

PSPはなぜ進化的プロセスなのか

- PSP演習から見えてきたこと -

2005年10月14日

発表: PSP/TSP分科会リーダー 宮脇 祥子

はじめに

- P S Pは「進化的プロセス」だとか、C M Mの次はP S P / T S Pだとか、言われています。
反対に、「P S Pはたくさんの記録をとらなければならないので業務には適用できない」などの否定的な話もよく聞かれます。
- でも、その本当の姿はあまりよく理解されていないのではないのでしょうか。
- 我々はこの「本当の姿」を探すべく、勇気をもってP S P演習の旅に一步を踏み出しました。果たして、その結末は。。。

PSP/TSP分科会メンバー

	氏名	会社名
1	遠藤 勉	日本電気(株)
2	角田 文広	日立ソフトウェアエンジニアリング(株)
3	宮脇 祥子	(株)日本科学技術研修所
4	牧田 宏	(株)日立製作所
5	小室 睦	日立ソフトウェアエンジニアリング(株)
6	松井 昌子	富士通(株)
7	村田 雅繁	三菱スペース・ソフトウェア(株)
8	岩見 好博	オリンパス(株)
9	秋山 義博	日本IBM(株) PSP講師
10	川下 敬之	PSPネットワーク PSP講師
11	田中 裕彦	松下電器産業(株) アドバイザ
12	CCT分科会	講義担当としてコラボレーション

PSP/TSP分科会のスタンス

- CMMはSPIの1モデル、WhatはあるがHowがない
 - 個々の開発者はそれなりのプロセス・スキルを持っているが、作業規律や管理がうまくない組織にはCMMが有効である
 - 仕事の3要素である「人」「プロセス」「技術」のプロセスだけにフォーカス
 - 「プロセスを標準化すればよい」との短絡思考を生む弊害も
 - 開発者のプロセス・スキル自体に問題がある組織には？？
 - 「標準プロセス」をうまく使えない プロセス形骸化

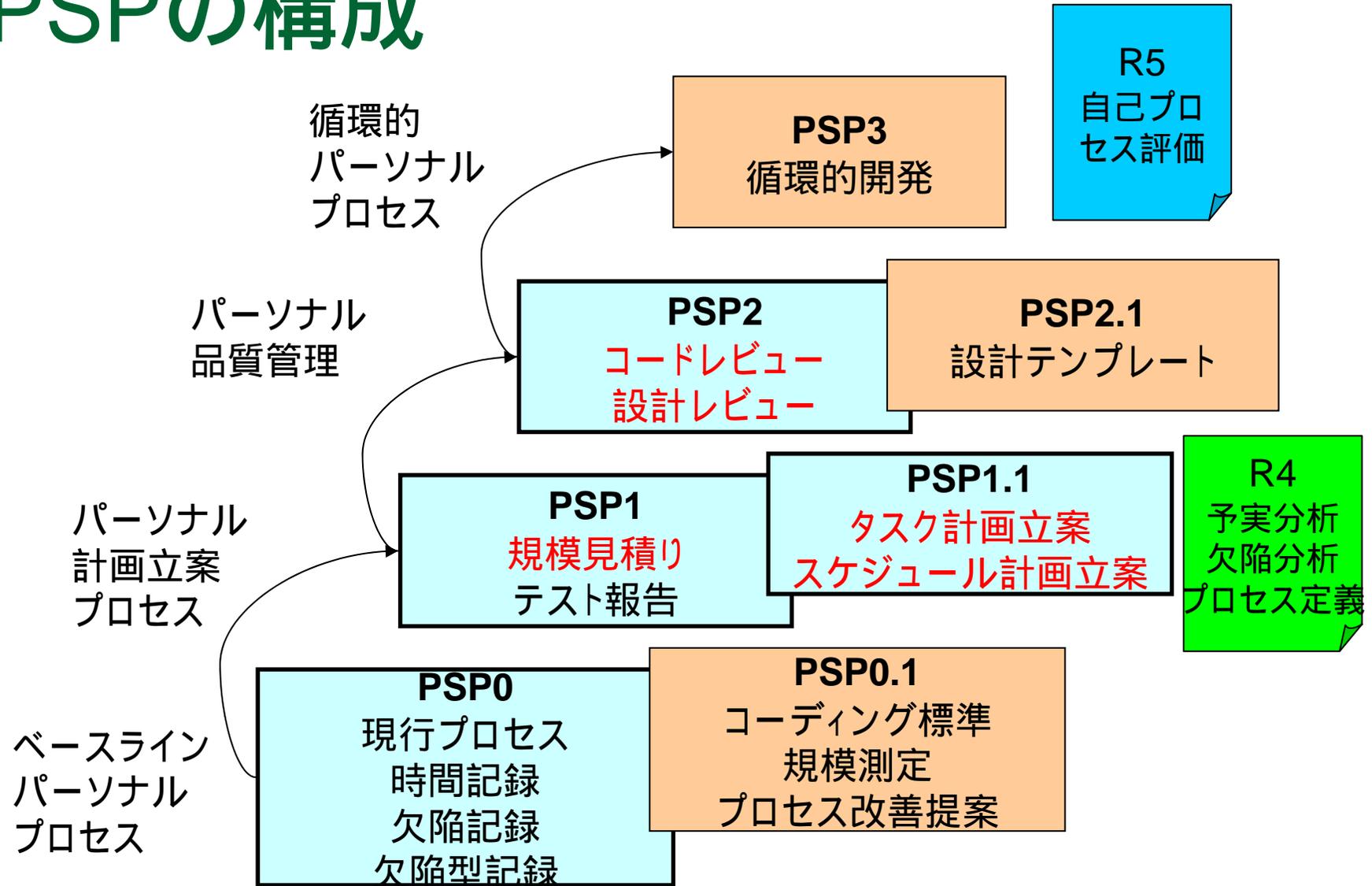
PSP/TSP分科会のスタンス(2)

