現場力を高める見える化手法
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～

株式会社永和システムマネジメント
株式会社チェンジビジョン

平鍋 健児

http://www.ObjectClub.jp

今日のお話の要点

- アジャイル開発の中には「現場活性化のヒント」がたくさんあります。これらは、ウォーターフォール、繰り返し型に関係なく使えます。
- さらに、これらの工夫は生産革新の現場で行われている「トヨタ生産方式」と本質が同じもので、「見える化」を基本にしています。
- 私の関係するプロジェクトの現場で、実際に行われている「見える化」の例を、写真を交えてお話します。
- さらに、その背後にある、「原則」と「価値」をお話します。
- これらを、私がミッションとしている開発現場の活性化手法として、プロジェクトファシリテーションという名前をつけ、体系化しています。
- みなさんの開発の中でも、利用できるものがあったらぜひ使ってみてください。また、新しいアイディアが発見できたら教えてください。

本日のお話

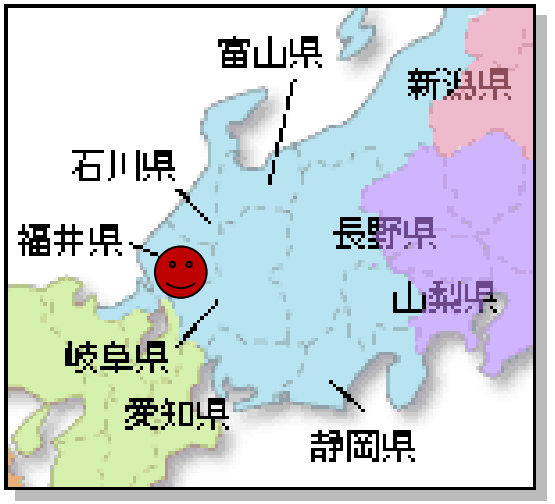
- 自己紹介（簡単に）
- プロジェクトの見える化を実践例で

- 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

- ㈱永和システムマネジメント
  - 本社は福井県福井市
  - 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
- 株式会社チェンジビジョン
  - 本社は東京都新宿
  - JUDE と TRICHORD で見える化
- 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
  - 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』



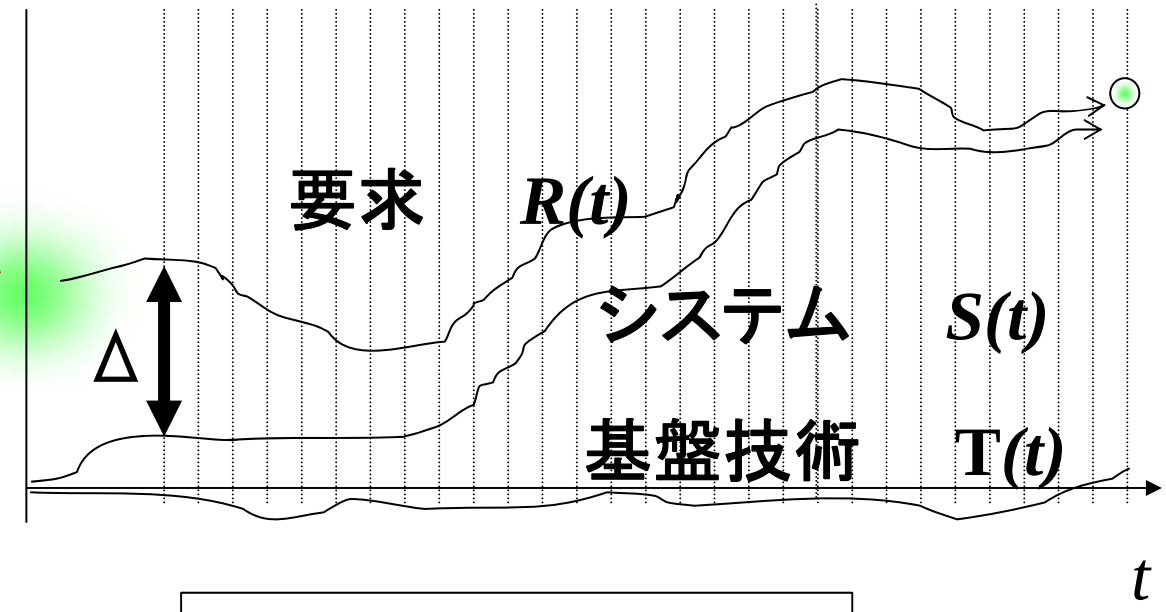
http://www.change-vision.com/

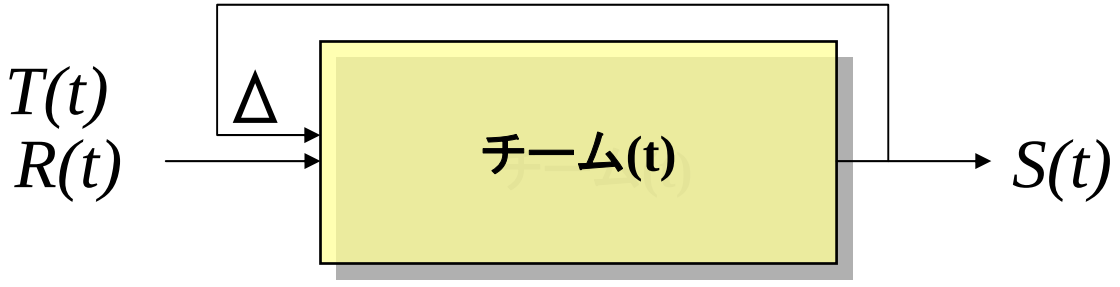
ChangeVision

なぜ
見える化か？

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

- 「最新の正の情報」が、「一箇所に」、「大きく」書かれていて、それを、「両チームのメンバー」、「審判」、「観客」が見ている。「次の行動」を誘発する。

POINT 全ステークホルダーが、行動を起こせるような、確かな、分かりやすい情報源

実践

タスクかんばん

- 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。
  - フォーマットは徐々にカイゼン。

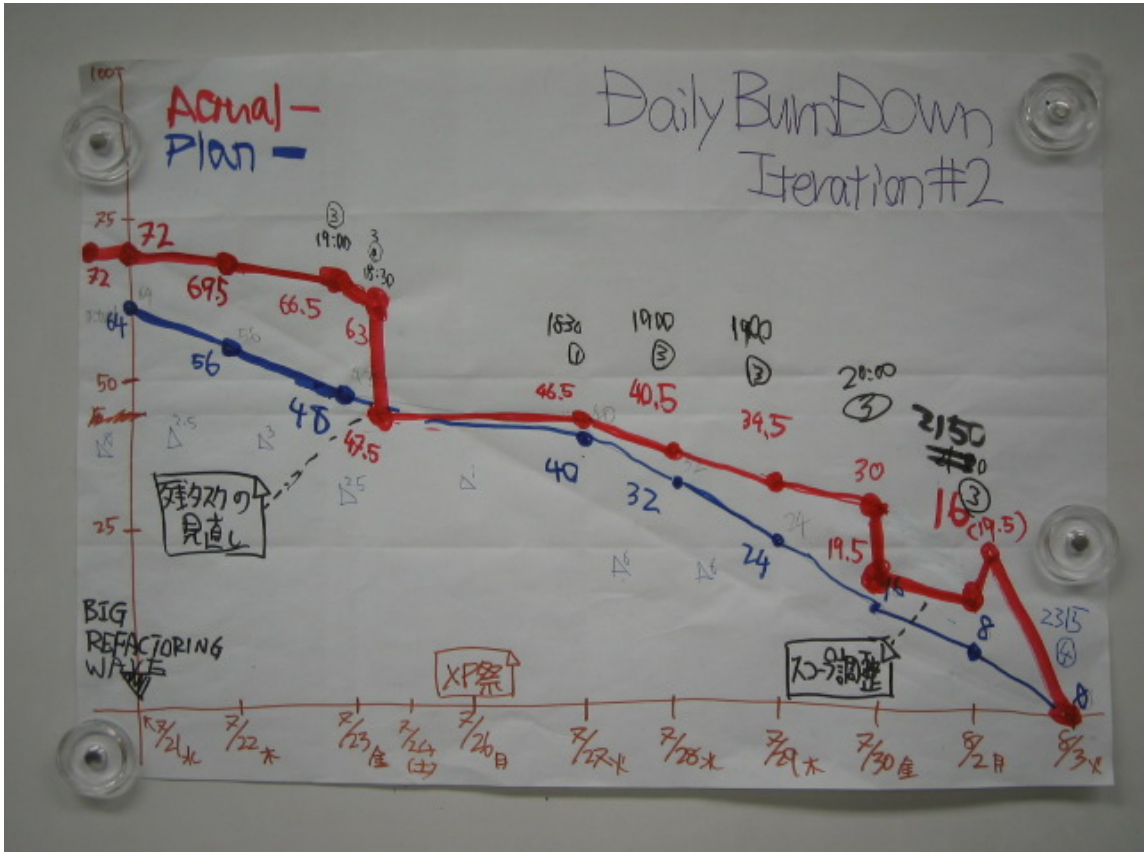
タスクかんばんの例

※バーンダウンチャーなどと共に、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、「タスクかんばん」で行なう。

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

ポータブルかんばん

(協力：CCS 佐藤竜一）

POINT 「かんばん-nano」スーツにもベストフィット！

（協力：ヤマハモーターソリューション）

朝会

- 作業の明確化
  - 自発的なサインアップ
  - 昨日やったこと、
    今日やること、
    問題点、の３点のみ。
  - 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.
  - 定時刻、定位置、短時間

朝会の例

PF実践編：朝会ガイド
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
  - 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

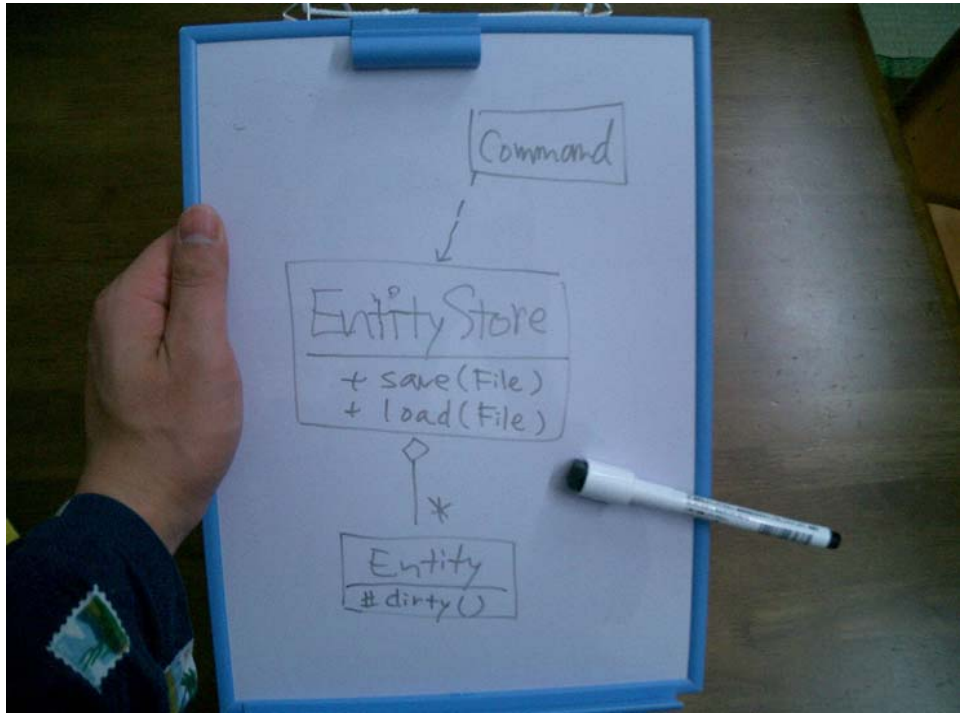
※　欠陥のムダ＝欠陥の大きさ×プロセス中の滞在時間

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）

協力：ヤマハモーターソリューション

My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。
  - 自作すると、なおよい。（200円）

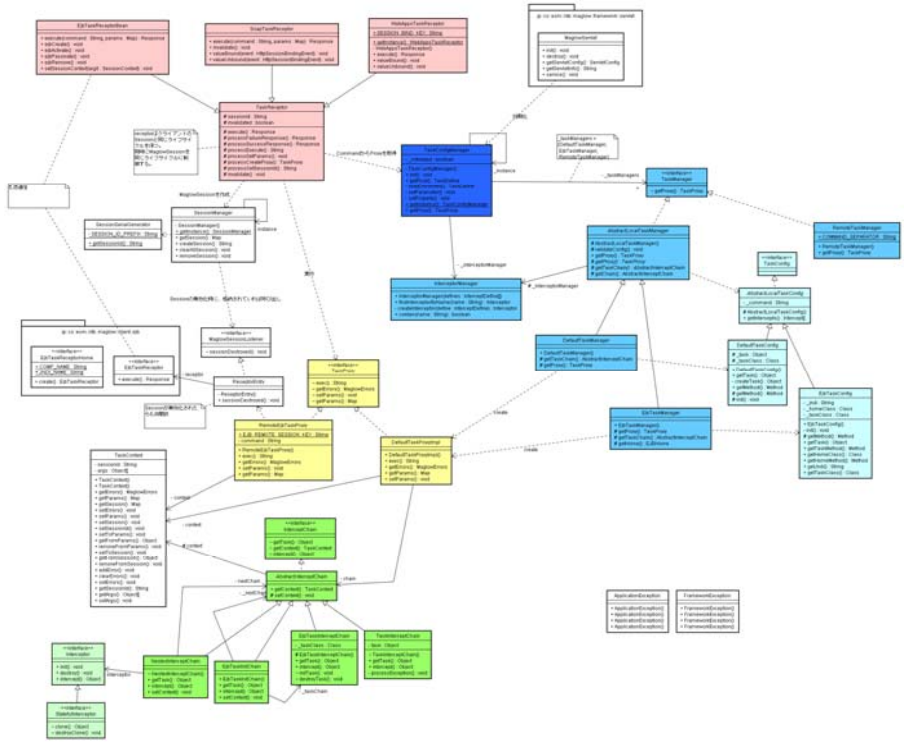


ペアボードの例(600円)

