

SEPG Japan

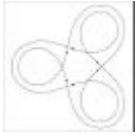
SPO : ソフトウェアプロセス最適化

組織のソフトウェアプロセスを客観的に評価することは、ソフトウェアプロセスを変更するためにある。しかし、評価という手段が目的化している。この現状に対して、最適化によるプロセスの変更のための枠組みを提案する。



株式会社ニルソフトウェア

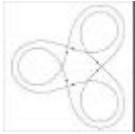




Agenda

- ✂ 動機
- ✂ プロセス最適化
- ✂ 考慮すべきポイント
- ✂ SPOフレームワーク
 - ✂ ベースモデル
 - ✂ プロセス技術
 - ✂ 技術的管理
 - ✂ ソフトウェア工学
 - ✂ 対自的活動
- ✂ 素朴な応用例
- ✂ まとめ



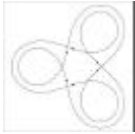


SPOの動機

- ✖ 実際的な問題がある。
 - ✖ ユーザの終わりになき要求変更と追加
 - ✖ 二直を前提としたスケジュール
 - ✖ ユーザがすぐ気が付く低品質と低い顧客満足度
 - ✖ コストの超過

- ✖ 以下の対処策は正しいか。
 - ✖ 要求管理の実施
 - ✖ プロジェクト計画の実施
 - ✖ プロセスとプロダクトの品質保証
 - ✖ プロジェクトのモニタリングと制御

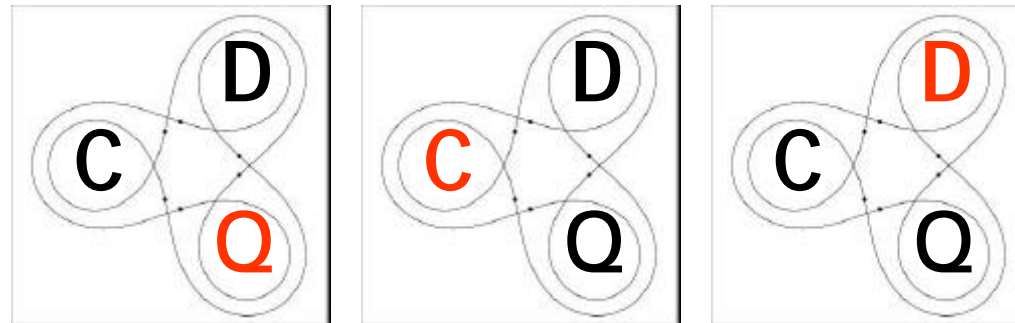
- ✖ 現象は複数の要素の組み合わせとなる。



SPO : プロセス最適化

- ある組織ないしはあるプロジェクトにおいて最適のプロセスを考える。

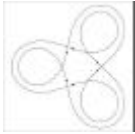
コスト(c)と品質(Q)と納期(D)は与えられたゴール中でトレードオフの関係にある



