



プロジェクトマネージャのための 実践的ガイダンス

- プロマネ拒否症への処方箋 -

2007年11月1日

アシストソリューション研究会
プロジェクトマネージャ育成分科会(2006年東日本)
発表 岩見 好博(オリンパス)



目次



- アシストソリューション研究会とは
- 分科会メンバー紹介
- 問題提起：「広がるプロマネ拒否症」
- メンバー間の議論
- プロマネ拒否症を解消するには
- TSPとは
- プロマネのための実践的ガイダンス
- Q&A



アシストソリューション研究会とは



■ 目的

- ソリューション研究会は、お客様相互の交流を育む場として、日頃お客様が抱えておられる課題や疑問をお客様同士で討議し、意見交換を行っていただくことを目的に活動しています。

■ 会員資格

- 弊社の提供するソフトウェア・パッケージおよび各種サービスをご利用いただいている企業、団体に所属する方であればどなたでもご参加いただけます。また、ひとつの団体から複数名の会員登録が可能です。

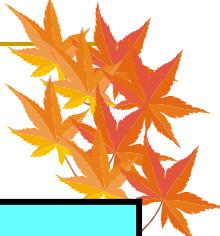
■ 会費

- 弊社のユーザーであれば、無料でソリューション研究会の活動にご参加いただけます。

□ <http://www.ashisuto.co.jp/users/sg/index.html>



2007年度 活動分科会



東日本	中日本	西日本
金融庁内部統制基準案への対応	JSOX法における適用範囲	内部統制におけるIT統制
内部統制におけるIT統制の構築方法と運用	IT部門の人材育成	ITILによる運用業務の可視化
IT投資対効果 ~投資の事後評価~	Webの今後と新技術 (Web2.0、Ajax等)	Webの今後と新技術 (Web2.0、Ajax等)
企業競争力アップに向けた「営業力強化」への取り組み	ソフトウェア開発品質の見える化	ソフトウェア開発品質の見える化
情報共有 / 知識定着化の事例検証		企業レベルでのオープンソース活用
プロジェクト・マネージャの育成		
Web2.0とビジネス・モデル		



2006年度分科会メンバー



- 飯島 郷 ITインフォメーションシステムズ(株)
- 佐々木 康夫 (株)アース製薬 サブリーダー
- 森 道夫 (株)エクサ アドバイザー
- 岩見 好博 オリンパス(株) リーダー
- 梶原 正人 (株)東海ソフトウェア
- 伊藤 裕規 日立ソフトウェア(株)
- 朝倉 高志 (株)アシスト
- 館 一広 (株)アシスト
- 板木 栄樹 (株)アシスト



問題提起



- 高品質のソフトウェアを予定のコストでしかもスケジュールどおりに開発できたら、それほど素晴らしいことはありません。
- しかし、開発現場は、度重なる要件変更やテストに次ぐテストで混乱し、開発者はやりがいを持てず「燃え尽き」ようとしています。プロマネはその対応に追われて「プロマネ拒否症」に。
- ソフトウェア開発を本来の「楽しくてやりがいのある」仕事にしたいと思い、当分科会はTSP、PSPに注目しました。



「広がるプロマネ拒否症」



- 「やりがいはあるが支援の体制や待遇に不満」があり73.0%がプロマネになりたくない。その理由として
 - 責任の重さに対して報酬が少ない
 - 現場のプロマネをサポートする体制／制度が十分でない
 - コンサルタントなど他の職種のほうが仕事内容や報酬が魅力的
 - プロマネの専門性が尊重されていない
 - 教育／育成のための体制や制度が十分でない
 - プロマネの意識調査(日経コンピュータ2005年7月11日号)
- 失敗プロジェクトが多くて成功体験が持てず、やりがいをなくしていくのが本当の理由？？



