

オレ流「身の丈」 開発プロセス

～テスト（最後の砦）で固めよう～
株式会社アジェンダ 小楠貴紀

リソースが足りない。オフショアでやろう。とりあえずやってみないとわからないよ。リスクの低いラボがいい。Ajaxいいね流行ってるし。やったことないけど。全部で130画面ぐらいかな。まだ途中だけど。とりあえずできたところから始めよう。設計書のフォーマットは変えよう。やってみると技術的に難しそう。思ったより工数掛かるけど**人数増やせば**2ヶ月あればいけるっしょ？受け入れの**テストは1ヶ月**ぐらいあれば足りるかな？**バグが収束しない**？なんで？仕様がおかしいの？また**テストやり直し**？仕様変更が収束しない？また**デグレ**？

あと2ヶ月あれば終わる？いつ終わるかって誰に聞けばわかるの？何が悪いかわかる？プロジェクト立て直ししよう。スケジュール作りなおそう。仕様も見なおそう。**テストは全部やり直し**だ。なんとか終わった。え？次は基幹システム？設計どうする？どこで開発する？**単体テスト**どうする？テスト設計やろう。どうしてそんなに**時間**かかるの？意味あるの？**バグ残**ってるから意味ないよ。効率よく...

壁にぶち当たってま～す

- 開発規模が急に大きくなった。
- 今までのやり方が通用しない。
- 「チーム開発」の経験不足。
- 手戻りが致命傷になる。

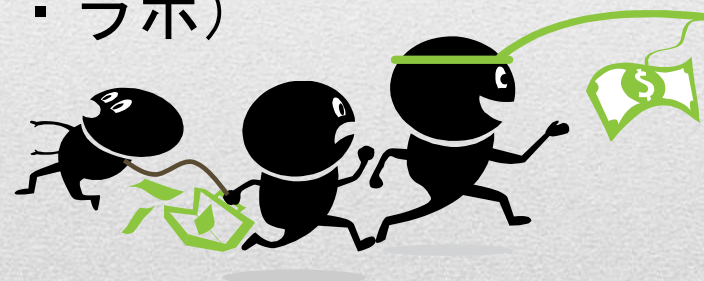


**気づいたら、すでに行き
詰まっていた。**

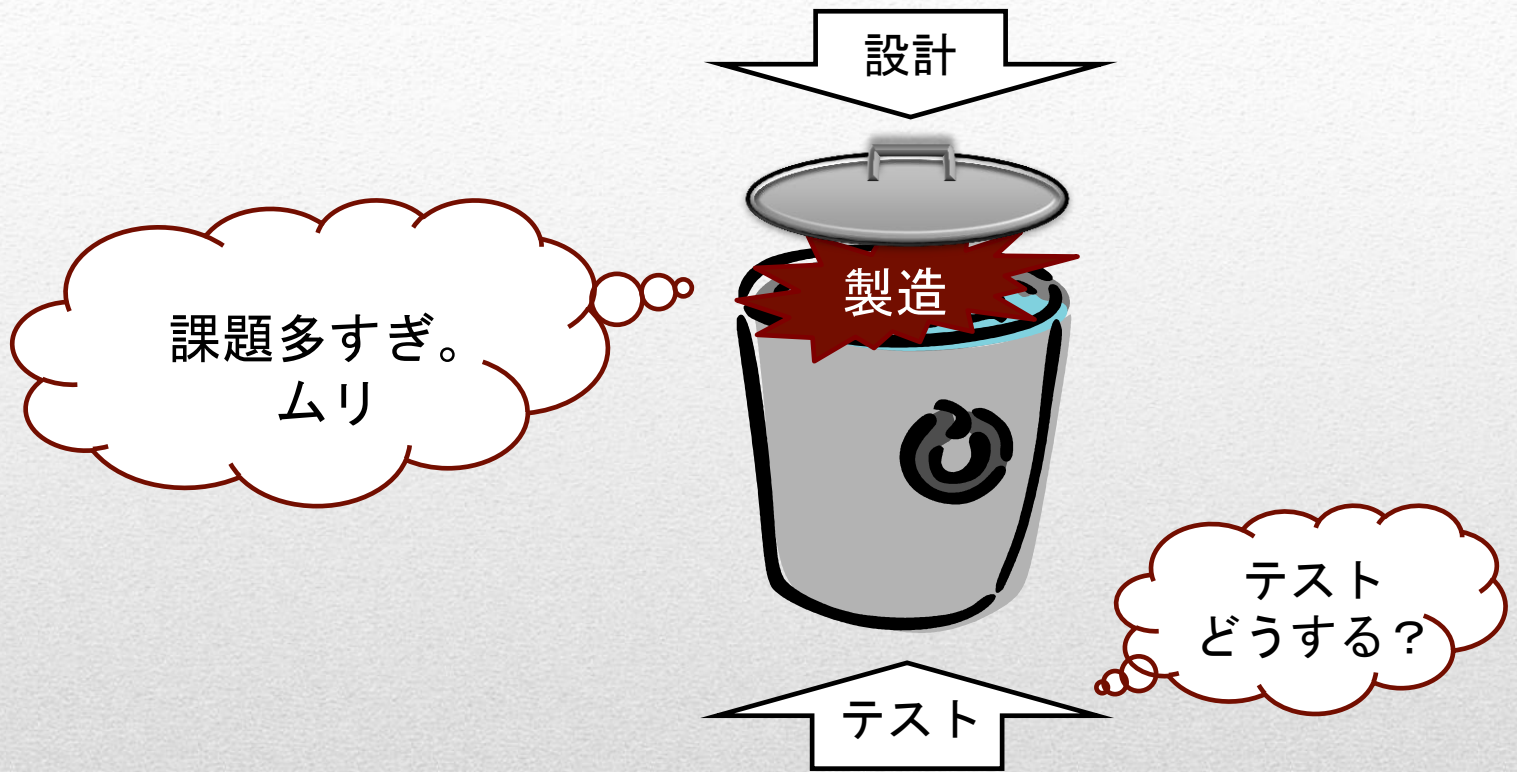
- 問題が大きすぎ
- 解決するノウハウがない
- そもそも、今まずい状態にあるとは思いたくない

**原因はいろいろ。本来は
分析するんだろうが・・・**

- 例：新プロジェクト
 - 業務系基幹システムの自社サービス、1年以上の開発規模
- チーム・メンバー
 - 設計・・・社内
 - 製造・・・社外（オフショア・ラボ）
 - テスト・・・社内

