



複雑化する DevOps を可視化しよう
～ CI / CD を整えてからの次の一歩～

Takao Mitsuyoshi
Solutions Architect
Amazon Web Services Japan G.K.

自己紹介

光吉 隆雄 (Takao Mitsuyoshi)

所属 :

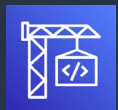
技術統括本部西日本ソリューション部

略歴 :

AWSソリューションアーキテクト

大手ゲーム会社の基盤開発にかかるDevOps推進

好きなAWSサービス :



AWS CodeBuild

西日本のエンタープライズ企業を担当するソリューションアーキテクトとして、新たに AWS の導入をご検討頂いているお客様 / 既に AWS をご利用頂いているお客様の課題解決を技術面から支援しています。



本日の目的

- CI / CD で起こる問題とはなにか？
- CI / CD の問題を分解して指標と測定対象を割り出す
- 指標と測定対象をもとに可視化する
- 可視化を利用して問題とどう向き合うか/闘うか



アジェンダ

- CI / CD の複雑化
- CI / CD の問題とは何か？

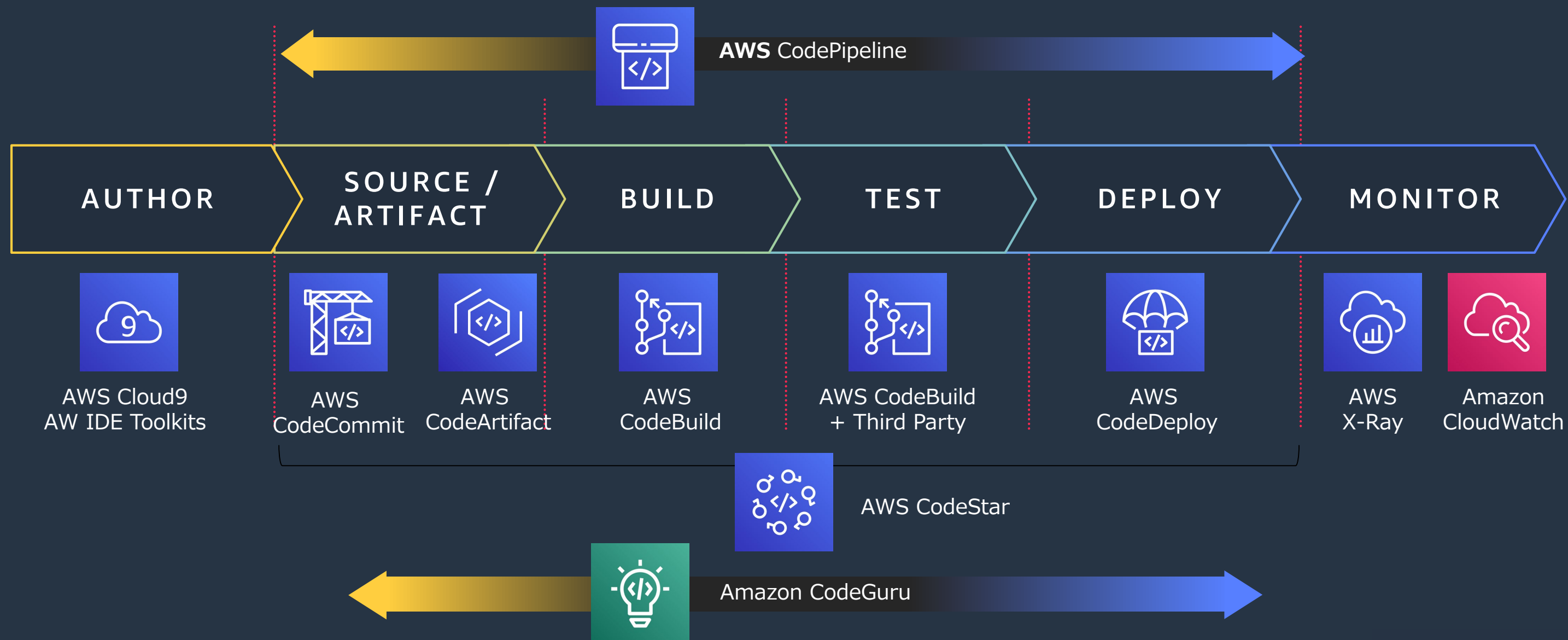
～幕間～

- 可視化のための指標づくり
- 指標を把握するための測定対象
- CI / CD を可視化してみる
- まとめ

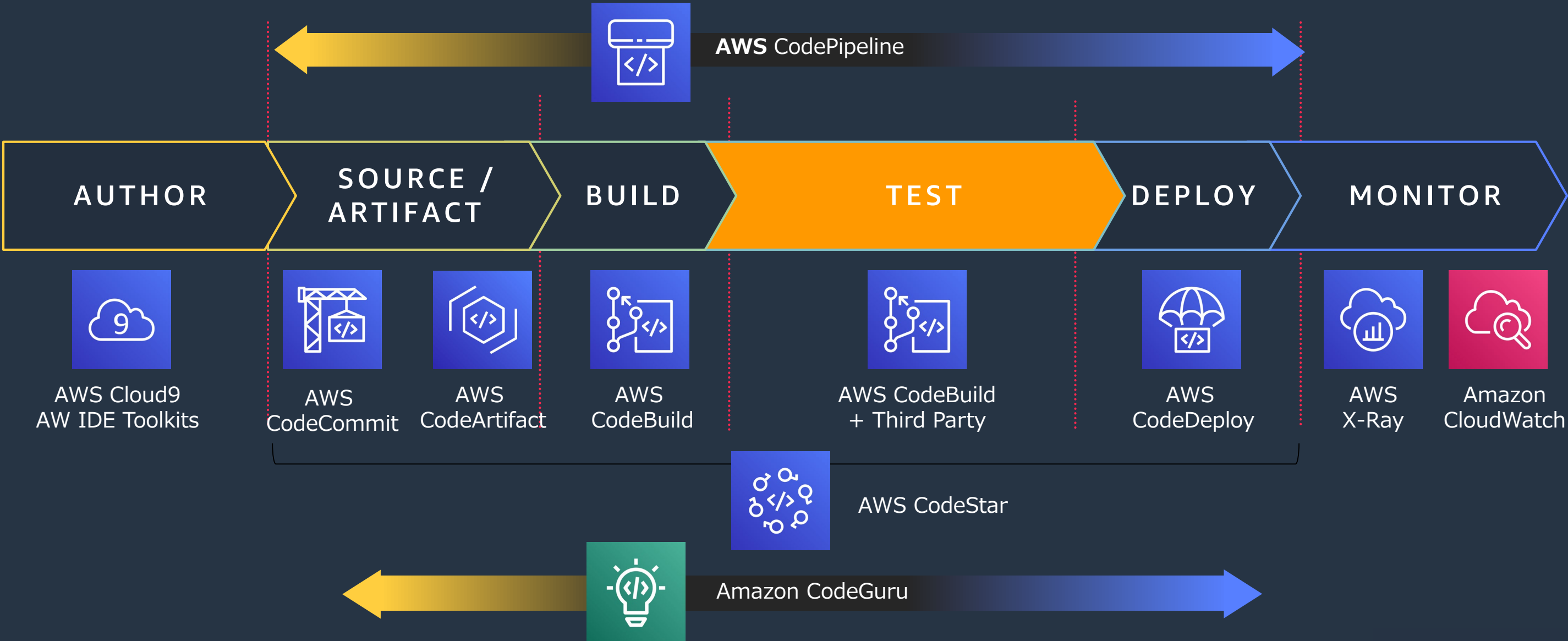


CI / CD の複雑化

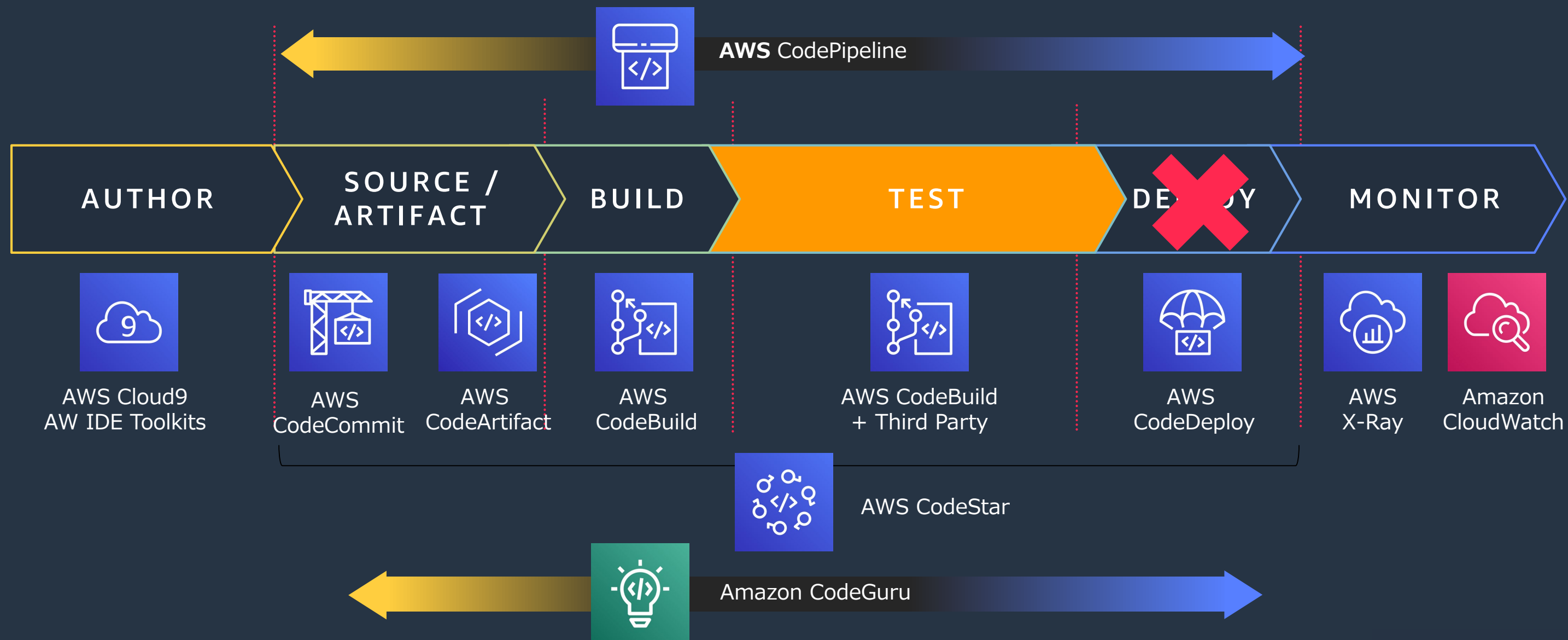
CI / CD を導入と複雑化



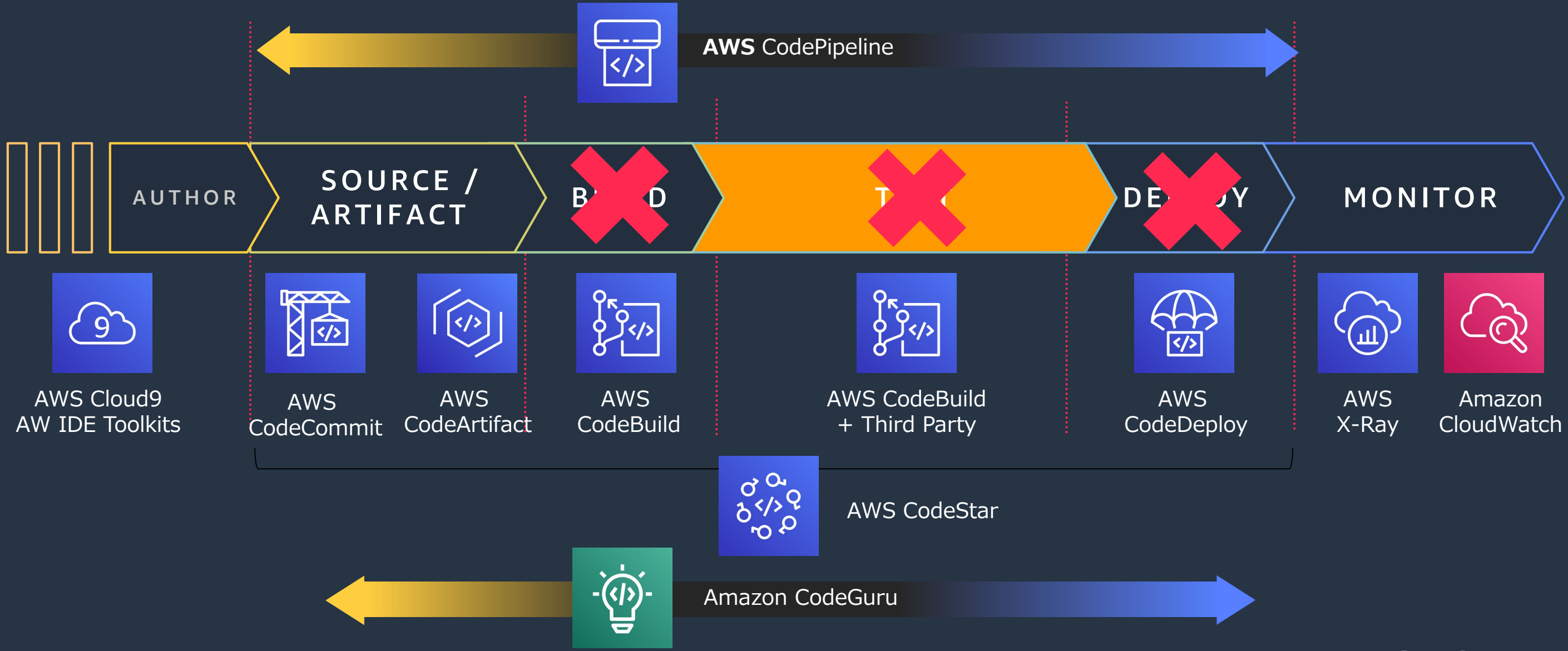
CI / CD を導入と複雑化



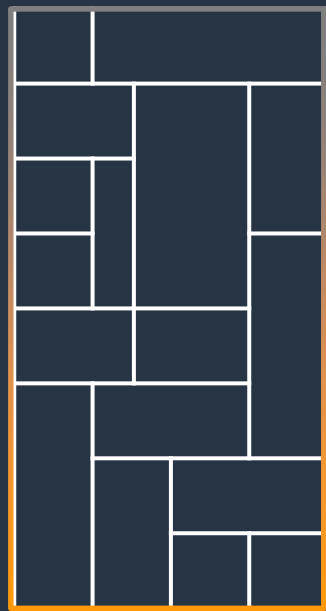
CI / CD を導入と複雑化



CI / CD を導入と複雑化



より CI / CD の目的を加速させたい . . .



- 好きな時にリリースしたい
- 変更のしやすさ (心理的安全性)
- 整理されたコード (生産性の向上)
- 柔軟なスケーリング
- すぐビルドしてすぐリリース

