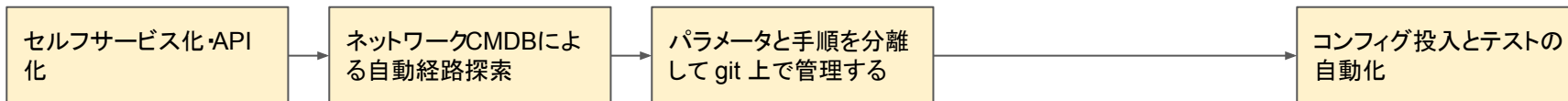
- ・ ネットワークのコンフィグを変更 (github)
- ・ 変更をトリガーにCI/CDが起動
- ・ Molecule が実行されテストシナリオが実行される
- ・ CML上でテストネットワークが構築され、変更が投入される。
- ・ 自動テストが実行される。
- ・ 成功した場合のみ本番に反映

組み合わせの例1

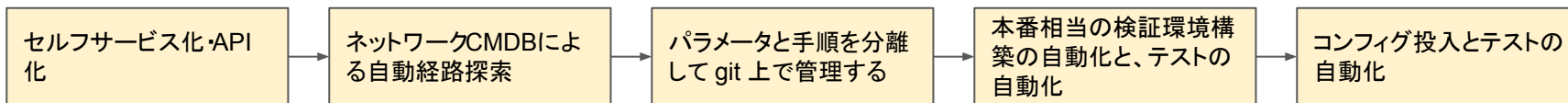
例1:FWポート開放設定やSWインターフェースの設定などの単体で完結する部分の自動化例



