

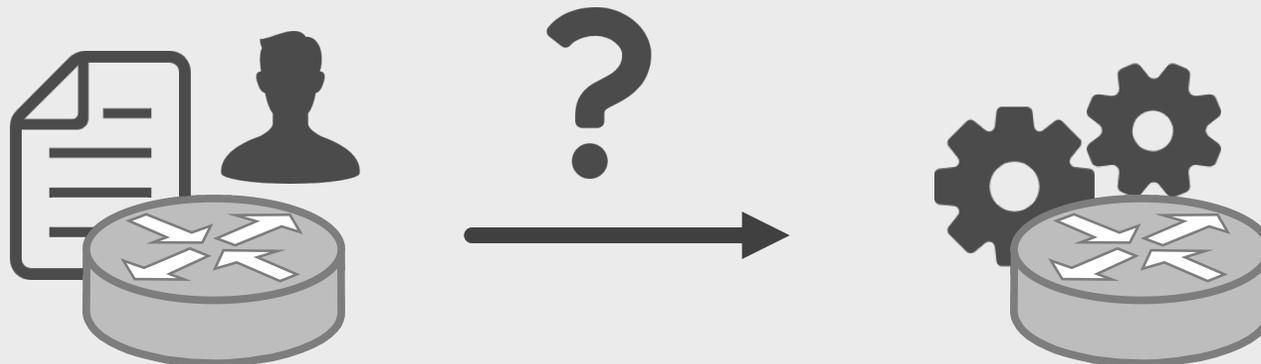
# 自動化の下ごしらえ

2020/08/28 JANOG46 Meeting Day3

株式会社エーピーコミュニケーションズ  
技術開発部 自動化グループ  
横地 晃

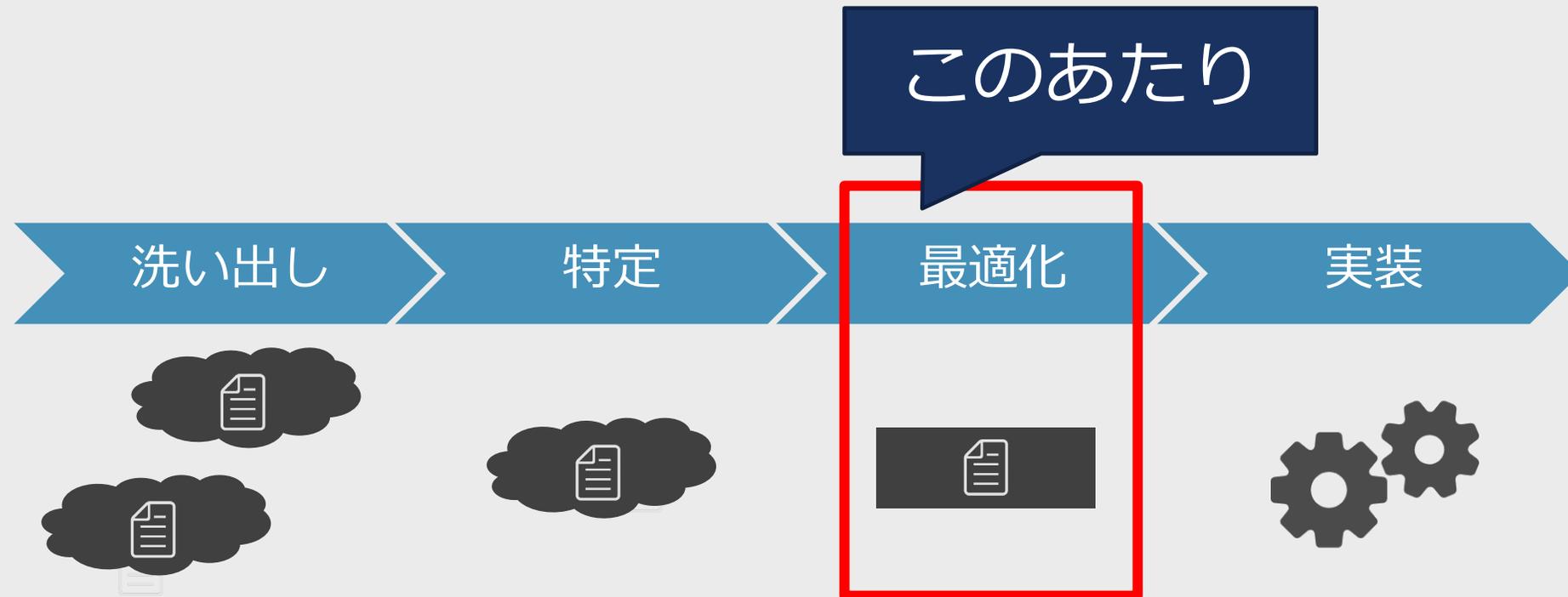
## はじめに

- ネットワーク運用自動化する場合、既存の手順書をなぞるようにそのまま実装していけばよいものでしょうか？
- まずは「下ごしらえ」が必要です。
- 自動化しやすくするためのポイントをいくつかご紹介します。



# 前提

- 業務の洗い出しや整理が終わり、自動化対象の手順書を特定した後のお話



# 自己紹介



横地 晃 @akira6592

所属

(株) エーピーコミュニケーションズ  
技術開発部 自動化グループ



業務

ネットワーク自動化の支援、各種アウトプット

JANOG

JANOG42: Ansible ネットワーク自動化チュートリアル  
JANOG43: 自動化の行き着く先は?  
JANOG44:ここからはじめよう、運用自動化

