PF(Project Facilitation)
プロジェクト・ファシリテーション

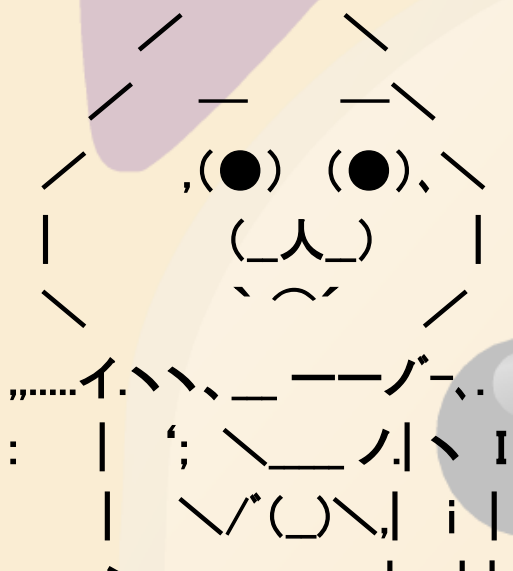
エンジニアを幸せにしようシリーズ（笑）

第１弾

きょうのおはなし

- 自己紹介
- PFとは
- 見える化とPF
- アジャイルとPF
- ツールと活用、効果

自己紹介



ゆーちです。

ハンドル名です。

本名は、内山康広といいます。

31歳（16進数）ですw

人生、波瀾万丈です（謎）

株式会社シーソフト代表取締役です。
現役のエンジニアです。プログラム書いてます。

メールソフト Becky! 用の BkReplyer ２をリリースしました。
バックアップ用 ProjecKit もよろしくw

