

回帰テストをほぼ100%自動化した世界

Automation Test推進パート
橋本 悠我

LINE Fukuoka

自己紹介



橋本 悠我

LINE Fukuoka株式会社
Automation Test推進パート

経歴

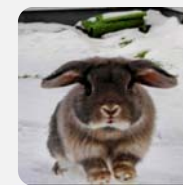
- ・プログラマ → システムエンジニア → QAエンジニア
- ※テスト自動化はほぼ未経験

現在

- ・ E2E自動テストの設計・実装・保守（モバイル/WEB）
- ・ 社内へテスト自動化の推進
- ・ CI環境、ジョブ設計・構築

趣味

- ・ サッカー
- ・ うさぎと遊ぶこと



Contents

1. 今日伝えたいこと
2. 発表の前提
3. 回帰テストをほぼ100%自動化した世界
4. テスト自動化初心者を知ってほしいポイント
5. 結論

1 今日伝えたいこと

いつやるの？ 回帰テスト自動化

成功事例もあることを知ってほしい

1 今日伝えたいこと

~~回帰テストをほぼ100%自動化=成功~~

テスト自動化で目的を継続的に達成=成功

1 今日伝えたいこと

なぜ回帰テスト自動化

回帰テスト自動化の重要度

- アジャイル・DevOps
 - 正しく動くソフトウェアが重要
 - 素早いリリース
 - 素早いフィードバック

テスト自動化のやりやすさ

- ノーコードのテスト自動化ツール
- CI/CD環境
- コンテナ(Docker,k8s)
- クラウドサーバ

回帰テストの実行頻度が増し、**回帰テスト自動化の重要度が高くなっている**
テスト自動化もやりやすくなっている

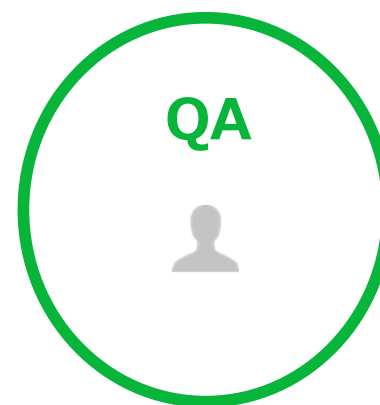
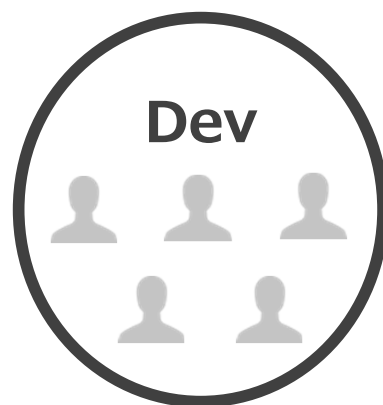
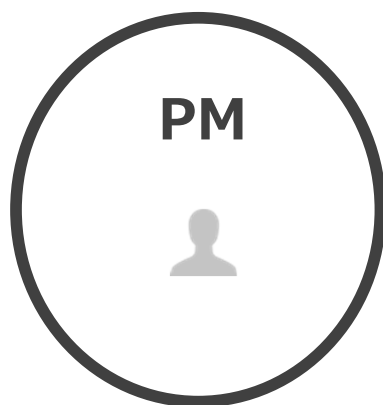
2 発表の前提

- ・ テスト自動化ツールは、ノーコードで自動化ができる**MagicPod**を使用
- ・ ツールやTipsを紹介する内容ではありません
- ・ テスト自動化初心者の方も理解できるような抽象度が高い内容です
- ・ 「ほぼ100%自動化した」とは98%で、2%はアニメーションの動作確認などで自動テストでは行えなかったものです
- ・ E2Eレベルで回帰テストを自動化した話です

2 発表の前提

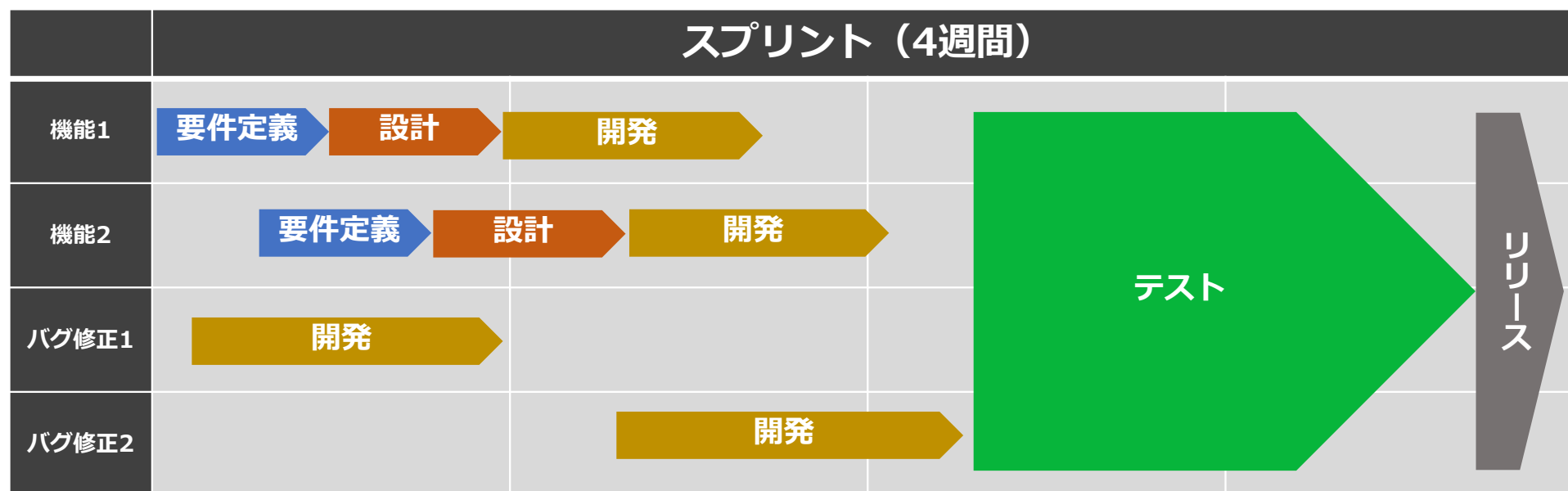
- ・ 弊社でMagicPodを用いたテスト自動化の**スモールスタート**として開始した社内WEBアプリの回帰テスト自動化の話です
- ・ 2020~2021までの話で、自動テスト完成の半年後に別担当者に引き継がれ運用されています
- ・ 同じことをすれば同じ成果になるとは限りません