例2:ACLのように通信経路上の全ての機器に設定を入れる場合の自動化例

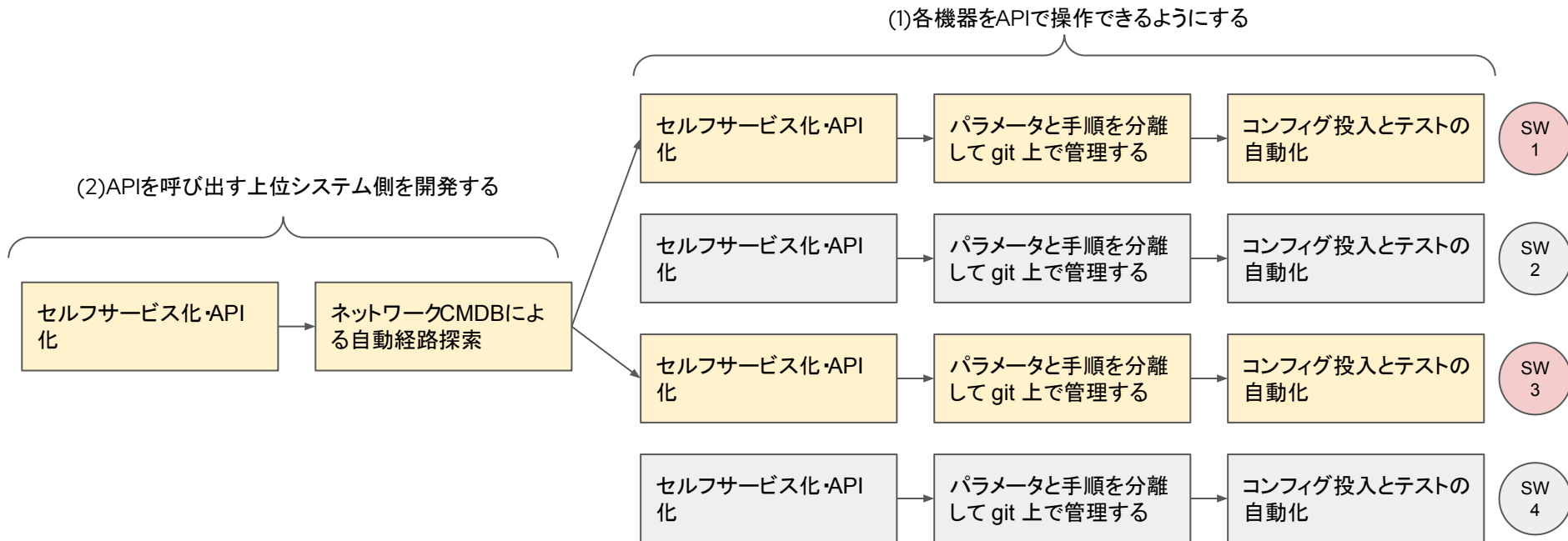


例3:ルーティングを含む複雑なネットワーク環境を自動化する例



組み合わせの例2

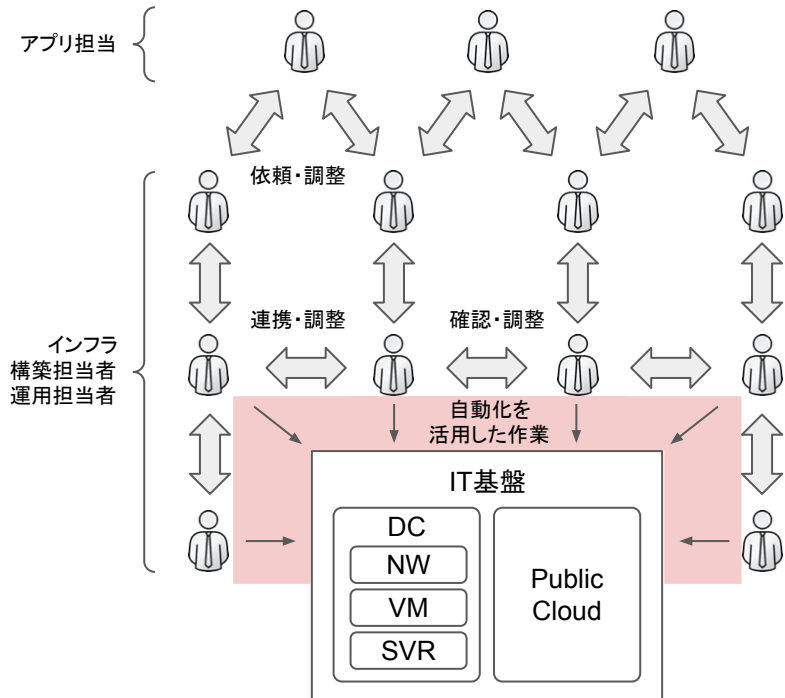
例4: 各組み合わせを階層化して利用することも可能です