PFとは

- プロジェクトファシリテーションとは

「ファシリテーション」って？

促進する、助ける、円滑にする

「日本ファシリテーション協会」https://www.faj.or.jp/

ファシリテーションの有効利用

会議の司会進行とか、アイスブレーキングとか、緊張を解く

- プロジェクトファシリテーション

造語：「プロジェクト」＋「ファシリテーション」

プロジェクトってキッツイよねぇ

- QCDSH

  品質＞コスト＞納期＞安全＞人間性

  デスマーチって人間性無視されてない！？
- 見える化

  ボトルネックの早期発見
- アジャイル開発

  XP（エクストリーム・プログラミング）

  リーンソフトウェア開発

ゆーちなりのPFの解釈

- 人間性＞品質＞コスト＞納期＞安全

働いている人の気持ちを、まず幸せにすることこそ重要。

見える化とPF

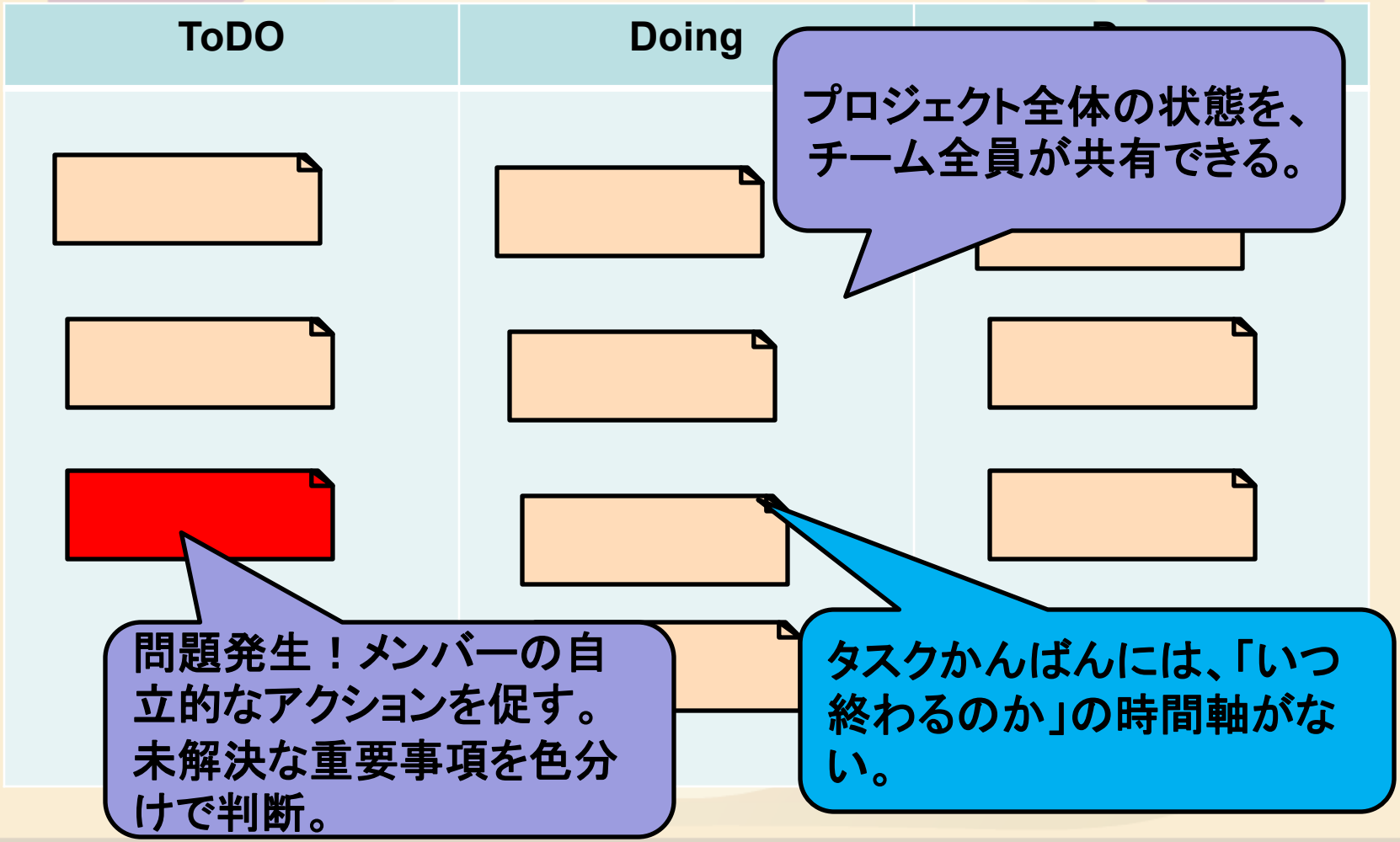
- 何のために「見える化」
- なぜ「見える化」
- どうやって「見える化」
- どこで「見える化」
- だれがやるの「見える化」

「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。

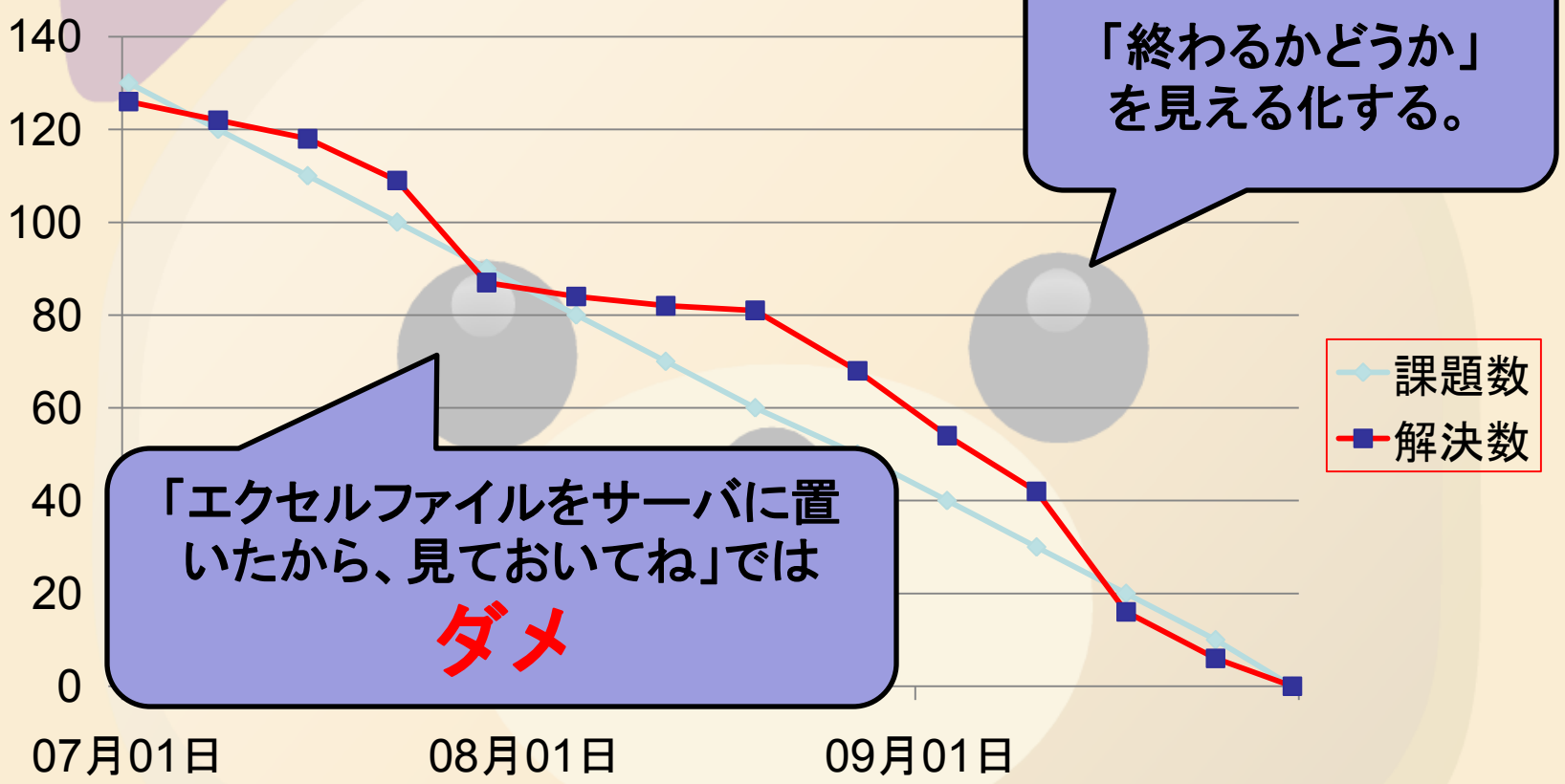
資料参照:
http://www.objectclub.jp/download/files/pf/ProjectFacilitation20071025.pdf