メンバー間の議論



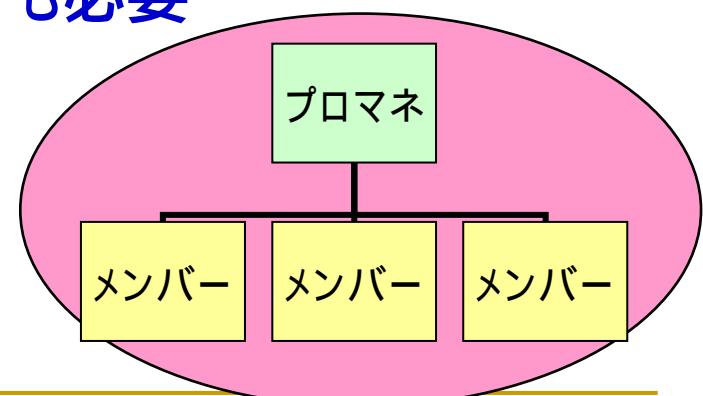
- 今のプロマネは、経営、現場、そして本人からNG！
 - 経営：QCD未達、赤字のプロジェクトが多い
 - 現場：決断しない、リーダーシップがない
 - 本人：やりがいのある仕事とは思えない
範となる先輩がほとんどいない
- 全くの新規開発はほとんどない。
 - ソフトウェア開発案件の80%（一説には95%）以上が既存システムの拡張あるいは保守
 - 既存システムのドキュメント不備で、担当が固定化
 - 同じことを長年繰り返していると「燃え尽き」る



メンバー間の議論(2)



- プロジェクトがうまく行かないので、プロマネ研修を強化したが。。。
 - PMPは増えたが、プロジェクトはうまく行っていない
 - プロジェクトマネジメントの強化は必要だが、そのやり方に問題があるのであるのでは？
- プロジェクトを成功させるには、
 - プロマネだけでなくメンバーの教育も必要
 - 「やりがい」を持たせる
 - 開発プロセスに裏付けられた
プロジェクトマネジメント力の強化



メンバー間の議論(3)



- プロジェクトはQCDだけで評価してよいか
 - プロダクトの品質だけでなく、プロセスの品質も評価すべきでは
 - プロマネ、メンバーのやりがい、達成感も評価すべきでは
- プロマネに成功体験を持たせることが育成につながるのでは
 - 小さな成功の積み重ねが重要
 - 実績を素早くフィードバックして、失敗を防止する



プロマネ拒否症を解消するには



- プロマネおよびメンバー向けの実践的なガイダンス
がって、それを習得できれば一番よい。
 - TSPは、リーダーとメンバーがプロジェクトの現場で活用できるガイダンスを提供し、実務経験とコーチング技術のあるコーチがチームを現場で支援するのが特徴である。
 - 開発プロセス(スクリプト、フォーム)とツールもある。
- プロマネになりたくない理由のうち、以下の解決にはTSPが役立つはず！
 - 「現場のプロマネをサポートする体制 / 制度が不十分」
 - 「教育 / 育成のための体制が不十分」

プロマネを組織でサポートする



TSPとは



- TSP(PSP)は以下の「守・破・離」を実践
 - まずは、TSPプロセスに従ってやってみて(守)
 - そのプロセスに問題があれば直し(破)
 - 新しいやり方を確立する(離)
- TSPおよびPSPは、当たり前のソフトウェアエンジニアリングをソフトウェア開発現場に導入するもの
 - 実際のプロジェクトで十分に活用できるプロセススクリプト(作業標準)、フォームや支援ツールも提供
 - これを使わないのでまさに「もったいない」
 - CMMはWhat、TSPはそれを実現するHowを提供



プロジェクトの成否に関するソフトウェア開発者の見方



- 参加して有意義な貢献をしていると感じる。
- 些細な成功でもマネジメントや同僚に認められ賞賛され、かつ、楽しくておもしろい環境。
- 製品とあなたがそれを開発した方法について、マネジメントと顧客からの前向きなフィードバック。
- あなたが最善と思うやり方でその仕事ができる自主性。

- Linberg; *Software Developer Perception about Software Project Failure*: Journal of Systems and Software 49 (1999): pp.177-192



TSP チームの開発者の反応



- これが本当に親密なチームだと感じた。
 - 時間を記録し追跡することは目からウロコだった。
 - このプロジェクトは本当に良いチームワークだった
作業の重複がなかった。
 - より生産的になれた。
 - プロジェクト実績が驚くほどよく分かった。
 - チームメンバーに特定の役割を割り当てるのは素晴らしい。
 - 計画を作るために、チームが本当に協力し合った。
 - チームに入っていたと感じ、権限も与えられていたと感じた。
-
- Humphrey, *PSP: A Self-improvement Process for Software Engineer*, 2005: p.323



