テスト

FU14 ソフトウェア工学概論

第13回

吉岡 廉太郎

前回の内容

- 開発環境
  - 開発とテストを繰り返すための環境
- テスト手法
  - 目的に応じた分類：ブラック、ホワイト
- 単体テスト
  - コードレビュー
  - テストケースとカバレッジ

今日の内容

- テスト工程の作業
- テストの構成
  - 結合テスト手法の分類
  - システムテストの種類
- テスト計画と障害管理

テスト工程の作業

7.1.3

- 欠陥を発見する作業
  - 実装したコードに明らかな間違いが無いことを確認した後で実施する
- テストの成功＝欠陥の発見
- 欠陥の特定
  - 障害の原因になった欠陥を明らかにする作業
- 欠陥の修正
  - 欠陥を取り除くための作業

欠陥の種類

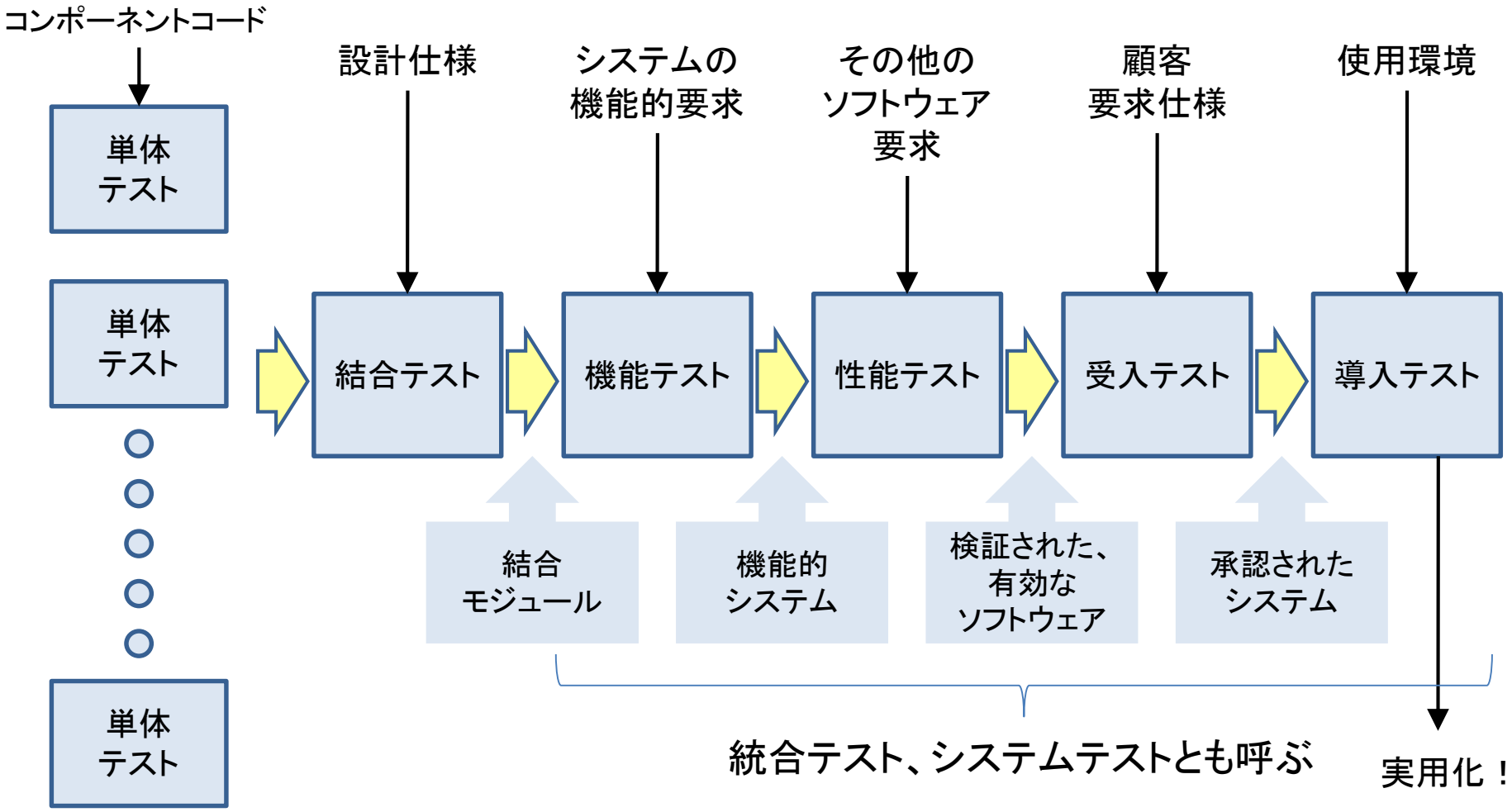
- 欠陥を発見するには、何を探せば良いかを知っていることが重要

アルゴリズムエラー	• 入力に対する出力が想定と違う：コードの間違い
計算・精度エラー	• 計算式の実装が間違っている
ドキュメンテーションエラー	• ドキュメントとプログラムの動作が一致しない
領域・境界エラー	• 特定の限界値においてシステムの性能が不十分
タイミング・連携エラー	• 複数のプロセスの順序または同時実行における問題
性能エラー	• システムが所定の性能を発揮しない
規定・手続きエラー	• 組織が定める規準や手順を順守していない

7.2.3

テストの構成

テストレベルともいう



7.3.1

テストと開発者の関係

- プログラミングの目的
  - 個人的なプログラム→自分の能力を示すもの
  - 顧客のための開発→どのような条件でも正常に動作する
- プログラムは個人の所有物ではない
  - プログラムをシステムの一コンポーネントととらえる
  - プログラムした人の所有物ではない
  - 欠陥を見つけたら、誰かを非難するのではなく、どのように修正するかのみを考える
- テストチームを独立させる
  - 不要な対立を防ぐ
  - 客観性の向上（思い込みの排除）
  - テストとプログラミングの同時進行が可能

