Web公開向け資料

NTT DATA
Global IT Innovator

大規模プロジェクトにおける
インフラ自動化導入の壁を打破せよ

2017年7月20日

株式会社ＮＴＴデータ
菅原　亮

アジェンダ

1. 自己紹介
2. 現状の課題
3. 提案フェーズ
4. 設計・開発フェーズ
5. まとめ

【自己紹介】

自己紹介(Self Introduction)

名前:

菅原　亮(すがはら　りょう)

所属:

株式会社NTTデータ 技術革新統括本部
システム技術本部 方式技術部

得意な分野:

インフラ自動化分野を得意とするインフラ技術
Puppetに関する記事や著書をいろいろ執筆
日本Puppetユーザ会 会長

【現状の課題】

日本におけるエンタープライズのIT事情（1/3）

“日本型SI”はクライアントから受注したシステム開発を個々の作業別で協力会社に発注という形態が多く、これに起因する課題も多くあります。

設計、構築・試験、保守・運用を担当する会社がバラバラになることが多く、
各フェーズを担当する会社間の連携が上手くいかないケースもよくあります。

設計（A社）

構築・試験（B社）

保守・運用（C社）

各フェーズを担当する会社間での連携がスムーズでないと、ライフサイクルを通じての
インフラ自動化導入の上で必ず大きな壁になります。

日本におけるエンタープライズのIT事情（2/3）

さまざまな理由によりブラックボックス化されてしまっているインフラでは
インフラ自動化導入に必要な情報が入手困難なケースが多々あります。

要員、ライセンス、政治的な問題など、さまざまな要因により自動化導入に必要な
情報を入手できず、自動化導入が困難を極める場合もあります。

担当者がプロジェクトを退場してしまい、
内部がどうなっているのか誰もわからない・・・

担当の会社が別会社なので
必要な情報を提供してもらえない・・・

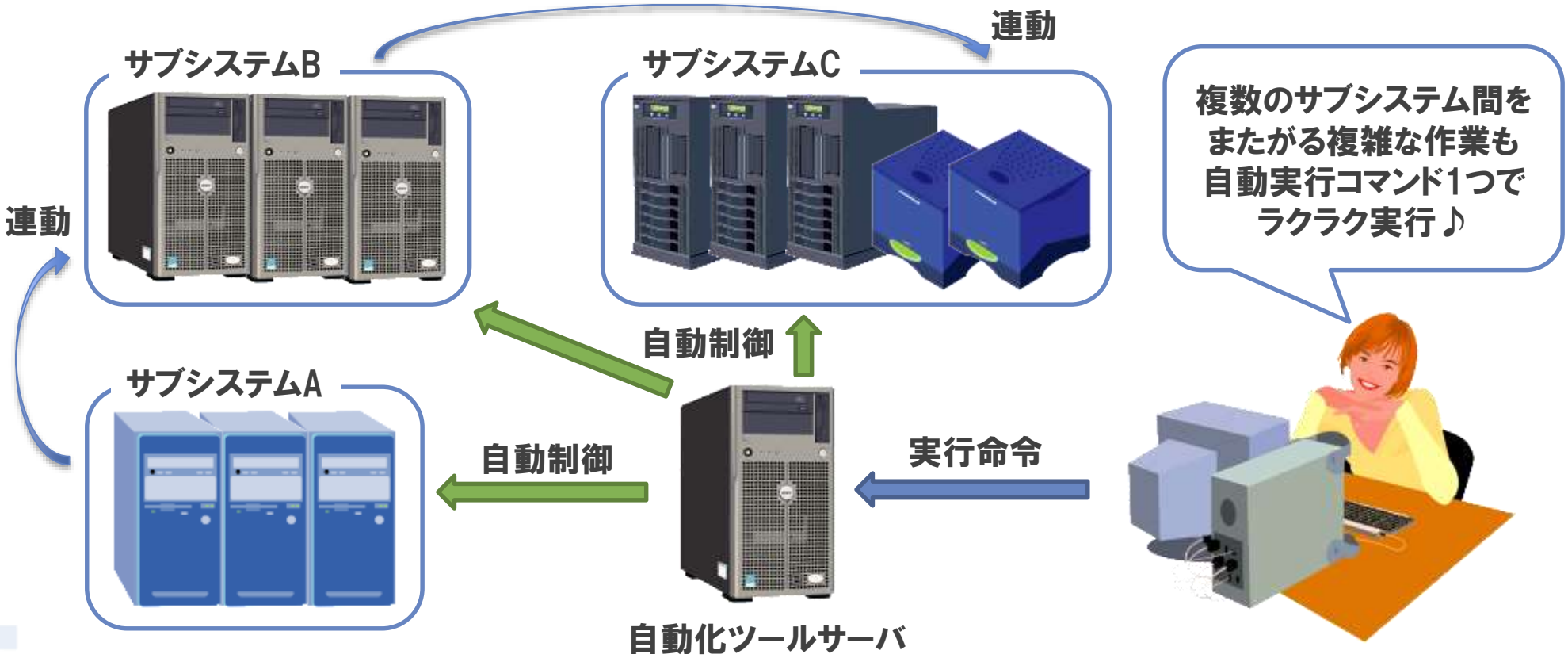
お客様にシステムの変更を許可してもらえず
自動化ツールの導入ができない・・・

特に既存のシステムを自動化しようとするなら情報が無ければ何もできません。
プラットフォームが多いほど情報も分散しているケースが多いため注意が必要です。

日本におけるエンタープライズのIT事情（3/3）

さまざまなサブシステムが複雑に連携しているシステムでは
システム全体を自動化しないと期待する効果が出ないケースもあります。

理想的な自動化導入後の姿

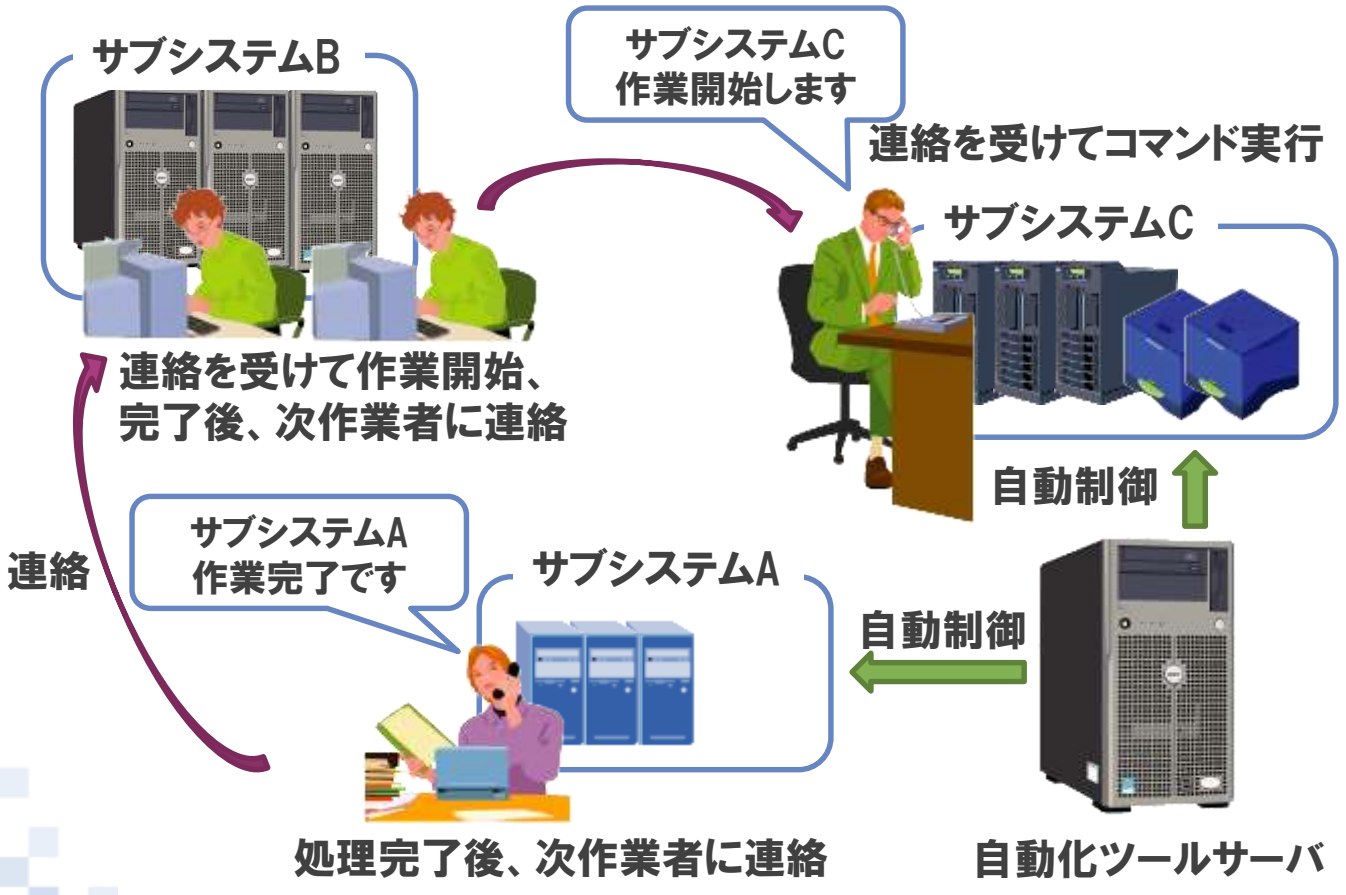