- プロセス・スキルに問題がある開発者とは？
 - 規模見積りやテストケース作成どころか、時間管理も出来ない開発者がいる
 - ソフトウェア開発で起こる問題の多くにはすでに答えがある。
 - 問題は、それを知らない、あるいは知ろうとしないこと
- 米国でTSPが普及しており、PSP Qualified Engineerが求められる

PSP/TSP分科会のテーマ

- 開発者の真の能力アップはどうすれば出来るのか
 - ただプロセス標準に従うだけでなく、コンピテンシーのある自立した開発者に
 - チーム、組織、顧客との関係でProject計画・運営が出来る
 - 開発者自身の見積能力や生産性を理解することが改善を始める一歩となり、PSPはそれを支援するツールである(らしい)。
- そうだ、自分たちもPSP演習をやってみよう。

PSPの構成



PSPの構成(2)

- PSPでは、各ステップに対するプロセススクリプトやテンプレートと1～2つの課題が与えられる。
- そのプロセス通りに課題の設計・プログラミングを行っていく。
- 見積・実績時間や欠陥などのデータを随時記録していく。
- 課題の難易度は様々である。
- プロセスはステップが進むにつれて追加、拡張される。
- 中間でレポートを作成する。
- 最終的には、自分自身の日常的なプロセスを計測・分析し、自分のゴール達成のためにプロセスを改善できるようになる。

PSPで体験したこと

PSPの考え方に慣れ、自分でよりよいプロセスを工夫し始める

PSP1

- +PROBE法により新規及び変更のLOCを見積もる
- +テスト計画をたててテストを実施する

- ✓計画をたててテストを行うことを体験する
場当たりにテストするより効率的
- ✓過去の実績から見積を行う方法を学ぶ
過去の実績の使い方を理解



PSP0.1

- +新規及び変更のLOCを見積もる
- +プロセスの改善点を記録する(PIP帳票)

改善の魂が芽生えてくる

- ✓見積があたらない
見積・実績を合わせたいくなる
- ✓こんな欠陥がある
少しでも欠陥を減らしたいくなる



PSP0

- 課題に対し、時間見積をおこなう
- プログラムを設計・実装する
- 各種作業にかかった時間、欠陥、規模の記録をとる

- ✓見積があたらない
時間見積が難しい事を再確認
- ✓こんな欠陥がある
自分の能力を知る

「工夫」の例

- レビューしてもらおう事を考えて、プログラムの構成をわかりやすく記述した説明書を作成した。
- あとからでも欠陥の意味がわかるように、欠陥の説明をわかりやすく記述した。
- 不具合を減らすために、コンパイル前にコードレビューを行った。

これらは、いずれPSPのステップで要求されるものである

自分が「不便」と感じるところから「改善」が始まる

PSPで体験したこと(2)