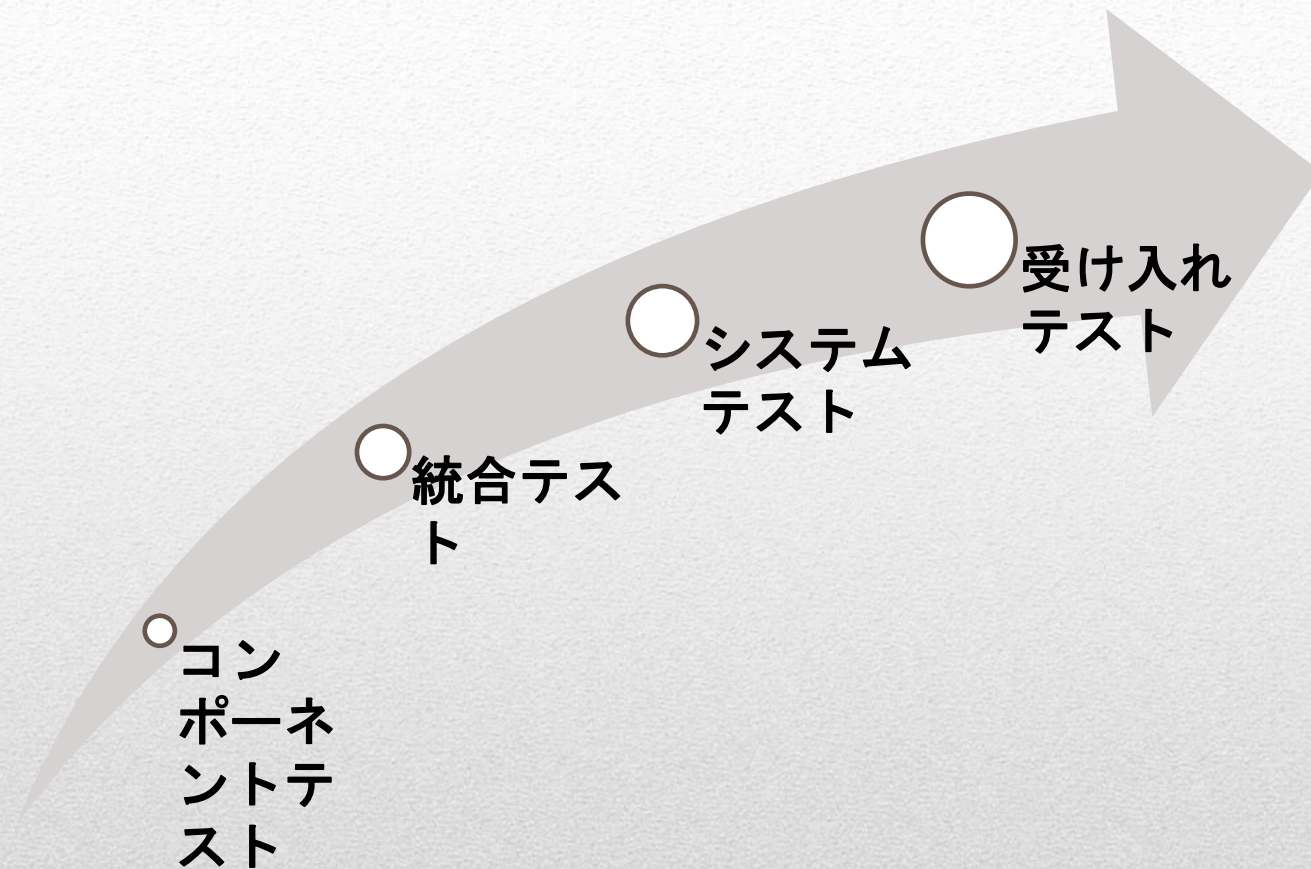


ゆっくり考えている時間など、ない。

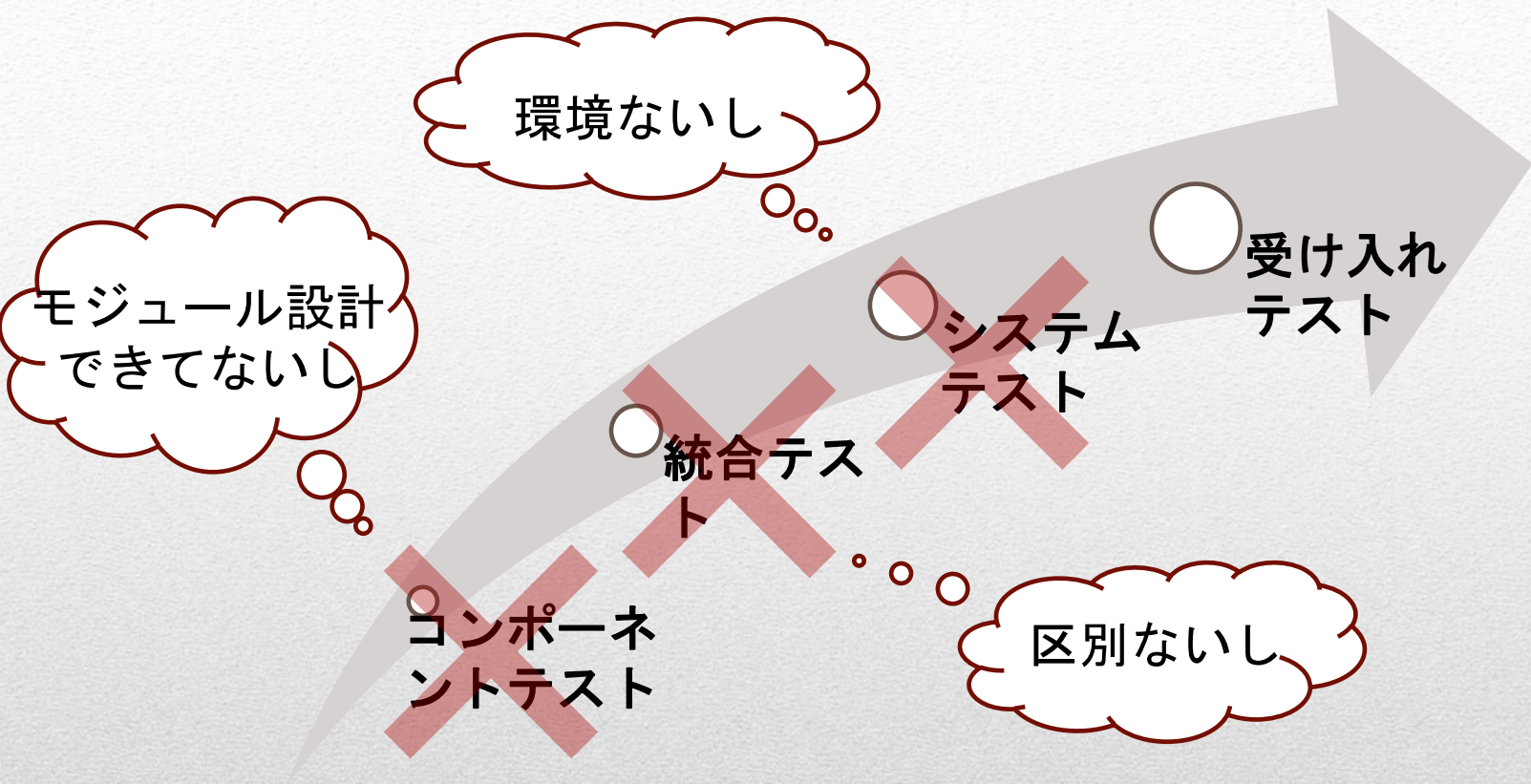


さくせん

→○○には蓋をしろ



よく見るテストの流れ。



環境ないし

モジュール設計
できてないし

受け入れ
