

見える化による プロジェクトファシリテーション

～モチベーションアップのツールと場づくり～

(株)永和システムマネジメント
平鍋 健児

<http://www.ObjectClub.jp>

Copyright © 2005 Kenji HIRANABE, Some rights reserved

本日のお話

- 自己紹介(簡単に)
- プロジェクトの見える化を実践例で
- プロジェクトファシリテーション(PF)とは？
- PFの価値
- PFの原則
- PFの実践

POINT

プロジェクトファシリテーションについて、実践をまじえてお話します。

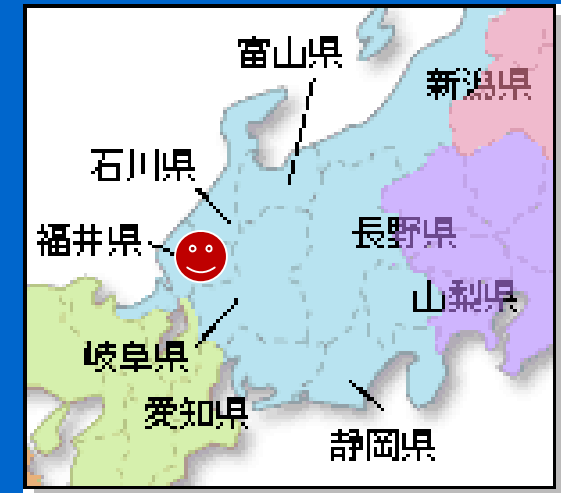
自己紹介

- (株)永和システムマネジメント

- 本社は福井県福井市, 200名
- 金融・医療・オブジェクト指向を使ったシステム開発
- 2002年より品川に東京支社

- 平鍋健児

- リアルタイム, CAD、オブジェクト指向の実践
- UMLエディタJUDEの開発
- オブジェクト倶楽部主宰
- アジャイルプロセス協議会、副会長
- XP-jpメーリングリスト主宰
- 翻訳、XP関連書籍、『リーンソフトウェア開発』、『アジャイルプロジェクトマネジメント』



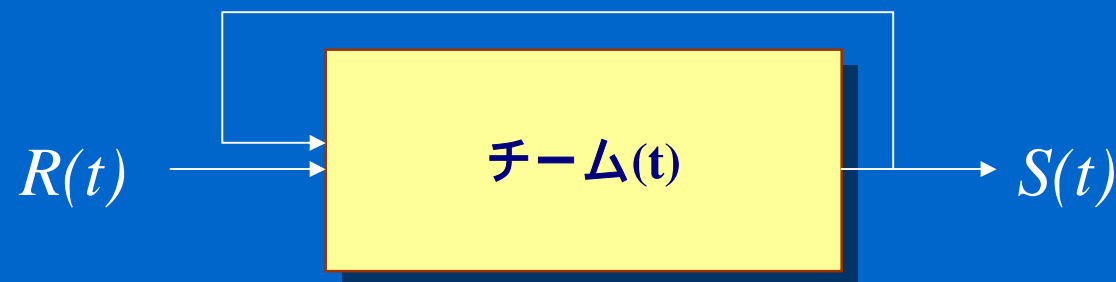
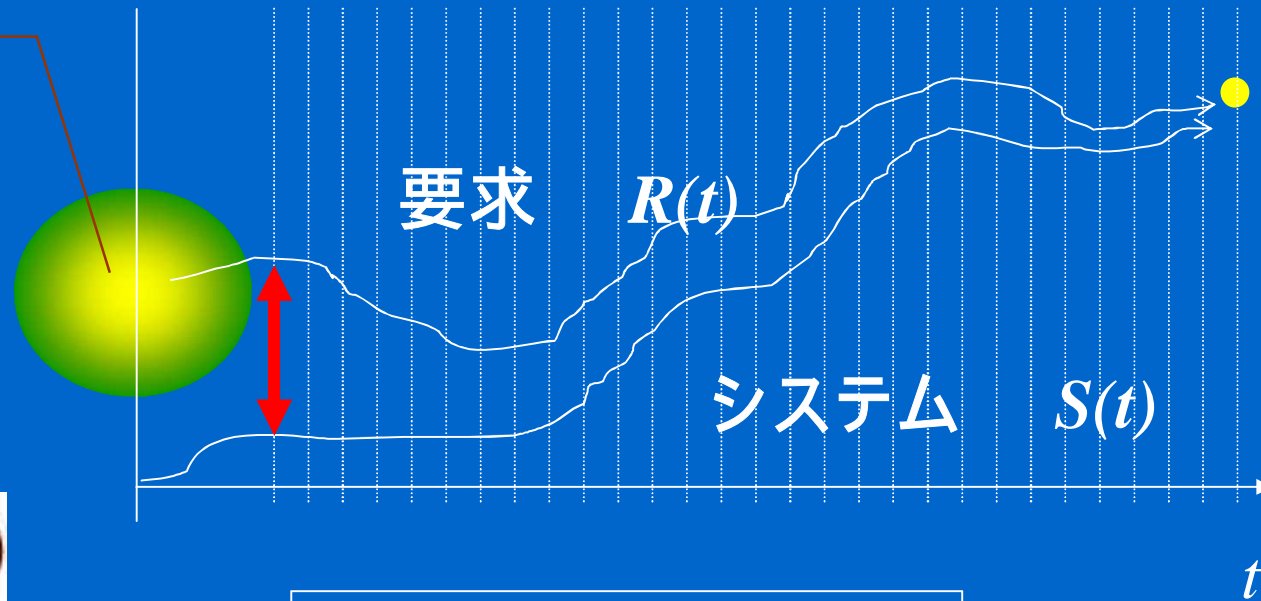
<http://jude.esm.jp/>

POINT

本日は、福井から来ました。

プロジェクトの成功は、Moving Target

不明確かつ不安定な要求。



POINT

見えなければ、制御できない。適応できない。カイゼンできない。

見える化

- 情報がパッとわかる
- 「現在の状態」も、「結果」もわかる

川之江	一	二	三	四	五	六	七	八	九	計	H	E
鈴木	和	藤	三	高	鏡	佐	高	加				
木田	原	好	雄	倉	木	石	藤					
7	4	9	5	2	1	8	3	6				
5	6	8	2	3	9	7	1	4				
仙台西	村上	鈴木	淵	横	今	藤	東	山				
上	司	木	口	沢	野	枝	林	田				

川之江	一	二	三	四	五	六	七	八	九	計	H	E
0	2	0	2	0	0	1	0	0		5	10	1
仙台西	0	0	0	0	0	0	1	0	0	1	5	2

本日の試合結果

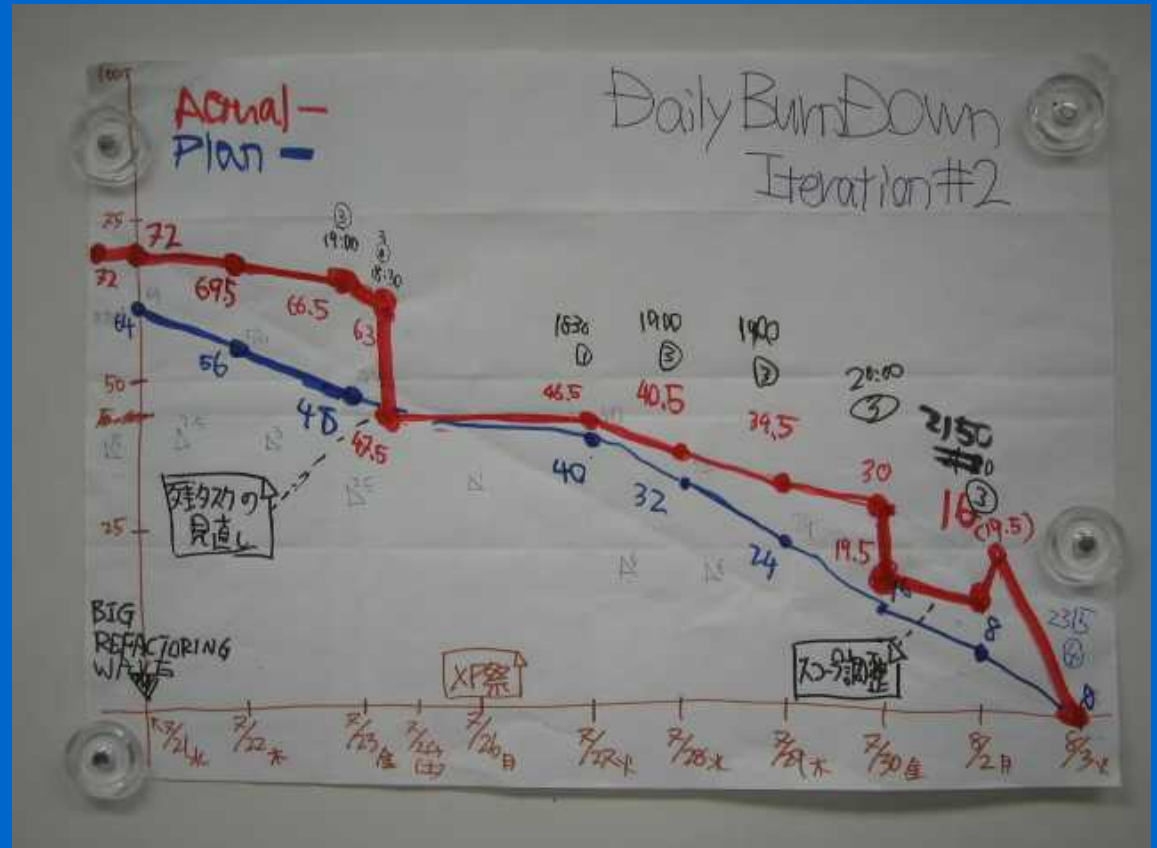
第一試合	第二試合
和智弁 5	東 邦 5
札幌一 4	大御経 3

POINT

ソフトウェア開発プロジェクトを、見える化しよう！

バーンダウンチャート

- 進捗の見える化
 - バーンダウン(下向き)
 - 中間成果物では計測しない。
 - 受け入れテストを通過した要求数でカウント。
 - メールでエクセルシートを配布したり、サーバに置いたから見てね、はナシ。