「成功するプロマネ」育成へ



従来のパターン

プロマネ研修

PMBOK理解

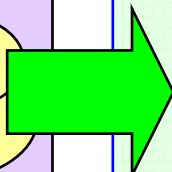


実際のプロジェクト
でどうすればいい
の？

サポートなし！
責められるだけ

失敗

プロマネ拒否症



そうか、
こうやれば
いいんだ！

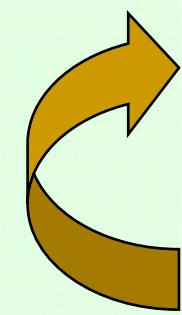


実践的ガイダンス

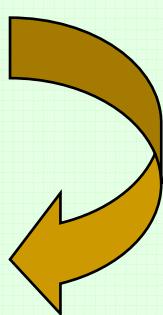


経験あるコーチ

成功



やる気



TSP導入のステップ



- 試行プロジェクトの選定
- マネジメント、チームリーダー研修
 - TSPへのマネジメントサポートを得る
 - 従来のプロジェクトとの違いを説明
- PSP研修
 - ソフトウェア開発者全員が2週間の研修を受ける
 - これを省くと、データが計測できずプロセスを守れない
- TSPサポートツールの研修
- チーム立ち上げ(コーチが支援)
- プロジェクト実施、プロセスデータを計測
- 開発サイクルの事後分析 次サイクルへ

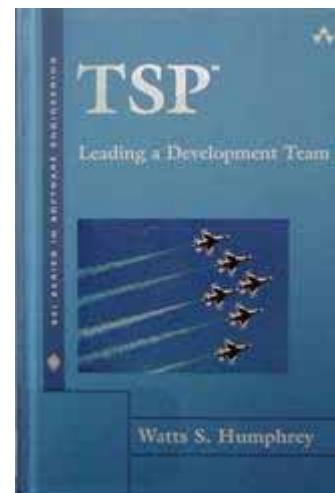


参考: プロマネのための実践的ガイダンス



- TSP: Leading a Development Teamから以下の要点をまとめた
 - プロマネの役割
 - チームの立ち上げ
 - スケジュール、コストを顧客、マネジメントと交渉する
 - リソースの確保
 - 進捗管理
 - プロセス品質
 - 要件変更への対応

ソフトウェアに限らず
現場リーダー向けガ
イドとして有用である
(出版担当者談)



プロマネの役割



■ 権限委譲

- 定義できる仕事はできるだけチームメンバーに委任
- プロマネは、本来の仕事であるマネジメントへの報告、チーム外との交渉と「予期せぬ事態」への対応に専念

■ 以下の「役割」をメンバー全員が担当

- 顧客インターフェース(要件)マネージャ
- 計画立案マネージャ
- 設計マネージャ、実装マネージャ
- テストマネージャ
- プロセスマネージャ、品質マネージャ
- サポートマネージャ

役割を自発的に
引き受ける



チームの立ち上げ



- 「立ち上げ」は表向き計画立案フェーズとなっているが、実は**自律的なチーム**としてチーム全員がまとまるチーム構築プロセス
 - チームがまとまつていれば、プロジェクトで困難が生じても乗り切れる、との考え
- 「立ち上げ」プロセスで求められる成果物をすべて作ったとしても、チームがまとまつていなければ「立ち上げ」は失敗したことに



自律的なチーム (self-directed team) とは



- 自律的なチームとは、
 - メンバーシップと帰属の意識
 - 共通のチームゴールに対するコミットメント
 - プロセスと計画に対する所有意識
 - 計画を作るスキルと、それに従う規律
 - 最高の仕事を目指す
- プロジェクトマネージャが計画を作り、それをメンバーに割り振るといった従来のプロジェクトマネジメントのあり方とは大きく違っている



