

品質管理

s161053 前田欣耶

資料

- ずっと受けたかったソフトウェアエンジニアリングの新人研修
- 第11章「品質管理」



目次

- 1、品質管理とは
- 2、構成管理
- 3、デザインレビュー
- 4、テスト管理

1、品質管理とは

- メトリクス

品質を評価する尺度のこと。食品の場合、
・細菌数・味覚 など。システム開発の場合、次の
ページの表へ。

このような項目があり、値を上回っているかどうかで、一定の品質を満たしているか判断する。

例えば、

- ・レビュー工数10%以上
- ・適合テスト時のテスト密度30%以上

メトリクス	意味
レビュー工数	全工数に対するレビュー工数の割合
テスト工数	全工数に対するテスト工数の割合
レビュー実施効率	レビュー工数に対するして指摘件数の割合
テスト実施効率	テスト工数に対するバグ件数の割合
設計工程バグ摘出数	設計工程で発見・解決したバグの数
レビュー時の誤り密度	開発規模(総ステップ数)に対する指摘件数
レビュー網羅率	開発規模(総ステップ数)に対するレビュー対象部分の割合
レビュー時の平均修正時間	レビューで発見された誤りの修正にかかった時間の平均
テスト時のバグ密度	全テスト項目数に対するバグ件数の割合
テスト時のバグ収束率	バグ予測件数に対する総バグ件数の割合
テスト密度	ステップ数に対するテスト項目数の割合
テスト網羅率	テストされた部分の割合。ステップ数やパス数などで算出
平均バグ寿命	バグの修正までにかかった時間の平均
テスト消化率	全テスト項目に対する実施済みテスト項目の割合
バグ摘出率	目標バグ数に対する実績バグ数の割合

- メトリクスは品質を高める手段ではない
メトリクスはあくまでも品質をはかる尺度。システムの品質を高めるには次の手段がある。

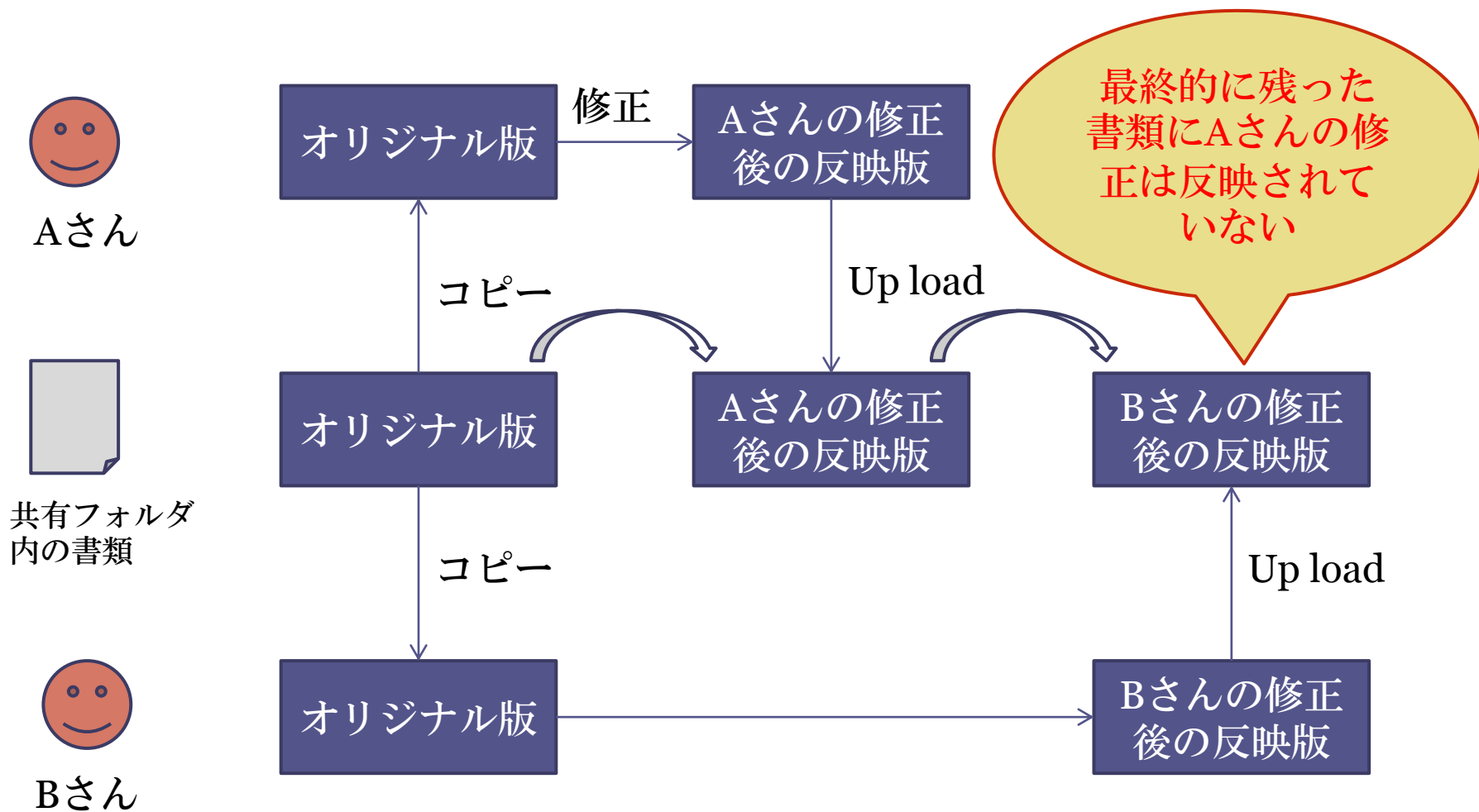
- 文書やプログラムを適切に管理する
- システム開発過程における設計書の品質を高める。
- 厳密かつ網羅的にテストを行う。

