

NTT DATA

AWS Summit 2025 ~Day2~

laCを活用したNTTデータ流一気通貫高速開発

2025/06/26

技術革新統括本部
Cloud & Infrastructure技術部
長谷川 樹

自己紹介



■ **所属**： 株式会社NTTデータグループ
技術革新統括本部 Cloud & Infrastructure 技術部

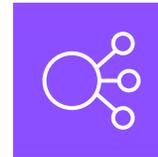
■ **入社**： 3年目

■ **業務内容**： クラウド案件の開発
→ 1年目からSREチームメンバーの人財育成にも取り組む。

■ **好きなAWSサービス**



AWS Lambda



Elastic Load Balancing



目次

1. IaC活用に向けて
2. 人財育成への取り組み
3. 仕組みづくりへの取り組み
4. まとめ



目次

- 1. IaC活用に向けて**
2. 人財育成への取り組み
3. 仕組みづくりへの取り組み
4. まとめ

突然ですが...

皆さんは業務の中で、IaCを活用できていますか？？

組織でのIaC活用には人財と仕組みが必要

IaC：構成の一貫性・再現性の確保、ビルドデプロイの工数削減による開発高速化が期待される。

IaCを活用する上で必要な要素

IaCを使いこなすスキルをもつ**人財**がいること

プロジェクトの制約に合わせた
開発の仕組みが整備されていること

設計から運用まで **一気通貫で** 実施できる人財・仕組みを作ることが重要。

アプローチ①：人財育成プログラム

アプローチ②：テンプレートや自動化ツール

これらの課題に対して、私の経験やNTTデータの取り組みについてご紹介します。



目次

1. IaC活用に向けて
- 2. 人財育成への取り組み**
3. 仕組みづくりへの取り組み
4. まとめ

運用までを見据えたlaCの実装が難しい

Q. laCコードの文法を理解すればlaCを活用できる... ?

→ A. No!!! ただパラメータを踏まえて実装すればよいのではなく、**運用しやすいlaCの実装**が重要。

「作ってはみたけど運用しづらくて、結局コードを破棄する」なんてことも...

- ・リリース方法
- ・コードの管理方法
- ・システムの運用方法
- ・コードの拡張性

などなど...

Q. なぜ運用しにくいコードができあがってしまうのか... ?

→ A. 設定するパラメータと文法がわかれば実装できてしまうため、システム方式設計や運用方法などの検討と独立して実装してしまう。

システム方式設計の時点から今後のlaCコードの運用方法を考慮し、laCの実装につなげていくことが重要。



実装

パラメータ

コードのロジック
(文法・コード規約など)

事前に考えるべきIaC自体の再利用性やメンテナンス性を考慮できない

リリースサイクルを加味したコードの配置場所

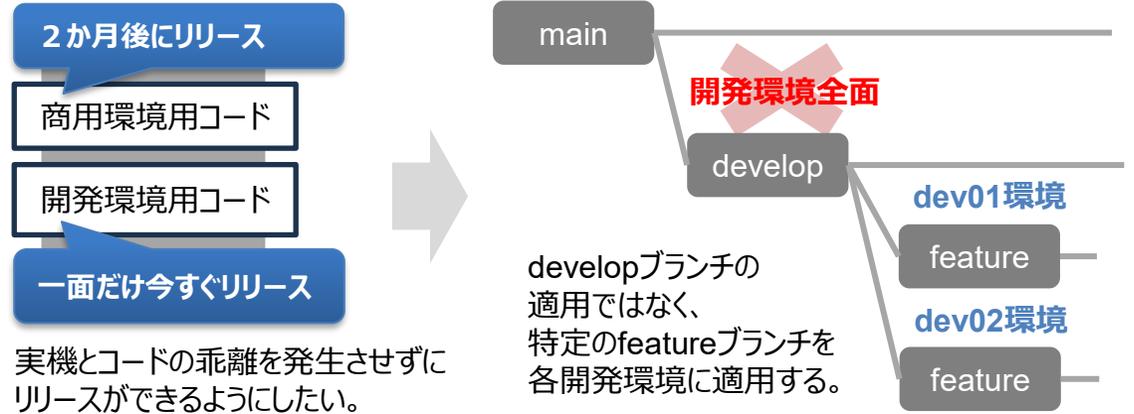
- Module A
 - Lambda
 - src
 - OO.py
- Module B
 - Amazon S3
 - S3バケット
 - オブジェクト
 - Data
 - △△.csv

インフラメンバーが3か月に一回程度アップデート

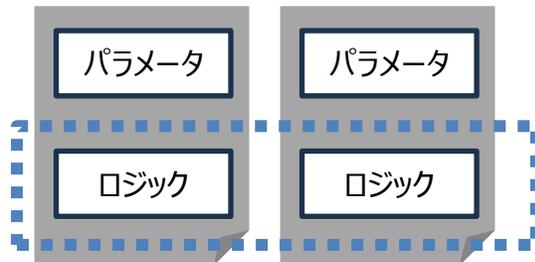
同じレポジトリ内で定義する?

