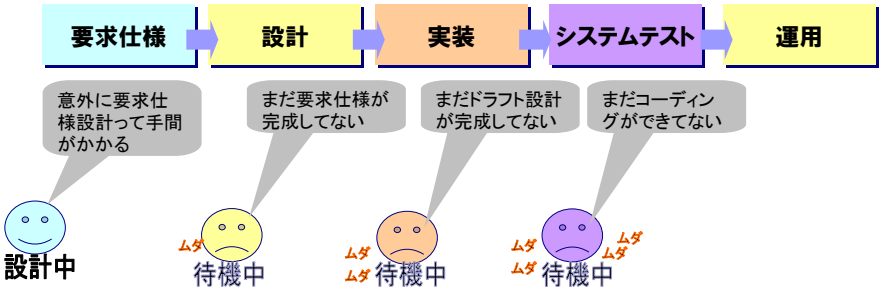
品質保証部門主導型ソフトウェア開発

シスメックス株式会社
小枝　徳晃

目次

- 従来型（ウォーターフォール型）システム開発の問題点
- アジャイル導入による改善
- アジャイル導入の難しさ
- 改善案1）アジャイル型とウォーターフォール型の融合
- 改善案2）改善案1に品質保証を融合した開発システム
- 品質保証部門主導型アジャイルシステム
- まとめ

従来型（ウォータフォール型）のシステム開発の問題点

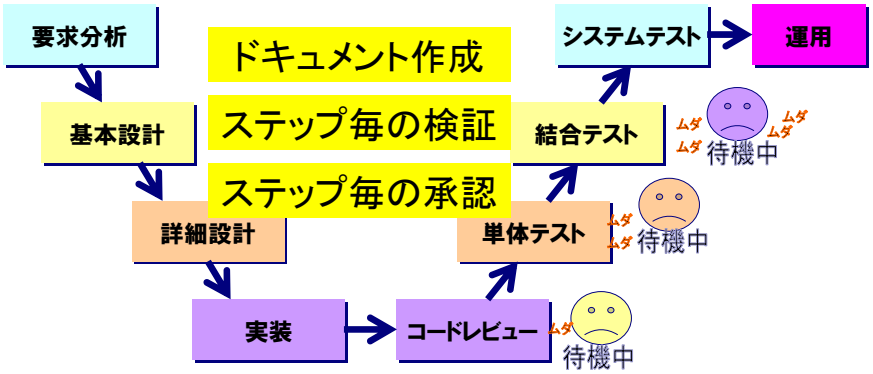


プロジェクトの規模が小さく、同一人物で全ステップをカバーしていれば問題は少ないが、プロジェクトの規模が大きくなるに従い、メンバーの数が増大化→待ちが各ステップで発生

規模が大きくなるほど、工程待ちのムダが発生する

品質向上を目指す余り、増えていく開発ステップ

ソフトウェアの品質向上を目的とした施策を行うことにより、開発ステップはますます増えていく。



★開発ステップが増えるということは、ステップ間の待ちが増えてしまう。★

品質向上の名前のもと、ますます待ちのムダが増える

更に問題が起こりやすいのは･･･システムテスト段階

一般的にシステムテストへの移行段階は、問題が起こりやすい

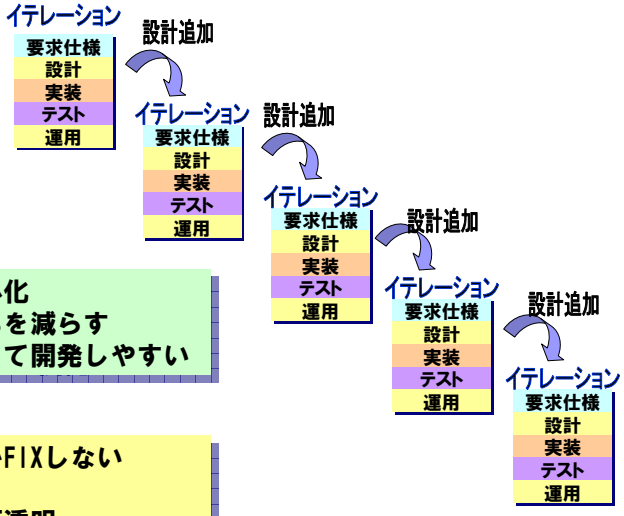
なぜなら、
- プロジェクトの最終段階で、それまでの矛盾がでやすい。
- 遅れ等により、クリティカルパスとなりやすい。

