

テストティング (Software Testing)

第1回講義
ソフトウェア・テスト概論



目次

- ソフトウェア・テストिंगの基礎
 - テストとは何か
 - テストでは何をすればいいか
- ソフトウェア・テストिंगの課題
- ソフトウェア・テストिंगのプロセス
- ソフトウェア・テストिंगの技法



(1) ソフトウェア・テストिंग の基礎

質問

- 清涼飲料の自動販売機のテストをすることになりました。
どういうテストを行ったらいいですか。

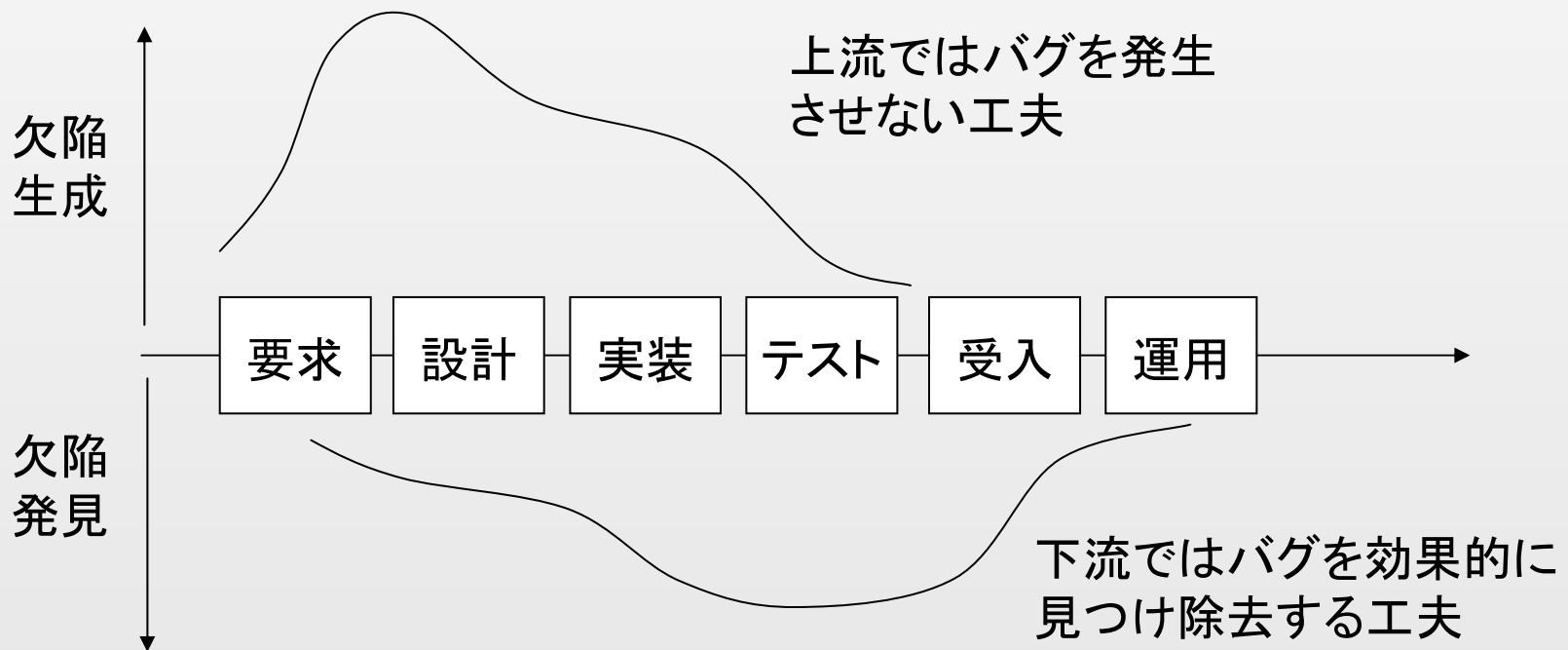
(ハードウェアも含めたシステム
のテストとして考えてください)