結合テスト

- 個別に動作が確認されたコンポーネントをつなげ、システムに組み上げる

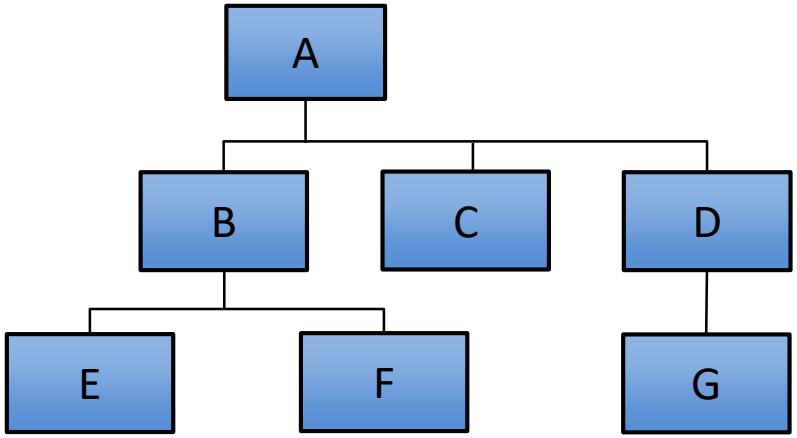
- 計画する内容
  - 順番
  - タイミング

- 計画の目的
  - 障害が発生した時、原因を特定できるように計画する
  - 効率化：開発、単体テスト、結合テストを同時進行

結合の手法

- ①ボトムアップ
- ②トップダウン
- ③ビッグバン
- ④サンドウィッチ
- ⑤変形サンドウィッチ

