

PSPSMトレーニングを 実務に生かすTips

長田智裕

(PSPネットワーク/日本光電工業株式会社)

2004/9/16

1 . PSPSM (Personal Software Process) をトレーニングする意義

1.1 PSPSMトレーニングの紹介

1.2 PSPSM 7つのステップ

1.3 PSPSMトレーニングによる変化

PSPSMトレーニングの紹介

■ 15回の講義

- 演習のバックグラウンドとなる知識を得る
- プロセス、見積もり、EVM、欠陥分析、品質管理、GQM、形式言語、プロセス進化

■ 10のプログラム演習

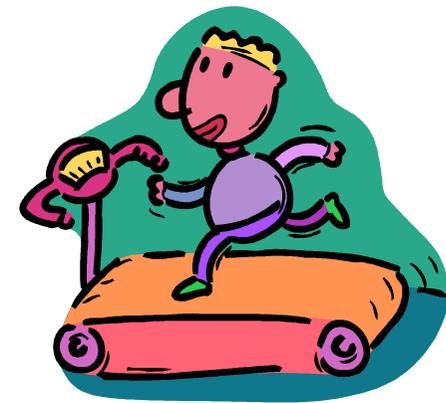
- 規定のプロセスに従ってプログラムを開発する

■ 5つのレポート

- 標準の作成、プロセスの分析、プロセスの定義と改善など

■ PSPトレーニングによる効果

- 規範に基づいたソフトウェア開発を行える
- 品質の高い製品を計画通りに開発する



PSPSM 7つのステップ

■ PSP3: 拡張プロセス

より大きなプロジェクトのために
PSP をスケールアップ

■ PSP2: 品質管理プロセス

品質管理技法, 設計テンプレートを導入

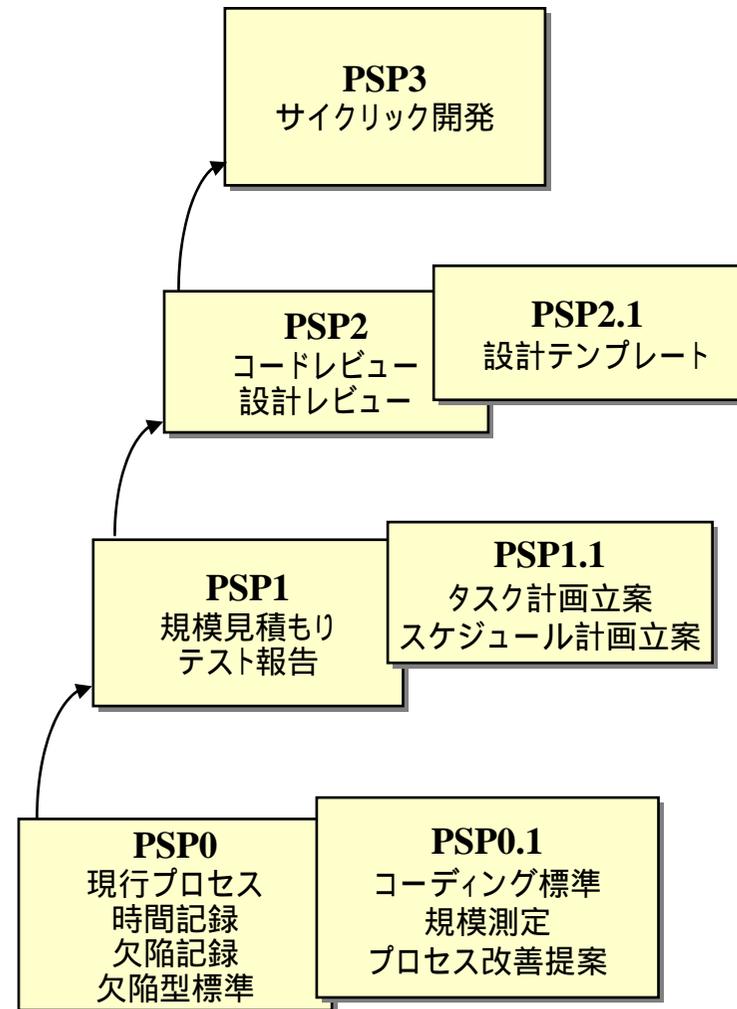
■ PSP1: 計画立案プロセス

規模および時間の見積もり
スケジュール計画と追跡の導入

■ PSP0: ベースラインプロセス

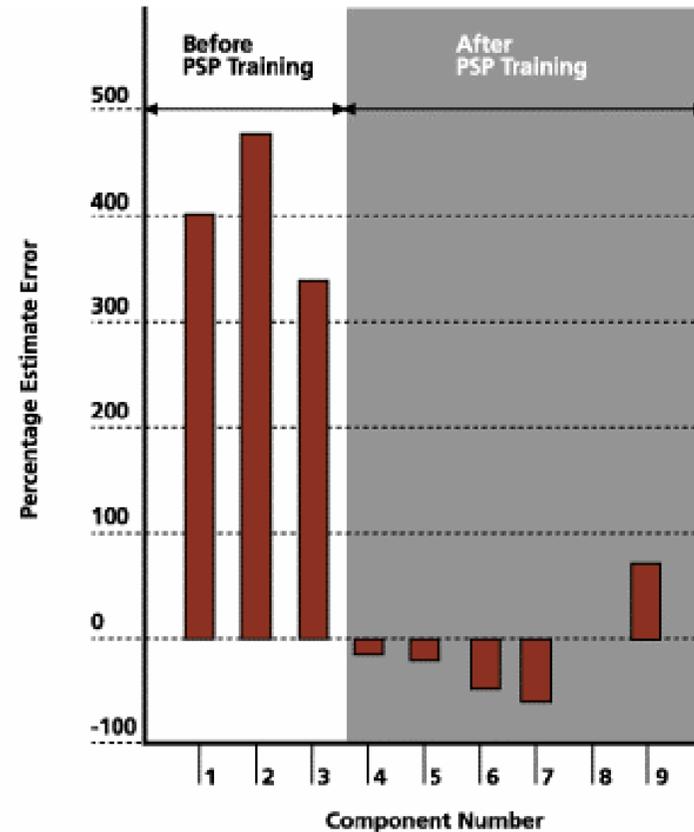
時間・欠陥・規模を測定し,
現行プロセスのパフォーマンスレベル
を把握

ベースライン確立



PSPSMトレーニングによる変化

- 見積もりや欠陥/テストに対して劇的な成果があったとの報告