3 回帰テストをほぼ100%自動化した世界 体制



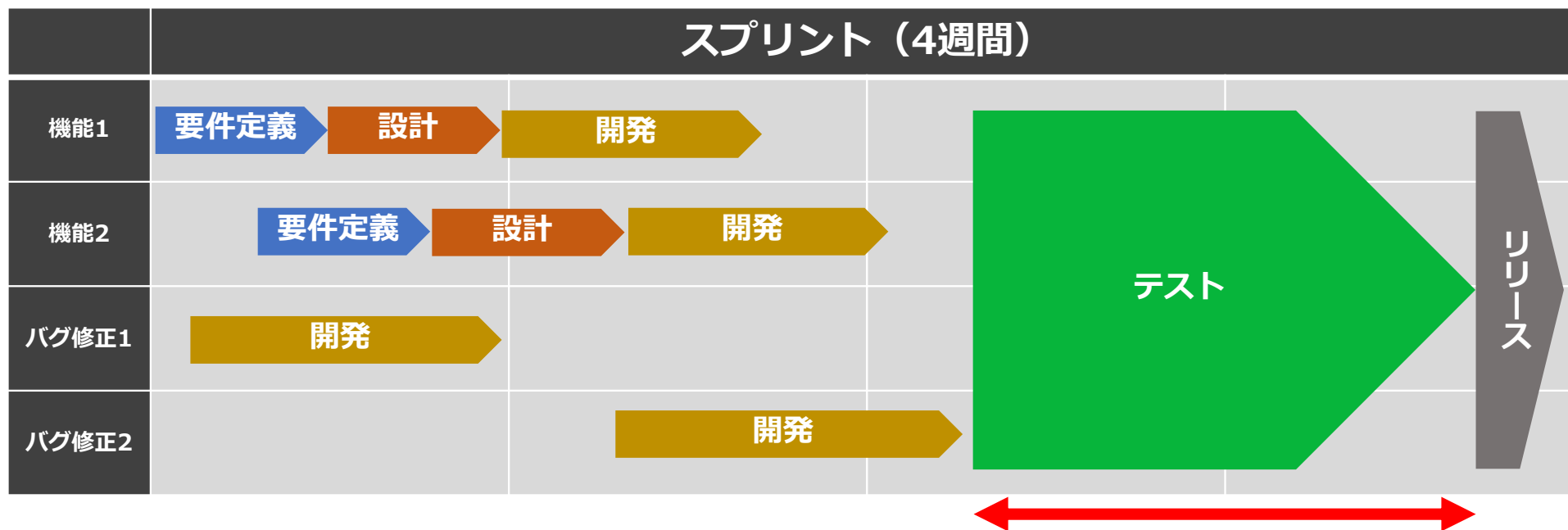
3 回帰テストをほぼ100%自動化した世界

開発サイクル（テスト自動化前）



3 回帰テストをほぼ100%自動化した世界

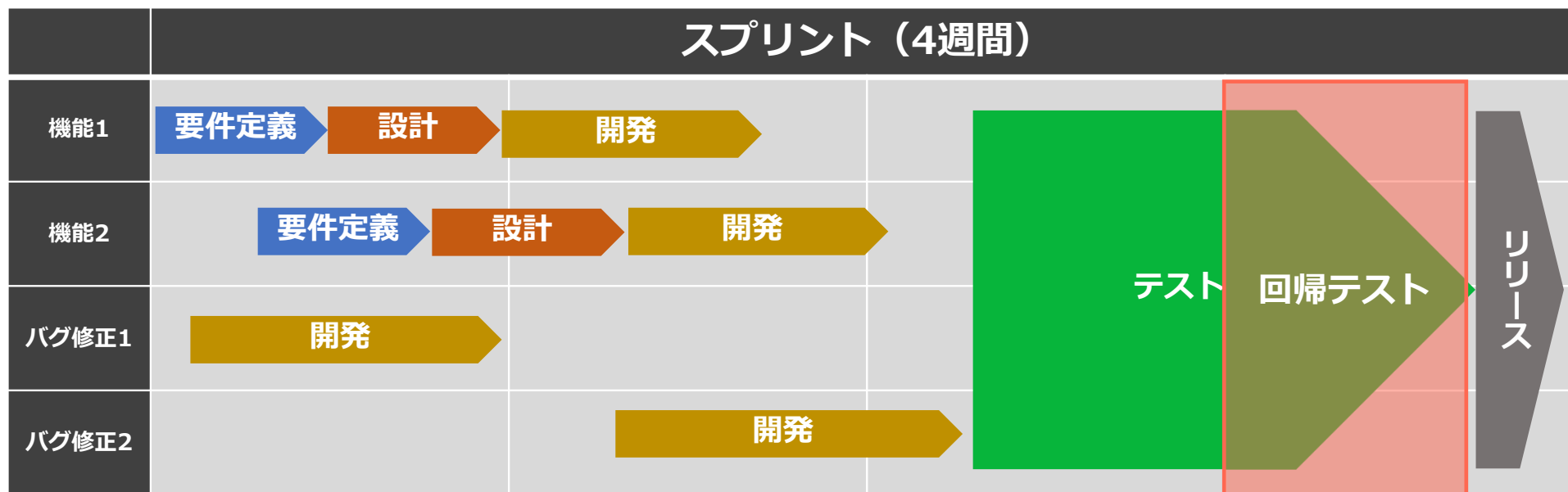
開発サイクル（テスト自動化前）



スプリント終盤でテストを実施

3 回帰テストをほぼ100%自動化した世界

開発サイクル（テスト自動化前）

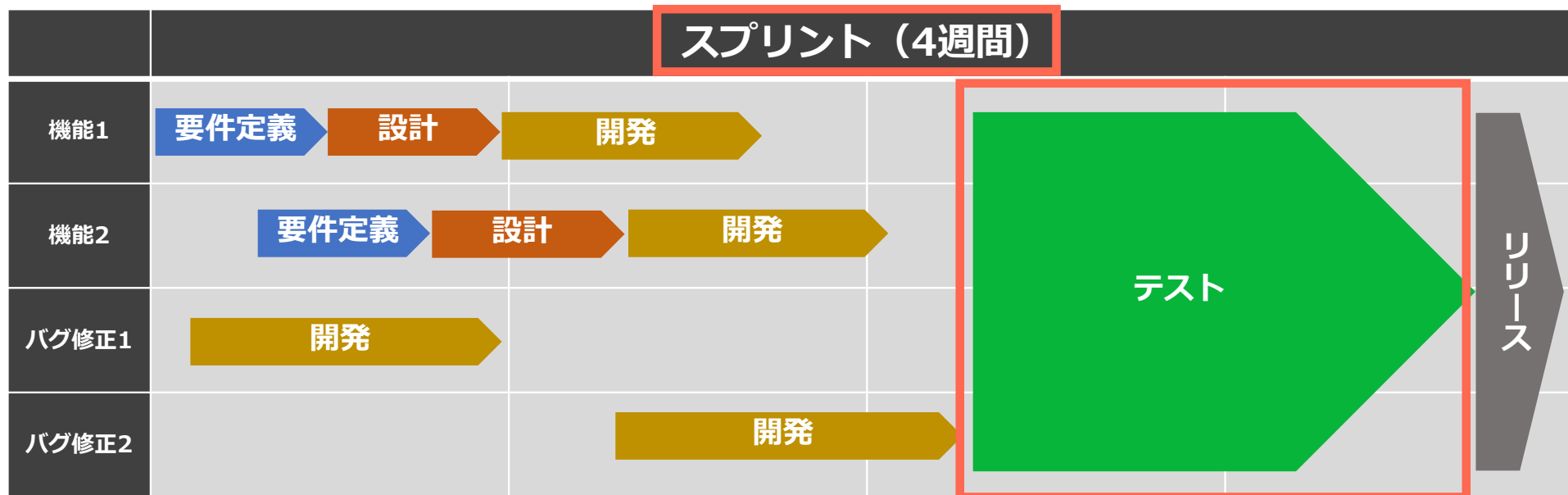