モノリスからマイクロサービスへ



柔軟なスケーリング



影響範囲の局所化



高速なデプロイ



よりフォーカスした
チーム



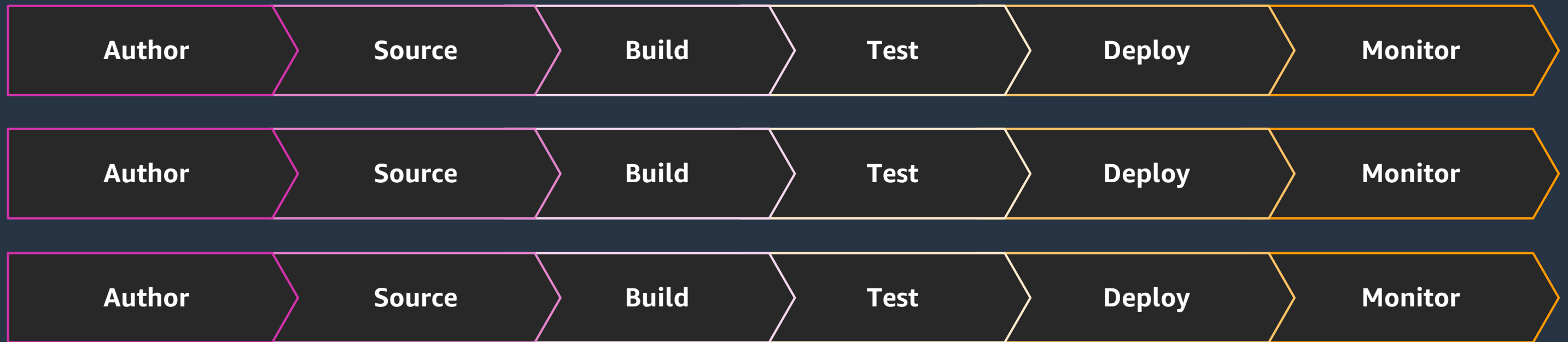
アジリティ



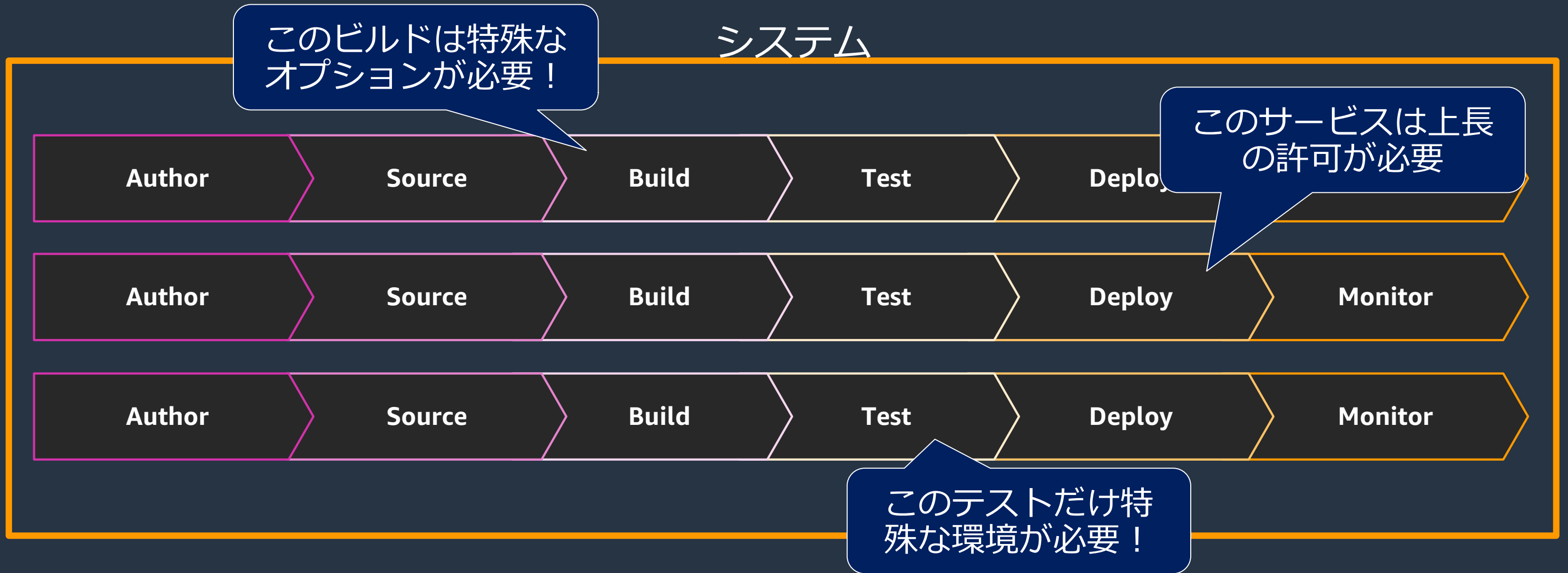
自由な技術選定

マイクロサービスによるパイプラインの増大

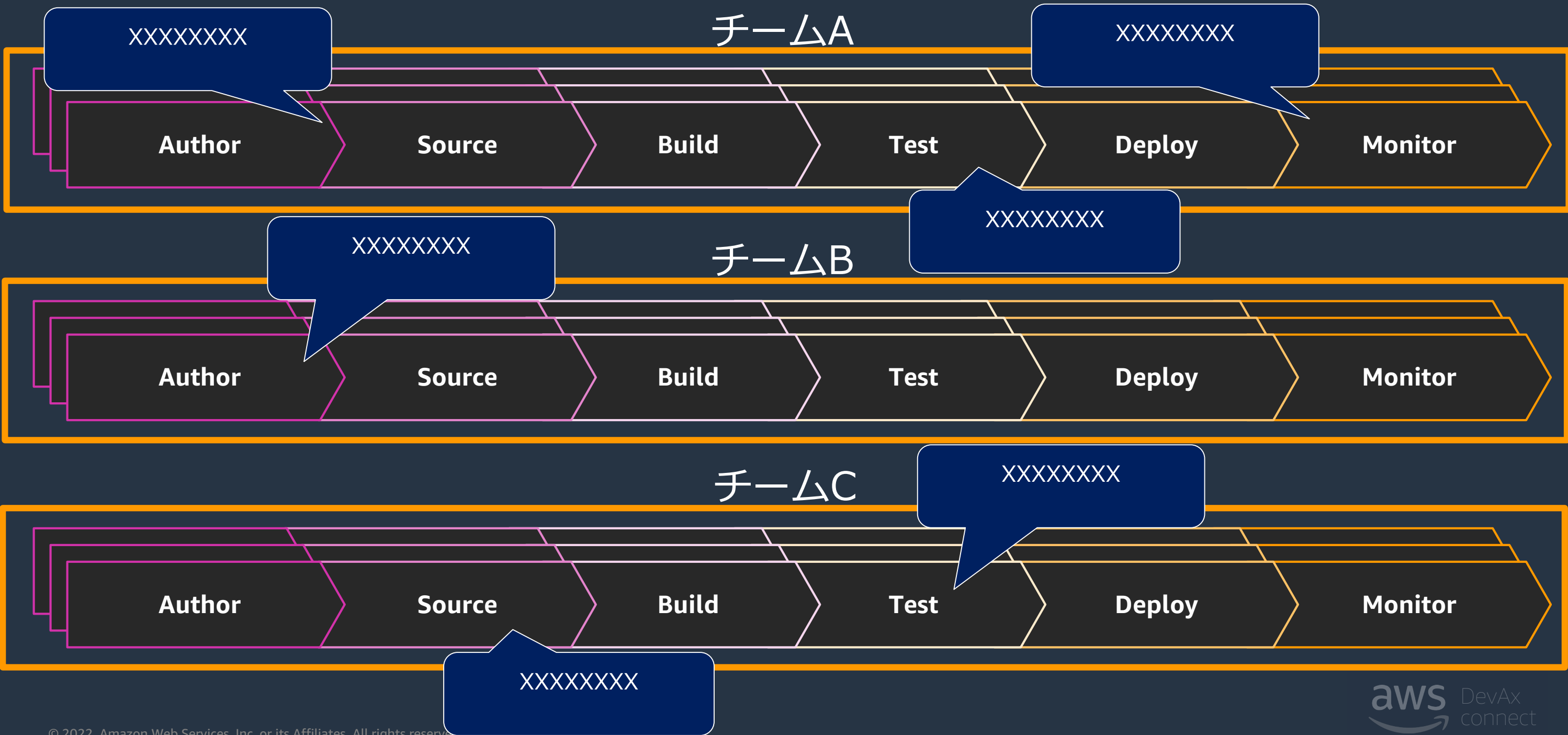
システム



マイクロサービスによるパイプラインの増大



マイクロサービスによるパイプラインの増大



CI / CD の問題とは何か？

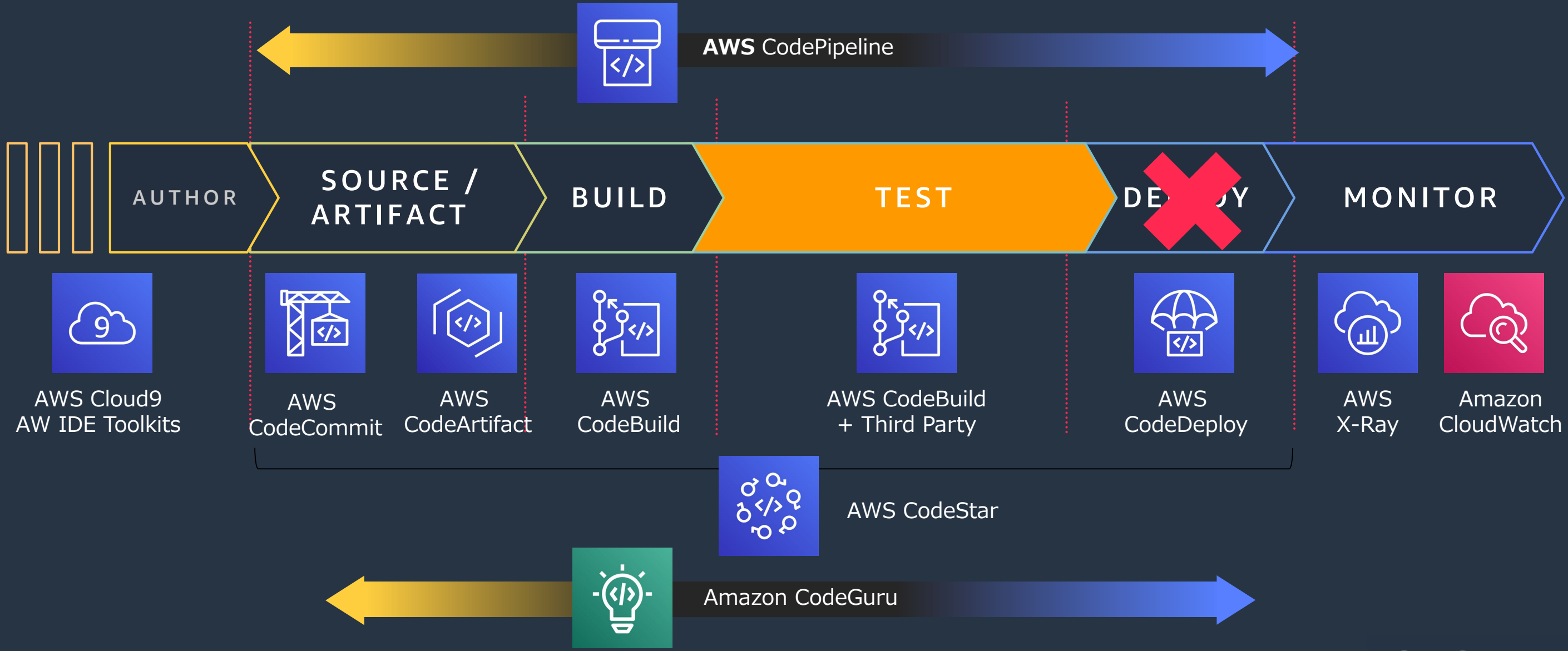