 - AIS社やボーイング社の事例^[1]
 - PSPのROIはCMMIやISO9001の4倍!?^[2]
- 一方、PSPSMコースを終了した技術者がPSPSMのコンセプトを実務に利用していないとの報告もある^[3]
 - コース自体は、ムダのないよくできたものである



[Ferguson 97]

→ PSPSMトレーニングを導入しただけで、AIS社やボーイング社のような効果を期待できるか？

2 . PSPSMによって効果を上げるためには

2.1 本当のトレーニングの目的は？

2.2 PSPSMを実務に応用できない理由

本当のトレーニングの目的は？ (1)

■ カークパトリックの4段階評価モデル

■ レベル1 反応(Reaction)

- 受講者がトレーニングに興味を持てたかを、事後アンケートや観察によって評価する

■ レベル2 学習(Learning)

- 受講者がトレーニングの内容を学習したか(パフォーマンスが上がったか)を事前事後テストや自他へのインタビューによって評価する

■ レベル3 行動(Behavior)

- 受講者がトレーニングの内容を実行しているかを、時間を置いた後(例えば3ヶ月後)に自他のインタビューや観察などによって評価する

■ レベル4 結果(Results)

- トレーニングに成果(生産性や品質の向上、コストの減少)があったかどうかを評価する

本当のトレーニングの目的は？(2)

- 4段階評価モデルから得られる知見
- 企業におけるトレーニングの目的は実務におけるパフォーマンスの向上である
- PSPトレーニングによってパフォーマンスを向上するためには、PSPのコンセプトを実務で実践することが必要
- 従って、成果を上げるためには、まず最初にPSPのコンセプトを実務で実践しやすくする工夫が必要