テスト工程の40%を占めており、手動実行に32時間かかっていた

3 回帰テストをほぼ100%自動化した世界

開発サイクル（テスト自動化前）

開発後すぐにユーザから
フィードバックが欲しい

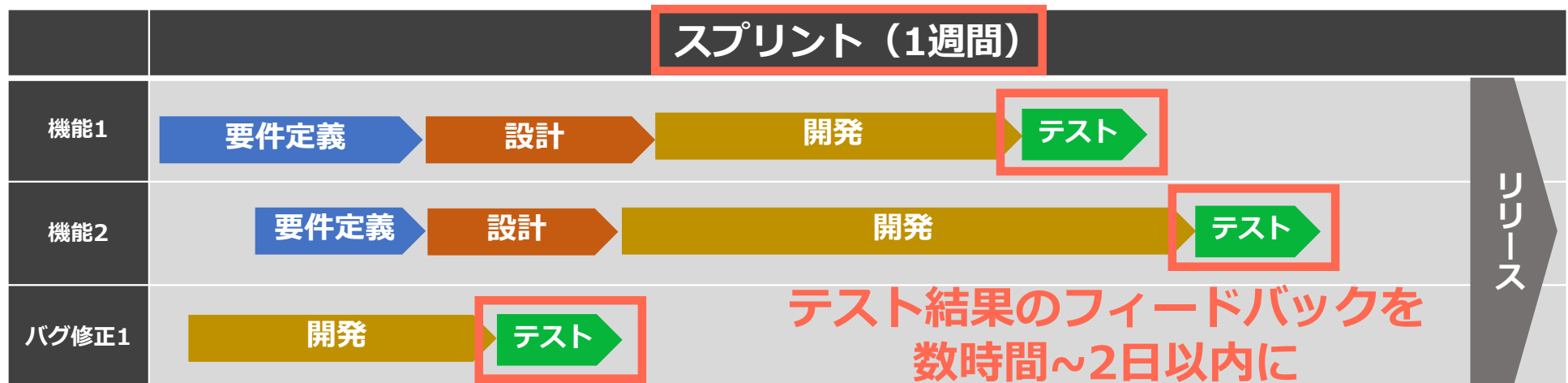


テスト結果のフィードバックが遅い

3 回帰テストをほぼ100%自動化した世界

テスト自動化の目的

週1リリース

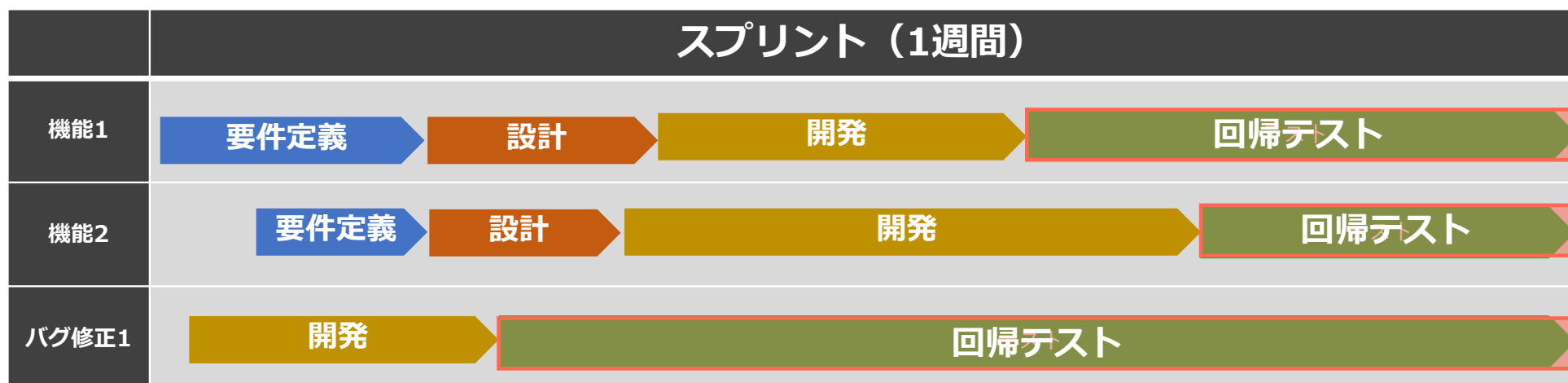


テスト結果のフィードバックを短縮化し

リリース頻度を上げる

3 回帰テストをほぼ100%自動化した世界

