

CircleCIの3000 万件の ワークフローから得られた DevOpsに関する知見

2020.02.13

Developers Summit 2020 Tokyo

Solutions Engineer @ CircleCI

Noboru Kurumai

自己紹介



名前: 車井 登 / Noboru Kurumai

ポジション: Solutions Engineer

経歴: パッケージソフト開発

クラウドサービスエンジニア

 @kurumai

アンケートよろしくおねがいします！



アンケートにご回答いただいた方の中から抽選で1名
にApple Watchが当たります！！



CircleCI Japan
@CircleCIJapan

Developers Summit 2020が今年も開催されますね! CircleCIは
ゴールドスポンサーとして参加予定で、2月13日11:05 ~
DevOpsやCI/CDの傾向についてお話する予定で
す: event.shoeisha.jp/devsumi/202002...
ご参加予定の方、ぜひ、参加できない方のためにも、[#circlecijp](https://twitter.com/circlecijp)
で呟いていただければ幸いです 🍌



13-C-2 CircleCIの3000 万件のワークフローから得られたDevOpsに関する知見 | Developers Summit 2020

event.shoeisha.jp

“

CircleCI知ってましたか？

”

“

CircleCI Japanがあるの知ってましたか？

”

CircleCI社のご紹介

- 世界最大規模のクラウドCI/CD サービス
- より良いコードをより速く、簡単にリリースすることを可能に
- 2011年設立、サンフランシスコ本社
- 250人+の社員（米国、東京、英国にオフィス）
- 19年7月 5,600万ドルのシリーズDを実施、合計1億1,550万ドルを調達

Representative Customers

SONY

LINE

mercari



DeNA

facebook

続) CI/CD戦国時代

CI/CD 戦国時代

Google GCP Cloud Build



Cloud Build

継続的なビルド、テスト、デプロイを実現します。

ビルドを自動化する

スピードと柔軟性

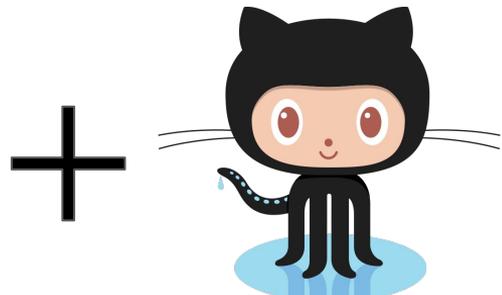
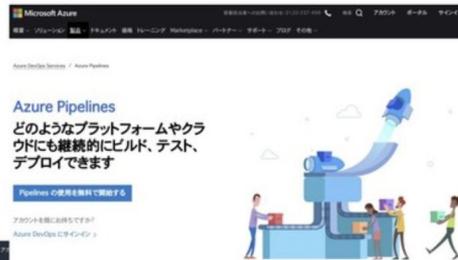
Cloud Buildでは、すべての言語でソフトウェアを迅速にビルドできます。豊富な機能 (VM、サーバーレス、Kubernetes、Firebase など) でのビルドが、テスト、デプロイやマルチプラットフォームワークフローの構築をさらに簡便にしてくれます。



AWS CodeBuild



Microsoft Azure Pipelines



CI/CDツールを 使う動機

- 自動化したい
- コストを削減したい
- 開発スピードを上げたい
- 品質も・・・
- セキュラ・・・

- 最高のチームづくりを
したいから

“

ハイパフォーマンスの開発とは何なのか？
自分達のチームはうまく出来ているのか？

”

“

優れたパフォーマンスを発揮しているチームは、
実際にはどのように稼働しているのか？

”