NTT DATA

日本におけるエンタープライズのIT事情（3/3）

さまざまなサブシステムが複雑に連携しているシステムでは
システム全体を自動化しないと期待する効果が出ないケースもあります。

システム全体を通じた自動化ができなかった場合



【提案フェーズ】

インフラ自動化の目的を“正しく”考える、その前に…

自動化導入
頑張るぞ！

おー！

この頃は作業ミスが
ホントに少ないよね
自動化導入のおかげだね

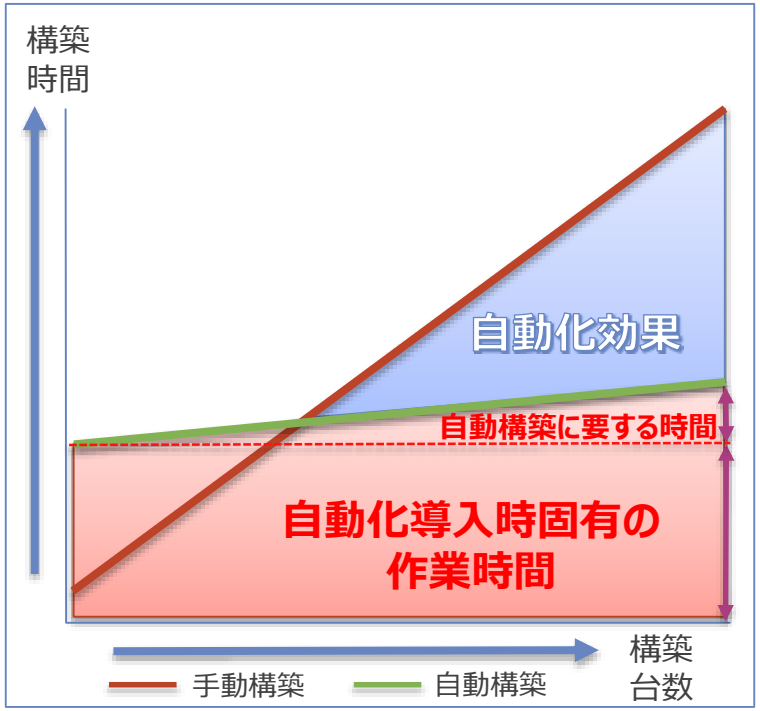
最近はスケジュールの
遅延がなくなりましたね
自動化の効果ですね

どちらがあるべき姿でしょうか？

インフラ自動化の目的を“正しく”考える

「自動化を導入したら余計コストが掛かってしまった」
そんな結果にならないため、“手段”を“目的”にしてはいけません。

自動化導入時、手動構築では必要の無い自動化導入時固有の作業が発生します。
工数見積もりやスケジュール作成時は、この固有作業の工数を考慮する必要があります。



自動化導入前のアセスメント

↓

自動化ツールの習熟

↓

自動化ツール周りの環境構築

↓

自動化スクリプトの開発と試験

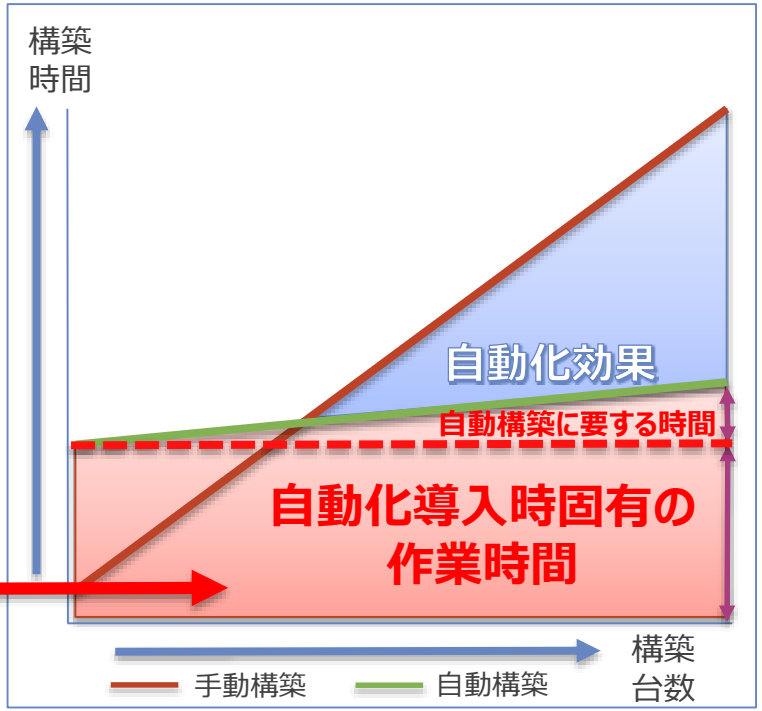