POINT 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。
  - UMLでなくてもよい。
  - 色をうまく使う。
  - 図の前で議論が始まる。

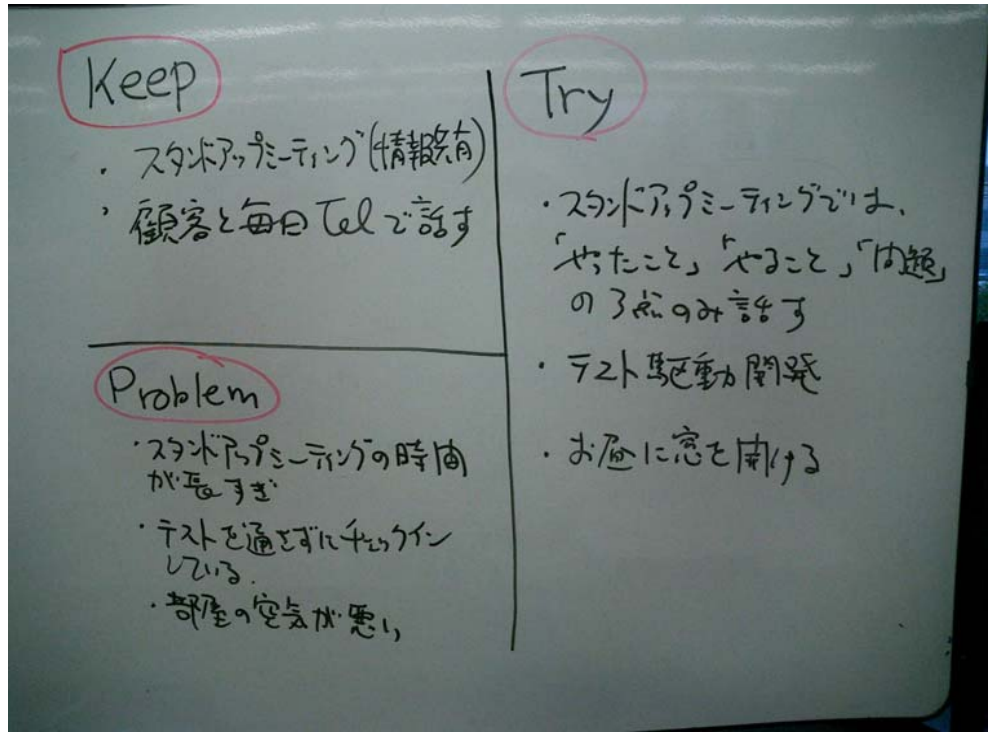


ソフトウェア内部構造のUML例

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」
  - 「課題ー解決リスト」、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。