スケジュール、コストを顧客、マネジメントと交渉する



- TSPチームはスケジュールとコストを「交渉」する
 - 「6ヶ月で開発してくれ」との要求にOKするまでは、マネジメントの責任
 - 無理だと知りつつ同意すれば、チームに責任が移ってデスマーチ・プロジェクトに陥ってしまう
- 交渉には、チームの計画が必要
 - 計画が出来る前に、顧客やマネジメントに「このスケジュール、コストでは出来っこない」と思わせることをしない、言わない
 - でないと、マネジメントは計画に不信感をもつ



リソースの確保



- チームリーダーはメンバーのスキルや適性、特にチームワークができるかを見極めなければならない
 - 職制上のグループやチームがプロジェクト・チームとなることが多い日本の組織ではメンバーの選択ではなく、部下のコンピテンシーを高めることが求められる
- チームメンバーは専任でなければならない。例外はデータベース専門家など限られたリソースだけである
 - 他プロジェクトとの兼任メンバーは、どちらのプロジェクト・チームにも帰属意識を持てず本当の意味でまとまりのあるチームになれない

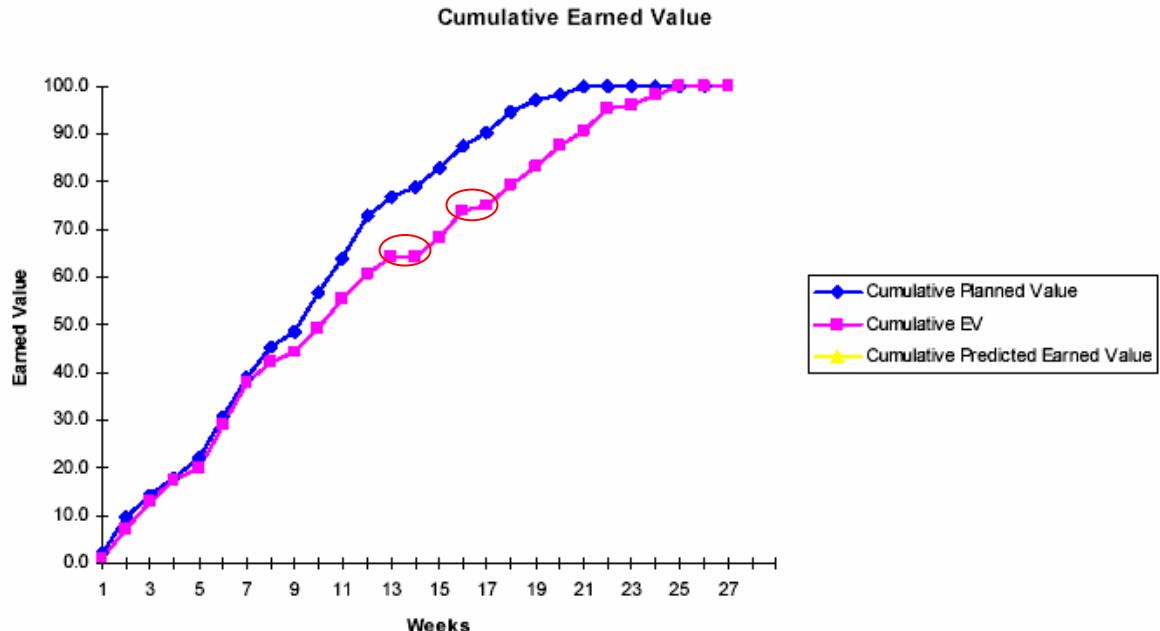


進捗管理



■ 進捗管理は獲得価値(EV)を計測して行う

- 作業の順序に無関係
- 完了予想日を計算できる
- 管理ツールが提供される



■ 定期的に報告

- 報告することで、必要なときにマネジメントの支援を得ることができる。



プロセス品質の先行指標



■ 品質計画

- プロセス品質尺度の目標値を設定
- この目標値を満たすように開発計画を調整

■ 開発時間比率をチェック

- 要件インスペクション / 要件開発 > 0.25
- 詳細設計 / コーディング > 1.00
- 詳細設計レビュー / 詳細設計 > 0.5
- コードレビュー / コーディング > 0.5



プロセス品質尺度(1)