トレーニングコースの運営自体をプロセスと捉える。トレーニングコースに対してレベル3・4の評価を行って分析し、改善する。



PSPSMを実務に応用できない理由(1)

■ PSPSMをどう実務に適応してよいかわからない

- 実際の業務はチームである
- 実際の業務がthe PSPと合わない
 - the PSP: トレーニングで用いるPSP0からPSP3までのプロセス。これに対してカスタマイズした、あるいはオリジナルの個人プロセスを a PSPという。
- そのまま適応すると仕事が煩雑になってしまう

■ データ収集が面倒

- ツールがない
 - 「もしデータ収集ツールをつかわないとしたら、仕事はペーパーワークになってしまう」^[4]
- データ収集をするだけではモチベーションが低下する
 - 目的は収集することではなく、分析してプロセス改善に生かすこと

→ the PSPをそのまま実務に持ち込むのは困難

PSPSMを実務に応用できない理由(2)

■ 管理者・同僚の理解、サポートがない

- データ収集、解析、PIP(プロセス改善提案)の時間が与えられない
- データ収集の意図が理解されない

■ PSPSMの原理が企業風土に合わない

- 規範に基づいて高品質なソフトウェアを作るより、不具合の修正が得意な技術者のほうがウケがよい
- 工数見積もりがうまい技術者より、残業する技術者のほうが給与が高い
- PSPの個人データを評価に使われるかもしれない不安

→ 技術者がPSPを実践しやすいような風土づくり

3 . PSPSMコース体験と 実務でのPSPSM活用経験から

- 3.1 タイムマネジメント
- 3.2 メトリクスの収集
- 3.3 EVM (Earned Value Management)
- 3.4 コードレビュー
- 3.5 プロセスエンジニアリング

タイムマネジメント

■ 知識労働者の条件として

- 実際は自分が思っているように時間を使っていない
- 無駄なことはせず、自分の強みを発揮できる最も重要なことに集中する
- アメリカのエグゼクティブは手帳に時間と行動を常にメモしている

■ PSPi (Introduction to the Personal Software Process)^[5]では、まずこれから

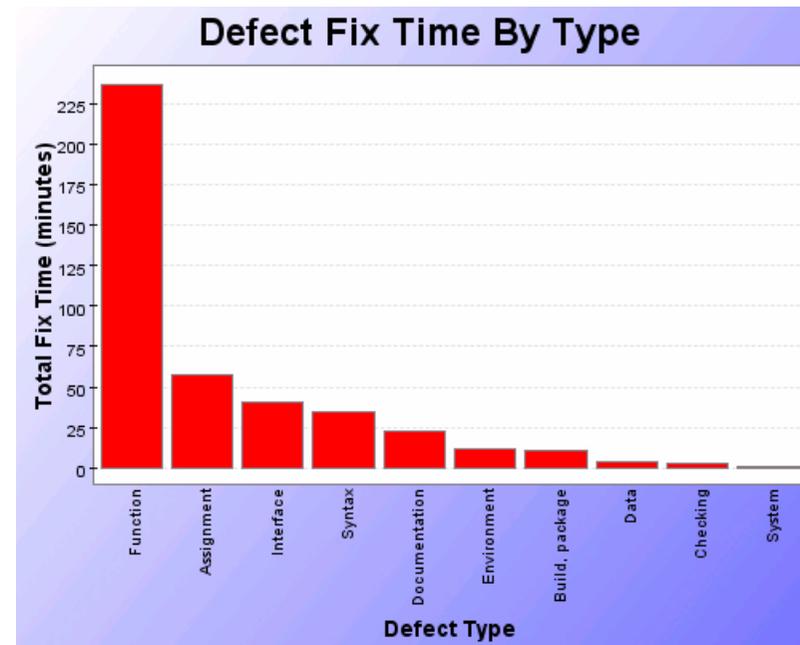
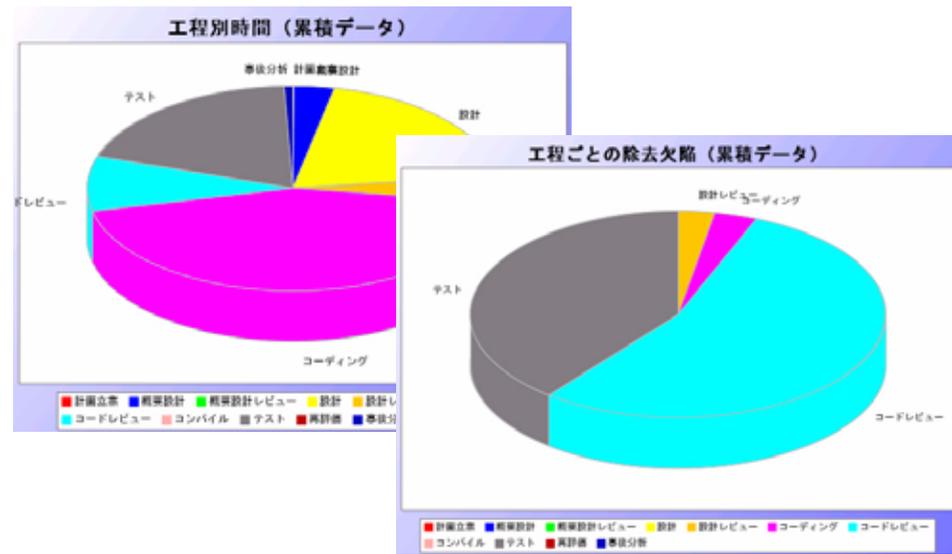
■ 実行することは以外と難しい