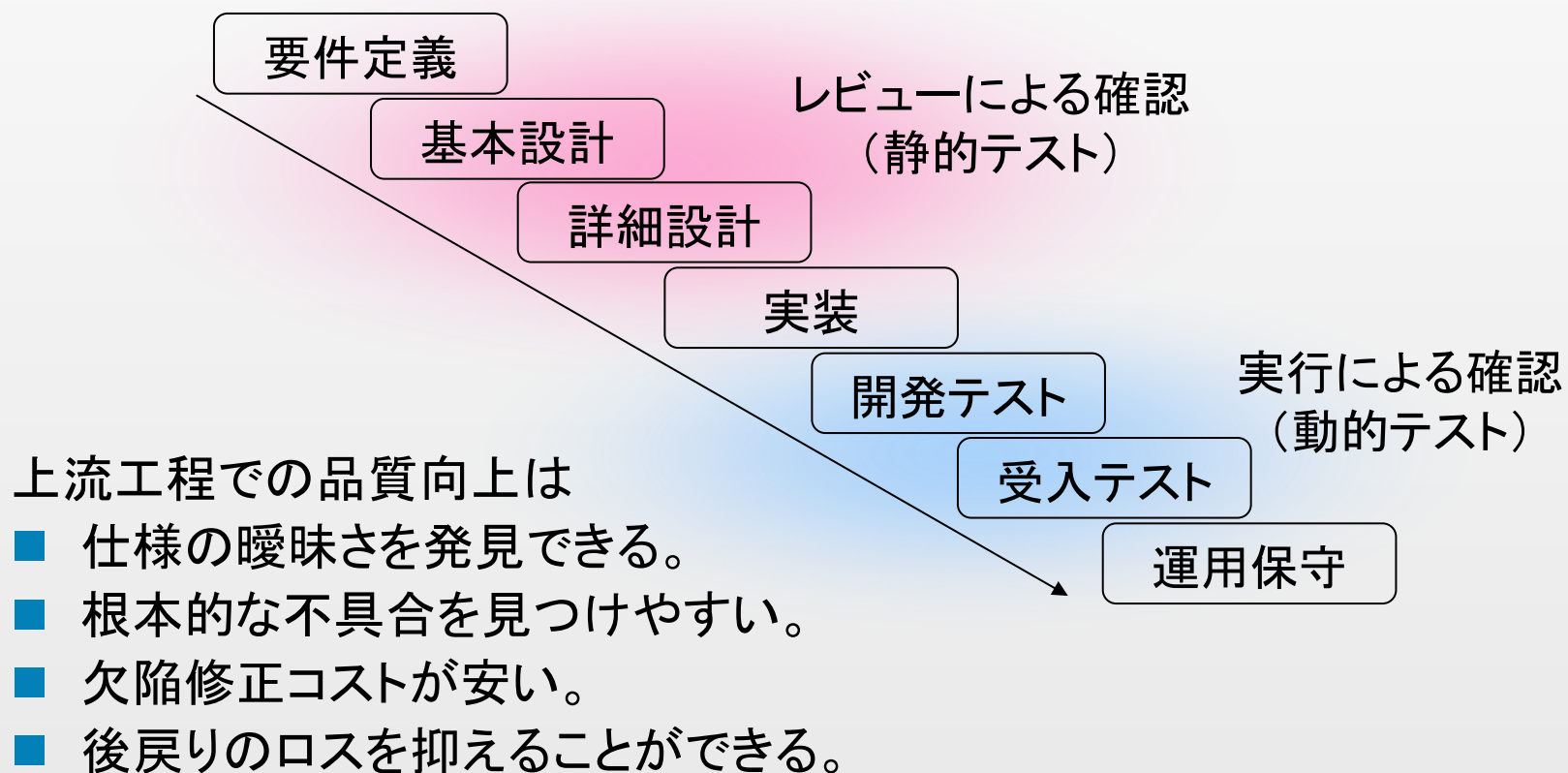
テストとは何か

- テストとは、ソフトウェア開発ライフサイクルにおいて混入した**欠陥を発見**する行為である。