バーンダウンチャートの例

POINT

全体進捗は、「バーンダウンチャート」で見える化、繰り返しのリズムづくり

かんばん

- 作業の見える化
 - ToDo(未実施)
Doing(実施中)
Done(テスト完)
で管理。
 - 各自の作業を指示しなくても、毎朝自発的に作業開始。



ソフトウェアかんばんの例

バーンダウンチャーターなどと共に、とにかく、壁に貼る。「情報発信器」とも呼ばれる。

POINT

作業の見える化は、「ソフトウェアかんばん」で行なう。

朝会

- 作業の明確化
 - 自発的なサインアップ
 - かんばんの前で、行なう。
 - 朝の仕事はじめが重要！
 - スタンドアップで15分。
 - PF実践編：朝会ガイドを準備しています！



朝会の例

<http://www.ObjectClub.jp/community/pf/>

POINT

毎朝、「かんばん」の前で全員で短い会議を行ない、リズムをとる。

あんどん

- 異常の見える化
 - 全受け入れテストを自動化。
 - 毎時バッチで流す。
失敗があれば、即時表示。
原因追及。
 - 欠陥のムダを排除。
 - 自働化とあんどんに対応
 - 欠陥の長期滞在を排除。



あんどんの例

欠陥のムダ = 欠陥の大きさ × プロセス中の滞在時間

POINT

異常の見える化は、「ソフトウェアあんどん」で行なう。(受け入れテストを回帰)

だるま

- 目標の見える化
 - なにか形になるもので目標をしめし、達成感を味わう。
 - プロジェクトキックオフで注目を得て、偶像化する。
 - グラフや、数値でもよい。
 - 目に入ることが重要。



選挙でおなじみのダルマ

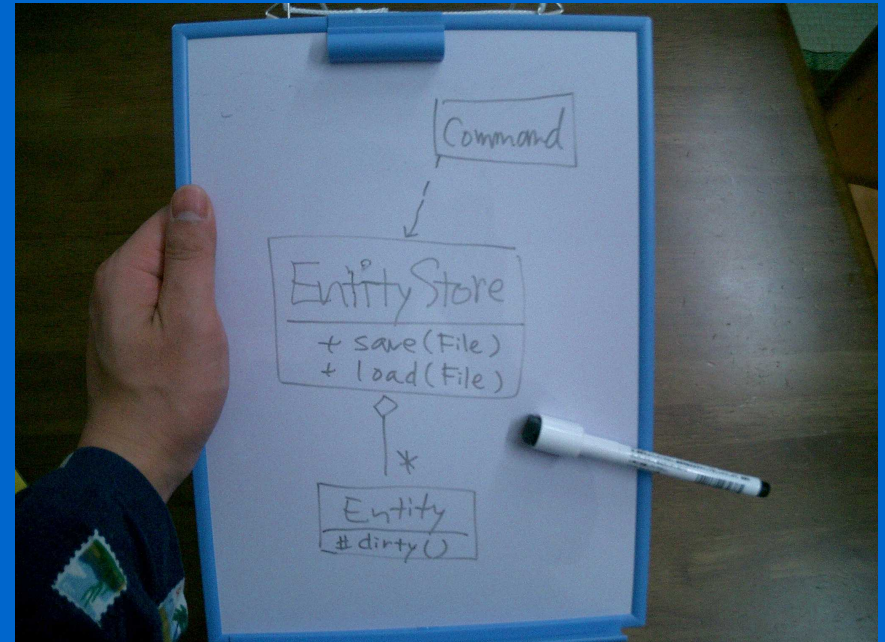
ダルマの目には、開始と終了がある。

POINT

目標を全員で共有しよう。

ペアボード

- ペアの討議内容の見える化
 - UMLなどを使って、二人の討議を見える化。
 - 議論が空中戦になるのを避ける。
 - 他の人を巻き込みやすくする。
 - ノートを捨てる。(蓄積 表現)
 - 記録は、必要ならそのままコピー！
 - 「問題vs私たち」の構図を作る。



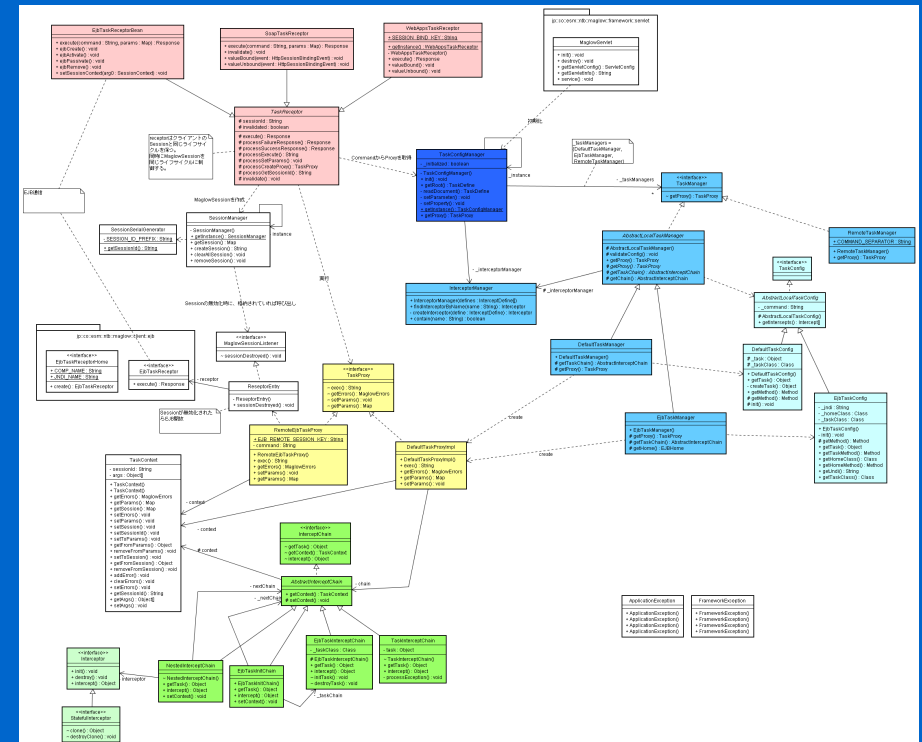
ペアボードの例

POINT

ペアの討議内容を、ボードにして見える化。ノートでなくボードで。

色つきUML

- ソフトウェア内部構造の見える化
 - UMLを使って、全員が意識する構造(アーキテクチャ、モデル)を貼り出す。
 - 手書きでもよい。
 - 色をうまく使う。
 - 図の前で議論が始まる。



ソフトウェア内部構造のUML例

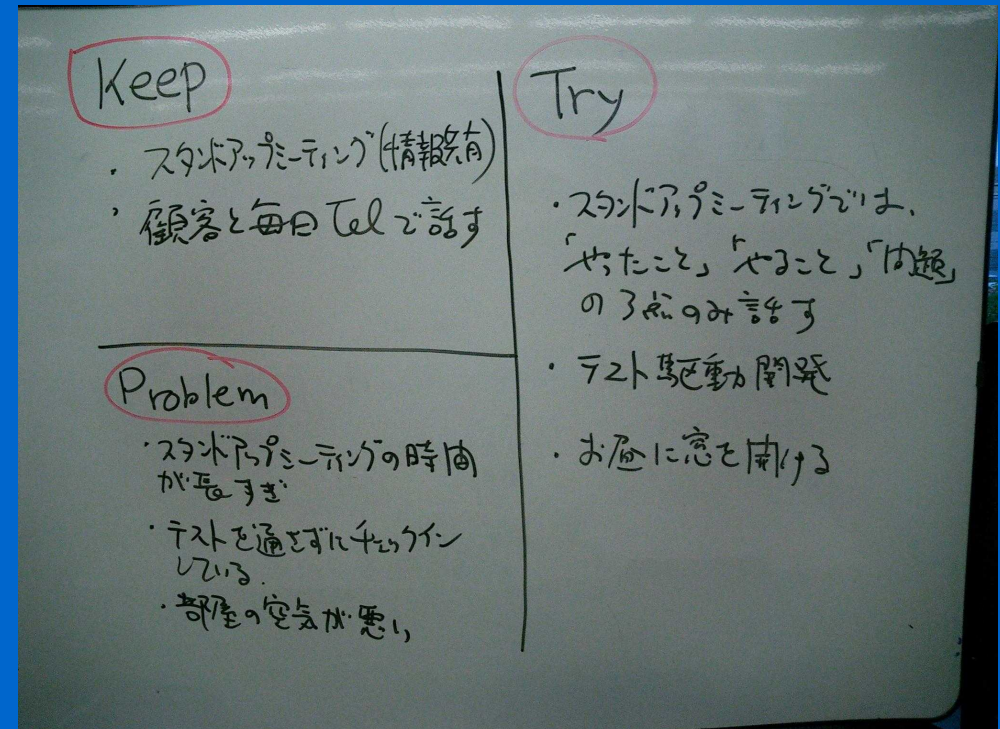
POINT

構造の見える化は、「色つきUML」で行なう。厳密でなくてよい。手書きでよい。

ふりかえり(1)

- カイゼンの気づき

- Keep(良い点)
Problem(悪い点)
Try(次回挑戦)
を出す。
- 全員で意見を出し、
暗黙知の共同化と
形式知化を行なう。「名前付け」
- 「課題 - 解決リスト」、
とは違う。
- とにかく、カジュアルな雰囲気
で全員発言することで、チームの
安全性を確保する。
- 「問題vs私たち」の構図になるように。



