



モダナイゼーションプロジェクト 立ち上げのポイント

Modernization 入門シリーズ

平岩 梨果

Solutions Architect
2024/1

自己紹介

名前：平岩 梨果

所属：アマゾン ウェブ サービス ジャパン合同会社
マイグレーション&モダナイゼーション
シニア ソリューション アーキテクト

経歴：外資系SIer にてIT アーキテクト

好きなAWSサービス：AWS Lambda



Modernization 入門シリーズ とは

モダナイゼーションについて初学者向けにステップを踏んで解説

更なる続編について順次公開予定

今回はここ



基礎編

モダナイゼーション
とは



モダナイゼーション・
パスウェイ



進め方編

モダナイゼーション
プロジェクト
立ち上げに向けたポ
イント



モダナイゼーション
プロジェクト
立ち上げのポイント

個別技術編

Modernize
Enterprise Java
Application

本セミナーの対象者と目的

対象者

- システム刷新を検討している責任者やリードする方
- システムのモダナイゼーションに取り組んでおり、進め方のポイントを確認したい方

目的

モダナイゼーションプロジェクトを立ち上げる際に、
検討・考慮すべきポイントについて理解を深めていただく

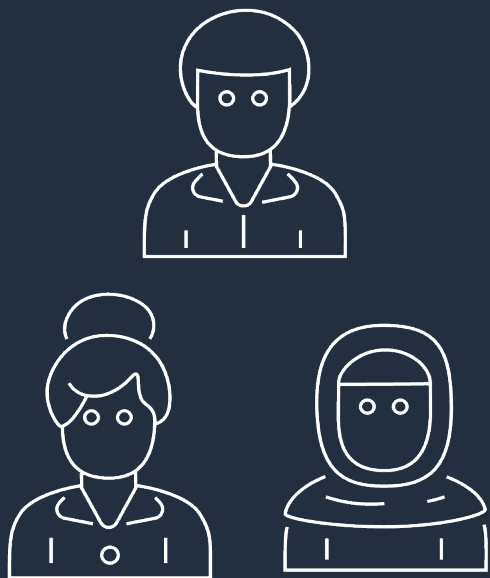
アジェンダ

1. モダナイゼーションの目的
2. モダナイゼーションプロジェクト立ち上げ前のポイント振り返り
3. モダナイゼーションプロジェクトの立ち上げ
4. People
5. Process
6. Technology
7. まとめ

モダナイゼーションの目的

モダナイゼーションの必要性

高い生産性やビジネス俊敏性を備えることで、
ビジネスの加速に貢献する



ユーザーの多様化



手段の多様化



社会や環境の変化

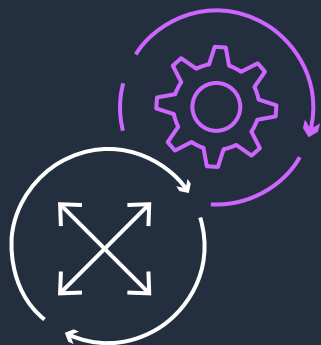
モダナイゼーションの目的と3つの柱

モダナイゼーションとは個別技術要素のことではなく
企業や組織が社会の変化にあわせて価値を提供し続けるために
組織やシステムを常に新しくしていくこと

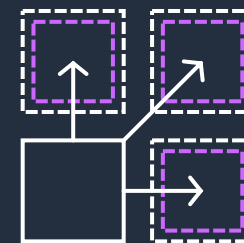
人・プロセス・技術 の3つを軸を組み合わせて推進するとよい



People



Process



Technology

モダナイゼーション プロジェクト立ち上げ前の ポイント振り返り

モダナイゼーションプロジェクト立ち上げに向けたポイント

AWS Black Belt Online Seminar

aws

モダナイゼーションプロジェクト
立ち上げに向けたポイント
Modernization 入門シリーズ

平岩 梨果
Solutions Architect
2023/09

© 2023, Amazon Web Services, Inc. or its affiliates. 0:00 / 31:59

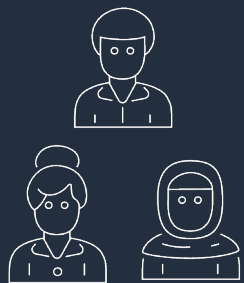
モダナイゼーションプロジェクト立ち上げに向けたポイント 【AWS Black Belt】

https://www.youtube.com/watch?v=mh3SpS_E24s

モダナイゼーションプロジェクト立ち上げの前に

どの対象になにを実施すべきか？

は見えていますか？



ユーザーの多様化



手段の多様化



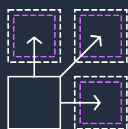
社会や環境の変化



People



Process



Technology



多くのIT資産

モダナイゼーション立ち上げ前の課題と対応策

よくある課題

対応策（検討のポイント）

1.モダナイゼーションは変更の度合いが大きいほど、多くのコストがかかるが、ビジネス目線を意識した合理的な理由付けができていない

ビジネス戦略や価値を念頭に目標設定を行い、KPI に沿った経済性評価を行う

2.“手段”に囚われ、“目的”の共通認識が図れていないまま進めることで、適切な移行方針が描けない

取り組むシステムや機能の現状把握（棚卸し）をした上で、取り組む優先順位を決め、ニーズに応じた適切な移行方針を検討する

3.人材育成や組織改革、マインドチェンジを怠ることで、テクノロジーを正しく活かすためのプランが描けていない。

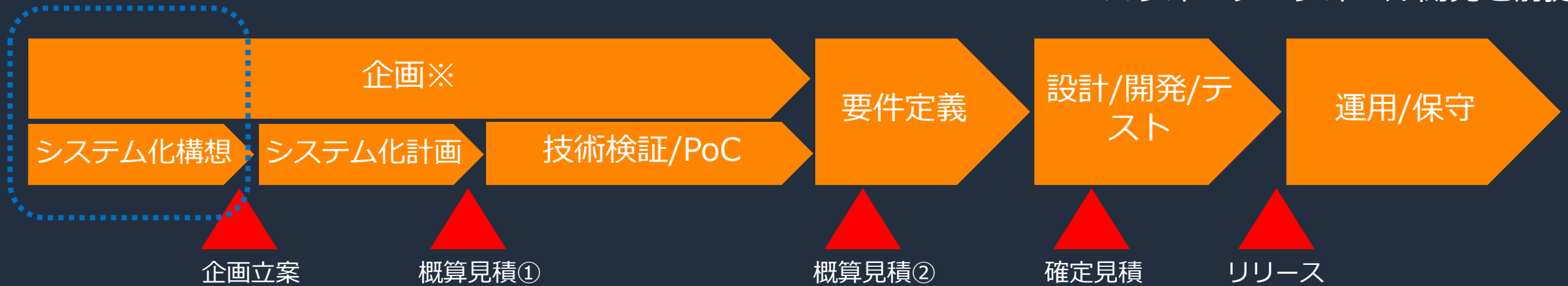
新しいスキルの習得コストやToBe にあった組織/チームの変革を考慮する

4.変化の激しい市場のなかで、数年単位の更改プロジェクトではニーズへの柔軟な追従が難しい

対象システムを選定し、仮説検証型で計画を行う。具体的には、KPI を利用して効果を計測し、投資対効果をみながら継続モダナイゼーション、さらに他への横展開を行う

一般的な開発プロジェクトの流れと前回のスコープ

※ウォーターフォール開発を前提



【企画】

IPA の「共通フレームワーク2013」では、要件定義の前にシステム化構想/システム化計画の企画プロセスを定義。

- システム化構想：経営課題を解決するための業務課題とシステムの構成を立案
- システム化計画：システム化構想を具現化するためのシステム化計画およびプロジェクト計画を具現化し、利害関係者の合意を得る。

モダナイゼーションプロジェクト立ち上げ前の検討ポイント

企画：システム化構想



1. ビジネスニーズと
価値の定義



2. KPI の策定



3. 現状把握
(棚卸し)