共著  
寄稿

Ansible 実践ガイド 第3版  
Software Design 2018年12月号 他



<https://tekunabe.hatenablog.jp/>



<https://book.impress.co.jp/books/1118101094>

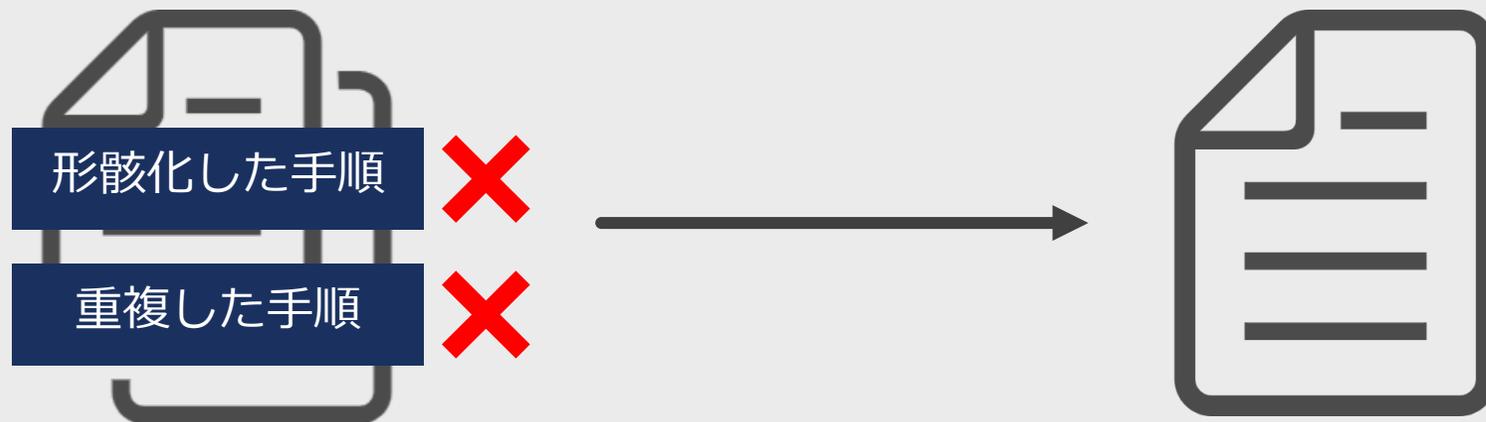
# 内容

1. 引く
2. 足す
3. 決める
4. 分ける
5. 整える
6. 入れ替える
7. 戻す

# 1. 引く

## 【引く】 不要な手順を削除する

- 余計な手順は負債化
- 残した手順はあとから削除しにくい



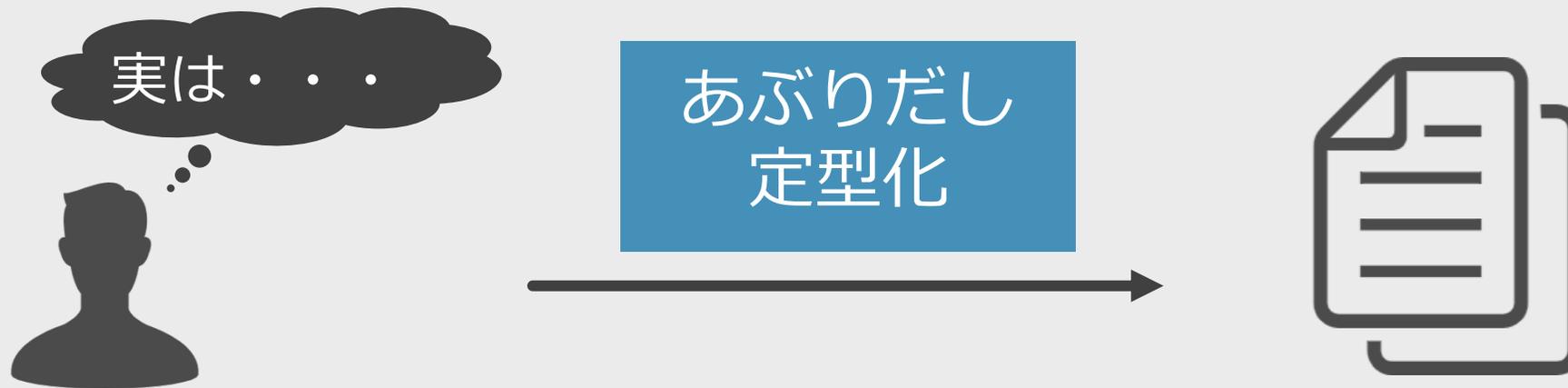
## 【引く】 不要な手順を削除する（例）

- 形骸化した手順を削除
  - 手順の経緯を追い、不要だと判断できれば削除
  - 手順書上は「確認」とあるが「記録」で十分なら確認を削除
  - 記録すら不要ならまるごと削除
- 重複した手順を削除
  - 監視システムなどでカバーできている確認手順は削除
- 手作業ならではの手順を削除

## 2. 足す

## 【足す】 隠れ手順をあぶりだす

- 手順書になくても、人がよしなにやっていた作業
- 機械は脳内補完できない
- 暗黙知を形式知



## 【足す】 隠れ手順をあぶりだす (例)

- 手順書で表現しきれていないもの
  - 例: 手順書上は1回の show コマンド実行だが、実際はOSPFネイバーが確立されるまで繰り返し実行して確認
- 重要だがそもそも書かれていない手順
  - 個人の経験、暗黙知

	A	B	C	D	E	F	G
1							
2		項番		手順	実行コマンド		
3		1		コアルーター事後確認			
4		1	1	OSPF ネイバー状態を確認する	show ip ospf neighbor		
5		1	2	ルーティングテーブルを確認する	show ip route		
6		1	3	:			
7		1	4	:			

