XPJUG-Kansai
日本XPユーザグループ関西
http://www.xpjug.jp/

アジャイル開発における品質保証

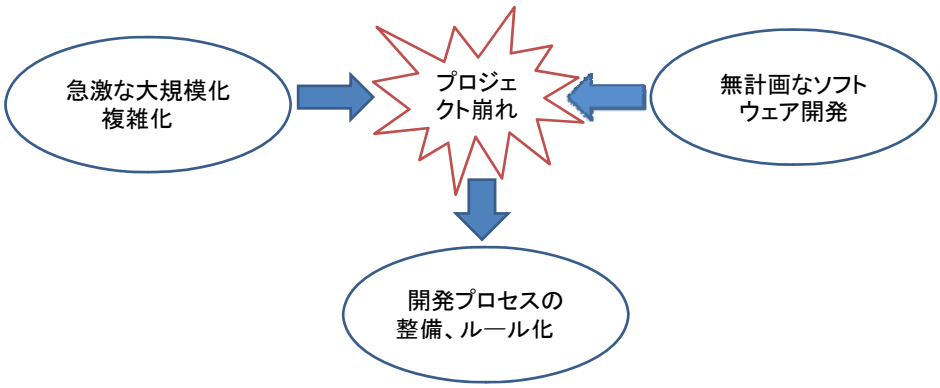
‘08.6.5 細谷泰夫
JaSST Kansai ‘08

アジェンダ

- 何故アジャイルプロセスか？
- アジャイルプロセス導入の障壁
- 品質保証とは？
- ウォータフォールモデルでの品質保証のおさらい
- XPにおける品質保証
- まとめ

