

テストパターン

「実装レベルのテストパターンの必要性について」

株式会社オーエスケイ
小井土 亨

アジェンダ

- ◆ 求められているソフトウェア開発
- ◆ テストの問題点と解決策
- ◆ テストパターンとは
- ◆ テストパターンサブタスクの紹介
- ◆ テストパターン紹介
 - UIテストパターン (ViewState)
 - DBテストパターン (多層ドランザクション)

求められているソフトウェア開発

◆ ビジネスの変化

- 顧客のニーズに合わせて、ビジネスを柔軟に変化させることが成功の鍵
- ビジネスの変化に合わせて、ソフトウェアも変更を行うことが求められている
- 多くの変化は予測不可能

変化に強く、継続的で迅速なソフトウェア開発

継続的で迅速な開発に必要な項目

◆ 明確な要求定義

- 早い時期での顧客による仕様確認
 - ◆ 継続的な短期リリース

◆ 変更コストを一定に保つ

- プログラムを常に変更しやすい状態に保つ
 - ◆ シンプルな設計
 - ◆ リファクタリング(設計の改善)

◆ 高い品質

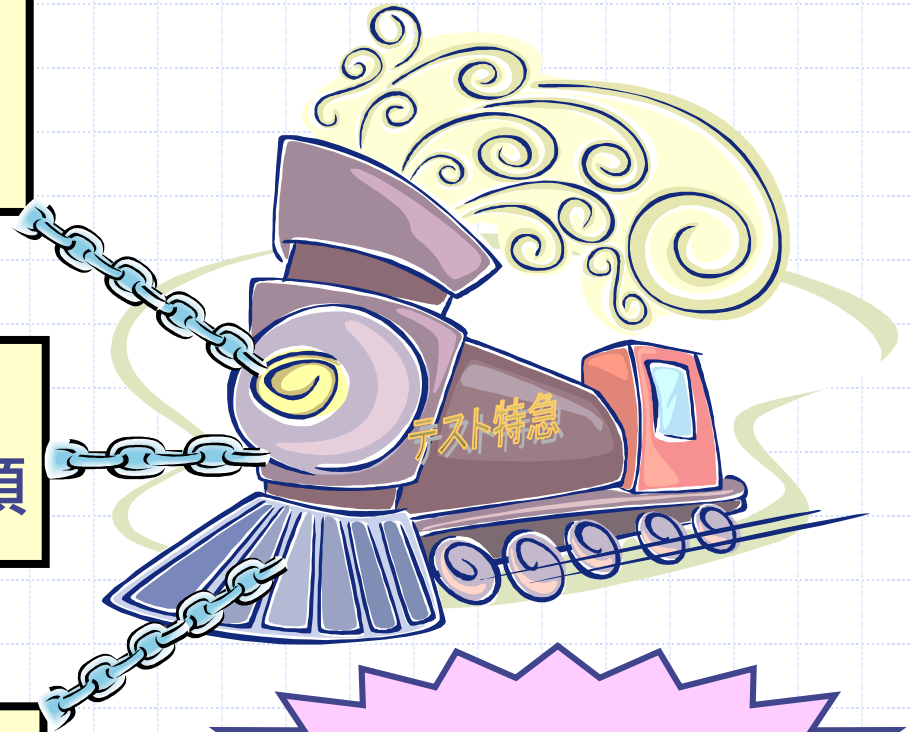
- 機能の正しさ、実行速度、使い勝手など

当たり前ですが「テストが重要」

継続的なリリース
継続的なテストが必要

リファクタリング
自動テストによる確認が必須

高い品質
テストによる確認が必要



テストが開発を牽引

テストの問題点と解決策

◆問題点1

- 計画が遅れて、テスト期間が確保できない
- 解決策 Test Driven Development

◆問題点2

- 手作業による人海戦術
- 解決策 テストフレームワークの導入

◆問題点3

- 品質の良いテストが書けない
- 解決策 テストパターン

テストパターンとは

◆ テストパターンの目的

- 効率的なテストの知識を共有する

◆ 対象とするテスト

- 自動化することができるテスト
- 単体テスト(単体クラスレベルのテスト)
- 総合テスト(シナリオレベルのテスト)
- 受入れテスト(システムの外部機能レベルのテスト)

