TSPの俊敏性向上に関する取り組みと評価

九州工業大学

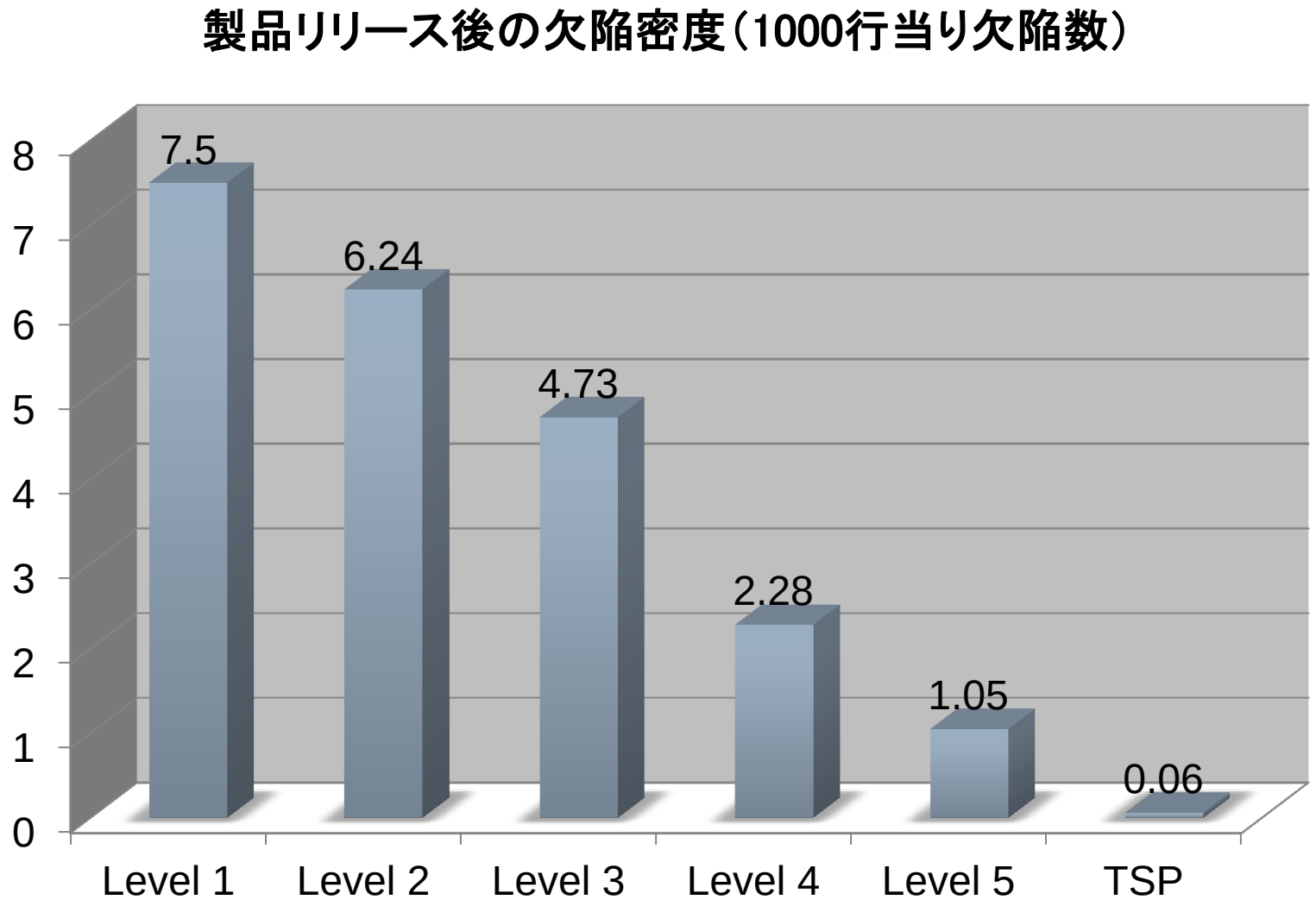
○片峯 恵一　梅田 政信　荒木 俊輔　橋本 正明

e-mail：katamine@ci.kyutech.ac.jp

ソフトウェア開発の課題

- 大規模化
  - 複雑化
  - 多種多様化
  - 高品質
- 短納期化
  - 要求変更への対応
    - 要求不備を前提とした開発
  - 派生開発
- チーム
  - メンバー変更への対応

CMMとTSPの品質比較



Ref.: CMU/SEI-2003-TR-014

ソフトウェア規模による開発手法の有効性

	小規模 (< 1000FP)	中規模 (1000-10,000FP)	大規模 (> 10,000FP)
1	Agile	TSP/PSP	TSP/PSP
2	TSP/PSP	Agile	CMM 3,4,5
3	Waterfall	CMM 3	RUP
4	CMM 1,2	RUP	Hybrid

Ref.: Software Engineering Best Practices, Capers Jones, 2010

TSPアジャイル化方程式

- ✕ TSP = Agile ?
- ✕ substitute(TSP, ”cycle”, “sprint”) = Agile ?
- ✕ TSP – ProcessData = Agile ?
- ✕ TSP – Process – ProcessData = Agile ?