何故アジャイルプロセスか？

SW開発が直面する問題と取り組み



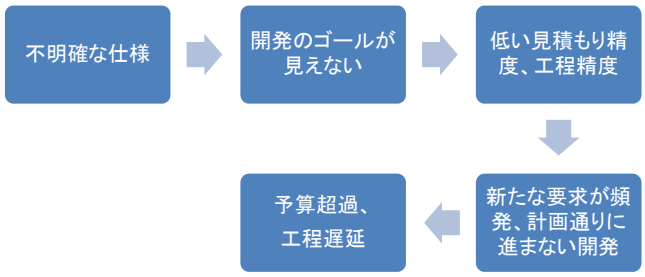
無計画な開発の減少、結果としてプロジェクト崩れの減少

一方で・・・　ルール化による弊害も現れている

ルール化により現れた弊害

ルール化が間違っている訳ではない。
ルール化により初歩的な問題が解決し、新たな段階の問題が見えてきた

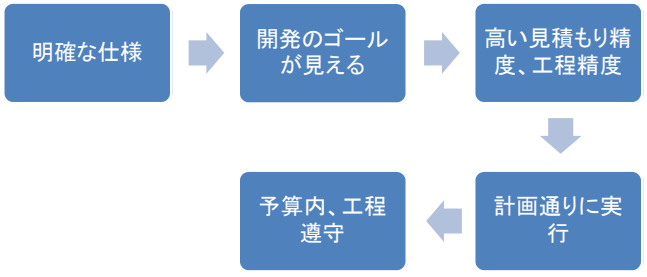
ルール化による改善パターン　改善前



ルール化により現れた弊害

ルール化が間違っている訳ではない。
ルール化により初歩的な問題が解決し、新たな段階の問題が見えてきた

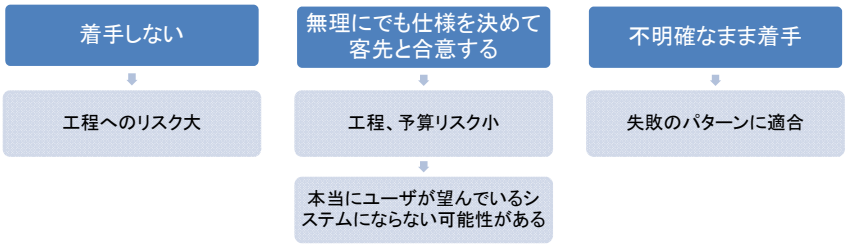
ルール化による改善パターン　改善後



これで開発が失敗することはない、めでたし、めでたし・・・

本当にそうだろうか？

明確な仕様が決まらなければ？



いずれのパターンも失敗する。反省会の言葉はいつも「仕様は早期に決めましょう」だったりしませんか？

実際問題として、ユーザが満足する仕様を早期にFIXするのはとても難しい。

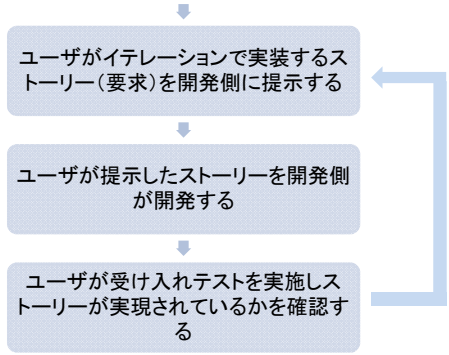
アジャイル開発では？

「要求は変化するもの」が前提

開発側とユーザが協調してゴールを目指すことに重点

明確な仕様が決まらなくても

開発を短期のイテレーションに分割する



繰り返しの中で、

ユーザは、自身が満足する要求を正確に掴み、正確に開発側に伝えられるようになる。

開発側は、ユーザが満足する要求についてユーザとイメージを共有できるようになる。

ユーザが満足するソフトウェアを適正な予算と期間で提供するための有効な方法としてアジャイルプロセスを導入する

アジャイルプロセス導入の障壁

書籍、シンポジウム、雑誌等でアジャイルが度々紹介されている

認知度アップ。反面、実行している企業は多くはない

何故か？

アジャイルプロセスを導入する上で、様々な障壁が存在する。

契約　調達　**品質保証**

今回は品質保証に注目！

品質保証とは？

品質保証について考えてみる

- 「品質保証」とは「品質」を「保証」すること
- では、「品質」とは何か？

「ソフトウェア品質保証の考え方と実際」（日科技連）によると・・・

JIS、ISOの定義

- 「品質とは、ユーザの要求（ニーズ、使用目的）を満足させるために製品（含むサービス）の持つべき特性である。」

クロスビー（P.Crosby）の定義

- 「品質は「要求に対する適合」である」

ワインバーグの定義

- 品質は誰かにとっての価値である。・・・
- ここで価値という言葉は、「人びとはその要求が満たされるなら、喜んで対価を支払う（また何かをする）か？」ということを意味している。

品質保証の観点からの品質の定義

- ワインバーグの言う「誰か」とは品質保証の観点では「ユーザ」であるべきと述べており、品質保証の観点での品質を以下のように定義している。
- 『品質は、ユーザにとっての価値である』
- すなわち、**『品質は、ユーザの満足度である』**