4. モダナイゼーション対象システム
の優先順位付けと移行方針策定



5. 人・組織の
変革プラン



6. 経済性評価



7. 対象システムの選定



8. ロードマップ策定

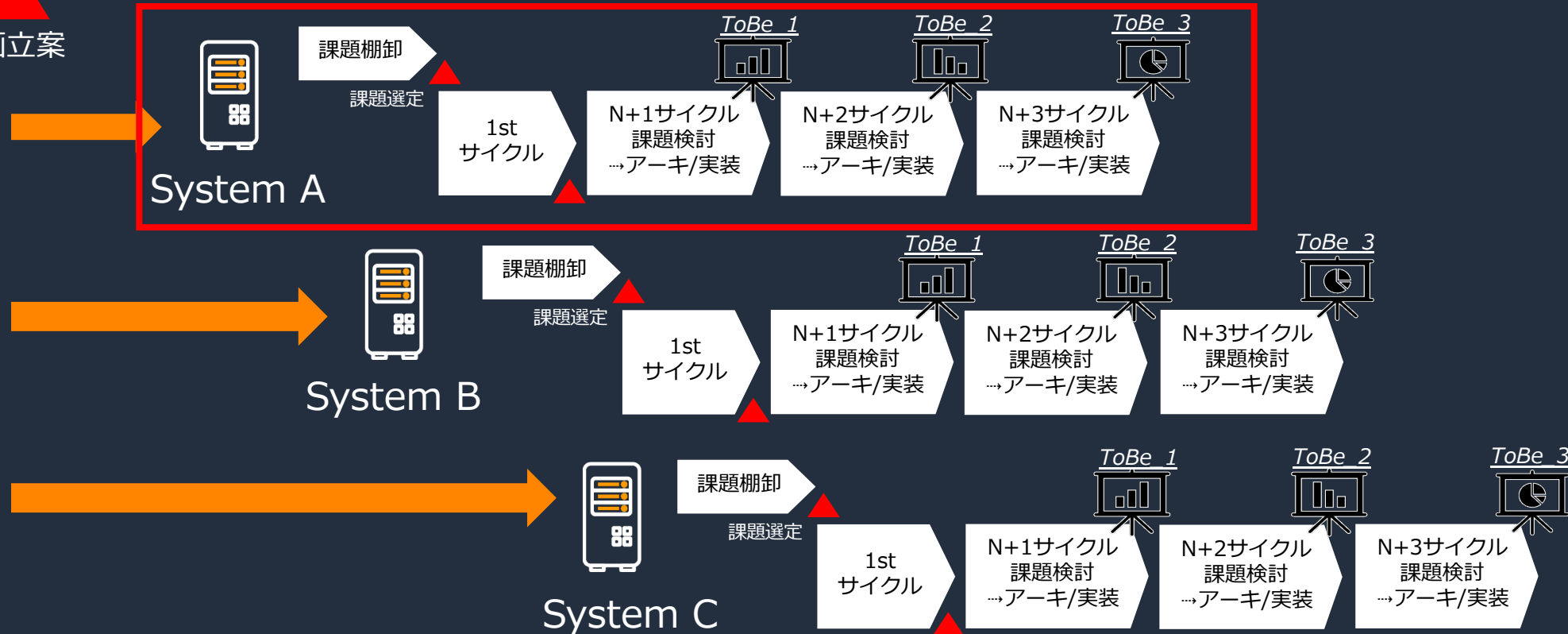
モダナイゼーション プロジェクトの立ち上げ

モダナイゼーションプロジェクトの立ち上げ

”選定”したシステムを対象に
モダナイゼーションプロジェクトを立ち上げ

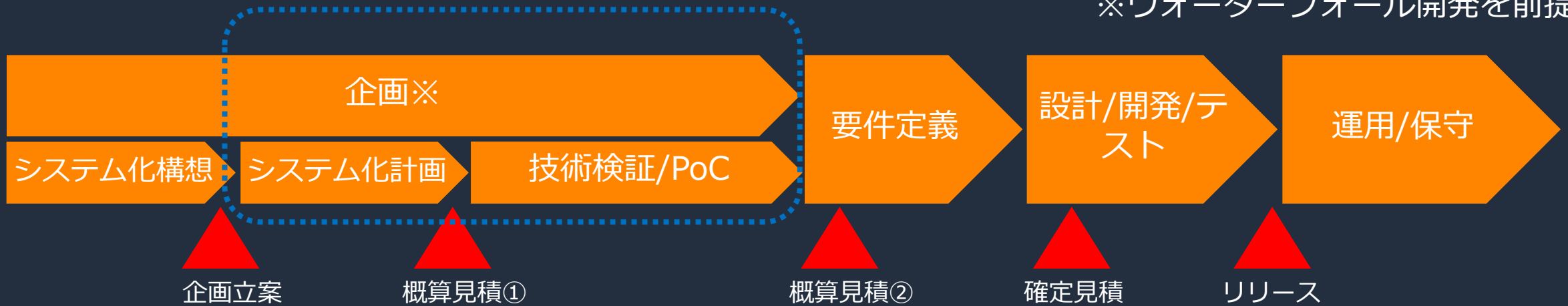
企画：システム化構想

企画立案



一般的な開発プロジェクトの流れと今回のスコープ

※ウォーターフォール開発を前提



【企画】

IPA の「共通フレームワーク2013」では、要件定義の前にシステム化構想/システム化計画の企画プロセスを定義。

- システム化構想：経営課題を解決するための業務課題とシステムの構成を立案
- システム化計画：システム化構想を具現化するためのシステム化計画およびプロジェクト計画を具現化し、利害関係者の合意を得る。
- 技術検証/PoC：利害関係者の合意を得るにあたり、その懸念を払拭するために実施。

モダナイゼーションプロジェクト立ち上げのポイント

“People”、“Process”、“Technology”の3つの軸に沿って
1.AsIs の分析、2.ToBe を描く、3.PoC の実施の3ステップにおけるポイントの解説

企画：システム化構想

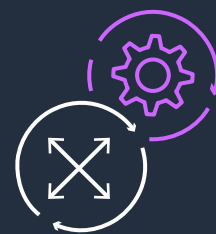


システム化構想企画書

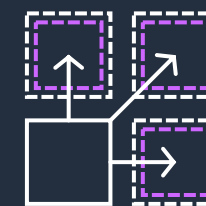
企画：システム化計画/技術検証/PoC



People



Process



Technology



1.AsIs の分析
(現行評価)



2.ToBe を描く



3.PoC の実施

People



People のポイント

- チームや組織のモダナイズには時間がかかることを念頭におく、最初から理想を目指しすぎない。
- 自社にはできないと諦めるのではなく、世の中で成功している取り組みなどを参考に、意識して実践することで変化を起こす。

【People】 1. AsIs の分析



AsIsの分析
(現行評価)

【システム化構想】

全体を俯瞰した分析

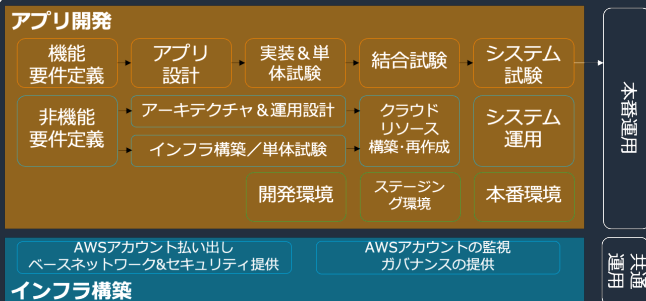
自社人材が保持する
既存スキルの把握



アプリケーション
エンジニア



インフラ
エンジニア

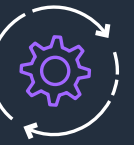


アプリとインフラチーム
の役割

プロジェクト対象のシステムを 担当するチームの詳細分析

● 分析対象

- ✓ スキル
 - ✓ 現状チームが保持しているスキル
 - ✓ 本人が今後が望むエンジニア像
- ✓ チームの種類とその役割
 - ✓ プロジェクト横断で活動するチームとの役割分担
 - ✓ アプリとインフラの責任分界点
- ✓ 担当チームがそれぞれ抱える課題の整理
- ✓ 社内メンバーの評価制度とその満足度
- ✓ 要件定義から運用までの開発要員の種類（社外/社内/IT子会社など）とその比率