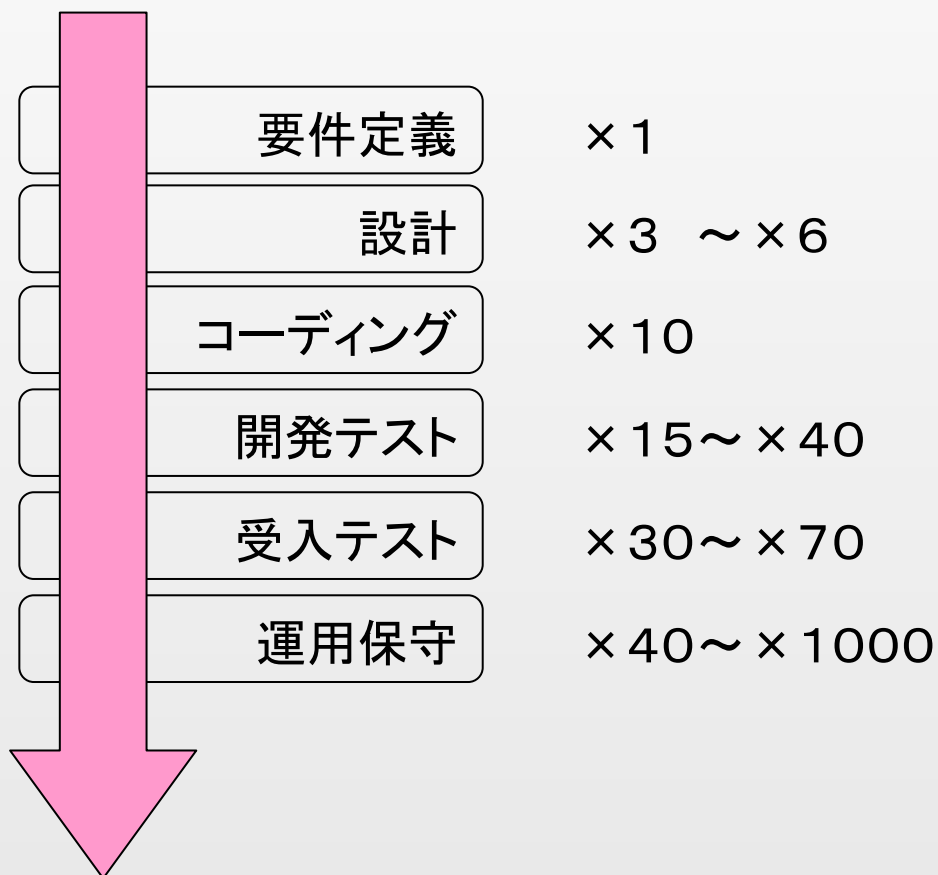


上流工程からの品質向上



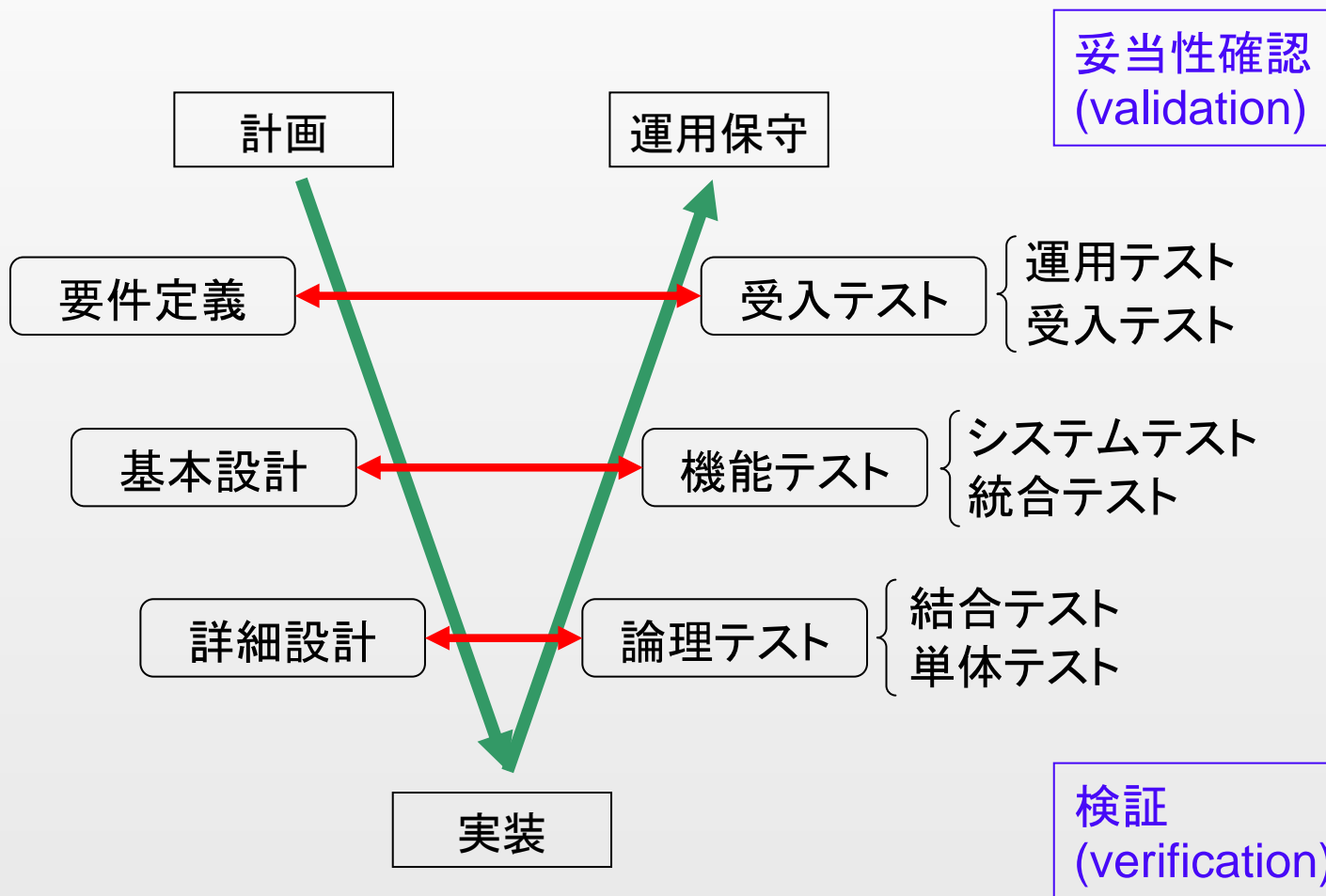


欠陥修正コスト



出典 : Boehm (1981)