品質とは、ユーザの満足度である

品質保証とは、製品が「ユーザの満足」を実現することを保証すること

ウォータフォールプロセスにおける品質保証

Verification（検証） & Validation（妥当性確認）

Verification（検証）

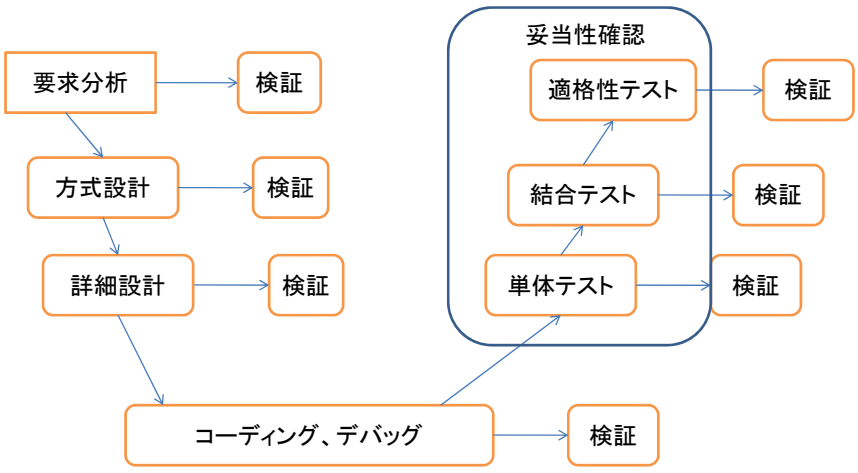
- 各開発工程が適切に実行されているかどうかをチェックすること

Validation（妥当性確認）

- ソフトウェア開発工程の最後に、ソフトウェアの要求事項に従っているかどうかを確認するためにソフトウェアを評価する工程

参考文献：ソフトウェア品質保証入門（日科技連）

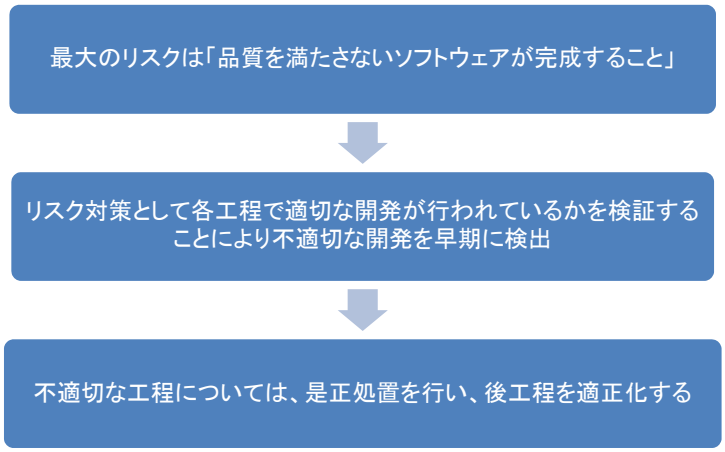
V&Vの概念



検証の手段：品質メトリクスの実績と指標との差異分析による評価

評価対象となる品質メトリクスと実際の品質（ここでは要求を満たすこと）の相関関係を定義し、評価の根拠としている。

各工程で検証を行う理由



eXtreme Programmingにおける品質保証

XPにおける開発の流れ

イテレーション

- 1～2週間の定間隔の開発周期。

計画ゲーム

- イテレーションで実施するストーリをユーザと開発側との共同作業により決定する。ストーリーは作成された時点で見積もりを実施する。

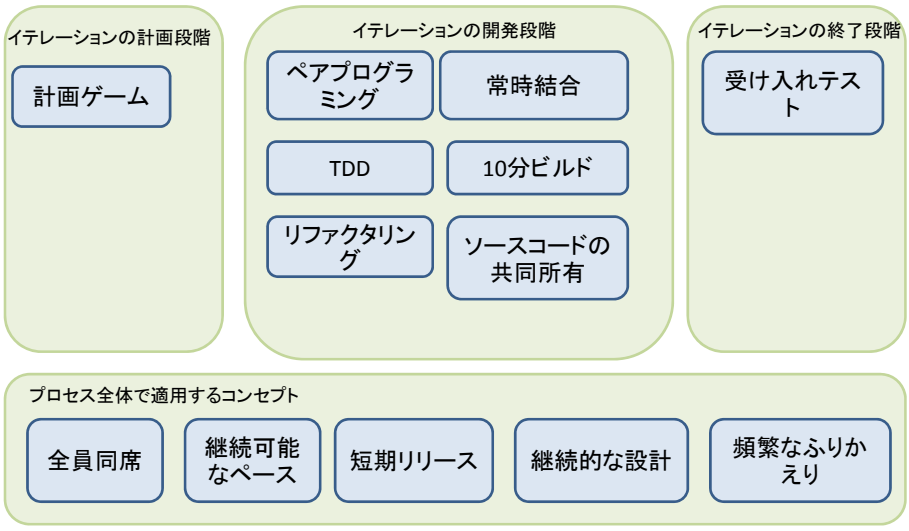
開発

- 開発が決定したストーリーをタスクに分割し、実装する。タスクへの分割、共同設計（メタファ、概略モデル）、実装の流れ

受け入れテスト

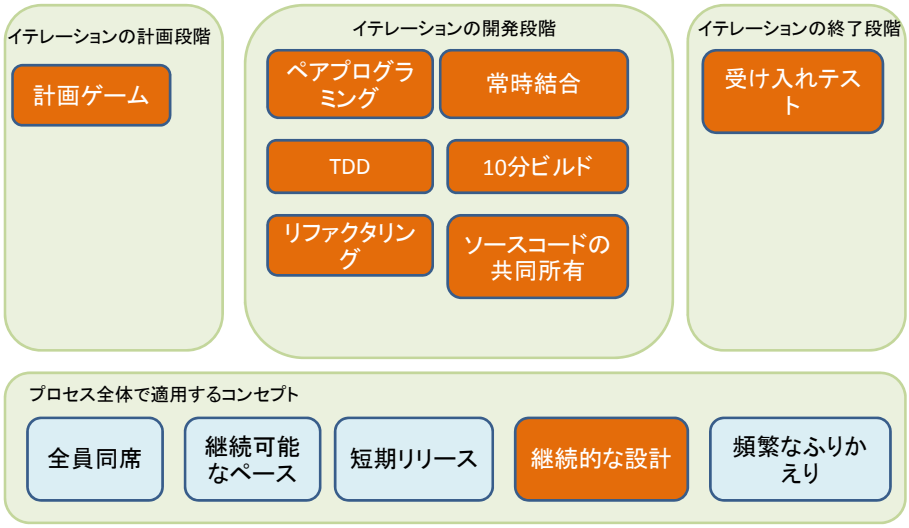
- ユーザがイテレーションの成果物であるソフトウェアがストーリーを満足しているかをテストする