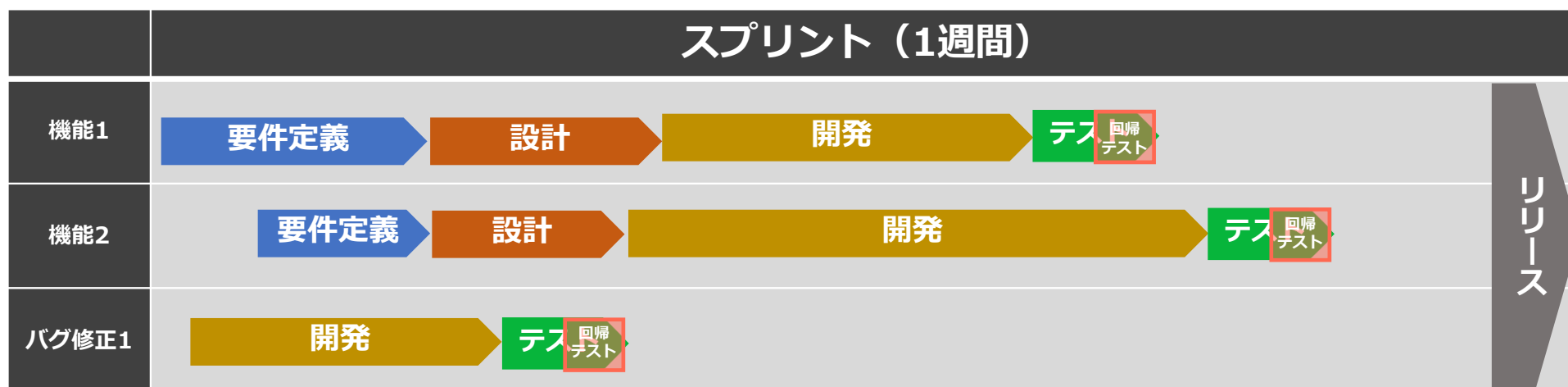
テスト自動化の目的



回帰テストが目的達成のボトルネックになっていた

3 回帰テストをほぼ100%自動化した世界

テスト自動化の目的



回帰テストの実行時間を短縮する必要があり
できる限り100%自動化が必要だった

3 回帰テストをほぼ100%自動化した世界

目的達成までにやったこと

1	自動化ツールのトライアル
2	作業の無駄をなくす
3	実行環境の設計・構築（開発者と一緒に）
4	自動化するテストの優先度を定める
5	手動のテストケースを自動テスト用に書き換える
6	テスト自動化
7	週1リリースの開発プロセス検討

