

効果測定の4指標表

研修後に測る指標と、研修前との突き合わせ方です。

レビュー工数を1つの数字で追うと、うまくいきません。工数は測る人の記憶に依存し、改修の大きさでばらつき、比較にならない数字が残ります。

代わりに、GitLab の MR データから機械的に取れる4つに限定します。人が入力する欄を1つだけ残し、残り3つはデータから取ります。入力欄が増えるほど、3か月後には誰も書かなくなります。

4指標

N o	指標	定義	取り方	人の 入力
1	MR あたりの AI 指摘件数と採用率	1つの MR で AI が出した指摘の件数と、そのうち対応した件数の割合	指摘件数は AI レビュージョブの findings.json を artifact に残して数えます。採用率は委譲判定シートの「即修正」の件数を分子にします	採否 だけ
2	人が採否を判断した件数	指摘のうち、CI で自動的に落ちたものを除いた件数	指摘の総数から、ゲートジョブが落とした P0 と P1 の件数を引きます	なし
3	ゲートで止まった MR の割合	ゲートジョブが失敗した MR の数 ÷ 全 MR の数	パイプラインのジョブ結果から数えます	なし
4	マージまでの時間と手戻り MR 率	MR 作成からマージまでの時間の中央値と、マージ後に同じ箇所を直す MR が出た割合	created_at と merged_at の差。手戻りは同じファイルに触る後続 MR を数えます	なし

指標2が、回答書に書かれた課題に一番近い数字です。「品質は上がったが工数削減は限定的」は、指標2が減らないまま指標1が増えている状態です。指標2を減らす手は、基準表を厳しくすることではなく、機械で判定できる指摘を CI 側へ移すことです。指標3が上がって指標2が下がれば、狙いどおりに動いています。

取り方

GitLab の REST API から取ります。エンドポイント名は下のとおりですが、自社インスタンスの版で使えるかは確認してから使ってください(要確認)。研修の調査で一次情報を取得できているのは、pipeline schedules とプロジェクトアクセストークンと push options の3つだけです。

取りたいもの	エンドポイント	主なクエリ
MR の一覧と作成・マージ日時	GET /projects/:id/merge_requests	state=merged 、 created_after 、 created_before 、 per_page

取りたいもの	エンドポイント	主なクエリ
MR ごとのパイプライン	GET /projects/:id/merge_requests/:iid/pipelines	
ジョブの成否	GET /projects/:id/pipelines/:pipeline_id/jobs	scope=failed
ジョブの artifact(findings.json)	GET /projects/:id/jobs/:job_id/artifacts/:artifact_path	
MR で変更したファイル	GET /projects/:id/merge_requests/:iid/changes	手戻りの判定に使用します

認証は api スコープのプロジェクトアクセストークンです。masked variable に置きます。

集計の単位は週です。日次にすると改修の粒度のばらつきに埋もれます。月次にすると、施策を入れてから効果を見るまでが遅すぎます。

ビフォー値との突き合わせ

ビフォーは研修前の8週分です。突き合わせの前に、次の3つを揃えてください。揃っていない数字を並べると、施策の効果ではなく期間の性質を見ることになります。

揃えるもの	決め方
対象リポジトリ	ビフォーと同じリポジトリだけを数えます。研修で使った演習リポジトリは混ぜません
対象 MR の絞り込み	Draft のまま閉じた MR、リリース用のマージ MR、設定ファイルだけの MR を外すか入れるかを最初に決めて、両側で同じにします
指摘件数の数え方	ビフォー側は現行の AI レビュージョブが MR コメントに投稿した件数です。アフター側は findings.json の配列長です。数え方が違う期間は、切り替えた週に印を付けます

比較する形です。

指標	ビフォー8週(研修前)	アフター4週(研修後)	差
1 MR あたりの AI 指摘件数(中央値)			
指摘の採用率			
人が採否を判断した件数(1 MR あたり)			

指標	ビフォー8週(研修前)	アフター4週(研修後)	差
ゲートで止まった MR の割合			
マージまでの時間(中央値)			
手戻り MR 率			

ビフォー側にゲートジョブは存在しないので、4行目のビフォーは 0% です。0% から上がることが悪い変化ではありません。止まった分だけ、人の目を通らずに済んだということです。

アフターの初回値

ハンズオン2で数える「仕込み欠陥の検出数 ÷ 指摘の総数」を、アフター側の初回値として置きます。既知の欠陥が入ったコードに対して、レビューが当たるかを測る形です。過去の改修10本を使ったバックテストの代わりになります。

patterns.md のタグ集計は補助です。タグを付けるのは人なので、主指標には置きません。

数字が揃わない場合

ビフォー値を正確に取るのは難しいという話が先方から出ています。8週分が揃わないときは、4指標の表だけを渡して、ハンズオン2の検出率を唯一の実測値として扱ってください。揃わない数字を埋めた表を作るより、実測が1つある表のほうが後で使えます。

出典: <https://docs.gitlab.com/ci/pipelines/schedules/> / https://docs.gitlab.com/user/project/settings/project_access_tokens/