テスト

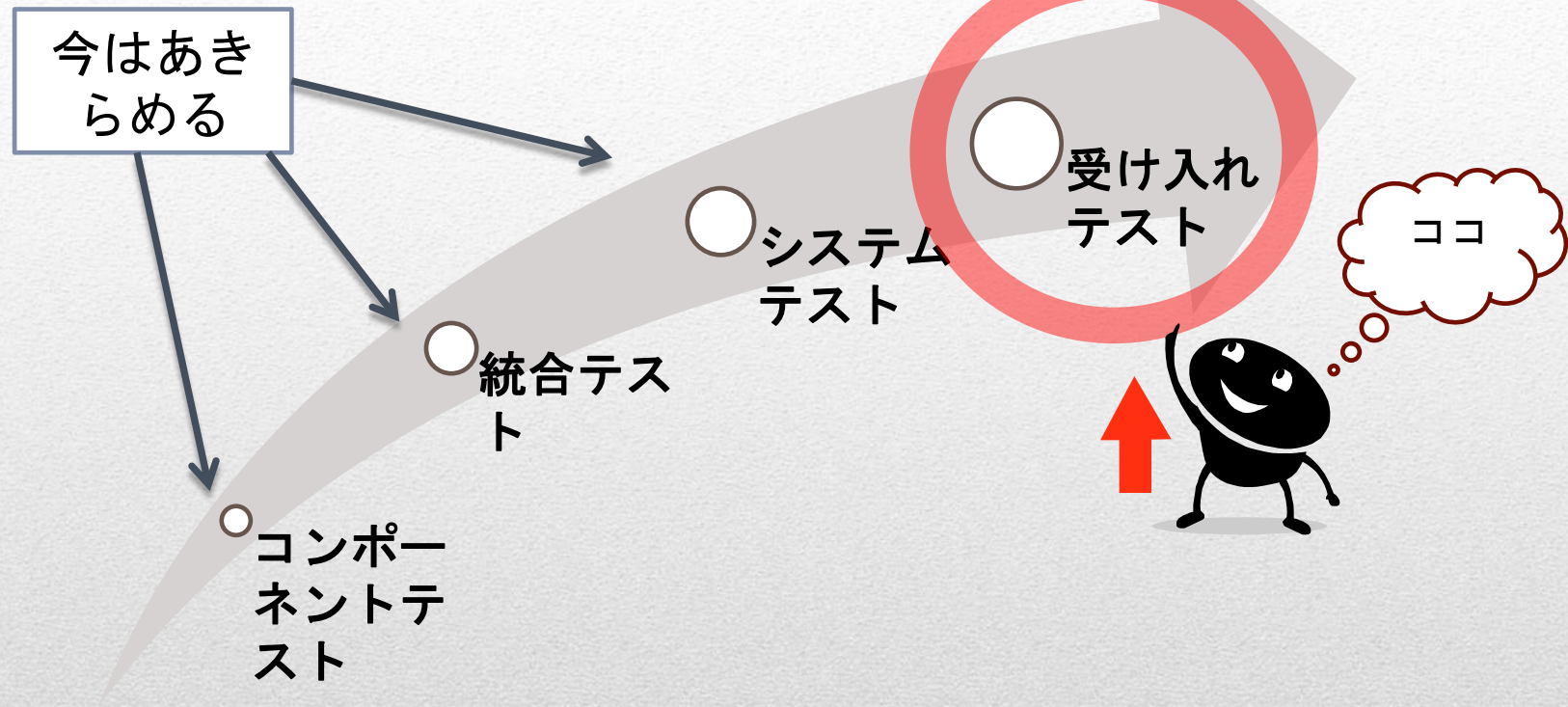
システム
テスト

統合テス
ト

コンポーネ
ントテスト

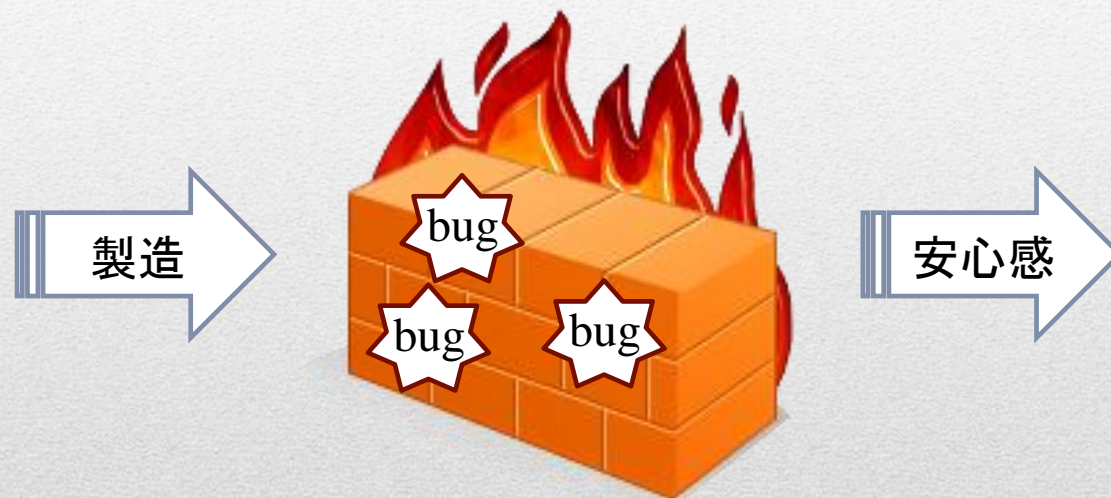
区別ないし

**実際の現場では、
ほとんど、できていない。**



**受け入れテスト＝最後の
砦に手を入れてみた。**

全機能を網羅するテストを、
必ず通してからリリース。



→ 「最後の砦」を作る

改善① 受け入れテスト

- レビューを増やす。
×量や時間 ○種類・タイミング
- できるだけ早い段階でレビュー。
基本設計→テスト設計
- 開発側のテストケースのレビュー
- やるよって言うだけで、引き締まる。



