

SAFe導入初期にアーキテクトは何を考えるか

SAFe® DAY JAPAN 第6回スポンサー共済ウェビナー

2021/9/10

本日のアジェンダ

1. 講演者自己紹介
2. SAFeアーキテクトの責任
3. SAFeアーキテクトの行動原理
4. SAFeの技術要素と導入時のポイント
5. まとめ

講演者自己紹介

梶原 直人

NTTデータ先端技術株式会社

ソフトウェアソリューション事業本部 APテクノロジー事業部

- 2006年 NTTデータ先端技術に入社。プライベートクラウド構築プロジェクトや、多数のアジャイル開発プロジェクトに従事。
- 2015年から当社でアジャイル専門組織を立ち上げ、近年は大規模アジャイル開発に関する導入コンサルティングやソリューション開発を専門に活動中。
- 2016年には製造業様のコネクテッドサービス開発に対しソリューションアーキテクトとしてSAFe導入に取り組んだ。
- Certified SAFe® Program Consultant、Certified ProductOwner、Project Management Professionalなどの資格を持つ。
- 技術評論社 Software Design 2020年12月号から、記事連載中「チーム開発の視点が変わる アジャイル開発の新常識」



講演者自己紹介

中谷 浩晃

Scaled Agile Japan

ストラテジックアドバイザー

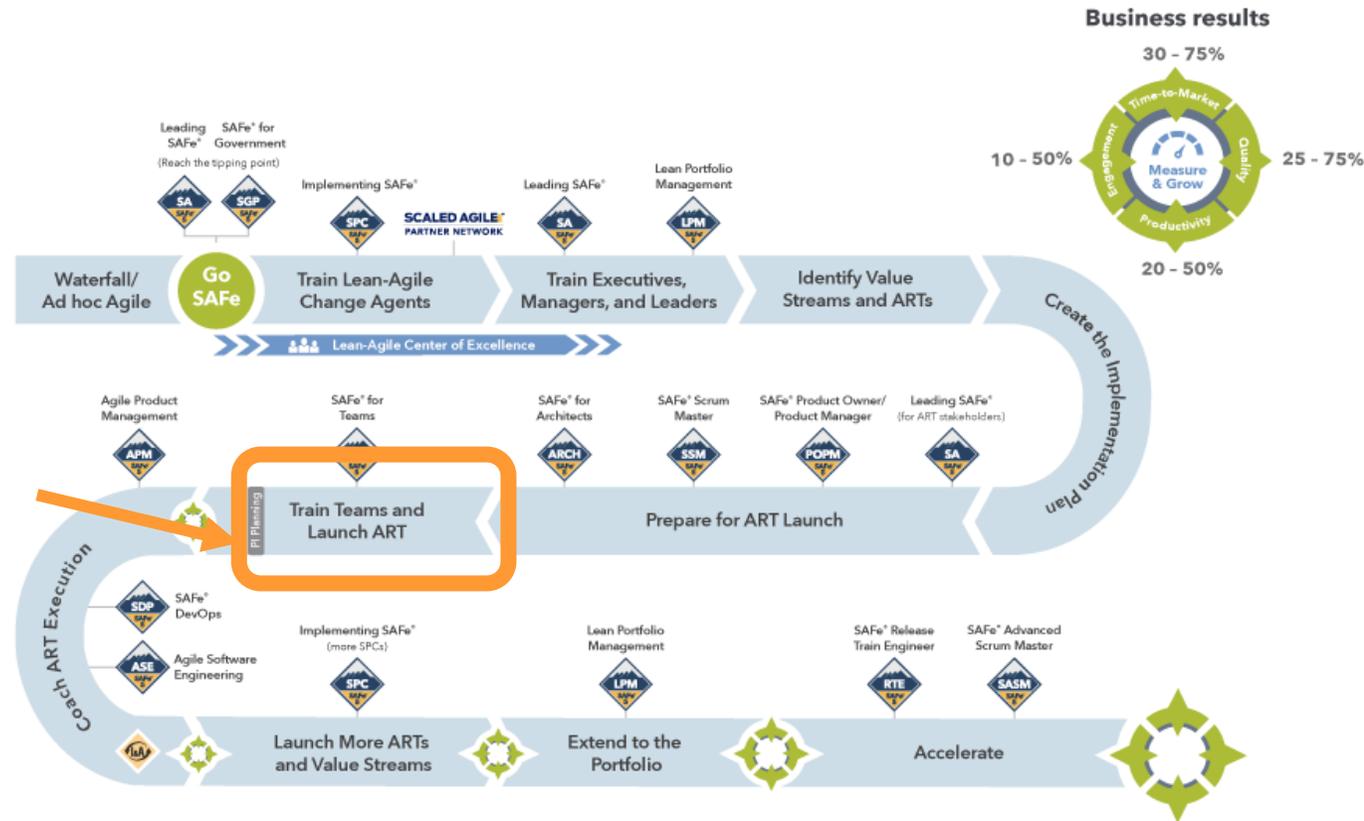
- 前職ウエスタンデジタルにて、自社のグローバルレベルのトランスフォーメーション推進のため、チェンジエージェントとして日本、米国、アジアの国々と連携し変革を推進。人材育成のためのSAFeトレーニング開催から、社内ARTメンバーやマネジメントに対するコーチングやコンサルティングに至るまで、SAFe実装の一連のプロセスを実践し推進。
- SAFe MeetupやScaled Agile主催ウェビナーでの講演によるSAFeコミュニティへの参画、およびScaled Agile In-country ReviewerとしてSAFeコースウェアのローカライズに貢献。日本人初のiSPCTとなる。
- 2021年9月よりScaled Agileに所属。東京勤務