問題：

これらが正しく実施されているか？

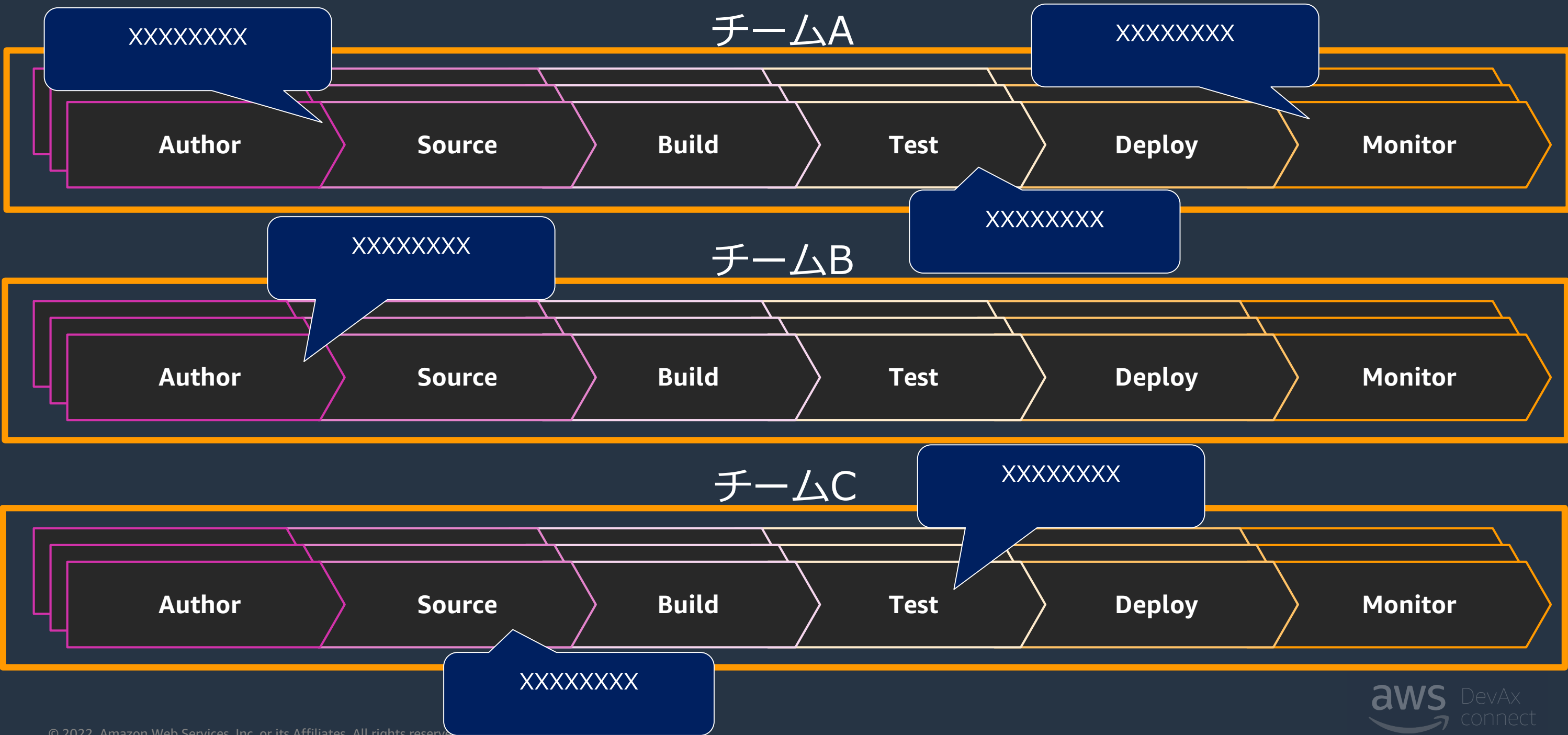
CI / CD の問題はチームが正しく動いているか？

- ソフトウェアのビルド・デプロイ・テスト・リリースというプロセスのあらゆる部分が関係者全員から見えるようにし、**共同作業をやりやすくすること**
- フィードバックを改善し、プロセスにおいて**できる限り早い時期に問題が特定されて解決されるようにすること**
- ソフトウェアの任意のバージョンを任意の環境に対して、**完全に自動化されたプロセスを通じて好きなようにデプロイ**できるようにすること

CI / CD を導入と複雑化

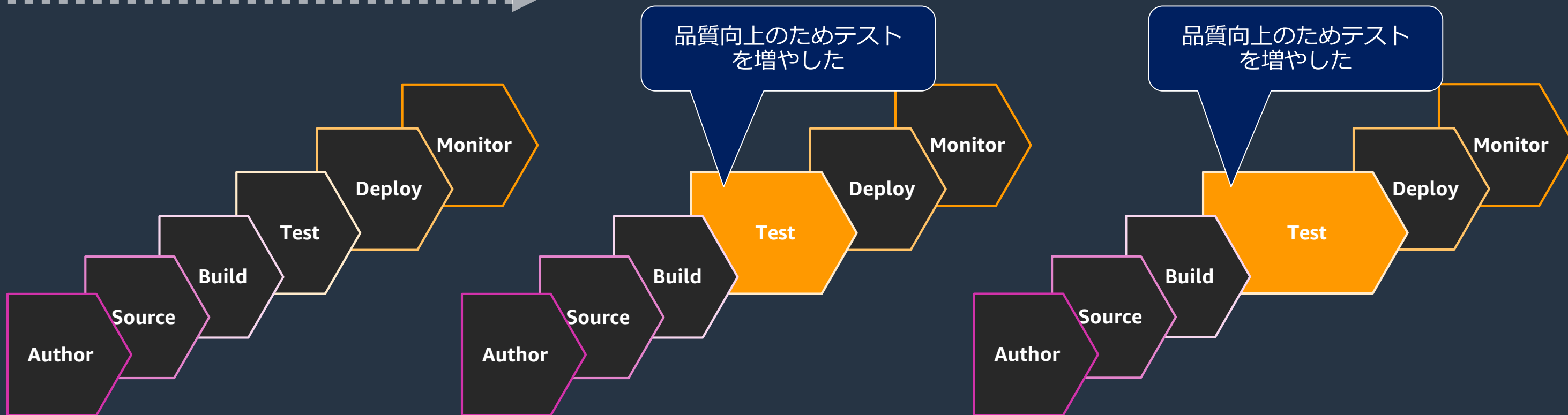


マイクロサービスによるパイプラインの増大



CI / CD の問題とはなにか？

イテレーション



時間軸

CI / CD の問題とはなにか？

イテレーション

変化は徐々に起こり「慣れ」てしまう
指標を明確にして可視化する必要がある

時間軸

幕間

可視化のための指標づくり

システムの可視化

可観測性

調査
トレー

可視化

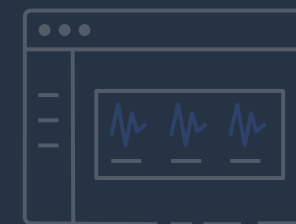
システムの状況を一箇所に集めて把握し
問題を素早く解決する



Log



Metrics



Profile



AI/Ops

セキュリティの可視化

SIEM (Security Information and Event Management)