ふりかえりシートの例

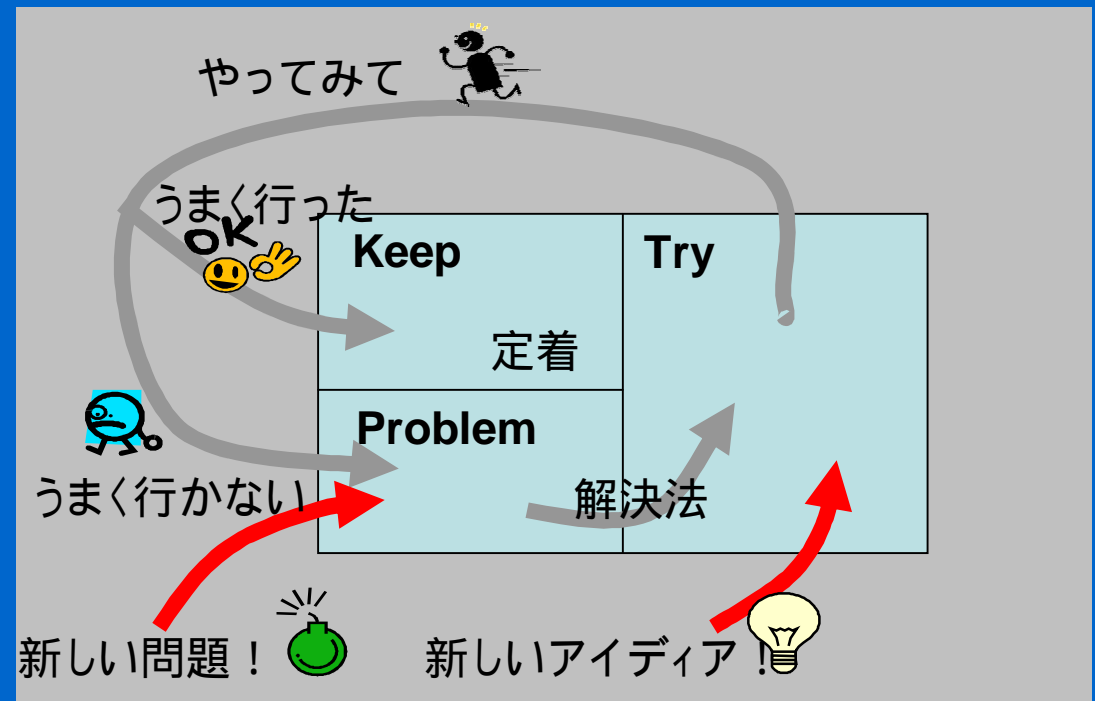
POINT

カイゼンの「気づき」を「ふりかえり」で得る。

ふりかえり(2)

- Keep/Problem/Try (KPT)

- Keepは定着する。
- ProblemはTryを生み出す。
- Tryは、KeepかProblemに移動する。
- 定着したものには、「名前づけ」を。



ふりかえりシートの例

POINT

イテレーション毎に「ふりかえり」を繰り返すことでプロセスが改善される。

ふりかえり(3)

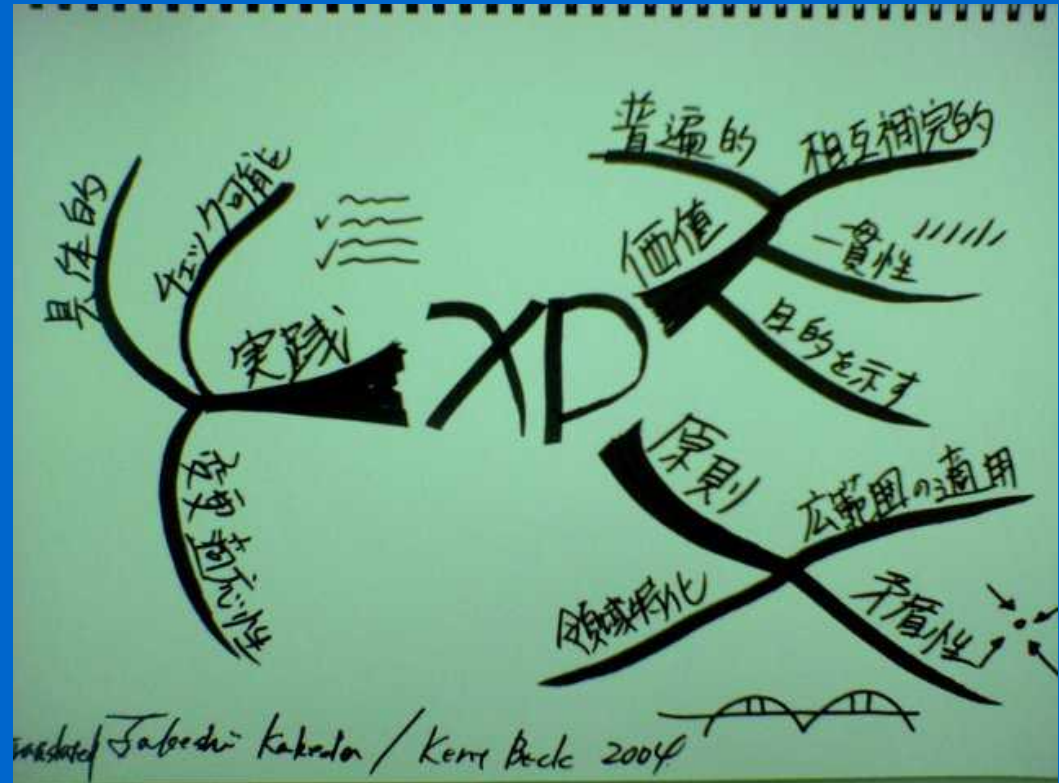
- プロジェクトやリリースの回顧
- タイムライン
 - プロジェクトを時間軸で振り返る。
 - 個々人の物語をチームの物語として表現
- 青 = 喜び
赤 = 怒り
黄 = 驚き
- 感情によって思い出を引き出す。
- プロジェクト終了のヒーリング、カタリシス、次のプロジェクトへの勇気



POINT リリース毎におおきな「ふりかえり」を。この後は、打ち上げを。(必ず！)

マインドマップ

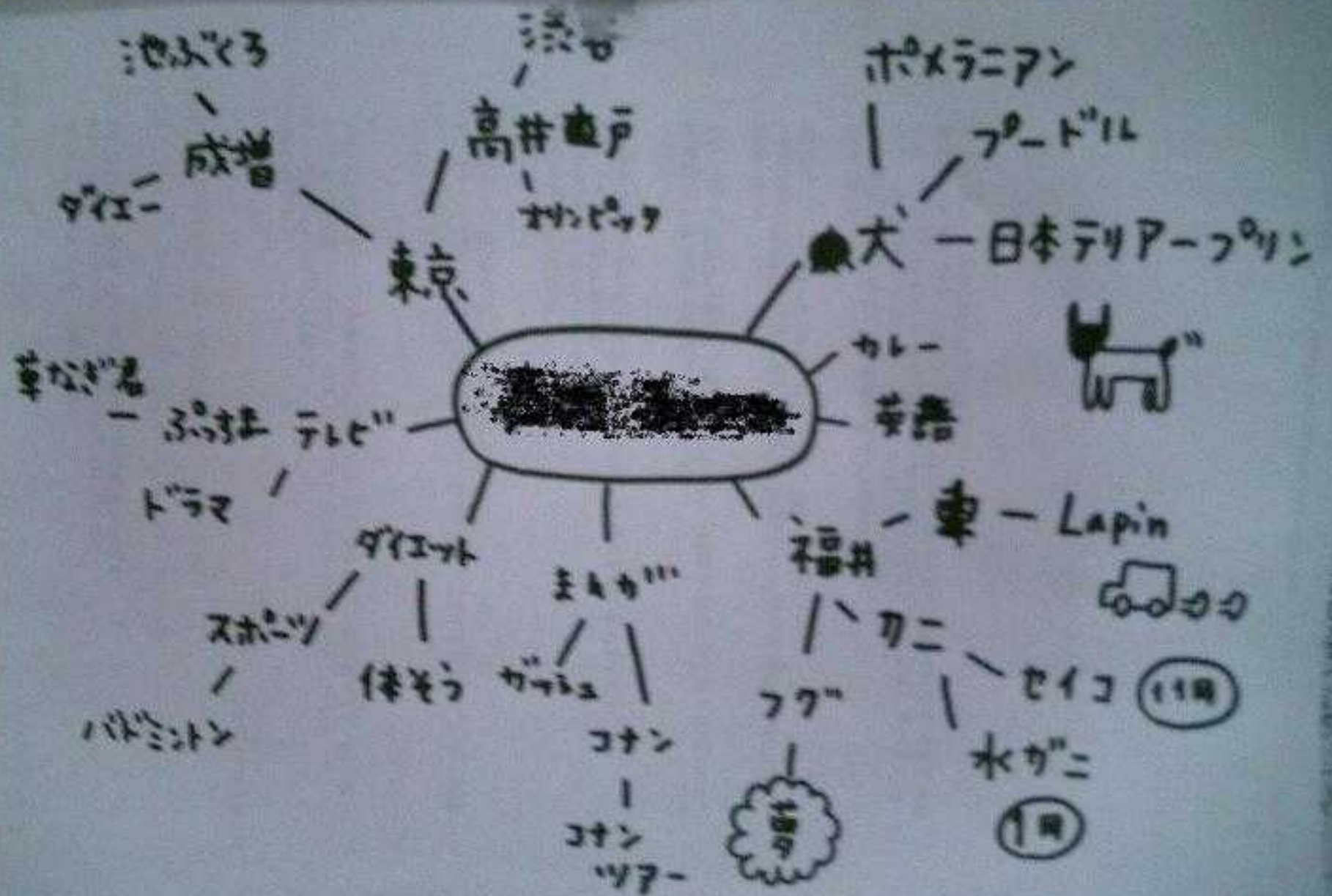
- 頭の中の見える化
 - 中心に話題をおいて
 - 放射状にキーワードを書く。
- すばやいノート術として
- 自己紹介として
- どちらかというと、
 - ブレインストーム
 - アイディア書き出し
 - アウトラインなどの発散系ツール