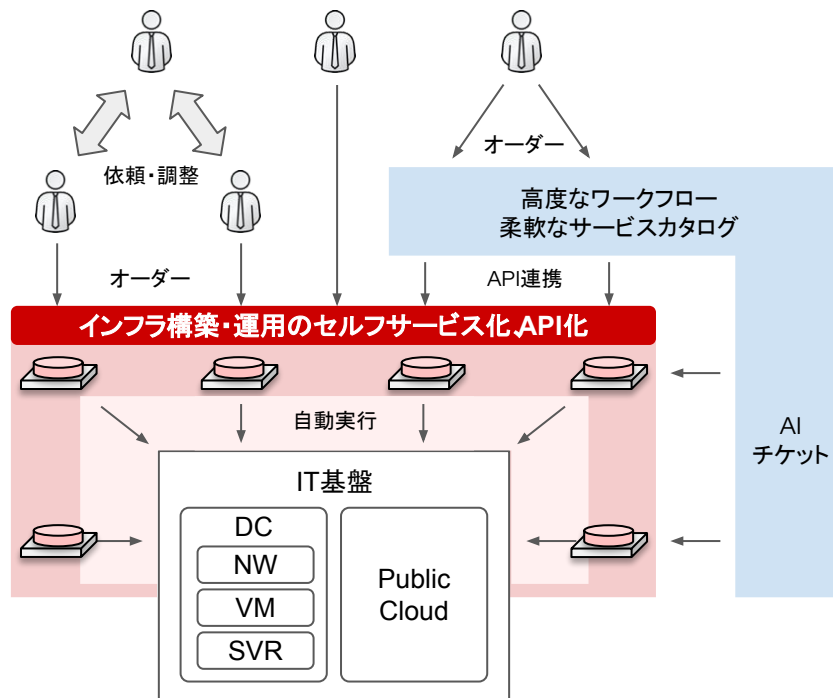
おすすめの始め方 ～Ansibleを使った場合～

Ansible の活用トレンドの変化

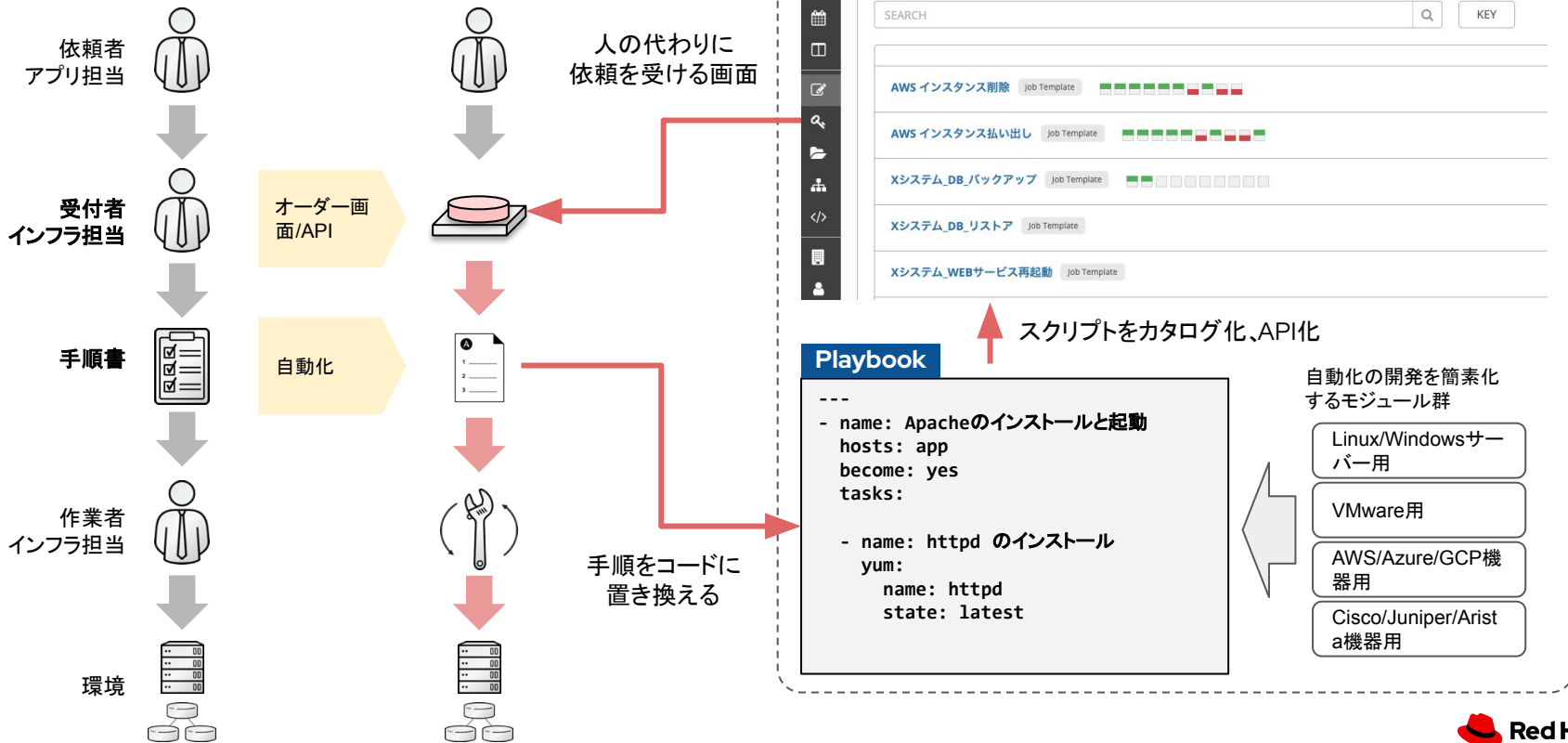
作業を助ける便利な道具



セルフサービス・API化を実現する手段



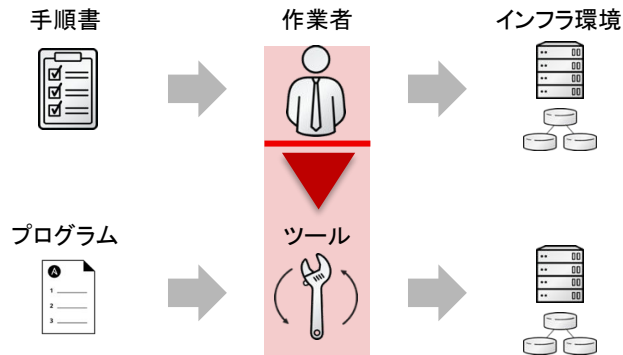
Ansibleの活用イメージ