実践

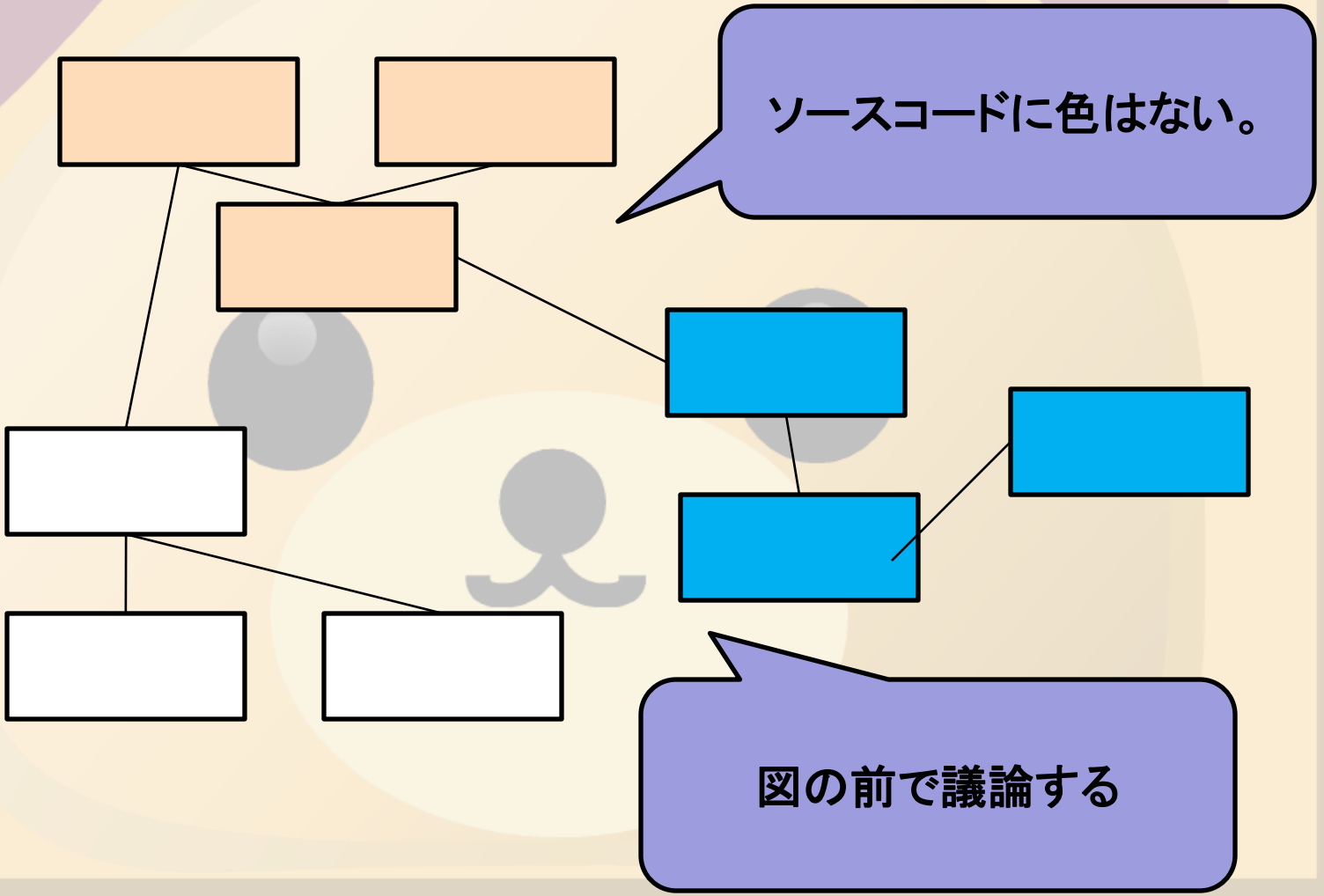
タスクかんばん


ToDO
Doing
プロジェクト全体の状態を、チーム全員が共有できる。
問題発生！メンバーの自立的なアクションを促す。未解決な重要事項を色分けで判断。
タスクかんばんには、「いつ終わるのか」の時間軸がない。