結果、　プロジェクトの成否がかかることが多い

開発ステップ間のギャップを埋めるために、
アジャイル型開発が有効である。

イテレーションに分割追加型で開発する

アジャイルの導入による改善



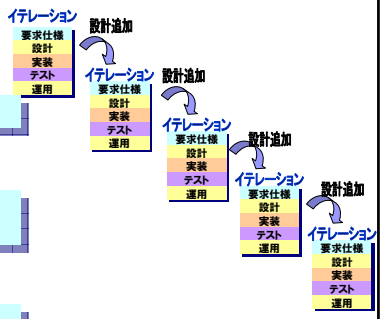
長所

★仕様分割、規模縮小化
　→ステップ間の待ちを減らす
★テストのことを考えて開発しやすい

短所

★要求仕様がなかなかFIXしない
★最終形が見えない
★総工数、総費用が不透明

アジャイル型開発導入の難しさ



■パートナー企業との契約

成果物を固定化できないことが契約で問題になる

■開発フローとの不整合

要求仕様を開発前に確定しなければならない開発規定となっている

■イテレーション毎の評価

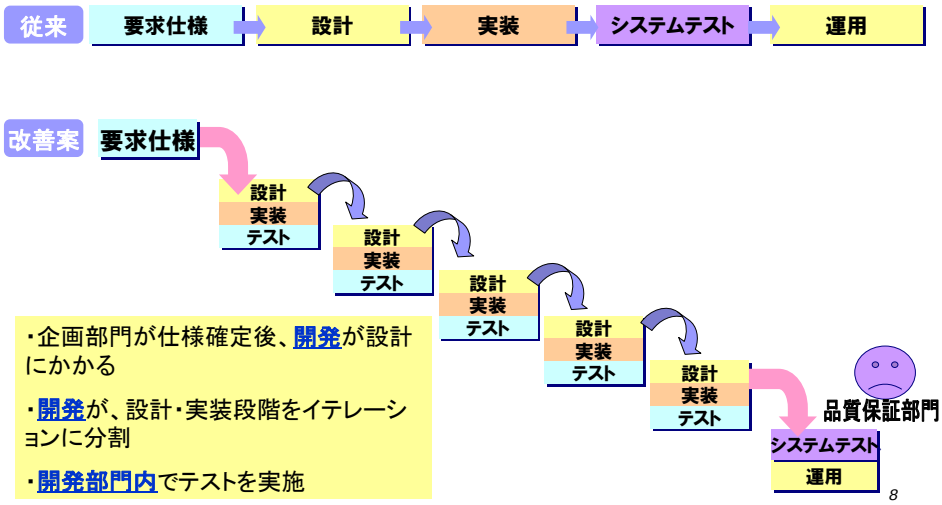
ユーザーレビューが頻繁に行えない。
→開発中は、開発に任されてしまっている。

■開発終了の判断

最終的にリリース可能かどうかの判断を行う根拠が不明確になる

良いことは判っていても簡単には導入できない。

改善案1．アジャイル型とウォーターフォール型の融合



・企画部門が仕様確定後、開発が設計にかかる

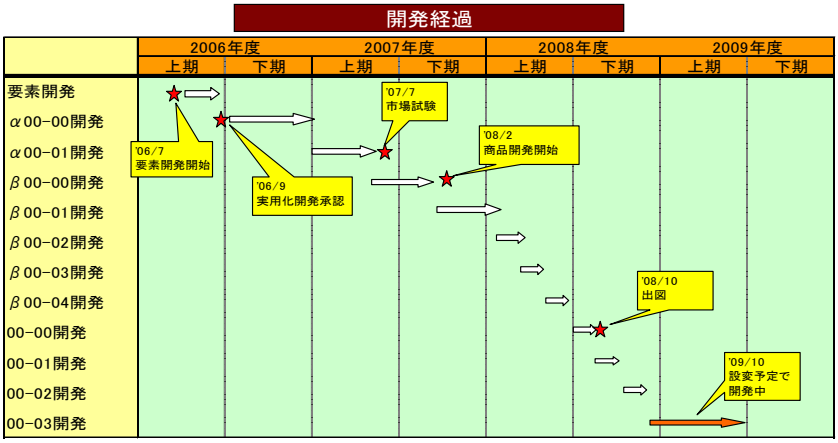
・開発が、設計・実装段階をイテレーションに分割

・開発部門内でテストを実施

弊社組み込みソフトウェア開発事例

- 工業用粉体等の微小粒子の分析装置
- ソフトウェア開発メンバー　最大１０名
- 開発期間　約2年
- 開発ステップ数　本体部　７万行（C++）
  解析部　12万行（C#）