CircleCIの動作実績と対象データ

動作状況

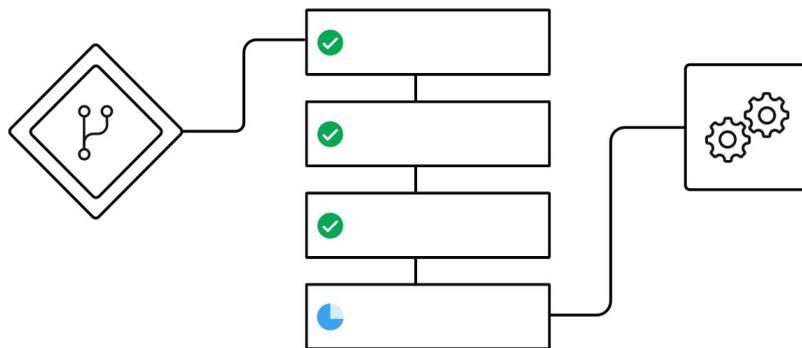
- 月間3,500万件以上のジョブを処理中
(Linux, MacOS, Windowsをすべて含むが、オンプレ版は含まない)

対象データ

- 2019年6月1日～8月30日に確認された3000万強のワークフローデータ
- 40,000以上のOrganization(≒組織・企業)
- 150,000以上のプロジェクト

少しだけCircleCI用語の説明

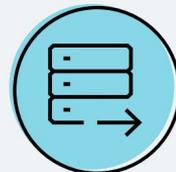
- ジョブはDockerコンテナやLinux VM、MacOSなど実行環境が起動する単位
- ワークフローは複数のジョブを組み合わせるCI/CDパイプラインの全体



4つのメトリクス



SOFTWARE DELIVERY PERFORMANCE



スループット

デプロイ頻度

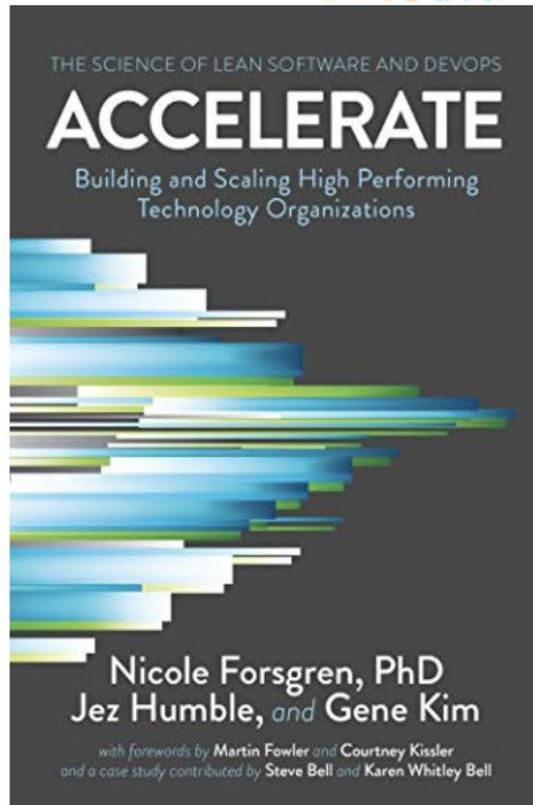
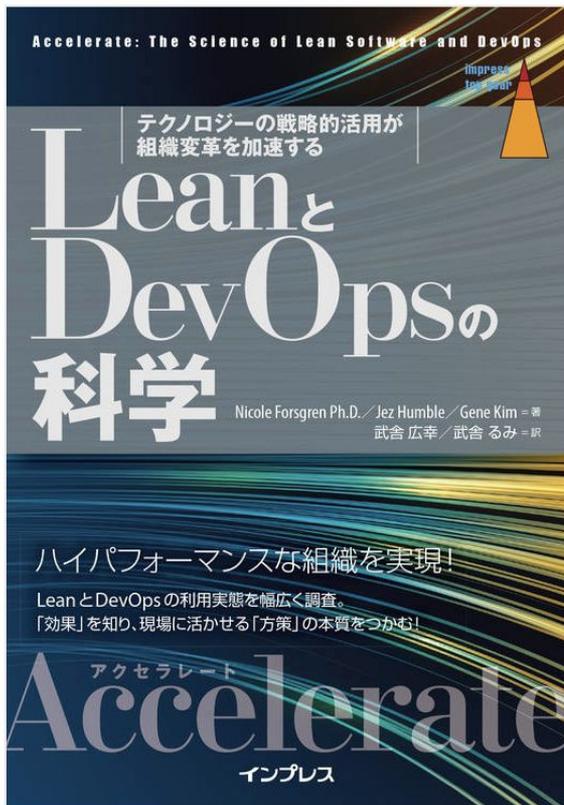
リードタイム