自分の「プロセス」を持てた！

PSP2.1

+適切な設計テンプレートを用いて設計をおこなう

✓テンプレートを用いると生産性が下がった？
プロセスに慣れるには時間がかかる

自分の弱みを克服しよう！

PSP2

+設計、コードレビューチェックリストを作り、
設計、コードレビューをおこなう
+欠陥数を見積もる

✓個々人の欠陥の特徴を分析
汎用ではなく、これを反映した
個人別チェックリストが有効！

課題が難しく挫折しそうになる
のでコーチが必要

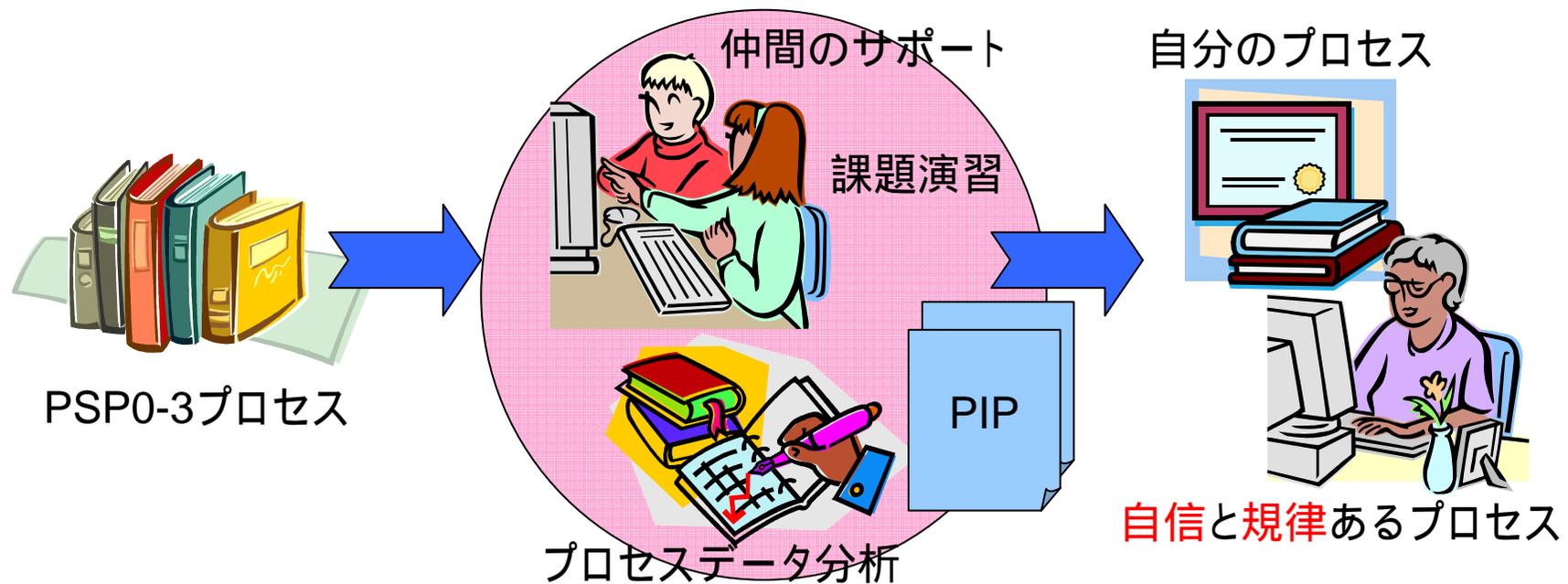
PSP1.1

+一日の中でPSPに使用できる時間を考えて