散らばった
ワークの
し分析や

セキュリティの状況を一箇所に集めて把握し
問題を素早く解決する

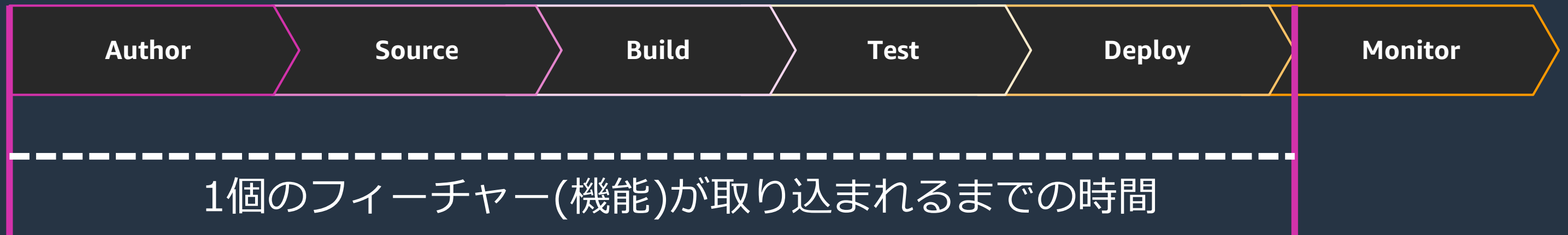
SIEM on Amazon OpenSearch Service

<https://aws.amazon.com/jp/blogs/news/siem-on-amazon-elasticsearch-service/>



CI / CD の指標とは何か？

機能の提供時間

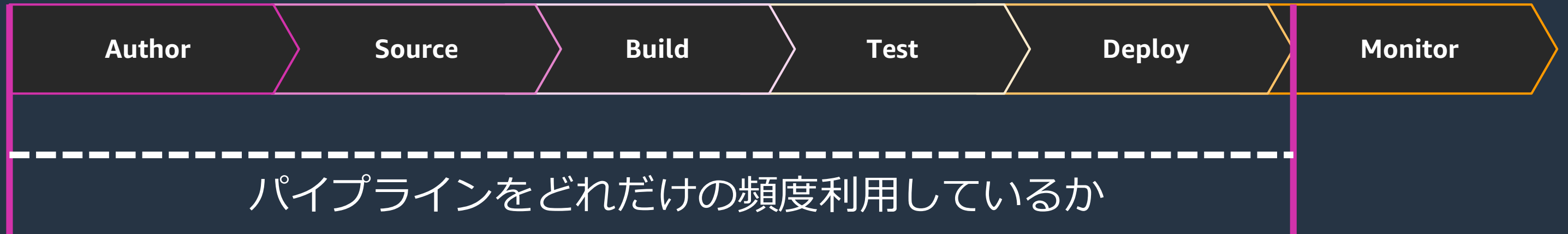


1個のフィーチャー(機能)が取り込まれるまでの時間

- チームのアジリティが想定しているものか？
- 機能の開発はパイプラインだけでなく、要件や仕様決定からのリードタイムをふくむべき場合も

CI / CD の指標とは何か？

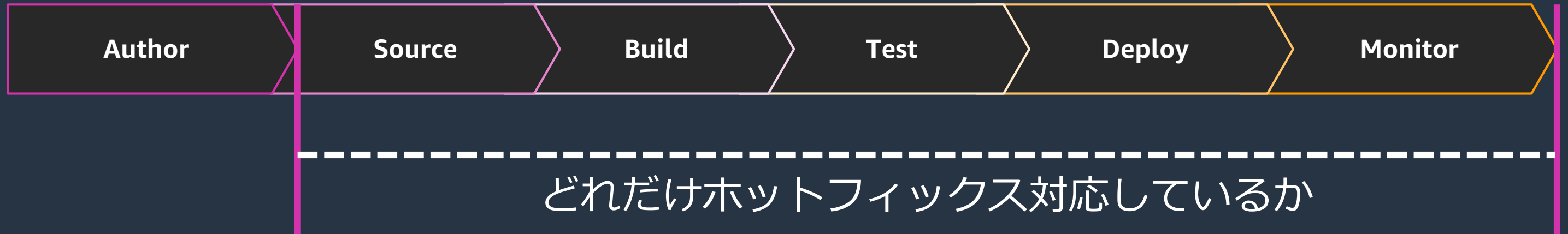
デプロイ回数



- アジリティを活かしてどの程度、活性化しているか？
- ライフサイクルの長さではなく頻度・回数

CI / CD の指標とは何か？

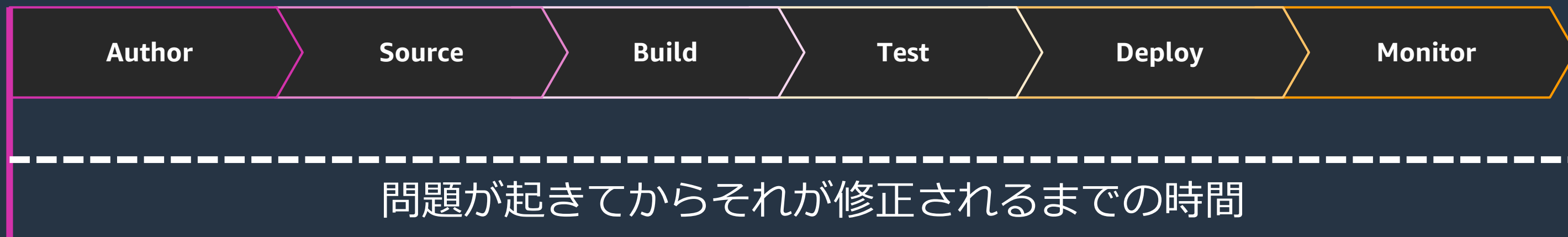
エラー発生率



- 通常デプロイとホットフィックスデプロイの割合は？
- そのチームがどの程度緊急対応に追われているか？
- テストの失敗とは混同しないこと

CI / CD の指標とは何か？

平均修復時間(MTTR)



問題が起きてからそれが修正されるまでの時間

- 問題を検知してからどの程度の時間で復旧できているか？
- 単純な修復時間だけでなく、具体的に何に時間がかかったか
- 調査や修正のための仕様策定のリードタイムも含まれる場合も

指標を把握するための測定対象

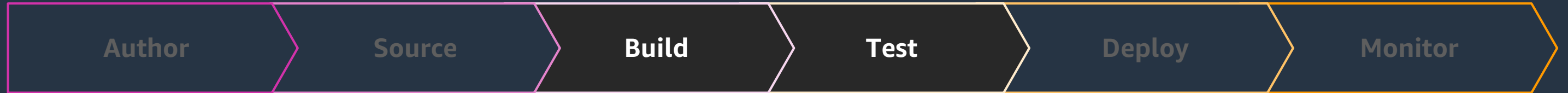
CI / CD の指標を把握するための測定対象



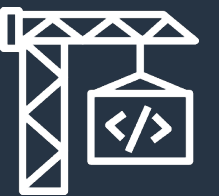
- 開発に紐づくチケットやかんばんなどのステータス変化
- 誰がどのリポジトリにコードをpushしているか？
- リポジトリ自体のコントリビュートは活発か？
- ホットフィックス対応をどの程度行っているか？