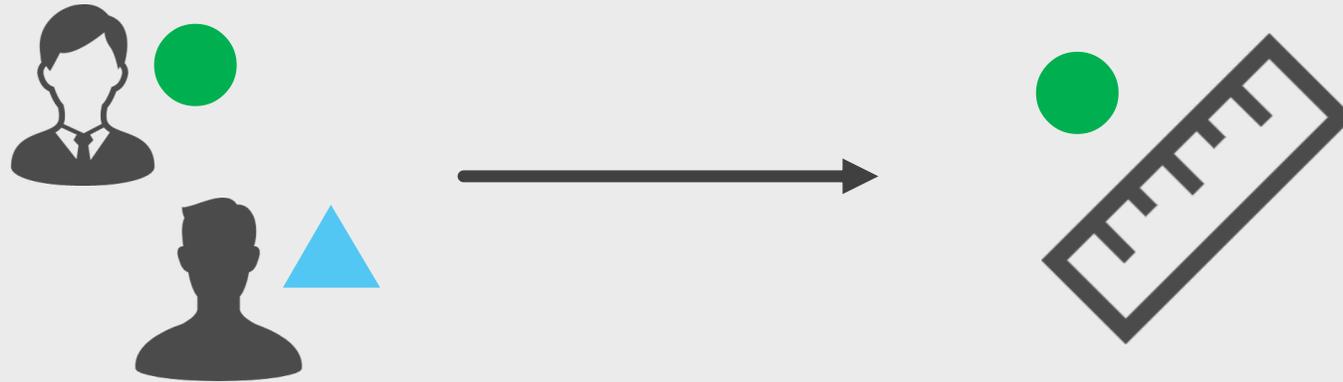
本当に1回の実行？  
結果のどこを確認？

暗黙的に実行している  
手順が抜けていないか？

# 3. 決める

## 【決める】 判断基準を明確に決める

- 統一した基準やロジックなしには自動化できない
- 新任者教育にも似た要領



## 【決める】判断基準を明確に決める（例）

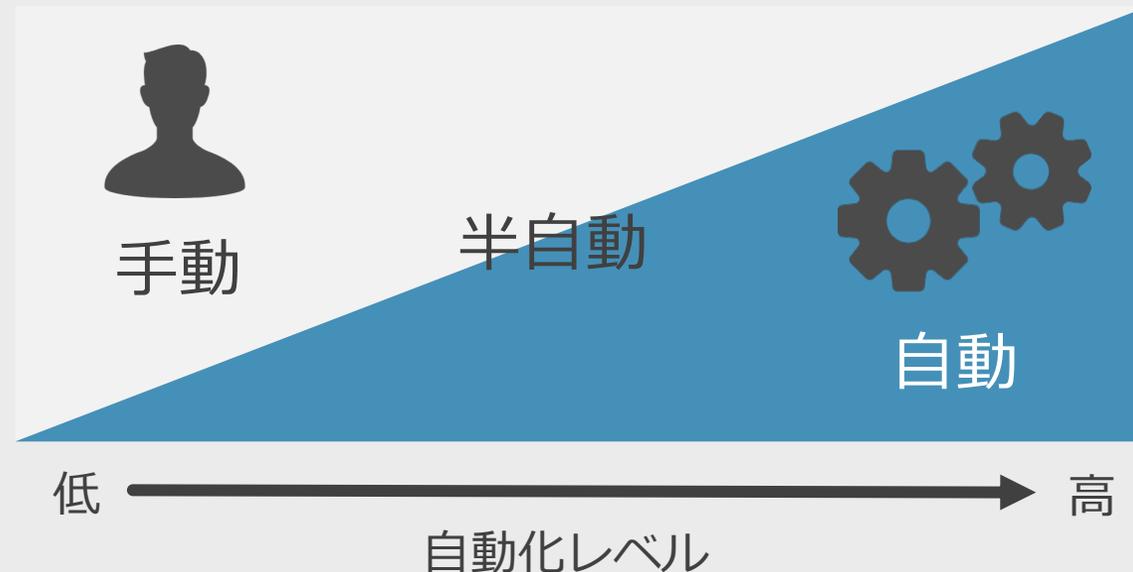
- ロジック化、数値化、統一

要注意ワード	書き換え例
異常がないこと	show log の結果に「ERROR」が含まれないこと
同程度であること	差分が5以内であること
増加していないこと	作業前と比較し、差分が0であること

## 4. 分ける

## 【分ける】 機械と人の役割を分ける

- 自動化する/しないだけでなく半自動という選択もある
- 手順を分解して、どこまで自動化できて、どこからできないかを見極める



## 【分ける】機械と人の枠割を分ける（例）

- 事前事後のコンフィグ差分の妥当性判断の自動化が難しければ、判断に必要な情報収集と比較までをいったん自動化
  - 「難しい」理由は基準の決めにくさや、リクスの大きさ、技術的な難易度など
  - 段階的に自動化していく