3 回帰テストをほぼ100%自動化した世界

目的達成までにやったこと-自動化ツールのトライアル

弊社では既に**MagicPod**と契約済みだったため
MagicPodを対象にトライアル

目的達成
できそうか

誰でも
自動化できるか

CI 連携
できるか

ツールの
サポートは迅速か

3 回帰テストをほぼ100%自動化した世界

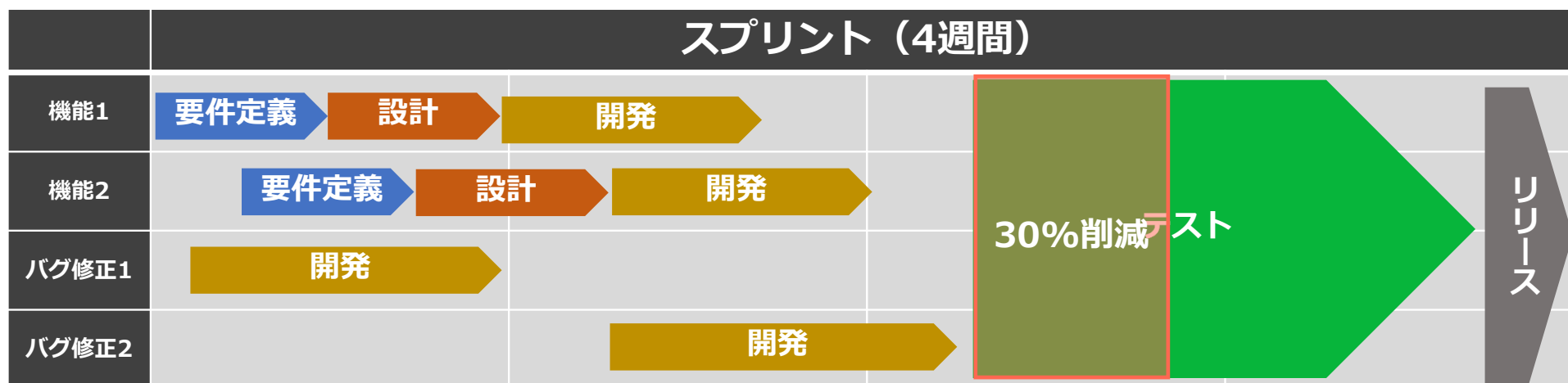
目的達成までにやったこと

1	自動化ツールのトライアル
2	作業の無駄をなくす
3	実行環境の設計・構築（開発者と一緒に）
4	自動化するテストの優先度を定める
5	手動のテストケースを自動テスト用に書き換える
6	テスト自動化
7	週1リリースの開発プロセス検討

3 回帰テストをほぼ100%自動化した世界

目的達成までにやったこと-作業の無駄をなくす

テスト自動化の時間を作るために
現状の無駄をなくす



目的が無い・優先度の低いテスト

3 回帰テストをほぼ100%自動化した世界

目的達成までにやったこと

1	自動化ツールのトライアル
2	作業の無駄をなくす
3	実行環境の設計・構築（開発者と一緒に）
4	自動化するテストの優先度を定める
5	手動のテストケースを自動テスト用に書き換える
6	テスト自動化
7	週1リリースの開発プロセス検討

3 回帰テストをほぼ100%自動化した世界

目的達成までにやったこと-実行環境の設計・構築（開発者と一緒に）

自動テストではユーザ操作やバッチ操作など
自動テスト以外の影響を受けない環境でテストしたかった

E2E環境

テスト自動化するための環境

E2E-CI環境

CIにより自動テストを実行する環境

3 回帰テストをほぼ100%自動化した世界

目的達成までにやったこと

1	自動化ツールのトライアル
2	作業の無駄をなくす
3	実行環境の設計・構築（開発者と一緒に）
4	自動化するテストの優先度を定める
5	手動のテストケースを自動テスト用書き換える
6	テスト自動化
7	週1リリースの開発プロセス検討

3 回帰テストをほぼ100%自動化した世界

目的達成までにやったこと

1	自動化ツールのトライアル
2	作業の無駄をなくす
3	実行環境の設計・構築（開発者と一緒に）
4	自動化するテストの優先度を定める
5	手動のテストケースを自動テスト用書き換える
6	テスト自動化
7	週1リリースの開発プロセス検討

3 回帰テストをほぼ100%自動化した世界

目的達成までにやったこと-手動のテストケースを自動テスト用に書き換える

手動テストケースは**テスト結果が曖昧なものが多かった**

例) 「ホームへ」ボタンをタップ

手動テスト	・ <u>ホーム画面が表示されていること</u> どうやって機械に表示されることを判断させる？
自動テスト	・ URLが「 https://〇〇/home 」であること

テスト結果が曖昧だと乖離が生まれる可能性がある

3 回帰テストをほぼ100%自動化した世界

目的達成までにやったこと

1	自動化ツールのトライアル
2	作業の無駄をなくす
3	実行環境の設計・構築（開発者と一緒に）
4	自動化するテストの優先度を定める
5	手動のテストケースを自動テスト用に書き換える
6	テスト自動化
7	週1リリースの開発プロセス検討