終わってから「自分たちは何がやりたかったんだっけ？」と気付いても手遅れです。

コスト削減 ～ 甘言に潜む落とし穴

「自動化を導入してコスト削減しましょう」は多くの場合で失敗します。
導入効果について正しく認識して導入を決定することが重要です。

耳あたりが良い「コスト削減」を強調すると、得られる導入効果ばかり目が行ってしまい
自動化導入時固有の作業について見落としてしまいかねません。

コストの話題は自動化導入に
向けて上司を説得する場面では
とても効果的でしょう。
しかし**導入効果に過剰な期待**を
生みやすい“両刃の剣”です。



自動化導入のコストメリットは、システムの初期導入時よりもむしろ
保守運用、追加開発や次期リプレースなどの段階になってから大きく見えてきます。

ライフサイクル全体を通じた自動化導入（1/2）

**自動化導入では「構築自動化」と考えるではなく「IT自動化」と考えて、
システムのライフサイクル全体を通じた自動化を計画すべきです。**

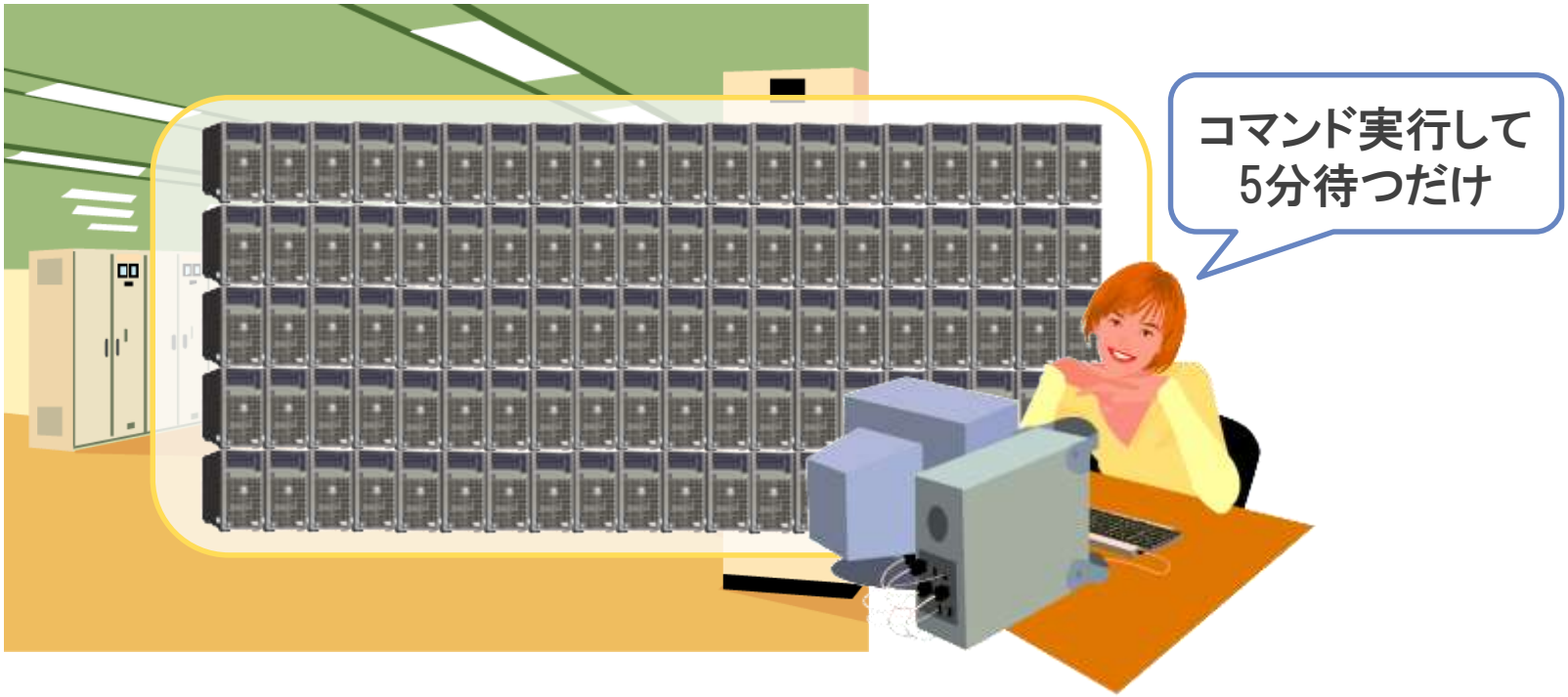
手作業で1サーバあたり4時間掛かるメンテナンス作業を実施するとした時、
1日8時間稼動、3人が同時並行で対応する場合では1日6サーバが限界です。