- 品質コスト(Cost of Quality: COQ)
 - 失敗COQ: $100 \times (\text{コンパイル時間} + \text{テスト時間}) / (\text{総開発時間})$
 - 評価COQ: $100 \times (\text{設計レビュー時間} + \text{コードレビュー時間}) / (\text{総開発時間})$
 - 評価の失敗に対する比率(A/FR): $(\text{評価COQ}) / (\text{失敗COQ})$
- A/FRのガイドラインは2.0である
 - これ以下だと十分に欠陥が除去されないし、これ以上だとレビューに時間をかけ過ぎとなり開発効率が悪くなることが知られている。



プロセス品質指標(PQI)



- 下記5つの値を掛け合わせる
 - 設計 / コーディング時間 = 設計時間 / コーディング時間
 - 設計レビュー時間 = $2 * \text{設計レビュー時間} / \text{設計時間}$
 - コードレビュー時間 =
 $2 * \text{コードレビュー時間} / \text{コーディング時間}$
 - コンパイル欠陥数 / KLOC =
 $20 / (10 + \text{コンパイル欠陥数} / \text{KLOC})$
 - 単体テスト欠陥数 / KLOC =
 $10 / (5 + \text{単体テスト欠陥数} / \text{KLOC})$
- PQIが0.4以上のプログラムには統合、システムテストで欠陥が見つからなかったことが分かっている



プロセス品質指標(PQI)の使い方

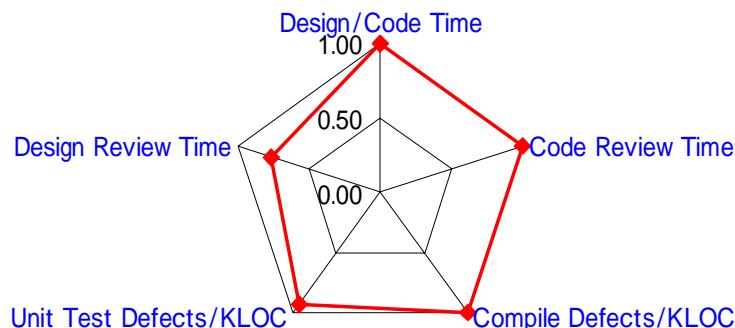


- PQIが基準値以下だと、以下の処置を決める

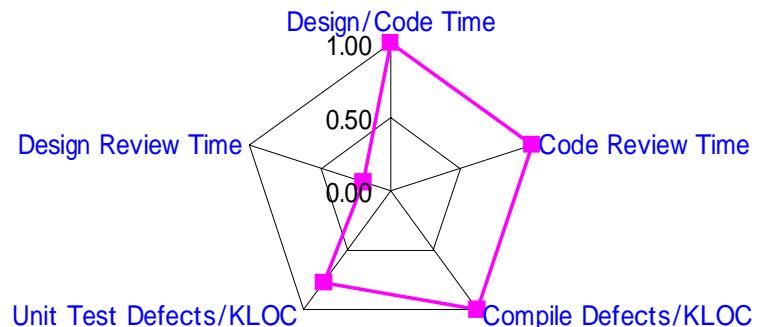
- 再インスペクションを実施する
- 作り直す
- 別のモジュールに入れ替える

品質のよくないモジュールを
統合テストに持ち込まない

PQI=0.71 Test defects=0



PQI=0.15 Test defects=1



要件変更への対応



- 要件が確定してから、と言っているといつまでも開発作業を始められなくなる
 - 今後一切変更されない「確定した要件」はないと考えよ
- 一つ一つではスケジュールにほとんど影響を与えない小さな変更が積もり積もって、計画を台無しにしてしまう
 - 小さな変更を計画やコストの見直しもせずに安易に認めてしまうと、「**変更はタダ**」と相手に言っているのと同じ
 - 修正工数の過少見積りが多い
- 初めからスケジュールやコストの変更が想定される「大幅な」変更のほうが扱いやすい！！
 - これをスケジュールやコスト差異を調整する機会になる



日本でTSPを導入する際の課題



- 日本語の分かる公認TSPコーチがない
- テキストが英語だけ
- PSPインストラクターも少ない

- でも、
 - TSP、PSPテキストの翻訳が進んでいる
 - 一部大学でPSP、TSPiの授業が始まった
 - 8月に日本でPSP/TSPi Faculty Workshopが開催され
今後、大学にPSP、TSPiコースが増えそう





ご清聴ありがとうございました