3 回帰テストをほぼ100%自動化した世界

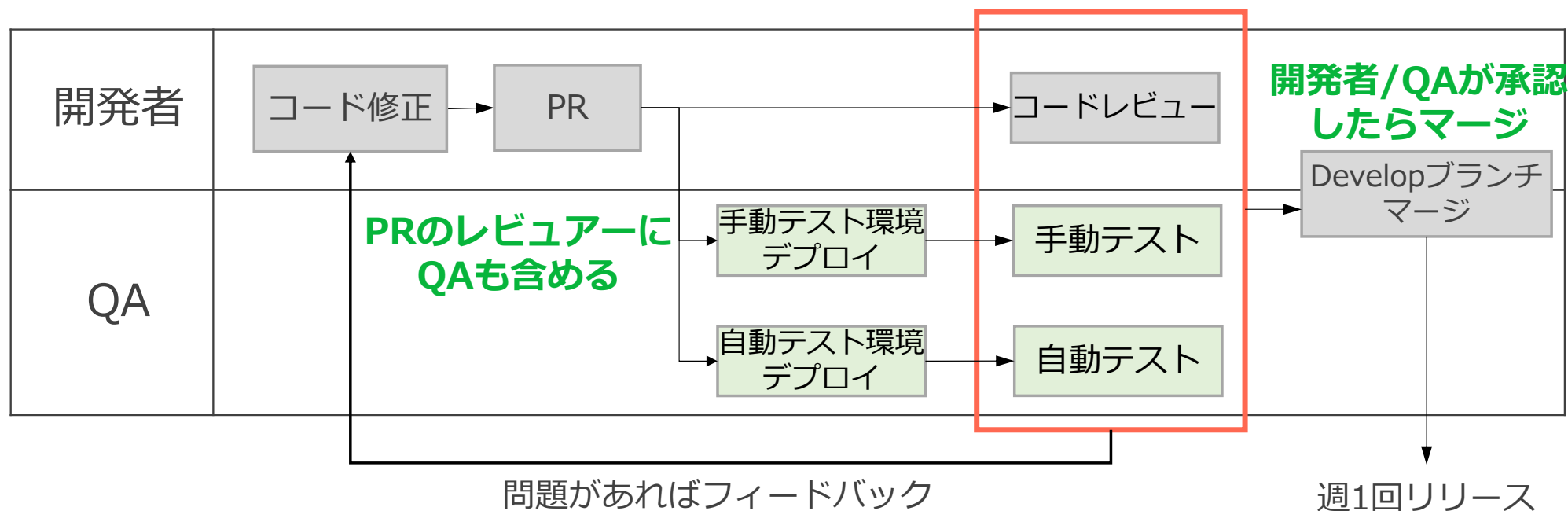
目的達成までにやったこと

1	自動化ツールのトライアル
2	作業の無駄をなくす
3	実行環境の設計・構築（開発者と一緒に）
4	自動化するテストの優先度を定める
5	手動のテストケースを自動テスト用に書き換える
6	テスト自動化
7	週1リリースの開発プロセス検討

3 回帰テストをほぼ100%自動化した世界

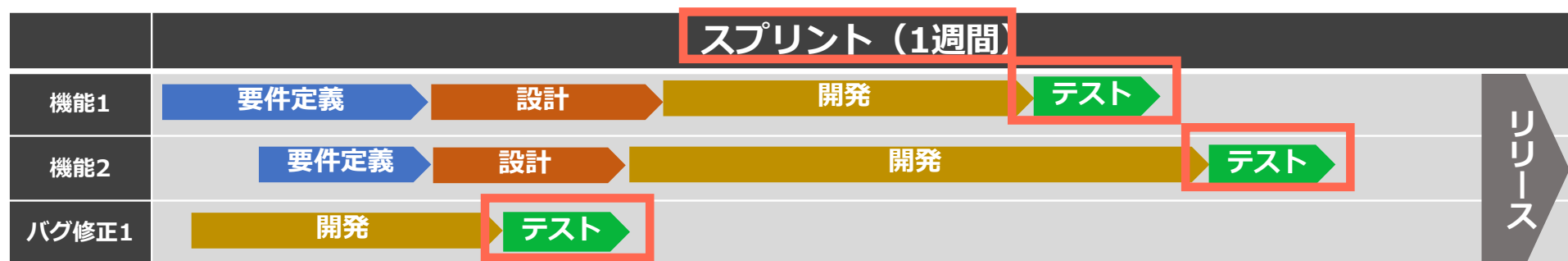
目的達成までにやったこと-週1リリースの開発プロセス検討

※PRとは、ソースコードなどを追加・修正した際、反映を自分以外の人に依頼する機能



回帰テストをほぼ100%自動化した世界

3 回帰テストをほぼ100%自動化した世界



成果	Before	After
リリース頻度	月1回	月4回(400%増加)
回帰テストの実行時間	32時間	2時間(94%削減)
テスト結果のフィードバック	1週間~3週間	2時間~2日(87%~95%削減)

テスト自動化の**効果**はこれだけではなかった

3 回帰テストをほぼ100%自動化した世界

目的達成 + 副次効果

半年間自動テストを運用し

開発者から自動テストの信頼を得れた

開発者からのコメント

- E2E が強力。feature をテストしすぐ開発者に返せるので、とてもいい。ローカルで試せると更に良い

Product・Organizationの新規作成後に、真っ白の画面まさに私の直した場所でした・・・心強い・・・！（編集済み）

あまりの的確さにちょっと怖くて軽く引いてます！！

成果	After
全体のPR数	150%に増加
リファクタのPR数	170%に増加
本番障害数	0件(半年間)