タスク計画をたてる。
+課題を行うため、統計の基礎を理解する。

✓PSPに使用できる時間はとても
少ない事を知る
スケジュールの立て方を学ぶ
✓基本的な統計が自分のデータを
活用するのに役立つ事を知る

PSPの本質？

- PSP演習は個人のソフトウェアプロセスの体験
- 演習を通して、自分のプロセスの問題、課題に気づき、自分のソフトウェアプロセスを開発する



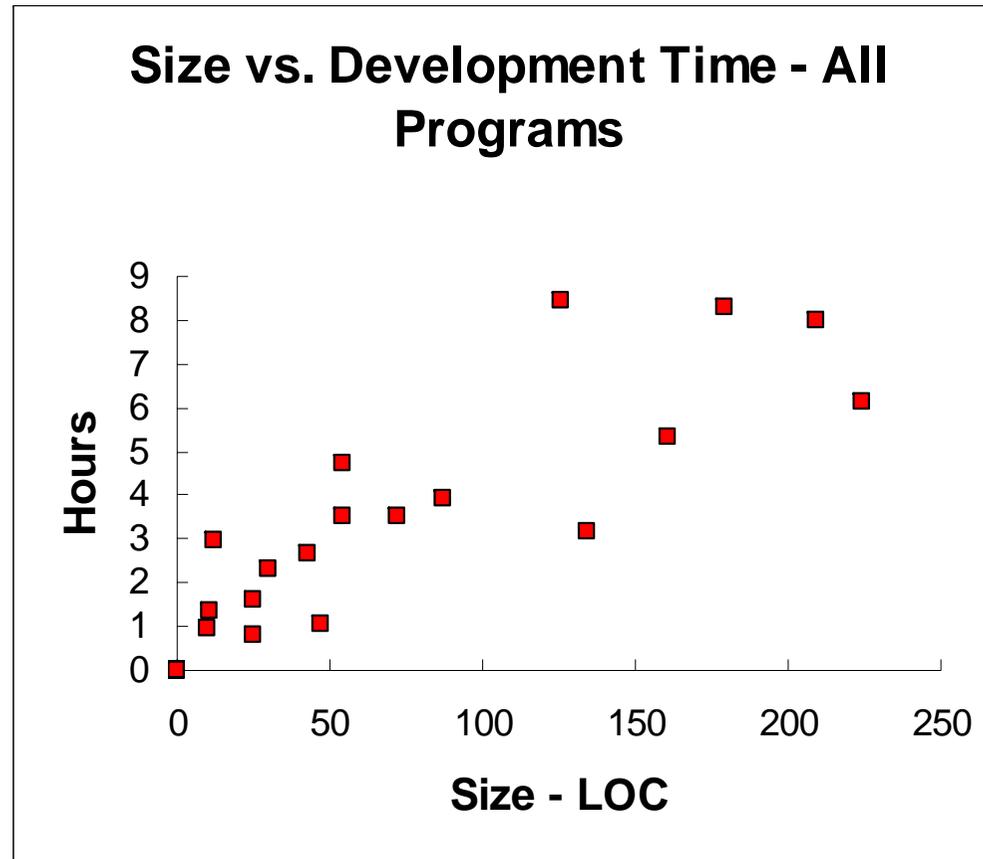
PSPは、よく考えられている！

- 「予測して管理する」
 - 履歴データ(見積もり根拠)は有用
