Engineer

変更のリードタイム

⧗ 変更をコミットしてから本番にデプロイするまでの時間

どのようなものか

変更のリードタイムは、変更をバージョン管理にコミットしてから、その変更を本番にデプロイするまでの経過時間です。DORAは、ソフトウェアの変更がどのくらい流れているかを調べるために、この時間を使います。実際に作業した時間と、次の工程を待った時間の両方が含まれます。月曜の朝にコミットし、火曜の午後に本番で動き始めたなら、夜間やレビューを待っていた時間も含みます。

始点はコミットの記録です。顧客から要望が届いた時刻、実装を始めた時刻、機能を利用者に公開した時刻を測ると、それぞれ別の問いに答えられます。新しいコードを本番に入れて、機能のスイッチは切ったままにすることもあります。コードの配置と公開は別々の出来事として記録できます。複数地域へ順に出す場合は、どの状態を成功した本番デプロイと呼ぶかをあらかじめ決めます。

まず対象サービスと本番の宛先を決め、集計に入れる変更を定義します。それぞれの変更を、条件を満たし、その変更を含んでいる最初の本番デプロイと結び付けます。複数のコミットを一つにまとめると、対象の履歴にまとめたコミットだけが残ることがあります。そこから測る場合、統合前のレビュー時間は入りません。比較の意味が分かるように、使った始点を明記します。

概念を判断に生かす

「変更のリードタイムとは？」を実際の判断に生かしましょう。その概念を、開発または購入するシステムに結び付けて考えるお手伝いをします。 変更を検証し、リリースし、元に戻せる仕組みを整え、金曜の深夜ではなく火曜の午後にお客様へ届けられるようにします。

サービスを見る お問い合わせ →

用語の背後にある前提を検証する

解説とリンク先の実績を参考に、プロジェクトについての質問を整理してください。ツールを選ぶ前に、前提を一緒に検討できます。 GitHub ActionsでGIS SaaSのデプロイを自動化 — デプロイ、データ、報告を手作業から解放

GitHub ActionsでGIS SaaSのデプロイを自動化 お問い合わせ →

なぜ重要なのか

待っている時間は、作業している時間より見えにくいことがあります。ビルドに六分かかり、変更がその後二日間デプロイの順番を待つとします。コンパイラを速くしても、全体への効果は小さくなります。レビューの完了、成果物の作成、事前検証、承認、本番に届いた時刻を記録します。少数の実例を並べるだけでも、どこに待ち行列があるか分かります。

この指標はチームの届け方を検討するために使います。小さな修正と、複数のサービスを横断する変更では条件が異なります。個人の順位にすると、簡単な変更を選んだり、コミットを遅らせたりする動機になるかもしれません。同じサービスを、具体的な手順変更の前後で比べます。普段の事例と遅い事例の両方を見て、本番障害や復旧の状況も確認します。

承認の待ち時間を減らせば、重要な修正を早く届けられる場合があります。それでも不具合が検出できるか、障害を復旧できるかを確かめます。DORAは、流れの指標と障害や復旧の指標を並べています。結果を一緒に読むと、工程が改善したのか、問題が次の工程へ移ったのかを検討できます。

実際の活用例

GISソフトウェアのデプロイを自動化した事例では、手作業だったテスト、ビルド、デプロイをGitHub Actionsで実行できるようにしました。公開されている記録には、リリースの期間が短くなり、設定をそろえたことが書かれています。コミットごとの時間は示されていません。同様の工程を測るなら、ソースの履歴と本番の記録を結び付けるところから始められます。

あるサービスに、午前九時と十時半にコミットが一つずつあるとします。両方を含むパッケージが午後二時に本番で動作を始めました。リードタイムは、それぞれ五時間と三時間半です。正午に事前検証の環境へ入っていた場合、その時刻は経路を説明する材料になります。計測を終えるのは本番の条件を満たした時点です。以前のビルドが失敗していた場合は、調査のため履歴に残します。

まず他の担当者が確かめられる記録を残します。含まれるリビジョン、パッケージの識別子、環境、完了を確認した時刻です。これらをソースの履歴と突き合わせます。一つのリポジトリで複数サービスを扱うなら、デプロイの記録に対象サービスを含めます。ツールや配布方法を変えた後は数件を手で確かめ、同じ意味の時刻が使われていることを確認します。

Gitでは、作者の日付とコミットを登録した人の日付は別の項目です。履歴を書き換えると一致しない場合があります。どちらを使うかを記録し、タイムゾーンをそろえて負の時間がないか調べます。復旧のため同じパッケージを再度配置しても、その変更が最初に本番に届いた時刻は変わりません。

問題が起きるところ

自動化されたジョブの終了は、別のシステムへ依頼を送ったことだけを表す場合があります。インスタンスはまだ起動中かもしれません。本番で動作したこと、所定の割合のトラフィックが切り替わったことなど、サービスにとって意味のある終点を定義します。同じサービスでは条件をそろえ、変更時はいつから新しい条件になったか残します。

レポートからは、本番に届かなかった変更が見えなくなります。中止した変更には終点がなく、作業中の変更はまだ待っているかもしれません。件数と滞留時間を別に記録します。成果物とコードを結び付けられない場合は、計測できなかった件数も記録します。値をゼロにすると待ち時間がなかったという別の意味になります。

平均値だけでは、長く待った例が埋もれます。代表的な値と、遅い側の傾向、対象件数を並べます。件数が少ないときは精密に見える数値に頼らず、実際の時系列を一件ずつ見ます。特定の工程で繰り返し待っているなら、そこで改善を試し、その後も同じサービスを同じ条件で測ります。

参考資料

- DORAのソフトウェアデリバリー指標は、変更のリードタイムを定義しています。
- DORAのバリューストリーム可視化ガイドは、作業、待ち時間、引き継ぎの記録を説明します。
- Gitのコミット出力形式は、作者とコミッターの日付項目を示しています。
- DORAのデプロイ自動化は、デプロイを再現可能にする手順を扱います。



用語集

実際の制約に合わせて選ぶ

デプロイの流れ、運用上のリスク、緊急対応の担当者をお知らせください。検査、復旧手段、明確な責任の所在を伴う変更の範囲を定められます。

料金を見る お問い合わせ →

用語を実務に結び付ける

「変更のリードタイムとは？」がお客様の組織にどう当てはまるかをご相談ください。現在のシステムの具体例が一つあれば、話を始められます。

サービスを見る

welcome@engineer.company +45 50 10 61 84 デンマーク、コペンハーゲン

このページを持ち歩く

連絡先カード

心を込めて丁寧に制作： D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

著作権 © 2025 – 現在 · Engineer ApS · すべての権利を留保します。

印刷元： https://engineer.company/ja/glossary/change-lead-time/