3 回帰テストをほぼ100%自動化した世界

成果-全体のPR数:150%増加

テスト結果フィードバックが短縮されたこと

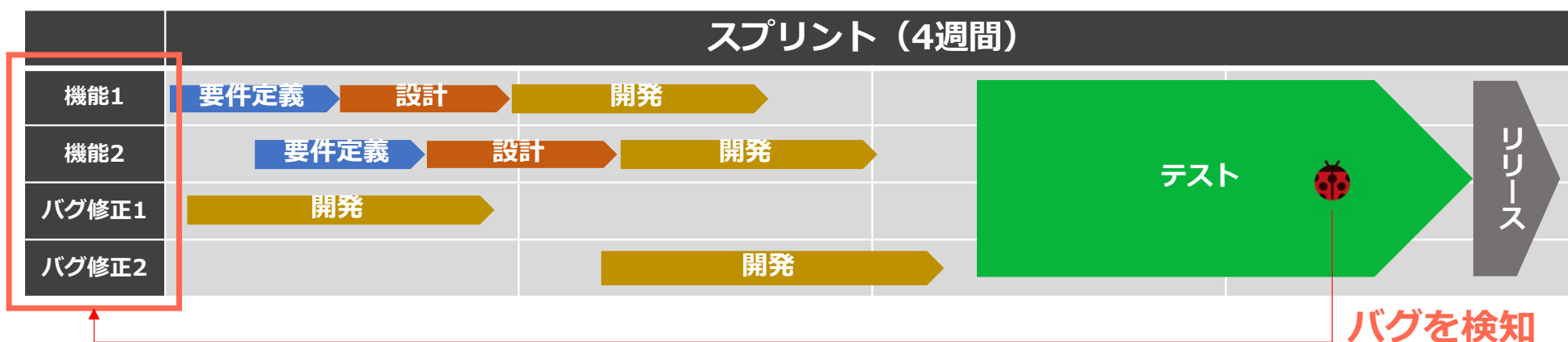
PR単位でテストを実行することで

開発速度にも影響

3 回帰テストをほぼ100%自動化した世界

成果-全体のPR数:150%増加

今まで



開発者



どの修正によるバグか調査しよう

開発者Aさんが修正した、バグ修正1が原因みたいだ

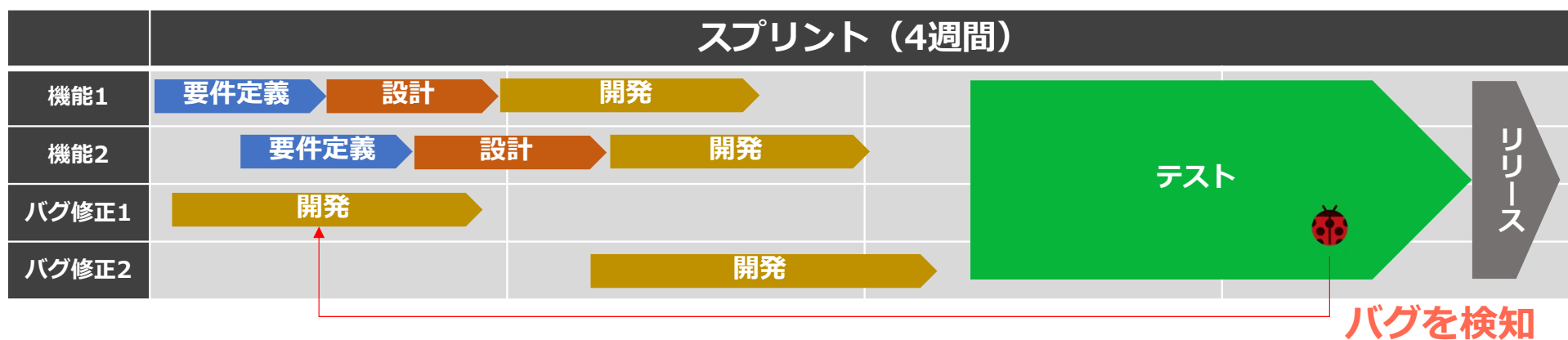
開発者Aさんに連絡しよう

バグ分析/別開発者とのコミュニケーションが必要だった

3 回帰テストをほぼ100%自動化した世界

成果-全体のPR数:150%増加

今まで



開発者



3週間前の修正内容なんてもう忘れちゃったよ…

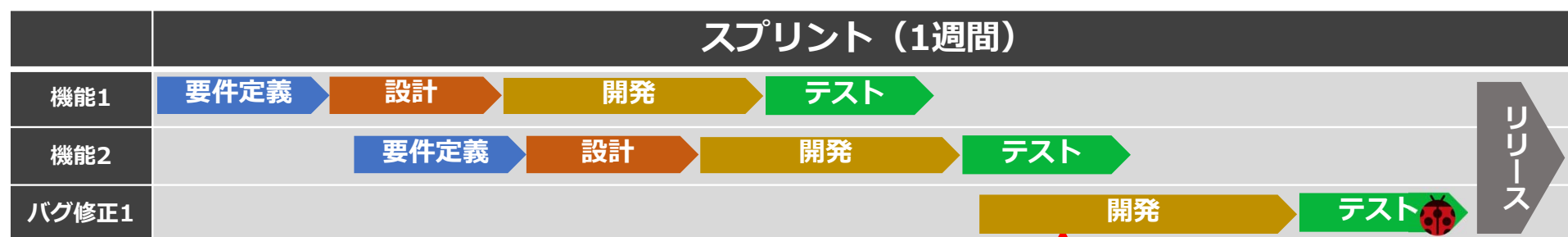
修正内容を思い出すところからスタートだ

本来不要な作業を行っており、**バグ修正に時間を要していた**

3 回帰テストをほぼ100%自動化した世界

成果-全体のPR数:150%増加

自動テスト運用後



開発者



この修正によるバグならこう修正すればよさそう

バグを検知

今まで発生していた**本来不要の作業を削減**

3 回帰テストをほぼ100%自動化した世界

成果-リファクタのPR数:170%増加

※リファクタリングとは、

外部から見たときのソフトウェアの機能や振る舞いを変えずに、ソフトウェアの内部構造を変えること

回帰テストが自動化されたことによって

気兼ねなくいつでもリファクタリングが行えるように

開発者



- テスターの負荷状況を気にしなくて良くなった
- テスト結果のフィードバックが早く、ほぼ全部テストしてくれるのでありがたい
- 壊してしまうかもしれない不安がなくなったため、積極的にチャレンジできる