手法ごとにテストの手間と進行が異なってくる



例）7コンポーネントが3階層に分布

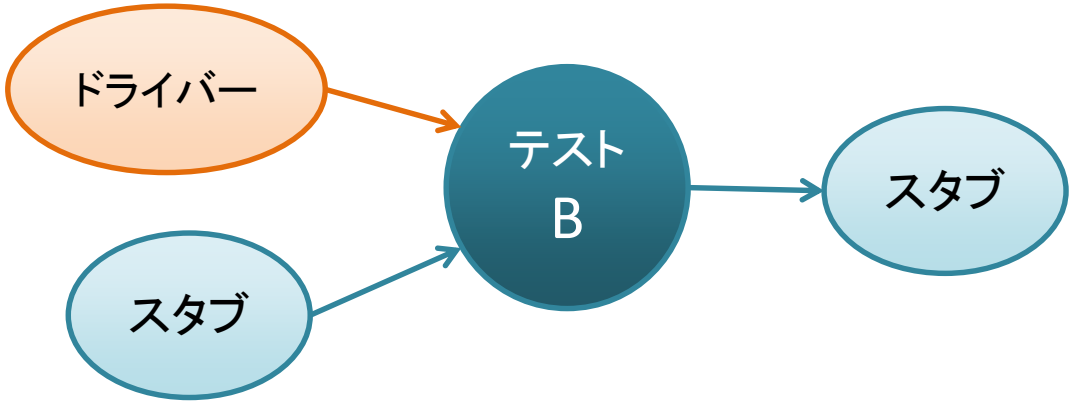
一般的には、システムの性質を考えながら、各部に最適な手法を適用する。
そのため、一つのシステムのテストに複数の手法を用いる。

7.2.3

ドライバーとスタブ

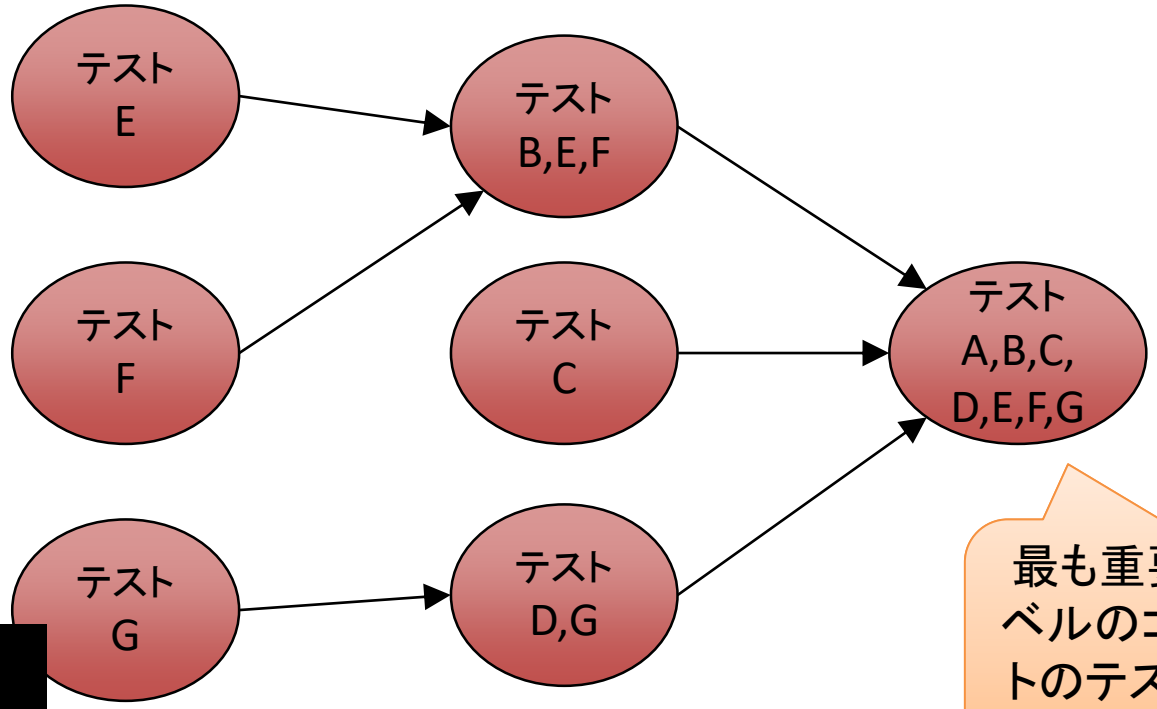
- コンポーネント・ドライバー
  - 対象のコンポーネントを呼び出しテストケースを渡す役目のコード