XPのプラクティス（抜粋）



この中で品質確保に直接関連するプラクティスは？

XPのプラクティス（抜粋）



多くのプラクティスが品質確保に寄与している。

ペアプログラミング

- 2人1組で1台のPCに向かってプログラミングを行う。

テスト駆動開発（TDD)

- 「今から実装するプログラムはどのように動作すべきか？」という観点でテストコードを作成し、テストにパスする実装を行う。テストコード作成→実装→テスト結果確認を5分程度のスパンで繰り返す。

リファクタリング

- 「動作を変えずにプログラムの構造を改善する」こと。XPにおいては、TDDで作成したテストをパスした状態を保ちながら構造を変更していく。

10分ビルド

- ビルドにかかる時間を10分以内に抑えることにより、1日に何度もビルド可能な状態にする。

常時結合

- 最低1日に1度は各メンバーのコードを結合し自動的な結合試験をパスするかどうか確認する。

コードの共同所有

- コードの所有権を決めずに、どのペアがどのコードを変更するかは柔軟に決める。ルールは必ずテストを通ること。

継続的な設計

- ユーザが作成するストーリーを満たす最適な設計を継続的に実施する。ユーザのストーリーから洞察できる確度の高い変更予測を元に変更コストが小さくなる設計を実施するが、決して、開発側の視点に偏った拡張性を追求しない。

再び、品質保証とは？

品質とは、ユーザの満足度である

品質保証とは、製品が「ユーザの満足」を実現することを保証すること

XPにおける品質保証

XPにおけるVerification（検証） & Validation（妥当性確認）

Verification（検証）

- プラクティスが適切に実行されているかどうかを検証する

Validation（妥当性確認）

- イテレーション毎に、計画ゲームにおいて実装対象となったストーリーに従っているかどうかを確認するためにソフトウェアを評価する工程

XPにおけるVerification

ウォータフォールプロセスにおける考え方

検証の手段：品質メトリクスの実績と指標との差異分析による評価

評価対象となる品質メトリクスと実際の品質（ここでは要求を満たすこと）の相関関係を定義し、評価の根拠としている。

基本はXPでも変わりなし。ただし、XPでは「プラクティスが正しく行われていること」を評価するためのメトリクスを収集、評価する必要がある。

例）TDD：実装コードとテストコードの割合
テスト実行によるカバレッジ率 etc.
リファクタリング：コードクローンの測定結果
ペアプログラミング：ペアプログラミングでのコード作成率
ペアの組み合わせの妥当性 etc.

XPではイテレーション毎に検証する（イテレーション内では検証しない）

最大のリスクは「品質を満たさないソフトウェアが完成すること」

↓

リスク対策として開発を短期のイテレーションに分割し、イテレーション毎の仕様を計画ゲームでユーザと合意する。

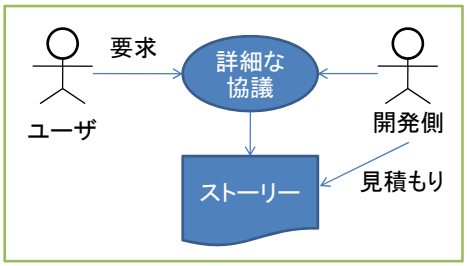
↓

イテレーションの検証の結果、不適切な部分については、次イテレーションでの改善項目としてフィードバックし是正を行う。

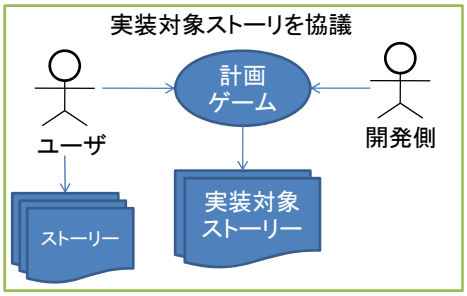
XPにおけるValidation

ユーザの満足が実現できるか？

①ユーザは要求を開発側に提示し、開発側との詳細の協議を経て、ストーリーを作成する。
ストーリーの粒度は小さくなるようにする。（ただし、あくまで要求レベル）



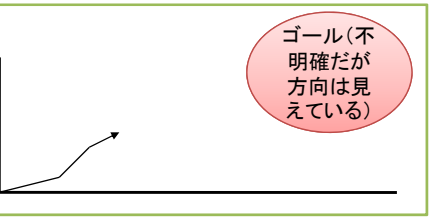
②計画ゲームによりイテレーションで実装するストーリーを決定する。この時、ユーザと開発側で「このストーリーを満たすことで最終的なユーザ満足に近づく」ことを共有する。



