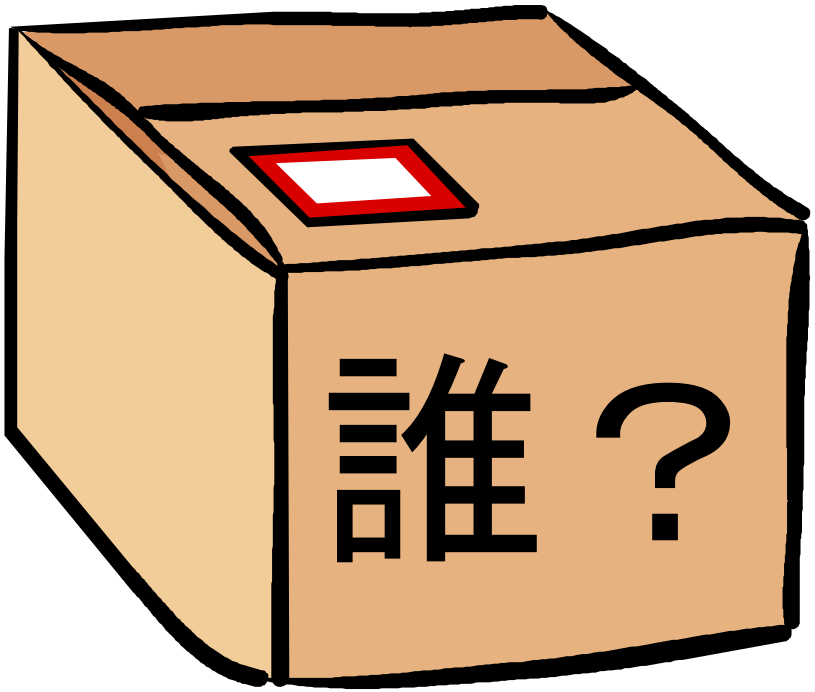
顧客を満足させ続けるための プラクティスについて

Asuka Kamijo



かみじょう ♂ あすか

上條　飛鳥(@nekoasuka)

はじめに

“顧客を満足させ続ける”ためには？

“顧客を満足させ続ける“ためには？

- 以下の4関係を満たし続けるという努力

1. 契約や交渉　　＜　顧客との関係
2. プロセスやツール＜　人との対話
3. 計画の遵守　　＜　変化への対応
4. ドキュメントの精度＜　動作するプロダクト

一歩踏み込んで

“顧客を満足させ続ける”ための プラクティスとは？

まず注意事項

これから紹介するプラクティス（手法）は
適用しただけでは、
“顧客を満足させ続けること”
にはつながりません

極力専門用語を避けて書いています
そのため若干の意味の違いがあります
※本質は変わりません

プラクティスのおしながき

- ユーザの言葉で書いた要件定義書
- 開発関係メンバー全員での工数見積もり
- 付箋に作業項目を書き出して貼り出す
- WBSのビジュアル化
- 2名態勢でのプログラミング
- 検証用プログラムを作ってから開発を始める
- プログラムの内部構造のみを綺麗にする
- いつでもリリース可能な状態にしておく

プラクティスのおしながき

- ユーザの言葉で書いた要件定義書
- 開発関係メンバー全員での工数見積もり
- 付箋に作業項目を書き出して貼り出す
- WBSのビジュアル化
- 2名態勢でのプログラミング
- 検証用プログラムを作ってから開発を始める
- プログラムの内部構造のみを綺麗にする
- いつでもリリース可能な状態にしておく

導入コストが比較的低く、
効果の高い3つをご紹介

プラクティス①

開発関係メンバー全員での工数見積もり

開発関係メンバー全員で見積もる

具体例：読破までの所要時間

(子ども向けの装丁だ)

(何かの専門書のようだ)

(細かい文字が多そうな雑誌だ)

どれくらいかかるかを、1・2・3・5ptのいずれかで見積もる
※各資料は開いてはいけない

具体例：読破までの所要時間

(子ども向けの装丁だ)

(何かの専門書のようだ)