◆ テストパターンの範囲

- テストのパターン
- テスト及びテスト対象のクラスから構成されるパターン



テストパターンサブタスク紹介

◆ テストパターンサブタスクとは

- パターンワーキンググループ実践タスクのサブタスクのうちの一つ
- 実践段階のテストに関する知識を、パターンを通じて共有する

◆ 目的

- 実装段階のテストに必要な様々な知識や知恵を、パターンあるいはパターンランゲージとして理解しやすい形で文書化する
- これを広く一般に共有・普及させ、さらに洗練させる
- 実装に携わる技術者を支援し、テストをより簡単により確実に作成できるようにする

◆ 活動

- テストパターンの収集と分類
- テストに関する情報の収集
- テストパターンランゲージの構成
- テストフレームワーク構築可能性の検証

テストパターンの例

現在、検討中のパターンです

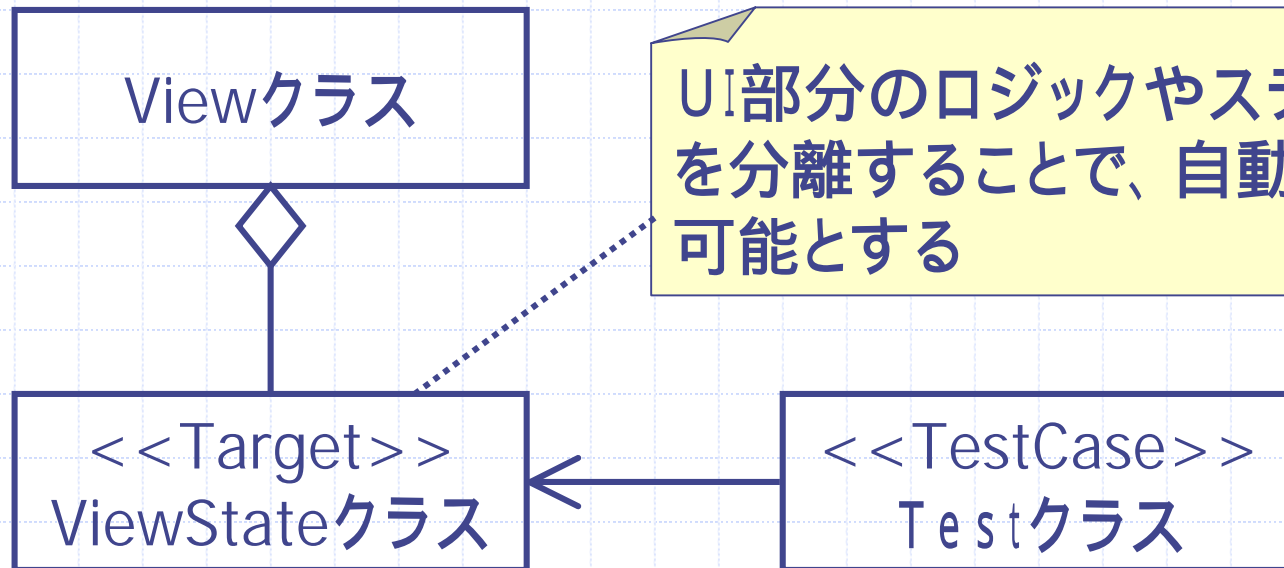


UIテストパターン (ViewState)