マインドマップの例(協力:Kent Beck、作:懸田)

POINT

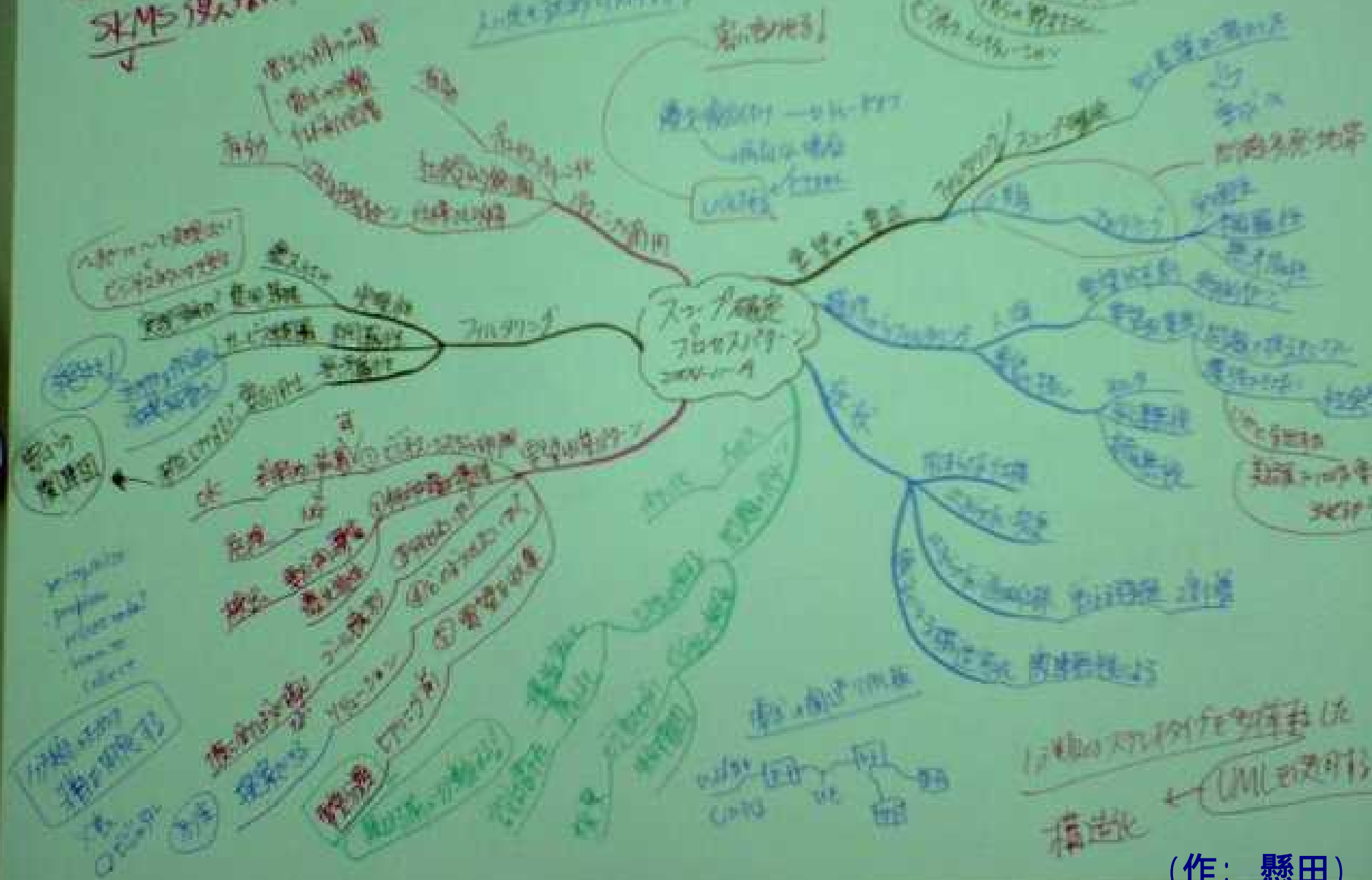
頭の中は、「マインドマップ」で見える化。



分類は
SKMS 使えん?

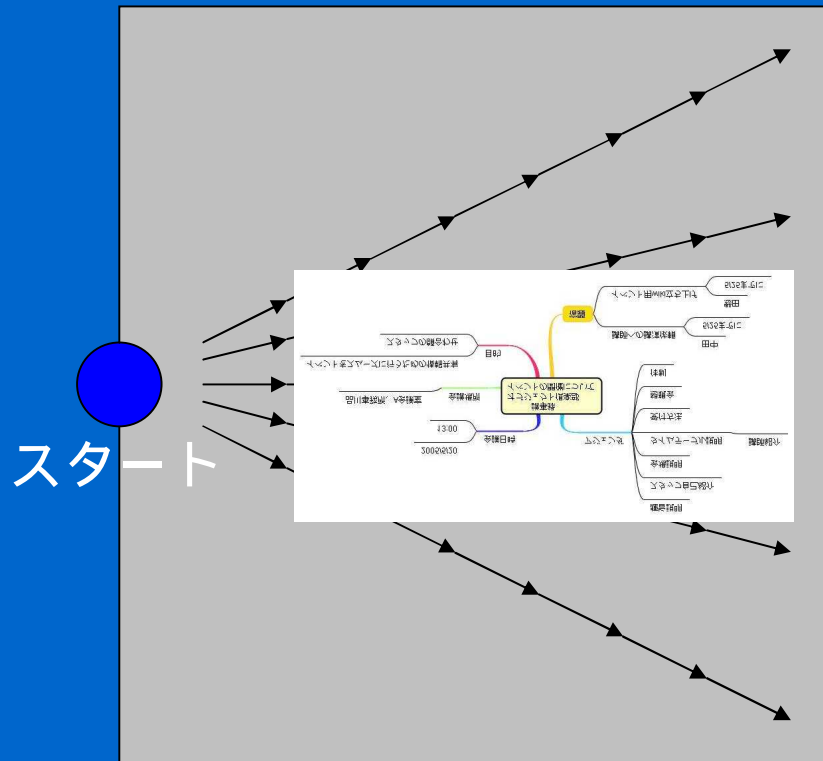
分類はSKMS 使えん?
分類はSKMS 使えん?

分類はSKMS 使えん?
分類はSKMS 使えん?

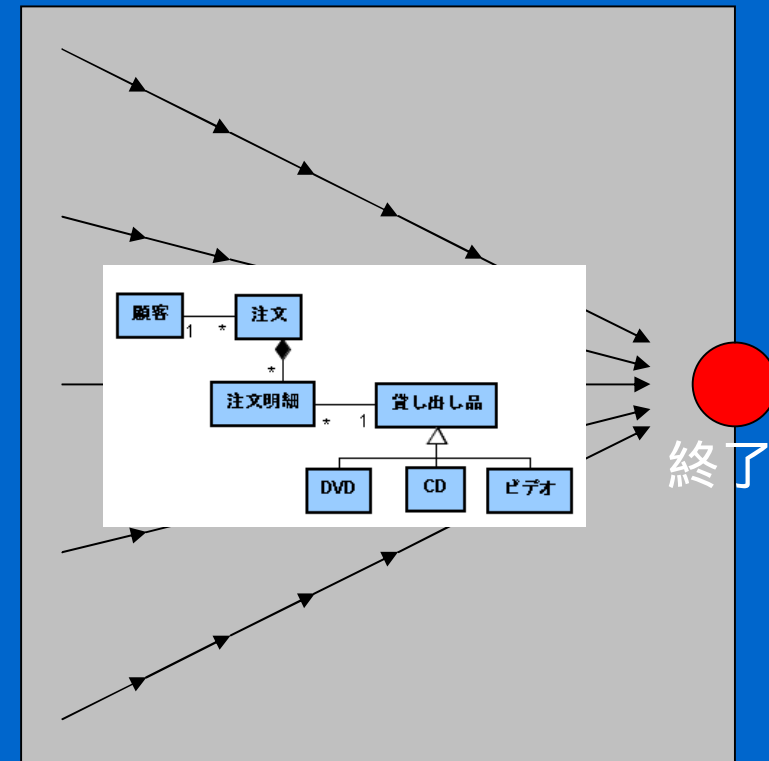


(作: 懸田)

要求分析に、マインドマップを使うという試み



思考の発散・概念の収集
「要求ギャザリング」活動
(マインド・マップの得意分野)



思考の収束・概念のモデル化
「要求モデリング」活動
(UMLの得意分野)

KY (危険予知) ミーティング

- 建設・土木現場で行なわれる、「リスク管理」の手法
- 明示的にリスクを書き出し、それに対する対策を書く。
- 担当者の名前必須。
- テスト期間、大事な日、大きなリスクがある場合、これを行なう。

月	日	危険予知活動表
グループ名		
危険の		
ポイント		
私達は		
こうする		
氏名		作業員 名

POINT

朝会で、リスクの確認を

SKMS(Structured Knowledge Management System)