SAFe導入の流れ

アーキテクトは、一般的にいう「本格的な開発作業」が開始されるAgile Release Trainのローンチまでに**必要な技術要素を準備する**

ARTローンチ



従来の大規模システムのアーキテクトが抱える悩み



本日のテーマ

Q. SAFeアーキテクトは、
SAFeの導入初期フェーズにおいて、
どのような考えで、何を準備すべきか

説明の流れ

1. SAFeアーキテクトの責任を知る

2. SAFeアーキテクトの行動原理を知る

3. SAFeの技術要素を知る

導入時のポイント

SAFeアーキテクトの 責任

SAFeアーキテクトの責任

主にARTやアジャイルチームと**コラボレーション**しながら、**リーダーシップ**を発揮し、以下に責任を持つ

1 システムをどのように動かすか？

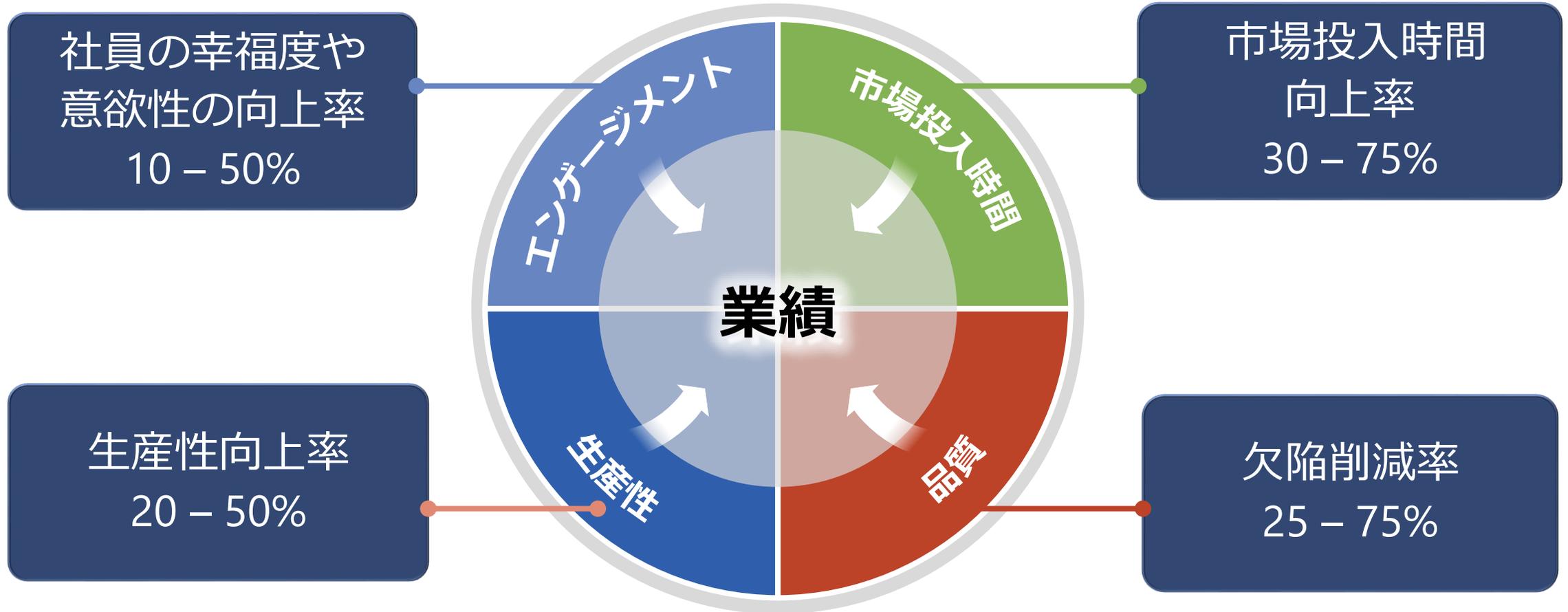
- Architectural runwayの維持とそのバックログ(Enabler)の定義
- 設計が現在と将来をサポートすることの保証
- 非機能要件(NFR)の定義

