

アジャイル・ソフトウェア開発を快適にする テストパターン紹介

— テストパターンサブタスク活動報告 —

パターンワーキンググループ
実践タスク テストパターンサブタスク
小井土 亨

アジェンダ

- ◆ ソフトウェア開発を取り巻く状況とテスト
 - アジャイルソフトウェア開発
 - テストが重要
 - 良いテストと効果
- ◆ テストパターンの紹介
 - テストパターン
 - サブタスクの活動紹介

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(高いリスク)

◆ ソフトウェア開発のリスク対策

- ビジネスチャンスには、リスクがある
- リスクを軽減する開発方法論が必要

アジャイル・ソフトウェア開発

◆ アジャイル・ソフトウェア開発の原則 (抜粋)

- 早期かつ頻繁にソフトウェアを納品する
- 顧客のビジネス競争の優位性を確保する
 - ◆ 継続的に、要求の変更に対応する
- 顧客とは、契約交渉ではなく協調
- より効果的なことを探るために見直しを定期的に行う

変化に強く、継続的で迅速なソフトウェア開発
アジャイル・ソフトウェア開発

アジャイル開発で必要となる項目

◆ ニーズへの的確な対応

- 早期に動作するソフトウェアを顧客に提供し、フィードバックを繰り返す

◆ 変更コストを一定に保つ

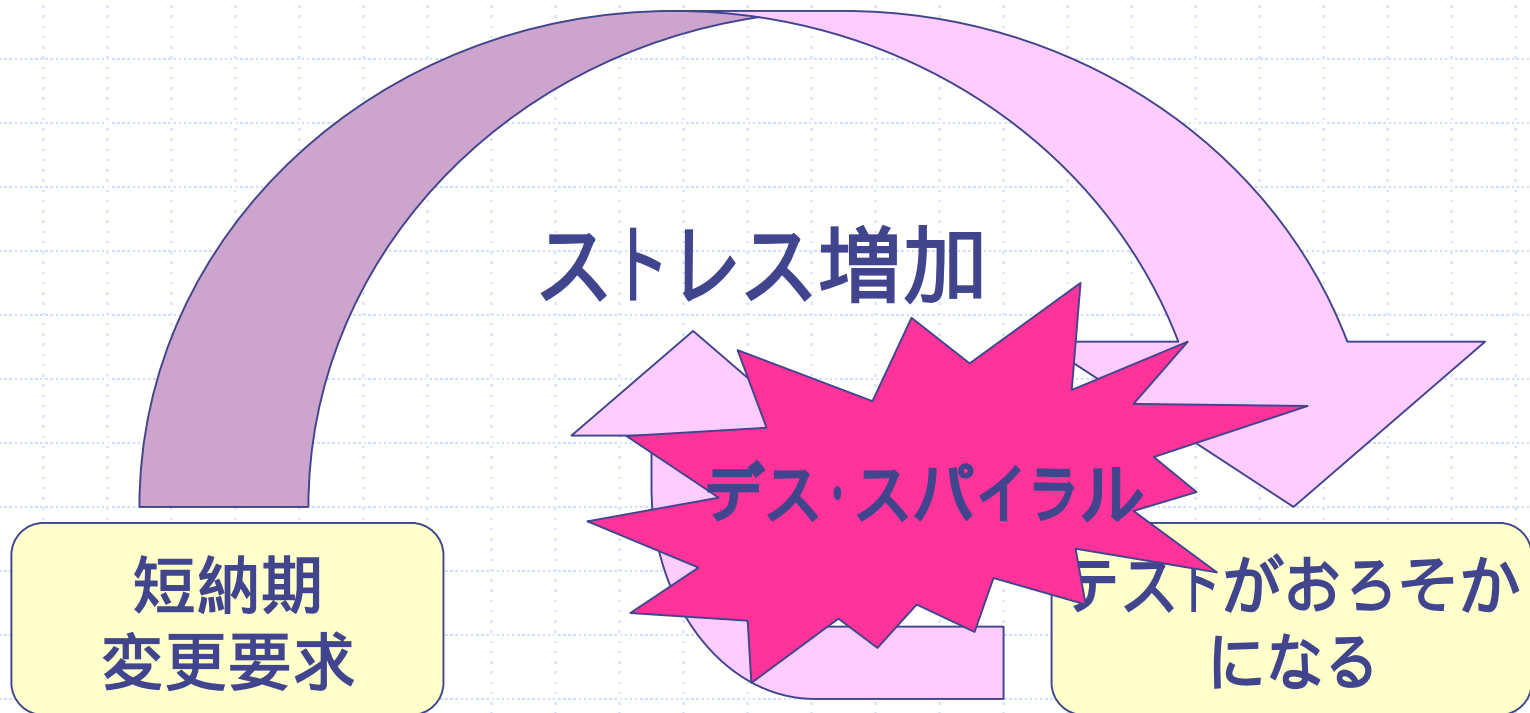
- プログラムを常に変更しやすい状態に保つ
 - ◆ シンプルな設計
 - ◆ 設計の改善(リファクタリング)