- 複数人の頭の中を一気にまとめる
- 赤、青、黄の付箋紙。
- 赤 = 分類
青 = 原理・原則
黄色 = インスタンス



資産工学研究所:坂本善博

POINT SKMSは、ナレッジを構造的にまとめます。

にこにこカレンダー

- チームのムードを見える化する。
- 帰宅時の気分を、
 - － 気持ちよく仕事が終わられた
 - － フツウ
 - － ダメダメ
- チームが自発的にモチベーションマネジメント



(株)富士通ソフトウェアテクノロジーズ

実践!!IT屋のトヨタ生産方式 あるソフトウェア会社の挑戦

POINT

チームムードは、にこにこカレンダーで見える化

見えなければ行動ができない

- とにかく、「壁に貼れ」(Excelシートのメールでは見えない)
- 進捗は「バーンダウン」で。
- 日々の作業は「かんばん」で。
- 「朝会」を行い、作業を自発的に宣言。
- 異常は「あんどん」で検出。
- 「見える目標」を置いて。
- 「ペアボード」でその場で話し合いながら。
- 重要なモデルやアーキテクチャは「色つきUML」で。
- イテレーション毎に「ふりかえり」を。
- 頭の中のアイデアを「マインドマップ」で。
- リスク管理を「KYミーティング」で
- 複数人のナレッジを「SKMS」で。

POINT

見える化とリズムは、行動をうみだす第一歩。

PFとは

「ファシリテーション」とは

- 促進する、助ける、円滑にする、場を作る
 - － 個人の能力を100%以上発揮する、チームの場作り
 - － ファシリテーター：会議の司会者、案内者、議論のプロセスに責任を持つ。
 - － 例：街づくりのための市民合意形成、組織改革、プロジェクト推進
- ファシリテーターのスキル
 - － ホワイトボードの使い方、机の並べ方
 - － 司会進行、合意形成プロセス
 - － ポストイットや模造紙、マジックの使い方
 - － アイスブレーキング(会の初めに緊張を解くアトラクション)
- プロジェクト・ファシリテーション(造語)
 - － プロジェクト(ソフトウェア開発を含む)の中でのファシリテーションのあり方

POINT

PF は、ソフトウェア開発の中での「ファシリテーション」に注目しています。

アジャイルソフトウェア開発

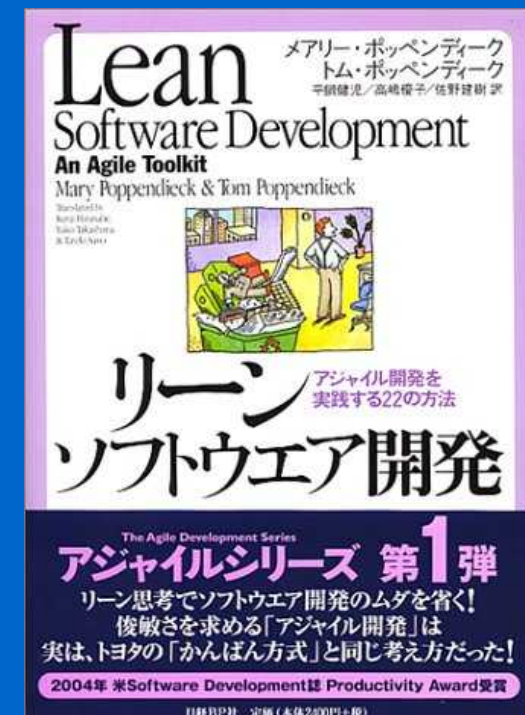
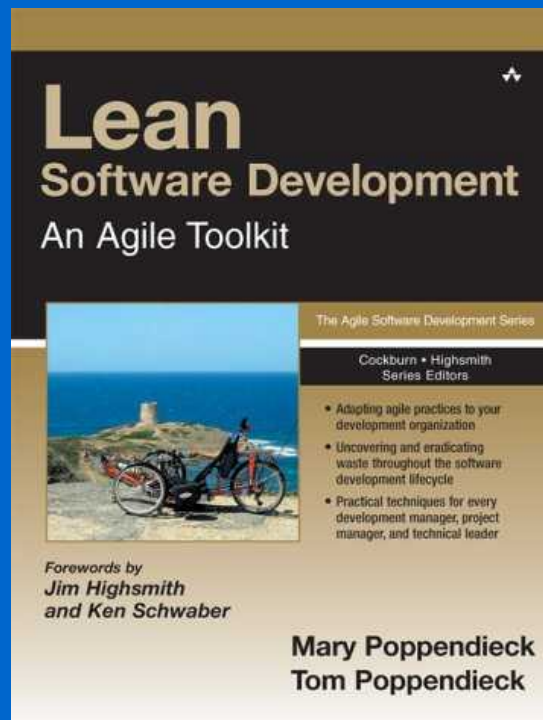
- 『XP2 (Kent Beck)』
 - 「コミュニケーション」、「シンプルさ」、「フィードバック」、「勇気」、「敬意」
 - テスト駆動開発
- 『リーンソフトウェア開発』 (Mary Poppendieck)
 - トヨタ生産方式をソフトウェア開発へ
 - もの作りはチーム作り
 - バーンダウン、かんばん、7つのムダ
- 『Crystal Clear』 (Alistair Cockburn)
 - プロジェクトを「安全地帯」へ導くチームづくり
 - プロセスからピープルへ
- 『アジャイルプロジェクトマネジメント(APM)』 (Jim Highsmith)
 - 変化に対応するチームづくり
 - 「コマンド - コントロール」 「リーダーシップ - コラボレーション」
 - 「Plan-Do」 「Envision-Explore」
- 『達人プログラマー』 (David Thomas, Andrew Hunt, Mike Clark)
 - Pragmatic Version Control/Pragmatic Unit Testing/Pragmatic Project Automation

POINT

PF は、アジャイルソフトウェア開発の一群から、ファシリテーション要素を抽出しています

リーンソフトウェア開発

- トヨタ生産方式を、ソフトウェア開発に応用



POINT

詳しくは書籍にて...

リーン思考の7つの原則

- **ムダを排除する**
 - － ムダ、とは顧客にとっての価値を付加しないもの、すべてである。ソフトウェア開発における7つのムダ(未完成作業のムダ、余分なプロセスのムダ、余分な機能のムダ、タスク切り替えのムダ、待ちのムダ、移動のムダ、欠陥のムダ)を発見し、ムダを排除しよう。
- **学習効果を高める**
 - － ソフトウェア開発プロセスは、繰り返し可能な「生産」ではなく、常に「発見」を繰り返す「学習活動」である。この学習プロセスを機能させるために、活動を見える化し、フィードバックを得ながら自己を改善していく仕組みを作ろう。
- **決定をできるだけ遅らせる**
 - － 不確定要素が多い場合、確実な情報を元に決定を下せるように、「オプション」を維持したままで前進することを許容しよう。このためには、システムに変更可能性を組み込んでおくことが戦略的に重要である。
- **できるだけ速く提供する**
 - － 「完璧主義」に陥らず、とにかく早く提供する。顧客からフィードバックを得ることで、発見と学習のサイクルが生まれる。このためにも、顧客からのプル型で開発を進めよう。
- **チームに権限委譲する**
 - － 現場の開発者が、100%の力を出せるようにする。中央集権で管理しようとしてはいけない。自発的な決定ができるようにチームをエンパワーする。見える化の手法をうまく使って、チームが自分の意思で状態を確認しながら前進できるようにしよう。
- **統一性を作りこむ**
 - － 統一性が感じられるシステムには、一貫したビジョンと思想がある。これはプロセスや手順で作ることができない。リーダーシップとコミュニケーションが、統一性の源泉となる。
- **全体を見る**
 - － 部分最適に陥ってはならない。個人や一組織のパフォーマンスのみで評価すると、部分最適が起ころてしまう。一つ上のレベルで評価するようにし、個人や組織の協調が生み出されるようにしよう。

POINT

7つの原則が、プロセスを編み出すためのガイドライン。

ムダを認識する

- トヨタ生産方式の「7つのムダ」とソフトウェア開発をマッピング

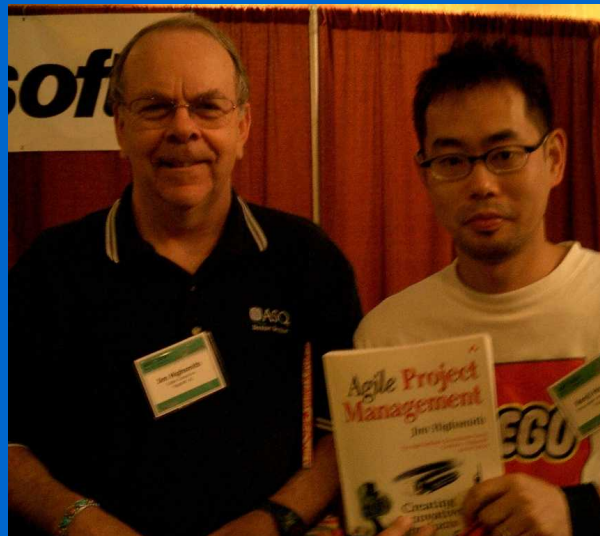
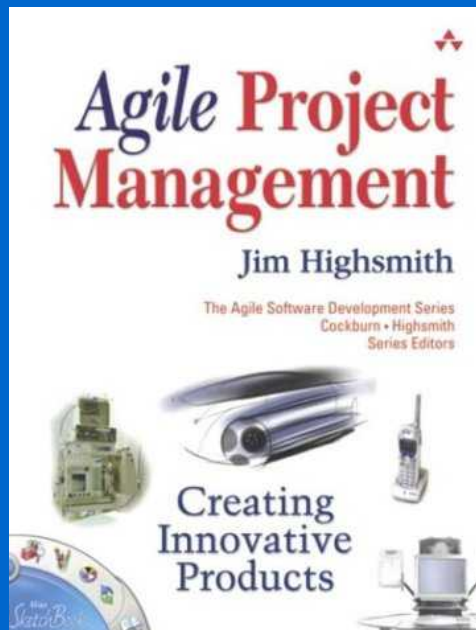
生産工程の7つのムダ	ソフトウェア開発の7つのムダ
在庫のムダ	未完成の作業のムダ
加工そのもののムダ	余分なプロセスのムダ
作りすぎのムダ	余分な機能のムダ
運搬のムダ	タスク切り替えのムダ
手待ちのムダ	待ちのムダ
動作のムダ	移動のムダ
不良を作るムダ	欠陥(バグ)を作るムダ