→ 引き算によるAgile化

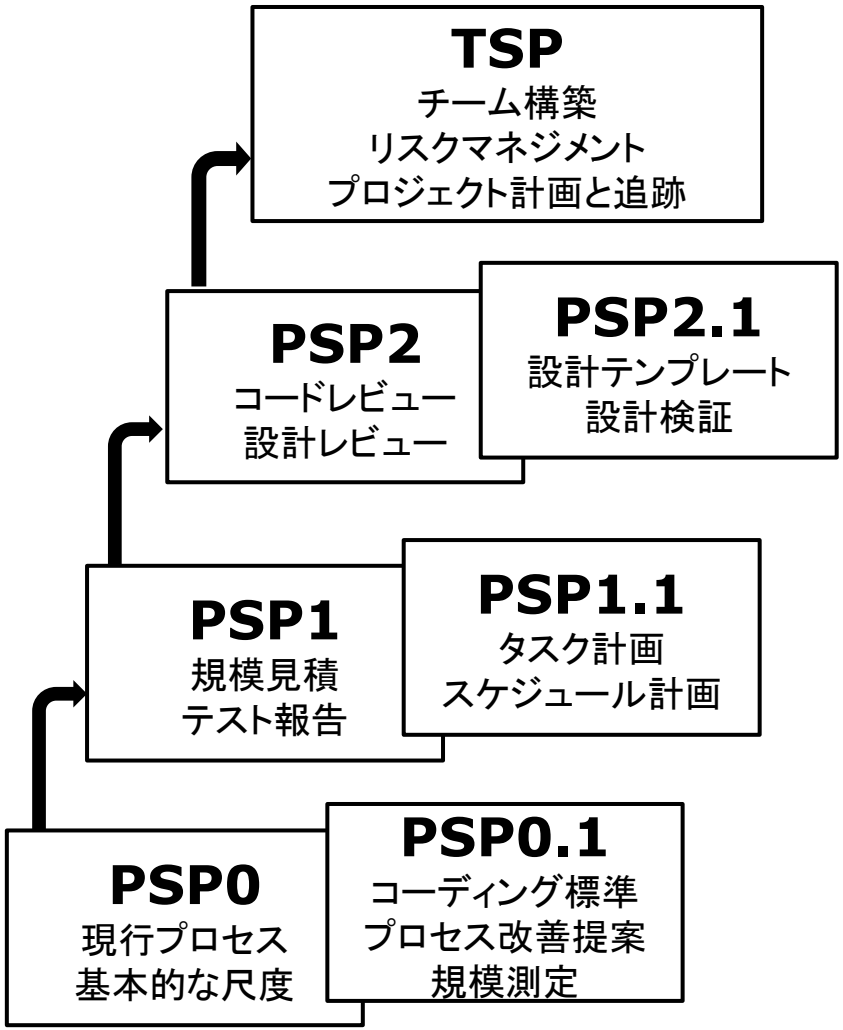
- ✓ TSP + Agility = AgileTSP

→ 足し算によるAgile化

研究の目的と方針

- 目的
  - 俊敏で高品質なソフトウェア開発手法の構築
- 方針
  - 品質重視
    - PSP/TSP
  - 俊敏性を向上させる方法
    - 開発戦略の修正
    - ベストプラクティスの導入
  - トライアル
    - 大学院のTSPコース＆PBL
    - PM学会九州支部SPI-WG