弊社組込ソフトウェア開発事例



弊社開発事例を通して

- 各イテレーション毎に1週間のテスト期間を設定
- 各イテレーションを成果物として契約することで、パートナー企業との契約がしやすい。
- 絶えず発生するハードウェアからの変更要求に迅速な対応が可能となった

改善案1の効果・問題点

効果

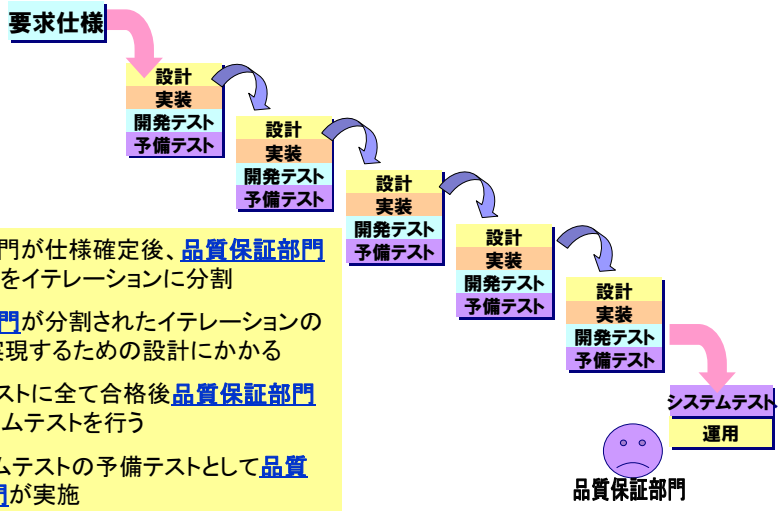
- ウォータフォール型の大きな枠組みを変更する必要なく、改善が図れる
- イテレーション毎の追加型開発のため、仕様追加や修正に応じやすい。
- イテレーション毎の目標が明確になり、開発の進捗状況がわかりやすくなる。
  開発にリズムが出て、予測が正確になる。

問題点

- 各イテレーションの評価が不十分であり、システムテスト段階で不具合が発生しやすい。
- タスクに対する優先度の付け方がプログラム作成側の都合になりやすく、各イテレーションが有効に使われない。