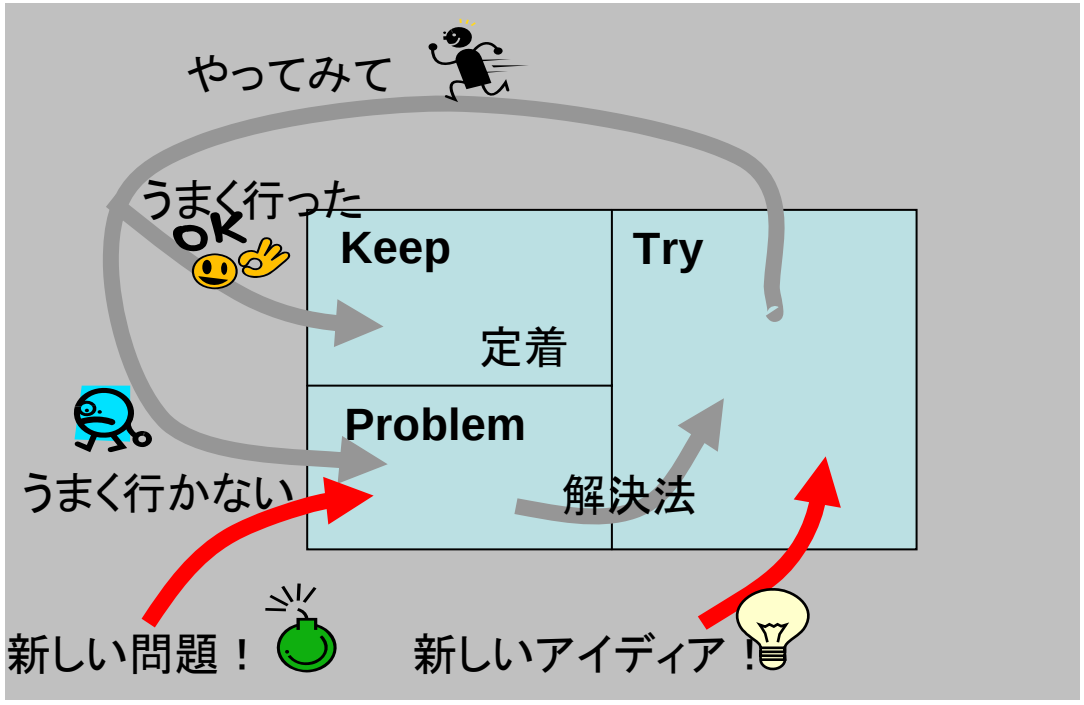
ふりかえりシートの例

実践編:ふりかえりガイド
http://www.ObjectClub.jp/community/pf/

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。



ふりかえりがカイゼンを導く

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

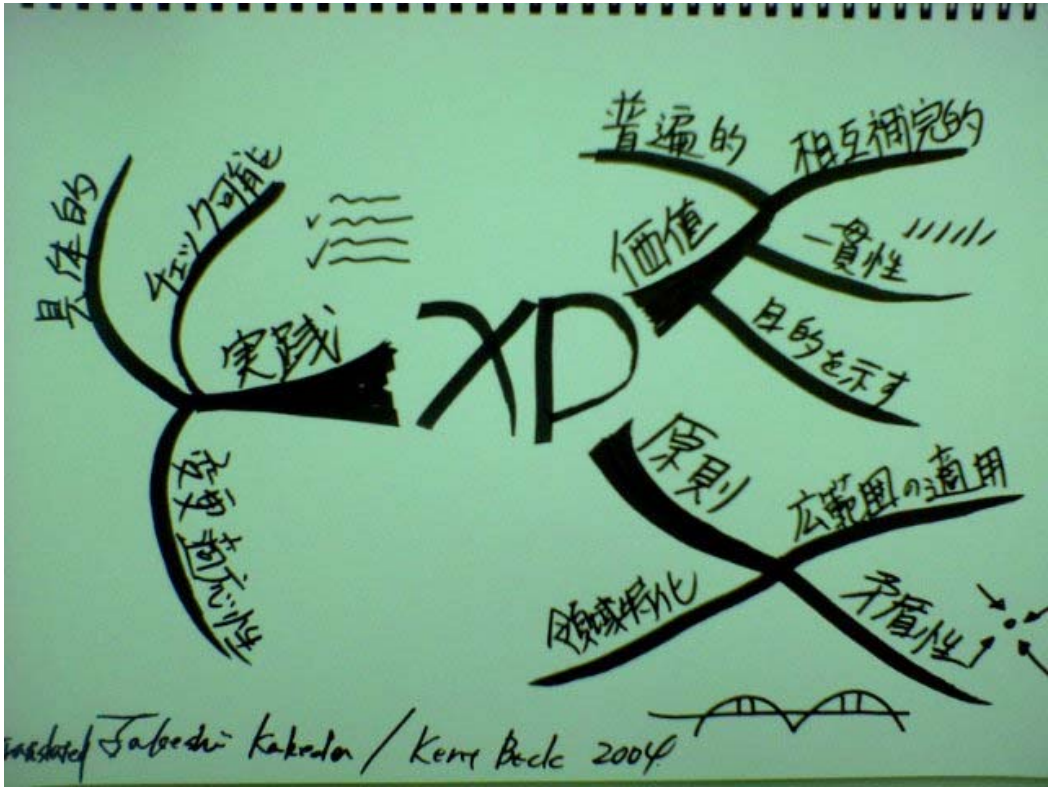
タイムラインの例

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

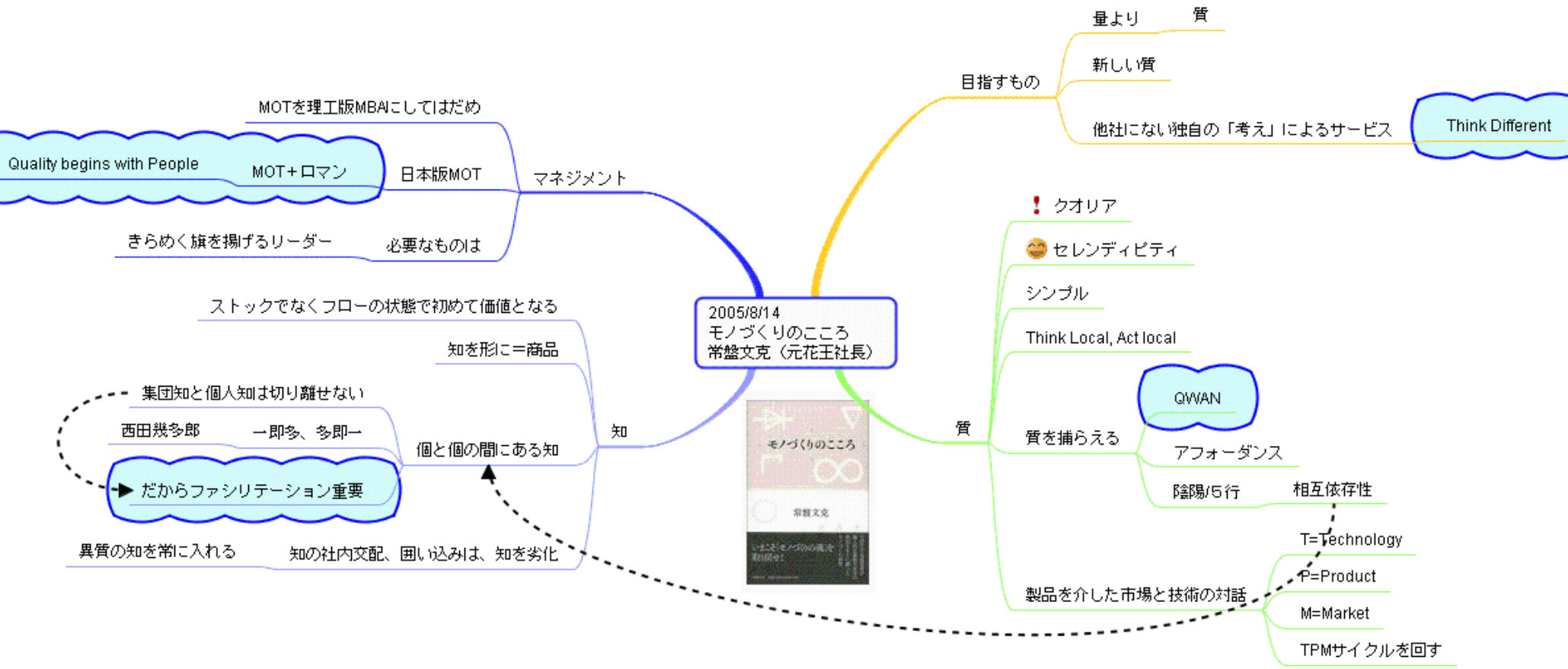
  などの発散系ツール



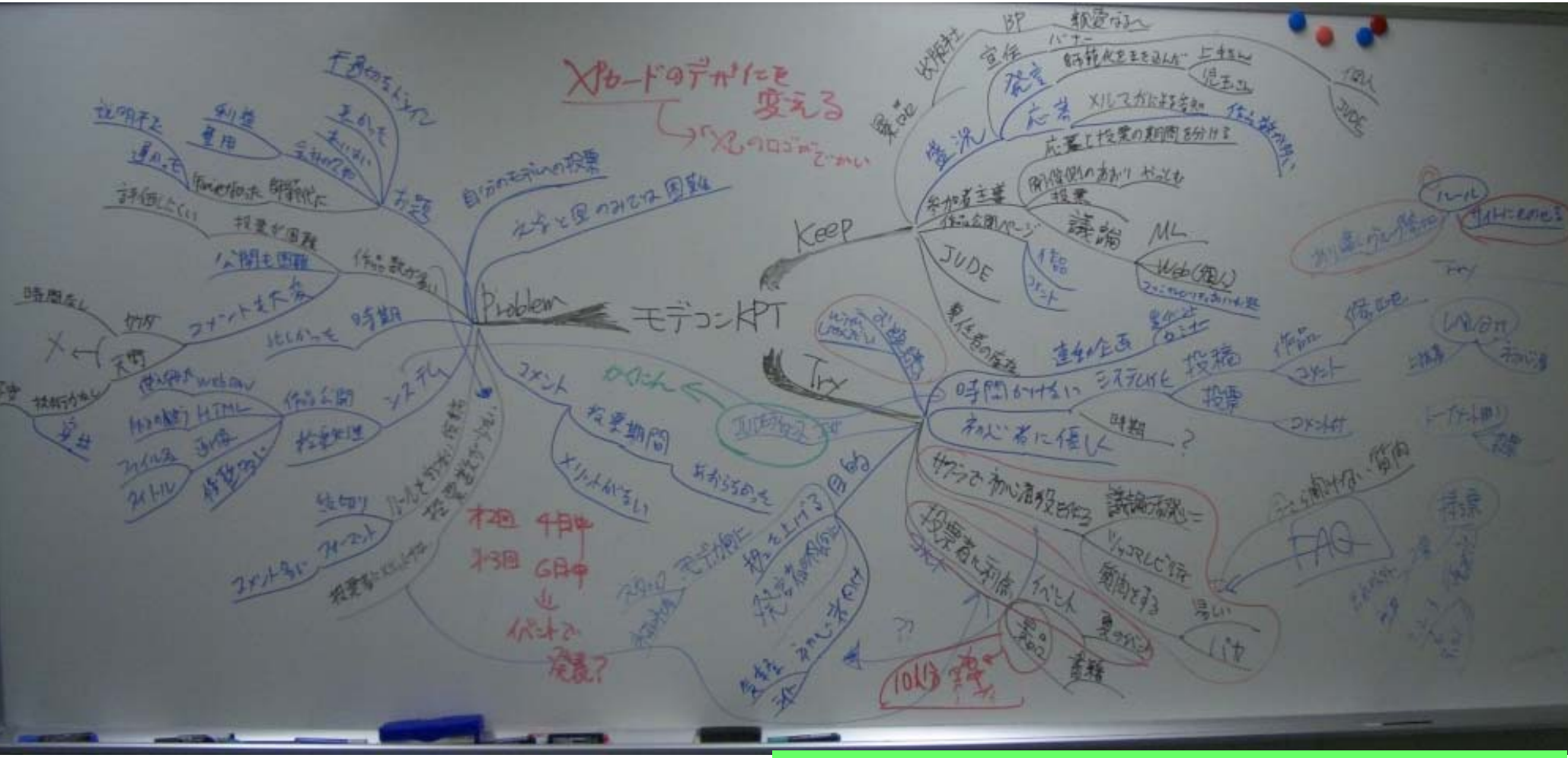
マインドマップの例（協力：**Kent Beck**、作：**懸田剛**)

POINT 頭の中は、「マインドマップ」で見える化。

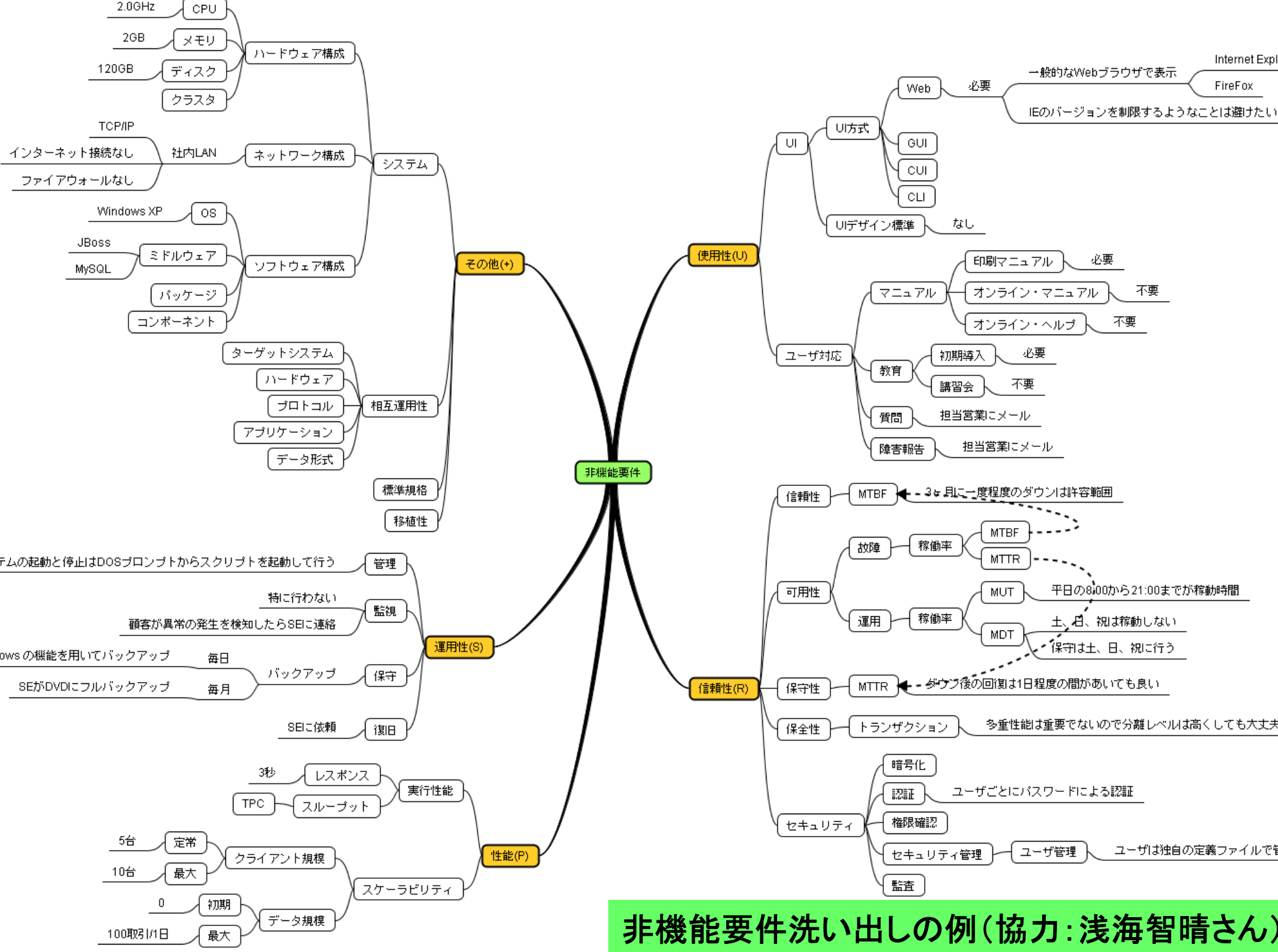
マインドマップの例: 作:水越明哉



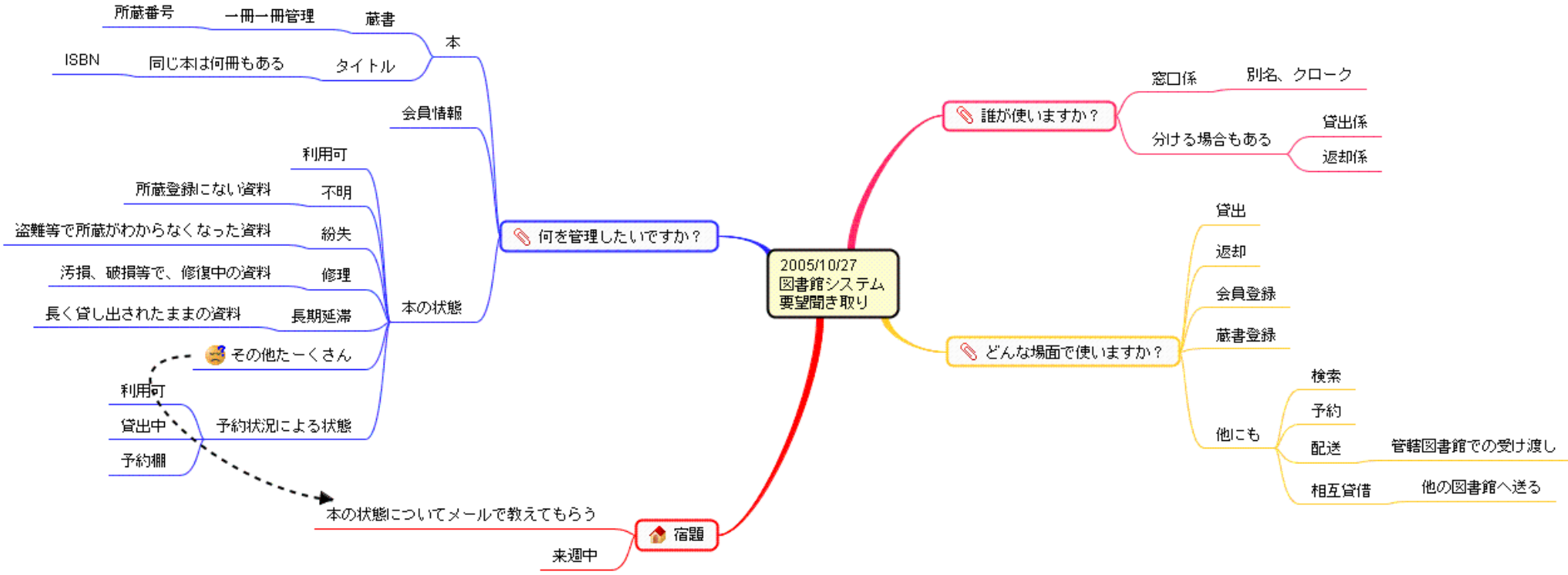
読書のまとめの例（常盤文克著『モノづくりのこころ』



ホワイトボード上のブレインストーミングの例



非機能要件洗い出しの例(協力:浅海智晴さん)