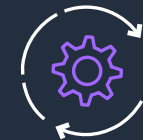


【People】 2. ToBe を描く ～ToBe を考える上でのポイント

- People-1
 - 自社人材（人財）の育成をする
- People-2
 - 役割分担のモダナイズを実施する
- People-3
 - 実現可能な内製化プランを立てる
- People-4
 - 組織やチームの変革に取り組む

【People】 2. ToBe を描く ～自社人材（人財）の育成

People-1

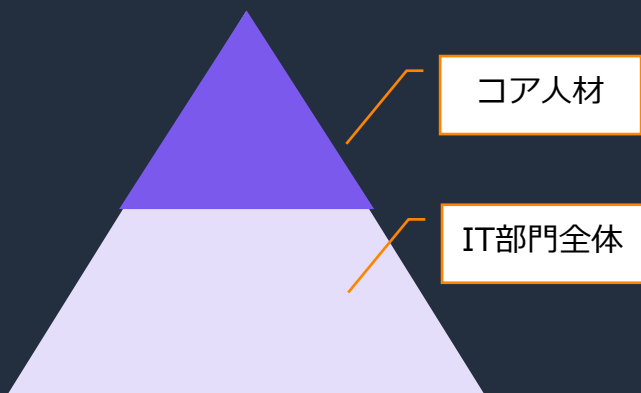


ToBeを描く

外部リソースの有効活用
世の中で実績のあるツールの採用も有効

育成の方向性

1. IT 部門全体の基礎スキルの底上げ
2. コア人材の育成

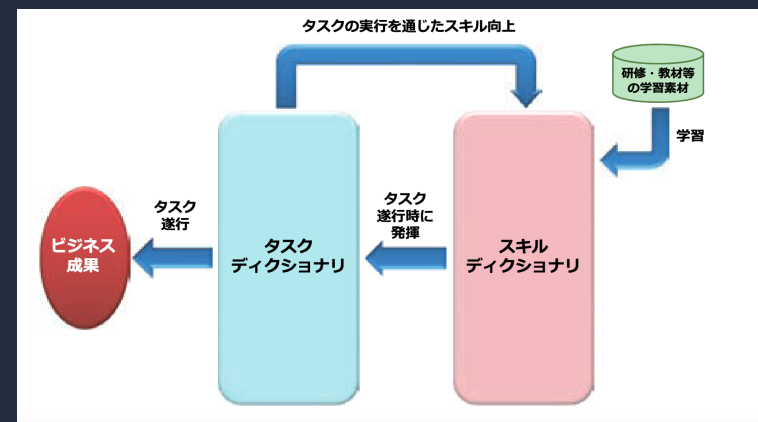


育成プランと実施

- 現状保持しているスキルと本人の意向を加味した育成プランの策定
- 育成プログラムの開発には、外部講師の活用も有効
 - AWS でもトレーニングサービスを多数提供 <https://aws.amazon.com/jp/training/>
- 伝道師を作る
- 社内/社外コミュニティの活用
 - AWS でも、JAWS-UG (AWS Users Group – Japan) の活動が活発 <https://jaws-ug.jp/>

モチベーションを維持するために

- スキルの見える化（デジタルバッジやジョブ・カード）
- エンジニアの評価制度の見直し



Source : IPA
コンピテンシディクショナリ解説書

【People】 2. ToBe を描く ～役割分担のモダナイズ 1/2



“ビジネスの俊敏性”が求められる領域には、
開発者のさらなる生産性向上を目指す

例) プラットフォームエンジニアリング

AsIs

マイクロサービスにコンテナ、IaC とか、
アプリ開発者としては、
考えることが多く、もう
パンク状態・・・



アプリ開発チーム

ToBe

インフラの自動化と
セルフサービス機能を
再利用可能な標準
として提供するよ



プラットフォームチーム

デリバリパイプライン
や低レベルなインフラ
管理の業務から解放さ
れた。よりアプリ開発
に集中できるよ



アプリ開発チーム

【People】 2. ToBe を描く ～役割分担のモダナイズ 2/0

People-2



ToBeを描く

顧客に価値を提供しつづける、品質を維持するため、
運用フェーズにも注目する

例) SRE (Site Reliability Engineering)

AsIs

進化をとめないために、
運用フェーズでも継続的
な改善が必要だけど、次
の開発もあるし、今の体
制だけでは心配だ。



??

アプリ開発チーム



ToBe

コーディングによる手作業や
繰り返し行われる作業（ Toil ）
を減らしたり、システム
を自動化して作業量の増大に
対応しよう



SRE チーム

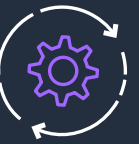
システム運用からア
プリ開発まで、信頼
性に関わることなら
任せて

リリースした後も
アプリの価値や品
質を維持できるね。



アプリ開発チーム

【People】 2. ToBe を描く ～内製化プラン 1/2

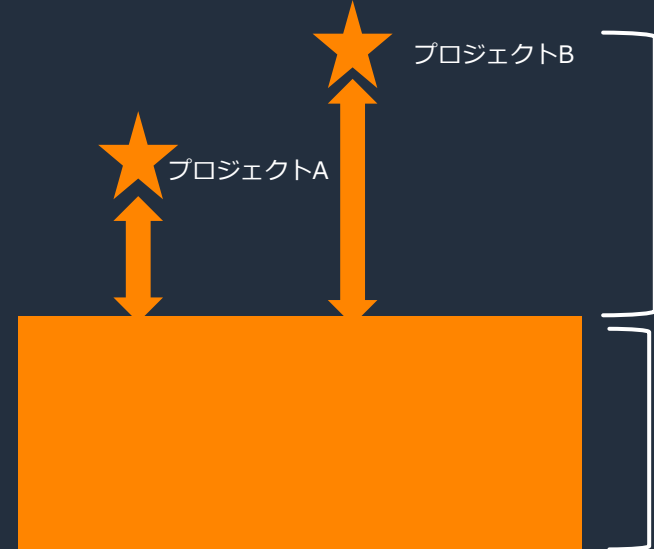
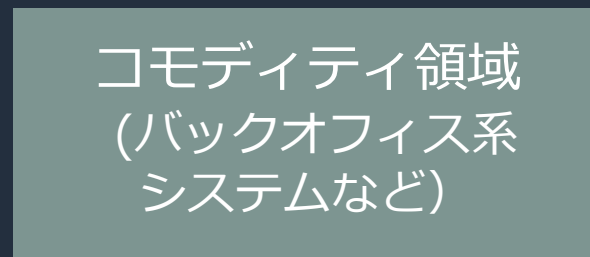
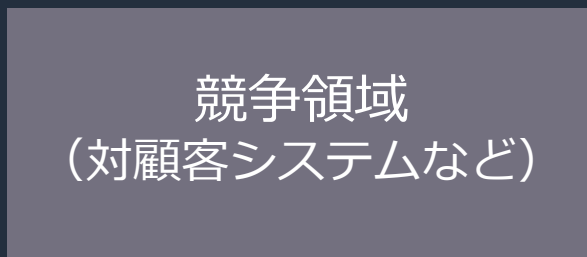


内製化の目的は、迅速な対応が必要なコア・コンピタンスに集中すること。そのための対象や社員数の適正を見極める

企業の経営戦略

内製化対象をどうするか

社員の適正数はどれくらいか



外部
パートナー活用

内製化要員

内製化の対象

外部パートナーも活用

- 内製化要員は固定費
- 日本は法律上、急な解雇が困難

【People】 2. ToBe を描く ～内製化プラン 2/2



各工程の内製化比率をブレイクダウン
現状のレベルからみて、現実的な目標を設定する

工程	内製レベル			
	L0 AsIs	L1 ToBe	L2	L3
要求定義/受入テスト	自社	自社	自社	自社
要件定義/システムテスト/ UI テスト	外部パートナー	自社	自社	自社
外部設計/ インテグレーションテスト	外部パートナー	外部パートナー	自社	自社
内部設計/製造/レビュー/ ユニットテスト	外部パートナー	外部パートナー	外部パートナー	自社