対象が100サーバある場合だと作業完了までに単純計算で最短17日必要です。

ライフサイクル全体を通じた自動化導入（2/2）

**自動化導入では「構築自動化」と考えるではなく「IT自動化」と考えて、
システムのライフサイクル全体を通じた自動化を計画すべきです。**

自動化導入で1サーバあたり5分の作業になっているのなら、
100台のサーバでもパラレル動作させれば最短5分で全ての作業が完了します。

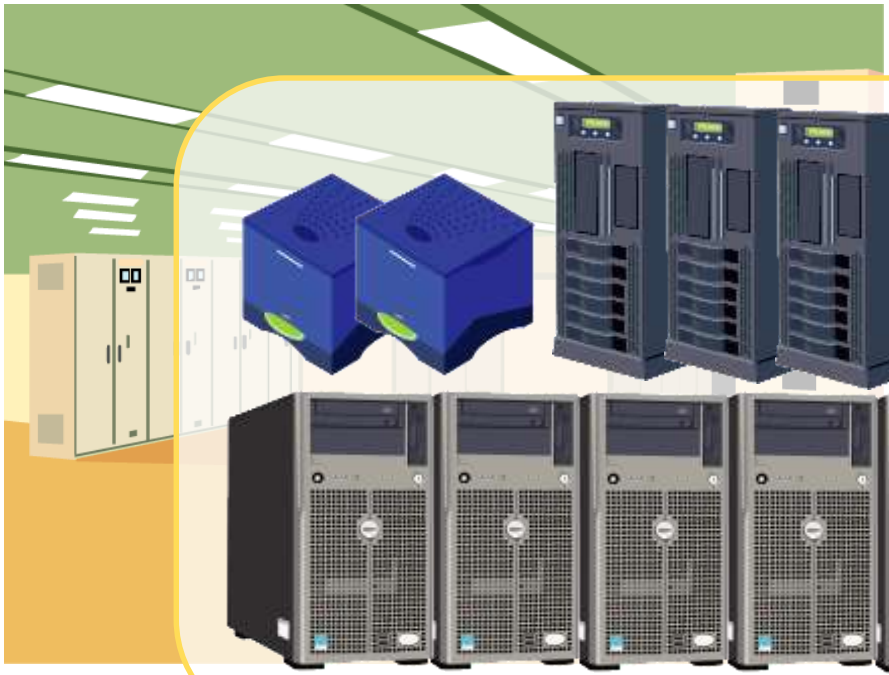


導入時にコストが掛かったとしても、後から保守運用作業などで回収可能です。

“心の壁”も打破する

自動化導入の壁を打破しても“心の壁”はなかなか崩せません。
今の時代もキーパーソンと会話を積み重ねることの重要性は変わりません。

インフラ自動化の導入には、多岐に渡るサーバの内部情報が必要になります。
そのため特に既存システムへの導入時、キーパーソンを味方につけるのは非常に重要です。



【設計・開発フェーズ】

開発体制の考え方（1/2）

ごく一部の有識者が全てを抱え込んで疲弊するケースがあまりに多いため
一極集中させない開発体制になるような配慮が必要です。

「あいつに聞けば大丈夫」「あいつが良く知っているから」「あいつが作ったから」
これらは一極集中してしまっているサインです。

NTT DATA

開発体制の考え方(2/2)

開発チーム自身で自動化スクリプト開発できるようメンバを教育して
自動化全体統括チームが開発チームの成果物を統合する体制にします。

自動化スクリプト
開発チーム
(サブシステム毎)