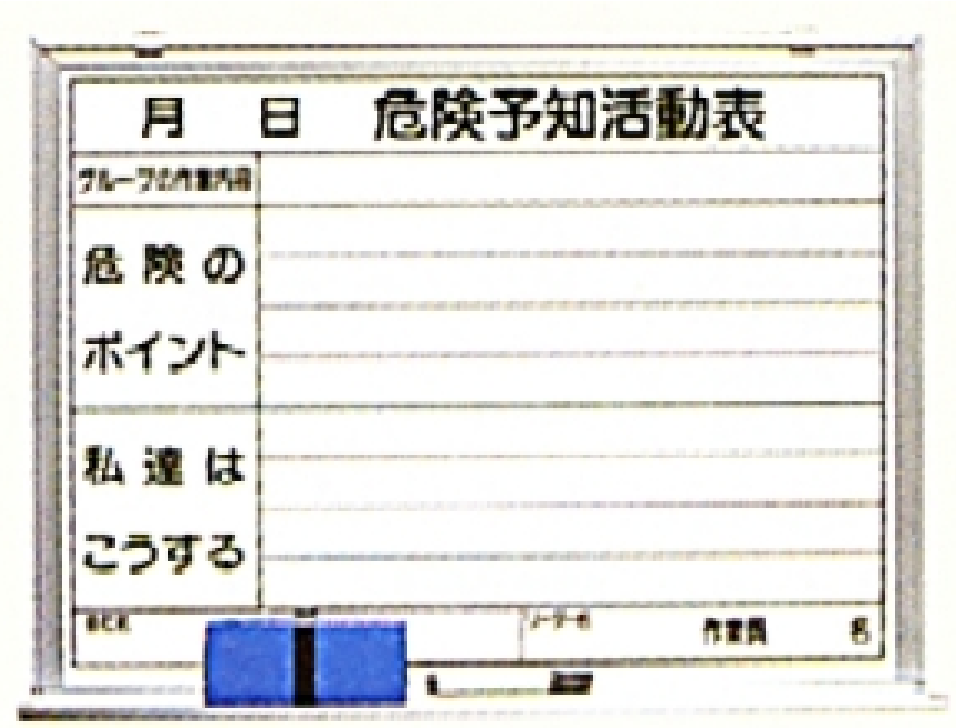


ユーザ要求聞き取りの例

新入社員教育

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。



危険予知版（土木現場で使われる）

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

SKMSの例

※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

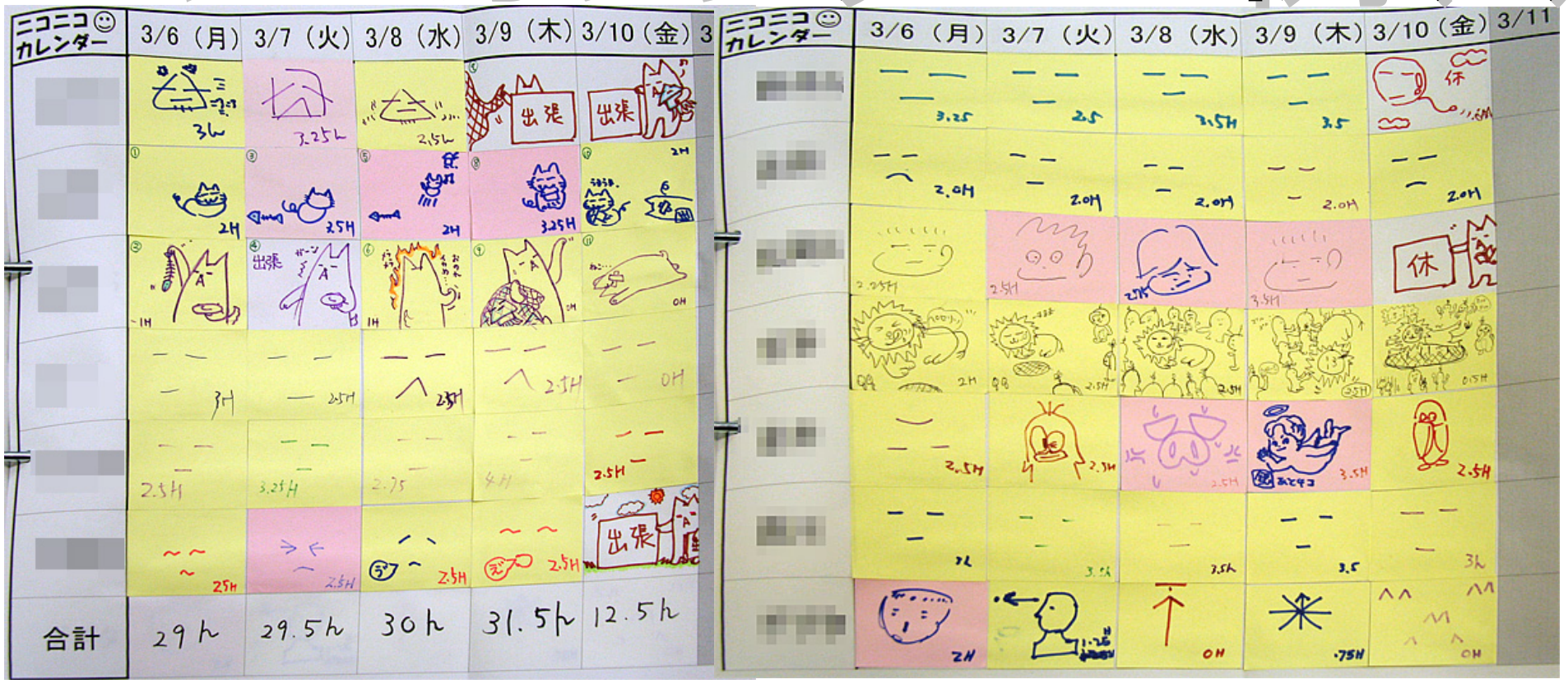
- チームのムードを見える化する。
- 帰宅時の気分を、
  - 気持ちよく仕事が終えられた
  - フツウ
  - ダメダメ
- チームが自発的にモチベーションマネジメント

※(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

にこにこカレンダーの例(1)



この掲示は、メンバが自発的に参加しないと成り立たないが、うまく機能することでメンバのモチベーションがアップする。この例では、カレンダー上で掛け合いをしたりして楽しんでやっている。

このカレンダーは保存されており、日次の作業(＝付箋の枚数や種類)とつきあわせることで、作業負荷とモチベーションの関係をつかむこともできる。

（協力：NECシステムテクノロジー）

にこにこカレンダーの例(2)

name	05/29	05/30	05/31	06/01	06/02	06/05	06/06	06/07	06/08	06/09	06/12	06/13	06/14	06/15	06/16	06/19	06/20	06/21	06/22	06/23	06/26	06/27	06/28	06/29	06/30	07/03	07/04	07/05	07/06	07/07
an_away_guy																														
GelMarkerer																														
gussan																														
Mr. Foie gras																														
Rice Field																														

- リリース日越えると、スマイルが増える。
- リリース中央から、リリースへ向けて、が難所。

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)
- 日々の作業は「タスクかんばん」で。
- 進捗は「バーンダウン」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。
- 「My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。
- 頭の中のアイディアを「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。
- チームムードを「ニコニコカレンダー」で。

POINT 見える化とリズムは、行動をうみだす第一歩。

プロジェクトファシリテーション

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。**

アジャイルプロセスとは

- 良いソフトウェアを手早く無駄なく作る手法の総称
  - 特定の開発手法ではない
- 以前は、ライトウェイト（軽量級）プロセスと呼ばれていた開発プロセス
  - Agile の日本語訳は

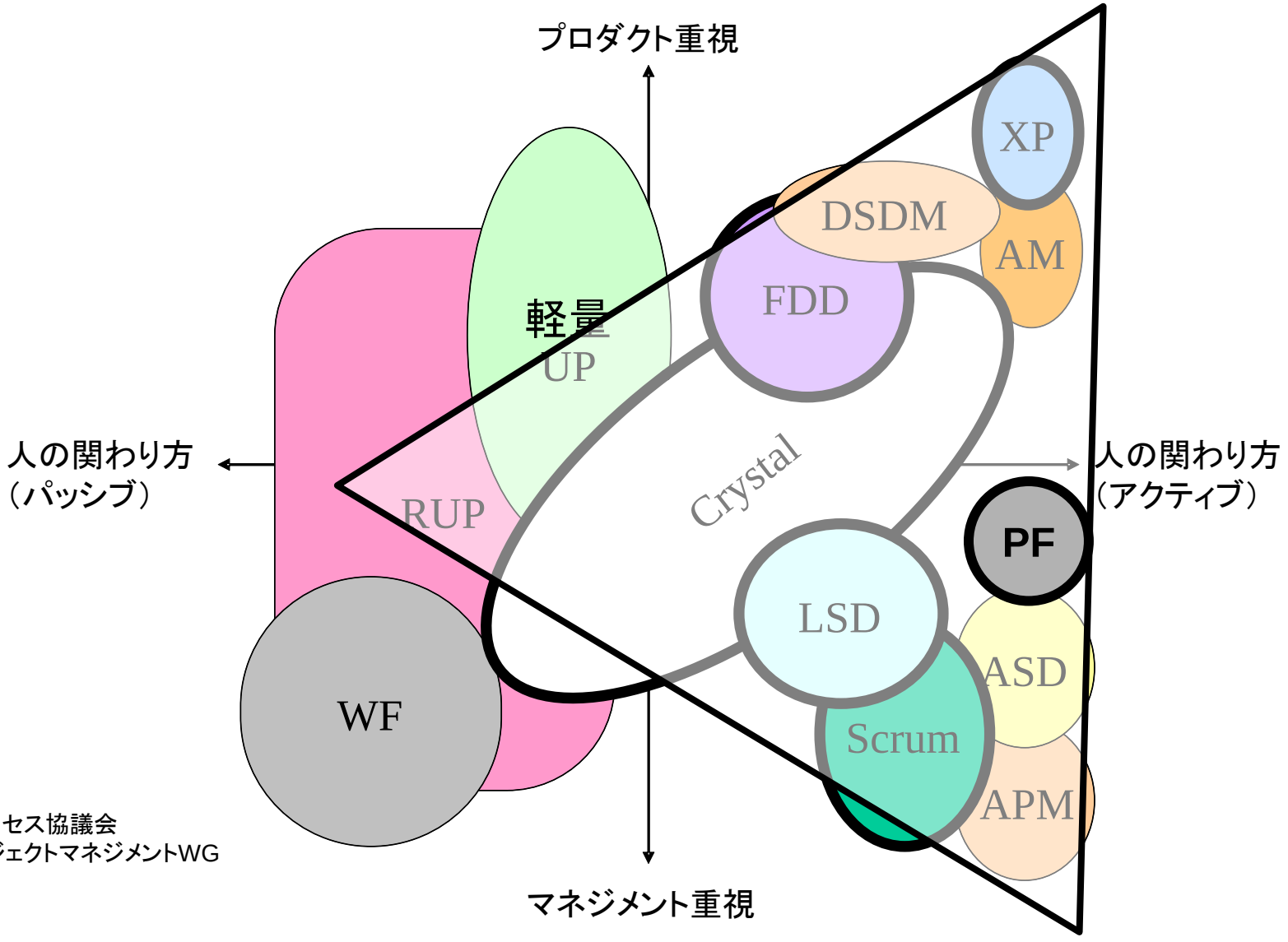
    敏捷な、いきいきした、頭のきれる、すばやい、機敏な、身のこなしの軽い、身の軽い、機動的な、鋭敏な、敏活な、活気のある、頭の回転の速い、しなやかな
- 繰り返し型であり、ウォーターフォールと対比させられることも多い。
- 「人」を中心においていることも特徴の1つ。

アジャイル関連書籍

- 『 XP(エクストリーム・プログラミング)』(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
  - 「コマンドーコントロール」⇒「リーダシップーコラボレーション」
  - 「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**