改善② レビュー

伝える

ユーザーの困っていること、重要な機能を、開発者にもちゃんと伝える。



聞く

開発者自身に説明させる。

ちゃんと理解していないと、説明できないからすぐに見つかる。
とってもお手軽で効果の高いテスト方法です。おすすめ！



改善③ コミュニケーション

できるだけ早く、
リスクを「知りたい」



その分、早く動ける

気づくと、集約してきた

テストの前に、

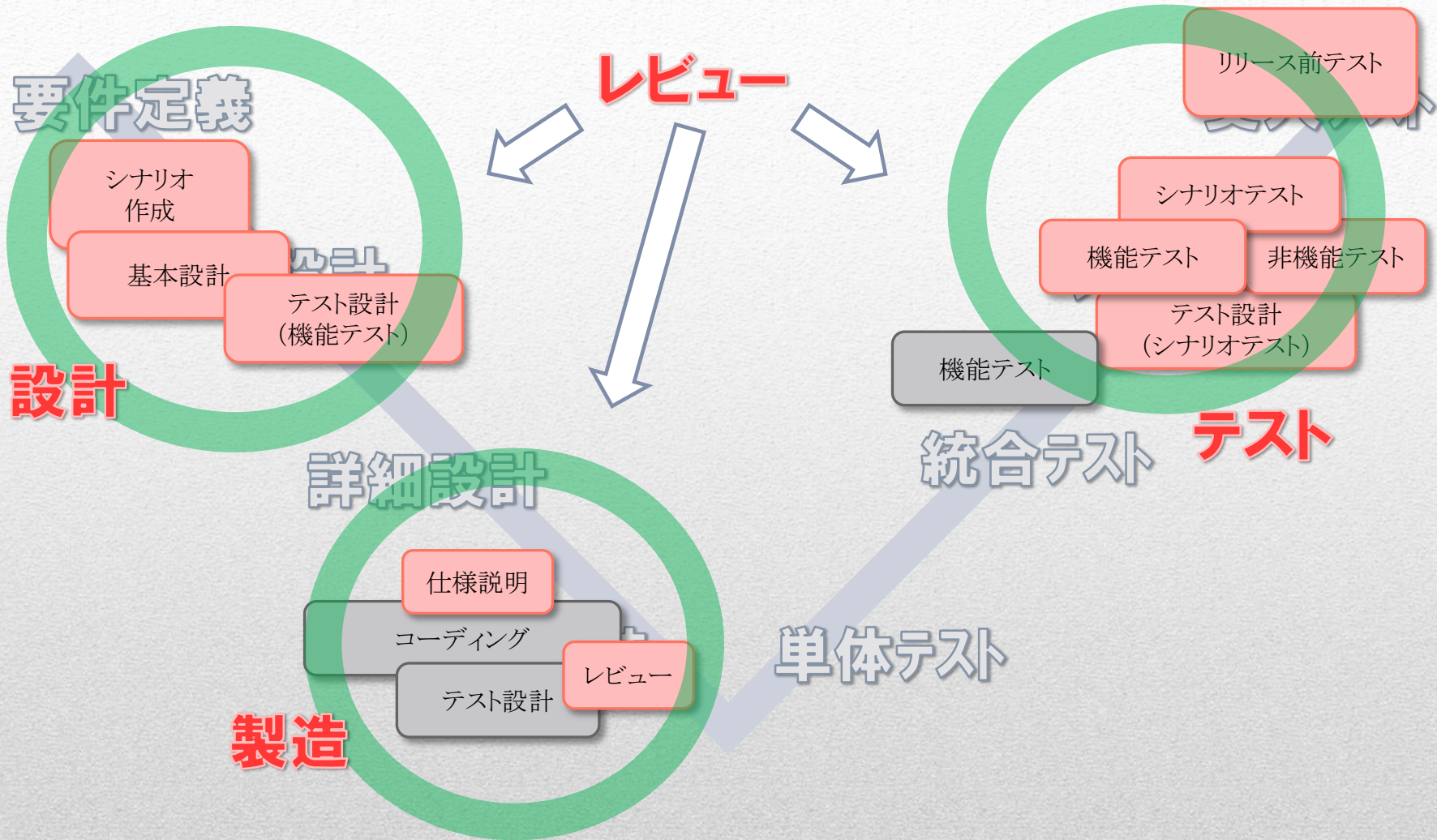
テスト工数が見積もれるようになった。

どんな品質で出てきそうか見える。

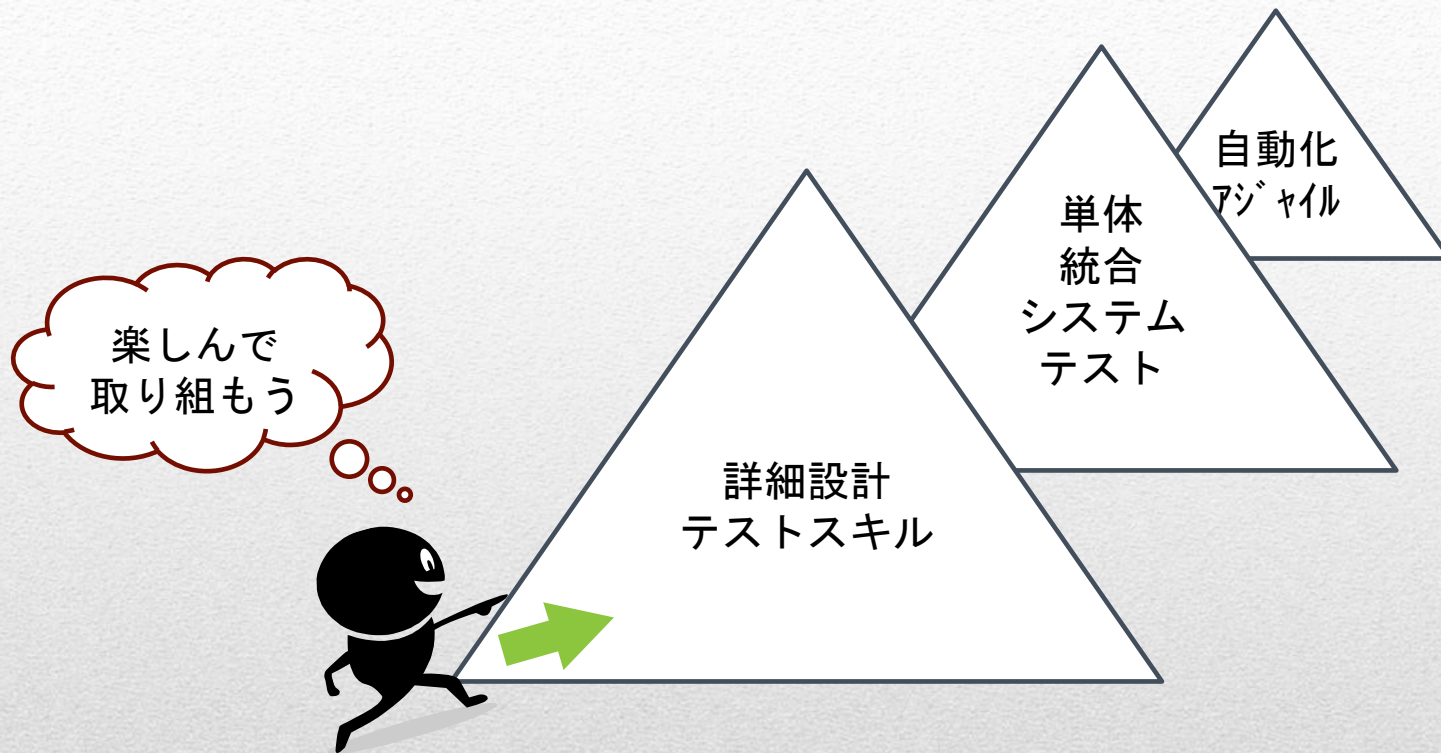
テストの印象

苦しい→ (少し) 楽しい

こんな効果が。



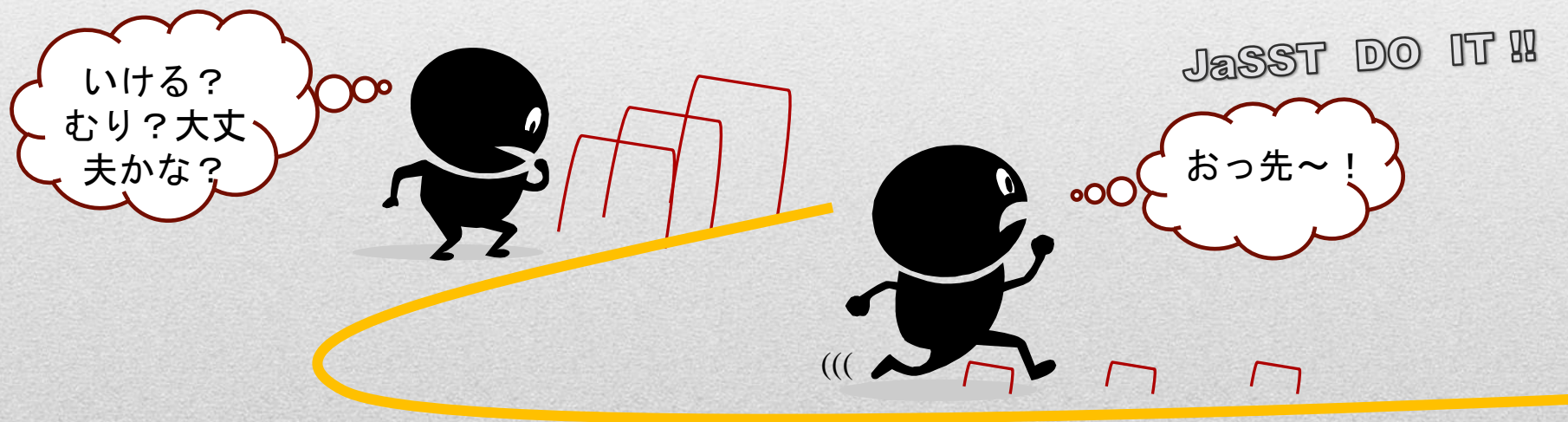
V字上だと・・・まだ穴だらけ



したっけ、次の手を打つ。

悩んでいたら、テストから始めよう！

ただし、まともにやると上流から下流まで巻き込んで大掛かりになってしまうので、受入テストやレビューなど、**できるところから**手をつけて、徐々に改善の輪を大きくしていく
→これが俺の（現場の）流儀です。



まとめ

な~んも、やってみりゃ
いいべさ！

JaSST DO IT !!