セルフサービス化がおすすめな理由

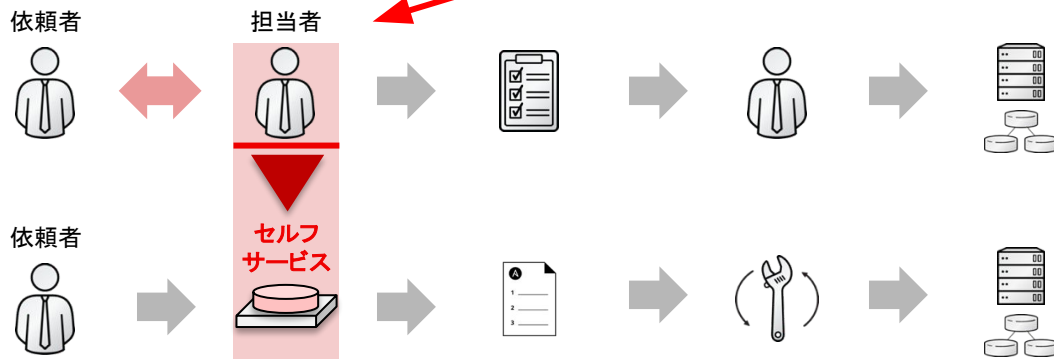
いままでの自動化

- ▶ 人間が実行する手順に注目
- ▶ 手順の実行をツールやプログラムに置き換える



これからの自動化

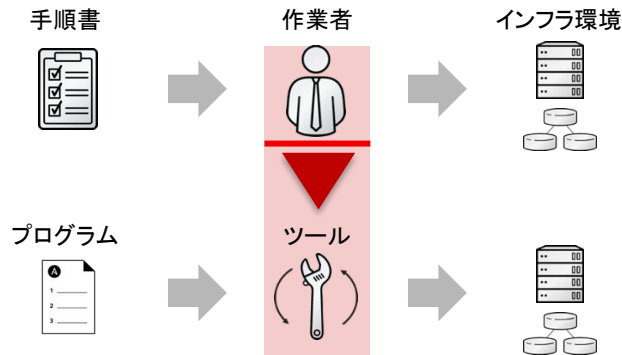
- ▶ コミュニケーションの発生箇所に注目
- ▶ コミュニケーションの相手をツールに置き換える



