Link and Motivation Group

開発生産性のために、ユニットテストについて考える

株式会社リンクアンドモチベーション 代慶 真

2023/2/10

© Link and Motivation Group

自己紹介

代慶 真（よけい まこと）

株式会社リンクアンドモチベーションQA

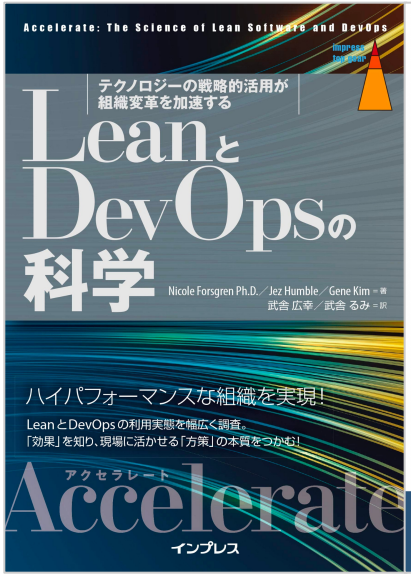
- **モチベーションクラウドの開発でバックエンドエンジニアとして2年間経験**
- 昨年よりQAエンジニアとして、主に自動テストの保守・運用を担当する

© Link and Motivation Group

これまでの取り組み

開発生産性

four keysの開発メトリクスを用いて、開発生産性の可視化と向上をおこなってきた



© Link and Motivation Group

これまでの取り組み

4 Key Metrics	Low	Low-mid	True-mid	High-mid	**High**
デプロイ頻度	1ヶ月に1回	1回/月~1回/週	1週間に2~4回	1日に1回	**オンデマンド 1日に何度でも**
リードタイム	1ヶ月以上	1ヶ月~7日	4-6日	1-3日	**1日以内**
MTTR	1週間以上	5-7日	2-4日	1-2日	**24時間以内**
変更失敗率	46%以上	31-45%	21-30%	16-20%	**15%以下**