2 システムをどのように作るか？

- 継続的デリバリパイプラインの有効化、促進、改善の主導
- DevOpsの促進
- バリューストリームとARTの特定や設計、およびそのサポート

SAFeアーキテクトの 行動原理

SAFeは何のためにあるのか？



顧客へ継続的に価値を提供
ビジネスアジリティの実現

アジャイルとリーン

アジャイル宣言の背後にある原則

顧客満足を最優先し、
価値のあるソフトウェアを早く継続的に提供します。
要求の変更はたとえ開発の後期であっても歓迎します。
変化を味方につけることによって、お客様の競争力を引き上げます。
動くソフトウェアを、2-3週間から2-3ヶ月という
できるだけ短い時間間隔でリリースします。
ビジネス側の人と開発者は、プロジェクトを通して
...
...
技術的卓越性と優れた設計に対する
不断の注意が機敏さを高めます。
シンプルさ（ムダなく作れる量を最大限にすること）が本質です。
最良のアーキテクチャ・要求・設計は、
自己組織的なチームから生み出されます。
チームがもっと効率を高めることができるかを定期的に振り返り、
それに基づいて自分たちのやり方を最適に調整します。

リーン開発の原則

1. ムダを無くす
2. 知識を作り出す
3. 決定を遅らせる
4. 早く提供する
5. 人を尊重する
6. 全体を最適化する
7. 品質を作りこむ

SAFeもこれらの原理原則に従っている

アジャイルアーキテクトの行動原理

アジャイルの原理原則

① 「要求の変更はたとえ開発の後期であっても歓迎します。」



変更容易性を高めておくこと

② 「価値のあるソフトウェアを早く継続的に提供します。」



小さな価値の単位で頻繁にリリースができること

③ 「シンプルさ(ムダなく作れる量を最大限にすること)が本質です。」



シンプルなアーキテクチャを継続的に維持すること

アジャイルアーキテクトの行動原理

大規模に適用する

アジャイル開発の原則

リーン開発の原則

巨大で複雑な
システム

ミッションクリ
ティカルな
システム

大企業の多数の
非エンジニア

SAFeリーンアジャイルの原理原則

SAFeアーキテクトの行動原理