CI / CD の指標を把握するための測定対象



- ビルド/テストの成功率/失敗率は？
- ビルド/テストの頻度は活発か？
- ビルド/テストの実行時間はどの程度か？
- テストのカバレッジはどの程度か？
- ビルドサイズはどの程度か？



CI / CD の指標を把握するための測定対象



- デプロイの成功率/失敗率は？
- デプロイの頻度は活発か？
- デプロイ所要時間はどうか？
- 正常にシステムが動き続けているか？



指標と関係する測定対象のまとめ

| | 機能の提供時間 | デプロイ回数 | エラー発生率 | 平均修復時間 |
|---------|-----------------|------------|----------|--------------------|
| その他 | Issueステータス | Issueステータス | | Issueステータス |
| Source | 誰がPushしたか | ホットフィックス | ホットフィックス | 誰がPushしたか フィックス |
| Build | | | | 失敗率 サイズ |
| Test | | | | 失敗率 リジ |
| Deploy | 成功率/失敗率 所要時間 | 実施頻度 | 成功率/失敗率 | 成功率/失敗率 |
| Monitor | 正常稼働状態か | | 正常稼働状態か | 正常稼働状態か |

これらを使って CI / CD を一箇所に集めて把握し
問題を素早く解決する

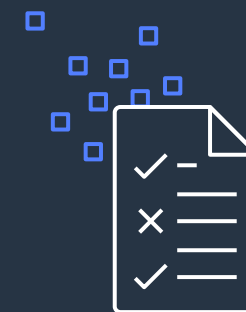
CI / CD の指標を把握するための測定対象の注意点

定量的な数値が指標にしにくい、もしくはするべきでないことも

- 変更コード量 / コミット数が多いと良いわけではない
- デプロイ頻度が高いから素晴らしいか



- あくまで CI / CD の問題を発見するもの
- あくまで CI / CD の改善のためのもの



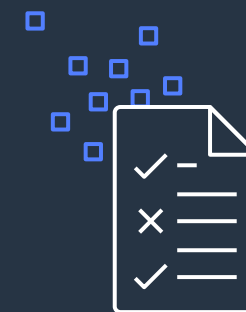
CI / CD の指標を把握するための測定対象の注意点

具体的な数値が出てくるとベストプラクティスに振り回されてしまう

- 10分ビルド
- 10分テスト
- トランクベース開発



- あくまでチームの目標や指標に対してのもの
- 近視眼的ではなく時系列によるチームの変化を見る



CI / CD を可視化してみる

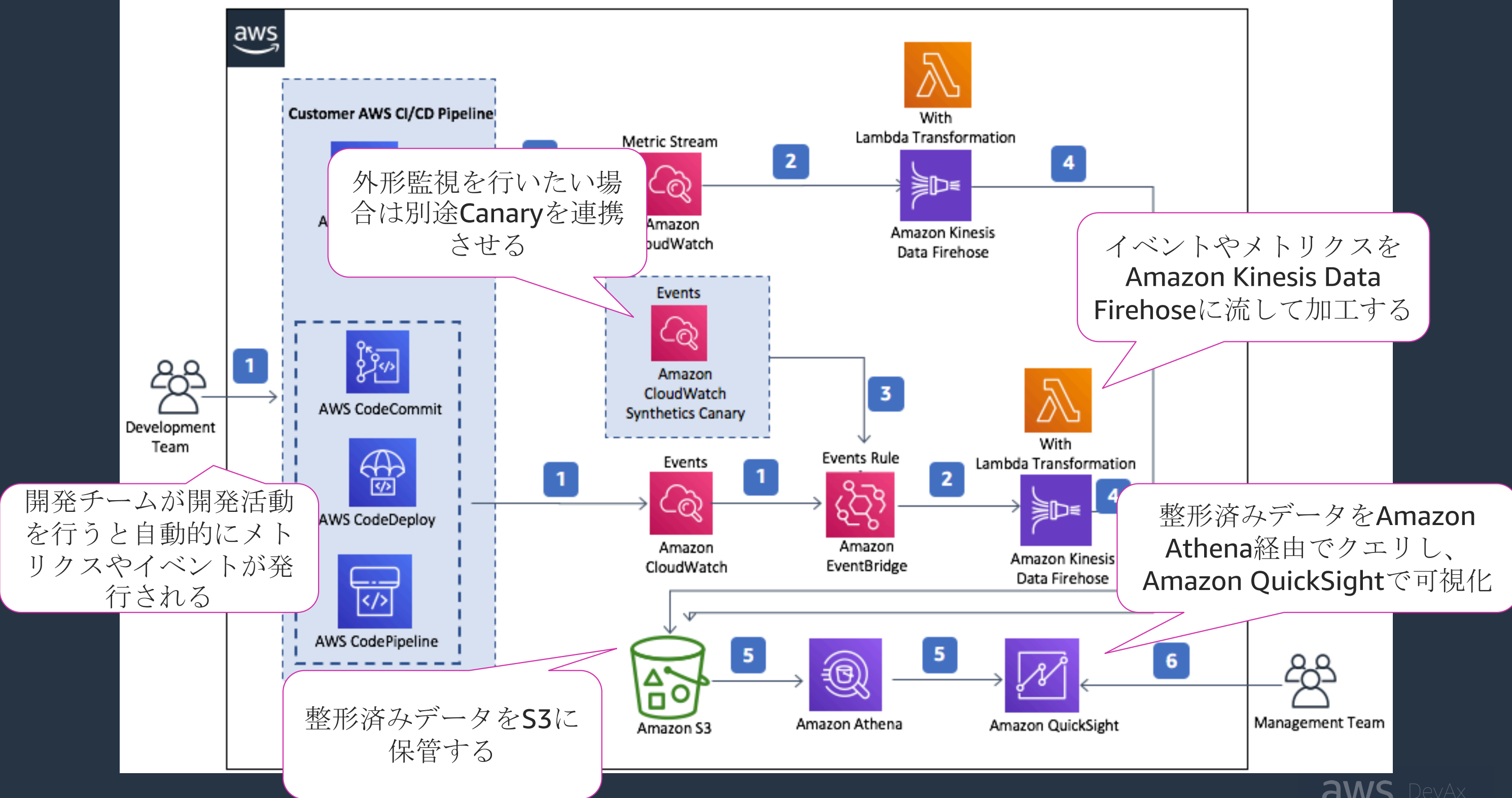
AWS DevOps Monitoring Dashboard

CI/ CD に関わるメトリクスを収集し、分析および可視化のプロセスを自動化することで、DevOps に関わる改善の指標を提示する

AWS DevOps Monitoring Dashboard

<https://aws.amazon.com/jp/solutions/implementations/aws-devops-monitoring-dashboard/>