- ゴールが一定として -

- 止揚するには、何れかの手段（プロセス，手法，ツール）を取らなくてはならない。

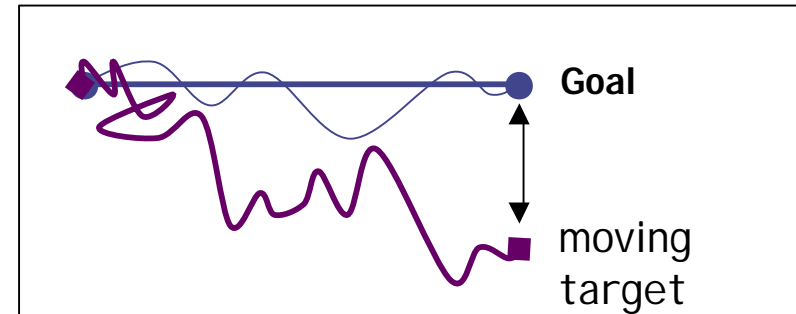
問題とその関係に対する気付き，重み付けの変更による広域最適化の予測



考慮すべきポイント

👁️ 視点

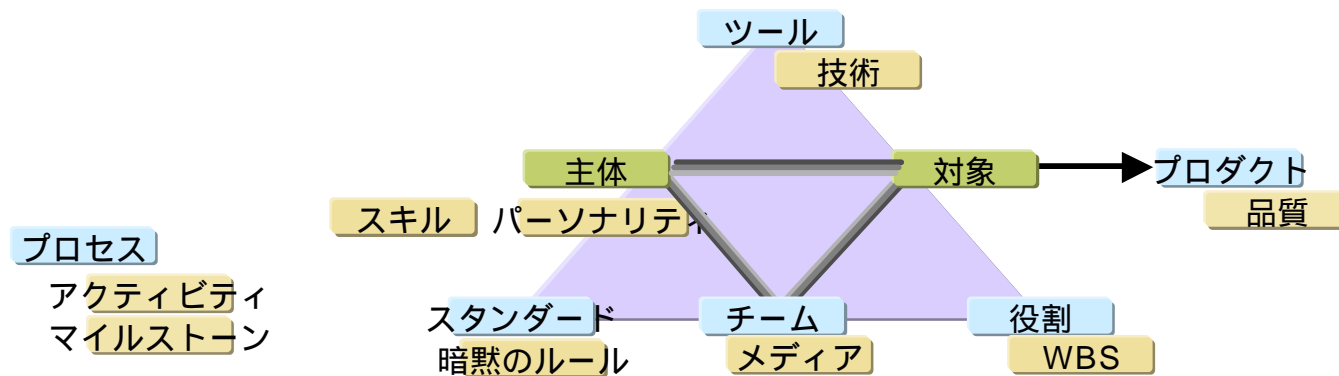
- 🗑️ ゴール
- 🗑️ 時間
- 🗑️ 組織

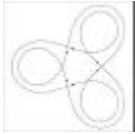


👁️ 最適化のためのバランスされる要素

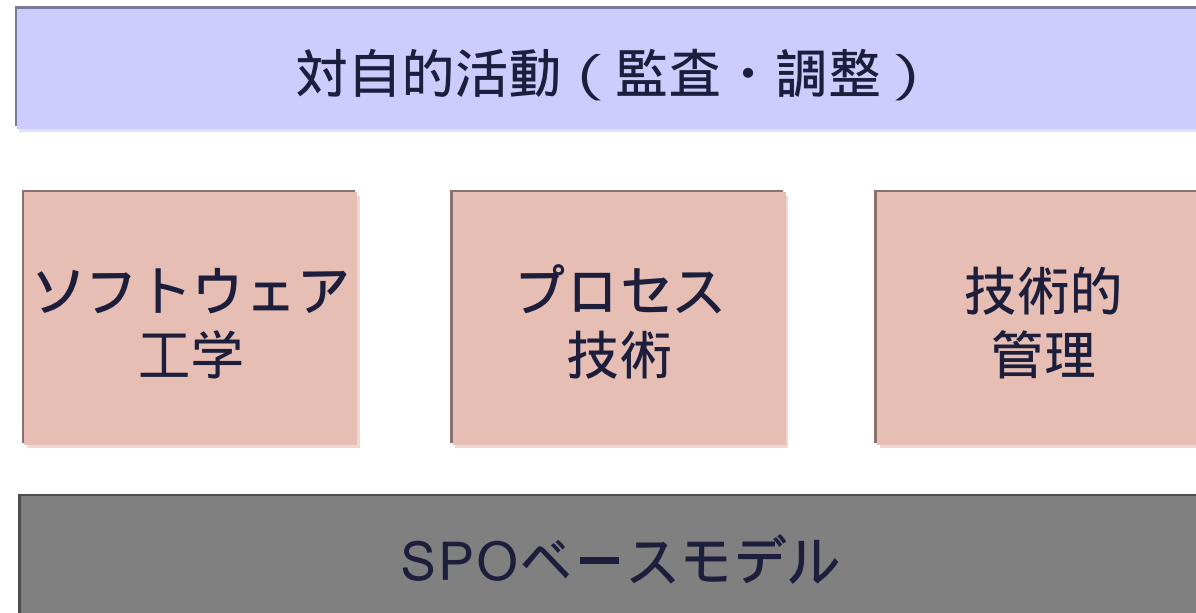
🗑️ QCD + G

👁️ Q : Quality C: Cost D: Delivery G: Goal

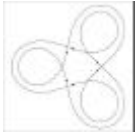




SPOフレームワーク



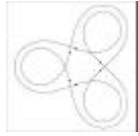
SPOとは、ソフトウェアを作るプロセスを、複数の関与者がそれぞれ見直すことによって、QCDを向上させるために対象に働きかける活動全般を指す。