アジリティ

低

高

体制構築

易

難

外部依存リスク

高

低

【People】 2. ToBe を描く ～組織やチームの変革

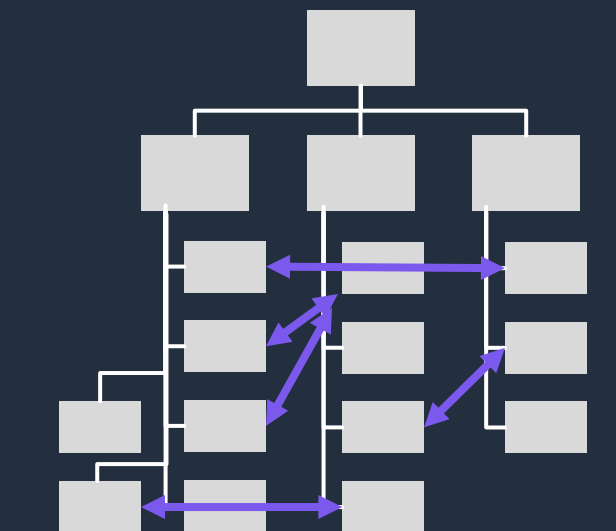
People-4



ToBeを描く

意識をし、まず取り組むことから始める
そこから、マインドと文化が醸成される

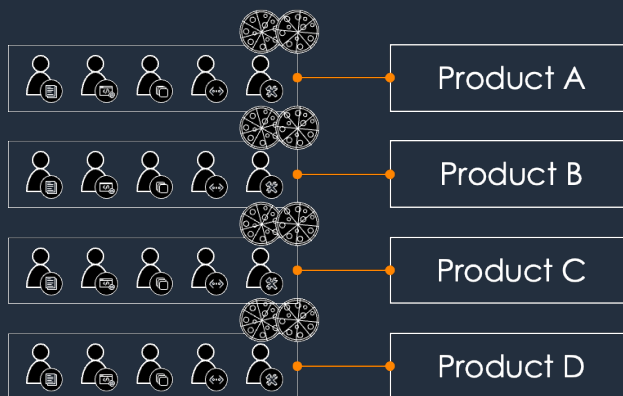
組織図からの脱却



創造性と問題解決のために
推奨すべき実業務における
コミュニケーション

逆コンウェイ戦略

アーキテクチャのための
組織をあらかじめ作る



チームファースト思考



信頼関係を保つための
小さいチームを作る



何を作るかから実行まで
すべての権限を持つ
(オーナー)



個人ではなくチーム
に報酬を与える



チームの多様性を
受け入れる



【People】 3. PoC の実施



新トレーニングの評価やスキル
習得度の検証



見積もり精度の向上



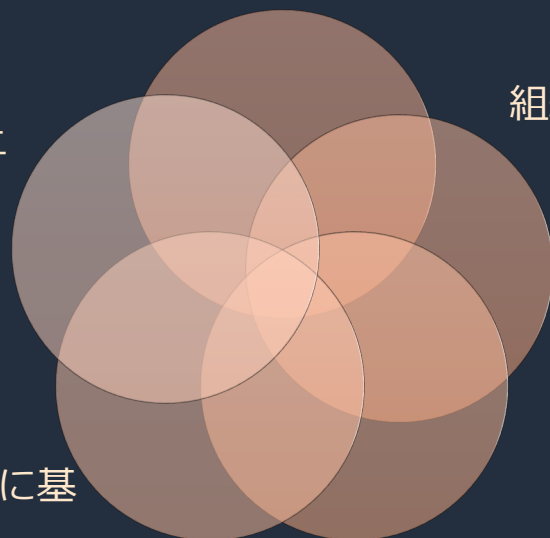
効果の予測や数値に基
づいた効果の検証



組織/チーム構成変更の実
現可能性検証



評価制度やスキル確認
ツールなど仕組みの検証



PoC における目的とそこで得られる 成果や基準を明確にする

- “People”の観点のPoC では、
- ・ 期間を限定したり、特定の組織などスコープを限定して試してみる。
 - ・ 成果や基準に達したかどうかの検証は、自社で判断が難しい場合、人材アセスメントの第3者評価などを活用することも有効である。
 - ・ 一方で、即時性の効果がでにくい活動であることを考慮し、実現可能性の高い目標レベルの設定に注力する。

【PeopleにおけるPoC の主な目的】

Process



Process のポイント

- 無駄を省き、品質を維持向上させるために、自動化する
- 人間の承認処理やチーム間のコミュニケーションなど、システムに依存しない無駄もある。
そのため、ときにPeople の施策とあわせて取り組む必要がある。
- 効果の確認しやすいところから徐々に取り組む

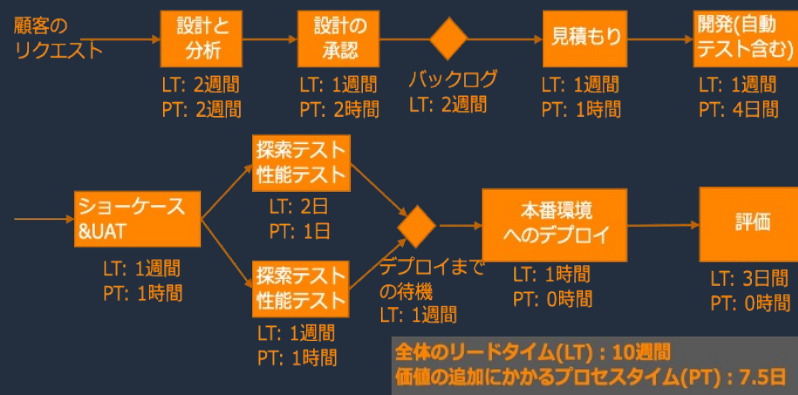
【Process】 1. AsIs の分析



AsIsの分析
(現行評価)

【システム化構想】
全体を俯瞰した分析

・ 開発プロセス全体の可視化とボトルネックの発見

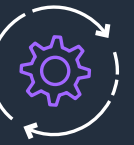


Value Stream Mapping (VSM) の例 (The DevOps Handbook日本語版より抜粋)

プロジェクト対象のシステムやチームを対象にした開発・運用プロセスの詳細分析

● 分析対象

- ✓ 下記に関するボトルネックの原因の深掘り
- ✓ 現状利用している開発プロセス
- ✓ 開発/運用周りで利用している仕組み
- ✓ 人間による承認処理や使用しているドキュメントの種類や内容
- ✓ 実施しているテストの種類と実施方法、実施にかかる工数
- ✓ テスト自動化の範囲と内容、実施方法およびその課題



【Process】 2. ToBe を描く ～ToBe を考える上でのポイント

- Process-1
 - 開発プロセスを効率化する
- Process-2
 - 開発や運用の周辺の仕組みを効率化する
- Process-3
 - テストを効率化する

【Process】 2. ToBe を描く ～開発プロセス



各プロセスの本質を理解して取り入れる
成果がでるまでに時間がかかることを理解する

例) アジャイル開発

アジャイルソフトウェア開発宣言

私たちは、ソフトウェア開発の実践
あるいは実践を手助けをする活動を通じて、
よりよい開発方法を見つけたそうとしている。
この活動を通して、私たちは以下の価値に至った。