アプリメンバーが1か月に一回程度アップデート

運用方法を踏まえた各環境面に対するコードの対応関係



機能A 機能B



- ✓ 今後類似機能を実装するときはどのように実装するか
- ✓ 同一モジュールで類似機能を作成する場合、システムに制約が発生するか

共通化

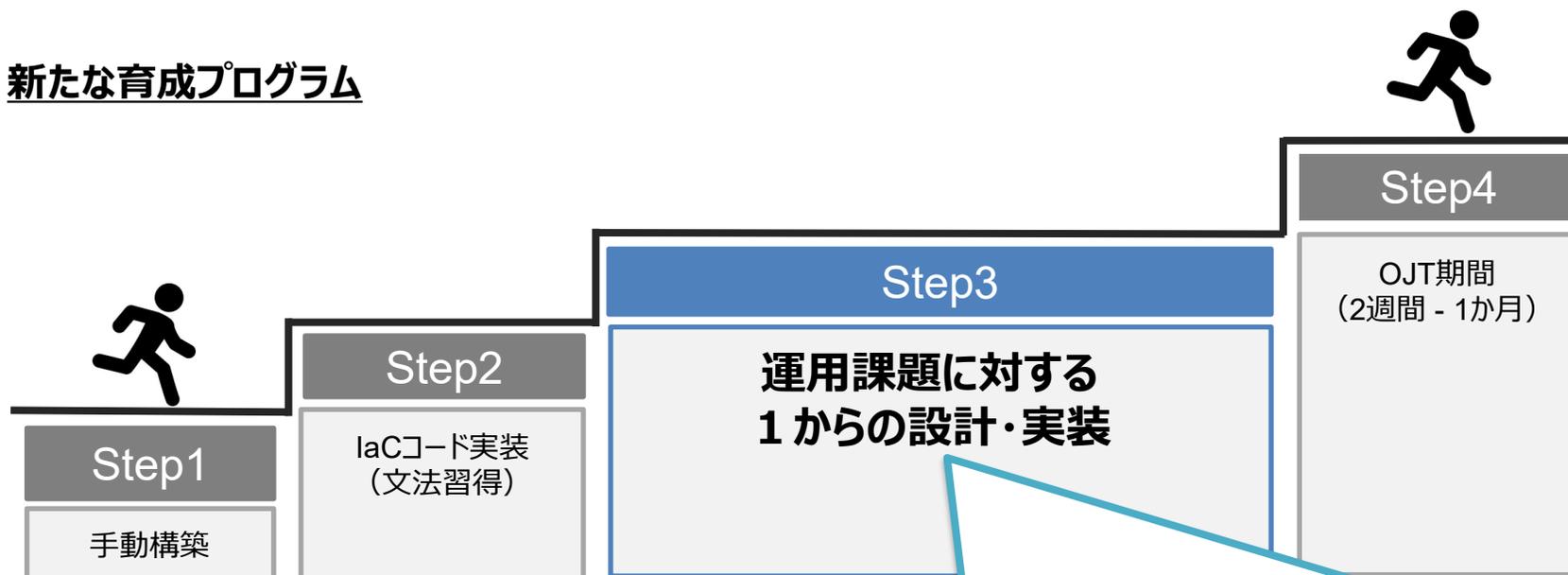
IaCコードの拡張性を加味したシステム方式設計

単純なIaC実装の練習だけでなく、実際の運用を見越して設計する経験が重要。

育成プログラムの作成

設計から運用まで一気通貫の開発経験を積む育成プログラムを作成

新たな育成プログラム



研修内容

実現場で直面している **運用改善の課題** をテーマとして、
実際のシステム・laCコードをベースに、**設計から一気通貫**で実施してもらう。



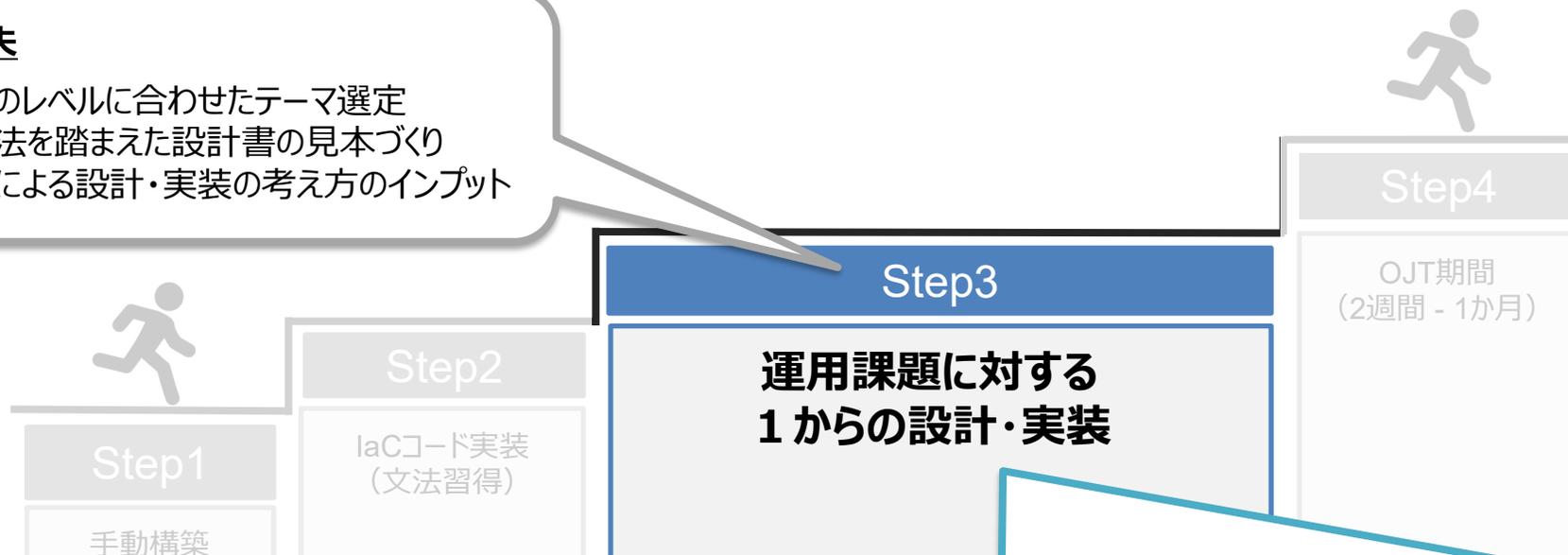
【課題】
DBのスロークエリログ改善
EOSL検知改善

育成プログラムの作成

一気通貫で開発ができる人材を育てる方法を確立してチームを育成

運営の工夫

- ✓ 受講者のレベルに合わせたテーマ選定
- ✓ 運用方法を踏まえた設計書の見本づくり
- ✓ ペアプロによる設計・実装の考え方のインプット



- ・実装経験のみの参画者に、設計の進め方を習得してもらうことができる。
- ・保守運用性・拡張性の高いIaCコード実装を念頭に一から設計してもらうことで、考え方習得のきっかけを作ることができる。
- ・構築後に短期間利用してみて運用改善を検討してもらうことで、設計時の検討事項をより考えてもらうことができる。