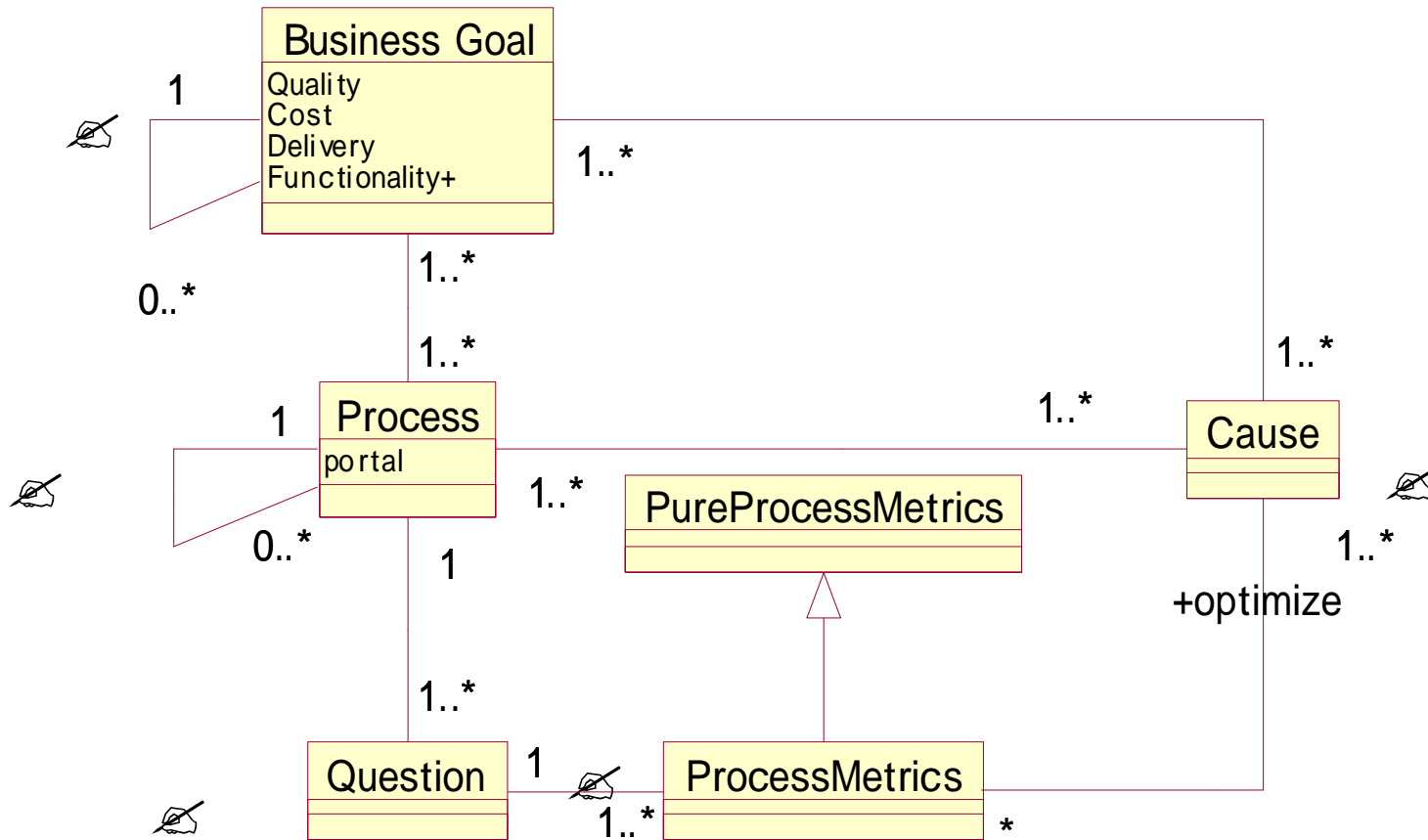
SPOベースモデル

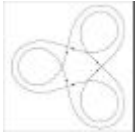
- ✂ 制御モデルを用いる（SPI：工場メタファ）
 - ✂ TQC
 - ✂ PDCA
 - ✂ シックスシグマ
 - ✂ D（問題定義）+ MAIC（測定・分析・改善・管理）
 - ✂ CMM
 - ✂ I（初期-Initiating）+ D（診断-Diagnosing）, E（構築-Establishing）, A（実行-Acting）, L（学習-Learning）

- ✂ 変化を予測する（最適化：山登り）
 - ✂ リスクを道しるべとする。
 - ✂ リスクの識別 / 評価
 - ✂ リスクの分析と解決策の検討
 - ✂ リスクの追跡
 - ✂ リスク情報のプロセスリポジトリへの蓄積



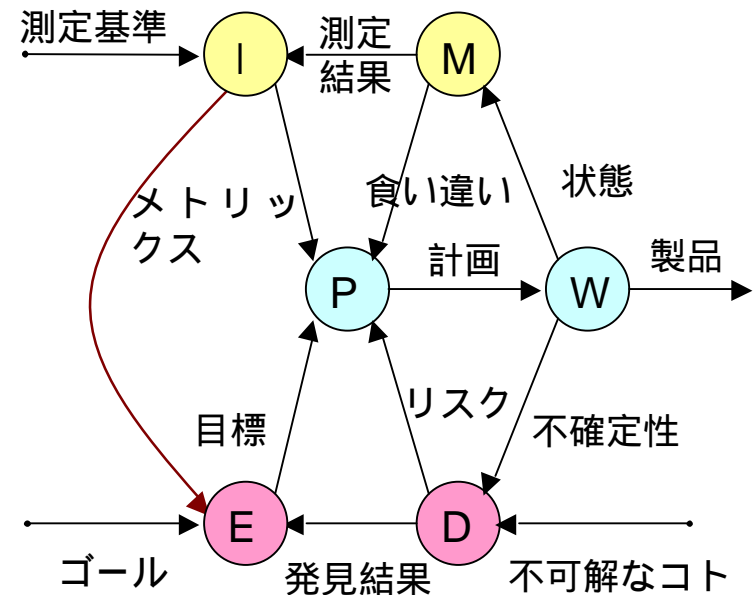
概念モデル





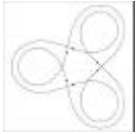
プロセス技術

- プロセス定義
 - モデル
- プロセス実行
 - 実行 (enaction) 機構
 - 変更は .
- プロセス計測
 - プロダクト計量との違い
 - GQM, FCM QFD, ami approach
- 欠けているもの
 - プロセスメトリクスを用いた気づきと予測



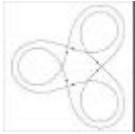
【凡例】 P: 計画, W: 作業, M: 計測, I: 向上活動, E: 洞察, D: 発見

Hall, E. MANAGING RISK, Addison-Wesley



技術的管理

- ✂ (技術的な裏づけを持つ) 管理或いは技術面での管理 (≠リソース管理)
- ✂ 例: スケジュール管理における一般的な方法
 - ✂ PERT記述 (WBS記述を含む) Ganttチャート
 - ✂ しかし,
 - ✂ 基本は基本としてバリエーションが必要
 - ✂ BOX法, 制約論理法, バックログ法
 - ✂ システムダイナミックス
 - ✂ プログラマはマネージャになり得るか.
 - ✂ 構成管理の例を考えよ.
- ✂ 欠けているもの
 - ✂ リスク主導のアプローチ (プロセス技術に同じ)