POINT

まずムダを認識。顧客価値に結びつかない、「すべて」がムダ。

アジャイルプロジェクトマネジメント

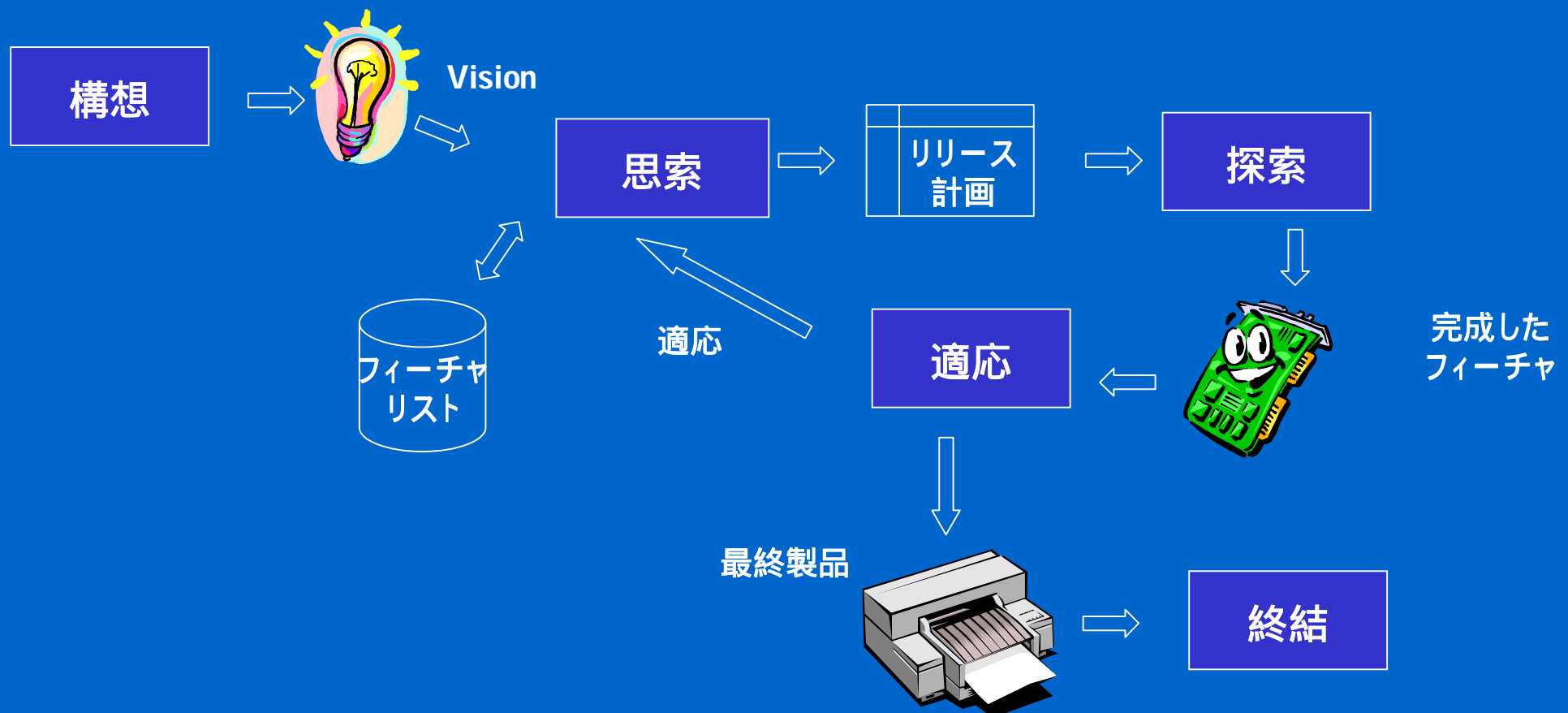
- アジャイル開発を、新製品開発(ソフトウェアに限らない)に拡大。イノベーションをつくるマネジメントとは。
- 計画・実行ではなく、構想・探索
- コマンド + コントロールではなく、リーダーシップ + コラボレーション



POINT

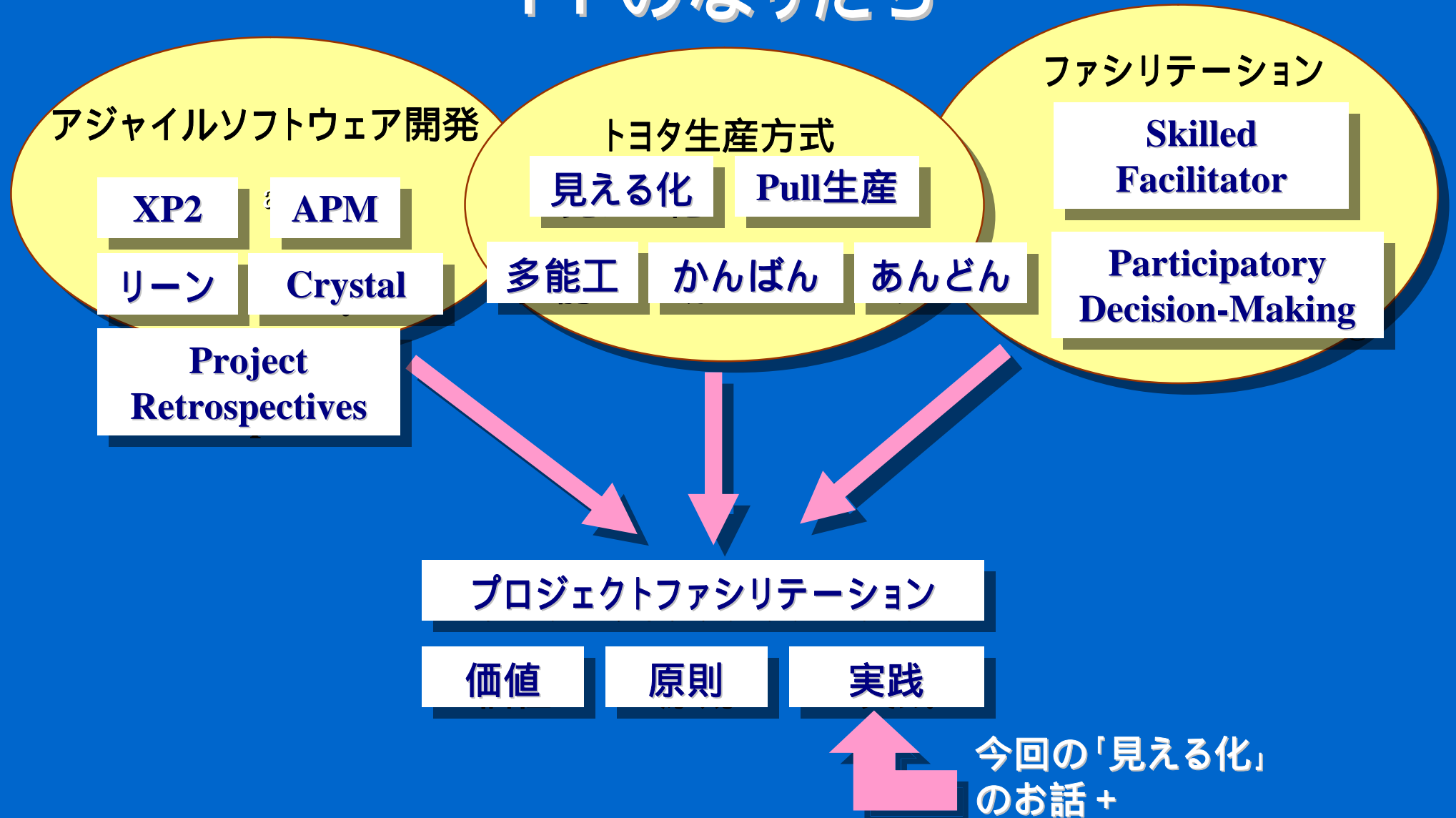
新刊です。アジャイル開発の集大成がつまっています。

APMのライフサイクル



価値	変化への対応、動作する製品、顧客との協調、個人と対話		
原則	製品の提供	反復型で機能ベースの提供、顧客価値の提供、技術的優位性の重視	
	リーダシップ - コラボレーション	適応型チームの構築、探索の奨励、シンプルにする	
プロセス フレームワーク	<div><div>構想</div><div>思索</div><div>探索</div><div>適応</div><div>終結</div><div></div></div>		
プラクティス	構想フェーズ	ビジョン	製品ビジョンボックスとエレベーターテスト
		スコープ	プロジェクトデータシート
		コミュニティ	適任者の確保
			関係者の特定
			顧客チーム・開発チーム間のインタフェース
		アプローチ	プロセスやプラクティスのテラリング
	思索フェーズ	FBS(機能ブレークダウン構造)	製品の機能リスト
			機能カード
			性能要求
		反復型計画	リリース、マイルストン、イテレーション計画
	探索フェーズ	目的の実現	作業負荷の管理
		技術的プラクティス	低コストでの変更
		コミュニティ	コーチングとチーム作り
			毎日の統合ミーティング
			参加型意思決定
			毎日の顧客との対話
	適応フェーズ		製品、プロジェクト、チームのレビューと適応