着任

成果物

自動化講習会

受講

新規メンバ

講師

自動化全体統括チーム

引渡し

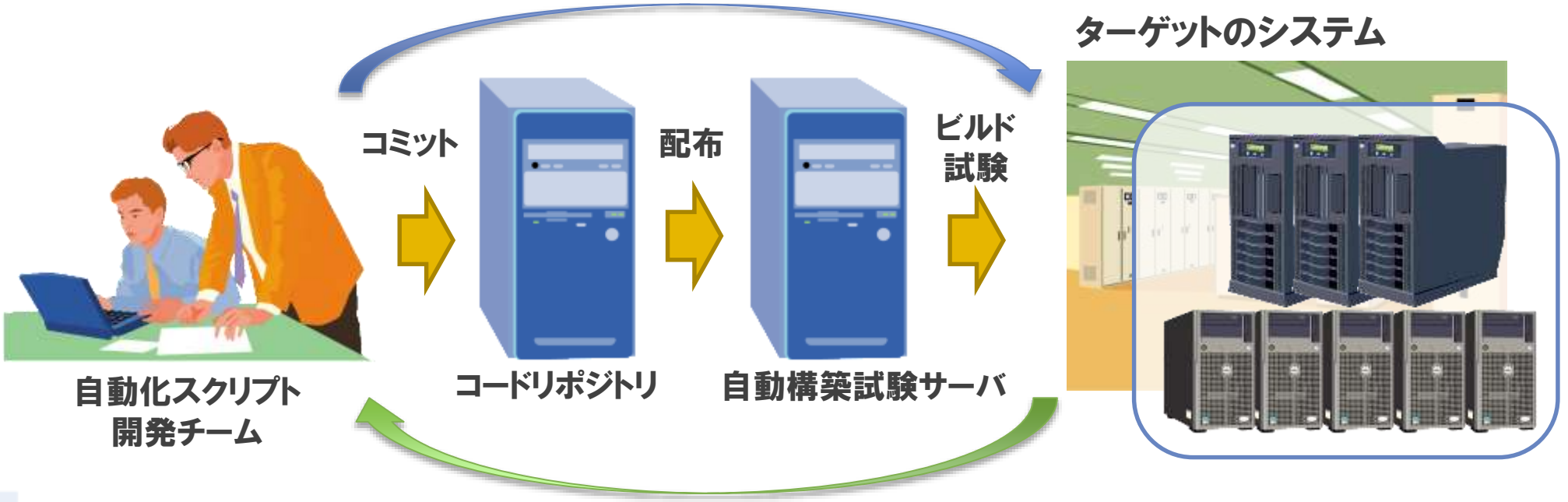
運用チーム

フィードバック

ソフトウェア開発のプラクティスを取り入れる工夫

インフラ自動化のスクリプト類は“アプリケーション”と認識することで、一般のアプリケーション開発のプラクティスを取り入れることができます。

バージョン管理やNightlyリリースなど、従来ソフトウェア開発で実践されてきたプラクティスをインフラ分野にも導入することが可能になります。



ツールを正しく評価する（1/2）

フラットな視点でツールを評価し、ツールの得意分野を理解した上で、適用範囲を決めてシステムのツールの役目を定義します。

ツールの導入が目的ではなく、ツールで課題を解決することが目的です。
ツールの特性を理解し課題を解決するのに適したツールを選定することが重要です。

ツールを正しく評価する（2/2）

フラットな視点でツールを評価し、ツールの得意分野を理解した上で、適用範囲を決めてシステムのツールの役目を定義します。

Our Solution

以下の3点を重視してツールを選定しました。

1. 実現したい自動化の範囲とツールが得意とする分野の重なり具合
2. 社内でのツールの利用実績とプロジェクトメンバのノウハウ蓄積具合
3. 自動化スクリプトの可読性の高さと記述のしやすさ

ツールの優劣ではなくツールの特性を理解した上で選定するよう心がけましょう。

インフラ自動化専用の開発環境の必要性（1/2）

自動化スクリプト類は専用の開発環境を準備するのがベターです。
検証環境や試験環境と相乗りすると環境の取り合いが発生しがちです。

インフラが使うのは最初だけだから相乗りにしてしまいがちですが、
試験、構築、運用保守の各フェーズでも開発環境を使うシーンは必ずあります。

自動化用開発環境

分離

インフラ用検証環境

スクリプト開発環境と検証/試験環境とを分離することによって、
環境の取り合いでスケジュール遅延を起こすリスクを回避できます。

インフラ自動化専用の開発環境の必要性 (2/2)

自動化スクリプト類は専用の開発環境を準備するのがベターです。
検証環境や試験環境と相乗りすると環境の取り合いが発生しがちです。

Our Solution

仮想マシンで本番環境の自動化ツールのみ動作する環境を構築し、
開発中に環境を壊しても即座に修復できる状態にして開発しました。
試験環境については実機と同じ構成で環境を構築して試験しました。

NTT DATA

要員育成を考慮した計画

コーディングスキルのあるインフラ要員確保は困難なケースが多いため
要員の育成を考慮した計画を立案すべきです。

インフラ用の自動化スクリプト開発は汎用言語を使えるだけでは難しいです。
無理に探すよりも自前で育成した方が結果的に効率がよい場合もあります。

Our Solution

- 有識者によるプロジェクトメンバに対する自動化ツール研修を実施
- 受講者は自動化スクリプト開発の実地演習を実施後に開発着手
- コーディング規約を規定して一定のコード品質を保つ工夫

STEP1: 有識者による研修

STEP2: 実地演習

STEP3: 本格着手

“車輪の再発明”はしない

インフラ自動化を導入する目的は“ラクをすること”です。
自作にこだわらず出来合いのモジュールなどを積極的に使うべきです。

一般的にアプリケーション開発では便利なライブラリが無いか最初に探します。
同様に自動化スクリプトでもまず最初に出来合いモジュールが無いか探すべきです。

Our Solution

メンバ内にノウハウが少ないパッケージのモジュールを中心として
“Puppet Forge”で出来合いのモジュールを検索し、レーティングの
高いモジュールを選んで活用しました。