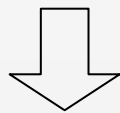
V字モデル



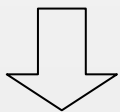


テストは何故必要か

- テストは品質を確認する行為である。



- 品質を確認しておかないと、プロダクトリスクが分からない。



- 品質事故が発生して、多大な損失を起こす。

品質を確認しておけば、リスクが許容範囲にあるかどうか事前に分かる。



ソフトウェアテストの原則

- すべての組合せをテストすることは不可能。
- 将来の実行環境は想定できないこともある。
- ↓
- 完璧なテストはあり得ない。
 - 『テストはバグが存在することは示すが、バグが存在しないことを決して示さない。』(E. W. Dijkstra)
- ↓
- **顧客が求めている重要な要求**は何か。
- 限られた時間でどのテストを優先して実施すればいいか。



ソフトウェアテストの原則

- すべての組合せをテストすることは不可能。
 - 将来の実行環境は想定できないこともある。
- ↓
- 完璧なテストはあり得ない。
 - 『テストはバグが存在することは示すが、バグが存在しないことを決して示さない。』(E. W. Dijkstra)
- ↓
- 顧客が求めている重要な要求は何か。
 - 限られた時間で**どのテストを優先して**実施すればいいか。



リスクを考慮したテスト

- 顧客にとって影響の大きい(リスクの高い)機能からテストする。
 - テスト容易性(実施コスト)を加味することもあるが、基本は顧客の視点で考える。
- FMEA(故障モード影響解析)を利用する。

$$\text{発生頻度} \times \text{影響度} \rightarrow \text{優先度}$$

足切りラインを設定する。

※FMEA: Failure Mode and Effect Analysis



FMEAの一例

■ ATMの例

リスク	発生度	影響度	優先度	
引き出し	3	3	9	} 綿密に実施
預け入れ	2	3	6	
セキュリティ	2	3	6	} 適度に実施
振り込み	2	2	4	
ローン返済	1	2	2	} 粗く実施
残高確認	1	2	2	
パフォーマンス	1	2	2	

発生度
3:高
2:中
1:低

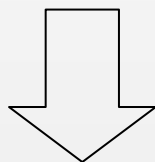
影響度
3:大
2:中
1:小

発生度 × 影響度

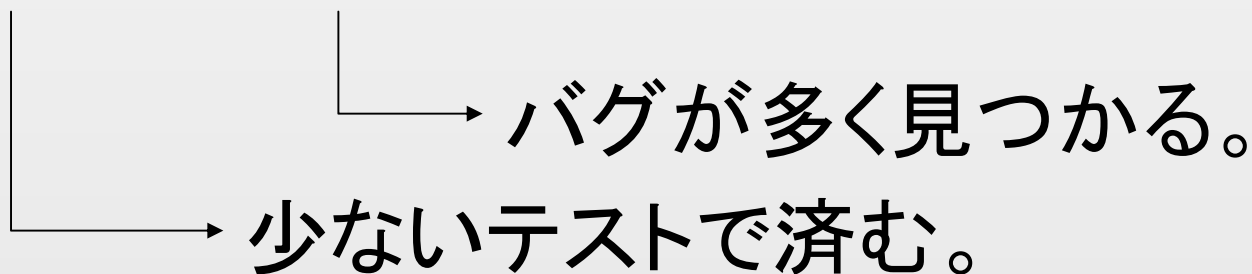


効率的で効果的なテスト

- テストすべきことを考え出したら、きりが無い。
- すべてをテストすることはできない。



- 効率的で効果的なテストが必要。





(2) ソフトウェア・テストニング の課題



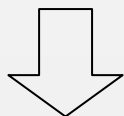
グループ討論

- 自己紹介
- ソフトウェア・テストの現状
 - どのようにテストを進めているか
- ソフトウェア・テストの課題
 - 何が難しいのか
- グループのまとめ
- 発表