➡ 実際の運用を見越してIaCの実装を検討できるように。

実装のみでなく、「IaCを用いて設計から運用まで一気通貫で実施できる人材」を育てる方法を確立。

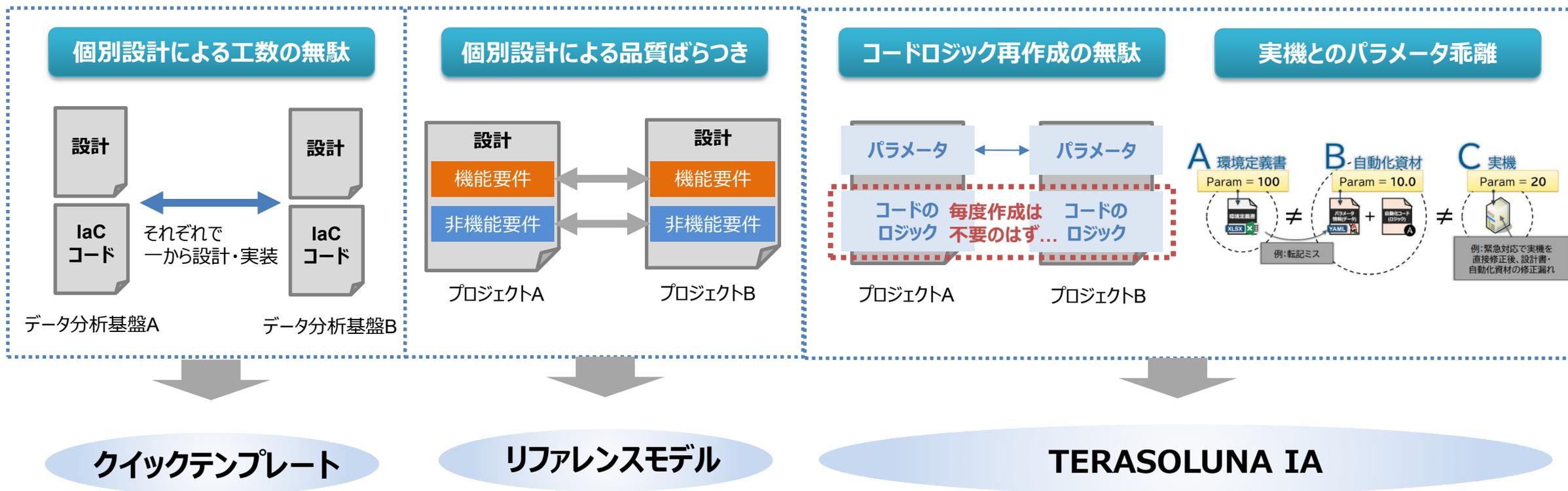


目次

1. IaC活用に向けて
2. 人財育成への取り組み
3. **仕組みづくりへの取り組み**
4. まとめ

さまざまなプロジェクトで再現性をもってIaCを活用するための仕組みを提供

IaCを活用するためのポイント



これらの課題を解決するために、NTTデータ独自のソリューションを用意。

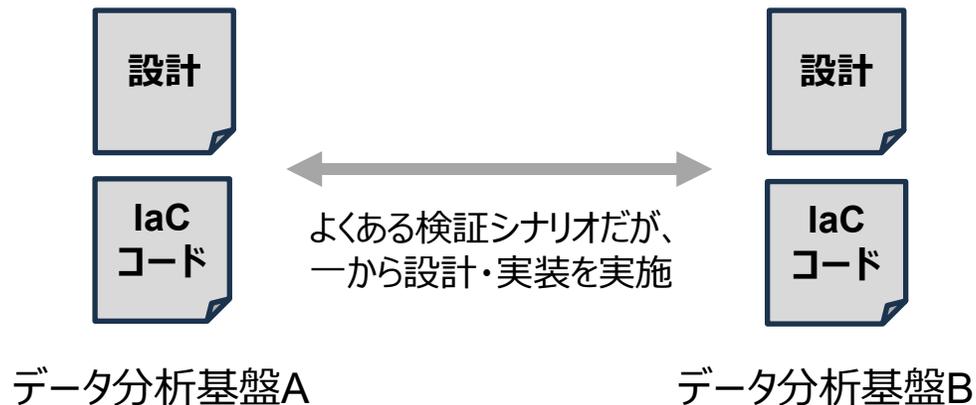
個別設計による工数の無駄・品質ばらつき改善

個別設計による工数の無駄

データ分析基盤・モバイルアプリなどのよくある構成を検証用に構築したい場合、
一から設計・実装すると工数が無駄になってしまう。



検証用に使える設計・laCコードのテンプレートがあれば...