SAFeリーンアジャイルの原理原則

1 「システム思考を持つ」



SAFeアーキテクトの行動原理

システムを俯瞰して深く理解し、全体最適化を図ること

2 「変動性を想定する、オプションを持つ」



複数の設計の選択肢を用意し、学習により収束させること

3 「WIPの視覚化と制限、バッチサイズの削減、キューの長さの管理」



スループットを向上させ、価値の提供を加速すること

SAFeの技術要素と 導入時のポイント

SAFeの技術要素

システムをどのように動かすか？



Architectural Runway
アーキテクチュラル・ランウェイ

システムをどのように作るか？



Continuous Delivery Pipeline
継続的デリバリパイプライン

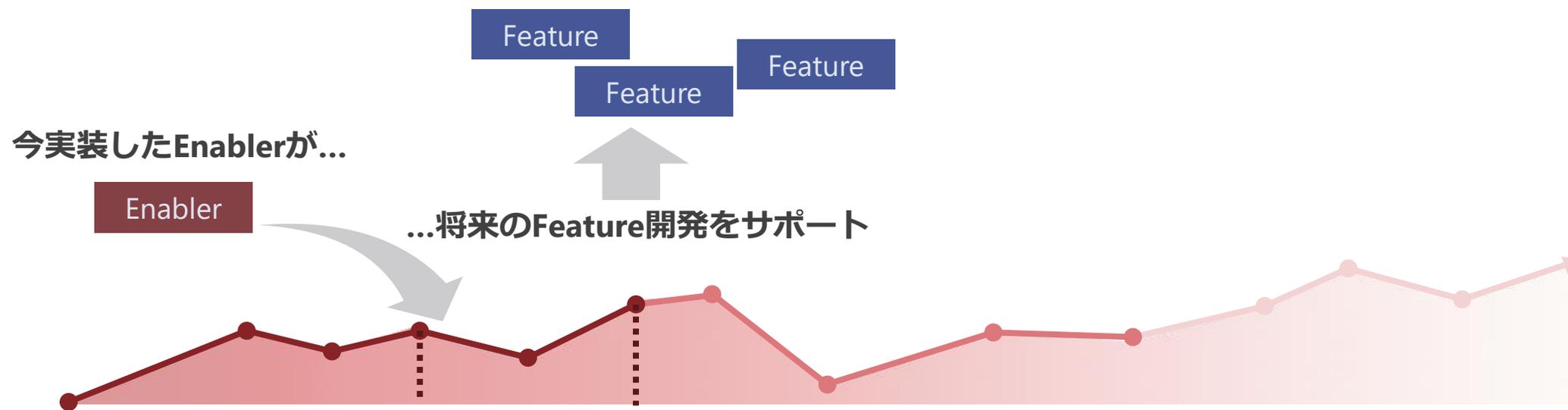


Agile Team
アジャイルチーム

Architectural Runway -アーキテクチャル・ランウェイ-

Featureの短期間でのリリースを支援するソースコードやコンポーネント、インフラ等の技術的な要素。

- ✓ Architectural Runwayを作るためのバックログはEnablerとして定義
- ✓ 継続的に維持・改善する
- ✓ 高速なデリバリーを実現するためのもの
- ✓ アジャイルチーム自身がEnablerを開発する



Architectural Runway導入時のポイント

1

BDUF(Big Design Up-front)の回避

- 最初は正解がわからない。不確実性に対処するために複数の選択肢を用意し、仮説検証の中で選択肢を狭めていく(Set-Based)
- 次のFeatureの増分を実装および検証するのに必要な場合にのみ、アーキテクチャを拡張する

2

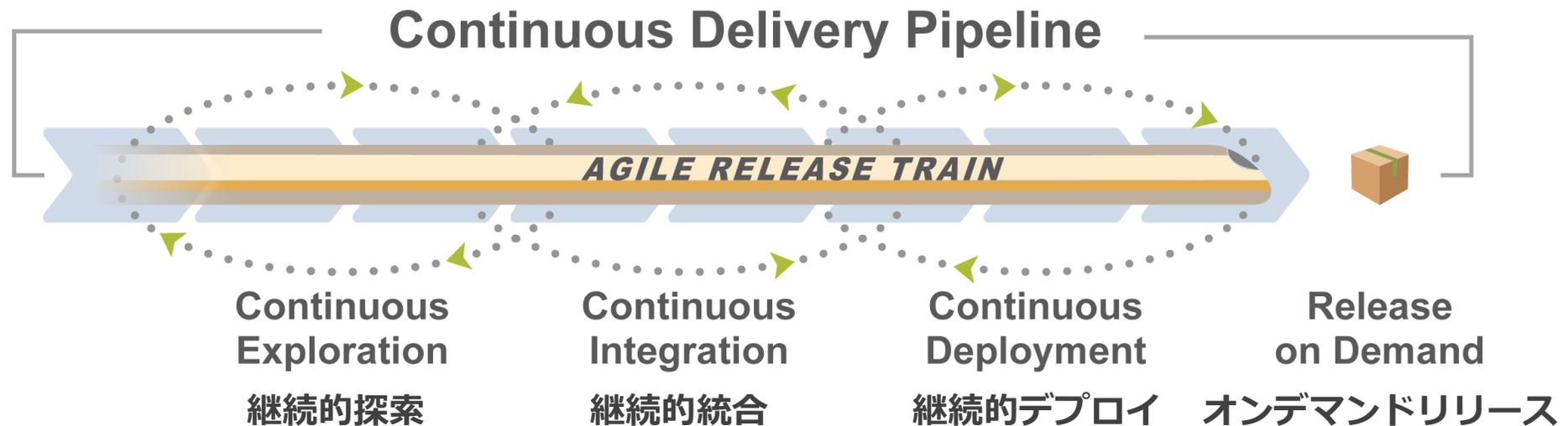
志向性と創発性のバランスを取る

- 創発的な設計に頼りすぎると「過度な再設計」、「システム統合の難航」、「ソリューションの重複」、「チーム間のコラボレーションの減少」が発生
- 志向的な設計が増えると、「チームの自律性の阻害」、「アジリティの低下」が発生