セルフサービス化がおすすめの理由

いままでの自動化

- ▶ 人間が実行する手順に注目
- ▶ 手順の実行をツールやプログラムに置き換える

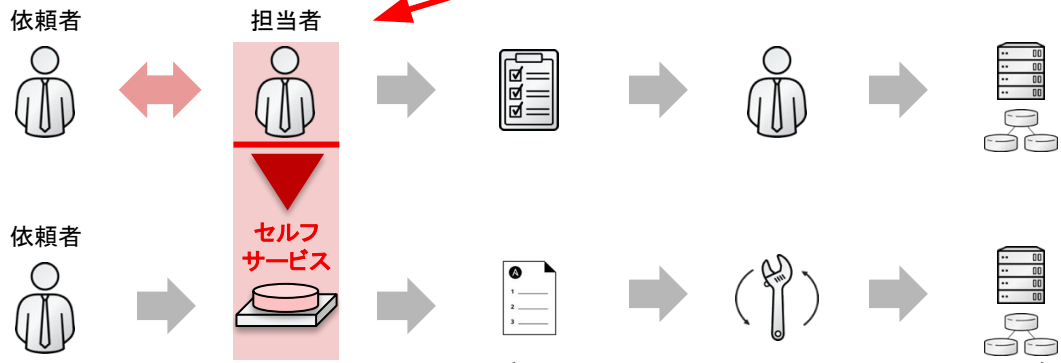


小さく始められる

即効性が高い

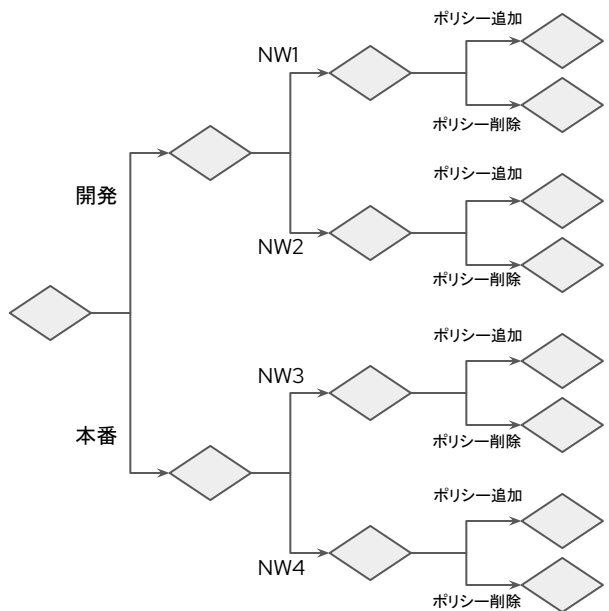
これからの自動化

- ▶ コミュニケーションの発生箇所に注目
- ▶ コミュニケーションの相手をツールに置き換える

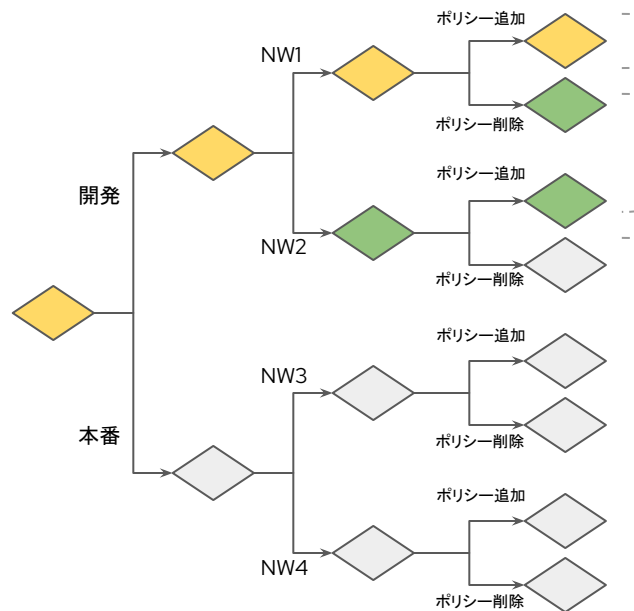


自動化1.0の延長で
取り組める

小さな自動化から初めて、徐々に拡大



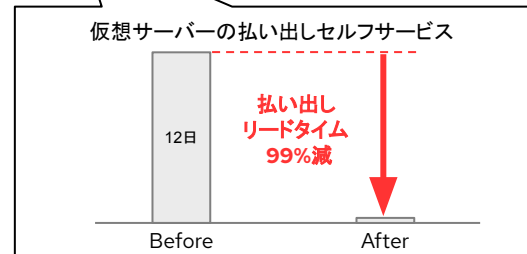
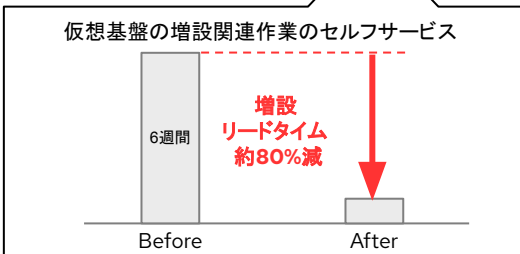
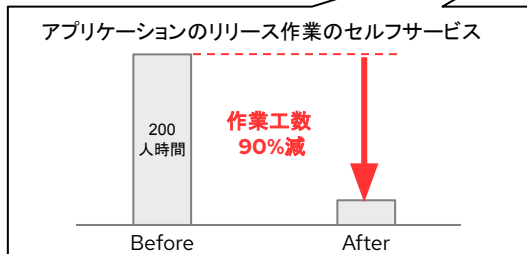
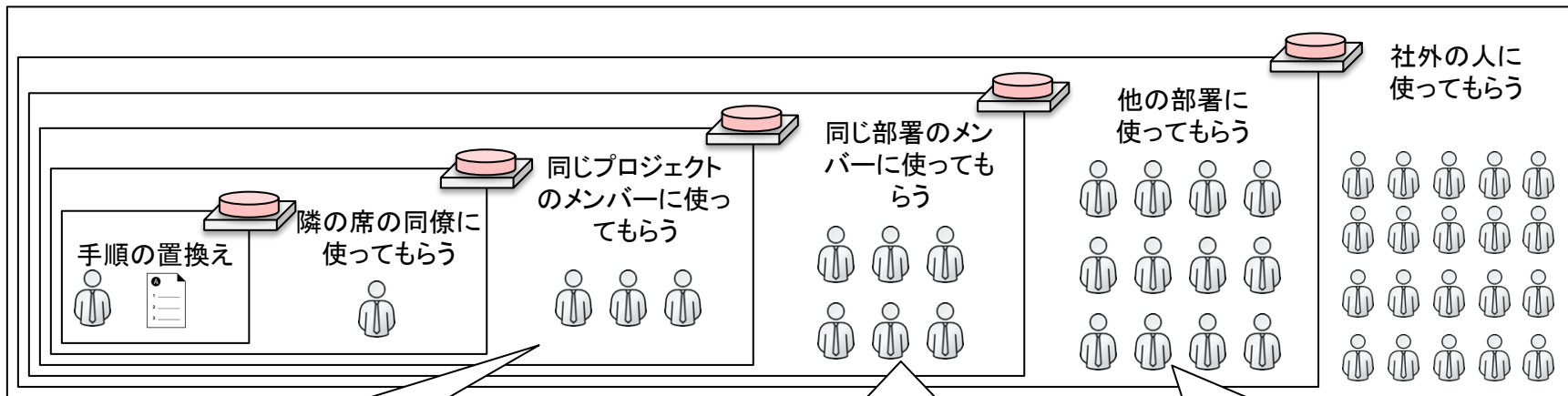
最初から全てのパターンを網羅したセルフサービスやAPIを作るのは難しい



最低限の機能を持った限定的なパターンへの対応からスタートして、徐々に機能を拡張していく

- 1st step(ver1.0)
特定用途における最低限の機能をセルフサービス化・API化する
- 2nd step(ver1.1)
ver1.0に少しの機能を追加して、対応するパターンを少し拡大する
- ⋮
- 3rd step(ver1.2)
- 4th step(ver1.3)
- 5th step(ver1.4)

はじめは隣の同僚に使ってもらうセルフサービスから



どこをゴールにするかは自分、プロジェクト、組織で決めてよい

まとめ

「効果が出る」ネットワーク自動化の始め方とは

- ▷ 全体の業務プロセスを可視化し、課題とアプローチの共通認識を作る
- ▷ 課題に合わせたアプローチを採用する(手順を置き換えるだけが自動化ではない)
- ▷ 始める箇所に迷ったらセルフサービス化・API化がおすすめ

Interactive Lab によるAnsible Automation Platform と新機能の体験

instruqt > Controller CLI > Host01 CLI Editor Dashboard

Red Hat Ansible Automation Platform

admin

Progress 59%

Introduction

<<最小のPlaybookの開発と実行>>

以下の手順に従い、演習環境を進めてください。

左側のウィンドのロードに少し時間がかかる場合があります。表示されるまで少々お待ち下さい。

↓をクリックすると次の手順が展開されます。

- タスク1: 環境の確認
- タスク2: Playbook の作成
- タスク3: Playbook の実行 (サービスの停止)
- タスク4: Playbook の実行

Skip Check



<https://red.ht/rhja-lab>

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.



[linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)



[youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)



[facebook.com/redhatinc](https://www.facebook.com/redhatinc)



twitter.com/RedHat