安定性

復旧時間

失敗の頻度



4つのメトリクス

スループット

変更のリードタイム



ワークフローの動作時間

デプロイ頻度



ワークフローが開始される頻度

安定性

平均修復時間 (MTTR)



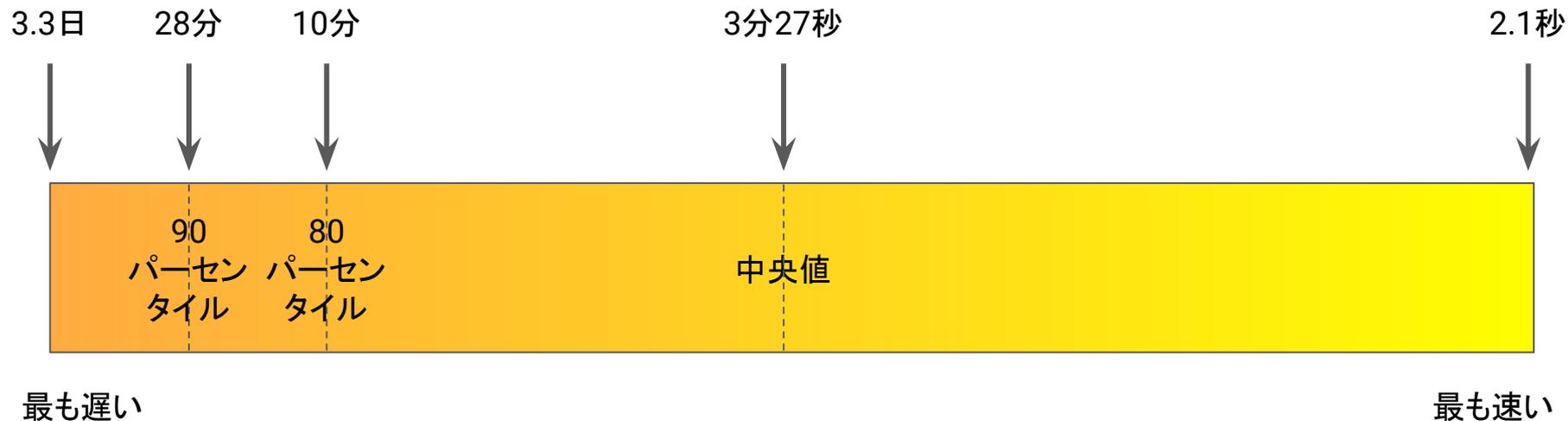
レッドビルドがグリーンビルドになるのに
要する時間

失敗の頻度



ワークフローの失敗率

ワークフローの動作時間(分析結果)



ワークフローの動作時間(考察)

- リードタイムはプロジェクトに強く依存するため、一概に何分(何秒)以内がベストとは言い切れない。
- ただし、3日以上というのは明らかに長過ぎるし、2.1秒では何も出来ない。
- CircleCIの経験上、最適なワークフローは数分から数十分程度