PSPプロセス



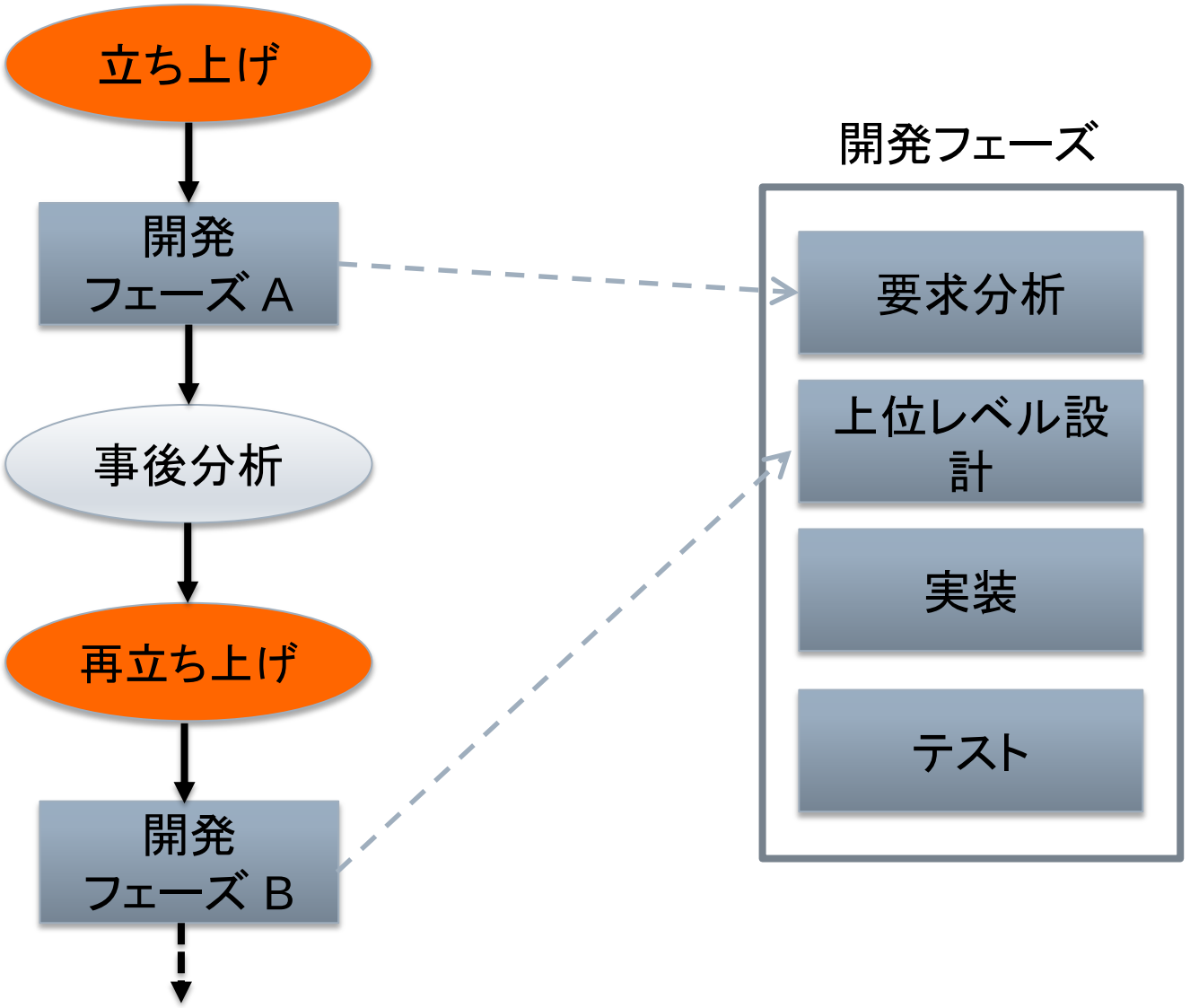
品質マネジメントと
設計標準の導入

見積もりと計画の導入

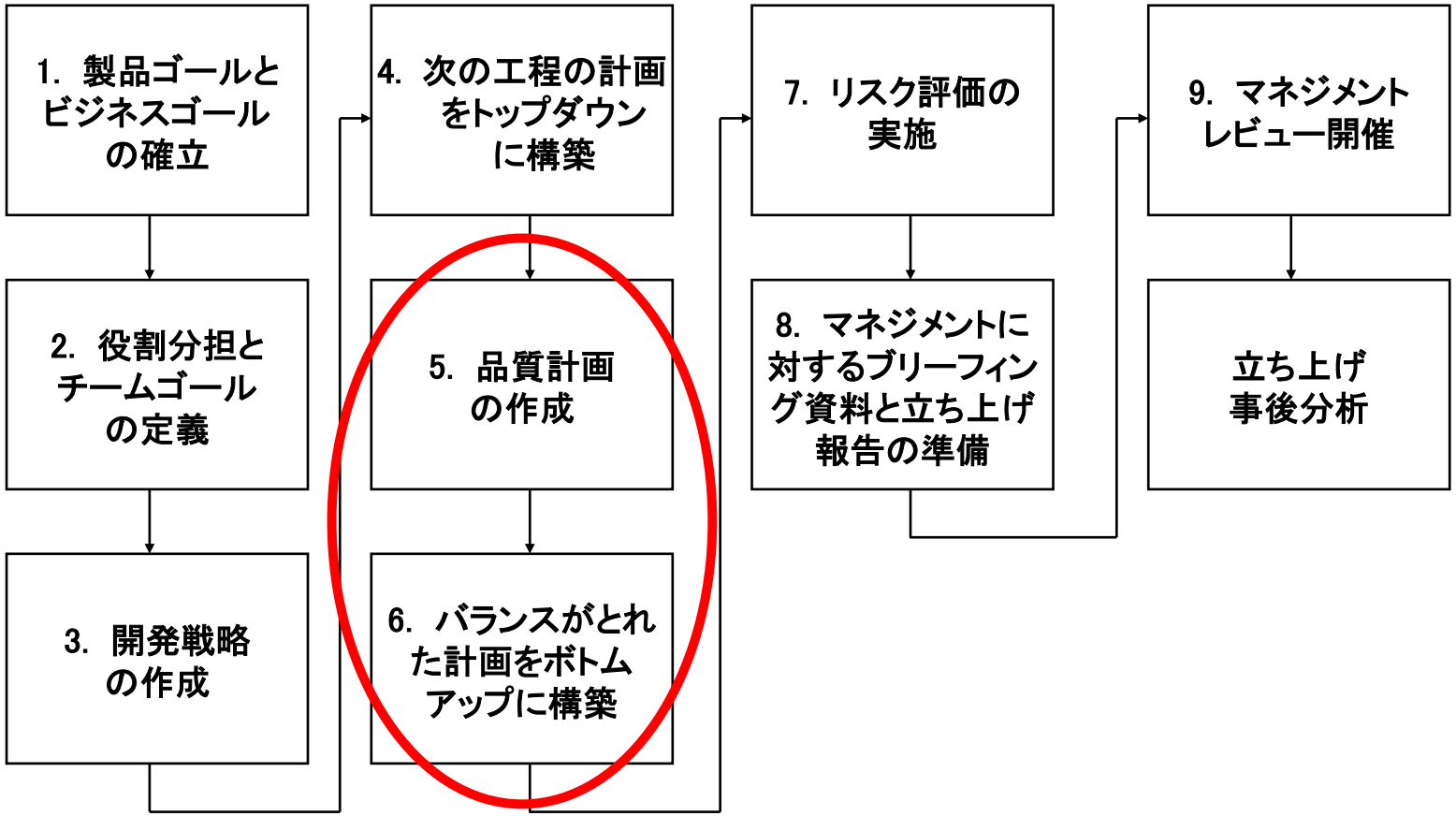
プロセス規律と
測定の導入

Ref. PSPガイドブック：ソフトウェアエンジニア自己改善

TSPプロセス



TSPの立ち上げプロセス



Ref. TSPガイドブック：リーダー編

TSPチーム

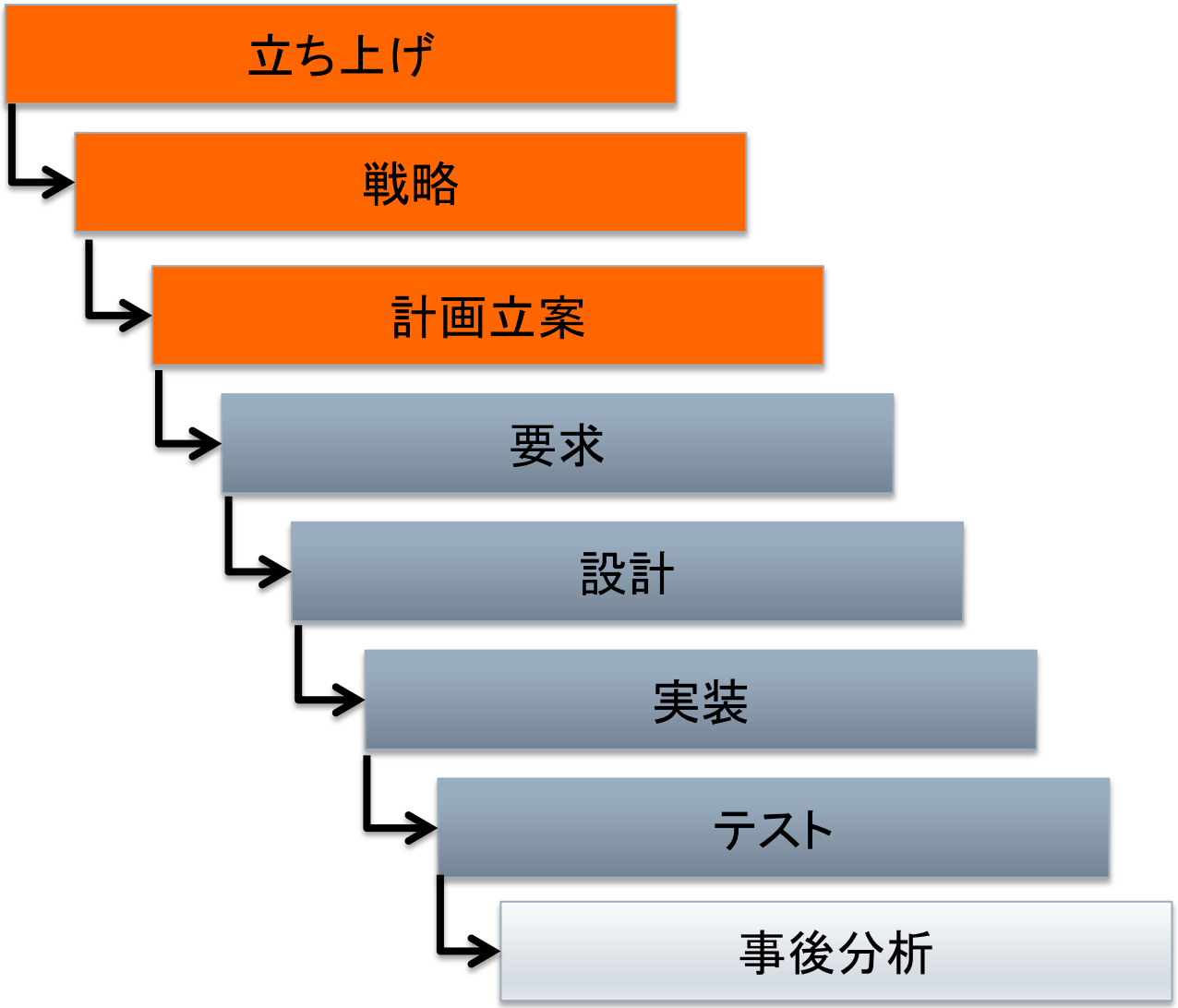

プロジェクトマネージャ
プロジェクトメンバー

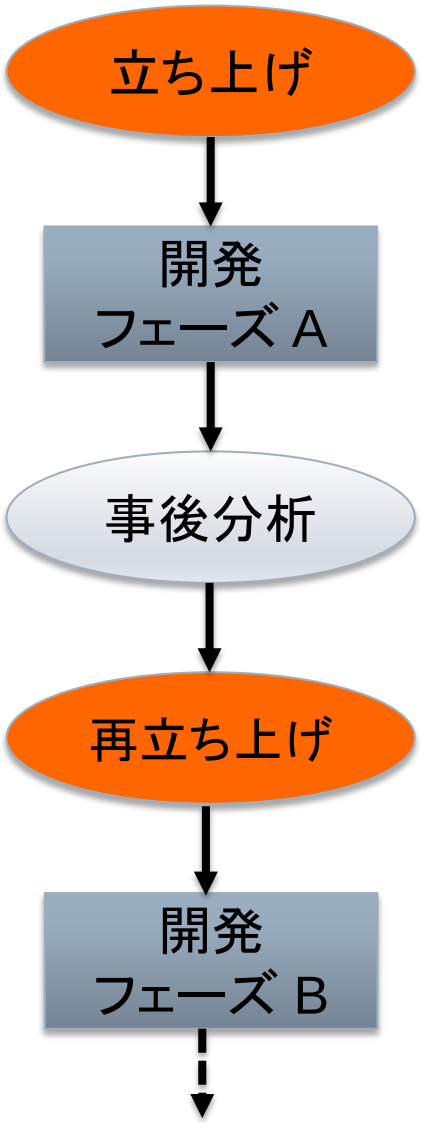
TSPコーチ
チームリーダー
サポートマネージャ
顧客インタフェイス
マネージャ
プロセスマネージャ
設計マネージャ
計画立案マネージャ
実装マネージャ
テストマネージャ

TSPの俊敏性向上に関する取り組み

- TSPi プロセスを採用
  - プロセスの追加・変更なし
- 開発の戦略基準を変更
  - 開発サイクル数の増加
  - 戦略基準の追加
- 要求変更
  - 第2サイクル後

TSPiとTSPの違い





戦略基準の変更

- TSPiの基本戦略
  - 循環プロセスの使用（2～3サイクル）
  - *サイクルの開発規模を小さくし、3サイクル以上に*

