

12章 プログラミング工程

ずっと受けたかったソフトウェアエンジニアリングの授業2

鶴保 征城

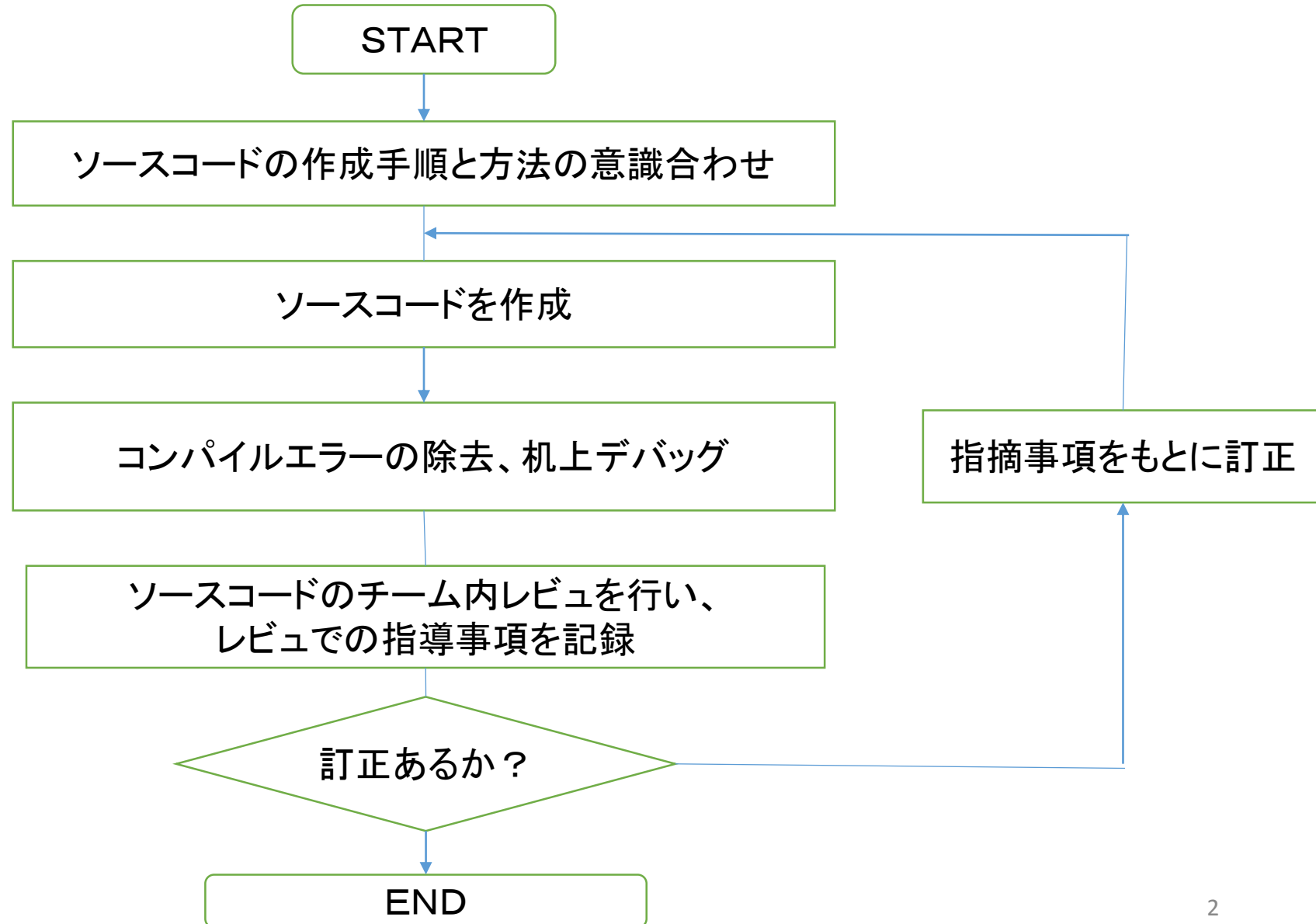
駒谷 昇一

60181027 芝 悠利

プログラミング工程

- 内部設計の仕様に基づきソースコードを作成しテストすること
- ソースコードの作成
- コンパイルエラーの除去
- ソースコードレビューの実施
- スタブ、ドライバ、テストデータの作成
- 単体テストの実施
- バグ対処

ソースコードの作成手順



ソースコードの作成

- 作成前に取り決める
 - プログラミング規約(コーディング規約)
 - 変数などの命名規則やコメントなどの記述方法、インデントや空白の入れ方など
 - 共通的なモジュールやクラスの設計
 - 1つのファイルでどのくらいのソースコードを作るか
 - 共通フォルダの更新方法
 - 更新とバックアップをとるタイミングの決定
 - テスト項目の選定方法、テスト実施方法
 - 障害処理票の書き方、管理の仕方

ソースコードレビューの準備と実施

- 目的

- 内部設計の内容がソースコードに反映されているかどうか

- 実施方法

- インспекション

- 特定の部分を決めて精密チェックし決められた規約に基づいて作成されているかを確認

- ウォークスルー(一般的にこちら)

- プログラムの軌道から最後までの流れを一通り追って、処理手順に間違いがないかを確認する

単体テスト

- ソースコードが内部設計書で記述された機能を満たしているかどうかを確認するためのテスト
- 実施方法
 - 全ての分岐を通るように作成
 - テスト項目を内部設計書から作成
 - 判定条件を満たしているかを確認
 - ホワイトボックステストで実施