手順書

事後コンフィグを取得し、事前コンフィグとの差分が想定通りであること

コマンド実行

比較

判断



自動

手動

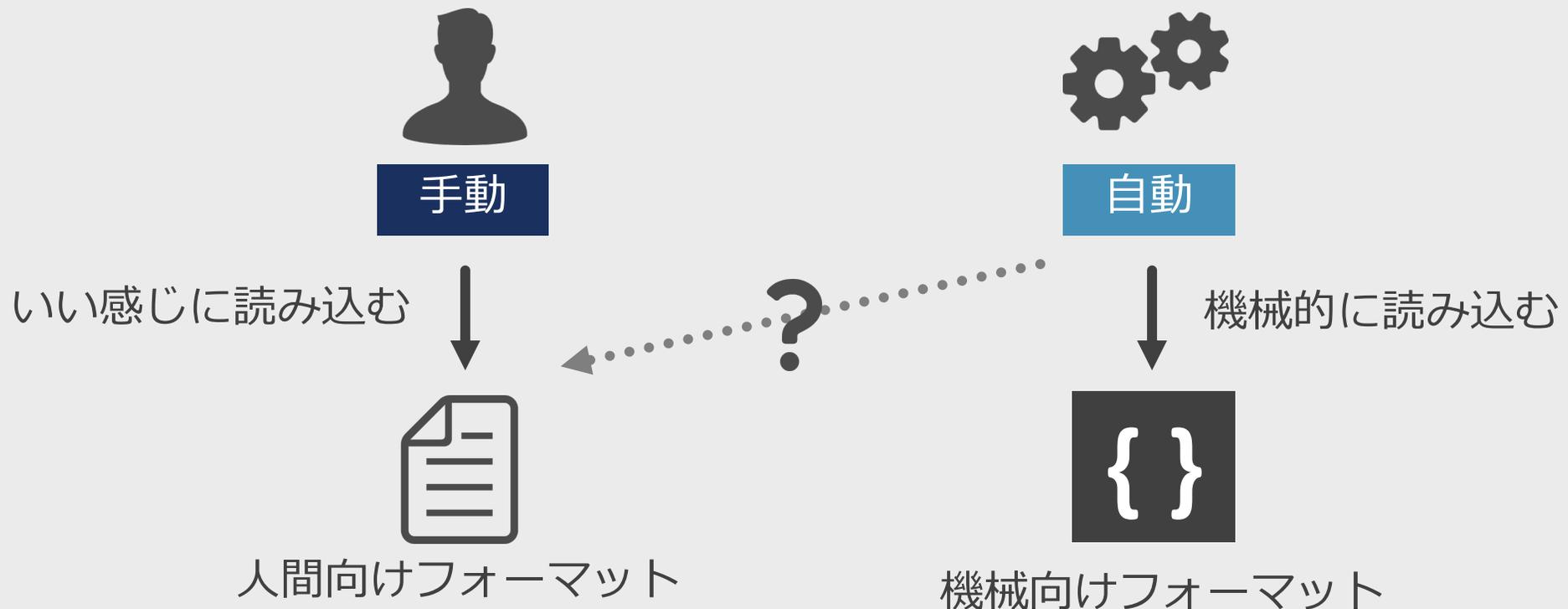


※基準を決めて自動化が理想

# 5. 整える

## 【整える】フォーマットを整える

- 人が扱いやすいフォーマットを機械が扱いやすい構造化データにする
  - YAML や JSON など



# 【整える】フォーマットを整える（例）

- 人間向けの方眼紙による申請書を YAML にする

どのセルに何が？

データ構造が壊れやすい



自動

読み込みやすい

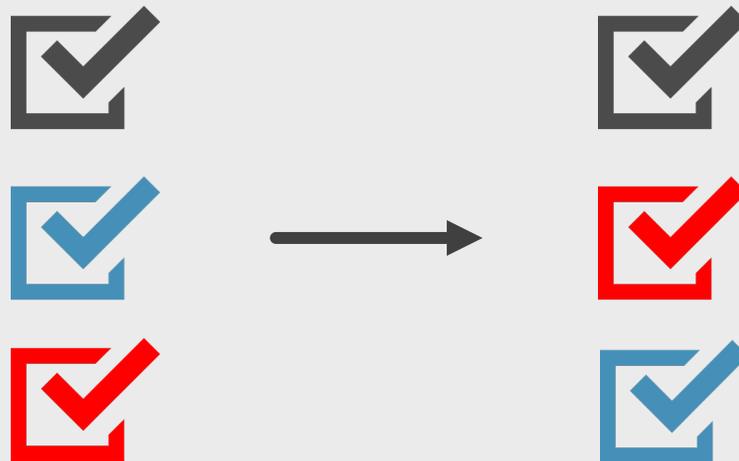
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1																					
2																					
3																					
4																					
5																					
6																					
7																					
8																					
9																					
10																					
11																					
12																					
13																					
14																					
15																					

```
---
- source:
  prefix:
    - 10.0.0.0/24
    - 10.0.1.0/24
  protocol: any
destination:
  prefix:
    - 172.16.1.0/24
  protocol: tcp
  port: 22
action: permit
```