ゼロから開発すると
全てのモジュールの開発と試験も
実施しなければなりませんので
手間も時間も掛かります。

スクラッチ開発

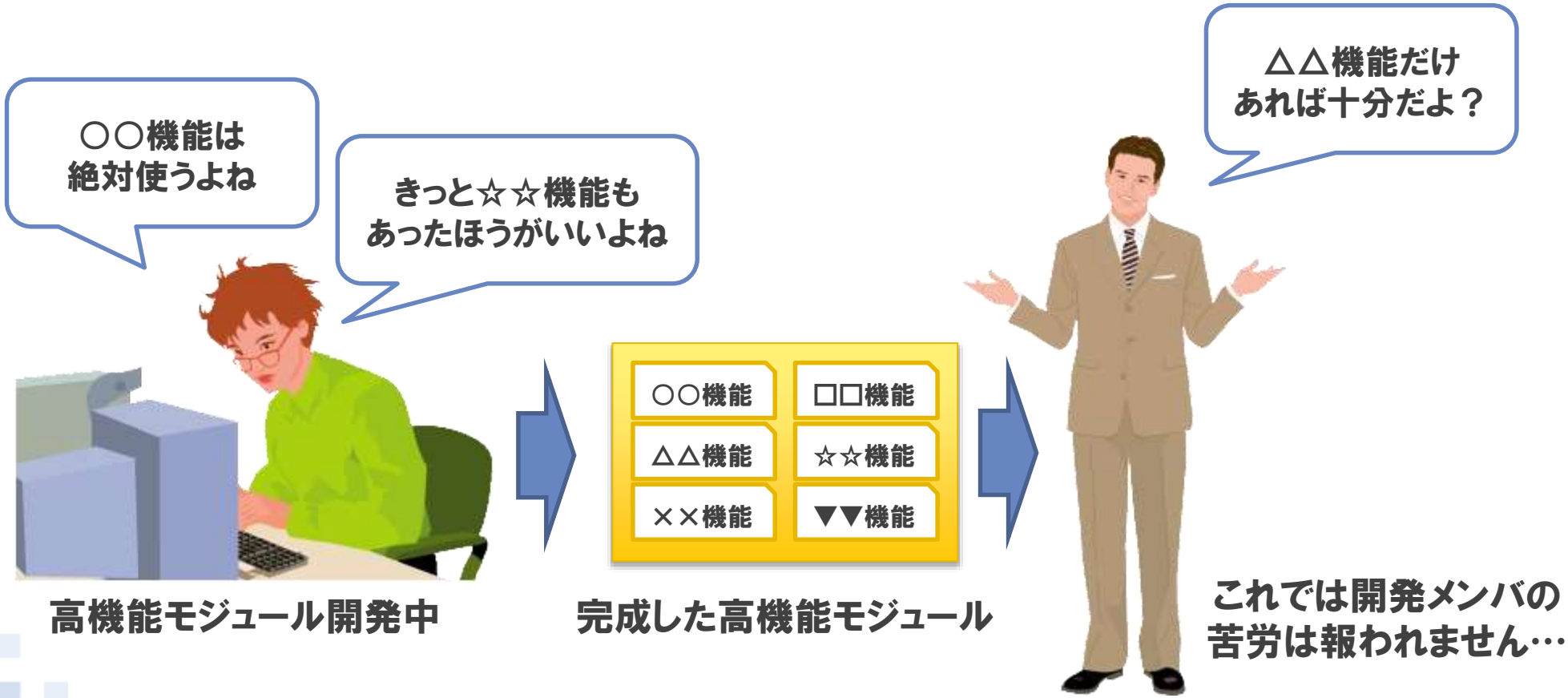
出来合いのモジュールを使えば
モジュール単体の試験は省略し
結合試験のみで済むため
手間も時間も削減できます。

出来合いモジュールでの開発

最初は必要最小限の機能から（1/2）

高度な機能を提供しても使わないケースは多々あります。
ラクをしたいなら機能を実装することで自己満足に陥ってはいけません。

使いそうな機能を予想しながら開発すると必要以上に高機能化してしまいがちですが、
導入に掛ける手間と得られる効果のバランスをよく考えて機能は実装すべきです。

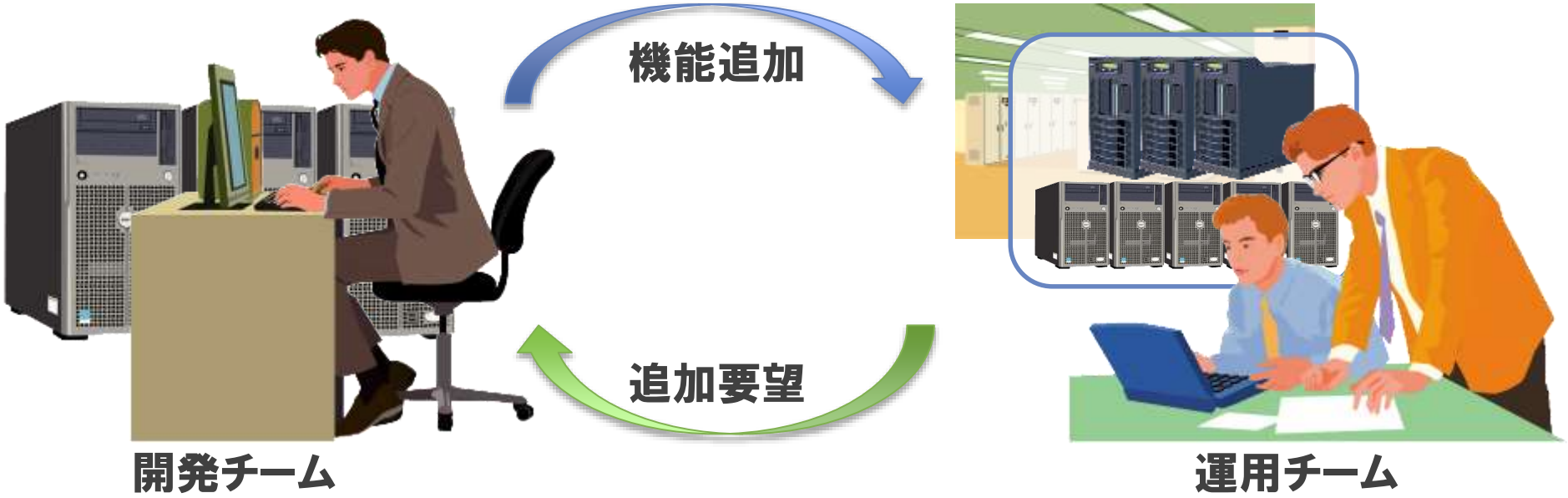


最初は必要最小限の機能から(2/2)