**プロセスやツールよりも個人と対話を、
包括的なドキュメントよりも動くソフトウェアを、
契約交渉よりも顧客との協調を、
計画に従うことよりも変化への対応を、**

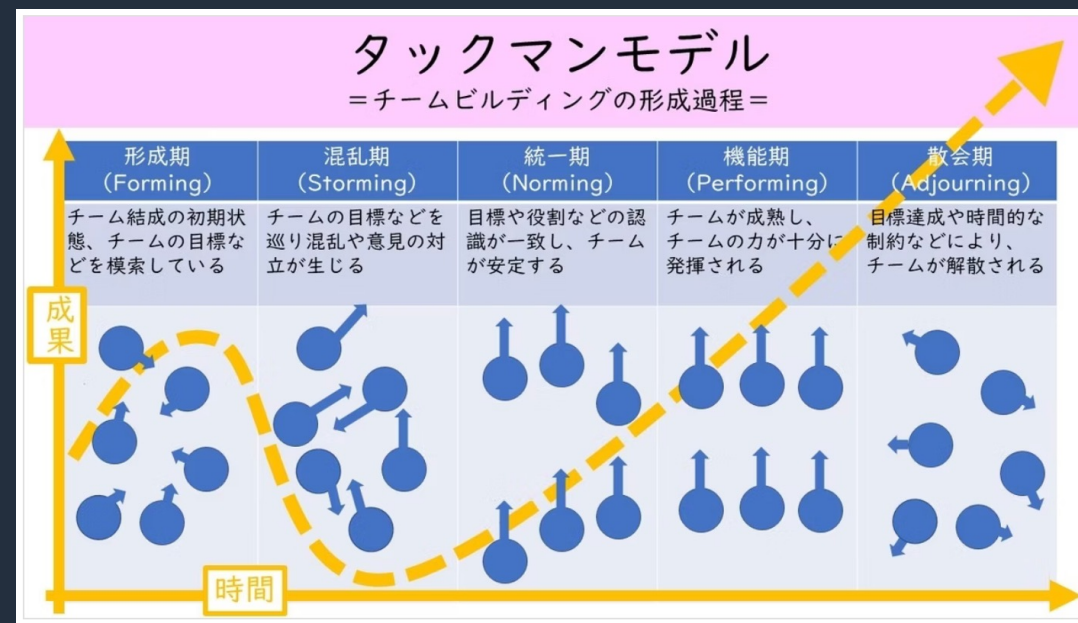
価値とする。すなわち、**左記のこと**がらに価値があることを
認めながらも、私たちは**右記のこと**がらにより価値をおく。

Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Jeff Sutherland
Martin Fowler	Brian Marick	Dave Thomas