◆ 目的

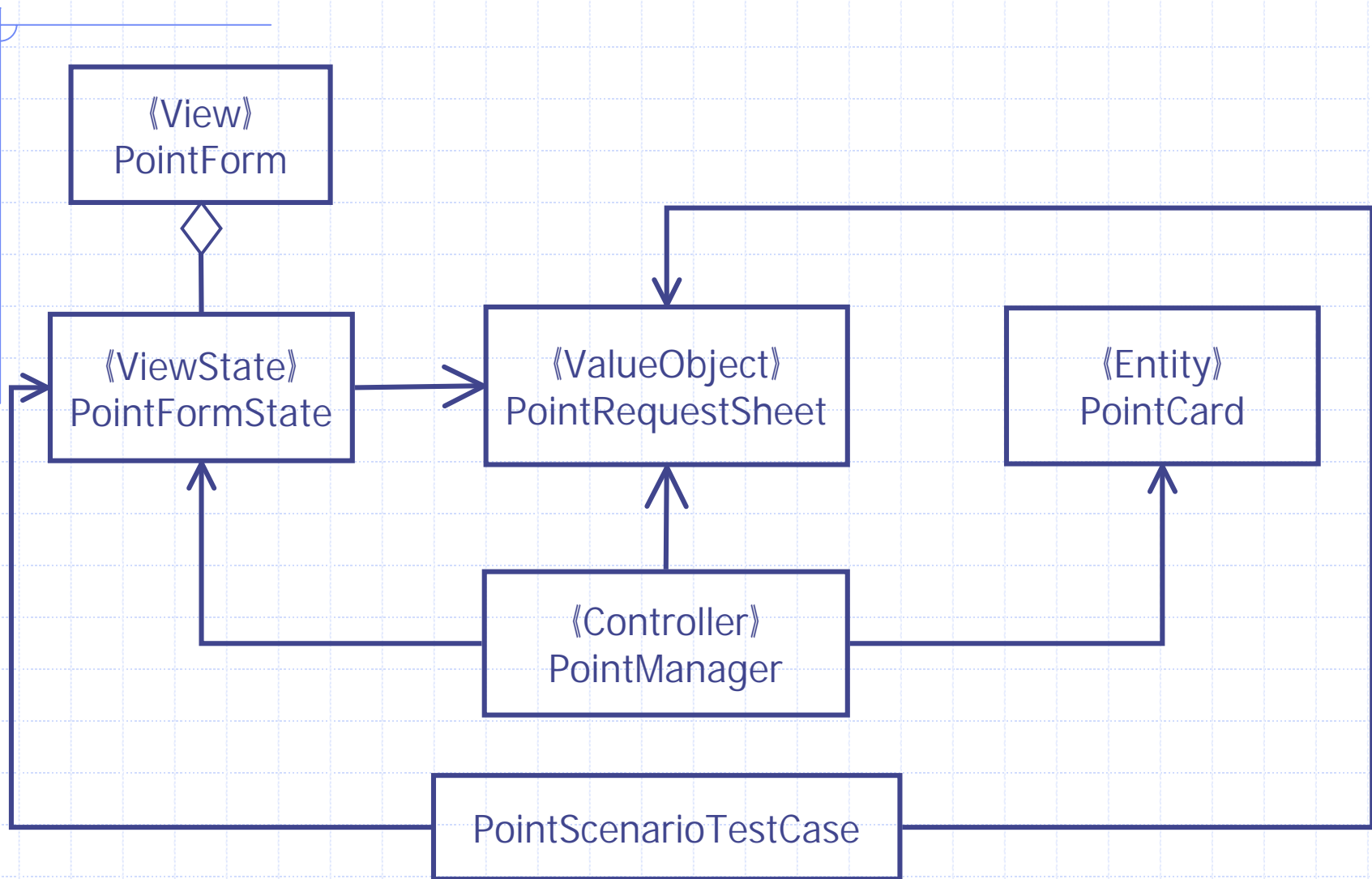
- UIロジックをテストする
- シナリオテストを実行する

◆ 構造



Visitorパターンでパラメータをクラス化すると実行しなくてもシナリオテストができる

UIテスト - 例

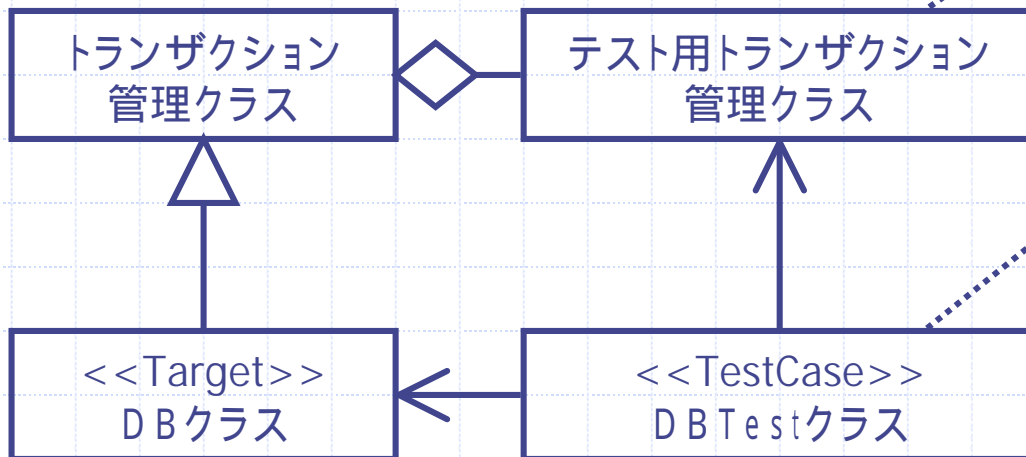


DBテスト(多層トランザクション)

◆ 目的

- データベースロジックをテストする
- 繰り返しテストを実現する

◆ 構造



テスト時だけ、処理のトランザクションとテストのトランザクションを同一化することでテストの実行とDBを基に戻す(ロールバック)ことを可能とする。