- ツールが必要
- モチベーション
 - 「見積もりをしない」「見積もりしても当たらない」



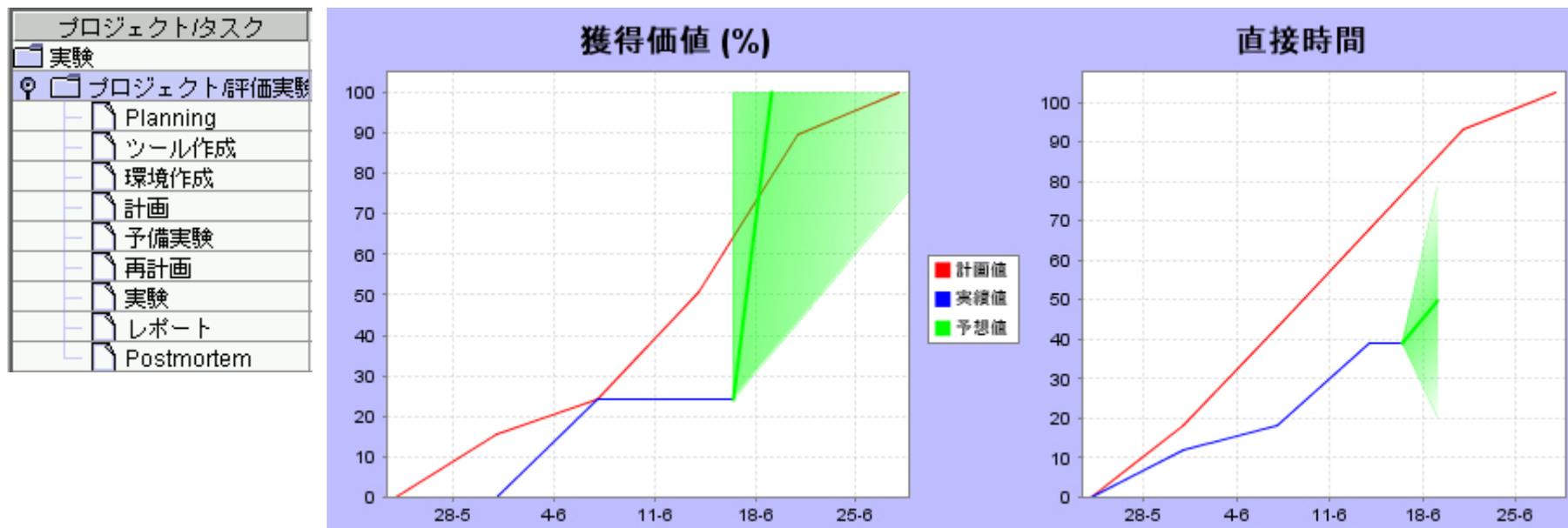
メトリクス収集

- 技術者自身が興味を持ってメトリクスを収集する
- ツールを利用する
 - 昨年紹介のProcess Dashboard^[6]など
- 技術者の他に、管理者の興味・理解が必要
- 分析する時間の供与
 - 分析し、改善に利用しないデータには意味がない