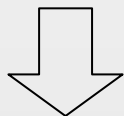


テスト現場の課題

- 大規模複雑かつ短納期
- コストダウン
- 顧客ニーズの変化への追従



- テスト工程にしわ寄せがくる。
- できる範囲でテストをする。



- **正しい**テストを**正確に・最速に**実行

原因

現象

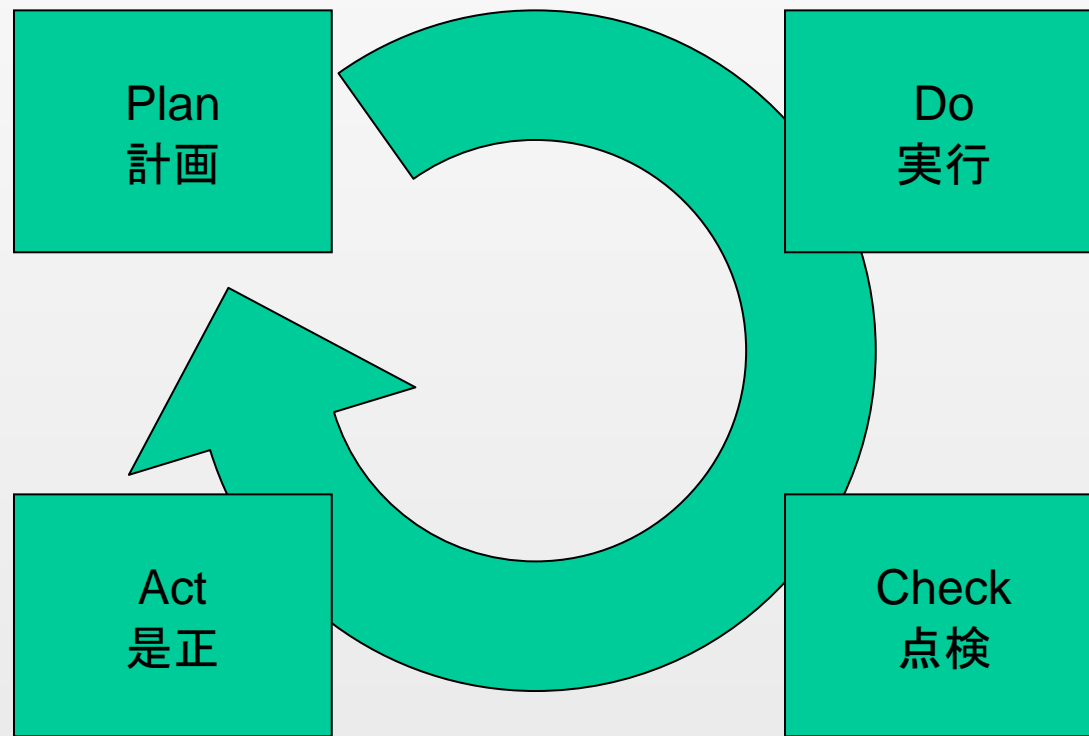
対策



(3) ソフトウェア・テストिंग のプロセス



一般のプロセス



品質管理の父デミングの提唱

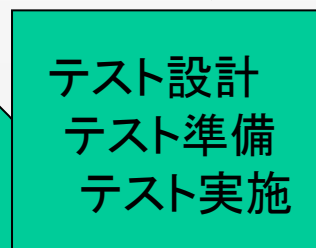


テストプロセス

どうやるか



実行する

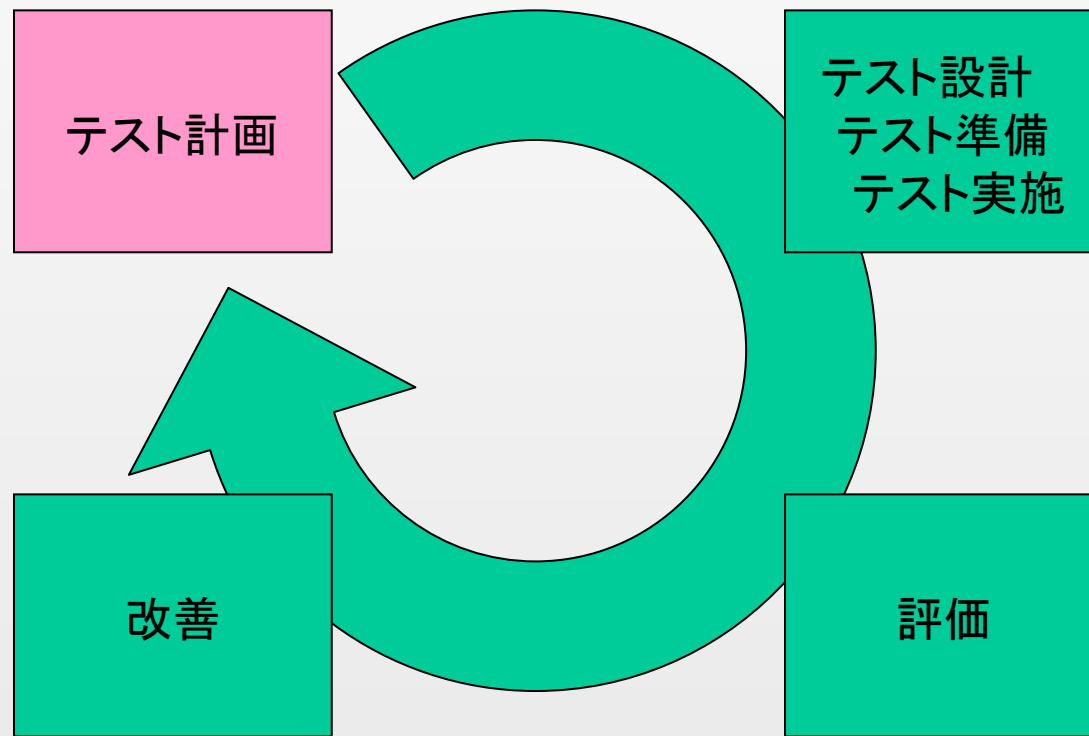


失敗は繰り返さない
成功はやり続ける

うまくいったか