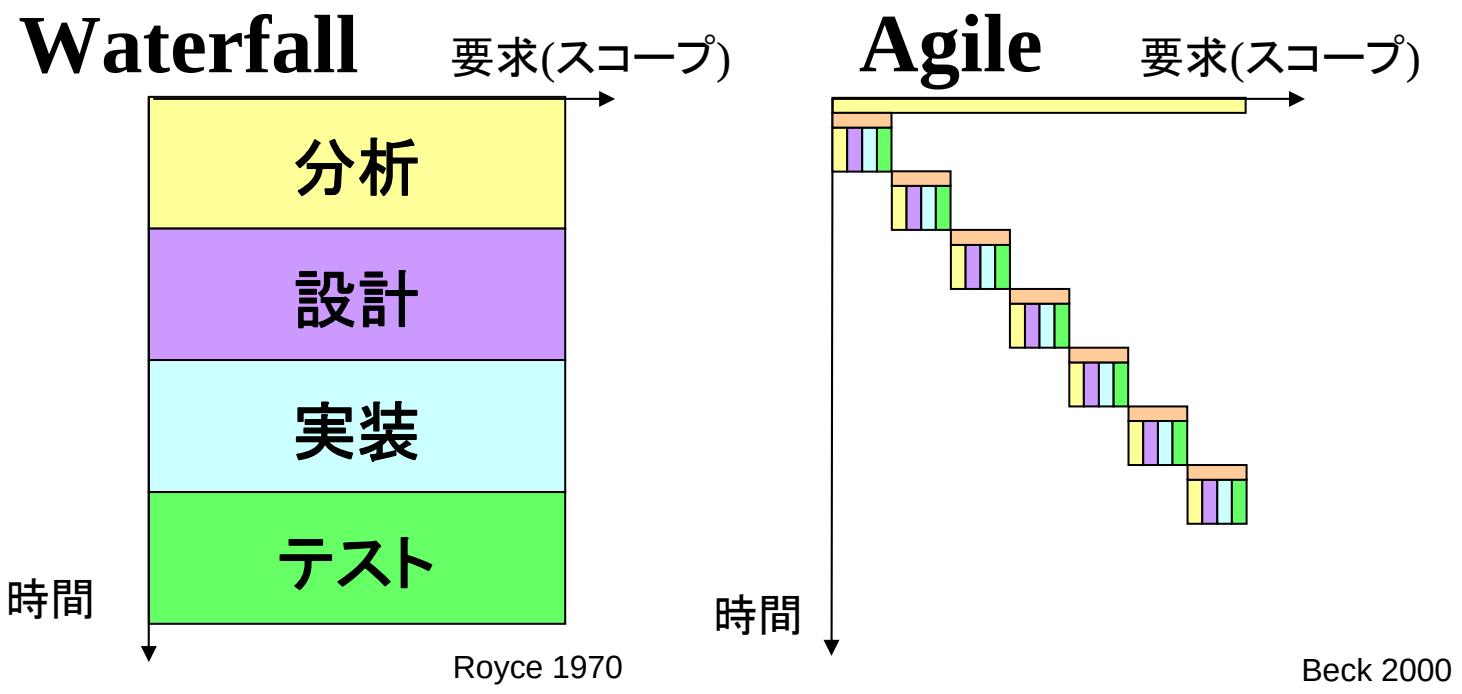
各プロセスの分類図



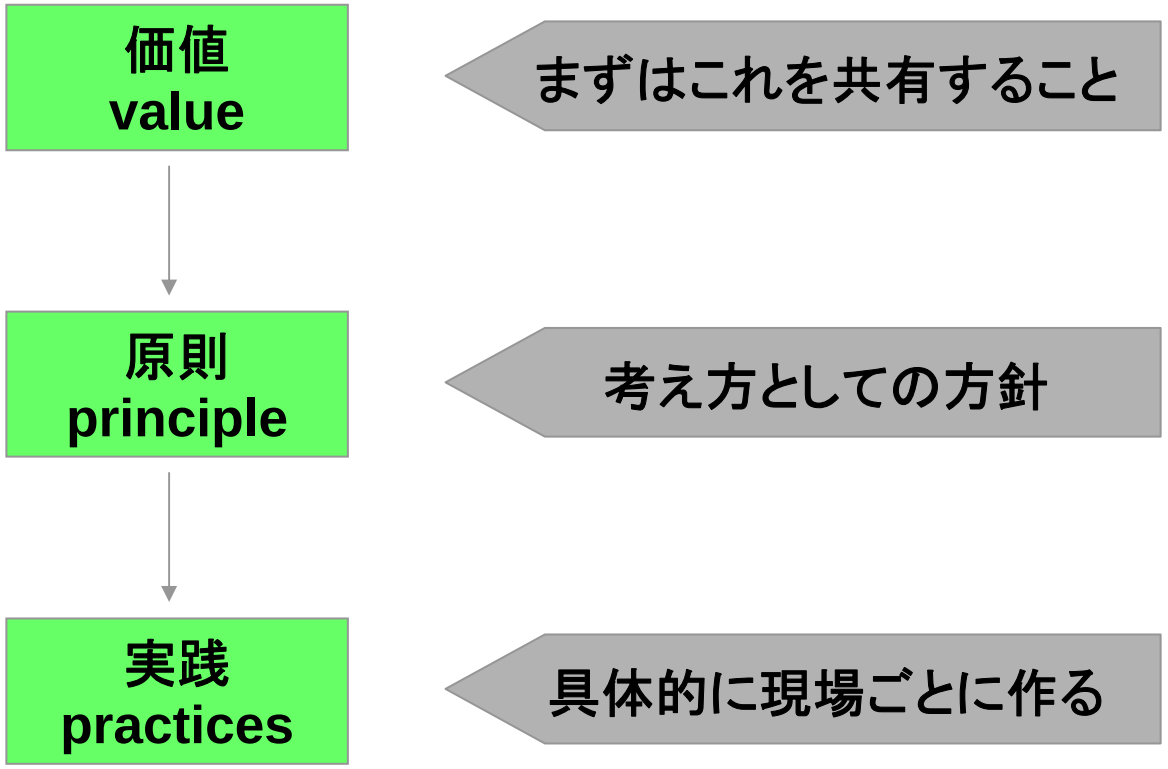
©アジャイルプロセス協議会
アジャイルプロジェクトマネジメントWG

プロセスとしてのAgile

- 短いサイクルで、分析、設計、実装、テストを並列に行う
- 進化型開発



アジャイルの価値、原則、実践



POINT 「価値」、「原則」、「実践」のレイヤがあり、それぞれの存在意義がある。

アジャイルの価値

私たちは，

プロセスとツールよりも ..... 個人と対話に．
包括的なドキュメントよりも ..... 動くソフトウェアに．
契約交渉よりも ..... 顧客との協調に．
計画に沿うことよりも ..... 変化に対応することに．

価値をおく．

出展：アジャイル宣言(agilemanifesto.org)

POINT 価値を共有することからはじめる

アジャイルの原則

- **顧客価値の優先**
  価値のあるソフトウェアをできるだけ早い段階から継続的に納品することによって顧客満足度を最優先します。
- **変化に対応**
  要件の変更はたとえ開発の後期であっても受け入れます。変化を味方につけることによってお客様の競争力を引き上げます。
- **短期のリリース**
  動くソフトウェアを2~3週間から2~3ヶ月というできるだけ短い時間間隔でリリースします。
- **全員同席**
  ビジネスをする人と開発者はプロジェクトを通して日々一緒に働かなければなりません。
- **モチベーションと信頼**
  意欲に満ちた人々を集めてプロジェクトを構成します。環境と支援を与え仕事が無事終わるまで彼らを信頼してください。
- **会話**
  情報を伝えるもっとも効率的で効果的な方法はフェイス・トゥ・フェイスで話をすることです。
- **動くソフトウェア**
  動いているソフトウェアこそが進捗の最も重要な尺度です。
- **持続可能なペース**
  アジャイル・プロセスは持続可能な開発を促進します。一定のペースで永続的に保守できるようにしなければなりません。
- **技術**
  卓越した技術と優れた設計に対する不断の注意こそが機敏さを高めます。
- **シンプル**
  シンプルさ – ムダなく作れる量を最大限にすること – が本質です。
- **自己組織的チーム**
  最良のアーキテクチャ、要件、設計は自己組織的なチームから生み出されます。
- **ふりかえりと改善**
  チームがもっと効率を高めることができるかを定期的に振り返り、それに基づいて自分たちのやり方を最適に調整します。

POINT 「原則」は、「価値」と「実践」を繋ぐ。

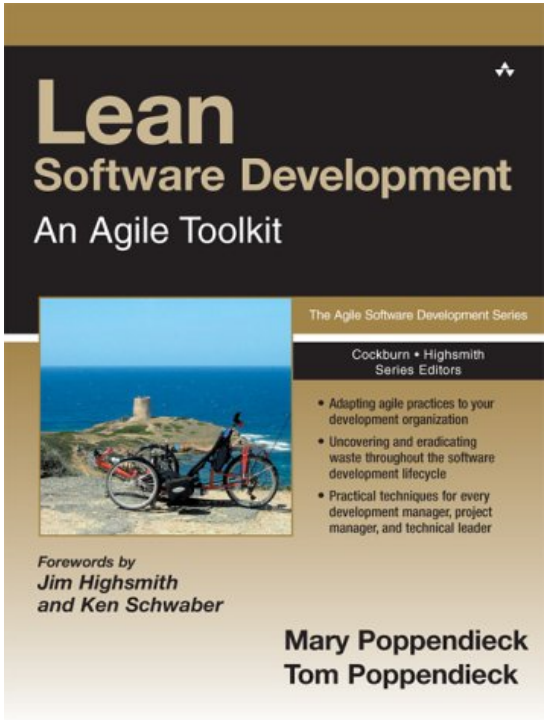
アジャイルの実践（例　XP）

- **計画ゲーム**
  - ビジネス優先度と技術的見積により次回リリースの範囲を早急に決める。現実が計画と変わったら、計画を更新する。
- **小規模リリース**
  - シンプルなシステムを早急に生産に投入する、それから新バージョンを非常に短いサイクルでリリースしていく。
- **メタファー**
  - どの様に全体のシステムが機能するかを示すシンプルな メタファーを共有することで全ての開発を導く（ガイドする）。
- **シンプルデザイン**
  - いつでもシステムは出来る限りシンプルに設計されるべきだ。余分な複雑さは見つけ次第取り除かれる。
- **テスティング**
  - プログラマは継続的にユニットテストを書く。顧客は、機能の開発が終わったことを示す受け入れテストを書く。
- **リファクタリング**
  - ２重コードを取り去り、単純化し、柔軟性を加えるために、プログラマは、システムの動作を換えることなくシステムを再構成する。
- **ペアプログラミング**
  - 全てのコードは２人のプログラマにより一台のマシンで書かれる。
- **共同所有権**
  - 誰でも、どのコードでも、どこででも、いつでも、プログラマはコードを修正できる。
- **継続的インテグレーション**
  - システムを一日に何回もインテグレードしビルドし、テストを 100% パスさせる.
- **週40時間**
  - 週４０時間以上仕事をしてはいけないのがルール。一日８時間を燃焼する。
- **オンサイト顧客**
  - 現実のユーザをチームに加えて、フルタイムで質問に答えられるようにする。
- **コーディング標準**
  - プログラマは、コーディング標準に従って全てのコードを書く

POINT　**「実践」は、やるかやらないか、わかるくらいに具体的。しかし状況で変化する。**

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、部分最適が起こってしまう。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

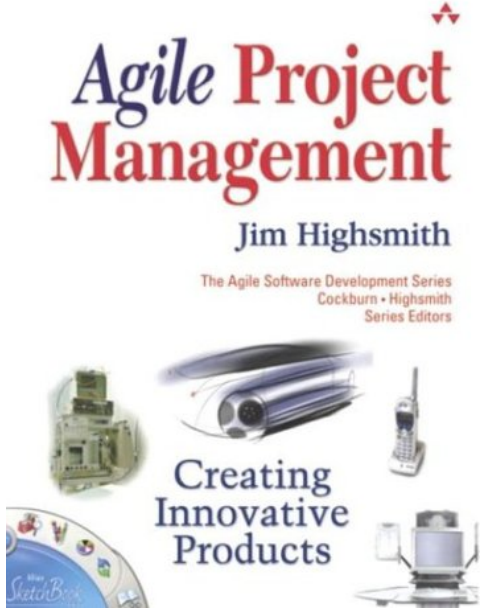
- トヨタ生産方式の「7つのムダ」とソフトウェア開発をマッピング

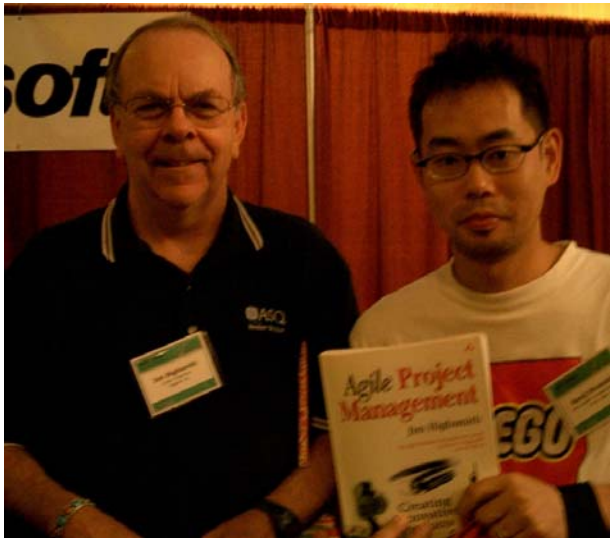
生産工程の7つのムダ	ソフトウェア開発の7つのムダ
在庫のムダ	未完成の作業のムダ
加工そのもののムダ	余分なプロセスのムダ
作りすぎのムダ	余分な機能のムダ
運搬のムダ	タスク切り替えのムダ
手待ちのムダ	待ちのムダ
動作のムダ	移動のムダ
不良を作るムダ	欠陥（バグ）を作るムダ