© 2001, 上記の著者たち
この宣言は、この注意書きも含めた形で全文を含めることを条件に自由にコピーしてよい。

Source : IPA
アジャイルソフトウェア開発宣言の読みとき方

タックマンモデルにおける混乱期



Source : 「チームビルディング」
タックマンモデル

【Process】 2. ToBe を描く ～開発/運用周りの仕組み

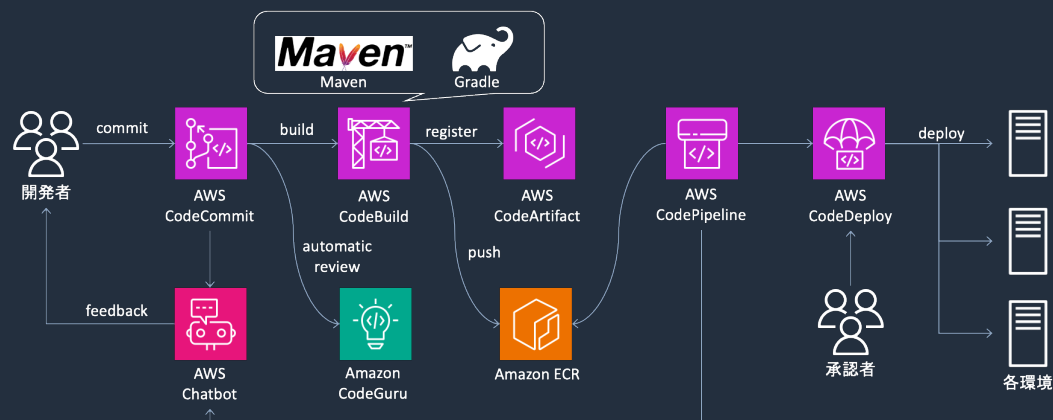
Process-2



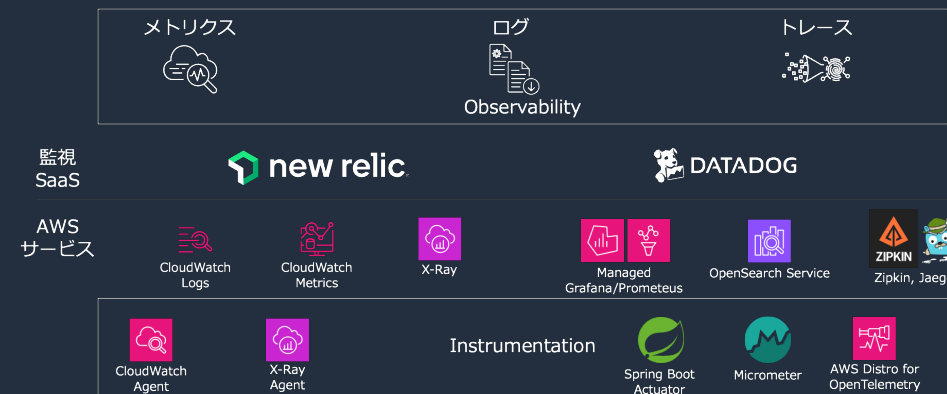
ToBeを描く

開発や維持運用の仕組みのアップデートは 品質向上や効率化の肝

- ・ 作業状況の可視化
- ・ 開発上の定形作業を自動化
- ・ 運用の効率化（可観測性の確保）

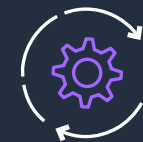


開発上の定形作業の自動化例

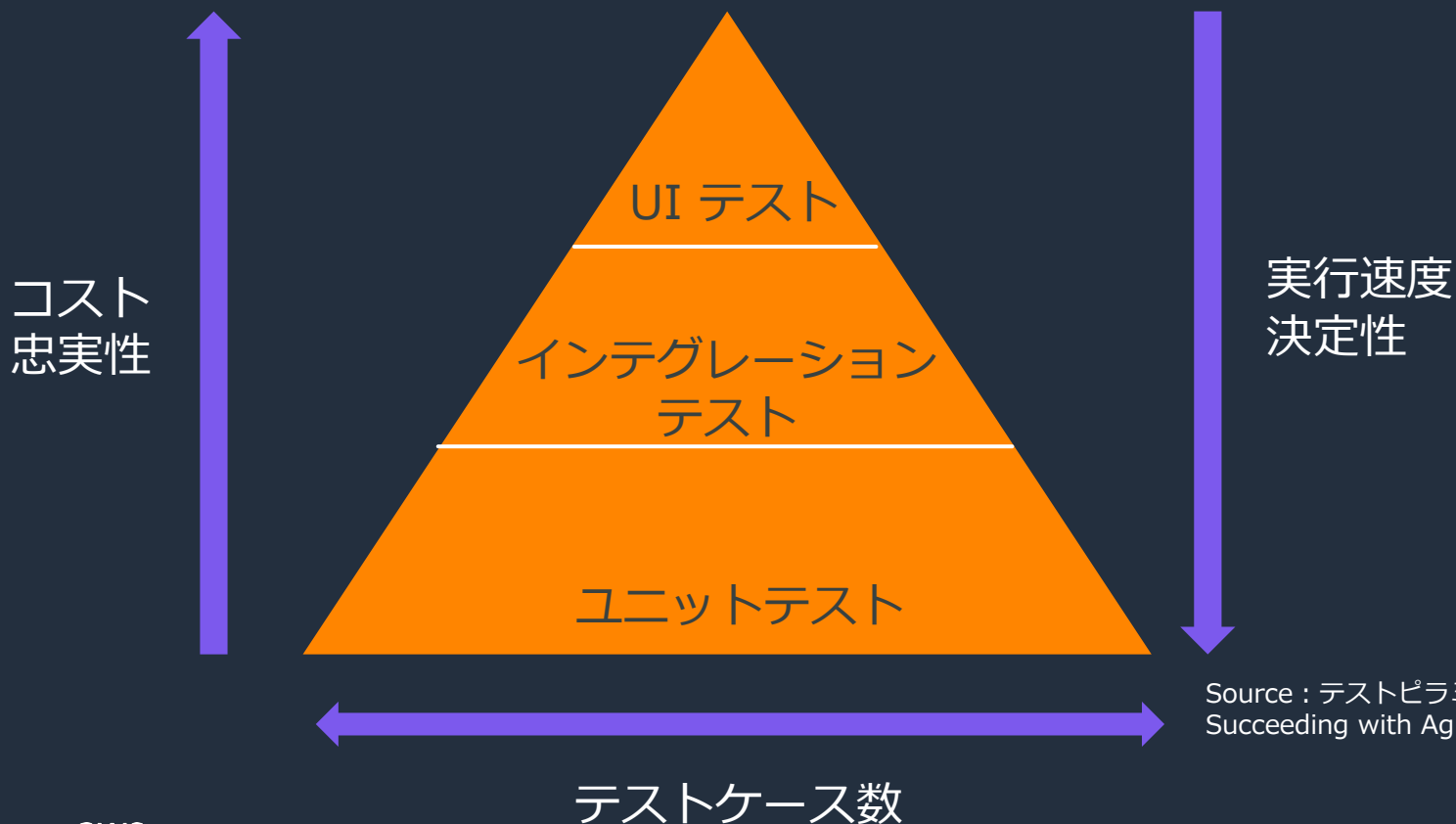


可観測性確保のための実装例

【Process】 2. ToBe を描く ～テスト（自動テストの信頼性 1/2）



自動テストの信頼を保つため、
最適なテストピラミッドを意識してテスト設計をする



- 検証の対象となるコード
範囲や粒度での分類



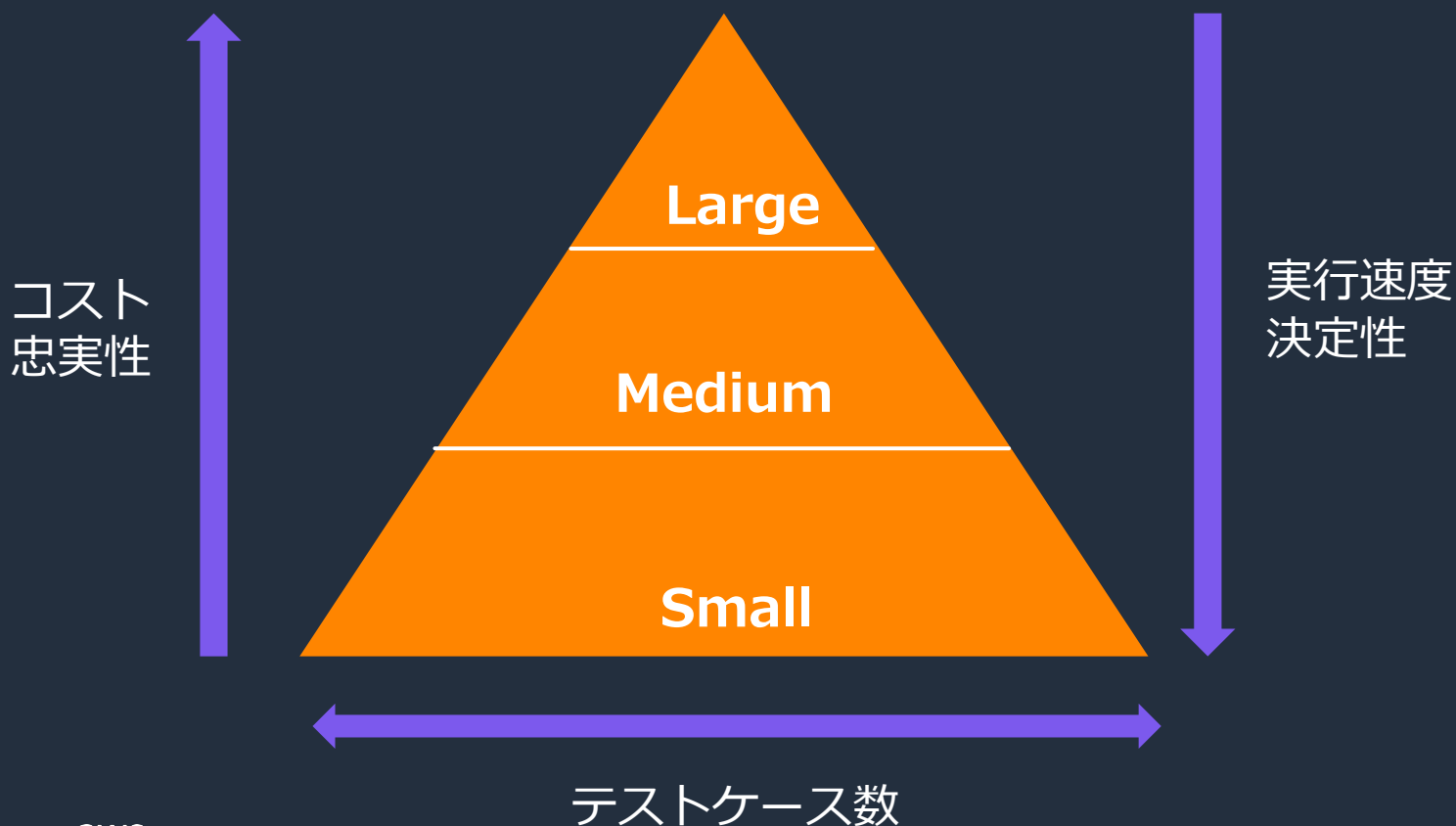
UI テスト、インテグレーションテスト、ユニットテストなど各テストレベルで、どのようなテストを実施するか認識、解釈が人やチーム、組織によって異なる

Source : テストピラミッド
Succeeding with Agile: Software Development Using Scrum(Mike Cohn,2009)