 - 某社でのPSP受講の目的は「ムリな開発工数や期間に反論できる手法を知りたい」が最も多い
 - 見積もりには幅がある！
 - 一点見積もりは、そもそも非現実的？
 - この幅は時間とともに狭くなるはず
 - でなければ、SEとして成熟していない？
- 「履歴データ」は正しく！
 - 過少実績申告は後で自分の首を絞める
 - Process Dashboardは履歴データ管理に有用
 - 自動計測できないメトリクスは管理が難しい？

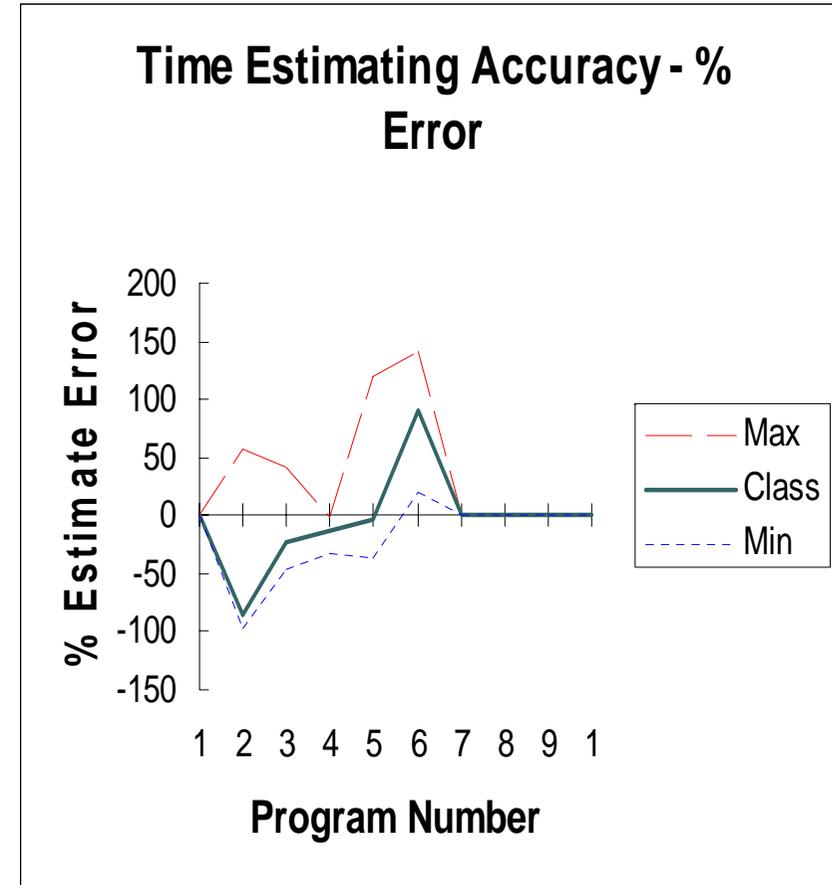
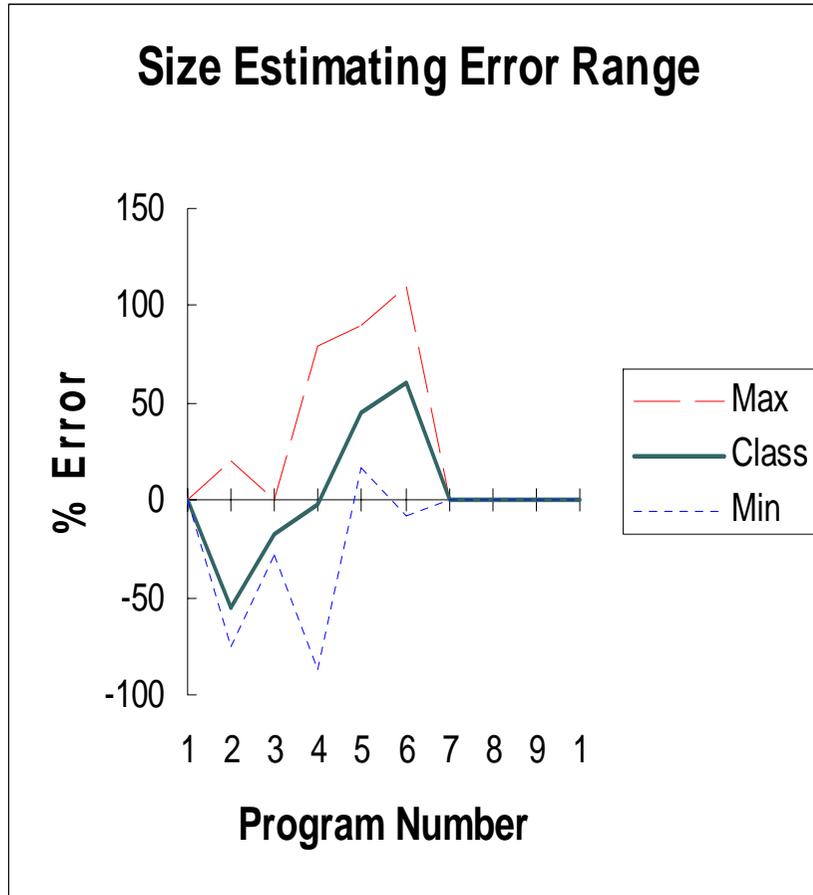
PSPは、よく考えられている(2)