2、構成管理

- 文書、ソースコードの性質

- 大量の文書やソースコードを取り扱う
- 文書やソースコードは常に変化する
- その際、複数の人が文書やソースコードを書き換える

構成管理の大切さ



3、デザインレビュー

誤りのない設計書をつくるには...

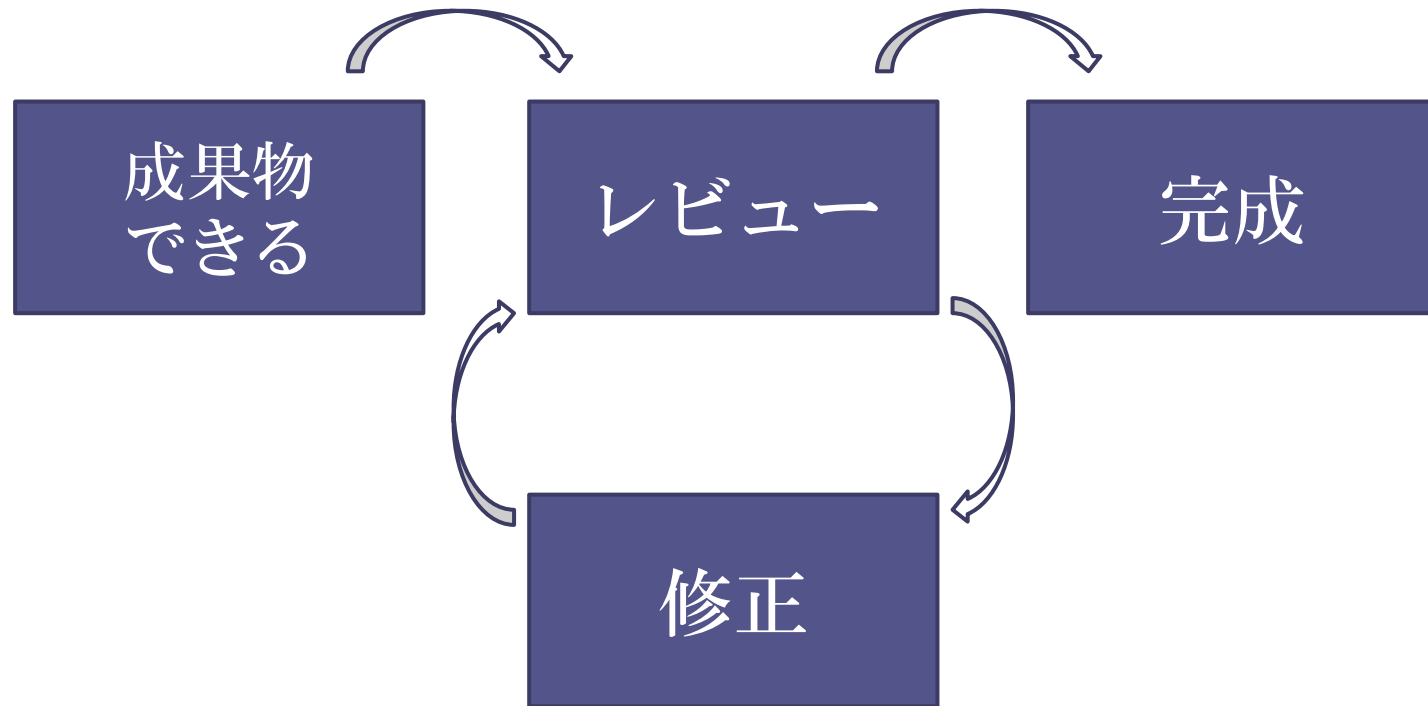
デザインレビューをおろそかにしない。

誤りのない設計書を作るということは、システムの品質の向上につながる。

設計書を確実なものにし、製造工程での手戻りをさける。

- レビューの準備

レビューはいつするか？



• レビューの準備

- 参加メンバーの決定

基本は設計書の作成に関与した人が参加する。

だが上司は参加しない方がいい。同僚からの指摘があると、上司からの評価が気になる。

- 日時の決定

- 資料の事前配布

• レビューの2種類の方法

- ウォークスルー

文書やプログラムを冒頭から最後まで順に確認していく方法。

- インспекション

範囲や対象を特定して、その部分を集中的に確認する方法。

レビューを重ね、最終的にはウォークスルーを使って、全体を通して確認。

- レビューで確認すること

レビューで確認すべき項目は「形式の正しさ」と「内容の正しさ」に分類できる。

- 「ですます調」か「である調」に統一されている。
- 「てにをは」の使い方が正しく、統一されている。
- 用語の定義が明確でぶれがない
- 数値が付いている
- フォント、スタイル、文字サイズが規則通りである。
- 表現が顧客の要請に一致している。