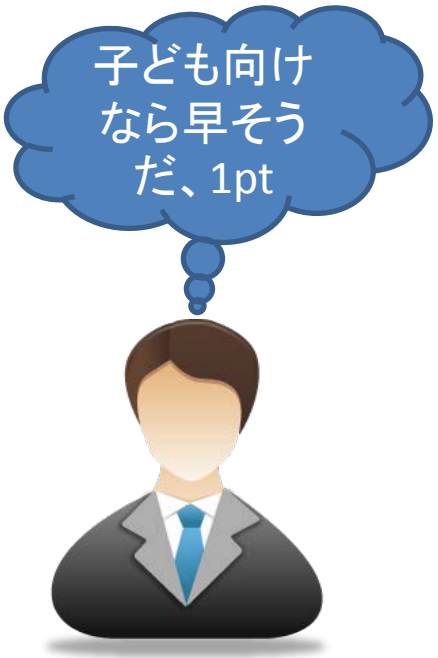
(細かい文字が多そうな雑誌だ)

1つずつ、開発担当全員で、相対的に見積もる

見積もりを公開する時まで口に出してはいけない

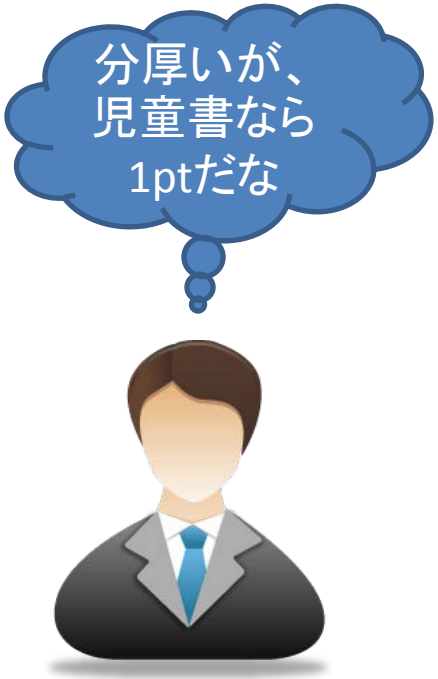
具体例：読破までの所要時間

(子ども向けの装丁だ)



(何かの専門書のようだ)

(細かい文字が多そうな雑誌だ)



具体例：読破までの所要時間

(子ども向けの装丁だ)

1pt

(何かの専門書のようだ)

1pt

(細かい文字が多そうな雑誌だ)

1pt

具体例：読破までの所要時間

(子ども向けの装丁だ)

1pt

(何かの専門書のようだ)

(細かい文字が多そうな雑誌だ)

具体例：読破までの所要時間

(子ども向けの装丁だ)

1pt

(何かの専門書のようだ)

(細かい文字が多そうな雑誌だ)

見積もりが一致しない場合は、少数派から順に意見を聴く

5pt

5pt

3pt

具体例：読破までの所要時間

(子ども向けの装丁だ)

1pt

なぜ、3ptなんですか？

(何かの専門書のようだ)

でも誰が読むかは分からないから、調べる時間は必要だと思う

(細かい文字が多そうな雑誌だ)

以前読んだが、調べながらでもさくさく読めたよ

具体例：読破までの所要時間

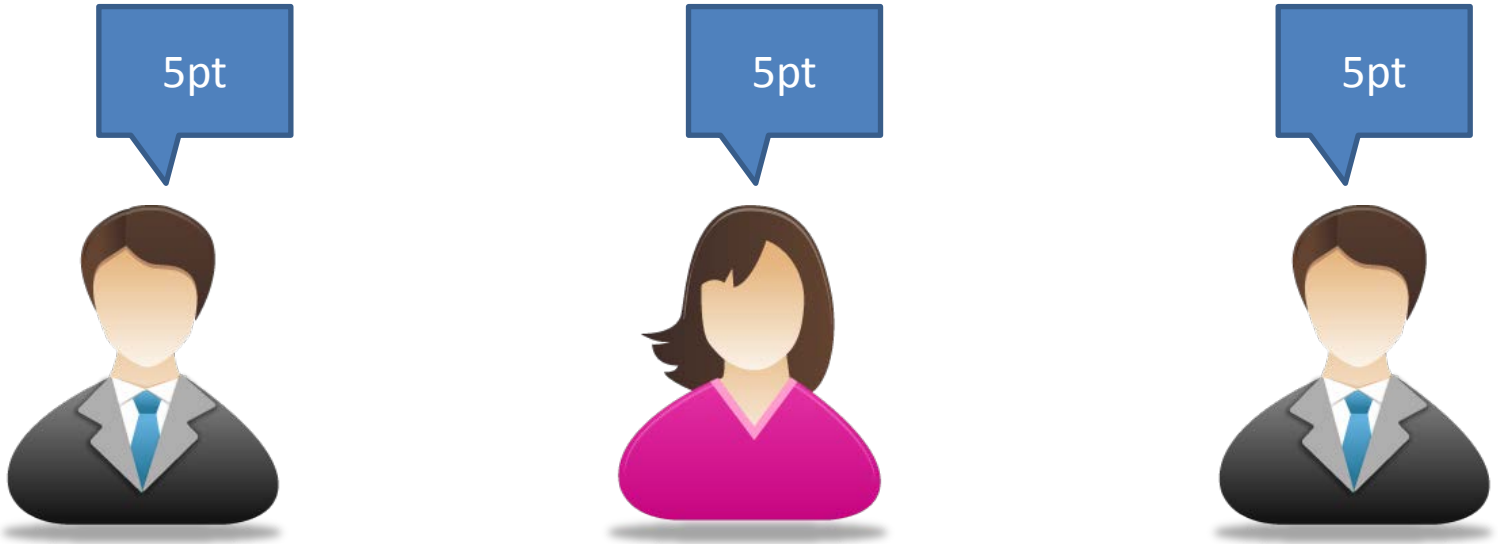
(子ども向けの装丁だ)