- 詳細設計書、テスト仕様書の必要性を体得
 - 開発していて「詳細設計書」、「テスト仕様書」があるといいな、と感じると、次のステップでそれが要求されている！！
 - 設計レビュー、コードレビューも自分の間違い傾向を自覚した後、それを防ぐよう個々人で作成する
 - 初めから提示されるより有難味がある
 - 強制と自発では定着度合いが違う
 - なぜPSP0があるのか、が分かった
- 気がつくと、設計書レビュー、コードレビューを自然にやっていた！
 - レビュー工程定義がなくとも、レビュー実績が採られる
 - 設計中、コーディング中に見つけた欠陥はレビュー実績に

プログラム規模と開発時間の対応

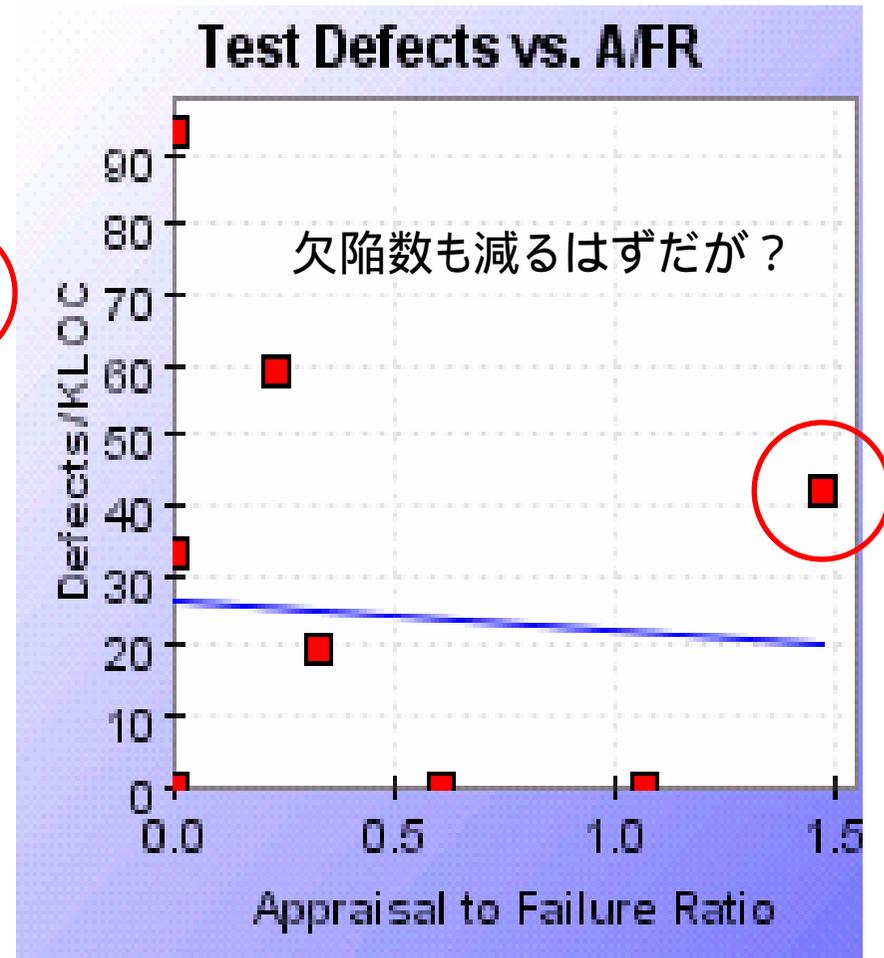
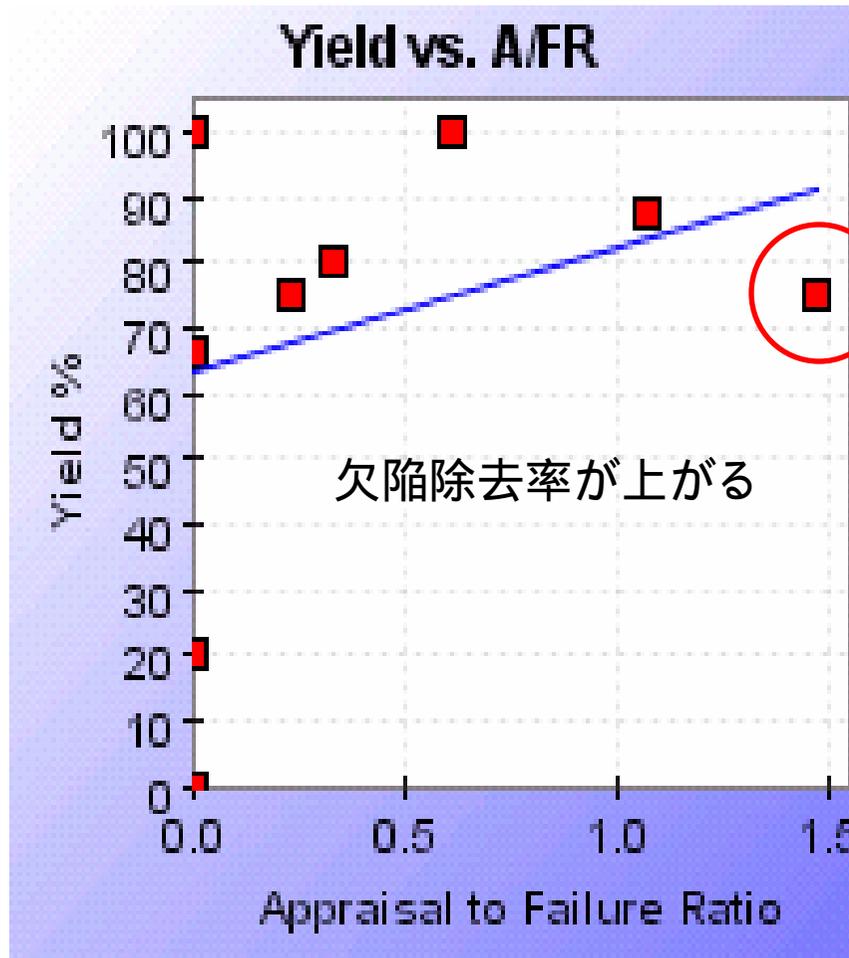