- 推奨する戦略基準
  - サイクル1で開発する製品は、最終製品のサブセットで最小の機能を提供する。
  - サイクル1で開発する製品は、容易に拡張できるベースを提供する。
  - サイクル1で開発する製品は、すべてが高品質であり、簡単にテストできる。
  - 製品の設計は、チームメンバーが個々に作業できるようにモジュール構造にする。
  - *各サイクルで開発する製品は、顧客が業務に部分的に利用し、要求の妥当性を確認できる機能を提供する。*
  - *製品の設計は、機能の追加、変更があることを前提とし、これに容易に対応可能となるようなアーキテクチャーとする。*

大学院生チームによるトライアル

- プロジェクト
  - 変更カウンタの開発
    - 簡易バージョン管理システム
    - TSPiテキストの機能要求記述書を使用（Ref. TSPiガイドブック）
- 開発期間
  - 9 月末から 2 月上旬の約 5 ヶ月弱
  - 一人当たり週 5～10 時間程度
- チーム
  - 3名 x 2チーム
  - 複数の役割を兼任

実施結果

- 開発サイクル
  - 第1サイクル：8週間、12週間
  - 第2～第4サイクル：3～6週間
- システムテスト欠陥
  - 5サイクルで0個
  - 1サイクルで1個
- プロセス品質指標(PQI)
- サイクルの時間配分