高度な機能を提供しても使わないケースは多々あります。
ラクをしたいなら機能を実装することで自己満足に陥ってはいけません。

Our Solution

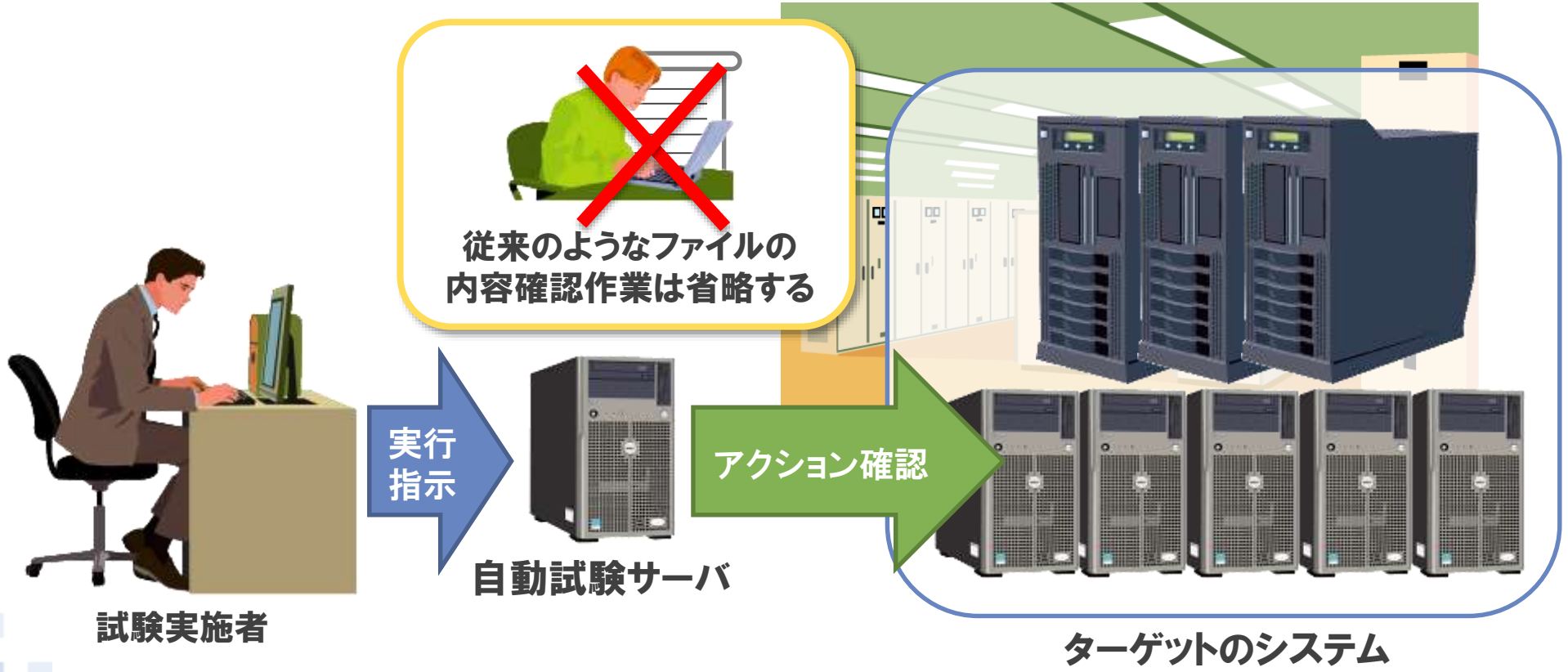
- 必要最小限の機能のみ実装して初回はリリース
- 必要に応じた機能拡張が容易となるモジュールの粒度や設計指針



真に実施すべき試験項目とは（1/2）

自動化ツールで設定したシステムの試験は設定を確認するのではなく
その設定で定義されるアクションが正しく発生しているかを確認します。

特定のポートをLISTENしているかなど、正しい設定で発生するアクションを試験します。
ツールで配布した設定ファイルの中身を確認するような確認はナンセンスです。



NTT DATA

真に実施すべき試験項目とは(2/2)

自動化ツールで設定したシステムの試験は設定を確認するのではなく
その設定で定義されるアクションが正しく発生しているかを確認します。

Our Solution

- Puppetで配布する設定ファイルのパラメータの確認は省略
- 手作業で設定される項目のみ設定値確認を実施
  - OpenStackではTempestを使用して試験を実施
  - OpenStack以外のサーバではServerspecを使用して試験を実施
- 結合試験は従来通りの方法で実施
  - Serverspecでは複数サーバをまたがる試験の定義が難しいのが理由

【まとめ】

まとめ

既存プロセスを単純に置き換えるだけでは効果が出ません。
自動化に最適化したプロセスとしてこそ大きな効果を得られます。

- 自動化の導入効果を正しく理解する
- アプリケーション開発と同様に考える
- 無駄な試験をしない

日本Puppetユーザ会について

3ヶ月に1回の頻度でPuppetユーザ会を開催しています。
インフラ自動化に興味がある方はぜひご参加ください。

https://japanpuppetusergroup.connpass.com/

日本Puppetユーザ会のメンバーが執筆した
Puppetを中心としたインフラ自動化に関する書籍が間もなく発売！

NTT DATA

NTT DATA

Global IT Innovator