規模・時間の見積り誤差

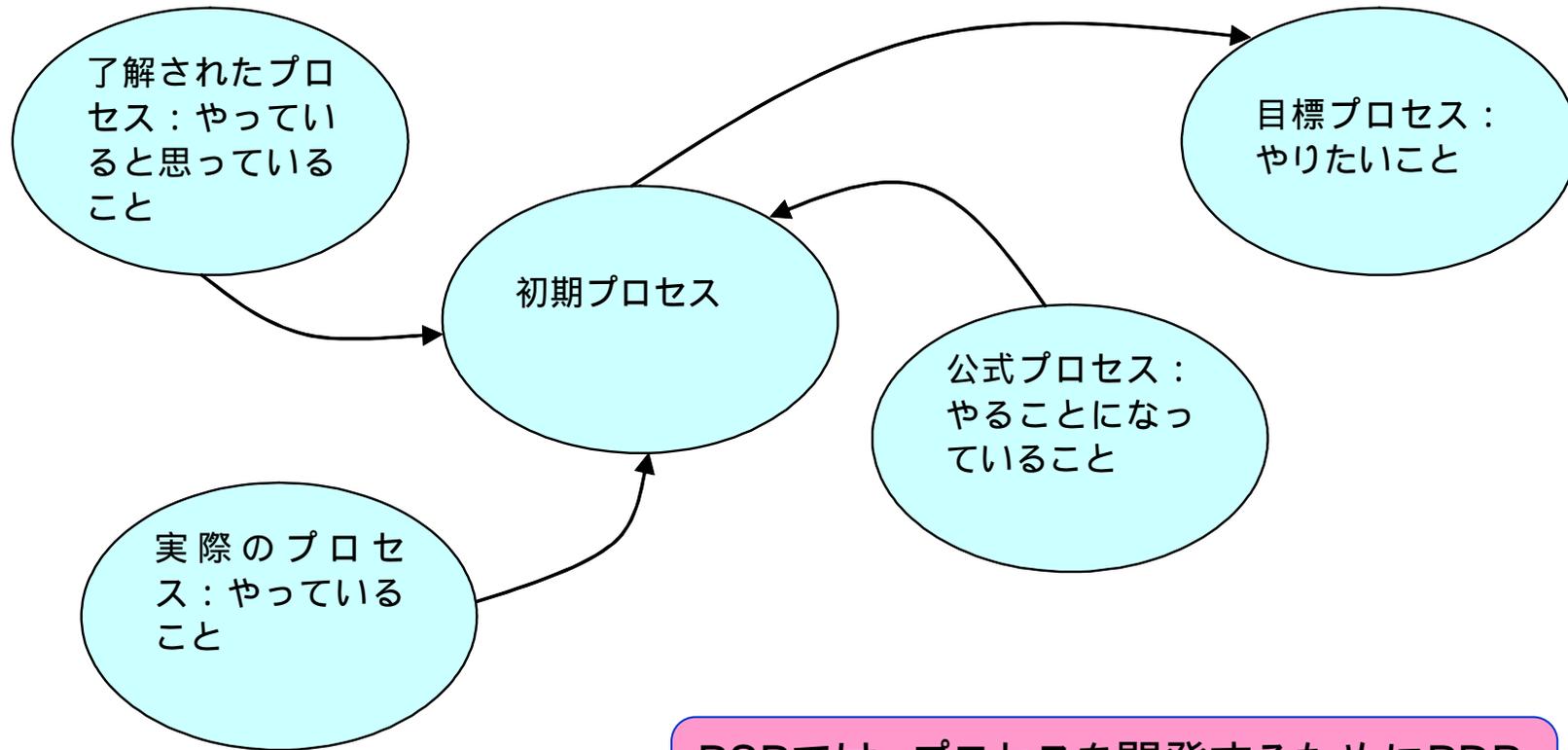


条件: 1A ~ 6A 3名のサマリーデータ

評価コストをかけるると



プロセスの進化



PSPでは、プロセスを開発するためにPDP
(Process Development Process) を提供

PSPを実施する上でのアドバイス

- ツールを使ってデータ記録の負荷を軽減する必要がある。(ProcessDashboard)
- 仲間同士で成果物をレビューしあうことで様々な「気づき」を得ることができ、さらに改善につながる。
- 言語知識や開発環境は整えてから実施する。
- 統計の知識が必要になるので、**コーチ役が必要**。
- 課題の提出ルールやレビュールールなどを決めておくと混乱をさけることができる。
- PSPはあくまでも教育のツール。演習をせずして業務に適用すべきではない。

企業内でPSP演習が可能か

- PSP演習を行う最大の課題は演習時間の確保
 - 比較的時間がとれる時期に演習を行うことが現実的
- PSPは、企業内研修というよりも教育機関で実施する事を目的に作られている
 - 日本の教育機関では「ソフトウェアエンジニアリング」を本格的に教えている所がほとんどないため、教育機関でPSPを行うこともむずかしい
- 自分達が企業内研修のインストラクタに

まとめ

- P S Pを実施することで自分自身をみつめ、かつ進化していく事を体験できた。(見積精度の向上、プロセス改善能力)
- P S Pの目的は、決められたプロセスに従って作業するのではなく、「自分自身のプロセスを定義する能力を身につけること」であり、それを可能にしてくれるツールだと感じた。
- P S Pは机上で勉強してもだめ。演習をしてみないとわからないことがたくさんある。
- みなさん、ぜひP S Pをやってみてください。かならず発見があります。

補足資料

PSP1 プロセススクリプト

工程番号	目的	モジュールレベルのプログラムの開発をガイドする
	開始条件	<ul style="list-style-type: none">・問題記述・PSP1プロジェクト計画概要・規模見積テンプレート・規模の見積値と実績値の履歴データ・時間記録ログと欠陥記録ログ・欠陥型標準・ストップウォッチ(任意)
1	計画立案	<ul style="list-style-type: none">・要求仕様を作るか手に入れる・PROBE法により、必要とする新規および変更のLOCを見積もる・規模見積テンプレートを完成する・必要とする開発時間を見積もる・プロジェクト計画概要に計画データを記入する。・時間記録ログを完成する。

出典: パーソナルソフトウェアプロセス技法 Watts S.Humphrey 著

PSP1 プロセススクリプト(続き)