1pt

(何かの専門書のようだ)

(細かい文字が多そうな雑誌だ)

話し合ったらもう一度見積もる



具体例：読破までの所要時間

(子ども向けの装丁だ)

1pt

(何かの専門書のようだ)

5pt

(細かい文字が多そうな雑誌だ)

3pt

全ての見積もり対象について、同様に実施する
3回実施しても意見が合わない場合は、
最大値か、平均値に最も近いポイントを採用する

プラクティス②

WBSのビジュアル化

よくある光景

機能ID	担当者	開始日	終了日	進捗
ID XXXX	Asuka	2013/3/30		80%

進捗を消してみる

機能ID	担当者	開始日	終了日
ID XXXX	Asuka	2013/3/30	

締め切り近くに一気に入る

本当に終わったの？

機能ID	担当者	開始日	終了日
ID XXXX	Asuka	2013/3/30	2013/3/31

切り口そのものを変えてみる

- 機能ごとに、相対的な値を振る
- 合計値を出す
- 一定期間毎に、どれだけの値を消化できる？

切り口そのものを変えてみる

相対的な値で見積もり、
機能一覧に加える
担当者は記入しない

機能ID	ポイント	期日	担当者
TEST01	1	2013/4/30	
TEST03	3	2013/4/30	
TEST06	3	2013/5/10	
TEST02	5	2013/5/15	
TEST05	8	2013/6/1	
TEST04	5	2013/8/1	

切り口そのものを変えてみる

実際に少し作り、
どの程度時間がかかるか計測する
担当者は立候補で募る

機能ID	ポイント	期日	担当者
TEST01	1	2013/4/30	Bob
TEST03	3	2013/4/30	Jane
TEST06	3	2013/5/10	
TEST02	5	2013/5/15	
TEST05	8	2013/6/1	
TEST04	5	2013/8/1	