【Process】 2. ToBe を描く ～テスト（自動テストの信頼性 2/2）



自動テストの信頼を保つため、
最適なテストピラミッドを意識してテスト設計をする



● テストサイズによる分類



Google 社から報告された、自動テストに使用されるリソースの量や実行場所・時間に注目した分類方法。

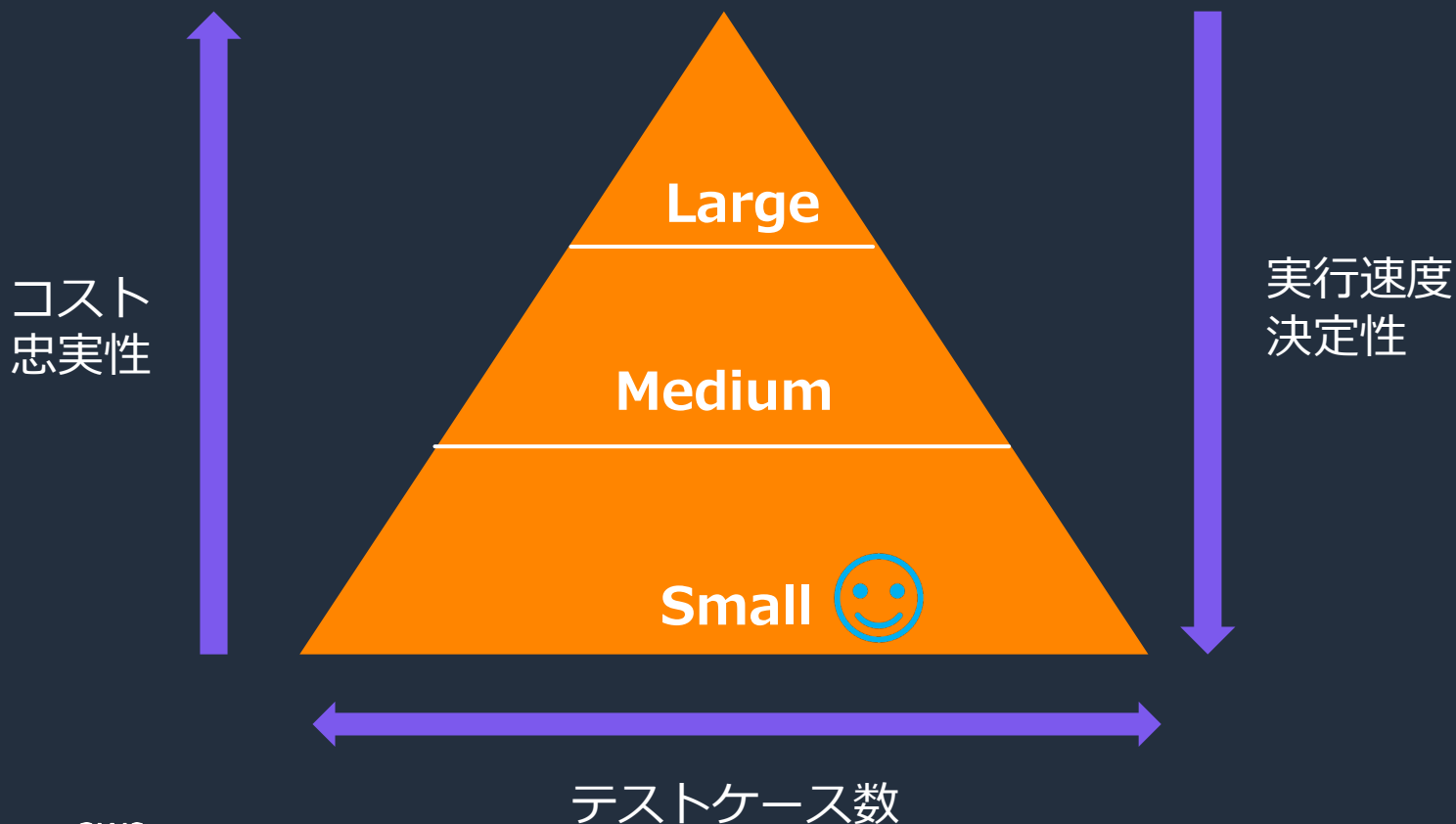
Small : 単一のプロセス内で実施するテスト
Medium : 単一のマシンに閉じた環境内のテスト

Large : 自動テストを実施するマシンから他のマシンへの接続を許容するテスト

【Process】 2. ToBe を描く ～テスト（自動テストの範囲）



効果の高い、難易度の低いところから取り組む



● テスト自動化の範囲

最終的に、Small、Medium、Large すべてのテストを自動化できることが望ましいが、すべてを最初から実装できている必要はない。まずは、かかる工数の観点で効果の高い、またテストサイズの観点から実装の難易度が低い“Small”から整備していくとよい。



【Process】 3. PoC の実施



開発プロセスの実践



見積もり精度の
向上



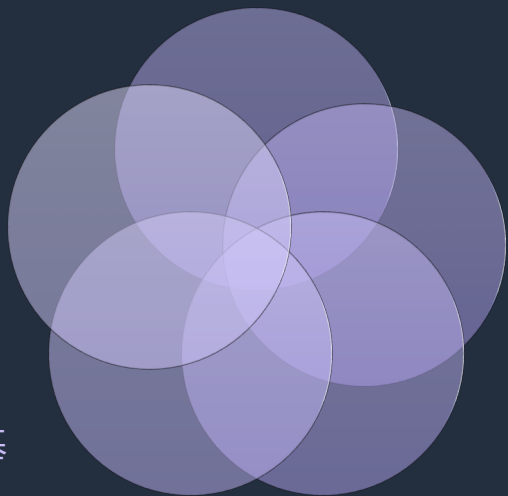
効果の予測や数値に基
づいた効果の検証



テスト自動化方
法と効果の検証



各種ツールの評価
とノウハウの蓄積



PoC における目的とそこで得られる
成果や基準を明確にする

- “Process”の観点では、
- ・ 開発プロセスやテスト全体の中での効率化であり、無駄が多く効果の高いところから取り組むとよい。
 - ・ Process に関する変更は、People の要素とあわせて初めて効果がでる場合もあるため、安易に効果がないと判断しない
 - ・ 上記を踏まえ、継続的な取り組みであることを念頭におく。

【ProcessにおけるPoC の主な目的】

Technology のポイント

Technology のポイント

- 手段の検討の前に、改めて目的やゴールの共通認識をはかる
- ゴールやコスト、スケジュール、他システムの都合などの制約を踏まえてToBe アーキテクチャを決定する。
そして、これからも段階的にアーキテクチャを刷新することを視野にいれる。

【Technology】 1. AsIs の分析



AsIsの分析
(現行評価)

【システム化構想】
全体を俯瞰した分析

特定のシステムやアプリケーション
を対象にした詳細分析

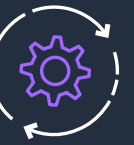
● 分析対象

- ✓ 現行資産
 - OS /ミドルウェア
 - ソースコード
 - フレームワーク
 - 各種定義体
- ✓ 業務運用/システム利用状況
- ✓ 業務知識の保持レベル
- ✓ アプリケーション保守の状況
- ✓ 外部I/F や他システムとの連携状況
- ✓ 非機能要件
 - セキュリティ・拡張性・信頼性・性能など
- ✓ アプリケーション開発上の課題

● 調査方法

- ✓ ヒアリング
- ✓ 実機調査
- ✓ ドキュメント調査
- ✓ ソースコード解析
 - 手動解析
 - ツール活用





【Technology】 2. ToBe を描く ～ToBe を考える上でのポイント

- Technology-1
 - 目標を意識して**ToBe Architecture** を描く
- Technology-2
 - 決定事項について**Architecture Decision** を残す
- Technology-3
 - **他システム連携**の方針決めと調整