© Link and Motivation Group

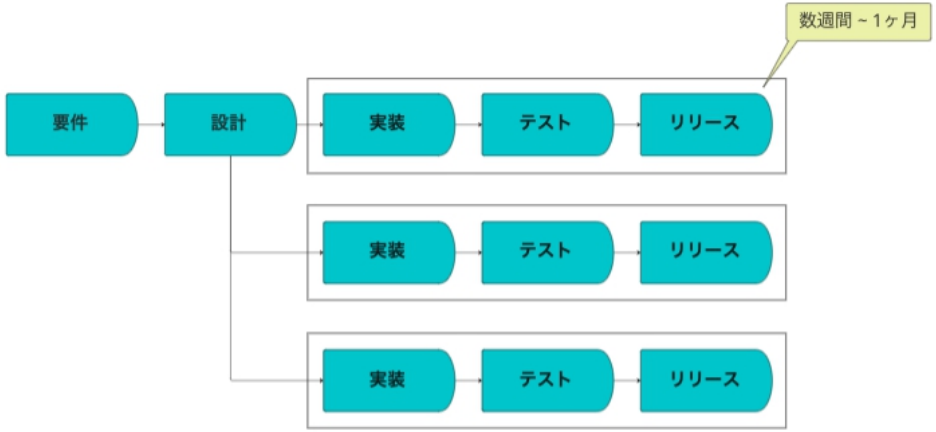
これまでの取り組み

Highレベルを基準においた時に課題が見えてきた

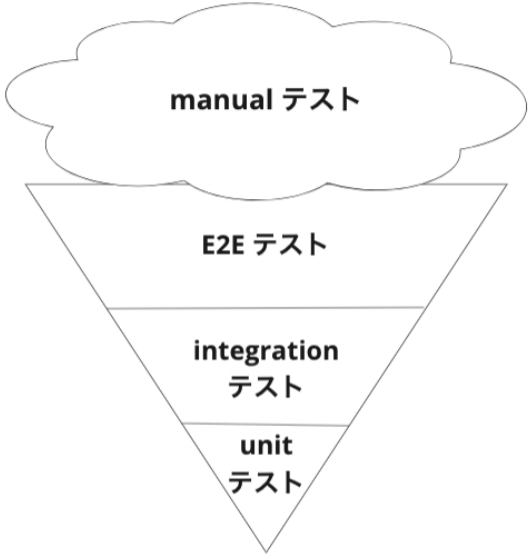
© Link and Motivation Group

現状の課題

開発プロセスの現状



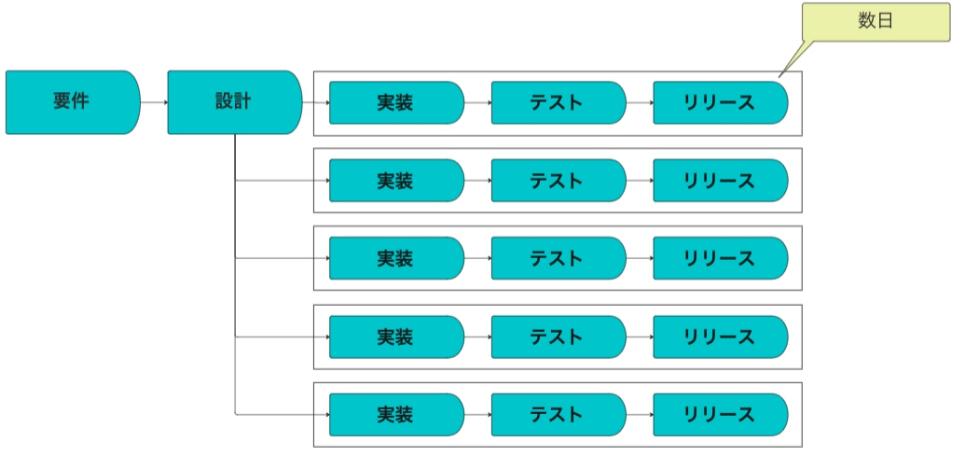
テスト戦略の現状



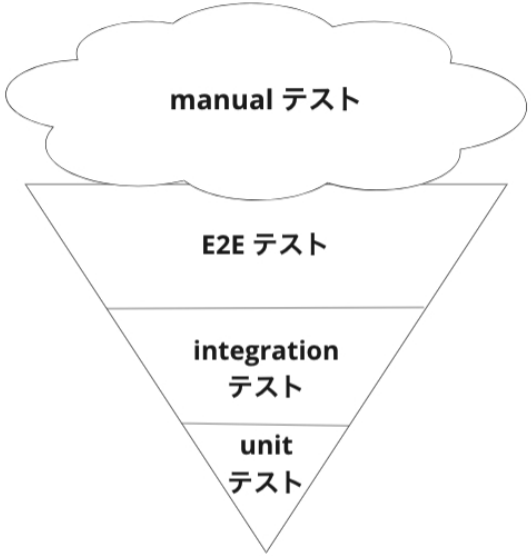
© Link and Motivation Group

現状の課題

目指すべき開発プロセス



テスト戦略の現状

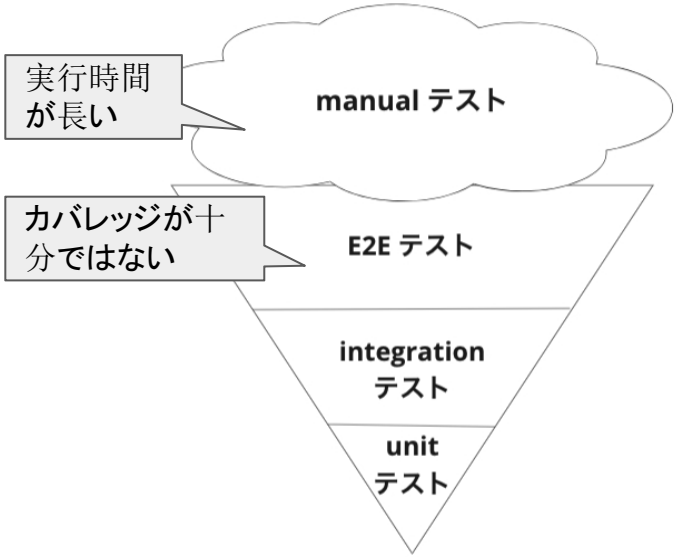


© Link and Motivation Group

現状の課題

目指すべき開発プロセス

テスト戦略の現状



© Link and Motivation Group

現状の課題

テストフェーズが原因で、
質とスピードがトレードオフの状態になる

© Link and Motivation Group

現状の課題

- 品質を優先すると、、、

実装は完了しているのにリリースをすることができない
結果的に**デプロイ頻度・リードタイム**が落ちてしまう

- スピードを優先すると、、、

新規機能のテストや、既存機能のリグレッションテストが十分に行えず、
インシデントが起こってしまう
結果的に変更失敗率が上がってしまう

© Link and Motivation Group

ユニットテストを拡充する

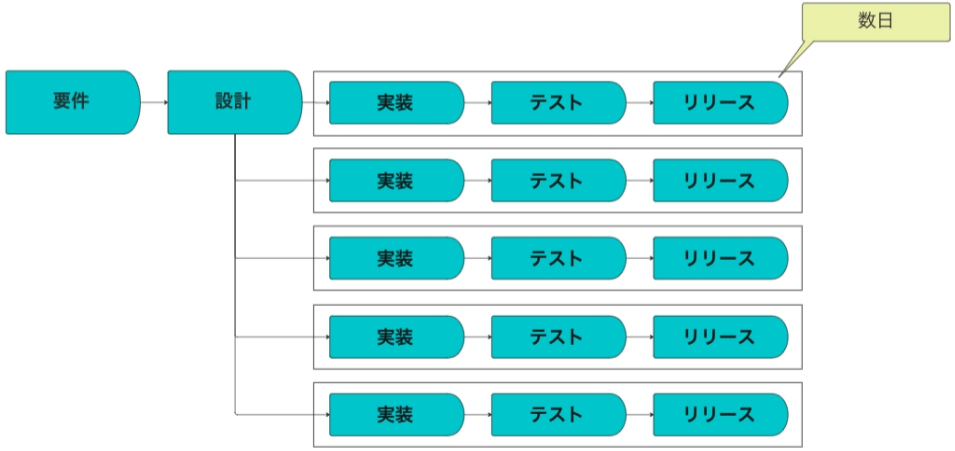
- 新規機能のテスト・・・一部の手動テストをユニットテストで品質担保することで、手動テストの工数を減らせる

- 既存機能のリグレッションテスト・・・ユニットテストで仕様カバレッジを十分に満たすことができれば、E2Eも含めた自動テストで十分に品質担保することができる

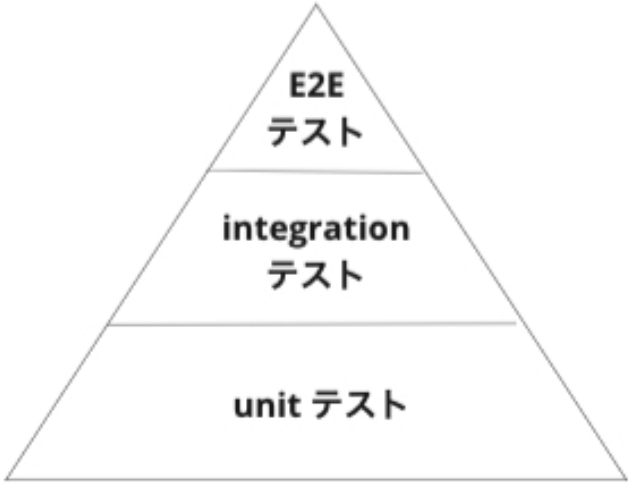
© Link and Motivation Group

打ち手

目指すべき開発プロセス



目指すべきテスト戦略



© Link and Motivation Group

ユニットテストの現状

実は、、、

© Link and Motivation Group