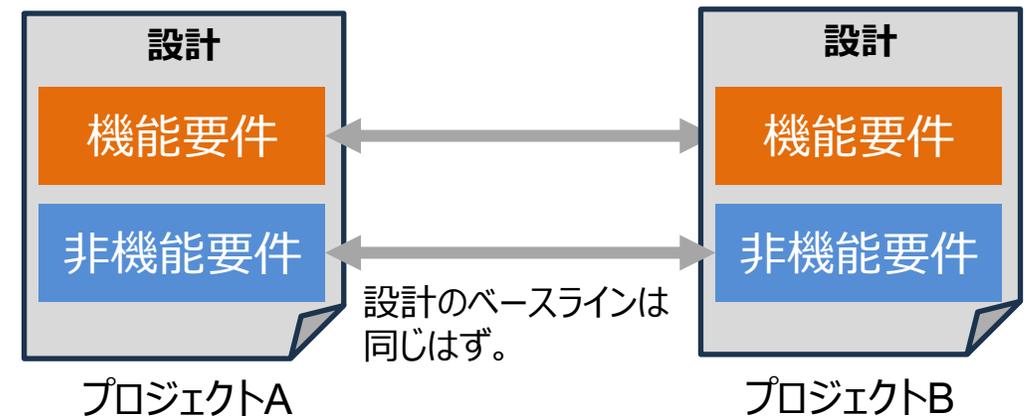


個別設計による品質ばらつき

プロジェクトごとに一から設計してしまうと、
システムの品質にばらつきが生まれてしまう。



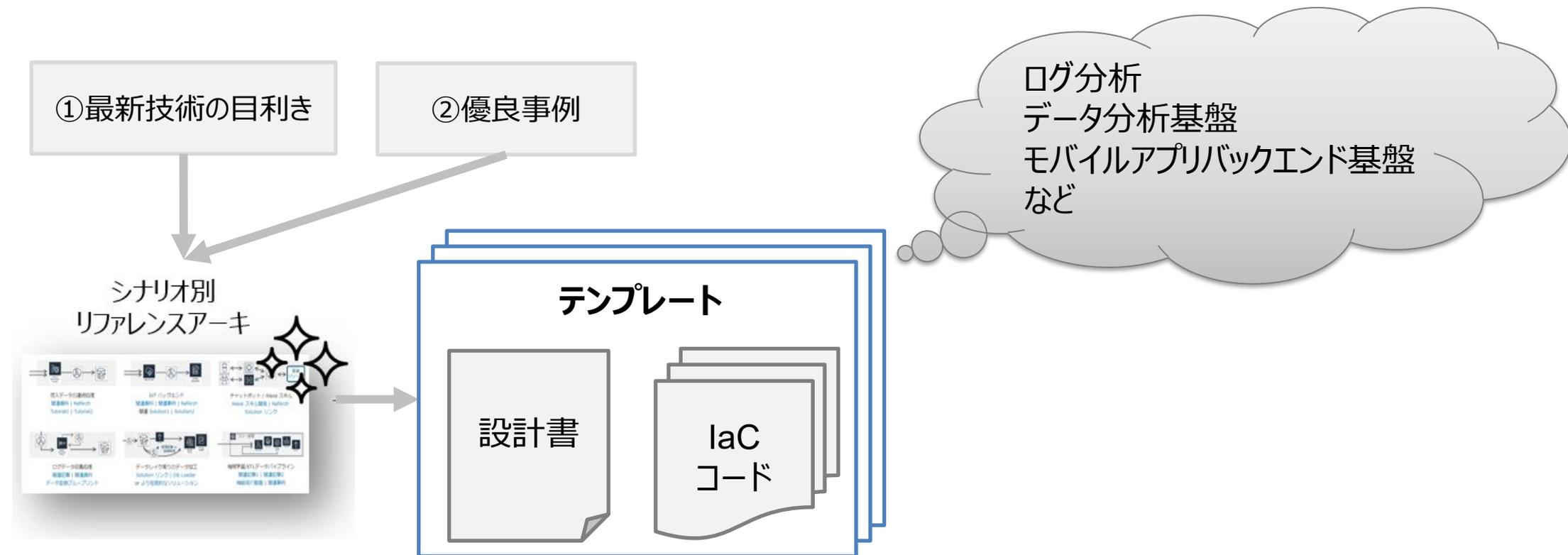
成功実績のある開発のベースラインがあれば...



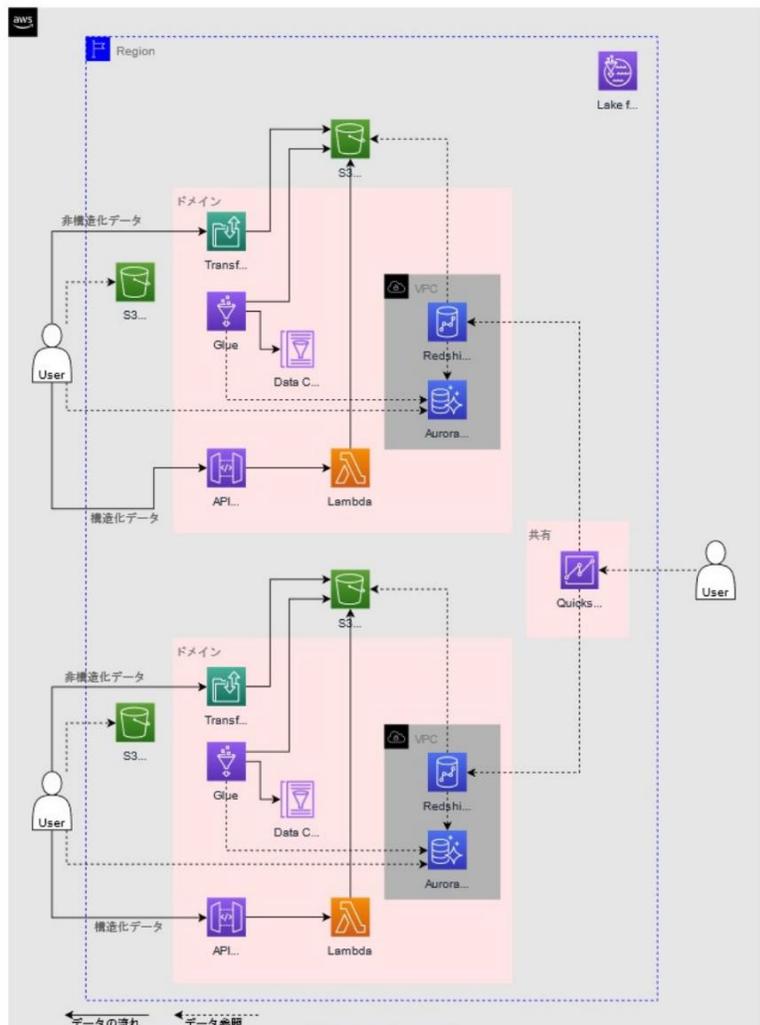
クイックテンプレート

特定シナリオ用の「テンプレート」で手軽にPoCの開始が可能に

クイックテンプレート：ログ分析・データ分析といった特定のシナリオにおける設計がセットになったIaCテンプレート
→ 特定システムのPoCに要する工数を大幅に短縮。