- ワークフローの実行時間とは、アーティファクトをデプロイするものだけでなく、そのワークフローの成果に到達する時間である。
- また、ワークフローの価値は、単にすばやく実行することではなく、開発者に適切なフィードバックがすばやく行われること。
- ワークフローには理想的な長さに普遍的な標準があるわけではない。
大事なものは、長さを短縮できるポイントを探すべき。
- 本データはCircleCI上で自動化された時間だけを計測している。もしまだ自動化されていないテストやセキュリティチェックなどがあれば積極的に自動化すべき。

デプロイの頻度 (2009年のFlickr)

10 deploys per day

Dev & ops cooperation at Flickr

John Allspaw & Paul Hammond
Velocity 2009



Dev and Ops

<https://www.slideshare.net/jallspaw/10-deploys-per-day-dev-and-ops-cooperation-at-flickr>

デプロイの頻度 (2012年のAWS)

AWS re:Invent

AWS LIVE

Werner Vogels
CTO, Amazon.com

Deployments at Amazon.com

11.6 Seconds mean time between deployments (weekday)	1,079 Max number of deployments in a single hour	10,000 Mean number of hosts simultaneously receiving a deployment	30,000 Max number of hosts simultaneously receiving a deployment
--	--	---	--

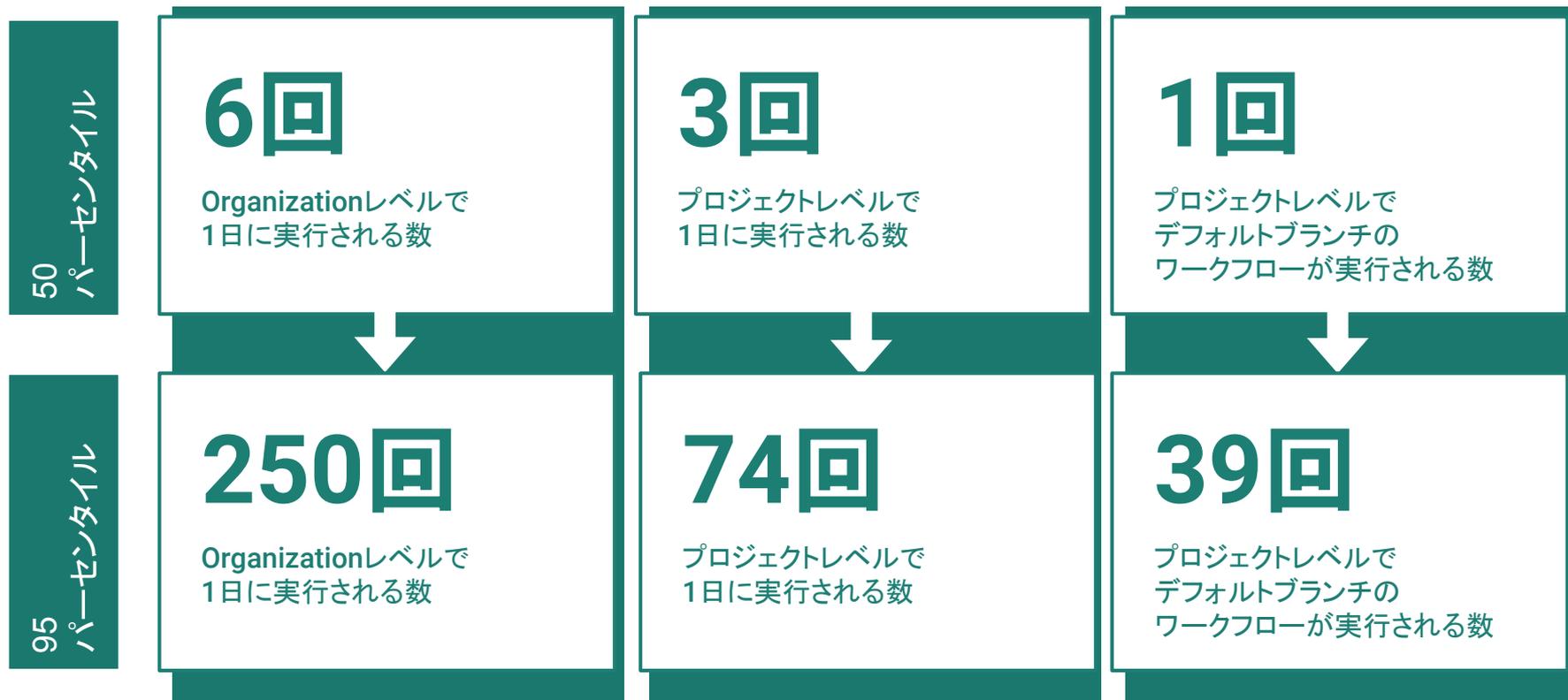
11秒に1回のデプロイ

最大で1時間1,079回のデプロイ

42:27 / 1:40:12

<https://www.youtube.com/watch?v=PW1lhU8n5So>

デプロイの頻度(分析結果)



デプロイの頻度(考察)

- Flickrのプレゼンテーション以降、数多くのプロジェクトが高頻度のデプロイに挑戦している。
- しかしながら、1日に10回以上デプロイしているプロジェクトは多くはない。

- 2019 Accelerate State of DevOps Reportでは、下記に分類されている。
 - 非常に優れたチーム(Elite): 1日に数回のデプロイ
 - ハイパフォーマンスのチーム: 1日から1週間に1回
 - 低パフォーマンスのチーム: 1ヶ月から6ヶ月に1回

- CircleCI を使用している組織は、必要なときにいつでもデプロイできるため、デプロイ回数は、組織の事情を反映したものになり、必ずしも優れた開発チームかどうかの指標にはならない
- 代わりに、組織がコードをテストおよび統合する頻度と、エラーが発生したときにどれだけ早く復旧できるかという指標は、開発チームのパフォーマンスが高いことを示します。



tarappo

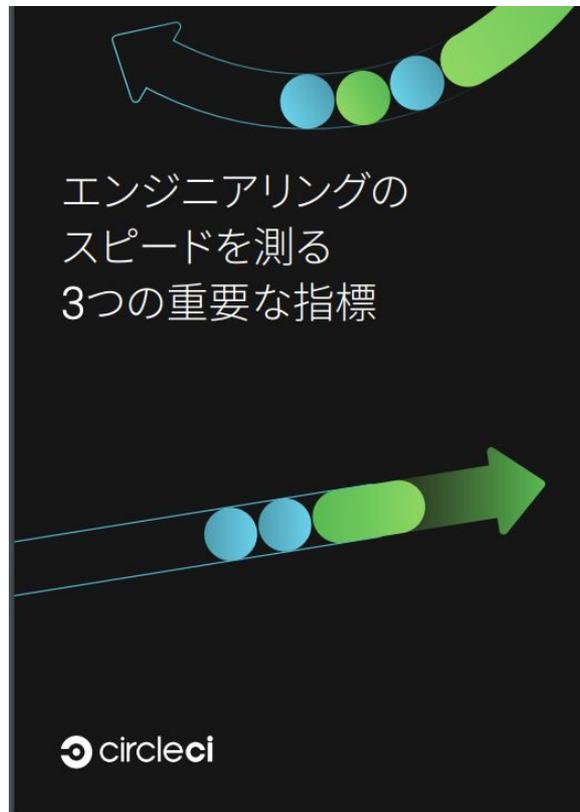
@tarappo



デプロイ頻度、リードタイム、MTTR、変更失敗率だけど。アプリの場合リリース頻度が高いことが必ずしも良いわけじゃないと思っているのだけど、どうなんだろう。昨今は自動アップデートをオンにしているユーザもある程度いるだろうし、頻度高いほうが良いといえば良いのかな。

午後2:00 · 2020年2月10日 · [Twitter Web App](#)