【Technology】 2. ToBe を描く ～To-Be Architecture

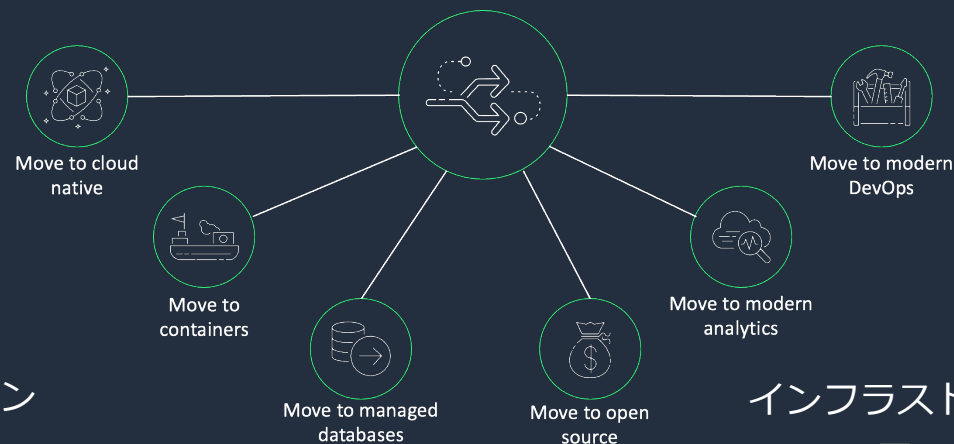
Technology-1



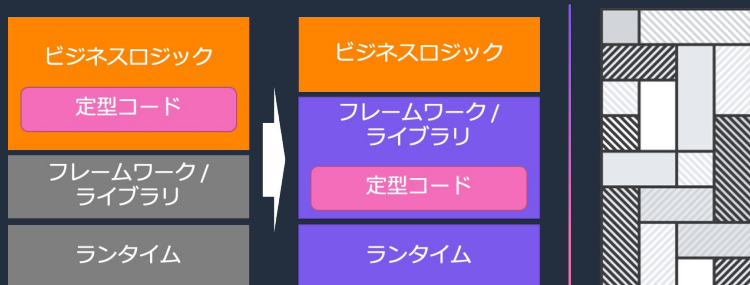
ToBeを描く

最新技術の採用が正解ではない
求められるビジネス価値と解決したいIT 課題のバランスをとる

モダナイゼーション・パスウェイ



アプリケーション・モダナイゼーション



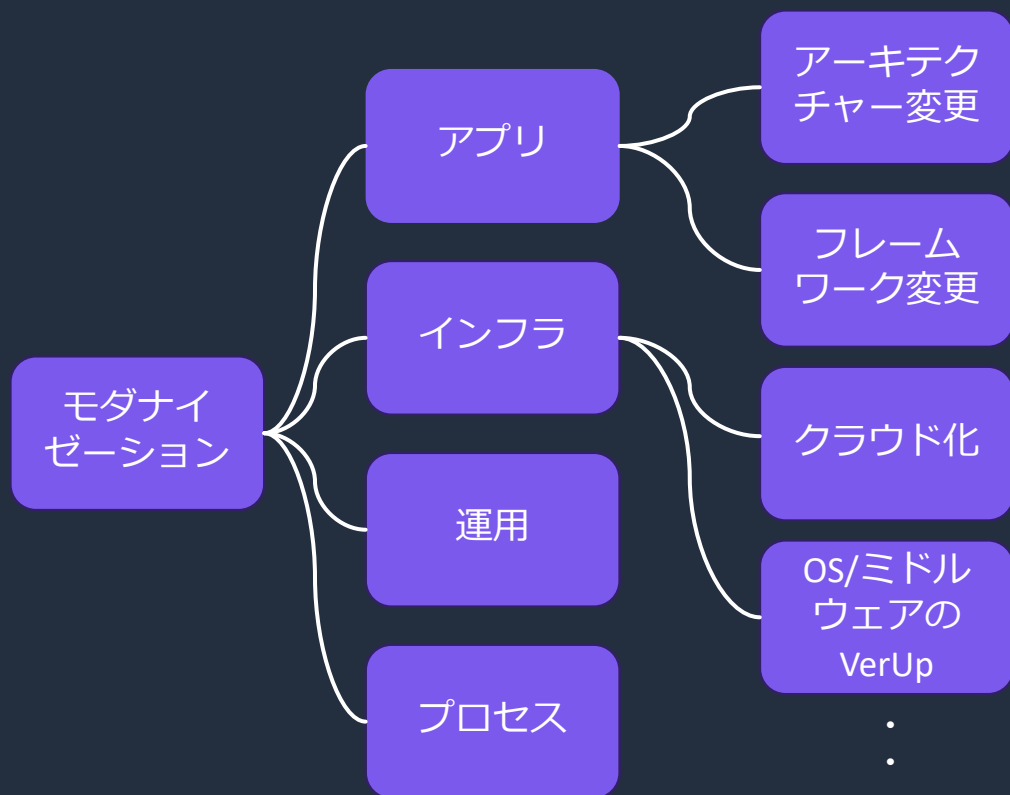
インフラストラクチャ・モダナイゼーション



【Technology】 2.ToBe を描く ～Architectural Decision



この時点での最善をディシジョンする
“検討経緯”の記録を残す



- ・ 段階的にアーキテクチャを刷新する場合、過去の検討経緯や案を振り返ることが必要（二度手間を防ぐという意味もある）
- ・ 決定事項のみならず、意思決定に至った議論や背景、制約条件などを記載する

【Technology】 2.ToBeを描く ～他システム連携



連携処理を分解し、可能な限りモダナイズを
他システムとは早めに調整を

● ToBe の連携方針

連携先に依存する処理

- ・ 全体方針を確認
- ・ 基本的には、密結合から疎結合な連携へ変更。その際、トランザクション/性能要件を考慮。
- ・ 制約などにより変更が難しい場合には、現行IF を維持

連携先に依存しない処理

連携先とのIF は維持する場合でも、連携前後の内部処理は、クラウドサービスの活用などモダナイズが可能なケースがある

密結合：ロジックの実装の中身、データの型やスキーマ構造に依存して使われる)
疎結合：機能の利用方法を厳密に定義して、利用してもらう



【Technology】 3.PoC の実施



PoC における目的とそこで得られる成果や基準を明確にする

“Technology”の観点では、

- ・ クラウドのインフラリソースを有効活用
- ・ やみくもに“細かい機能”の検証を実施しない
- ・ 他システムとの連携方法を変える場合、他システムとPoC 実施方法や役割分担についても調整が必要。

などといったポイントを考慮して取り組む。

【TechnologyにおけるPoC の主な目的】

まとめ



モダナイゼーションプロジェクト立ち上げのポイント

“People”、“Process”、“Technology”の3つの軸に沿って
1.AsIs の分析、2.ToBe を描く、3.PoC の実施の3ステップにおけるポイントの解説

企画：システム化構想

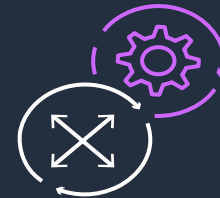
企画：システム化計画/技術検証/PoC



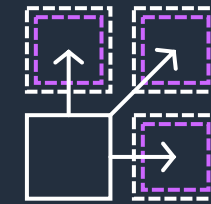
システム化構想企画書



People



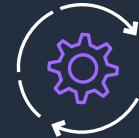
Process



Technology



1.AsIs の分析
(現行評価)



2.ToBe を描く



3.PoC の実施

本セッションのまとめ

- People

- モダナイゼーションの成功を左右する重要なポイントが多いため、長期的視点かつ徐々にできることから取り組んでいく
- まずは意識して取り組むことで、マインドやカルチャーが醸成される

- Process

- 現行の無駄を排除し、自動化や品質向上に取り組む

- Technology

- ゴールを意識してToBe Architecture を描き、継続モダナイゼーションを実施するため、決定事項はArchitecture Decision を残す
- 他システム連携は複数の処理から構成されるため、分解してモダナイズできる部分を考える。
連携先に依存する処理は、早めに調整を開始する

参考資料

◆ AWS Black Belt

- 「モダナイゼーションとは」 ～Modernization 入門シリーズ
(<https://www.youtube.com/watch?v=-Nu-KTsFk1A>)
- 「モダナイゼーション・パスウェイ」 ～Modernization 入門シリーズ
(<https://www.youtube.com/watch?v=wyp8EOkLUpk>)
- 「モダナイゼーションプロジェクト立ち上げに向けたポイント」 ～Modernization 入門シリーズ
(https://www.youtube.com/watch?v=mh3SpS_E24s)

◆ AWS Prescriptive Guidance

- Phases of the modernization process
(<https://docs.aws.amazon.com/prescriptive-guidance/latest/strategy-modernizing-applications/phases.html>)

AWS Black Belt Online Seminar とは

- 「サービス別」「ソリューション別」「業種別」などのテーマに分け、アマゾン ウェブ サービス ジャパン合同会社が提供するオンラインセミナーシリーズです
- AWS の技術担当者が、AWS の各サービスやソリューションについてテーマごとに動画を公開します
- 以下の URL より、過去のセミナー含めた資料などをダウンロードすることができます
 - <https://aws.amazon.com/jp/aws-jp-introduction/aws-jp-webinar-service-cut/>
 - <https://www.youtube.com/playlist?list=PLzWGOASvSx6FIwIC2X1nObr1KcMCBBlqY>



ご感想は X (Twitter) へ！ハッシュタグは以下をご利用ください
#awsblackbelt

内容についての注意点

- 本資料では資料作成時点のサービス内容および価格についてご説明しています。AWS のサービスは常にアップデートを続けているため、最新の情報は AWS 公式ウェブサイト (<https://aws.amazon.com/>) にてご確認ください
- 資料作成には十分注意しておりますが、資料内の価格と AWS 公式ウェブサイト記載の価格に相違があった場合、AWS 公式ウェブサイトの価格を優先とさせていただきます
- 価格は税抜表記となっております。日本居住者のお客様には別途消費税をご請求させていただきます
- 技術的な内容に関しましては、有料の [AWS サポート窓口](#)へお問い合わせください
- 料金面でのお問い合わせに関しましては、[カスタマーサポート窓口](#)へお問い合わせください (マネジメントコンソールへのログインが必要です)



Thank you!