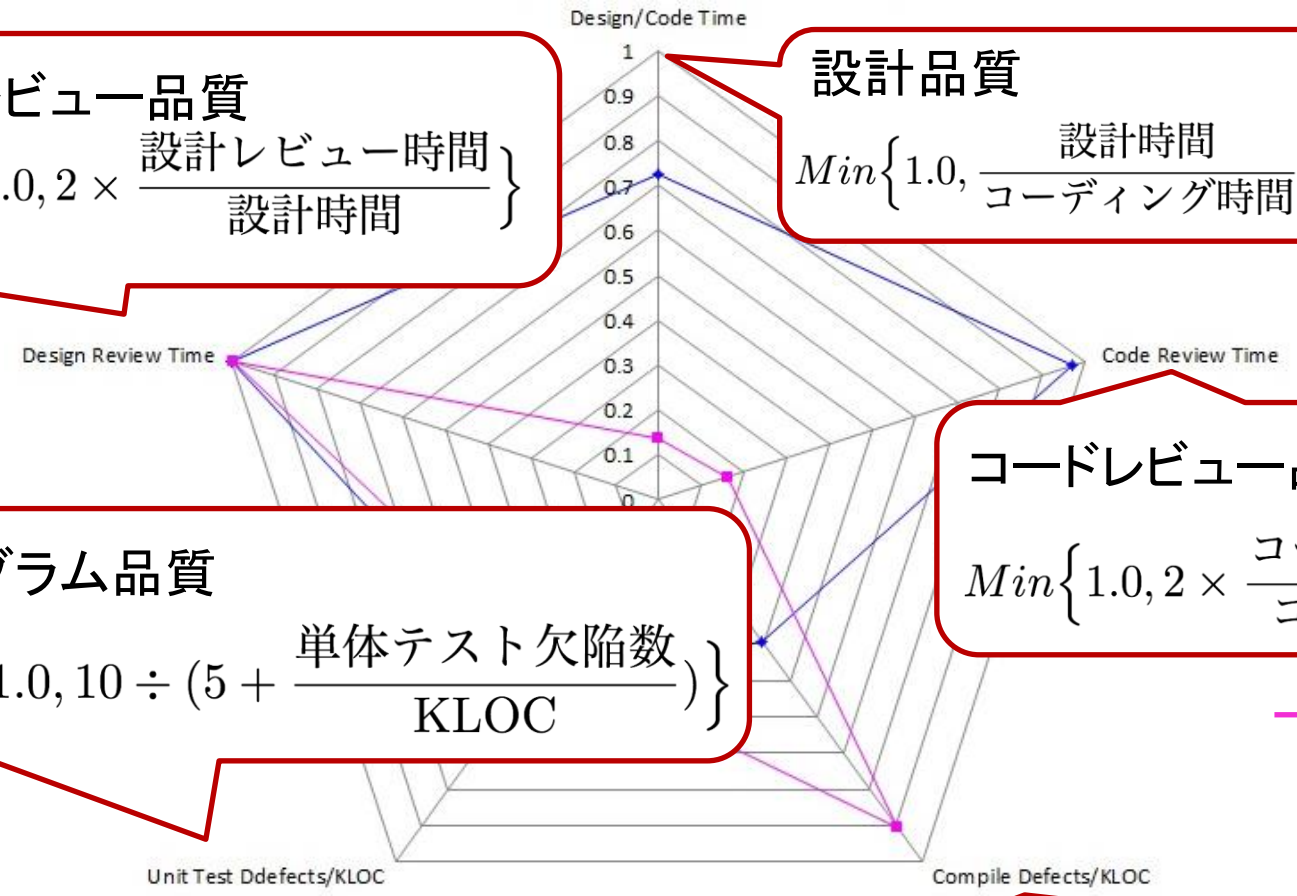
プロセス品質指標PQI

設計レビュー品質

$$Min\left\{1.0, 2 \times \frac{\text{設計レビュー時間}}{\text{設計時間}}\right\}$$

設計品質

$$Min\left\{1.0, \frac{\text{設計時間}}{\text{コーディング時間}}\right\}$$



コードレビュー品質

$$Min\left\{1.0, 2 \times \frac{\text{コードレビュー時間}}{\text{コーディング時間}}\right\}$$

プログラム品質

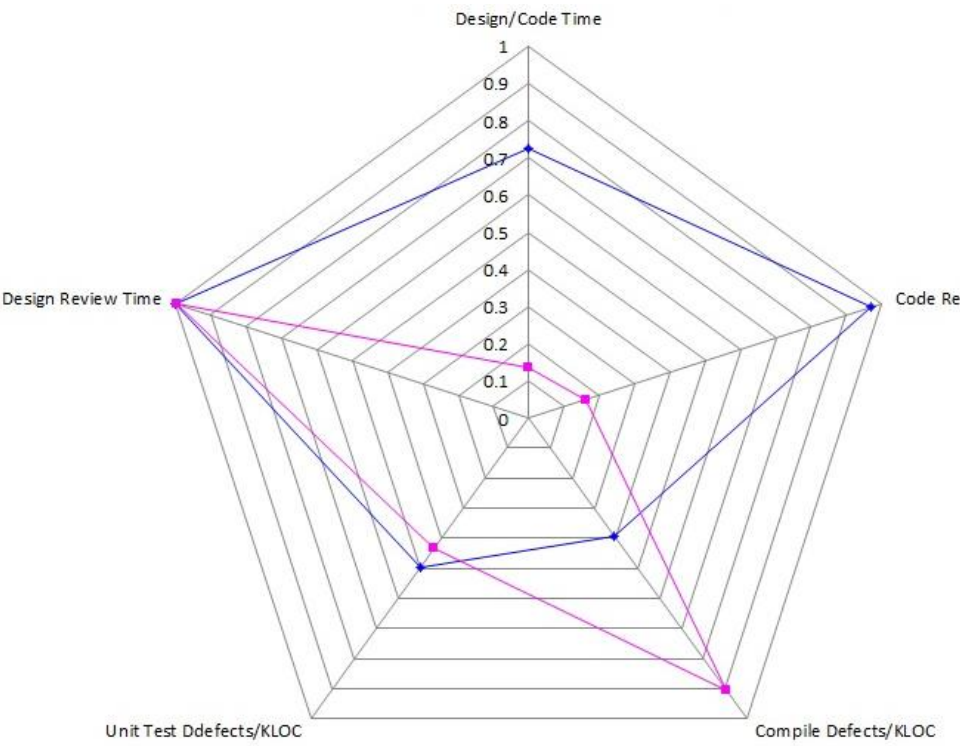
$$Min\left\{1.0, 10 \div (5 + \frac{\text{単体テスト欠陥数}}{\text{KLOC}})\right\}$$