Continuous Delivery Pipeline -継続的デリバリパイプライン-

アイデアからリリースまでの、ワークフロー、アクティビティ、自動化を表すもの。

- ✓ 一連のバリューストリームをマッピングする
- ✓ DevOpsを用いて実装し最適化する
- ✓ 継続的に維持・改善する



Continuous Delivery Pipeline導入時のポイント

1 DevOpsマインドを浸透させる

- 迅速に変更して価値を届けたいDevと、なるべく変更しないで安定を求めるOpsは対立構造にある
- セキュリティを加えたDevSecOpsを実現する

2 ビルトイン品質を実現する

- 「作ったらテストする」ではなく「作ったらテストされていた」を目指す
- テストファースト
- 継続的かつ高頻度なE2Eテスト

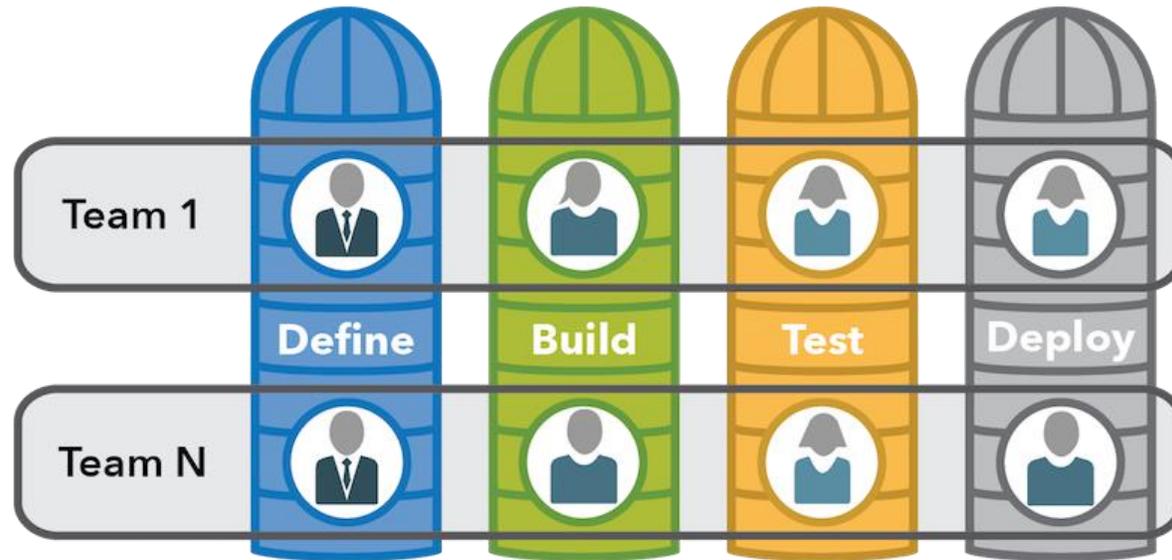
3 デプロイとリリースを切り離す

- デプロイは運用環境にシステムを適用すること、リリースは利用者が使えるようになること
- デプロイ頻度を上げつつオンデマンドリリースを目指す

Agile Team -アジャイルチーム-

短いタイムボックスで価値の増分を定義、開発、テスト、デリバリーする5～11人のスクラムチーム。

- ✓ 自己組織化&自己管理されていてクロスファンクショナル
- ✓ 複数のチームがコラボレーションしながら同じARTに乗る
- ✓ アーキテクチャ滑走路を実装しつつ創発的な設計もする
- ✓ DevOpsを実践する



© Scaled Agile, Inc.

Agile Team導入時のポイント

1 バリューストリームを中心に開発組織を考える

- アーキテクトは技術的視点で意見し、バリューストリームやARTの設計をサポートする
- 組織はアーキテクチャに作用することを認識する

2 アジャイル経験者を集める

- 単一チームのアジャイルが出来ないのに複数チームと同期が必要なアジャイルができるはずがない
- クロスファンクショナルでDevOpsが実践できるチーム

3 積極的にAgile Teamとコラボレーションする

- 要求、仕様、設計、ルールは一方通行にならないように
- 最終的に価値を作るのはアジャイルチーム。関係者は全力で彼らをサポートする

まとめ

- ✓ SAFeリーニアジャイルの原則に従う
- ✓ アジャイルチームとは対等
- ✓ リーダーシップを発揮しコラボレーションしながら良い設計を目指す