参考：BOX法（視覚的な管理のために）

開始前

機能番号及び機能名			
計画開始日		計画終了日	
実開始日		実終了日	
担当者			
設計	実装	単体	レビュー
その他			

予定通り実装まで

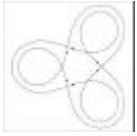
機能番号及び機能名			
計画開始日		計画終了日	
実開始日		実終了日	
担当者			
設計	実装	単体	レビュー
その他			

遅延発生（計画終了日前）

機能番号及び機能名			
計画開始日		計画終了日	
実開始日		実終了日	
担当者			
設計	実装	単体	レビュー
その他			

遅延発生（計画終了超過）

機能番号及び機能名			
計画開始日		計画終了日	
実開始日		実終了日	
担当者			
設計	実装	単体	レビュー
その他			



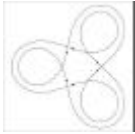
ソフトウェア工学（の応用）

✂ 定義

- ✂ ソフトウェアに関する一連の知識の体系化と応用

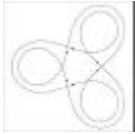
✂ 工学要素を固定して考えない

- ✂ プロセス（要素）と工学（要素）の間にはN:Mの関係がある．
- ✂ 全てが知られているわけではない．
- ✂ 変化する．



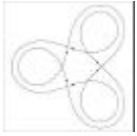
補足 : DSDM (Dynamic Systems Development Method)

重視する技術領域	補足
プロトタイピング	ビジネス 使用性 性能 / 容量 能力 / 技術の検証
モデル化技術	MoSCoWによる優先度付け 例として幾つかの開発用技術とデファクト標準の利用 ✖ オブジェクト指向モデル化 ✖ websiteナビゲーションモデル ✖ ビジネスプロセスモデル
テスト	リスク主導のテスト
構成管理	例えば, 週単位の配布物の管理のため.
支援環境	PCTE (Portable Common Tool Environment) に似た構造