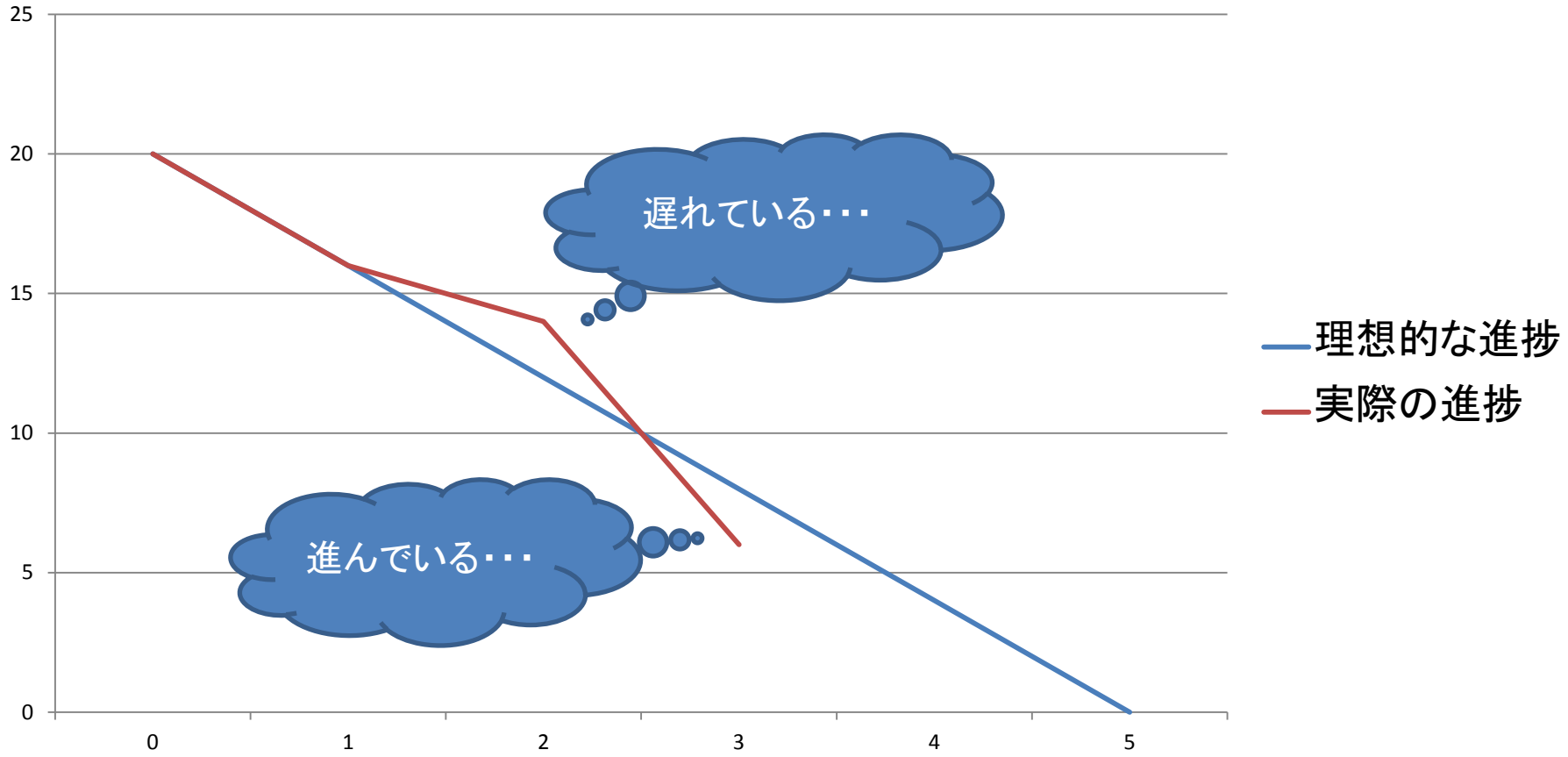
切り口そのものを変えてみる

実作業時間を記入しておく
→1d で 4ポイント消化できた
残り21ポイントは、1d * (20 / 4) ≒ 5dで消化できそう

機能ID	ポイント	期日	担当者	実作業時間
TEST01	1	2013/4/22	Bob	2h
TEST03	3	2013/4/22	Jane	5.5h
TEST06	2	2013/4/26		
TEST02	5	2013/4/26		
TEST05	8	2013/5/6		
TEST04	5	2013/5/6		