EVM (Earned Value Management)

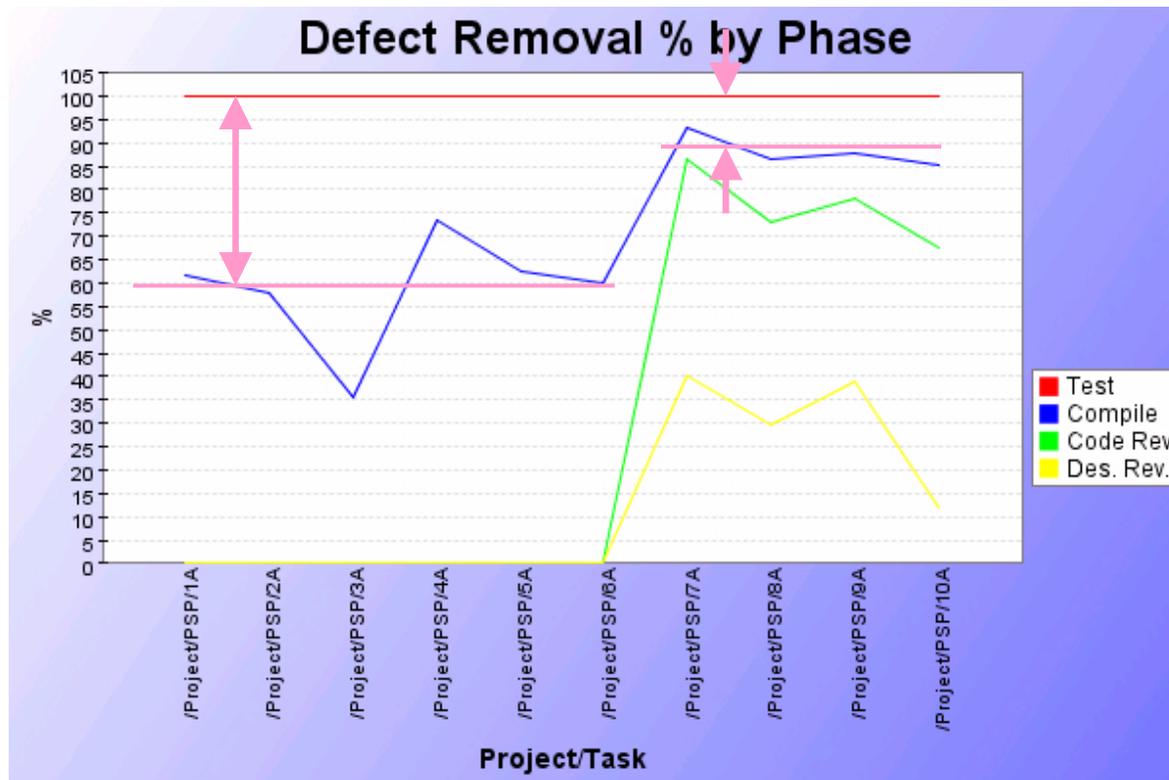
- 個人作業についてもWBSを作成し、EVで進捗をトラッキングする
 - プロジェクトマネージャになる前にEVMに慣れる
 - PSPの講義には簡易的な統計の知識もあり、PMBOK®の計画プロセス群の理解が深まる
 - 過去に経験のない仕事の見積もりは外れてあたりまえ。とりあえずは練習と経験(と履歴データ)のために