実は、、、

コードカバレッジ（C0）は90%以上を維持してる

© Link and Motivation Group

実は、、、

コードカバレッジ（C0）は90%以上を維持してる

© Link and Motivation Group

ユニットテストの現状

起こっていること

ユニットテストはコードカバレッジを満たすように書かれているが、
同じ箇所を手動テストでも確認している

なぜ起こっているか

自分たちが書いたユニットテストを自分たちが信頼できていない

© Link and Motivation Group

ユニットテストの現状

ユニットテストの価値を高めていく必要がある

© Link and Motivation Group

ユニットテストの価値が高い状態

メンテしやすく、信頼できて、フィードバックが早いユニットテストの実現
＝保守性、信頼性、性能効率性の両立

背景

- 質とスピードの同時実現には、安心して改修できるコードが必要
- 組織がスケールしてもコードのブラックボックス化を予防し、継続的なリファクタリングを可能にする必要がある

© Link and Motivation Group

ユニットテストの価値の可視化

Unit Test Metrics 現在値 / 目標値	2023/1時点	Low	Low-Mid	True-Mid	High-Mid	High
テスト実行時間（CI）	平均値：17:24 中央値：16:35 ※一部skipあり	~30min	~20min	~10min	~7min	~5min
テスト実行時間（localで遅いrspecファイル目安100件の実行時間）	20min以上	20min	16min	12min	8min	4min
テスト成功率(flakiness)	80%前後	80%未満	80%〜	90%〜	95%〜	99%〜
信頼できるテスト比率	なし	20%未満	20%	40%	60%	80%
定性指標改善				可読性改善	UT・プロダクションコードの保守性改善 開発チームからの信頼性向上	

© Link and Motivation Group

まとめと今後の展望

まとめ

- 開発生産性向上のために、ユニットテストの価値向上を行うことを決めた
- ユニットテストの現在地を可視化した

今後の展望

- QAチームとして、ユニットテストのリファクタリングを推進する
- 自動テストの拡充を行うことで、質とスピードを兼ね備えたデリバリーを実現する

© Link and Motivation Group