バーンダウンチャート

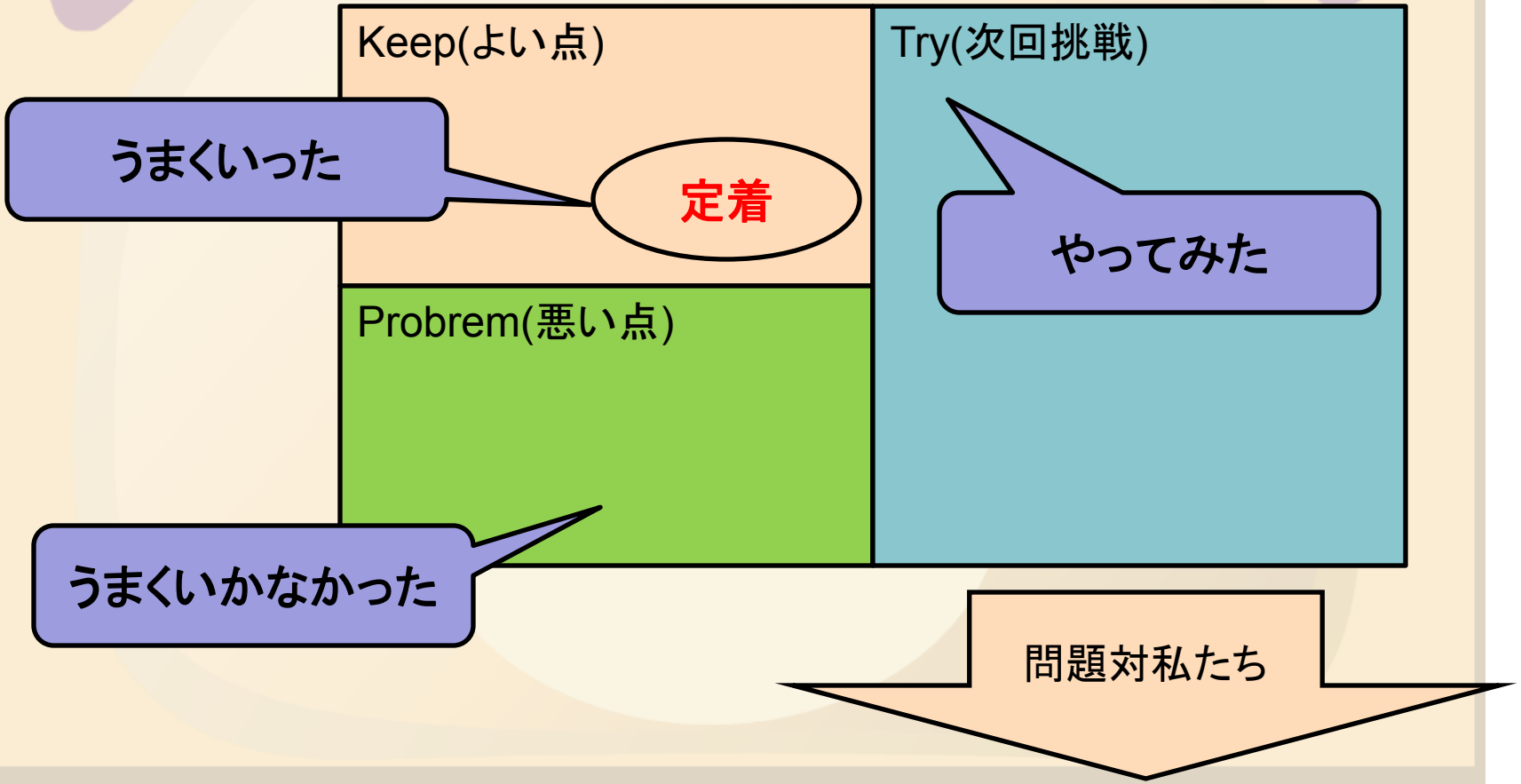

140
120
100
80
60
40
20
0
07月01日
08月01日
09月01日
課題数
解決数
「終わるかどうか」
を見える化する。
「エクセルファイルをサーバに置
いたから、見ておいてね」では
ダメ

色つきUML



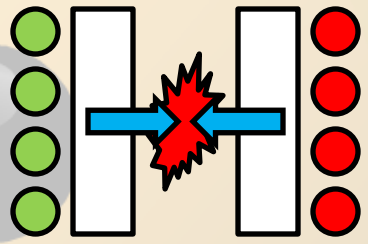
ふりかえり

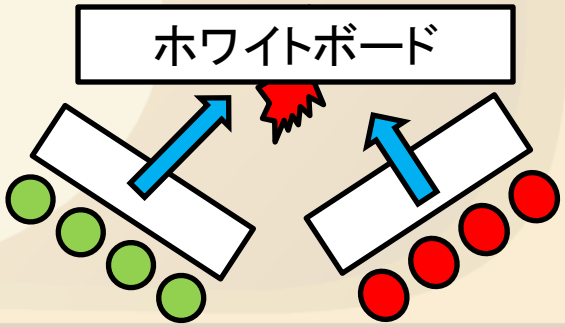
- KPT


Keep(よい点)
Try(次回挑戦)
うまくいった
定着
やってみた
Probrem(悪い点)
うまくいかなかった
問題対私たち

問題対私たち

- You vs. Me、You vs. Us になりがち。
- 問題と人を分離
- 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





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)