外形監視を行いたい場合は別途Canaryを連携させる

イベントやメトリクスをAmazon Kinesis Data Firehoseに流して加工する

開発チームが開発活動を行うと自動的にメトリクスやイベントが発行される

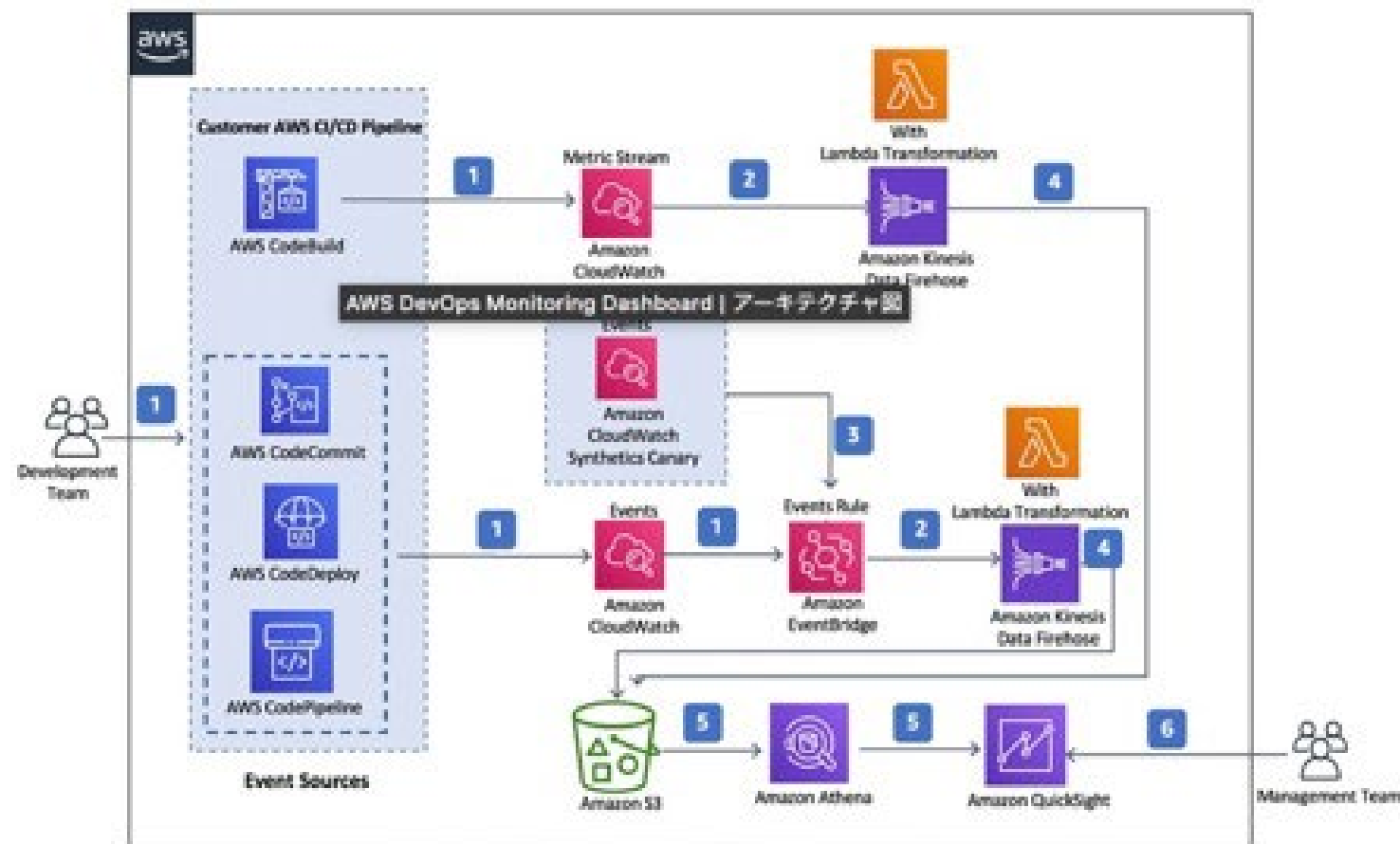
整形済みデータをAmazon Athena経由でクエリし、Amazon QuickSightで可視化

整形済みデータをS3に保管する

ソリューションのデプロイ

AWS ソリューション実装の概要

下図は、このソリューションの実装ガイドと付属の AWS CloudFormation テンプレートを使用して、自動的にデプロイできるアーキテクチャを表しています。



AWS DevOps Monitoring Dashboard

バージョン 1.1.0

リリース: 2021 年 6 月

作成者: AWS

推定デプロイ時間: 10 分

予想費用

[ソースコード](#)

[CloudFormation テンプレート](#)

実装ガイドを表示

AWS コンソールで起動する

AWS IQ エキスパートでデプロイする

デモ

欲しい指標を実現する

AWS DevOps Monitoring Dashboard
は標準では AWS のサービスしかメ
トリクスの収集対象にしていない

Github Actions を使って Github リポ
ジトリのメトリクスを AWS DevOps
Monitoring Dashboard に反映させる
など任意のメトリクスをカスタマイズ
することも

[https://aws.amazon.com/jp/builders-
flash/202204/devops-monitoring-dashboard/](https://aws.amazon.com/jp/builders-flash/202204/devops-monitoring-dashboard/)



CI / CD の可視化で複雑さと闘う

- DevOps 担当者の指標として利用
 - 問題があるチームの発見
 - アジリティの高いチームの分析
 - 各チームの知見の横展開
 - 各チームの状況を共有し合う環境の提供



CI / CD の可視化で複雑さと闘う

- イテレーションの振り返り時に利用
 - テスト、ビルド、その他の before / after を確認
- 月ごと、四半期などの大きなタームでの振り返りに利用
 - 指標の変化を確認してチームの変化を確認
- マネージメント層への訴求
 - 改善活動のための説得材料
 - KPIとして提示しやすい情報提供



まとめ

まとめ

- マイクロサービスやシステムの成熟によって CI / CD は複雑化する
- CI / CD の可視化はチームが正しく動いているかの可視化
- 可視化を行うには取るべき指標とそれにあった測定対象を決める
- 測定内容はあくまでチーム全体の改善のものとして扱う
- 可視化は DevOps 担当者だけでなくチーム内でも活用できる



参考資料

SIEM on Amazon OpenSearch Service

https://github.com/aws-samples/siem-on-amazon-opensearch-service/blob/main/README_ja.md

AWS DevOps Monitoring Dashboard

<https://aws.amazon.com/jp/solutions/implementations/aws-devops-monitoring-dashboard/>

自分の環境に合わせた DevOps のモニタリング環境を手に入れる方法

<https://aws.amazon.com/jp/builders-flash/202204/devops-monitoring-dashboard/>

Thank you!

Takao Mitsuyoshi

Solutions Architect
Amazon Web Service Japan G.K.