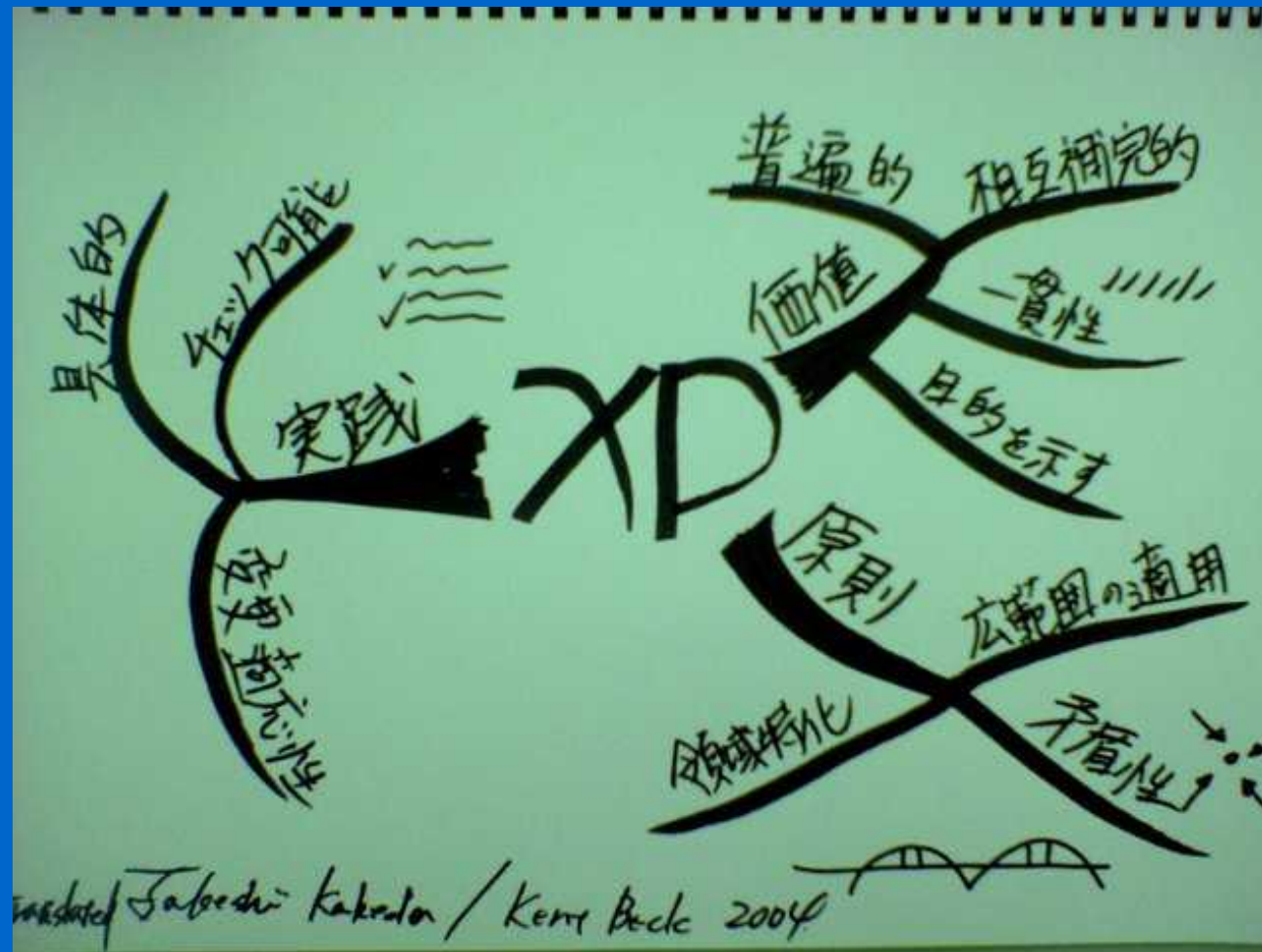
PFのなりたち



POINT

FP=アジャイル + TPS + ファシリテーション。ソフトウェア開発以外に適用可能。

価値、原則、実践



目的：なぜPFが重要か

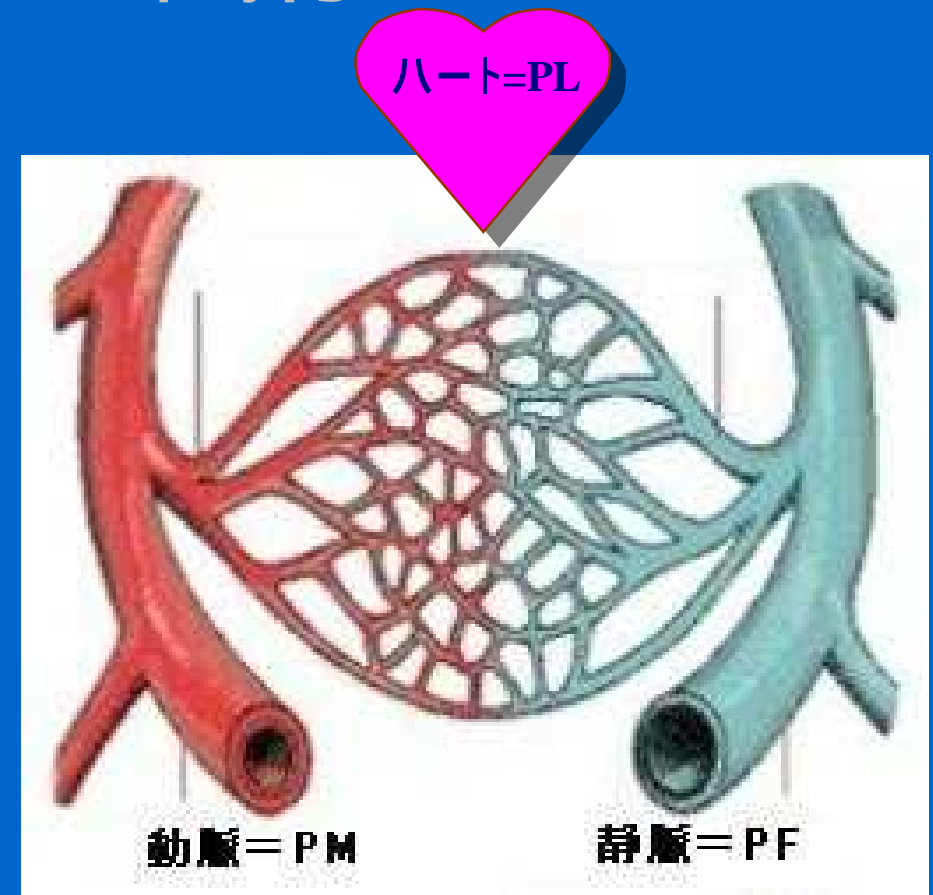
- プロジェクトを成功させるために。
 - － 行動を起こさせるために。
 - － ひとりひとりの能力を最大限に発揮させるために。
 - － 個人の総和以上の価値をチームとして発揮するために。
- エンジニアとして、よりよい人生の時間をすごすために。
(Quality of Engineering Life: QoEL の向上)
 - － 気づきを得るために。
 - － 仕事の中で、プロジェクトを越えて続く人間関係を得るために。
 - － やりがいと笑顔と信頼関係で、プロジェクトに取り組むために。

POINT

PF は、ソフトウェア開発を成功させ、あなたの人生の質を向上させます。

PM, PF, PLの関係

- PMは目標達成のために重要
- PFがないと, 行き詰ってしまう
- もう1つ, PL(プロジェクトリーダーシップ)がある.
- PM, PF, PLの素養が同一人格にあることは稀.
 - PM プロジェクトマネージャ
 - PL プロジェクトリーダー
 - PF プロジェクトファシリテータ



POINT

PFとPM は、相補関係です。両方とも, PLあつての仕事

PFの5つの価値

● コミュニケーション

- 必要な人と、必要を感じたときすぐ、対面で話をしていませんか？
- チームとして個人の総和以上の成果を上げるために、「コミュニケーション」を価値とします。

● 行動

- あなたの言葉に、行動はともなっていますか？
- 価値を現実のものとするために、そして気づきを得るために、「行動」を価値とします。

● 気づき

- 今日、何かに気づきましたか？気づきを、毎日誰かに話していますか？
- 個人そしてチームが成長するために、「気づき」を価値とします。

● 信頼関係

- あなたはチームのメンバーを信頼していますか？チームのメンバーはあなたを信頼していますか？
- 行動を起こす勇気の源として、「信頼関係」を価値とします。

● 笑顔

- 人からの非難をおそれてびくびくしていませんか？冗談を言える雰囲気はありますか？今日、みんなの笑顔は見えますか？
- 人生の貴重な時間を楽しくすごすために、「笑顔」を価値とします。

PFの4つの原則

- 見える化(Management by Sight)
- リズム(Rhythm)
- 名前づけ(Name and Conquer)
- 問題 vs. 私たちの構図(Problem vs. us)

見える化

- 情報がパッとわかる
- 「現在の状態」も、「結果」もわかる



POINT

ソフトウェア開発で使える、見える化手法のヒントは、まだたくさんある。

リズム

- リズムを「デザイン」する
 - － 会議
 - － 成果物のタイミング
 - － 日時のテスト
 - － 日時の朝会 (毎朝 10:00)
 - － 週次の会議 (毎週金曜は。。。)
- 朝会、ふりかえりのタイミング
- リズムが行動の「搬送波」



リズムがチームのハートビート

POINT

リズム(チームの鼓動)をデザインして、チームを前進させよう

名前付け

- 「気づき」をキャッチ
- ナレッジを,
 - 定着
 - 他のチームに伝播
- 例:
 - 「今日のお仕事」(by 坂田さん)
 - 「ぬかどこ」(by 倉貫さん)
 - 「にここカレンダー」