- レビュー結果の記録

レビューの結果は「レビュー記録表に」記録する。

項目	内容
指摘番号	指摘を特定する番号
分類	指摘の種類(R:訂正要請、Q:質問、C:コメント)
位置	指摘を受けた位置
指摘内容	指摘の内容
緊急度	指摘事項の訂正に対する緊急度
重要度	指摘の重要度
訂正担当者	指摘を訂正する担当者の氏名
訂正内容	訂正した内容
訂正月日	訂正した日付
訂正確認者名	正しく訂正されたことを確認した担当者の名前
訂正確認日	正しく訂正されたことを確認した日付

確認項目	意味
目的合致性	求められる目的に合致している
信頼性	求められる信頼性を実現している
可用性	求められる可用性を実現している
機密保護	求められる機密保護を実現している
操作性	求められる操作性を実現している
性能	求められる性能を実現している
保守性	求められる保守性を実現している
拡張性	求められる拡張性を実現している
互換性	求められる互換性を実現している
テスト容易性	求められるテスト容易性を実現している
可搬性	求められる可搬性を実現している
再利用性	求められる再利用性を実現している
接続性	求められる接続性を実現している

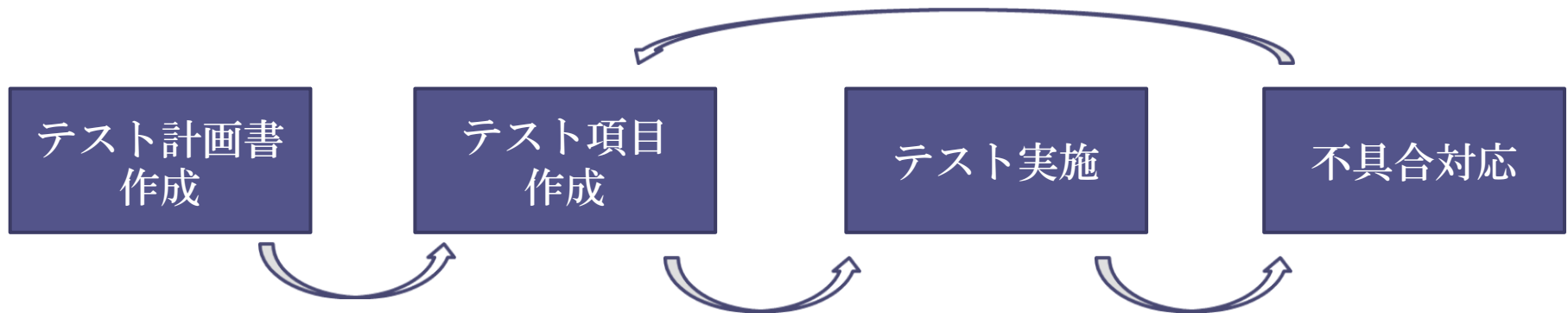
4、テスト管理

- テストの種類

システム開発で行うテストには大きく5つある。

- 単体テスト(プログラム単体をテストする)
- 結合テスト(複数のプログラムを結合したテストする)
- 総合テスト(システム全体を総合的にテストする)
- 受入テスト(納入されたシステムが要求を満たすかどうかをテストする)
- リリーステスト(リリース前に最終的に動作を確認する)

- テストの一般的な手順



- テスト計画書の作成
一般に次の項目を記述する。

- 品質管理
- テストスケジュール/場所
- テスト指標
- 終了条件
- 合格条件
- テスト対象
- テストケース
- 成績書
- テストツール
- テスト体制
- テスト環境

- テスト項目の作成
次のような観点から作成される。

- 機能テスト(機能の確認)
- 信頼性テスト(信頼性の確認)
- 性能テスト(性能の確認)
- 操作性テスト(操作性の確認)
- 複合テスト(複合的な条件での動作確認)
- 繰り返しテスト(繰り返し動作した場合の動作確認)
- 統合テスト(システム全体の動作確認)

- テスト項目を適切に作成するには
 - ホワイトボックステスト



- ブラックボックステスト
対象のプログラムの内部構造を気にせず、経験的に予測できる条件を中心に繰り返しテストし、網羅性を高めるアプローチ。

- テスト完了かどうかを判断する

テスト密度	バグ検出率	評価
高	低	品質良好
低	高	品質が良くない
低	低	テスト不足

- バグの状況から品質を判断する。

「予測されるバグ数の何パーセント」という考え方に基づいて、除去すべきバグ数の目標定めます。予測に対する実際のバグ数の割合が目標に近くなり、バグ発生件数の増加がゆるやかになったらバグは出尽くしたと判断できる。

- 代表的な予測方法としてテストの進み具合と累積バグ数のグラフを成長曲線にあてはめた方法がある。