テスト項目表

テスト項目表					
テスト区分 ☐ 単体テスト ☐ 結合テスト ☐ 総合テスト ☐ 移行テスト ☐ 運用テスト			管理番号 A-12-3-4	通番 1 / 2	作成者 佐藤
テスト対象名 (システム名、モジュール名) テーブルクライアント (Table-client.java)					
通番	テスト予定日 担当者	テスト内容	テスト実施日 担当者	テスト結果	対処内容
1	佐藤 (1/21)	商品ボタンを押した時、複数回同じ商品 ボタンを押すと、正常に数量が増えるか	1/22 岩本	OK	なし

ソースコード

```
#include<stdio.h>
int main(void)
{
  1000 行
  int a, b, c;
}
```

1/5 程度
テスト項目の数は
200 項目ぐらい

ホワイトボックステストとブラックボックステスト

- ホワイトボックステスト

- 全ての分岐、命令を実行するようにテスト項目を作成
- プログラムの制御構造を意識
- 命令網羅、分岐網羅、条件網羅がある
- 単体テストは分岐網羅

- ブラックボックステスト

- 入力に対して設計書で定義された出力が得られているかどうかの確認
- 制御構造ではなく、設計書の仕様に基づいてテスト

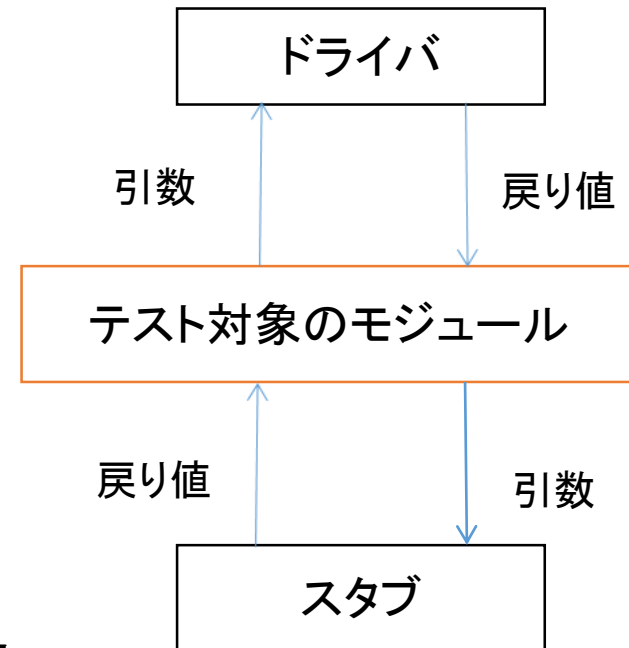
スタブとドライバ

- スタブ

- 呼び出されたこと、引数を表示
- 適切な戻り値をセット

- ドライバ

- モジュール実行に必要な引数をセット
↓ モジュールを実行
- 戻り値の内容を表示



単体テストの実施手順

1. ソースコードの作成
2. コンパイルエラーの除去
3. 詳細設計書の内容からテスト項目表の作成
4. スタブ、ドライバの作成
5. テストの実施
6. 障害処理票を起票
7. 障害管理簿に追記
8. 障害内容の原因を調査
9. 障害処理票の対処内容を記載
10. 障害対処(ソースコードの修正など)
11. 障害管理簿を更新
12. 障害管理簿の内容を分析

障害処理票

障害処理票				
障害発生時刻		2010 年 3 月 26 日 (金) 15 時 03 分		管理番号 12 - 34 - 56 - 7
テスト工程		☒ 単体テスト ☐ 結合テスト ☐ 総合テスト ☐ 移行 ☐ 運用		テスト項目管理番号 A - 12 - 3 - 4 - 5 通番
システム名 書籍検索システム		モジュール名 Search T		起票者 佐藤 太郎
トラブル分類	バグ混入工程	抽出すべき工程	抽出遅延理由	
☒ 1 設計バグ	1 要求分析	1 要求分析	1 設計レビュー漏れ	
2 製造バグ	2 システム提案	2 システム提案	2 仕様書訂正漏れ	
3 改造バグ	3 外部設計	3 外部設計	3 仕様書解釈誤り	
4 DB、OSバグ	4 内部設計	4 内部設計	4 ソースコードレビュー漏れ	
5 環境、HWバグ	5 製造	5 製造	5 テスト漏れ	
6 手順バグ	6 単体テスト	6 単体テスト	6 改修誤り	
7 提供データ誤り	7 結合テスト	7 結合テスト	7 ユーザ調整漏れ	
8 誤操作	8 総合テスト	8 総合テスト	8 ユーザ調整誤り	
9	9 移行	9 移行	9 仕様変更	
10	10 運用	10 運用	10 杜撰なバグ管理	
11 その他	11 その他	11 その他	☒ 11 その他	
障害状況				
Aを入力したが、BではなくCという出力となった。				
障害対処内容			対処者 久保田 二郎	
モジュール1を、入力AのときBを出力するよう修正した。			対処日時 4月1日21時03分	
			確認者 高橋 聖子	
			確認日時 4月2日14時32分	
備考 (関連項目に対する対策、次工程への提言など)				
単体テストをもっと丁寧に行なうことが必要。				

障害処理票の記入手順

1. 障害処理票を起票。障害管理簿に新しい行を追加(テスト実施者)
2. 障害処理票を提出(テスト実施者)
3. 障害内容を調査し、原因が判明・対応が決定した後、管理者に報告(障害対応担当者)
4. 障害管理簿に原因・対応内容を分析(管理者)
5. 障害処理票の障害処理内容を記入・管理者に提出(障害対応担当者)
6. 定期的な進捗会議で障害管理簿から障害の改善状況を確認(会議)

単体テストでの品質向上の工夫

- プログラミングのチェックはクロスチェックで行うことが望ましい
- 例