品質保証部門が、開発工程をコントロールすることで解決できないか？

改善案2．改善案1に品質保証を融合した開発システム



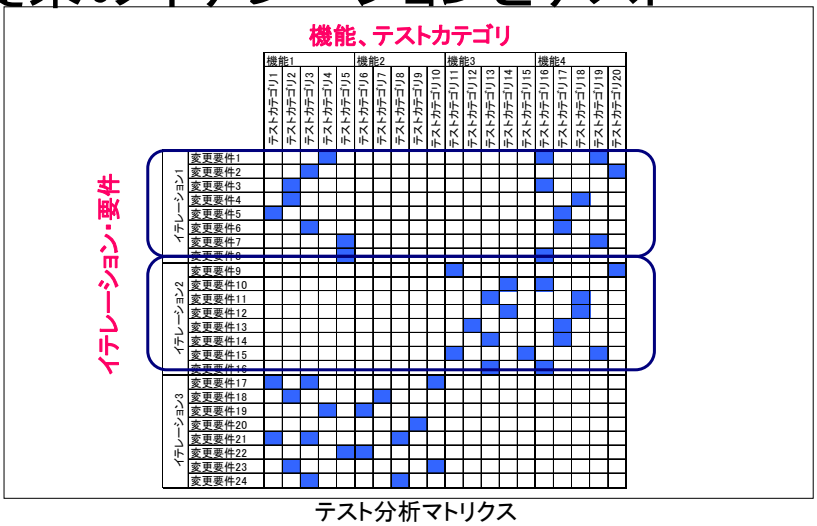
・企画部門が仕様確定後、品質保証部門が、仕様をイテレーションに分割

・開発部門が分割されたイテレーションの仕様を実現するための設計にかかる

・予備テストに全て合格後品質保証部門がシステムテストを行う

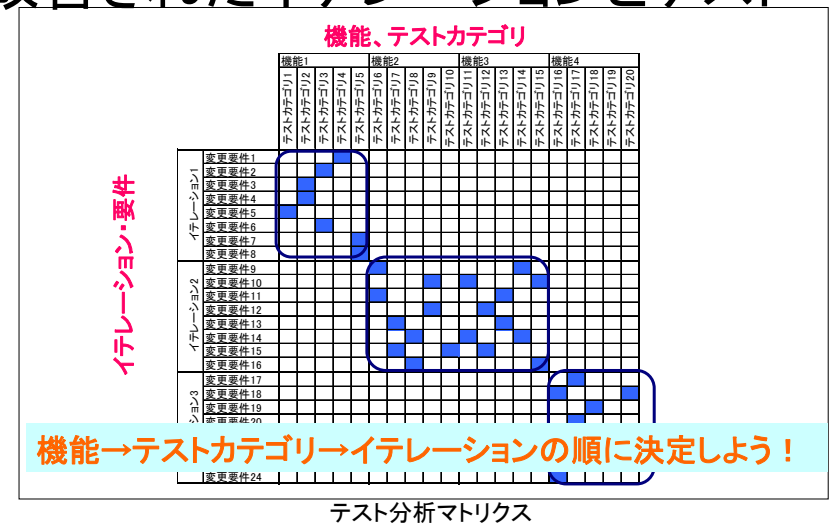
・システムテストの予備テストとして品質保証部門が実施

従来のイテレーションとテスト



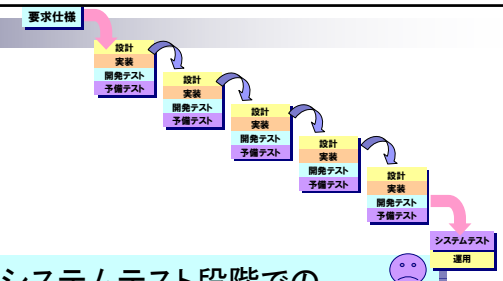
テスト分析マトリクス

改善されたイテレーションとテスト



テスト分析マトリクス

改善案2に期待できる効果

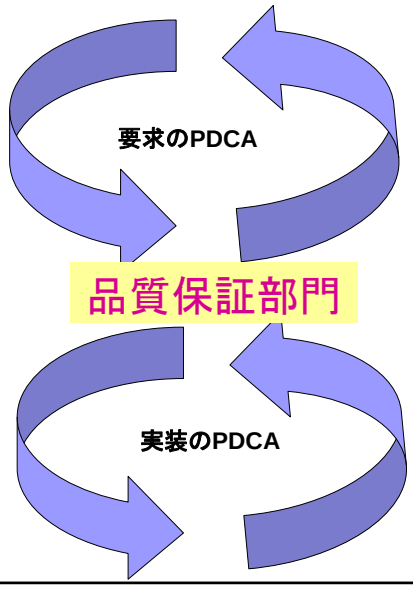


効果

- 予備テストの実施により、システムテスト段階での不具合が減少する
- 予備テストにより、テスト仕様上の問題が事前に修正でき、テストの品質が上がる
- 各イテレーションの品質が上がり、ユーザー等へのデモを実施できる
- システムテスト段階でのムダが減少し、効率化が図れる

やはり、品質保証部門が、開発工程をコントロールすることが効率化を産む

改善案3. 品質保証駆動開発システム



要求のPDCAと実装のPDCAを分けることが有効では？

そして。。。

品質保証部門が要求のPDCAと実装のPDCAを**繋ぐ役割**を持つ！

まとめ・今後の課題

- 従来のウォーターフォール型開発とアジャイル型開発を融合させることにより効率的な開発ができることが実証できた。
- 更にそれを発展させ、品質保証部門が設計、実装段階をコントロールすることが、開発の効率化に有効であると考えられる。
- 今後、要求のPDCA、実装のPDCAを品質保証部門が中心に行なうことにより、さらに効率的な仕組みを作っていきたい

- ご清聴ありがとうございました。