1. テスト計画



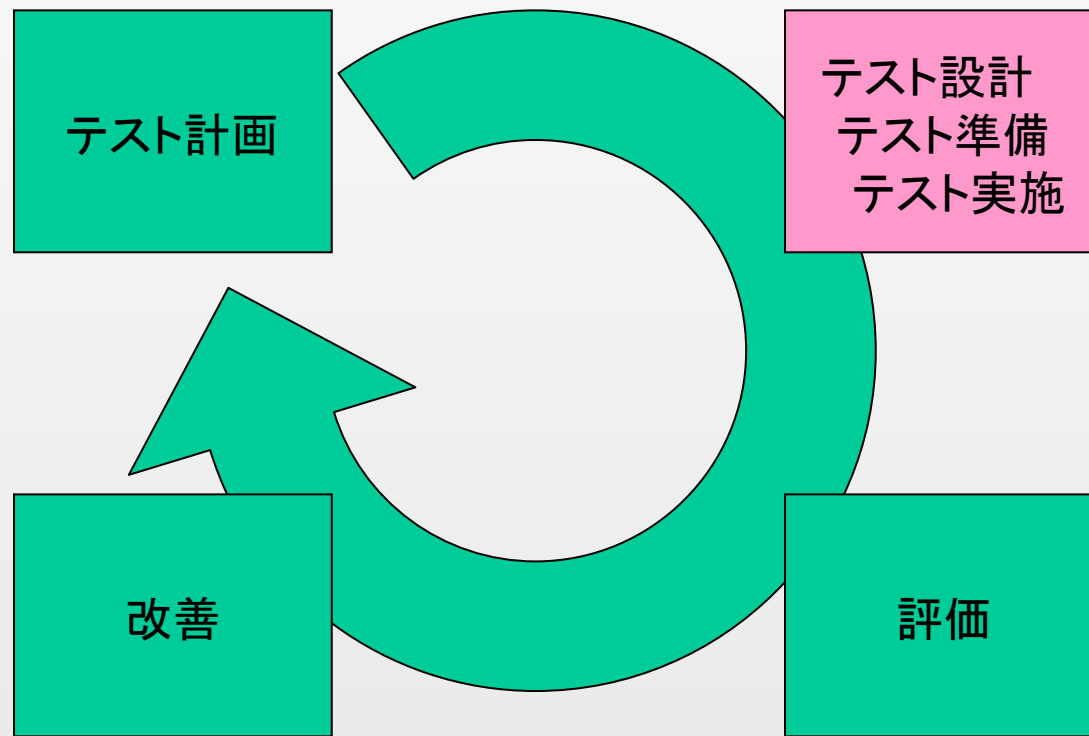


計画すべきこと

- **どこまでやるか** (スコープ)
 - 何をやらねばならないか、何をやらなくていいか
 - 期待されていることは何か (品質要件)
 - どうなったら終わりか (終了条件)
- **どのようにやるか** (工程、体制)
 - 責任者と担当者 (トレーニング)
 - 試験環境やツールの調達
- **制約は何か** (予算、納期)
 - スケジュール
- **管理方法** (進捗管理、バグ管理)
 - 共有ツールと管理方法



2. テスト設計と実施





テスト設計

- **網羅と効率**を考慮したテストケース・手順を設計する。
 - 入力や条件を網羅する。
 - 無駄を省く。
 - **期待値**を設定する。
 - テストデータや環境も設計する。