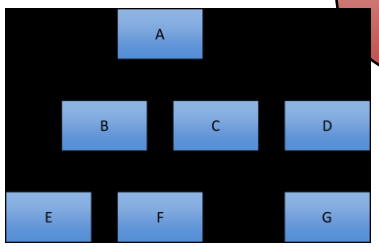
- スタブ
  - まだ結合しないコンポーネントの役目を模擬するためのコード



①ボトムアップ結合

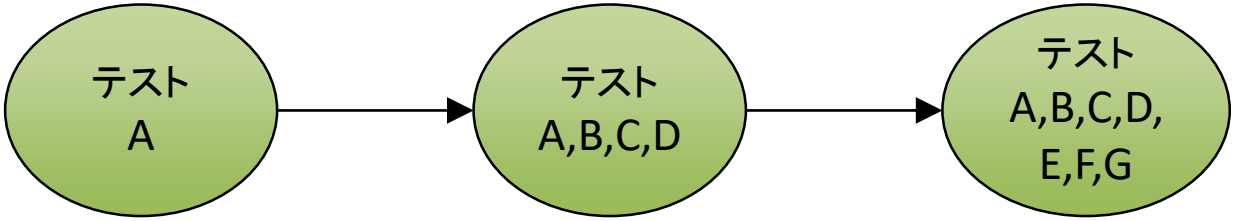
- 依存性を考慮して一段階ずつテストする



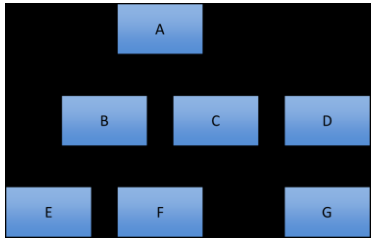


②トップダウン結合

- Aのみ単体でテストする

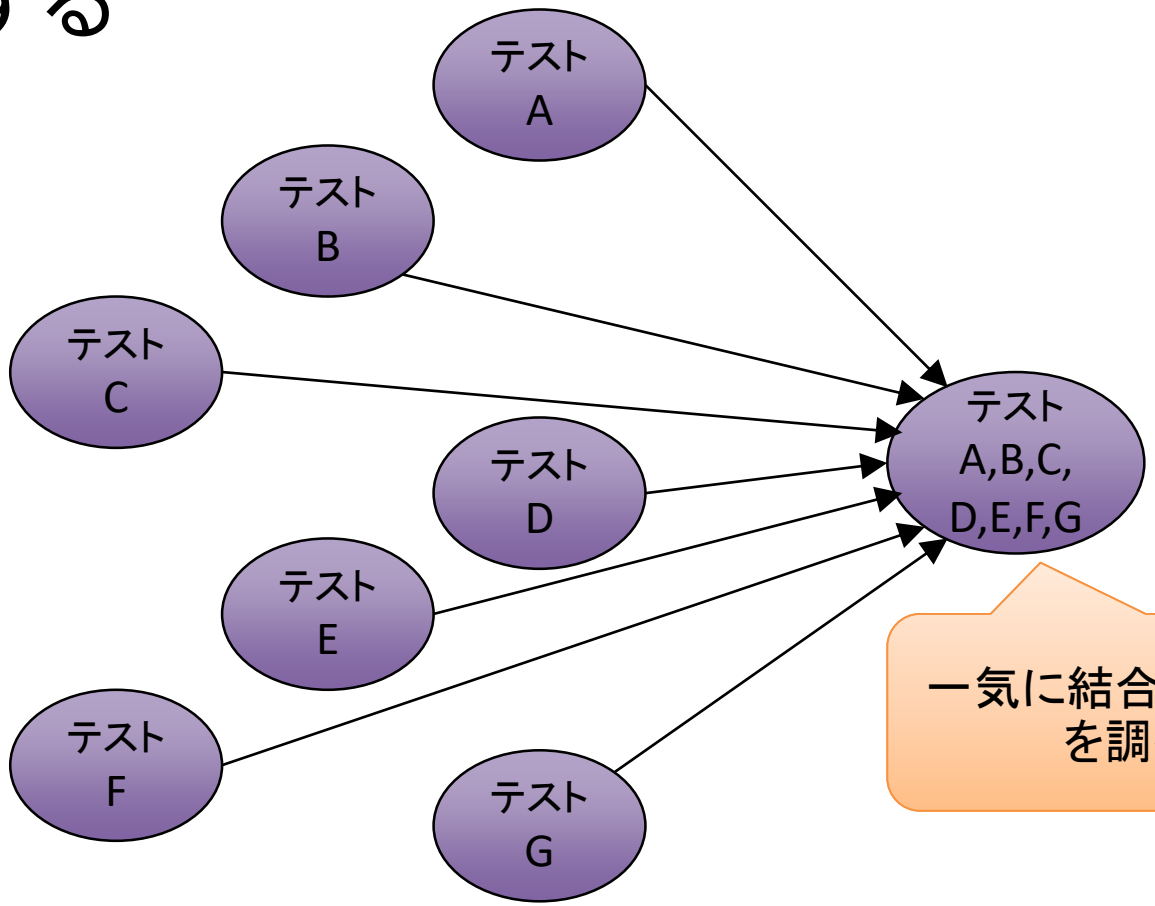


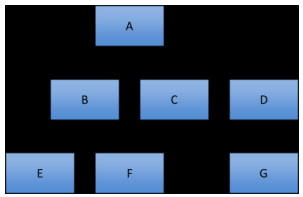
スタブがたくさん必要。
下位階層の個別テストが行われない。



③ビッグバン結合

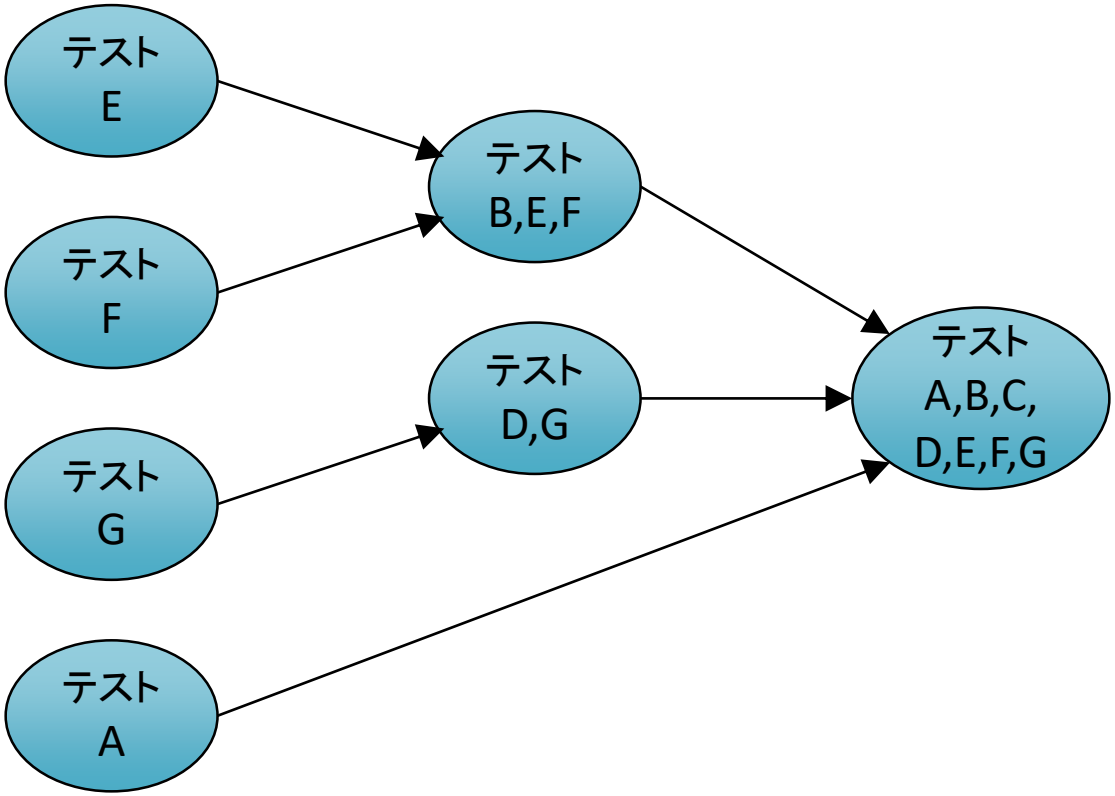
- スタブとドライバーで個別コンポーネントをテストする

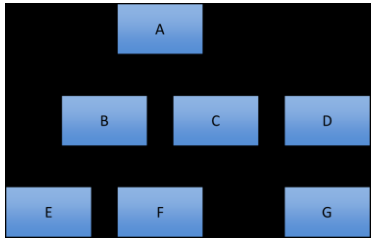