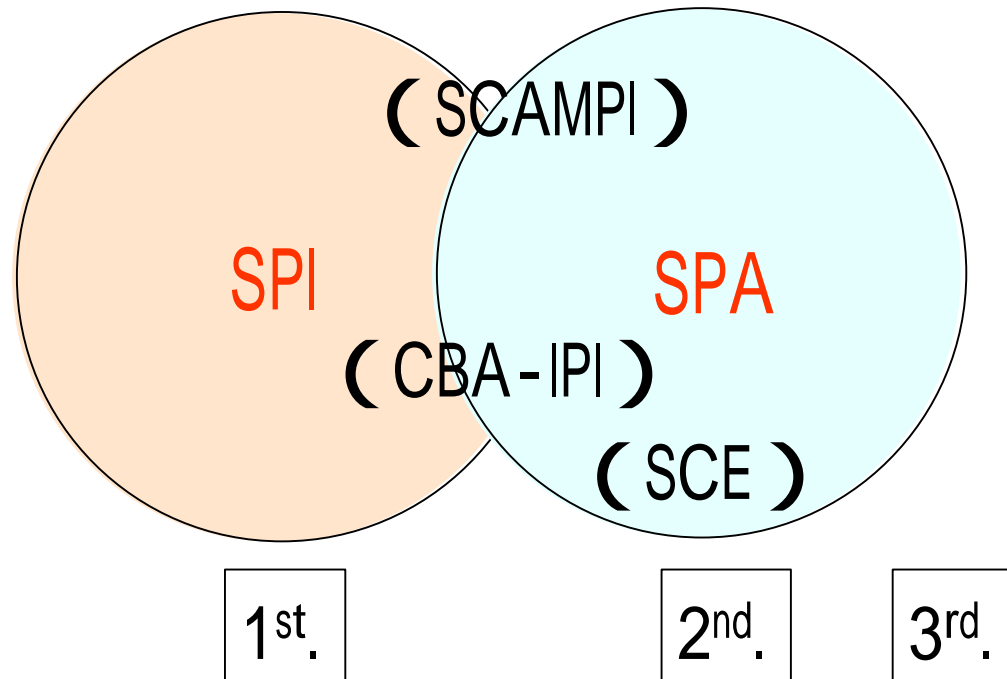
補足：CMMIの工学的プロセス領域

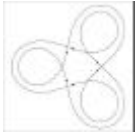
重視するプロセス領域	補足（プロセス領域に特有の実践-SP）
要求策定(development)	顧客の要求を策定する。 製品の要求を策定する。 要求を分析し、妥当性確認をする。
要求管理	要求を管理する
技術面での問題解決	製品コンポーネント問題解決 設計の実施 製品設計（結果）の実装
製品統合	製品統合の準備をする インターフェ이스の互換性を保障する 製品-コンポーネントの合成と製品の配布
検証（Verification）	検証の準備をする ピアレビューを実行する 選択された作業成果物の検証を行う
妥当性確認(Validation)	妥当性確認の準備をする 製品と製品-コンポーネントの妥当性確認を行う



（評価 / ）監査による改善モデル

- ✂ 第一者監査
 - ✂ 内部換算
- ✂ 第二者監査
 - ✂ ベンダ監査
- ✂ 第三者監査
 - ✂ 認証





対自活動としてのプロセス資産の確保

✂ 材料

✂ 明確なもの

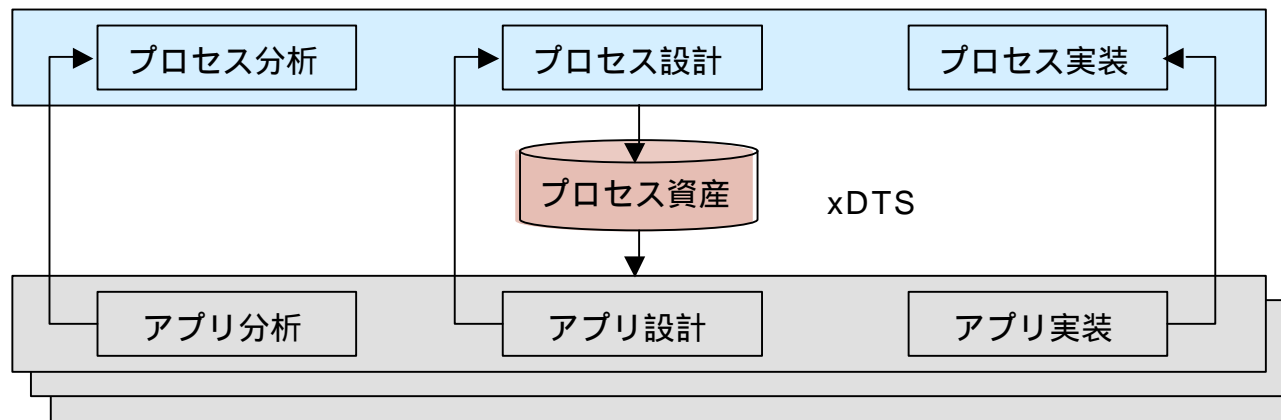
- プロセス及びプロセスメトリクス
- 既存のドキュメント（仕様書，内部設計書，チェックリスト，マニュアル，用語集，及び標準文書）
- 既存のプログラム及び関連スクリプト

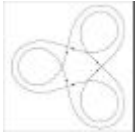
✂ 非形式的なもの

- 組織の文化 / 特性，コミュニケーションの状況

✂ ゴール

- 既存資産の利用（成功したもの，及び失敗したもの）





まとめと将来への課題

✎ ソフトウェアプロセス最適化 (SPO) は、現実的な解である。

🌟 技術だけでもプロセスだけでもプロジェクトは玉成しない

プロセス

- プロセスの全体像
- リスクに基づくプロセスの設計

技術

- マトリクスによる選択

ツール

- 広い範囲をカバーするツール群
- 全てカスタマイズ可能

🌟 SPOは3つの要素 { プロセス , 技術 , ツール } を総合する必要がある。