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。

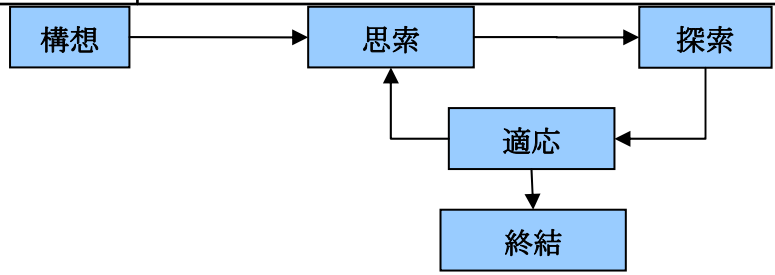
アジャイルプロジェクトマネジメント

- 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 アジャイル開発の集大成がつまっています。

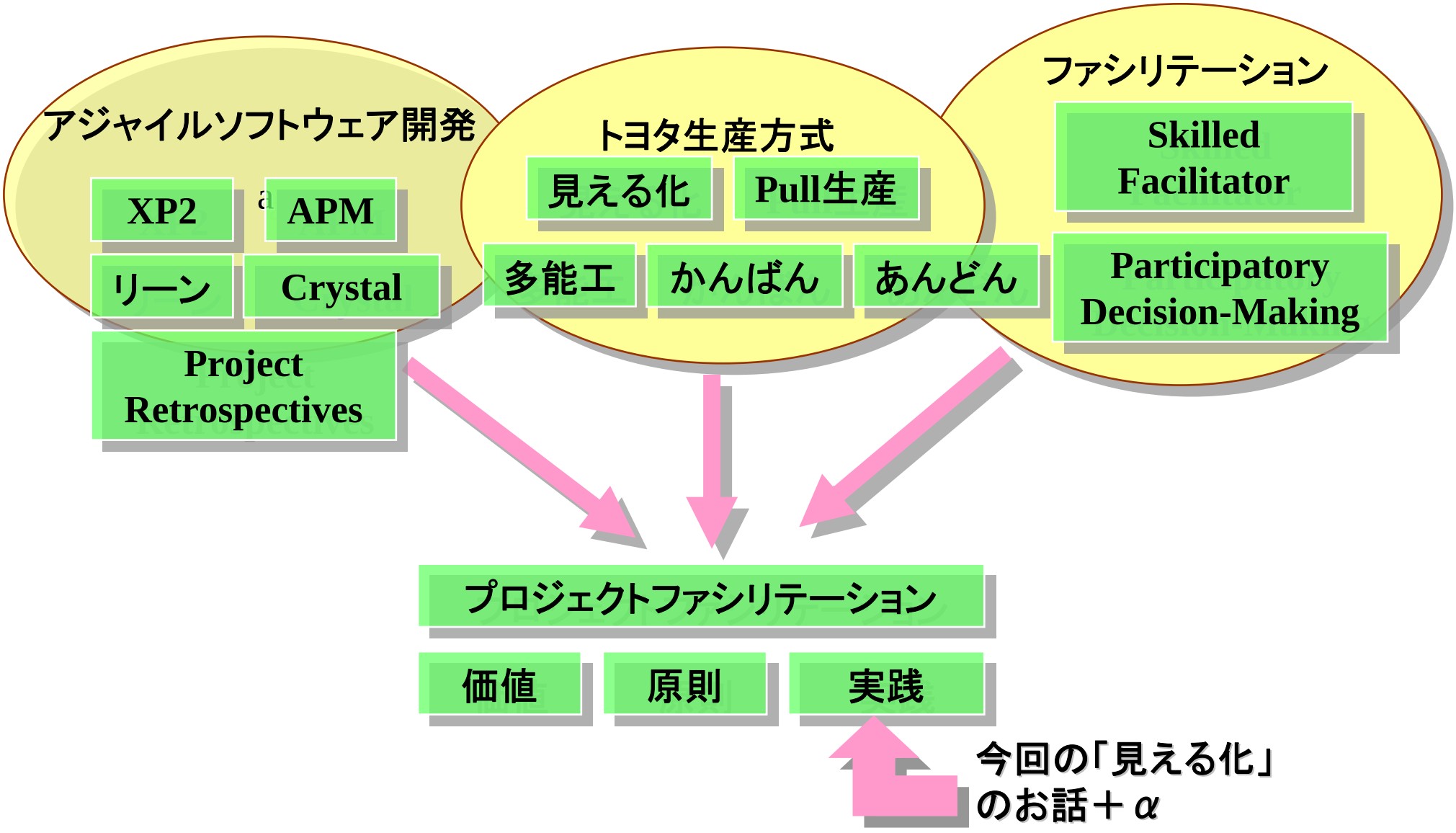
価値	変化への対応、動作する製品、顧客との協調、個人と対話		
原則	製品の提供	反復型で機能ベースの提供、顧客価値の提供、技術的優位性の重視	
	リーダシップ－コラボレーション	適応型チームの構築、探索の奨励、シンプルにする	
プロセスフレームワーク			
プラクティス	構想フェーズ	ビジョン	製品ビジョンボックスとエレベーターテスト
		スコープ	プロジェクトデータシート
		コミュニティ	適任者の確保
			関係者の特定
			顧客チーム・開発チーム間のインタフェース
		アプローチ	プロセスやプラクティスのテーラリング
	思索フェーズ	FBS(機能ブレークダウン構造)	製品の機能リスト
			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

トヨタ生産方式とアジャイルの共通点

- 品質を作りこむ
  - 100つくって検査で90はねてもダメ
  - 一人ひとりが工程で品質を作りこむ
  - テスト駆動開発、テストで進捗をはかる
- 自働化
  - 現場の一人ひとりにラインを止める権限（あんどん）
- より顧客に近い側から生産を制御（pushでなくpull）
  - 後工程引き取りによる「停滞のムダ」の排除
  - 今不要ないものは作らない(YAGNI)，ビジネス主導のプライオリティ付け（要求の一個流し）
- 誰にでも見える進捗のアナログな管理
  - 「生産管理版，カンバン，あんどん」と「付箋、カード、Myボード」
- 一人ひとりに工夫の権利
  - 多工程持ち，多能工
  - 出来る限り役割分担しない。トラックナンバーを上げる。ペアで作業をする。

参考：
「トヨタ生産方式」
「現場のムダどり事典」

PFのなりたち



POINT PF=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。

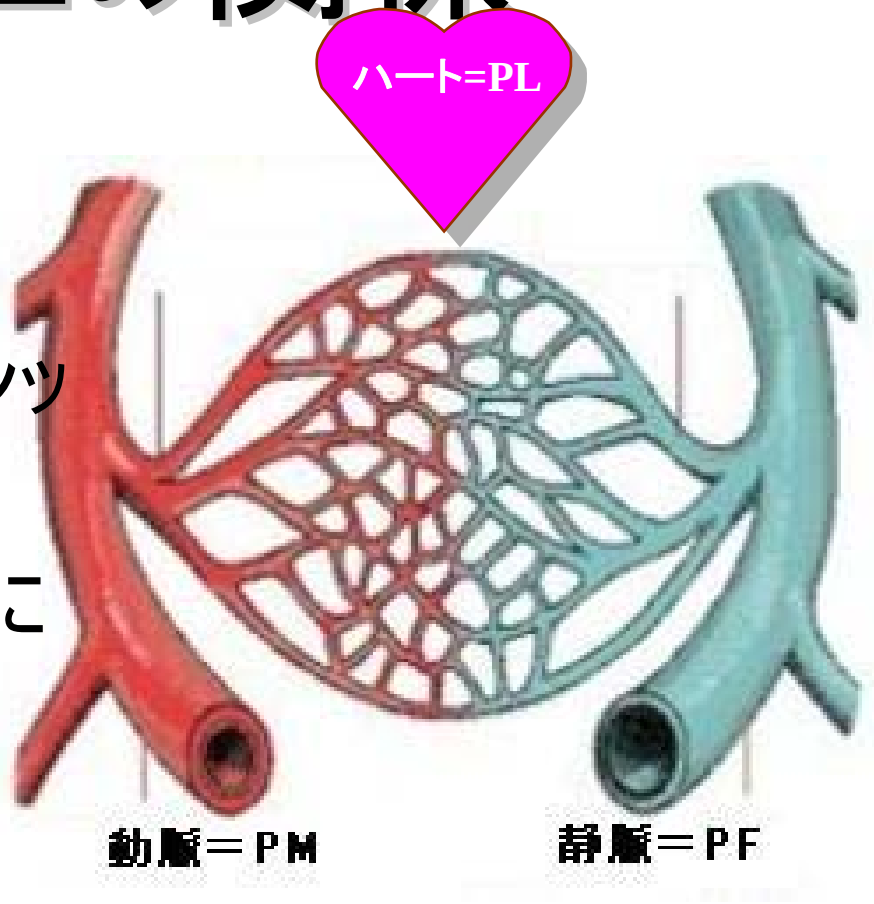
目的:なぜPFが重要か

- 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と、エンジニアのやりがいの両立を目指します。**

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 プロジェクトマネジャ
  - PL プロジェクトリーダ
  - 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.**

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すか？*
  - 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5つの原則

- 見える化(Management by Sight)
  - 目に見えるようにして、行動につなげる。
- リズム(Rhythm)
  - 人間活動として定期的なリズムを設計する。
- 名前づけ(Name and Conquer)
  - 気づいた概念に名前をつけておく。
- 問題 vs. 私たち (Problem vs. us)
  - 「問題」と「人間」を分離する。
- カイゼン(Kaizen)
  - 継続的に、今の自分たちできる、小さいことから。

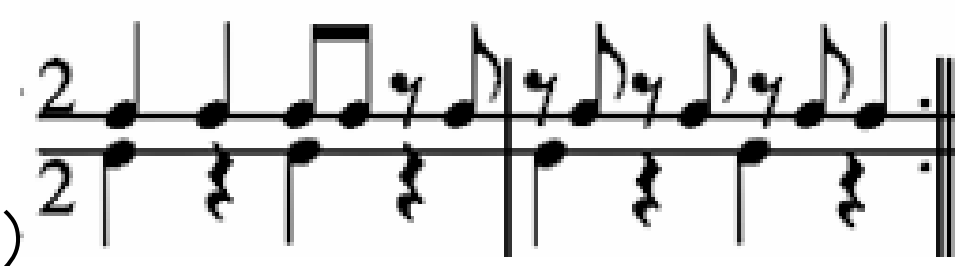
見える化

- 「最新の正の情報」が、「一箇所に」、「大きく」書かれていて、それを、「両チームのメンバー」、「審判」、「観客」が見ている。「次の行動」を誘発する。

POINT 全ステークホルダーが、行動を起こせるような、確かな、分かりやすい情報源

リズム

- リズムを「デザイン」する
  - 四半期、月、週、日
  - 会議や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

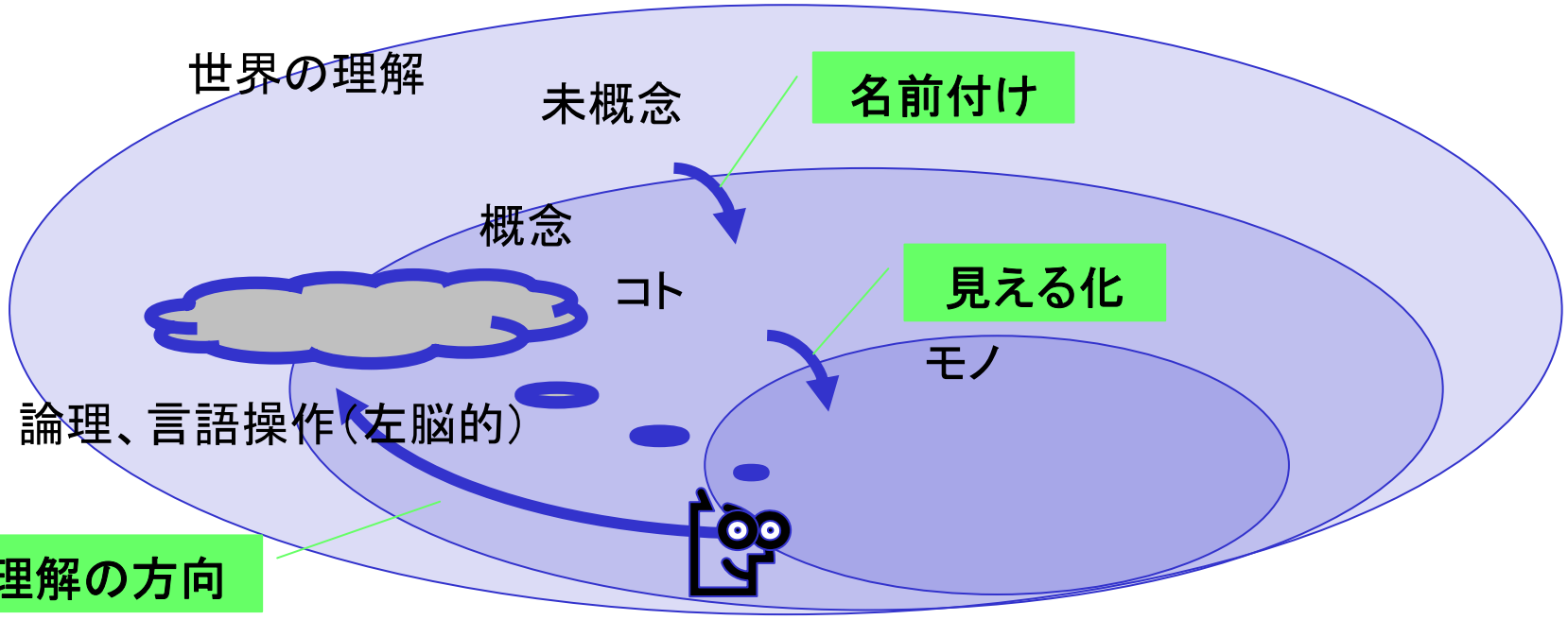
- 「気づき」をキャッチ
- ナレッジを，
  - 定着
  - 他のチームに伝播
- 例：
  - 「今日のお仕事」（by 坂田さん）
  - 「ぬかどこ」(by 倉貫さん)
  - 「にこにこカレンダー」