データ分析基盤のPoCに要する設計・実装工数を削減



概要設計書

課題

よくある典型的なデータ分析基盤をPoCしたいだけなのに、一からETL処理の設計が必要で、工数がかかる。



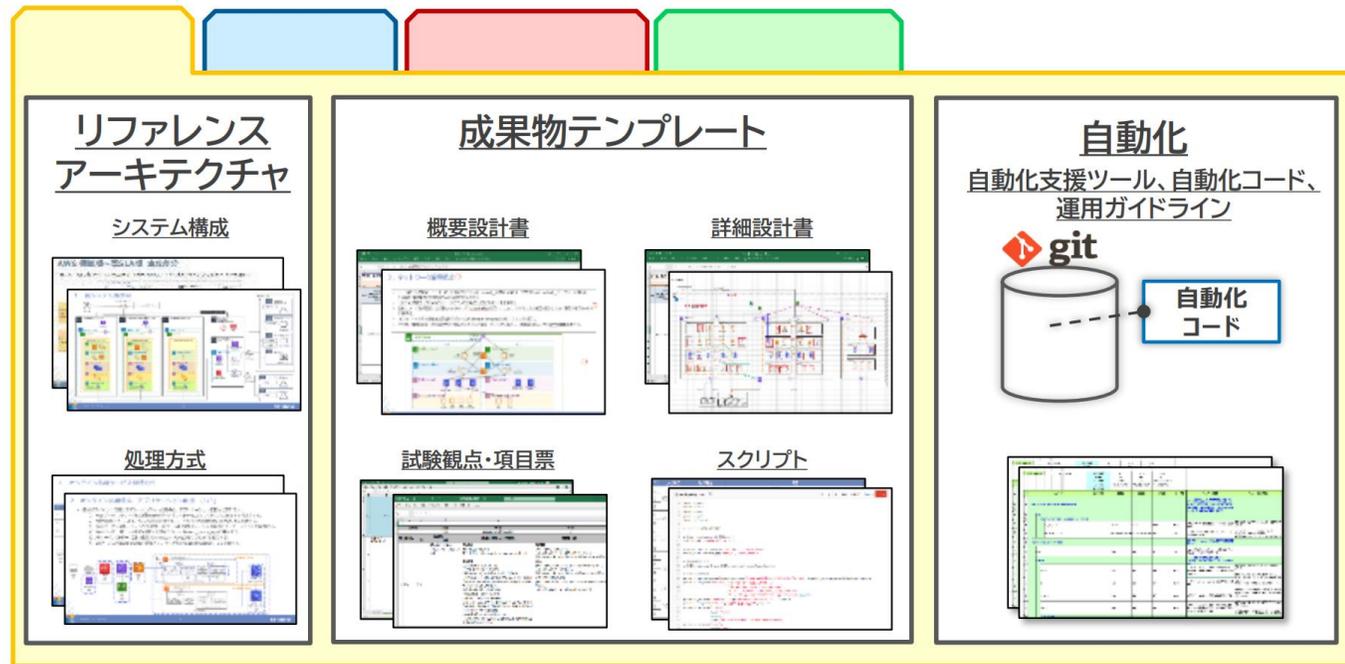
事前準備されたテンプレートを利用することで、PoC時の設計・実装にかかる工数を削減。



すぐにリソース構築・検証を行うことができる。

開発のベースラインとなる「テンプレート」を用いて、高品質・高生産性を実現

リファレンスモデル：プロジェクトの実績に基づき、要件定義からテストまでを一気通貫で集約したテンプレート



- ✦ 各開発フェーズの工数を削減。
- ✦ 実績のあるモデルの利用による品質向上。

社内の複数プロジェクトで生産性向上を確認。

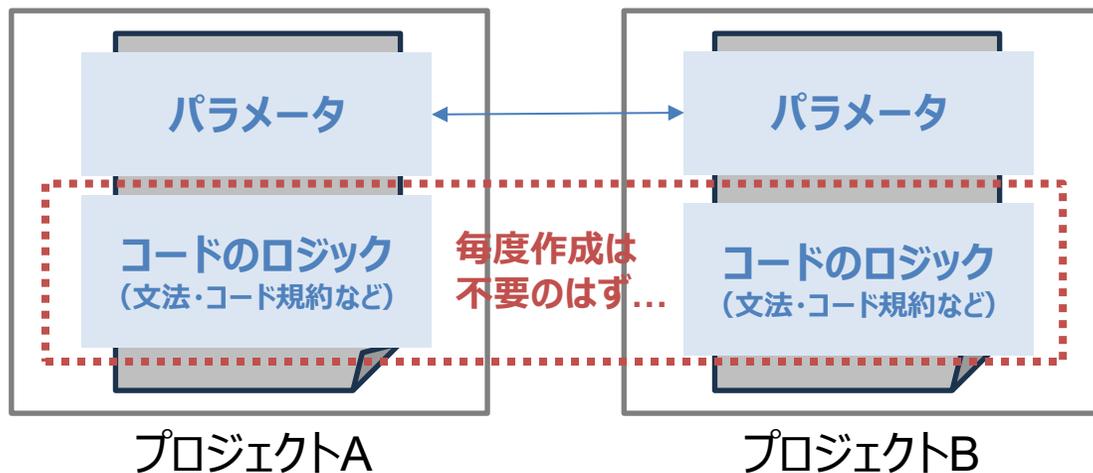
コードロジック再作成の無駄・実機とのパラメータ乖離の改善

コードロジック再作成の無駄

laCでやりたいことはパラメータの管理で、
コードのロジックは変わらないはずだがーから作成している。



コードロジック部分を再作成することなく、
パラメータの選定だけに注力することができれば...