テスト準備

- テストが実施できる準備をする。
 - テストデータを用意する。
 - テスト環境を整備する。
 - テストツールを用意する。

- テストリソースを配置する。
 - 人、テスト環境の競合を考慮して、スケジュールを立てる。

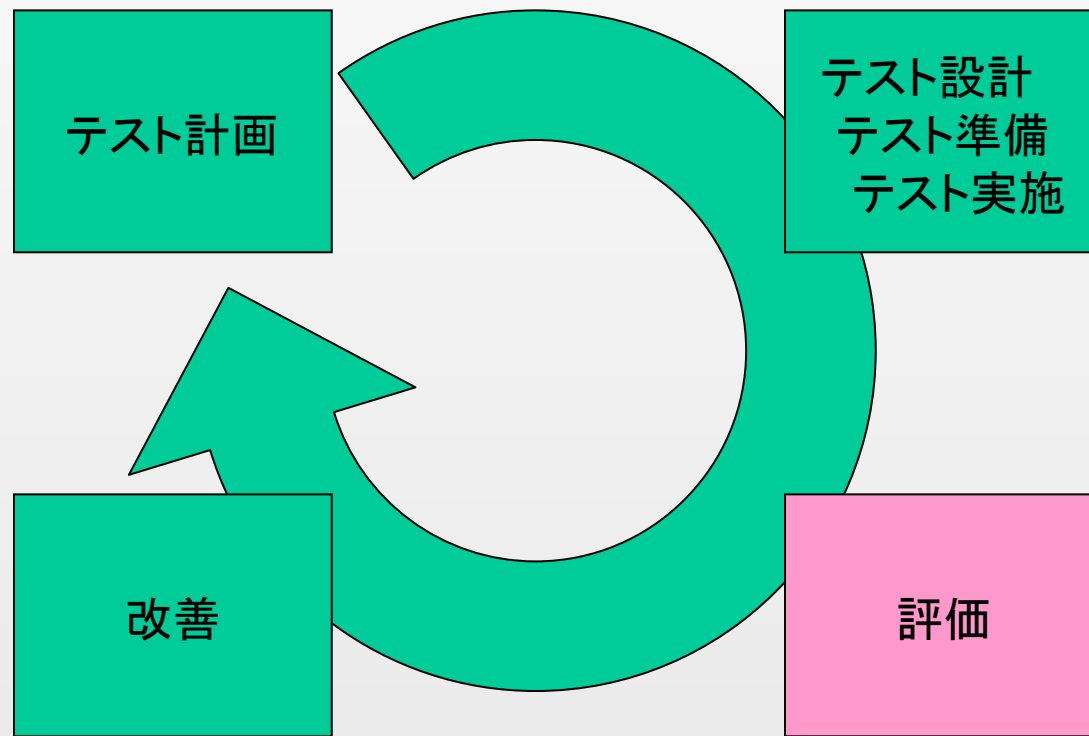


テスト実施

- テストケースを実行し、経過・結果を記録する。
 - どのバージョンのテストを行ったか。
 - 再現性はあるか。
 - 気づいたこと(インシデント)は何か。
- 不具合を報告する。
 - 不具合報告を作成する。
 - バグ管理ツールに登録する。
- テストの進捗を管理する。
 - どこまでテストが進んだか。
 - 終了基準を満たしているかどうか。



3. テストの評価



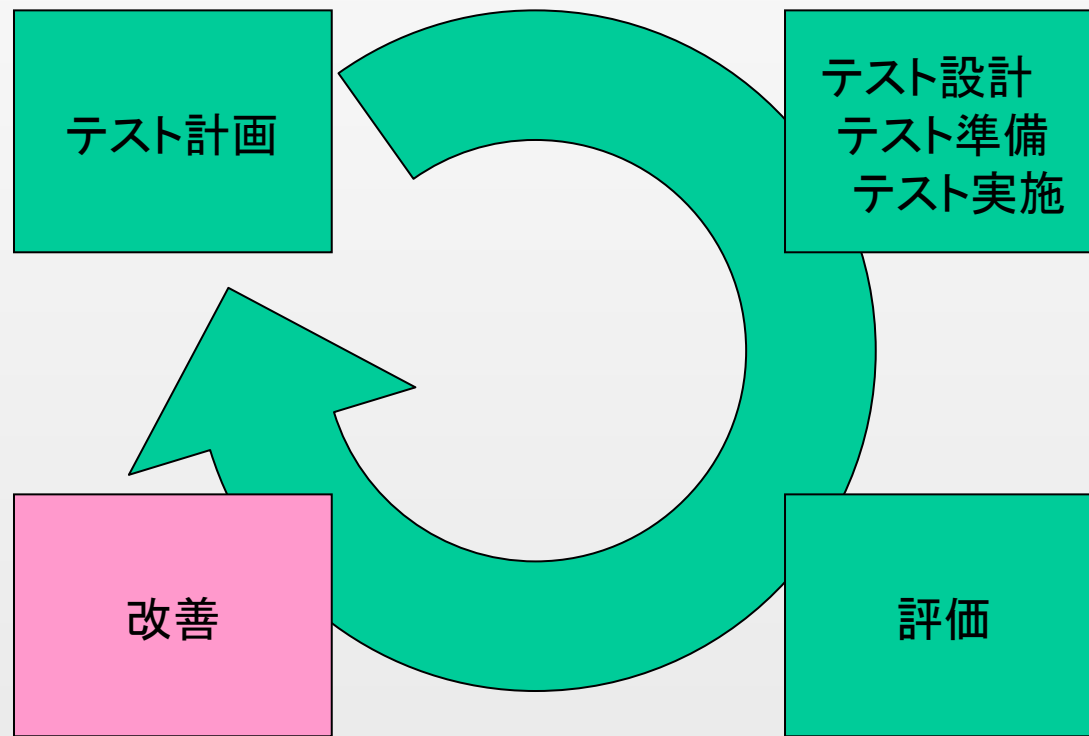


テストの評価

- 製品の品質を見極める。
 - バグの発見状況から、テスト対象の品質の良し悪しを見極める。
 - 想定した品質でない場合、テスト設計の見直しが必要かどうかを判定する。
- テストの品質を見極める。
 - 製品の品質だけでなく、テスト設計の品質の善し悪しも見極める。
 - テストの進捗と合わせ、より早期に多数の問題を発見できるように戦術を練る。
- 可視化する。
 - テストの進捗が把握できるように、グラフ化(バグ管理図)する。