3 回帰テストをほぼ100%自動化した世界

成果-本番障害数:0件

回帰テストを実施していた工数を
別作業に割り当てることができ本番障害発生抑制につながった

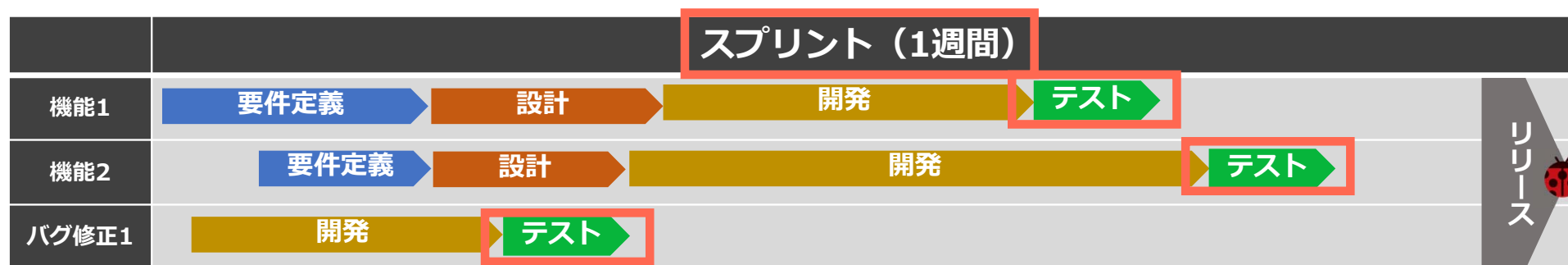
自動テスト
結果分析
メンテナンス

工数の都合上
検証できなかったこと

改善活動

3 回帰テストをほぼ100%自動化した世界

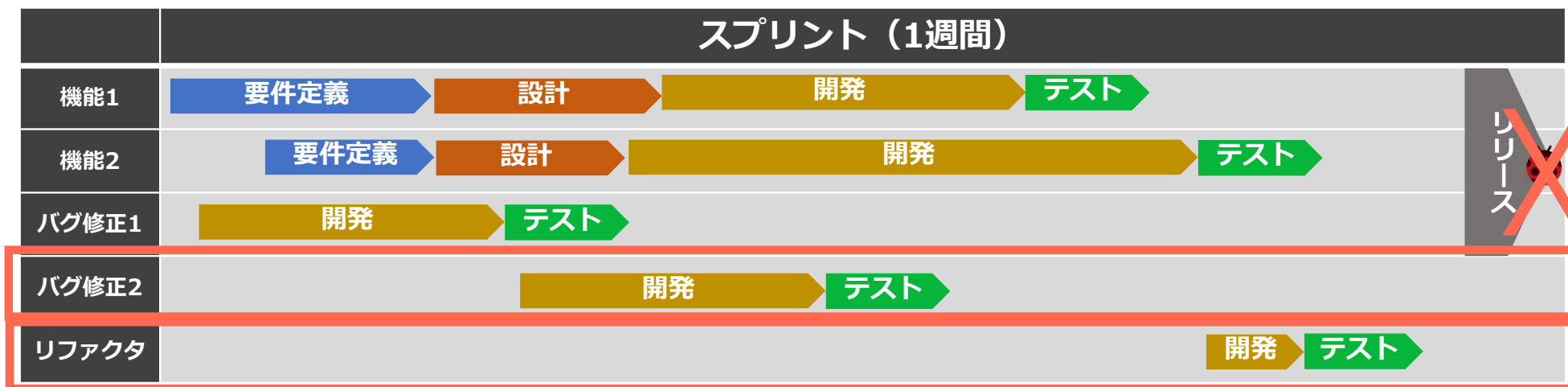
目的達成



テスト結果のフィードバックを短縮化し
リリース頻度を上げることができた

3 回帰テストをほぼ100%自動化した世界

目的達成 + 副次効果



開発体験の向上・本番障害の抑制
にもつなげることができた

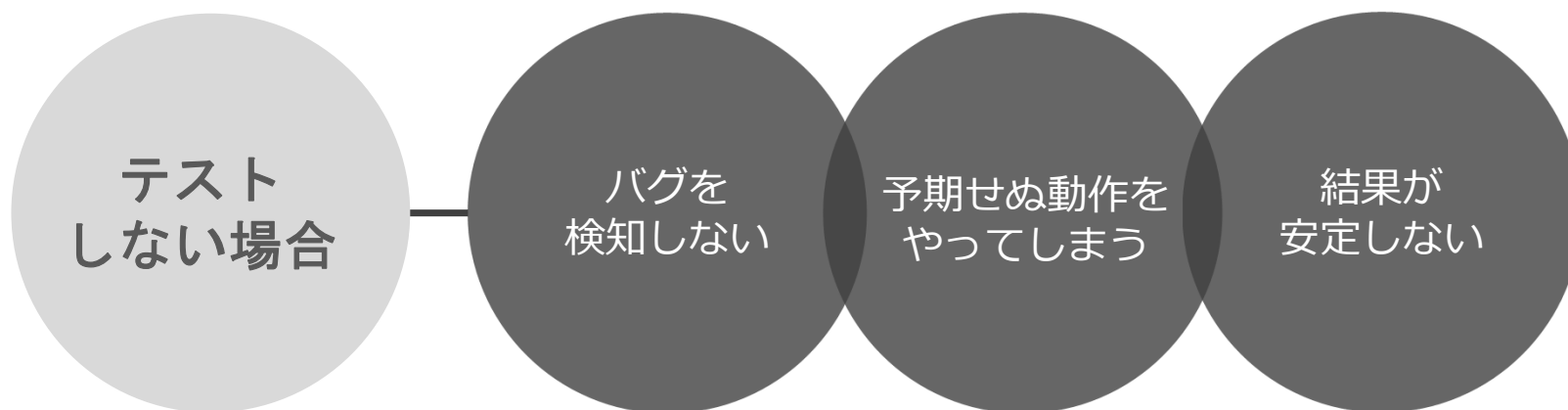
3 回帰テストをほぼ100%自動化した世界



みんなをHAPPYにすることができた

4 テスト自動化初心者にとってほしいポイント

自動テストのテストを絶対にすること！



5 結論

いつやるの？ 回帰テスト自動化 = **今でしょ**

最大のリスクはリスクを取らないことだ！

THANK YOU