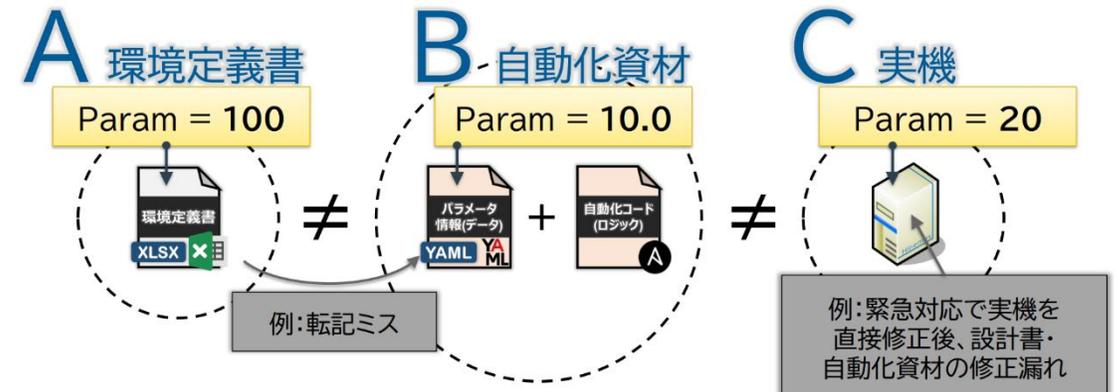


実機とのパラメータ乖離

laCコードとは別にパラメータ管理用のドキュメントを作成することがあるが、実機とのパラメータ乖離や資料作成分の工数増加が発生してしまう。

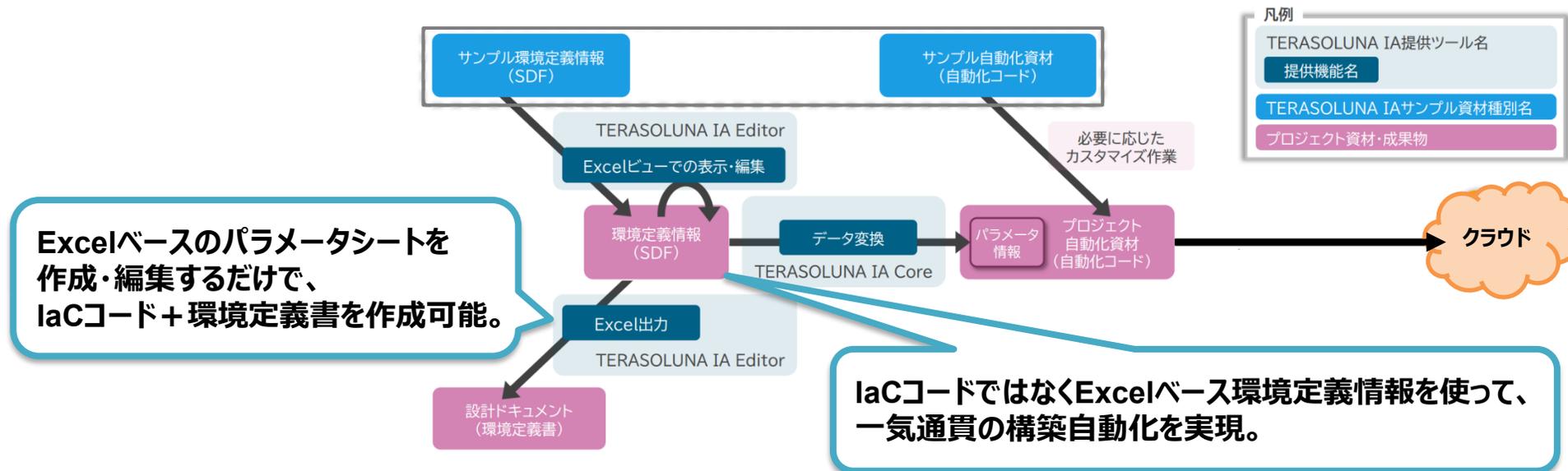


管理用ドキュメントの手動管理が不要になれば...



laCコード自動生成ツールで、作業負荷・実機との乖離を削減

TERASOLUNA IA：可読性が高く編集が容易なExcel形式をインプットとしたlaCコード自動生成ツール



課題①
コードロジック再作成の無駄

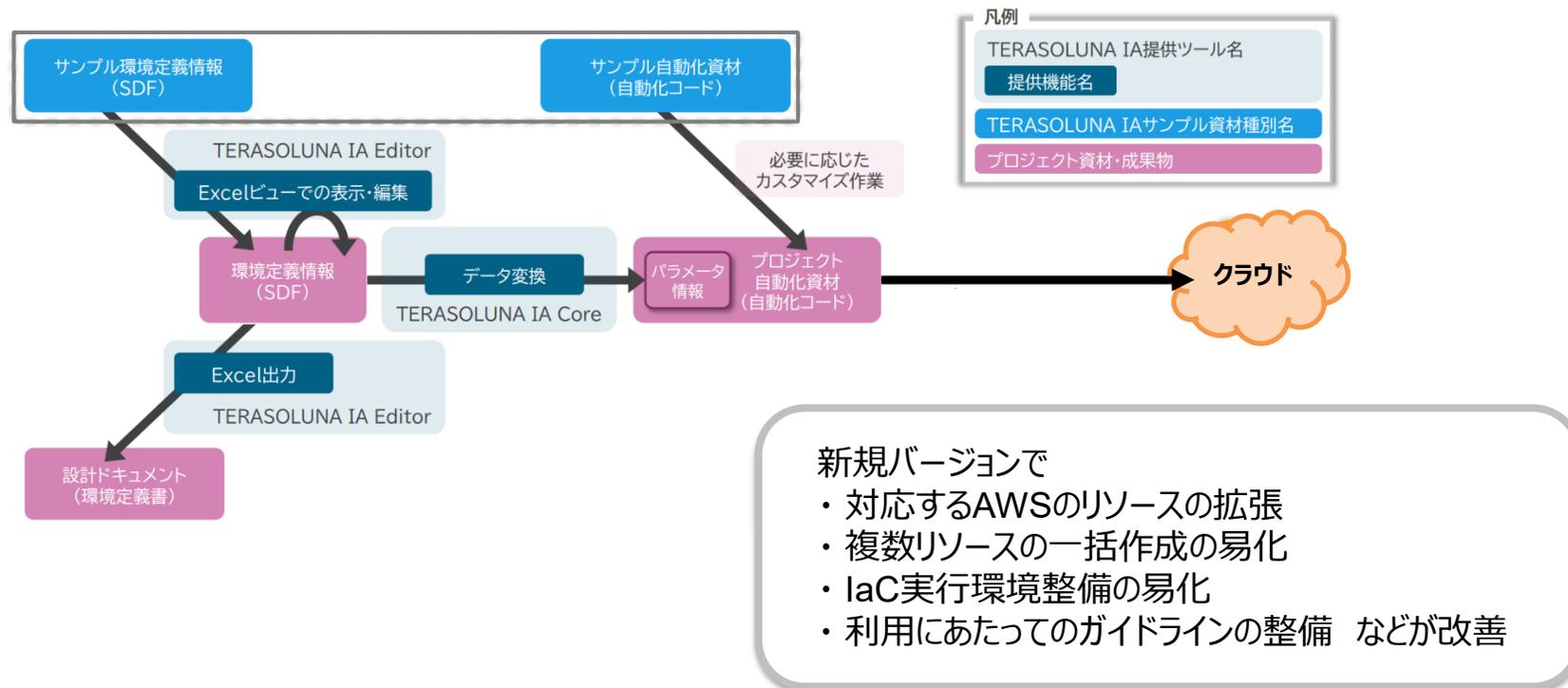
Excelベース環境定義情報を作成すれば自動でlaCコードを作成し、リソースの構築を行うことができるため、コードロジックの検討が不要に。

課題②
実機とのパラメータ乖離

Excelベース環境定義情報をもとにlaCコードの作成・実機への適用・設計ドキュメントの作成が一気通貫で実施され、パラメータ乖離を防止できる。

新規バージョンでは自動生成ツールの構築がさらに簡単に

TERASOLUNA IA：可読性が高く編集が容易なExcel形式をインプットとしたIaCコード自動生成ツール



AWS環境でサーバレスのインフラCI/CD基盤の構築とその基盤による自動構築を簡単に実現できるように。

プロジェクトごとに最適なソリューションを見極め、適用し、サポート

設計から運用まで一気通貫でサポートするソリューション

クイックテンプレート

特定シナリオをクイックに構築するためのIaCテンプレート

リファレンスモデル

設計・実装がセットになった開発のベースライン

TERASOLUNA IA

パラメータシートをインプットすることでIaCコードを自動生成できるツール

このプロジェクトでIaCを活用したいんだけど...