コードレビュー

■ PSPトレーニングの中でもっとも「ドラマティックな」体験

- 改善に根付いた、構造的なレビュー技法
- 目標「200LOC/時間以下」
- 直接実務適応できるプラクティス

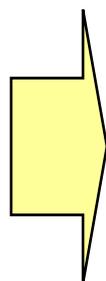


実務でのコードレビューの効果

■ コードレビューを導入前の以前の仕事と導入後の仕事の比較

- コードを動かす前なので修正がもったいなくない
 - 気が済むように修正し、その結果コードが洗練される
- 保守性、移植性の向上

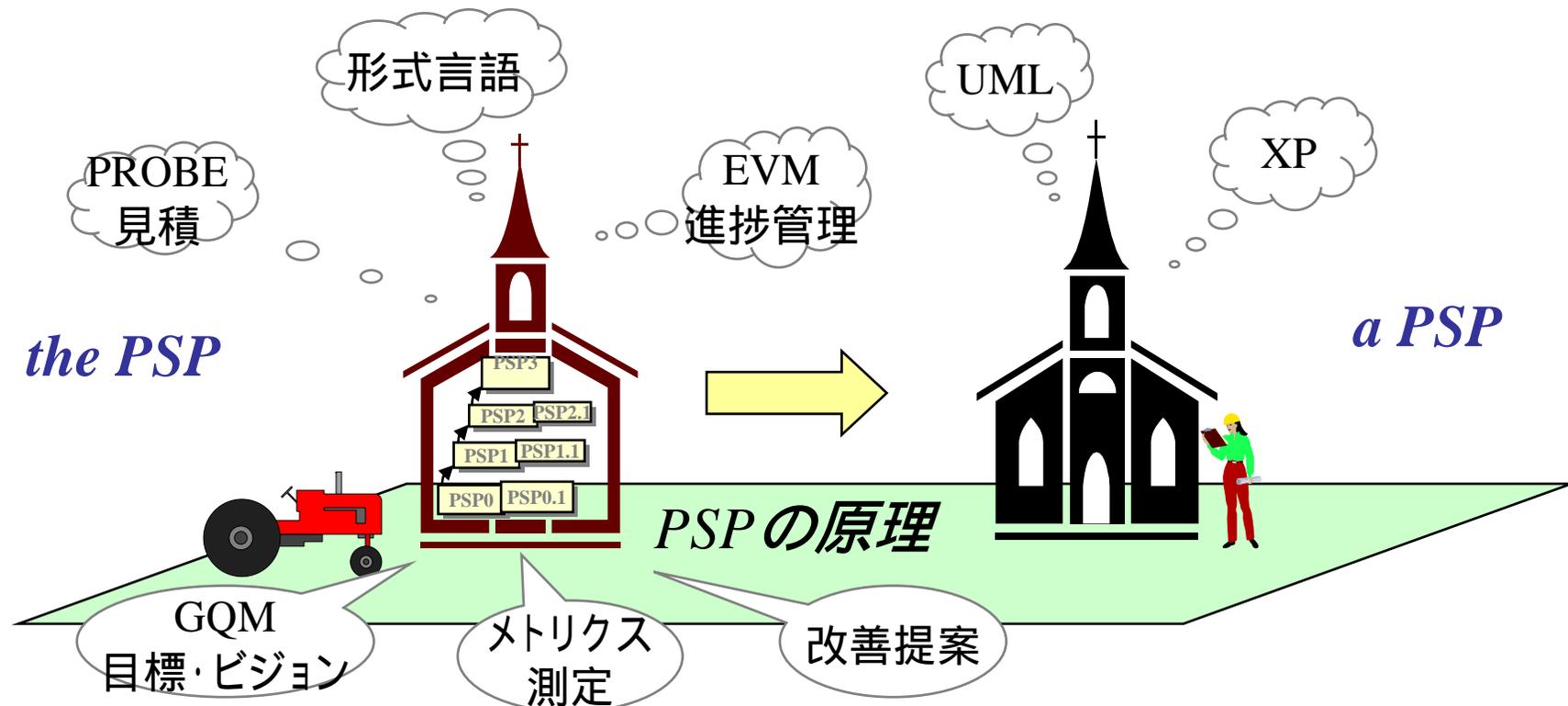
| | |
|--------------------|--------------|
| 対象 | 6778 LOC |
| テスト前の実行デバッグ 欠陥数 | 59 |
| 単体テスト 欠陥数 | 55 |
| 除去欠陥密度 | 8.11 欠陥/KLOC |
| 欠陥摘出率 | 94.8% |
| トータル 欠陥数 | 114 |
| 欠陥密度 | 16.8 欠陥/KLOC |