④サンドウィッチ結合

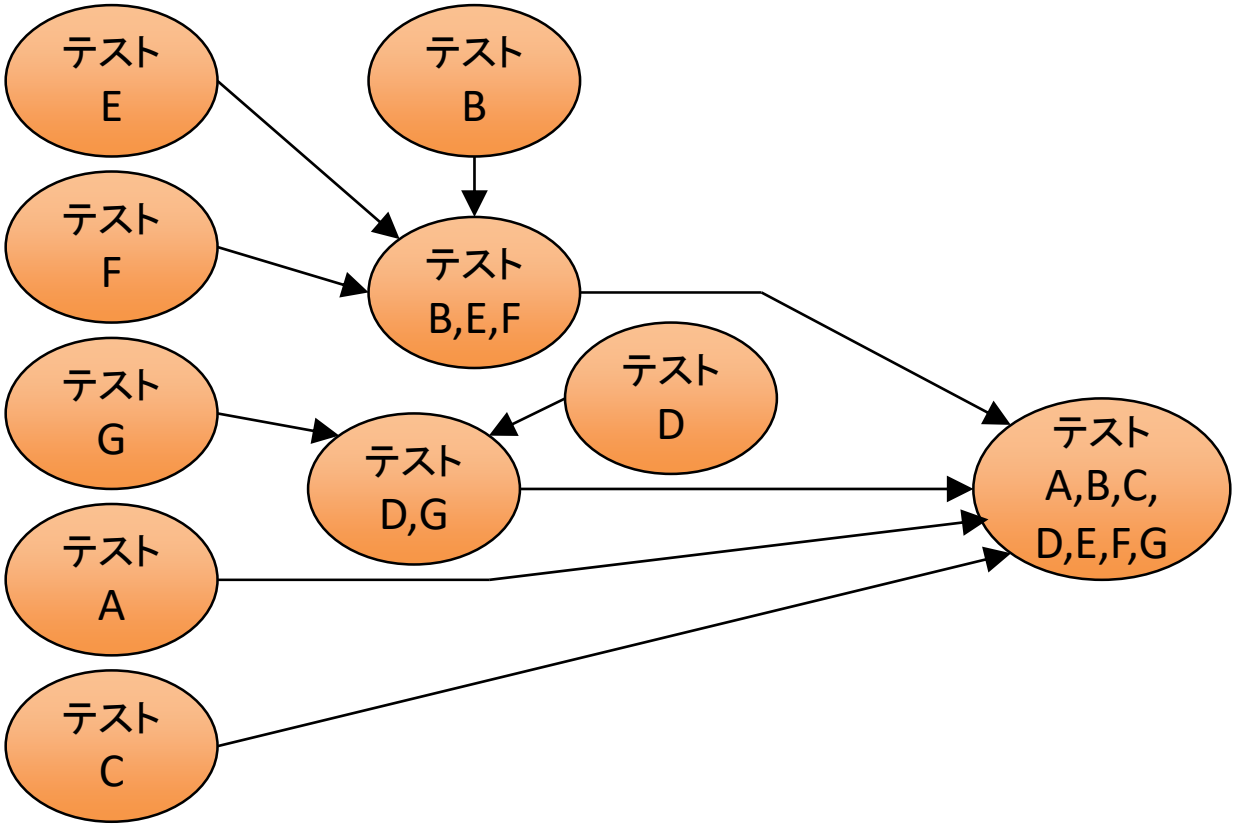
- システムを3階層に分ける

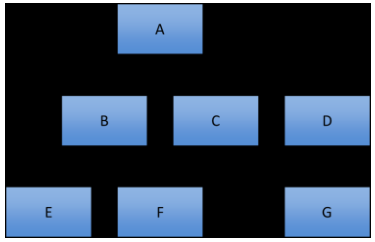




⑤変形サンドウィッチ結合

- 上位階層のコンポーネントを結合前にテストする





結合テスト手法の比較

	ボトムアップ	トップダウン	ビッグバン	サンドウィッチ	変形サンドウィッチ
結合	早い	早い	遅い	早い	早い
プログラムが実行可能になるまでの時間	遅い	早い	遅い	早い	早い
コンポーネントドライバの必要性	必要	不必要	必要	必要	必要
スタブの必要性	不必要	必要	必要	必要	必要
早い段階からの並列度	普通	低い	高い	普通	高い
特定のパスをテストする機能	簡単	難しい	簡単	普通	簡単
順番の計画と制御	簡単	難しい	簡単	難しい	難しい

- どの手法でも各コンポーネントは一度だけ結合する