フィーチャーフラグ



フィーチャーフラグ

非常に高度なテストでも、実際のアプリケーションの複雑さを捉えることはできません。CircleCI では、テストによって潜在的なあらゆる問題を検出する代わりに、機能のリリースに LaunchDarkly のフィーチャーフラグを利用して、リリースの安全性を高めています。

LaunchDarkly (英語) の Tim Wong 氏は、次のように述べています。「フィーチャーフラグは、デプロイとリリースを切り離すためのものです。従来の企業では、この 2つはひとまとめになっており、デプロイが完了した時点でコードがリリースされます。フィーチャーフラグを使用すると、コードをデプロイするタイミングと、コードをリリースするタイミングをそれぞれ決定できます。実世界でテストしてから、ユーザーに提供する準備ができているかどうかを判断できるのです。これは非常に大きな違いです」

フィーチャーフラグのさらなる効果として、インシデント中に防御策として使用できることが挙げられます。たとえば、本番環境で問題が発生した場合、開発者はその原因となっている機能を無効にするだけで、修正が完了するまで問題のあるコードを効果的に隔離できます。通常、これは変更全体をロールバックするよりも迅速なため、全体的な安定性が向上します。

https://circleci.com/landing-pages/assets/2017-VelocityReport-Updated-070219_JA.pdf

平均修復時間: MTTR (分析結果)

すべてのブランチの場合

30日

17.5時間

1秒未満

State of DevOps 2019では、Eliteで1時間以内



最も長い

最も短い

おそらく、一日の終わりに失敗を検出して、次の日に修正していることが多い

平均修復時間: MTTR(分析結果)