| | |
|---------------|--------------|
| コードレビュー 対象 | 3015 LOC |
| 時間 | 23.2 時間 |
| 欠陥数 | 110 |
| 除去欠陥密度 | 36.5 欠陥/KLOC |
| 欠陥除去効率 | 4.75 欠陥/時間 |
| レビュー効率 | 130.0 LOC/時間 |
| 単体テスト 対象 | 3444 LOC |
| 欠陥数 | 83 |
| 除去欠陥密度 | 24.1 欠陥/KLOC |
| 欠陥除去効率 | 2.7 欠陥/時間 |

プロセスエンジニアリング(1)

- PSPトレーニングの本質はプロセスの体験にある
- 各技術者が自分自身に適切なプロセスを設計する
- そして、その個人プロセスを継続的に改革していく



プロセスエンジニアリング(2)

プロセスはしもべである。

もしそれが役に立たないなら、変更しなければならない。 [7]

■ 個人プロセスの改革の経験はチームあるいは組織のプロセスの改革にも生かせる

- PSP体験により、技術者はCMMの80%を体験していることになる
- PSP体験により、プロセス改革に必要な知識や経験を獲得できる

■ プロセスのバランスを知る

- TSP/PSP^[8]
- Agile on TSP/PSP^[9]
- RUP with PSP^[10]



4. おわりに

4.1 The PSPを超えて

4.2 PSPネットワークの紹介

The PSPを超えて

■ 守・破・離

- 守(しゅ) : まずは師の教えを守り、指導を受けたとおりにやる
- 破(は) : 師の教えを基本として、自分に合った、自分なりのやり方でやる
- 離(り) : これまで学んだことすべてを、自分の形で体現し、師を超える

■ 「離」こそ目指すところ

- CMMもアジャイルも守・破・離



PSPネットワークの紹介

■ PSPネットワークの生い立ち

- PSPネットワークは、創価大学 (故)藤野喜一教授の遺産です。藤野教授は1997年度に修士課程の科目にPSPを採用され、翌1998年度には社会人向けのPSPコースも開設されました。この社会人向けコースを修了した20名のなかの有志が、同窓会的につくったのがPSPネットワークです。発足は1999年5月です。

■ PSPネットワークの連絡先

psp-secretary@yahoogleroups.jp

(メールアドレスが変わりました)

参考文献

- [1] Ferguson,P et al., "Pathways to Process Maturity: The Personal Software Process and Team Software Process," IEEE Computer, 30(5), pp.24-31.
- [2] Rico,D.F, "How to Estimate ROI for Inspections, PSPSM, TSPSM, SW-CMM[®], ISO 9000, and CMMISM," <http://davidfrico.com/dacs02b.pdf>
- [3] El Emam,K et al., "Implementing Concepts from the Personal Software Process in an Industrial Setting," Proc. Software Process 1996, IEEE computer Soc. Press
- [4] Moriso,M, "Applying the PSP in Industry," IEEE Software, 17(6), p.90-95
- [5] Humphrey,W.S,PSPネットワーク訳, "パーソナルソフトウェア入門," 共立出版, 2001
- [6] "Software Process Dashboard Initiative", <http://processdash.sourceforge.net/>
- [7] Humphrey,W.S,ソフトウェア品質経営研究会訳, "パーソナルソフトウェアプロセス技法 - 能力向上の決め手", 共立出版, 1999
- [8] Carnegie Mellon University, "The Team Software Process (TSP) and the Personal Software Process (PSP)", <http://www.sei.cmu.edu/tsp/main.html>
- [9] Humphrey,W.S, "Setting The Agile Context," <http://www.xpuniverse.com/slides>
- [10] Svensson,H, "Incorporating the Personal Software Process into the Rational Unified Process," <http://www-106.ibm.com/developerworks/rational/library/content/RationalEdge/may01/IncorporatingthePersonalSoftwareProcessintotheRUPMay01.pdf>