工程番号	目的	モジュールレベルのプログラムの開発をガイドする
2	開発	<ul style="list-style-type: none">・プログラムを設計する・設計を実装する・プログラムをコンパイルし、見つけた全ての欠陥を修正し記録する・プログラムをテストし、見つけたすべての欠陥を修正し記録する・時間記録ログを完成する
3	事後分析	<ul style="list-style-type: none">・プロジェクト計画概要に時間、欠陥数、規模の実績データを入れて完成させる
	終了条件	<ul style="list-style-type: none">・徹底的にテストされたプログラム・見積データと実績データが記入され完成したプロジェクト計画概要・完成した規模見積プレート・完成したテスト報告プレート完成したPIP帳票・完成した欠陥記録ログと時間記録ログ

PDP (プロセスの開発プロセス)

目的		<ul style="list-style-type: none"> ・プロセスの開発と拡張をガイドする
開始条件		<ul style="list-style-type: none"> ・新規プロセス、またはプロセスの改訂が必要である。 ・プロセス要求を記述した要件、またはPIPがある。 ・PDPフォーム、スクリプト、標準、レビュー・チェックリストが手元にある。
一般事項		<ul style="list-style-type: none"> ・開発者個人あるいはチーム全体がPDPプロセスに従う。 ・チームで使うとき、チーム・プロセスに組み込まれていること。 ・その指示に従って、スクリプト、フォーム、標準、他の要素を開発する。 ・大きなプロセスを部分に分解して、PDPを使って各部分を開発する。
Step	活動	説明
1	プロセス計画をつくる	<ul style="list-style-type: none"> ・(もしあれば)既存プロセスの在庫目録をつくる。 ・新しい/修正されるプロセスの概念設計を行う。 ・新規の、再利用、修正、削除されるプロセス要素を見積もる。 ・スクリプト、フォーム、標準、サポート資料、他のプロセス要素別に開発工数を見積もる。 ・プロセス要素ごとに総所要時間を見積もる。 ・プロセス、フェーズごとに総所要時間を見積もる。 ・PDP計画要約を完成させる。
2	プロセス要件	<ul style="list-style-type: none"> ・PIP、または他の要求文書を分析する。 ・新規/修正プロセスへの要件を定義する。
3	概要設計	<p>新規/修正プロセスについて</p> <ul style="list-style-type: none"> ・トレーニング、特別なツール、支援などのプロセス制約を定義する。 ・プロセス・フローと基本的フェーズを定義する。
4	プロセス・スクリプト	<ul style="list-style-type: none"> ・各プロセス・スクリプトを作成、または修正する ・レビュー・チェックリストを使って、各プロセス・スクリプトをレビューして訂正する。
5	プロセス・フォーム	<ul style="list-style-type: none"> ・各プロセス・フォームを作成、または修正する ・レビュー・チェックリストを使って、各プロセス・フォームをレビューして訂正する。

PDP (プロセスの開発プロセス)

6	プロセス標準	<ul style="list-style-type: none"> 各プロセス標準を作成、または修正する レビュー・チェックリストを使って、各プロセス標準をレビューして訂正する。
7	他のプロセス要素	<ul style="list-style-type: none"> 他のプロセス要素を作成、または修正する レビュー・チェックリストを使って、他のプロセス要素をレビューして訂正する。
8	プロセス・ノート	<ul style="list-style-type: none"> 必要ならば、プロセス・ノートを作成、または修正する レビュー・チェックリストを使って、プロセス・ノートをレビューして訂正する。
9	プロセスのテスト	<ul style="list-style-type: none"> すべてのフォームを完成するスクリプトに従って、そのプロセスをテストする。 可能であれば、実際のプロジェクトでそのプロセスを使う。 すべての問題と改善提案をPIPに記録する。 すべてのPIPに対応してプロセスを調整する。
10	ピアレビュー	<ul style="list-style-type: none"> 可能であれば、同僚に誤りを識別するためプロセスをレビューしてもらい、修正を提案してもらう。 レビュー結果に基づいてプロセスを訂正する。
11	プロセス・パッケージ	<ul style="list-style-type: none"> 配布、使用に適した形で出来上がったプロセスをパッケージする。 最終パッケージの誤りをチェックし、必要な修正を行う。
12	事後分析	<ul style="list-style-type: none"> すべてのプロセス・データを計画要約に記録する。 プロセス要素、フェーズの時間と比率を計算する。 将来の計画時に使えるよう、これらのデータをプロセス・データベースに記録する。 PDPプロセスの改善点があれば、PIPフォームに書く。
終了条件		<ul style="list-style-type: none"> 配布・使用の準備が整った完成した高品質のプロセス プロジェクト要約報告 将来使うためのプロジェクト・データ

PSP演習修了者の表彰



「PSPの種」
PSP演習の体験を
日々に活かして
花を咲かせましょう