このソリューションを使いつつ、プロジェクトに合った形で最適化すれば...

- ✓ プロジェクトの制約や課題に合わせて最適なソリューションを見極め、提案。
- ✓ 適用にあたって設計から運用まで一気通貫でサポート。





目次

1. IaC活用に向けて
2. 人財育成への取り組み
3. 仕組みづくりへの取り組み
- 4. まとめ**

laCやクラウドでお困りごとあれば、ぜひご相談ください

laC：構成の一貫性・再現性の確保、ビルドデプロイの工数削減による開発高速化が期待される。

laCを活用する上で必要な要素

laCを使いこなすスキルをもつ**人財**がいること

プロジェクトの制約に合わせた
開発の仕組みが整備されていること

設計から運用まで **一気通貫で** 実施できる人財・仕組みを作ることが重要。

アプローチ①：人財育成プログラム

アプローチ②：テンプレートや自動化ツール

NTTデータは**豊富な案件実績からの知見**をもとに、**顧客の「実現したい」**を全力でサポートします！

The image features a low-angle, wide shot of a modern city skyline under a clear blue sky. Two prominent skyscrapers with white facades and dark window bands are the central focus. Other buildings of varying heights and architectural styles are visible in the background and foreground. The overall scene is brightly lit, suggesting a clear day. The text 'NTT Data' is superimposed in the center of the image.

NTT Data

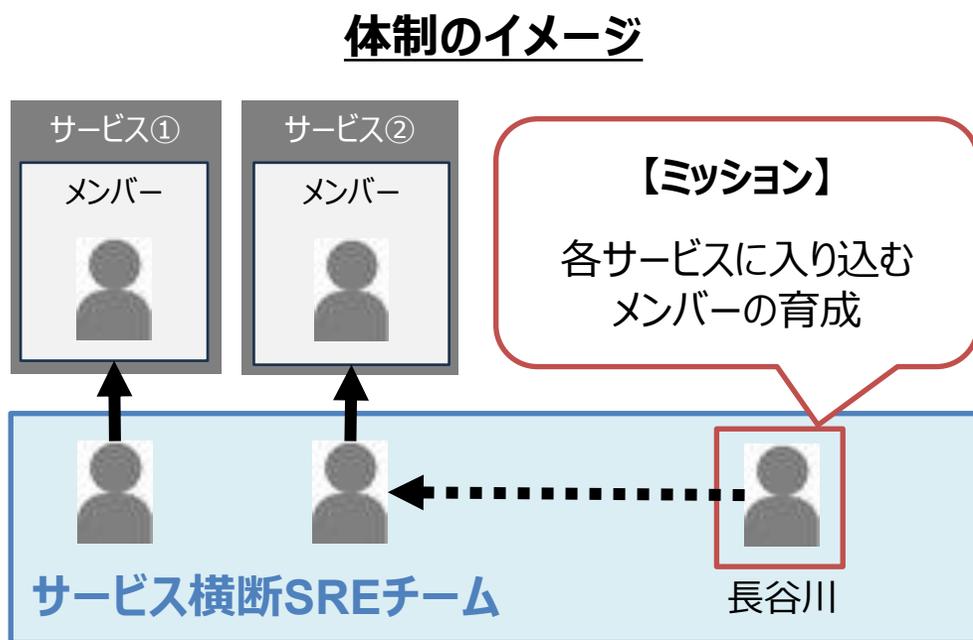
補足



人財育成のハードル

メンバー育成にあたり、IaCの習得がハードルに

- 私たちの組織では、複数のサービスでSREの文化を根付かせるために、横断的なSREチームから各サービスにチームメンバーを派遣する体制を採用。
- プロジェクトの都合で新規メンバーの獲得ではなく、既存メンバーを育成するという方針をとっていた。