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！

人間の理解と、名前、概念、モノ・コト、見える化、

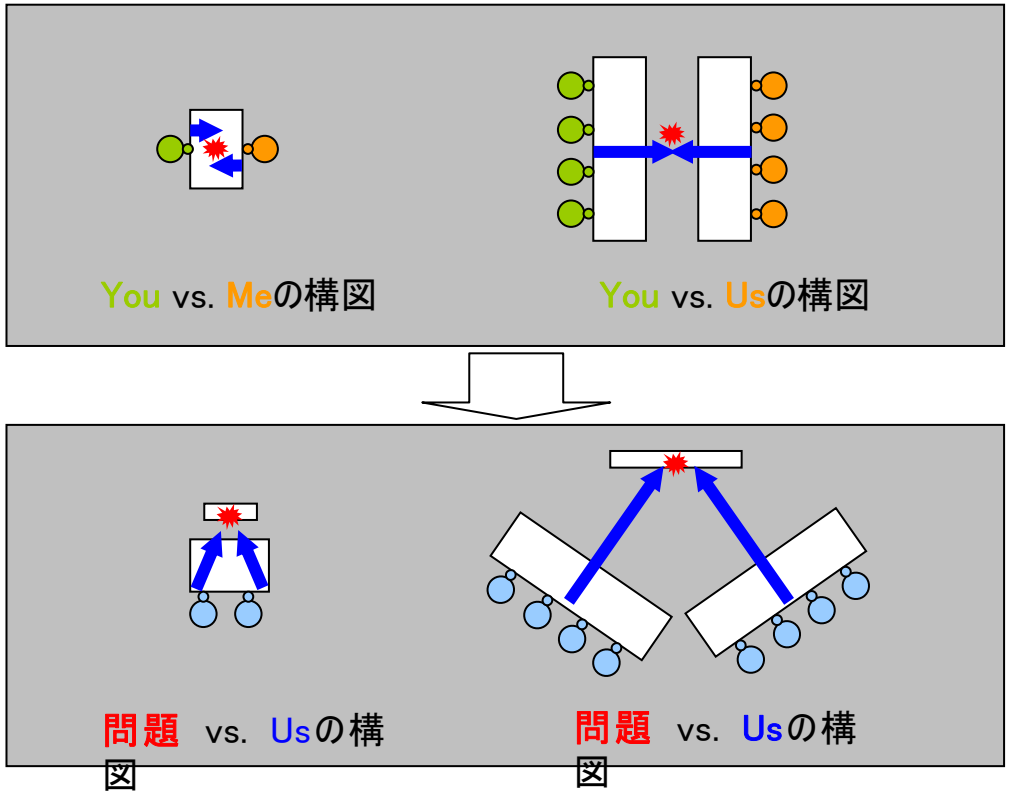
- **概念**＝名前のついたもの。私たちが頭の中で、あるいは、他者とのコミュニケーションに乗せてハンドリングできる単位。
- **モノ**＝概念のうち、目に見えるもの。物理法則に従うもの。



- **名前づけ**＝人間は、名前をつけることによってその対象物をはじめてはっきり認識できる。未概念の名称による概念化。輪郭の切り取り。言語操作対象化。
- **見える化**＝見えないものを見えるようにする（モノ化する）ことによって、よりはっきりと実体の存在感を得ることができる。

問題対私たち (Problem vs. Us)

- ともすると，議論は
  You vs. Me
  You vs. Us になりがち.
- 「問題」と「人」とを分離
- 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.

カイゼン

- 大きな改革でははじまらない
- 小さなカイゼンから
- 今の自分たちでできること
- 来週からできること
- よくなっていくことを体感しよう
- ふりかえり、が強力な武器
- For the better tomorrow
- 明日はきっと今日よりも、いい日に決まっている

POINT 継続的に、今の自分たちでできることからカイゼンを。うまくいったら自分たちをホメよう。

PFの実践

全体構成

色つきUML

- 見える化
- リズム
- 名前づけ
- 問題対私たち
- カイゼン

1週間

計画
半日

イテレーション開発
一日の繰り返し

ふりかえり
半日

要求

タスク

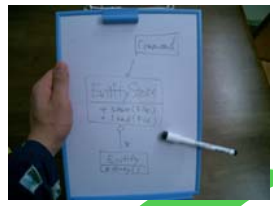
タスク

タスク

タスク

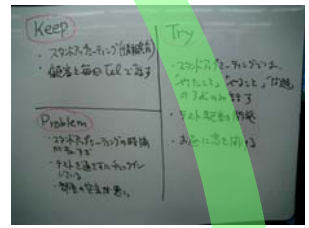
朝会、かんばん

バーンダウン

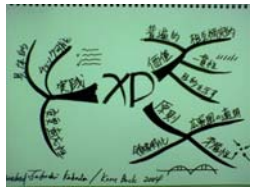


ペアボード

あんどん



ふりかえり



マインドマップ

にこにこカレンダー

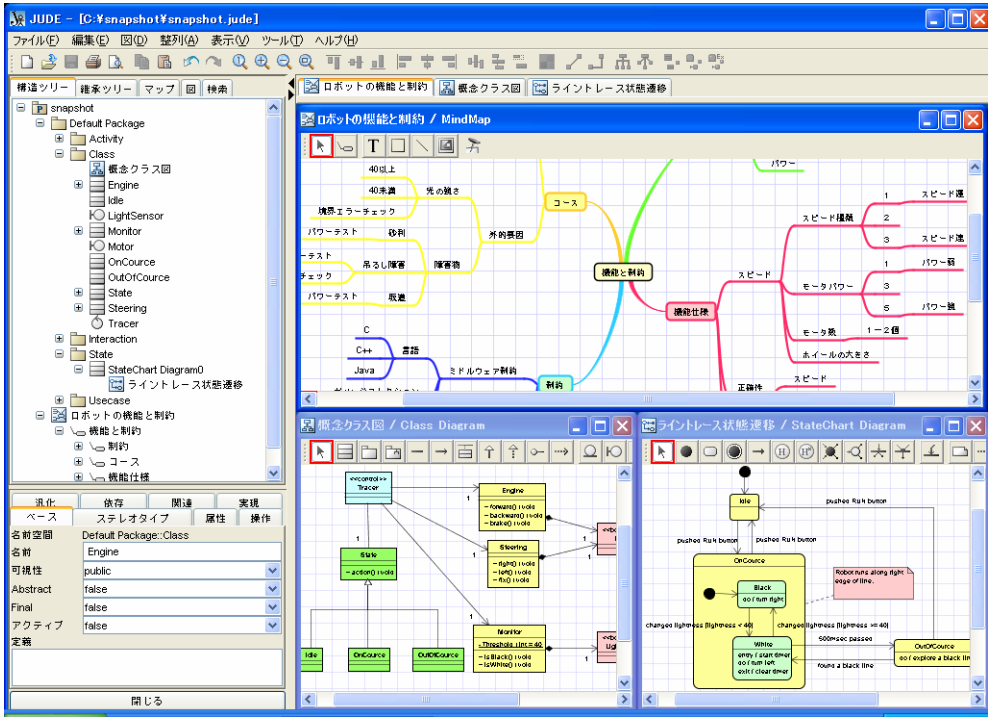
SKMS

JUDE開発チーム例

ツール/サービス

JUDE(ジュード)

- UMLとマインドマップの融合エディタ
- 無償版と有償版があります。
- 全世界に200,000人のユーザがいます。

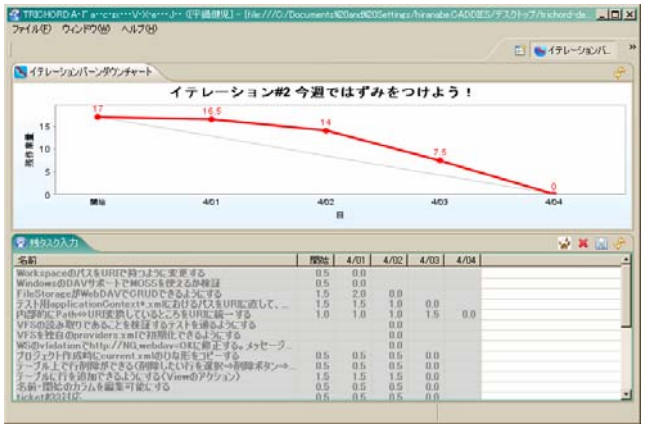


POINT ダウンロードは、http://jude.change-vision.com

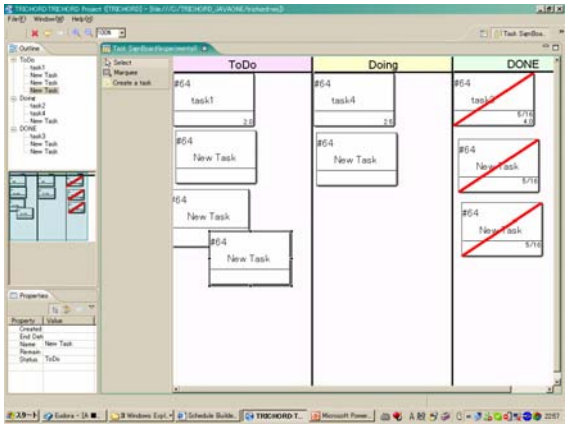
TRICHORD(トライコード)

- チームの情報共有板。
  管理者でなく、現場が使いたいから使う、情報発信ツール
- ニコニコカレンダー、バーンダウン、タスクかんばん、
  パーキングロット、カレンダー、、、などなど
- 製品版、無償版も公開中。