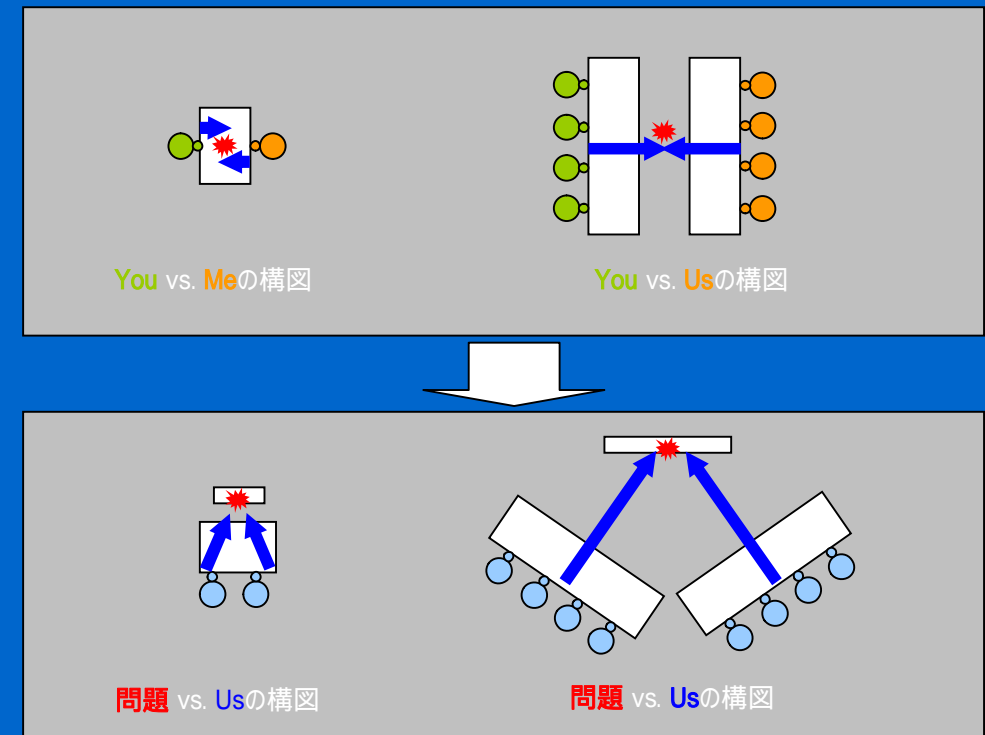


名前は大切(写真協力:平塚市博物館)

POINT 名前をつけないと、「気づき」が逃げて行っちゃう！

問題対私たちの構図(Problem vs. Us)

- とすると、議論は
You vs. Me
You vs. Us になりがち。
- Problem vs. Usにもちこむ。
 - ホワイトボードを使う
 - 座り方を替える
 - ペアプログラミング

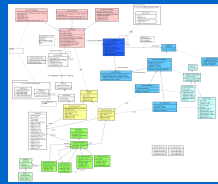


POINT

不毛なゼロサムゲームから、生産的な議論へ向かうカギ。

PFの实践

全体構成



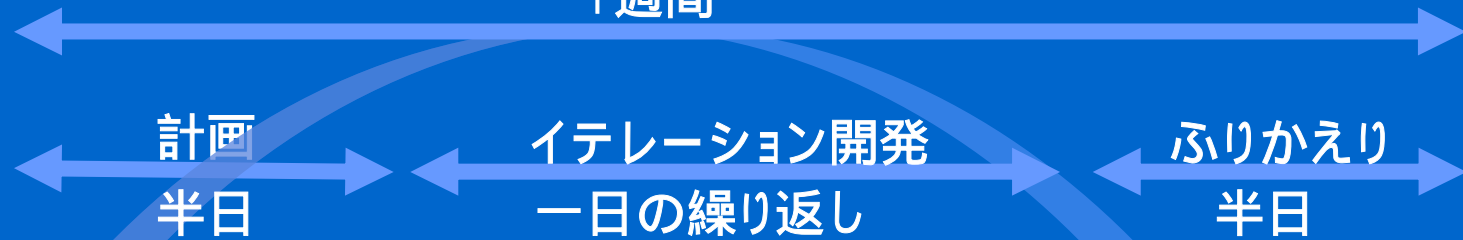
色つきUML



見える目標

1週間

- 見える化
- リズム
- 名前づけ
- 問題対私たちの構図



要求

タスク

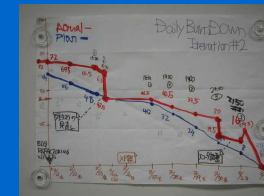
タスク

タスク

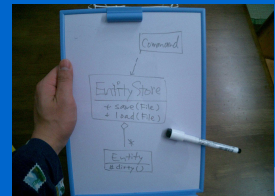
タスク



朝会、かんばん



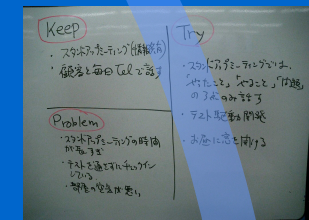
バーンダウン



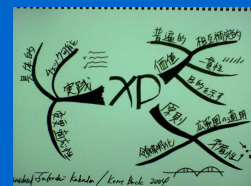
ペアボード



あんどん

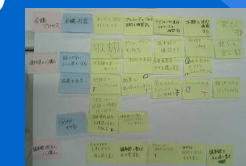


ふりかえり

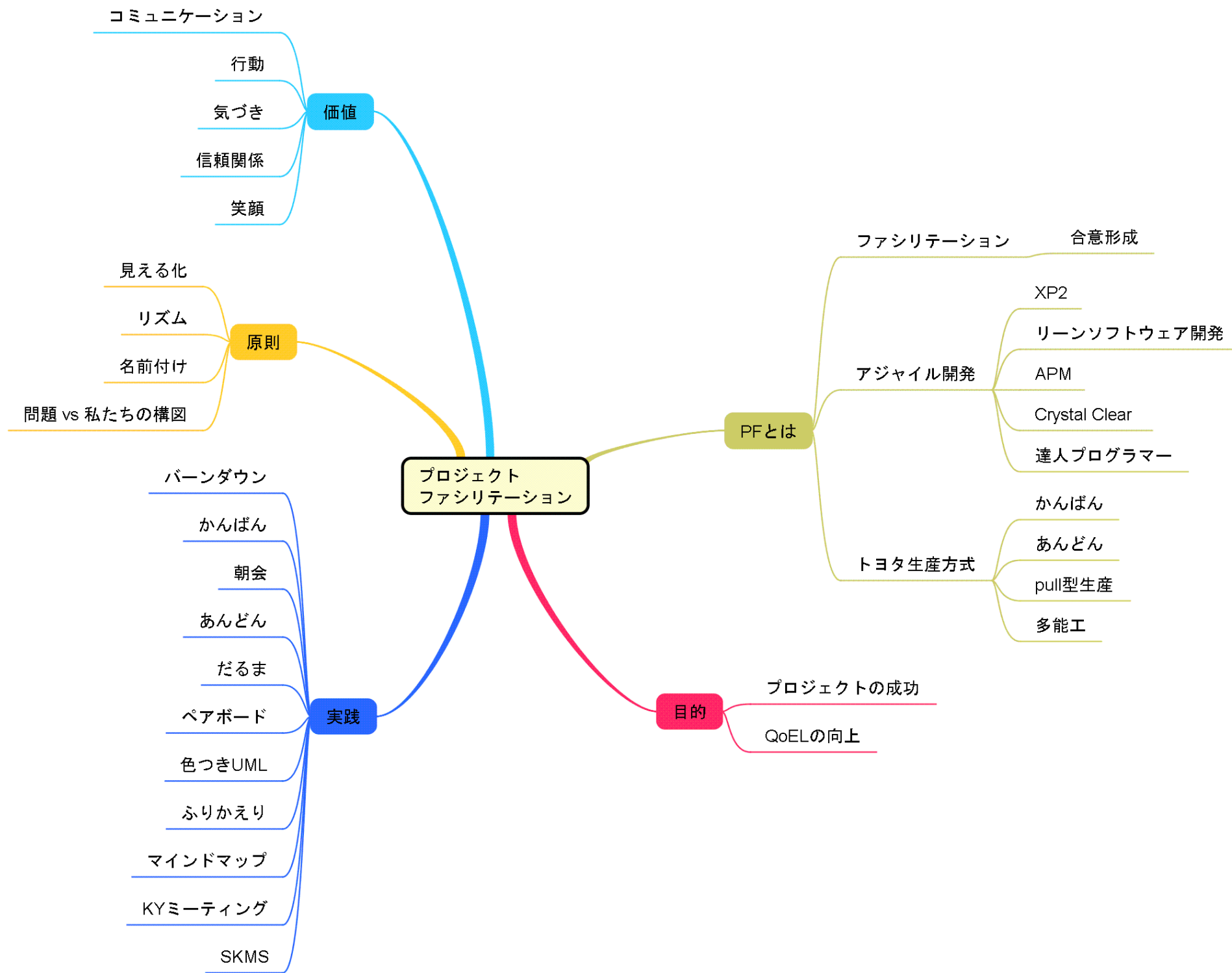


マインドマップ

テスト・ペアプロ・リファクタリング



SKMS



人々に学び

人々と一緒に計画し

人々が持っているもので始め

人々が知っていることの上に築きなさい。

リーダーが真に優れていれば、

終わってみると

人々は口々にこういう

「自分たちの力でやり遂げた」と。

ご清聴ありがとうございました。

[http://www.ObjectClub.jp/
community/pf/](http://www.ObjectClub.jp/community/pf/)