サイクル1
PQI = 0.0087

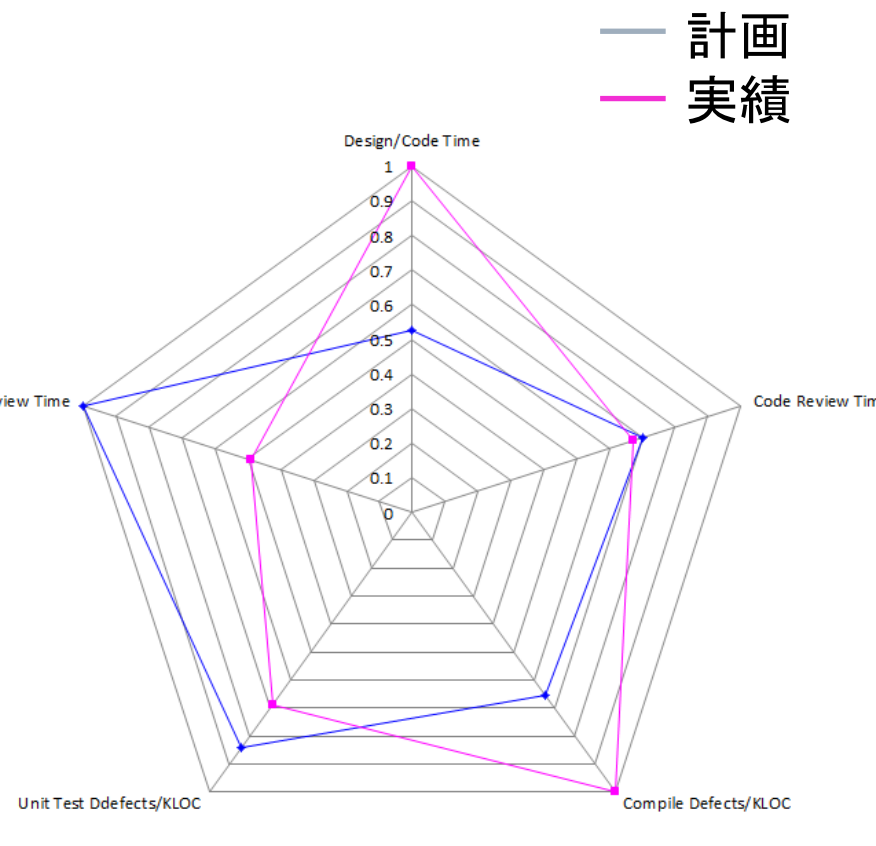
コード品質

$$Min\left\{1.0, 20 \div (10 + \frac{\text{コンパイル欠陥数}}{\text{KLOC}})\right\}$$

PQIの推移（サイクル1 → サイクル2）

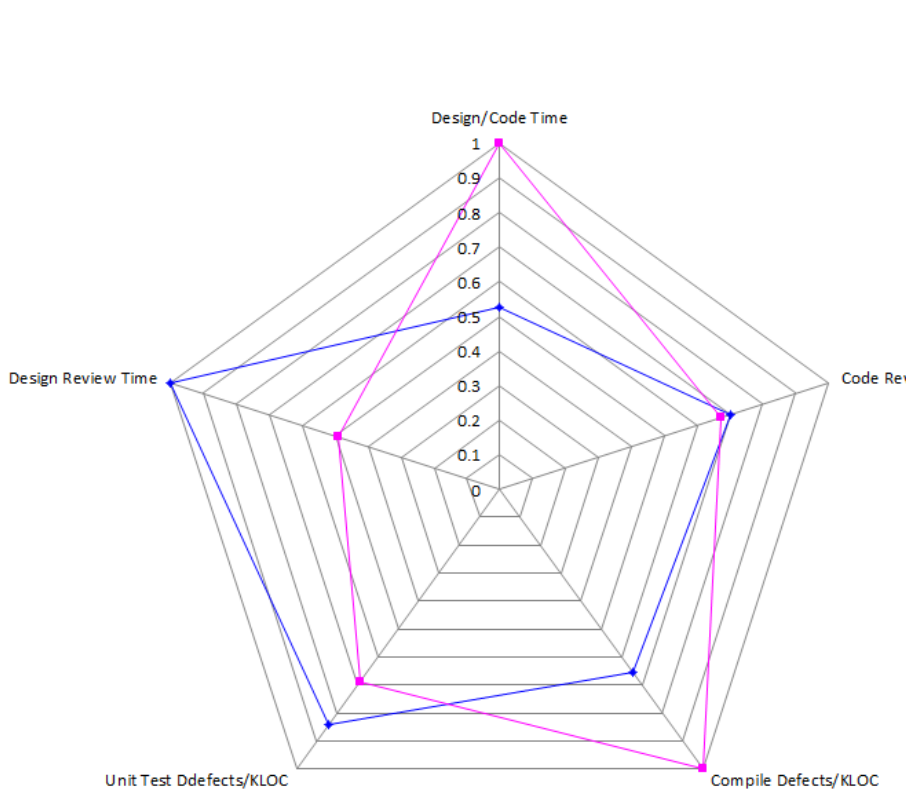


サイクル1
PQI = 0.0087

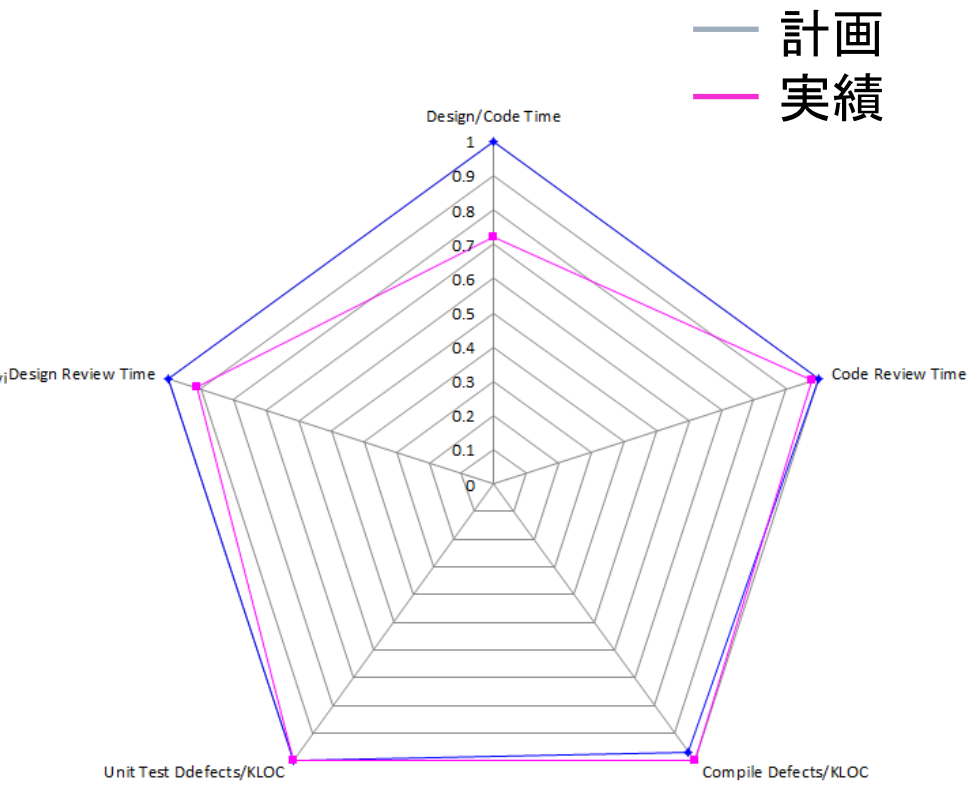


サイクル2
PQI = 0.2277

PQIの推移（サイクル2 → サイクル3）

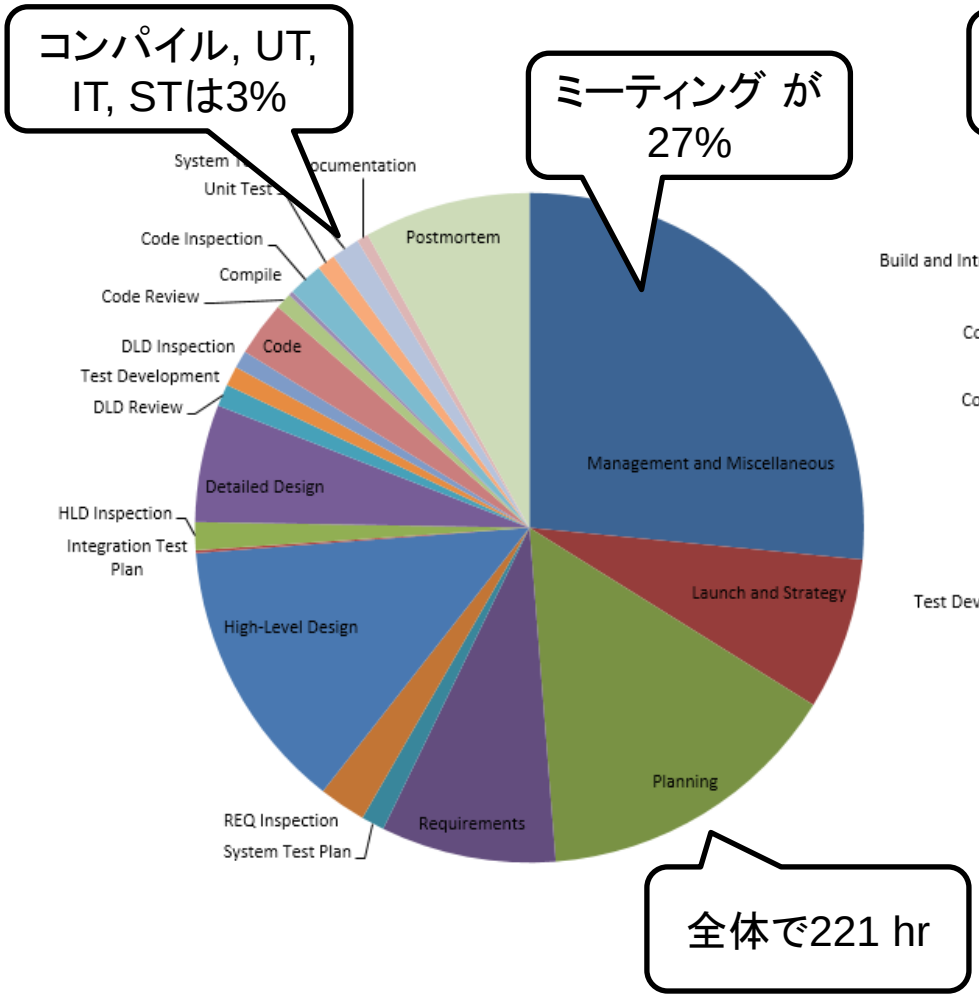


サイクル2
PQI = 0.2277