## 6. 入れ替える

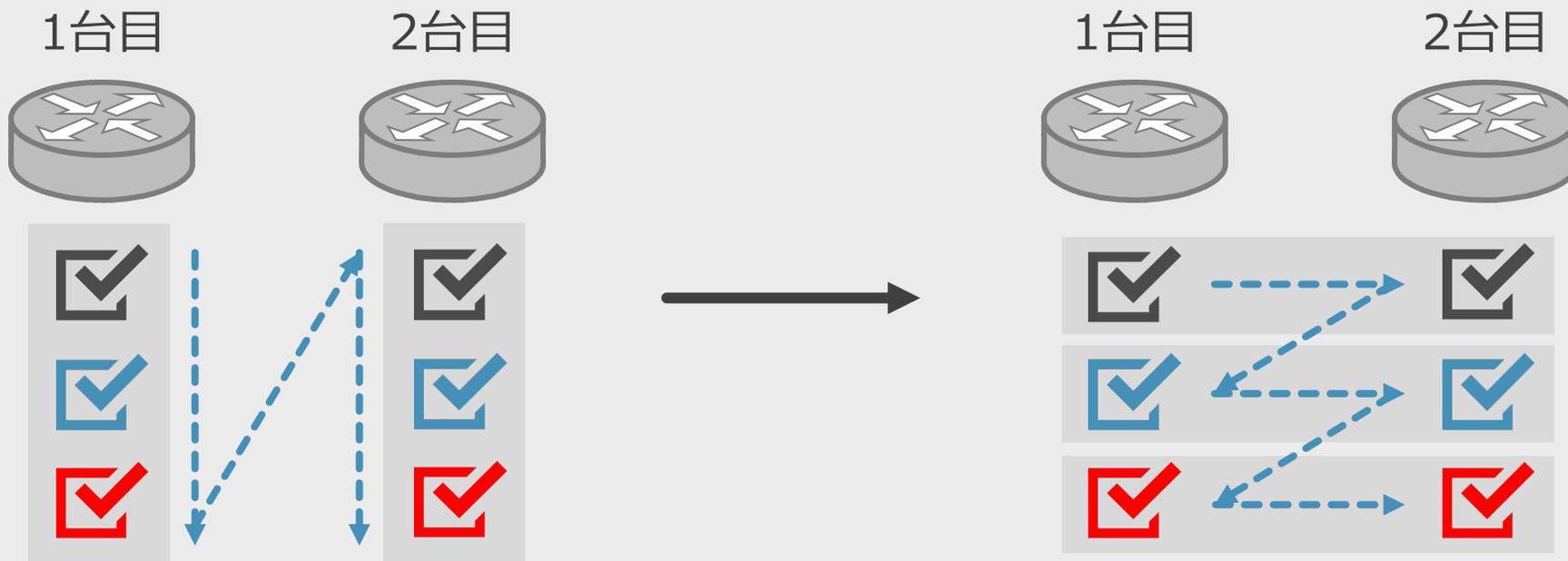
## 【入れ替える】順序を最適化する

- 特に順序に意味がない箇所は、自動化しやすく入れ替えてしまう



## 【入れ替える】順序を最適化する（例1）

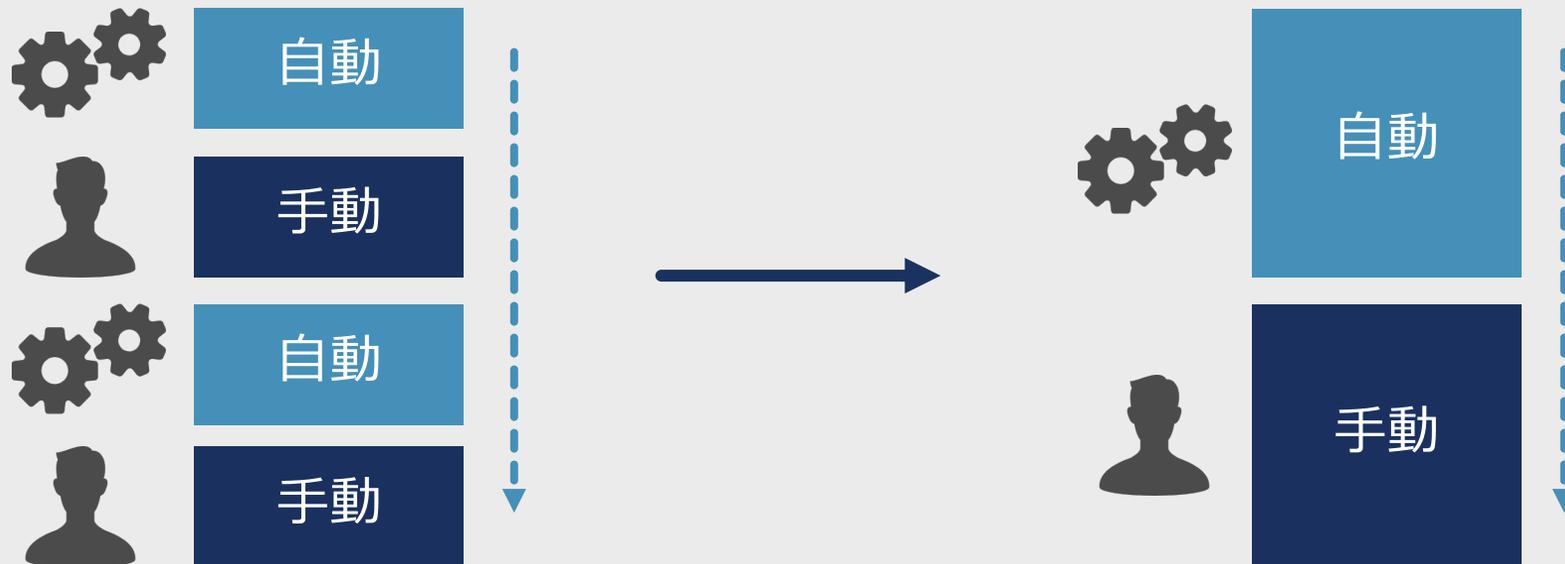
- 対象機器ごとの作業をタスクごとに変更
  - 実装との親和性を考慮



※Ansible のデフォルト動作  
(strategy: linear)

## 【入れ替える】順序を最適化する（例2）

- 人の介入回数を少なくする
  - 自動化しきれなかった手動による状態確認をまとめる



## 7. 戻す (オプション)

## 【戻す】 あえてデグレ

- 手動の効率化の延長上に自動化があるとは限らない
- いったんあえてデグレさせることで自動化しやすくなることも



## 【戻す】 あえてデグレ (例)

- ファイルによる一括インポートを個別作業に戻してから自動化
  - インポート用のAPIがないなどの事情により



まとめ

# まとめ

- |          |             |
|----------|-------------|
| 1. 引く    | 不要な手順を削除する  |
| 2. 足す    | 隠れ手順をあぶりだす  |
| 3. 決める   | 判断基準を明確に決める |
| 4. 分ける   | 機械と人の役割を分ける |
| 5. 整える   | フォーマットを整える  |
| 6. 入れ替える | 順序を最適化する    |
| 7. (戻す)  | あえてデグレ      |

## ディスカッション



- ・ 質問
- ・ 他に工夫をしていること
- ・ 事例
- ・ 成功や失敗から学んだこと

など

## 参考

- 書籍: 人と機械の共生のデザイン-「人間中心の自動化」を探る
  - 人の機械の役割を分ける際の、自動化レベルの考え方が参考になる
  - <https://www.morikita.co.jp/books/book/2510>

【本資料のお問合せ】  
株式会社エーピーコミュニケーションズ  
技術開発部 自動化グループ  
automation@ap-com.co.jp