- テストを簡略化するためにコンポーネントを変更しない→スタブ・ドライバーは別プログラム

システムテストの目的

- 単体テスト、結合テスト
  - 設計通りに作られていることを確認
- システムテスト
  - 顧客の要求を満たすように動作することを確認

システムテストの種類

7.2.4

① 機能テスト
- 統合されたシステムが要求仕様通りに動作するか
- 要求定義書に基づいてテストケースを作成する

② 性能テスト
- 非機能要求が満たされていることを確認する
- 検査対象：計算、レスポンスの速さ、結果の精度など
- テストチームが計画し実行する

③ 受入テスト
- システムが顧客が想定した通りに動作するか確認する
- 顧客が計画し、実施する

④ 導入テスト
- システムが顧客の環境で動作するか確認する
- 事前条件
  - システムを構成する、必要なデバイスを接続する、他システムと接続する
- 回帰テストを実施する

例）性能テストの種類

ストレステスト	• 高負荷状態で動作することを確認する
ボリュームテスト	• 大きなデータ量を扱えるかを確認する
構成テスト	• 要求されているさまざまなソフトウェア、ハードウェア設定での動作を確認する
互換性テスト	• 他システムとのインタフェースが仕様通りに動作するか確認する
回帰テスト	• 既存のシステムと置き換えても同等の機能が確保されることを確認する
セキュリティーテスト	• セキュリティーに関する要求が適切か確認する
タイミングテスト	• 応答や操作のタイミングに関する要求を満たしているか確認する

例）性能テストの種類（2）

環境テスト	• システムが導入・設置される環境で動作することを確認する
品質テスト	• 信頼性、保守性、可用性を評価する
復旧テスト	• 障害やデータ、電源、デバイス、サービスの喪失への対応を評価する
保守テスト	• 診断ツールと手順を評価する
ドキュメントテスト	• 必要なドキュメントが準備されたことを確認する
ユーザビリティーテスト	• ユーザインタフェースに対する要求が満たされているか確認する