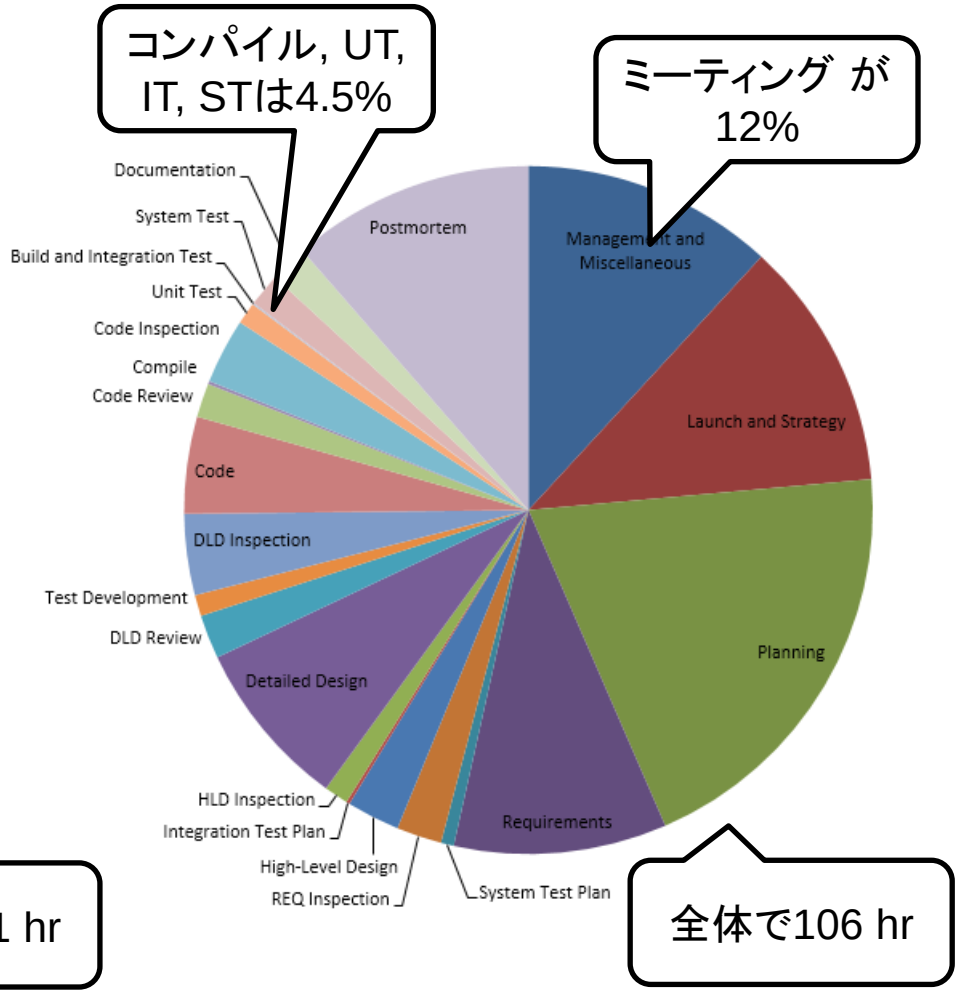


サイクル3
PQI = 0.6411

チームAの開発時間配分 -1

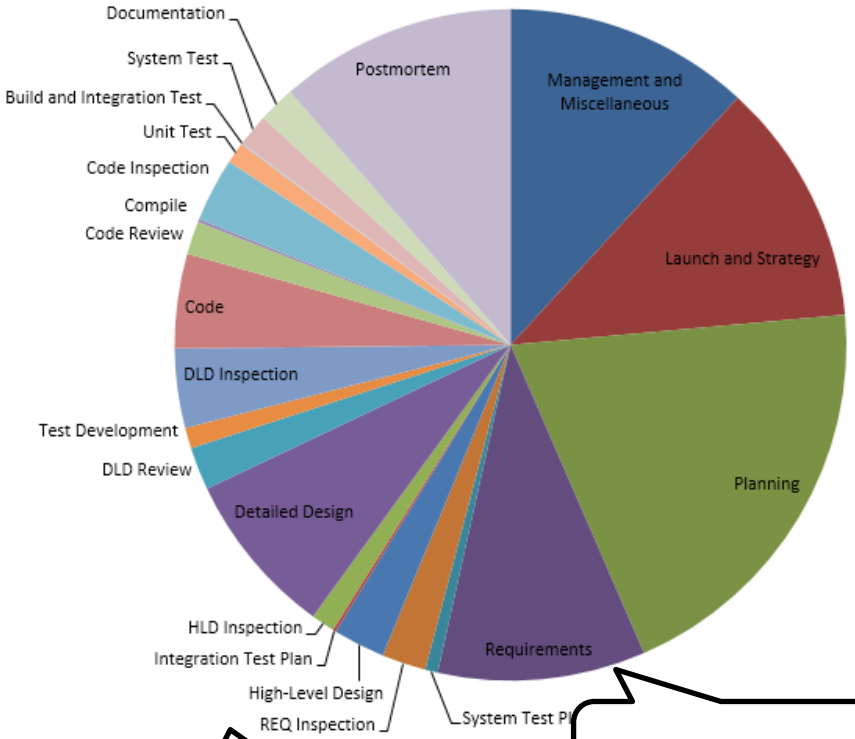


サイクル1



サイクル2

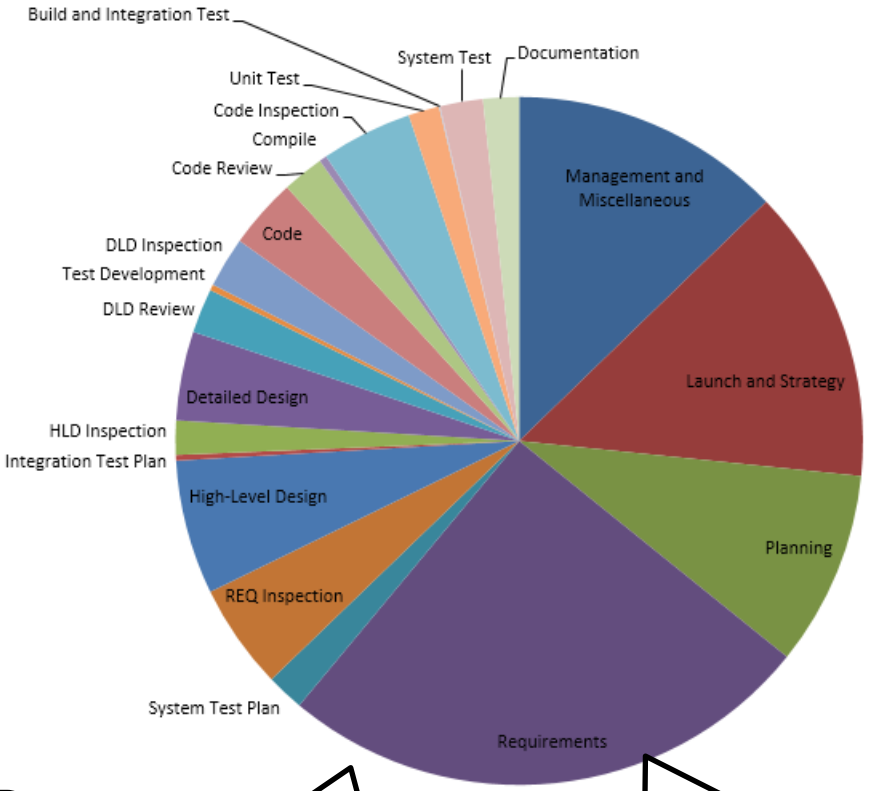
チームAの開発時間配分 -2



全体で106 hr

要求分析10%

サイクル2



全体で
57 hr

要求変更による
修正含め22%

サイクル3

実施結果 -2

- 改善活動
  - 品質の改善
    - レビュー品質（速度、方法など）
    - チェックリストの更新
  - リスク項目の洗い出し
    - 他講義（レポート）
    - 学会等の発表
    - 年末年始（休暇、行事）への対応
  - フェーズミーティングの導入
    - 週次ミーティングでは不足
    - 学生が自ら発想

考察

- 教育的な観点
  - サイクル増加
    - 自己改善活動の機会増加
  - 品質
    - 品質重視の考え方
      - レビュー、インスペクション
    - データに基づく継続的改善の意義

- 開発の観点
  - 小規模プロジェクトでもTSPiは有効
    - 60 ～ 220 人時
  - 要求変更への対応
    - エンジニアリング能力に依存
    - アーキテクチャ設計における指針等が必要

まとめと今後の課題

- TSPにおける俊敏性向上の取り組み
  - TSPiプロセス
  - 開発戦略
  - 要求変更
  - 大学院生チームへの適用と成果
- 今後の課題
  - 顧客参加における影響の分析
  - ベストプラクティスの選択と導入
  - マネジメント＋エンジニアリング
  - 適用範囲の拡大