◆ 高い品質

- 機能の正しさ、実行速度、使い勝手など

ソフトウェア開発で発生する問題

- ◆ ストレス(短納期、変更要求)で品質が確保できない
 - 従来の開発手法でアジャイル開発を行うとした場合

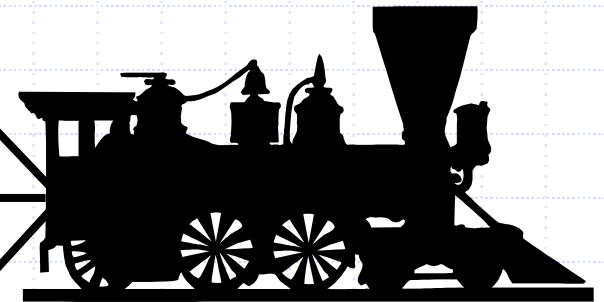


解決のヒントはテスト

継続的なリリース
継続的なテストが必要

テストが
アジャイル開発を牽引

設計の改善
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

テストパターンが
加速する

テストの位置付けが変化

◆ 以前のテスト

- 実施時期 - 設計・実装後
 - ◆ 設計の正しさの検証ではなく、動作を確認する
- 実施方法 - 仕様書を基に、人間が確認
 - ◆ 非常に多くの工数と期間が必要

◆ アジャイル開発でのテスト

- 実施時期 - 設計と同時あるいは前
 - ◆ 設計の一部としてテストを導入
 - ◆ 設計の正しさを実装の前に確認する
- 実施方法 - テストフレームワークの導入
 - ◆ フレームワークによる統一されたテストケース
 - ◆ 実装を進めながら、常にテスト(ビルド回数以上)
 - ◆ すべてのテストを定期的に行い、問題の発生を早期に発見

良いテストとは

- ◆ テスト目的が明確である
 - 何をテストするか明確である
 - その目的も明確で、更にひとつであること
- ◆ テストの判定が正しい
 - テストの成功、失敗が正しく判断されている
- ◆ テストを独立して実行することができる
 - テストが他のテストに依存することなく、独立している
- ◆ 繰り返し実行することができる
 - 何度でも繰り返して、テストを実行することができる
- ◆ テストを実行しても、状態が変化しない
 - テストを実行し、成功した場合でも失敗した場合でも、テストを実行する前と後で何も変わらない

良いテストの効果

◆機能的要求の確認

- 機能が要求通り作成されていることを実際に動作させて確認することができる

◆高い品質

- 多くのテストを作成及び実行することで、高い品質を維持できる

◆変更の影響への迅速な対処

- テストフレームワークなどを導入し、テストを自動化することで、変更の影響の把握と迅速な対応が可能になる

良いテストによるプログラム構造への効果

- ◆ **メソッドが一つの機能を実現している**
 - メソッドが明確な一つの機能だけを提供する
- ◆ **メソッドの結果を提供する**
 - 外部に対して、メソッドの処理結果を判断できるような何らかの方法を提供する
- ◆ **クラスやメソッドの独立度が高くなる**
 - クラスやメソッドが、他のクラスやメソッドの依存が低い
- ◆ **特定の環境への依存度が低い**
 - 特定のファイルやデータベース構造などに対する依存度が低い

まとめ(アジャイル開発でのテスト)

- ◆ **ビジネス競争を優位にするソフトウェアが求められている**
- ◆ **俊敏なソフトウェア開発**
 - アジャイルソフトウェア開発
 - リスクを軽減する開発手法
- ◆ **テストの重要性**
 - 守りのテストから攻めのテスト
 - 良いテストの効果

テストパターンの現状

◆ 様々なテストパターン

- テスト駆動開発に関連するテストパターン
 - ◆ テスト駆動開発のパターン、レッドバーに関するパターン、テストに関するパターン、など
- テストチームのテストに関するパターン
 - ◆ テストに関するパターン、テスト技法に関するパターン、テストプロジェクトのマネージメントに関するパターンなど
- 特定テストに関するパターン
 - ◆ 負荷テストに関するパターン
- テストプロセスに関するパターン
 - ◆ プロセス改善のパターンなど

テストパターンサブタスクの活動

◆ 活動目的

- 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して共有可能にする

◆ 現在対象としているテストパターン

- 設計及び実装に関するテストパターン
 - ◆ 幅広い開発プロセスのテストパターンが対象
- 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、シナリオテスト、受入テストなど
- 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から構成されるパターン

テストパターンサブタスクの活動内容

- ◆ テストパターンの収集と分類
 - 多くのテストパターンを分類し、テストパターンマップを作成する
- ◆ Kent Beck 著「テスト駆動開発入門」読書会
 - 月に1、2回のペースで開催
 - 書籍内のパターンについて、カタログ形式での記述や補足説明を作成
- ◆ テストパターンに関する意見交換
 - 収集したパターンの意見交換やメンバーが記述したパターンのレビューを行う

テストパターン紹介1

◆ テストファースト

- テストファーストを導入し、自動テスト実施



テストパターン紹介2

◆ ビューステート

■ 目的

- ◆ ユーザーインターフェイスをテストしたい

■ 解法

- ◆ ユーザーインターフェイスの状態やロジックを別クラスに分離し、自動テストを行う

◆ 半自動テスト

■ 目的

- ◆ 人でしか確認できないテストを効率的に実行したい

■ 解法

- ◆ 人による確認を含んだ半自動テストを作成する