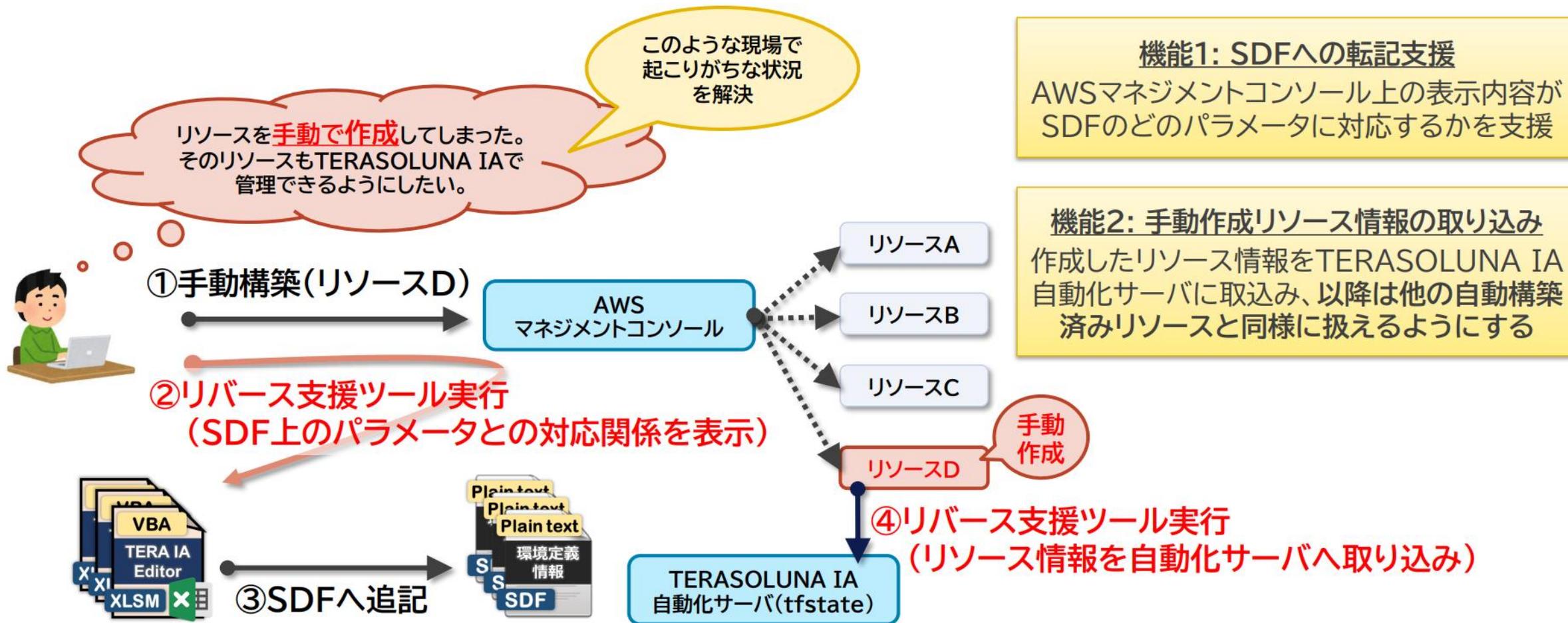


ゴール	各サービスの中でトイル削減や開発・運用改善を主体的に検討することができる。		
ハードル	【技術】 ・AWS ・開発ツール ・IaC	【人】 ・チームメンバーとの関係性 ・ステークホルダとの関係性	【サービス特性】 ・対象サービスのドメイン知識の理解 ・対象サービスの開発スタイルやルール

IaCの習得が大きなハードル

IaCの未経験者が多く、SREメンバーとしての立ち上がりに苦労しているメンバーが多かった。

TERASOLUNA IA (リバース支援ツール)



TERASOLUNA IA (従来版 vs CDKTF版)

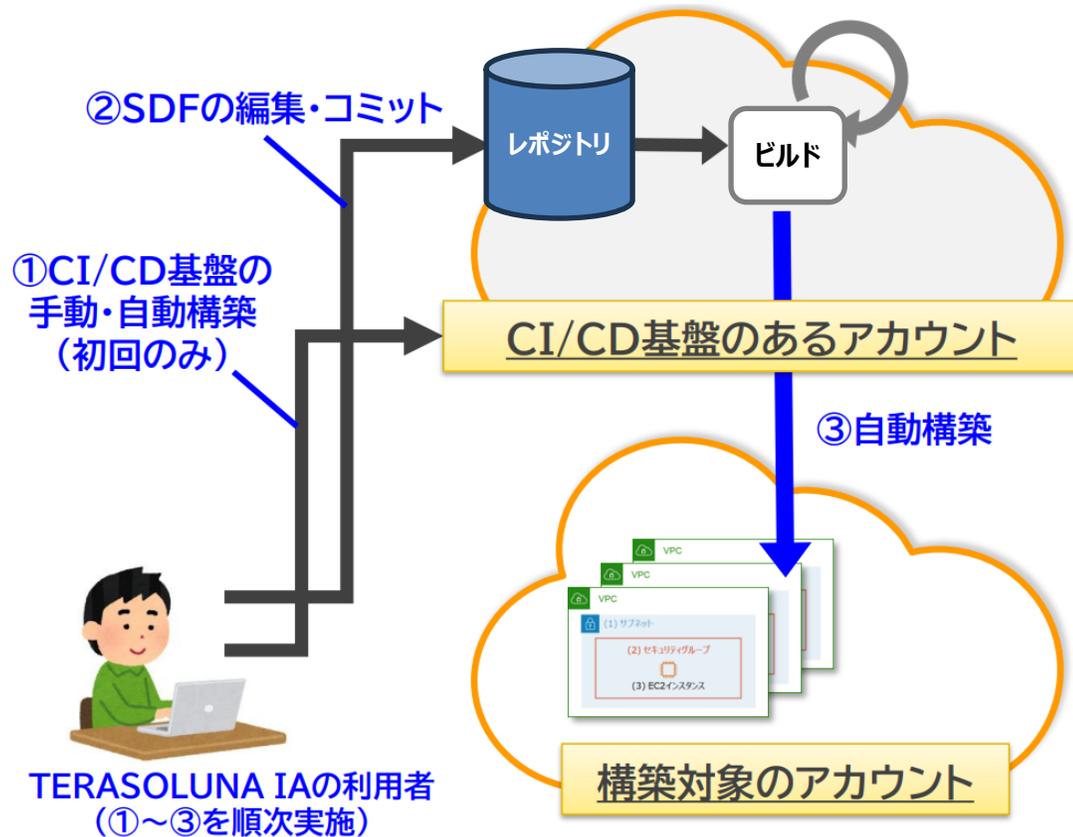
観点分類	観点詳細	CDKTF版(※1) v1.1.0	従来版(※1) v1.9.0
アーキテクチャ	複数種類リソースの一括作成	✓ 可	✗ 不可
	手動作成リソースの参照	✗ 不可	✓ 可
	サーバレスによる実行	✓ 可 (親和性は非常に高い)	✓ 可 (親和性は低い)
	インフラCI/CD	✓ 可 (親和性は非常に高い)	✗ 不可 (逐次的承認が必要)
	外部TFコードとの統合	✓ 可	✓ 可 (親和性は低い)
開発状況により変動	対応リソース数(※2)	98リソース (今後も増加予定)	155リソース (今後の開発は保留)
	リバース対応(※2)	✗ 不可 (開発予定は未定)	✓ 可
その他	Providerの指定	スタック単位 (全体で単一バージョンを使用)	リソース単位 (全体で異なるバージョンの混在が可)
	tfstateの管理単位	スタック単位で管理	リソース単位で管理

※1 CDKTF版 = TERASOLUNA IA for AWS (CDKTF版)、従来版 = TERASOLUNA IA for AWS

※2 2024年9月時点での情報

新バージョンではIaCによる構成管理がよりやり易く

2024年にCDK & Terraform対応バージョンがリリース。



新バージョンで追加されたメリット

- ✓ サーバレスにより自動化サーバが不要に。
- ✓ インフラCI/CDにより、SDF更新と同時に環境変更が可能に。
- ✓ 複数種類リソースの一括作成、変更が可能に。