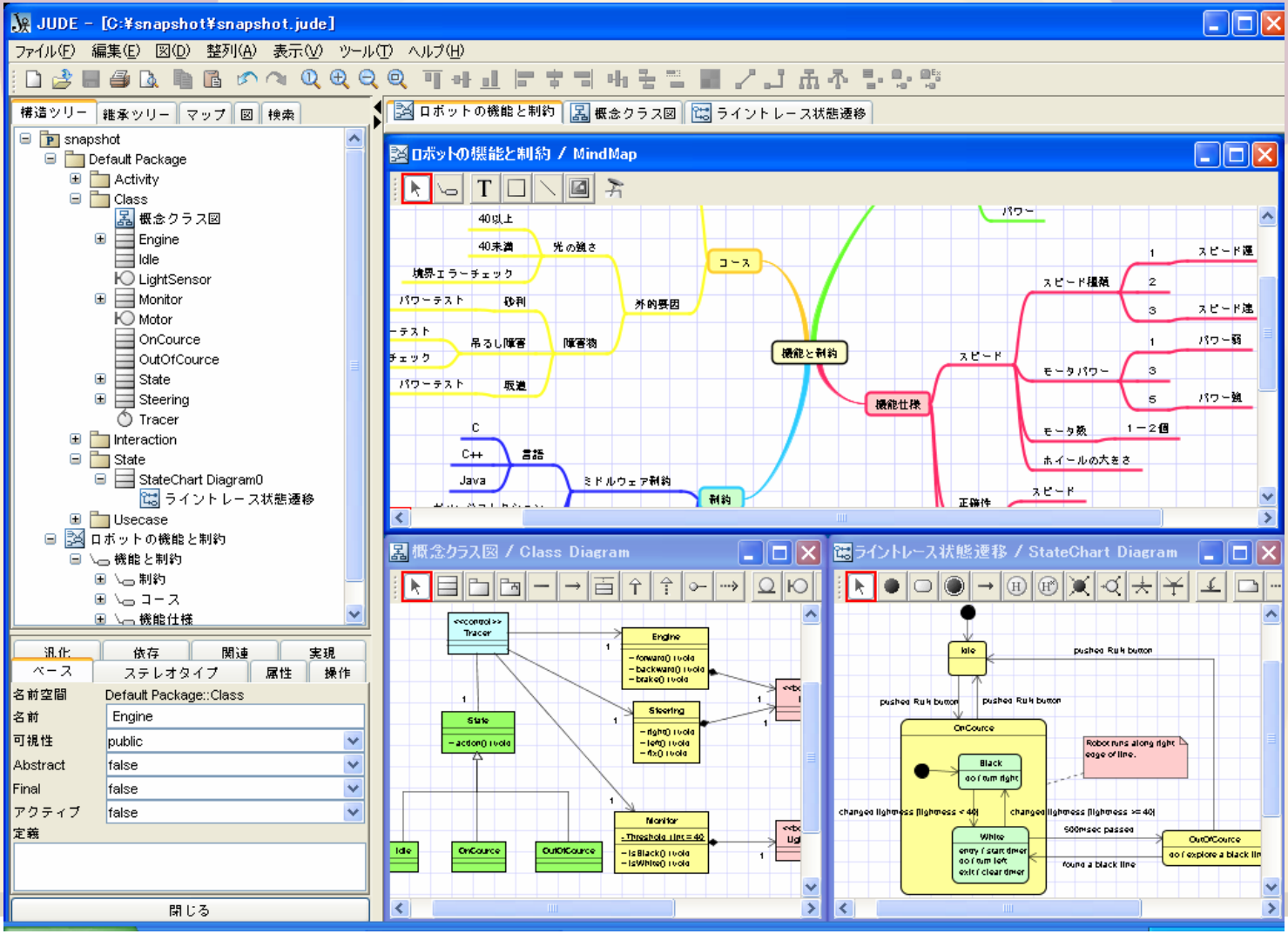
アジャイルの原則

- **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。

リーン思考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。

JUDE



わんくま同盟 福岡勉強会 #8

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

にこにこカレンダーシート

にこにこカレンダー
シールセット

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！

いますぐ始めてみよう

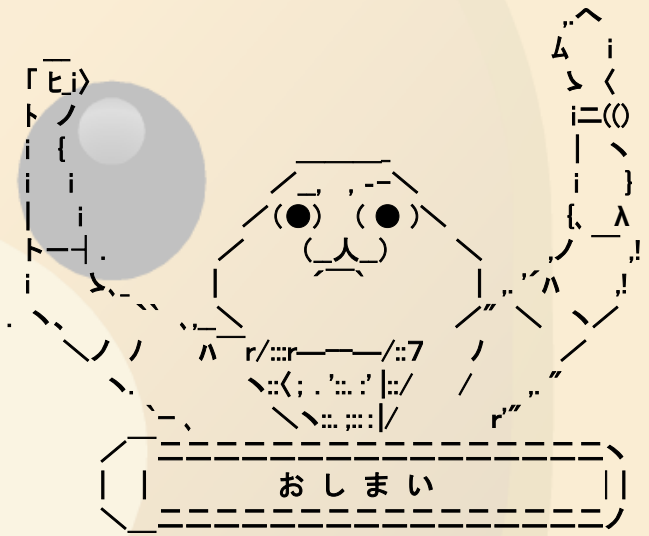
- 簡単なものからやってみようよ。
- 上からじゃなくて、現場から始めてみよう。

参考：オブジェクトクラブ
http://www.objectclub.jp/community/pf/
：にこにこカレンダーシート
http://www.seasoft.co.jp/

平鍋健児さんの文献をほぼ、パクリました。m(_._)m

ご静聴ありがとうございました。

m(_._)m



Special thanks for Yaruo charactors