4. テスト改善





テストを改善する

- テストの評価に基づいて、テスト設計や実行を見直す。
 - テスト設計の見直し
 - テストケースの追加
 - 優先順位の変更
 - テスト実施の見直し
 - 自動化
 - 担当の入れ替え
 - 管理情報の強化
- テスト計画そのものから変更するケースもある。
- テスト結果の分析は、テスト実行がすべて終わってからではなく、テスト実施中も随時実施する。



テストプロセスを改善する

- テストプロセスを総合的に評価する。
- この結果は、次のテスト、開発に活用する。
- テストに関する活動情報の要約も行う。
- テストに費やしたリソースの評価も行う。
- テストプロセス改善モデルとして、TPI(Test Process Improvement)モデルがある。



(4) ソフトウェア・テストニング の技法



良いテスト設計とは

- 「より少ないテスト実施回数」で
- 「より多くのバグが見つかるテスト」を行い
- 「テスト対象を漏れがないように網羅」する



そのためのポイント

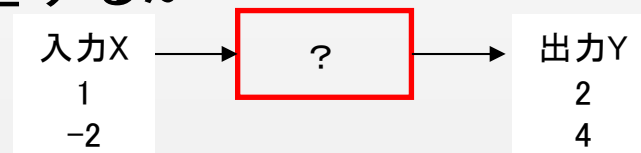
- 入力と条件の漏れをなくす(網羅性を上げる)
- 無駄なテストを省く
- 設計のコンセプトや省略時の理由を明確にする



テスト技法の分類

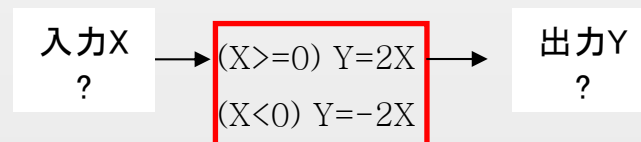
■ ブラックボックステスト

- プログラムの内部構造と動作に一切関知せずに、プログラムが仕様書通りの動作をするかを確認するテスト



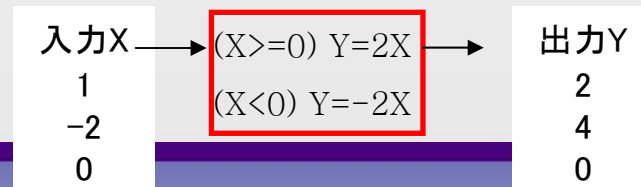
■ ホワイトボックステスト

- プログラムの内部構造（ソースコード、ロジック）を理解した上でそれら一つ一つが意図した通りに動作しているかを確認するテスト



■ グレーボックステスト

- 内部構造を理解した上で、ブラックボックス的なテストを行う技法





ブラックボックステストの技法

- 同値分割
- 境界値分析
- デシジョンテーブル
- 原因結果グラフ
- 状態遷移テスト
- 組み合わせテスト
- 直交表を用いたテスト
- エラー推測



ホワイトボックステストの技法

■ パステスト

- 制御フローテスト
- データフローテスト
- 状態遷移パステスト



(5) まとめ



ソフトウェア・テストの原則

- すべての組合せをテストすることは不可能。
- 環境の変化は想定できないこともある。



- 完璧なテストはあり得ない。
『テストはバグが存在することは示すが、バグが存在しないことを決して示さない。』(E. W. Dijkstra)



- 顧客が求めている重要な要求は何か。
- 限られた時間でどのテストを優先して実施すればいいか。



テストの定義

■ Myers (1979)

- テストとは、**エラーを見つける**つもりでプログラム（またはシステム）を**実行**する過程である。

■ Hetzel (1983)

- テストとは、プログラムまたはシステムの属性を評価することを目的とするあらゆる活動である。テストはソフトウェアの**品質を測定**することである

■ Rick D.Craig (2002)

- テストとは、対象とするソフトウェアの**品質を測定**して**改善**するための、テストウェアをエンジニアリングし、使用し、かつ保守しながら、同時並行的に進めるライフサイクルプロセスである



レポート課題(1)



レポート課題(1)

- 昨今のソフトウェア開発におけるテストイングの課題をまとめよ。
 - 本日の講義内容から
 - 本日のグループ討論における議論から
 - 日頃の問題意識から