テスト工程の文書の要素

① テスト計画
- 全機能と性質を確認にするための体制と計画

② テスト仕様
- 各テストの内容と評価の基準を詳細に記述する

③ テスト指示
- 各テストに用いるデータと手順を記述する

④ テスト報告
- 各テストの結果を記録する

⑤ 障害管理
- 発見した障害とその対応を記録する

例）テスト計画＆結果表

- 演習で利用している様式

大	大項目	中	中項目	小	小項目	テスト内容	確認内容	確認方法	作成日	作成者	検証日	検証者	検証結果
1	正常ケース	1	初期メニュー動作	1	四半期末精算登録画面の表示	初期メニューにおいて「四半期末精算登録画面を表示」ボタンをクリックする	「四半期末精算登録画面」がポップアップする。	画面のハードコピーを採取し、イメージを確認	5/29	星野	6/1	吉岡	×
				2	プログラムの終了	初期メニューにおいて「終了」ボタンをクリックする。	プログラムが終了する。	画面のハードコピーを採取し、イメージを確認	5/29	星野	6/11	吉岡	○
		2	四半期末精算登録画面の動作	1	四半期末精算登録	四半期末精算登録画面において、有有効な精算番号、材料業者コード、請求内容（複数の調達番号）、請求年月日を入力する。 以下の値を入力する。 精算番号　：01000 材料業者コード：010 請求内容　：01001 01002 01003 01004 01005 01006 01007 01008 010[illegible] 01010 請求年月日　テスト日（任意）	正しく処理された旨のメッセージが表示される。	画面のハードコピーを採取し、イメージを確認	5/29	星野	6/1	吉岡	○

テストケースを3段階程度に整理して、一覧表にする。
大項目は、正常処理／エラー処理といったレベルで良い。
中項目は、ユースケースを参考に考えると良い。
小項目は、ユースケースにおいて、処理を行う際の前提となるシステムの状態に着目し、確認が必要な状態毎にテストケースを設定する。

テストの内容（どのような状態で、何をするのか）を記述する

テストにおいて確認すること（正しく実装が行われた場合にどのような結果になるのか）を記述する。

確認事項を具体的にどのような方法で確認し、証跡を残すのかを記述する。

障害管理

- 想定と異なる振る舞いを障害として管理する

場所	• どこで問題が発生したか
タイミング	• いつ発生したか
症状	• 何が観測されたか
帰結	• その結果どうなったか
過程	• どのように起きたか
原因	• なぜ発生したか
重大度	• ユーザにどのような影響を及ぼしたか
費用	• どのような損失があったか

例）障害管理表

- 演習で利用している様式

障害番号	重要度	障害箇所（機能名）	現象	発見日	発見者	対応担当者	対応内容（原因と対応内容）	完了	完了日確認日
1	高	初期メニュー	「購入登録画面」がポップアップしない。	7/2	星野	星野	【原因】 InitMenu.java内のメソッドactionPerformedにおいて、「購入登録画面」の表示を行う箇所の条件式の誤り（スペルミス）。 【対応】 スペルミスを修正。	済	7/5
2	高	購入登録画面	不適切な社員番号を入力しても、エラーメッセージが表示されない。	7/2	星野	星野	【原因】 PurchaseControl.java内のメソッドPurchaseControlにて、クラスstaffを生成した後の判定を行っていない。 【対応】 判定処理を追加。	済	7/5

重要度で対応順を決定する

場所は機能名で示す

どのような現象がどのような条件の元で発生したかを記述する

原因と対応に分けて記述する。
【原因】なぜ発生したのか
【対応】どうやって回避または修正するかを示す

演習では、“テスト計画&結果票”＋“障害管理表”＝テスト報告書

本日のまとめ

- テスト工程の作業
  - より多くの欠陥を見つけるのが目的
- テストの構成
  - 結合テストの手法はコンポーネントの依存性で決まる
  - システムテストの種類は何をテストするのかで決まる
- テスト計画と障害管理
  - 計画と記録が大切

次回講義の事前学習:
8.1.2-3, 8.2.1-5, 12.1.2-3, 12.2.2, 12.3.3, 12.4.1, 12.5.3

期末試験

- 日時
  - 2018年8月9日（木） 3,4限　10:50-12:30
- 場所
  - M1 : [CS:std1]+[IT-SPR:std2]+[TGU:std2]
  - M2 : [SY:std3]+[SE-DE:std6]
  - M3 : [IT-CMV:std5]+[CN:std4]　※席も指定します
- 範囲・内容
  - 全14回の授業と演習のすべて
- 注意
  - 持ち込みはできません
  - 再試験はありません
- アドバイス
  - 授業資料のスライドを自分の言葉で補って説明できるようにする
  - 演習課題で行った作業を出来るようにしておく
  - http://borealis.u-aizu.ac.jp/classes/se1/
    ※質問に対する回答も掲載しています☺