ニコニコカレンダー



バーンダウン

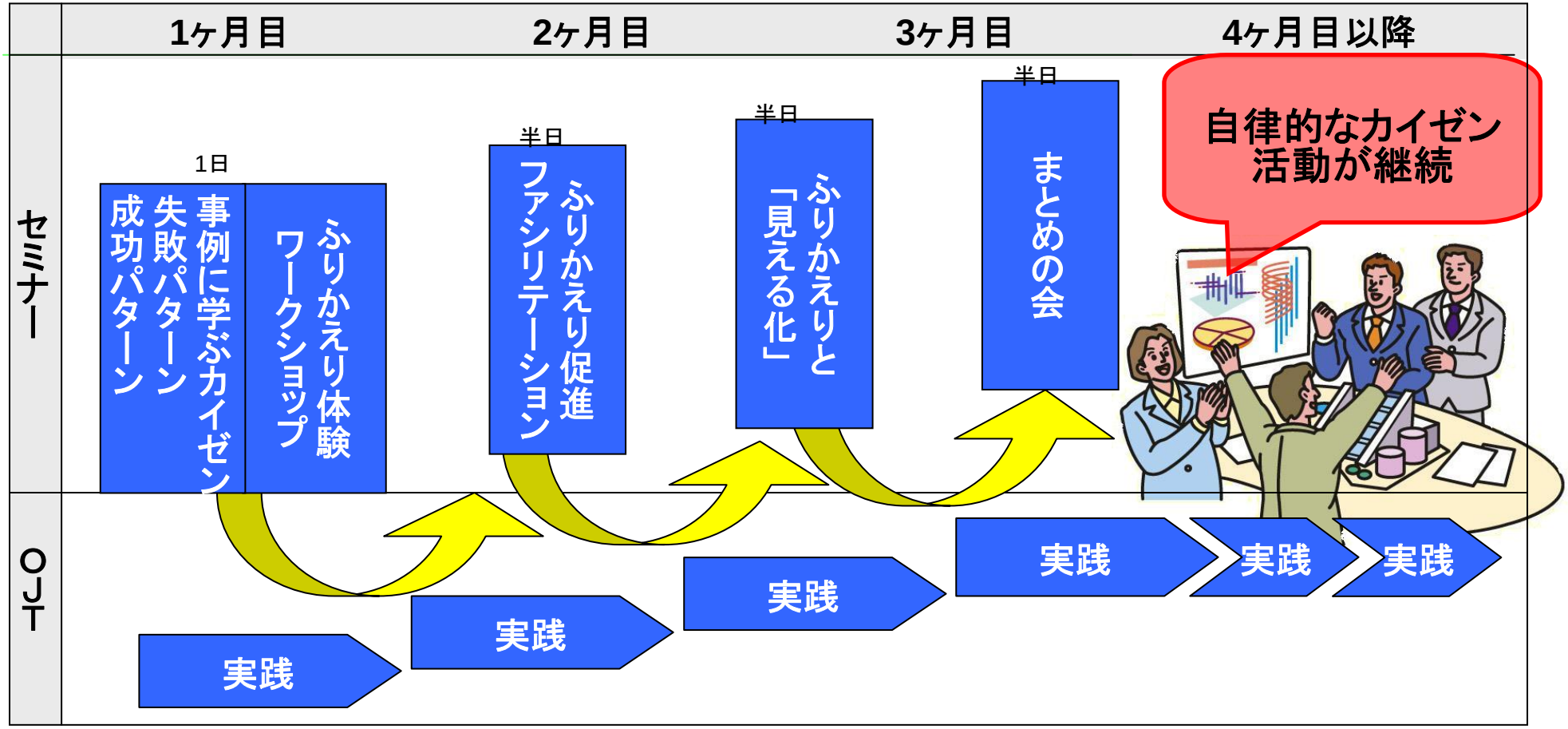


タスクかんばん

POINT ダウンロードは、http://trichord.change-vision.com

カイゼン入門セミナー

- 自律的なカイゼン活動を現場に定着させる
- セミナー(実習)とOJT(実践)のサイクルを通して体得できる



POINT 永和システムマネジメントが提供する教育サービスです。

最後に

PFの効果

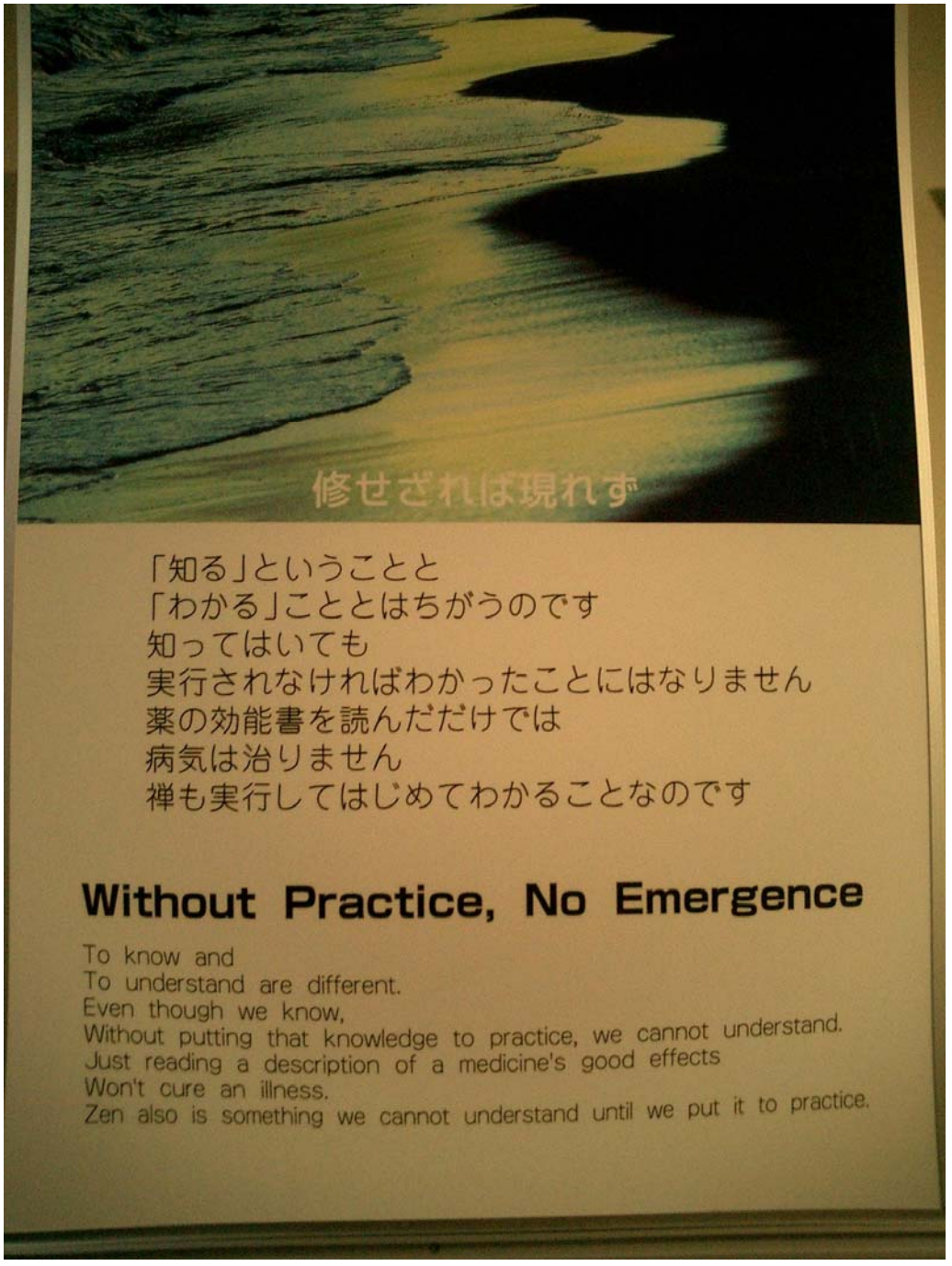
- 協調的なチームのムードを作り出す
- 笑顔の数
- 意外なリーダーの出現（人材の開発、発掘）
- 見える、マネジメント
- 早く分かるリスク（隠さない）
- 実感できる改善（くりかえし、ふりかえり）
- 自ら気づき、自ら行動することを、価値とする文化
- すぐ始められる！

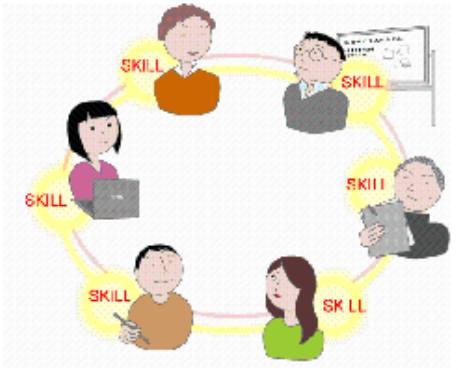
POINT **効果は、人材開発に関することが大きい。**

導入のポイント

- 簡単なものから、やってみる。
- そのままではうまく行かない。工夫して初めて定着する。
- 外から押し付けない。
  現場の若い人を必ず巻き込み、いっしょにやる。
- 現状を全否定しない。現状の良い点を生かし、課題を探そう。
- その過程で、ヒントとして使う。解決法を押し付けない。
- ポストイットや模造紙くらい、自分で買おう。（時間がもったいない）
- 人生の楽しみの一部、と考える。（Life Hacks）

POINT **現場を巻き込むプロセスが重要。**



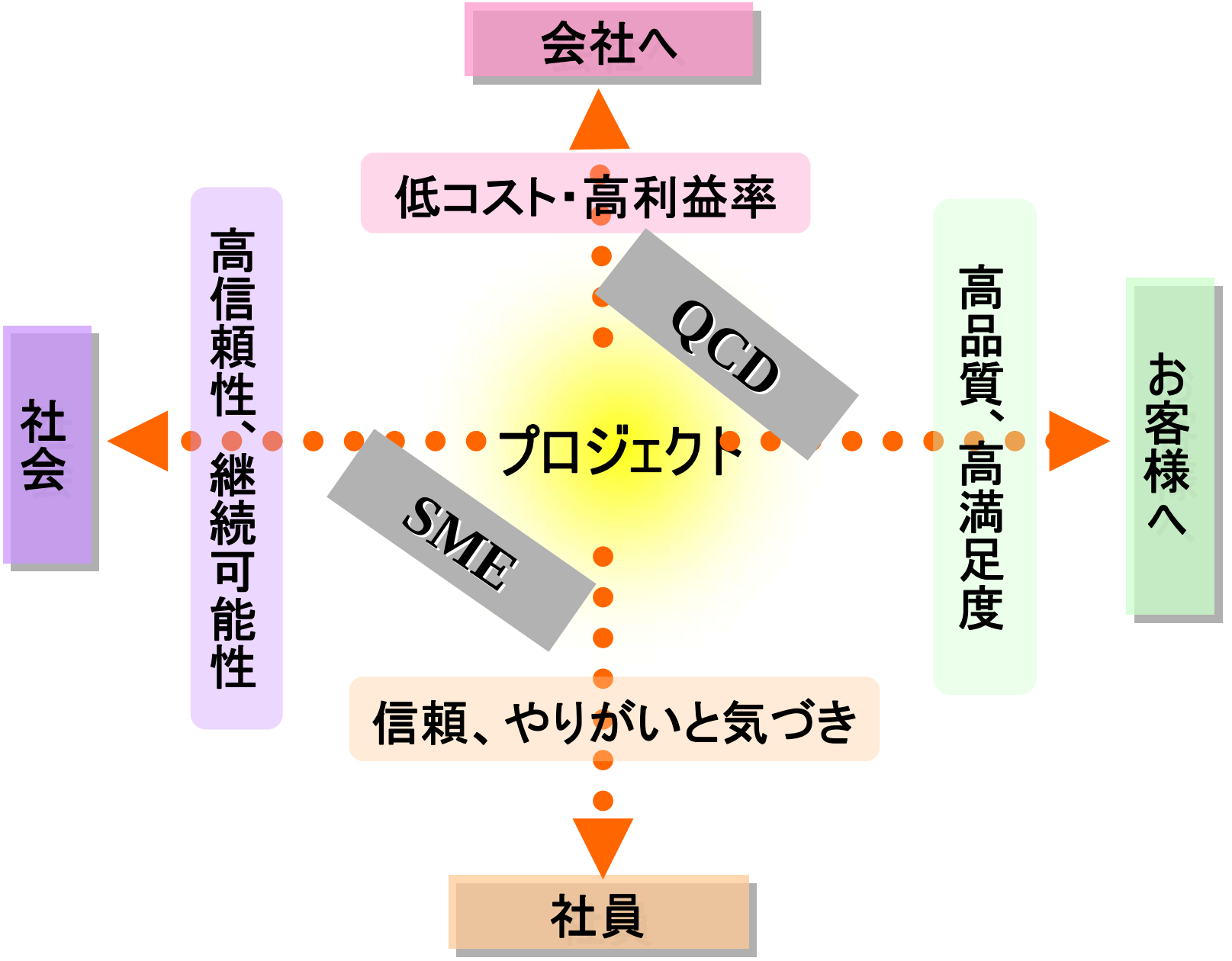


- プロジェクト ファシリテーション
  - 価値
    - コミュニケーション
    - 行動
    - 気づき
    - 信頼関係
    - 笑顔
  - 原則
    - 見える化
    - リズム
    - 名前付け
    - 問題 vs 私たちの構図
    - カイゼン
  - 実践
    - バーンダウン
    - かんばん
    - 朝会
    - あんどん
    - だるま
    - ペアボード
    - 色つきUML
    - ふりかえり
    - マインドマップ
    - KYミーティング
    - SKMS
    - にこにこカレンダー
  - PFとは
    - アジャイル開発
      - XP2
      - リーンソフトウェア開発
      - APM
      - Crystal Clear
      - 達人プログラマー
      - Project Retrospectives
    - トヨタ生産方式
    - ファシリテーション
  - 目的
    - プロジェクトの成功
    - QoELの向上

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
人々はロ々にこういう
「自分たちの力でやり遂げた」と。

老子

安定したプロジェクト、継続可能な価値創出のために



どうしたら変われるか？

- 与えられた仕事、から、気づく仕事へ。
- 自ら気づき、行動することを、価値とする文化を創ろう。
- 他人と過去は変えられない。
  自分が変わって未来をかえよう。
- ワークスタイルを変えよう、
  この業界を変えよう。

変化はあなたから始まる

- ***Social change starts with YOU.***

-- Kent Beck

ぼくは『周囲を変えたい』という状況になった場合、いつもこの言葉を忘れないように努力する--『自分が変えられるのは自分だけ』。でも、自分が変わるとすぐ、自分と周囲との関係が変化するんだ。そして、それで周囲の人が変化する、そして、自分の周りが変わり始める。大きな変化というのは、こういう風に始まるんじゃないかな、つまり、一人の人が世界と自分との関係を変えることを選択する。そして、その変化が広がっていく。

- ***You must be the change you want to see in the world.***

-- Mahatma Gandhi

参考文献・URL

- プロジェクトファシリテーション公式サイト
http://www.ObjectClub.jp/community/pf/
- PFP(プロジェクトファシリテーションプロジェクト)
http://projectfacilitationproject.go2.jp/
- 書籍:
  - 『プロジェクトを成功させる現場リーダーの「技術」』
  - 『アジャイルレトロスペクティブズ』
  - 『アジャイル・プロジェクト・マネジメント』
  - 『リーンソフトウェア開発』
  - 『ソフトウェア開発に役立つマインドマップ』
  - 『ポジティブ・チェンジ:主体性と組織力を高めるAI』
    (The Power of Appreciative Inquiry:
    A Practical Guide to Positive Change)

ご清聴ありがとうございました。