③ユーザは受け入れテストを実施し、計画ゲームで決定したストーリーをソフトウェアが満たしていることを確認する。この確認によって。最終的なユーザ満足に近づいた」ことをユーザと開発側が共有する。

XPにおける妥当性確認

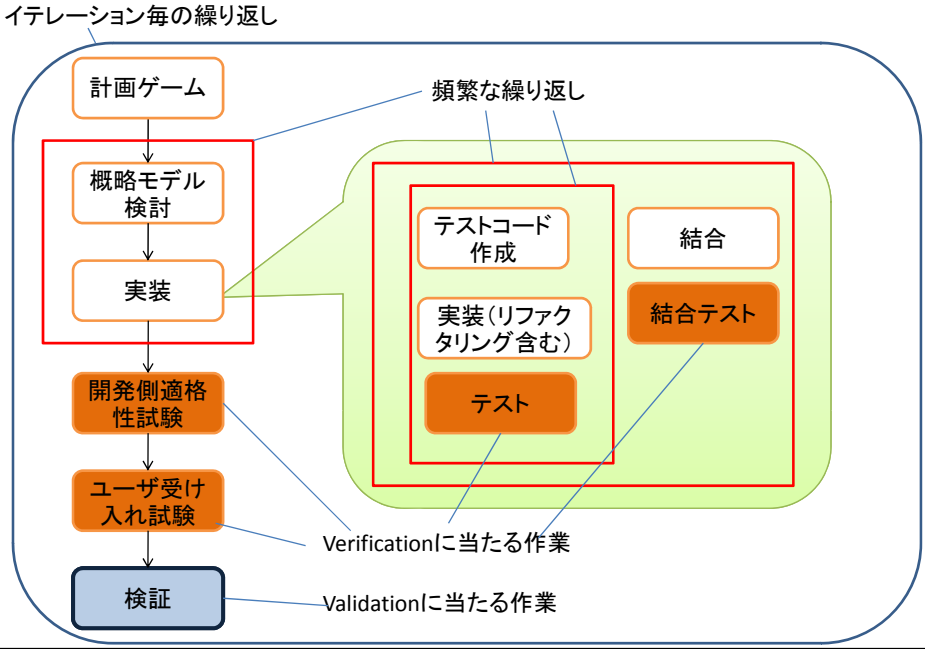
ユーザと開発側が協調し、ゴールへの方向性が正しいかをイテレーション毎に共有認識を持ちながら進める



ユーザにとって満足する方向で進んでいることは、これまで説明したプロセスで評価可能。

これでは、適格性確認にあたる部分しか妥当性を確認できていない？
→XPでは、「TDD」「常時結合」によって適格性確認以前の妥当性を常に確認しながら開発している。

XPにおける品質保証概念



まとめ

- 標準的なXPにおける考え方を示したが、いつも同じ方法が良いわけでなない。
- 例えばイテレーションが3か月の開発であれば、イテレーション内のフェーズを明確にして、フェーズ毎の検証が必要となる。
- 大切なのは、開発プロセスで変わらない「品質を保証する」という本質を捉えて、開発プロセス毎に達成のための手段を考えること。画一的に扱うべきではない。
- 本質が判れば、新しい開発プロセスや、特殊な状況に対応して、品質保証のプロセスを改善していくことができる。

ふりかえり

- アジャイルプロセスはユーザと開発側が協調して共通のゴールに向かう開発プロセスであることを説明しました。
- 品質保証とは、「ユーザの満足」を保証することであると説明しました。
- ウォータフォールプロセスの品質保証の考え方としてV & Vについて説明しました。
- XPの概要と、XPにおける品質保証の考え方を説明し、本質的な意味ではウォータフォールプロセスの場合と違いはないことを説明しました。

是非、開発プロセス選択のメニューの一つとしてアジャイルプロセスも加えていただければと思います。

ありがとうございました。

参考文献

- 「ソフトウェア品質保証の考え方と実際」
  （保田勝通　日科技連）
- 「ソフトウェア品質保証入門」
  （保田勝通、奈良隆正　日科技連）
- 「XPエクストリーム・プログラミング入門」
  （Kent Beck ピアソンエデュケーション）