デフォルトブランチの場合

1時間以下

15分



最も長い

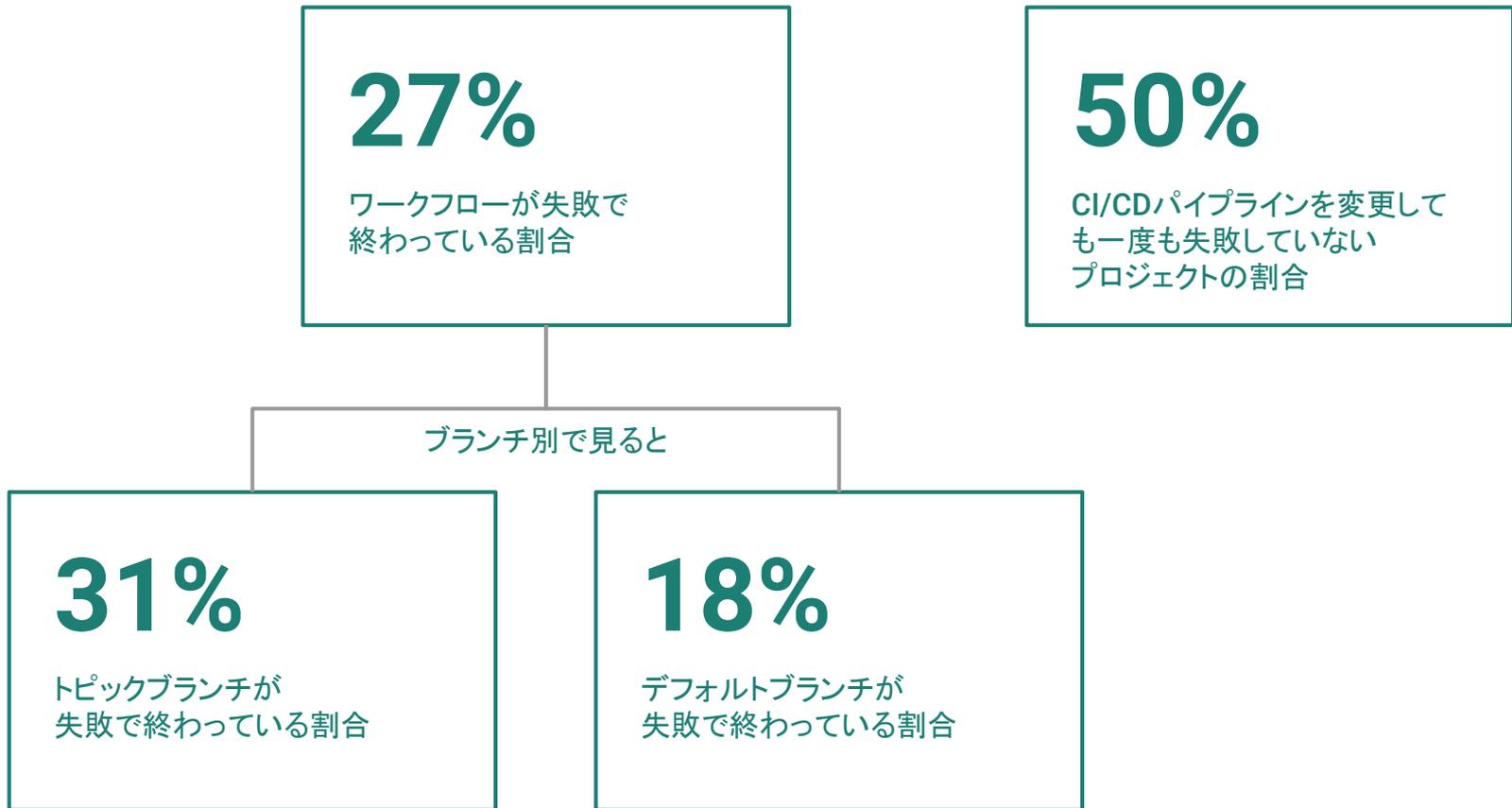
最も短い

30%のプロジェクトは90日間で一度も失敗していない(100% グリーン)

50%のOrganizationは1回で修正されている。

25%になると2回で修正されている

失敗の頻度(分析結果)



失敗の頻度(考察)

- 全体で27%のワークフローが失敗に終わっているものの、
多くはトピックブランチの失敗に起因している。
これはGit Flowなどのプラクティスが有効に働いているため
- デフォルトブランチでも18%が失敗に終わっているが、State of DevOps Report 2019ではEliteでも0-15%となっていて近い数値になっている。
- 50%のプロジェクトがCI/CDパイプラインの設定を変更してもビルドに失敗していない。
これは、設定ファイルを使い回すか、設定ファイルのパッケージング機能である Orbsを使っているため。
今後は、CI/CDパイプラインのベストプラクティスをいかに早く取り込むかが重要となってくる。

まとめ

ソフトウェアデリバリー パフォーマンスの要素	エリート*	平均的な CircleCI
デプロイ頻度 → ワークフローの開始頻度	オンデマンド： 1日に複数回デプロイ	1日に6件のワー
変更のリードタイム → ワークフローの期間	1日未満	3.5分
平均復旧時間	1時間未満	1時間未
変更による失敗率 → ワークフローの失敗率	0 ~ 15%	デフォルトブランチで 18%

回数は良いチームの指標にはならない
いつでもデプロイできる

最適なワークフローは
数分から数十分
開発者のフィードバック

修復時間を短くするには
プロアクティブな対応
と高度な自動化が必須
トピックブランチでの失敗
は恐れない。むしろ積極
的に結合する。
ベストプラクティスを取り
入れる

* 『2019 Accelerate State of DevOps Report』を出典とするエリートメトリクス。

次に向けて

- 言語特有の差異
 - 例えばPHPで書かれたプロジェクトの平均失敗率は7%だった
(全体平均は18%)
- ブランチ数の違い

世界のCircleCI ユーザーコミュニティ



ソウル



東京



大阪



シンガポール



福岡



バンコク



ジャカルタ

アンケートよろしくおねがいします！



アンケートにご回答いただいた方の中から抽選で1名
にApple Watchが当たります！！

Thank you.