切り口そのものを変えてみる

ポイントの消化をグラフ化して貼り出す



プラクティス③

付箋に作業項目を書き出して貼り出す

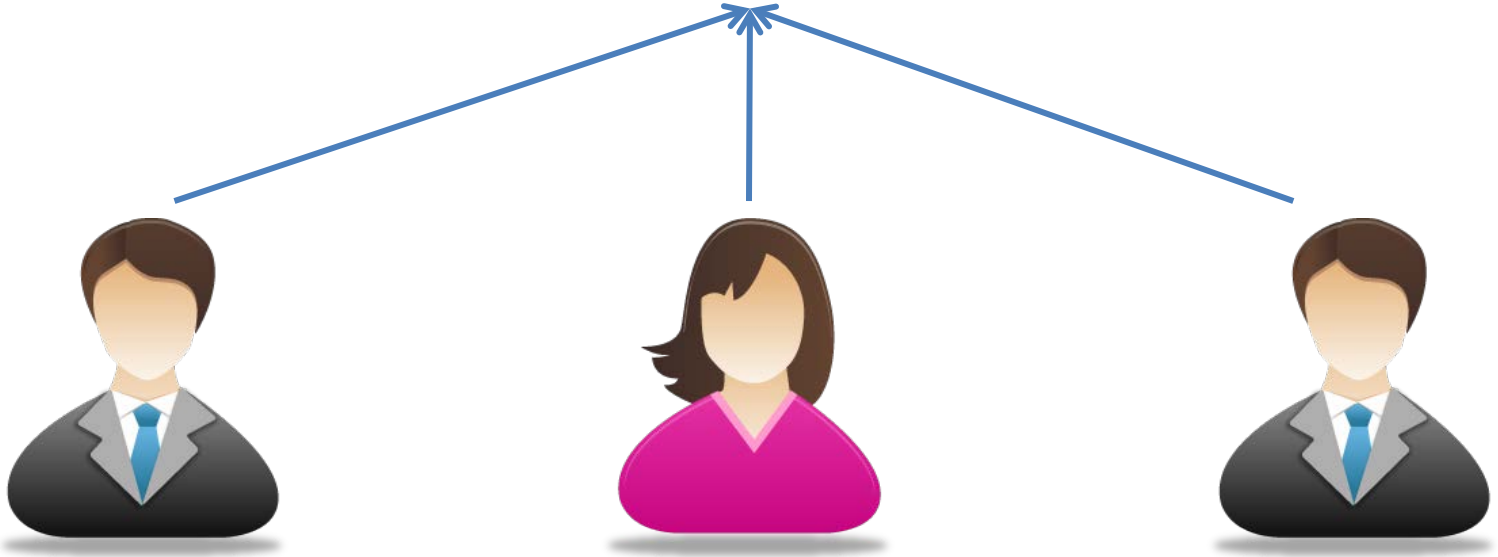
見えない作業はやる気を削ぐ

見えない作業量

作業を可視化する

環境構築 5pt	設計（在庫確認画面） 3pt	開発（在庫確認画面） 2pt
テスト計画書作成 8pt	テスト（在庫確認画面） 3pt	QA回答 #111 1pt

例えば付箋に全て書き出す



TODOとして貼りだす

TODO

- 環境構築 5pt
- テスト計画書作成 8pt
- 設計（在庫確認画面） 3pt
- テスト（在庫確認画面） 3pt
- 開発（在庫確認画面） 2pt
- QA回答 #111 1pt

担当するタスクに立候補し、DOINGへ

TODO

テスト計画書作成
8pt

設計（在庫確認画面）
3pt

テスト（在庫確認画面）
3pt

開発（在庫確認画面）
2pt

QA回答 #111
1pt

DOING

環境構築
5pt　Asuka

完了したら、DONEへ

TODO	DOING	DONE
テスト計画書作成 8pt		環境構築 5pt　Asuka
設計（在庫確認画面） 3pt		
テスト（在庫確認画面） 3pt		
開発（在庫確認画面） 2pt		
QA回答 #111 1pt		

全員が見る場所に配置・共有する

TODO	DOING	DONE
テスト（在庫確認画面） 3pt	テスト計画書作成 8pt　Jane	環境構築 5pt　Asuka
開発（在庫確認画面） 2pt		設計（在庫確認画面） 3pt　Bob
QA回答 #111 1pt		

実務へ展開するためには？

プラクティスの活用

実際にやってみましょう

- 説明したプラクティスを体験しましょう
- ４人１組のチームを作成します
- ４人の中で、以下のいずれかに該当する方は、“顧客”（兼まとめ）役を、他の方は“開発担当”役をしていただきます
  - 最も職位が高い方
  - 最も勤続年数の長い方
  - 説明したプラクティスについて熟知されている方

まずは、簡単なゲームをしましょう！

- 皆さんは歴史に残るプロジェクトのメンバーです
- プロジェクトの始まりから終わりまでを演じます
- セリフは1～100までの整数です
- 1，2，3，・・・と好きな数まで数え上げましょう
- 100になるまで、順番に数え上げ続けます
- プロジェクトに関する思いを胸に発声しましょう
- 今朝一番早く起きた方から時計回りに演じましょう

5分！

それでは歴史を紡ぎましょう！

100プロジェクト

本番に移りましょう！

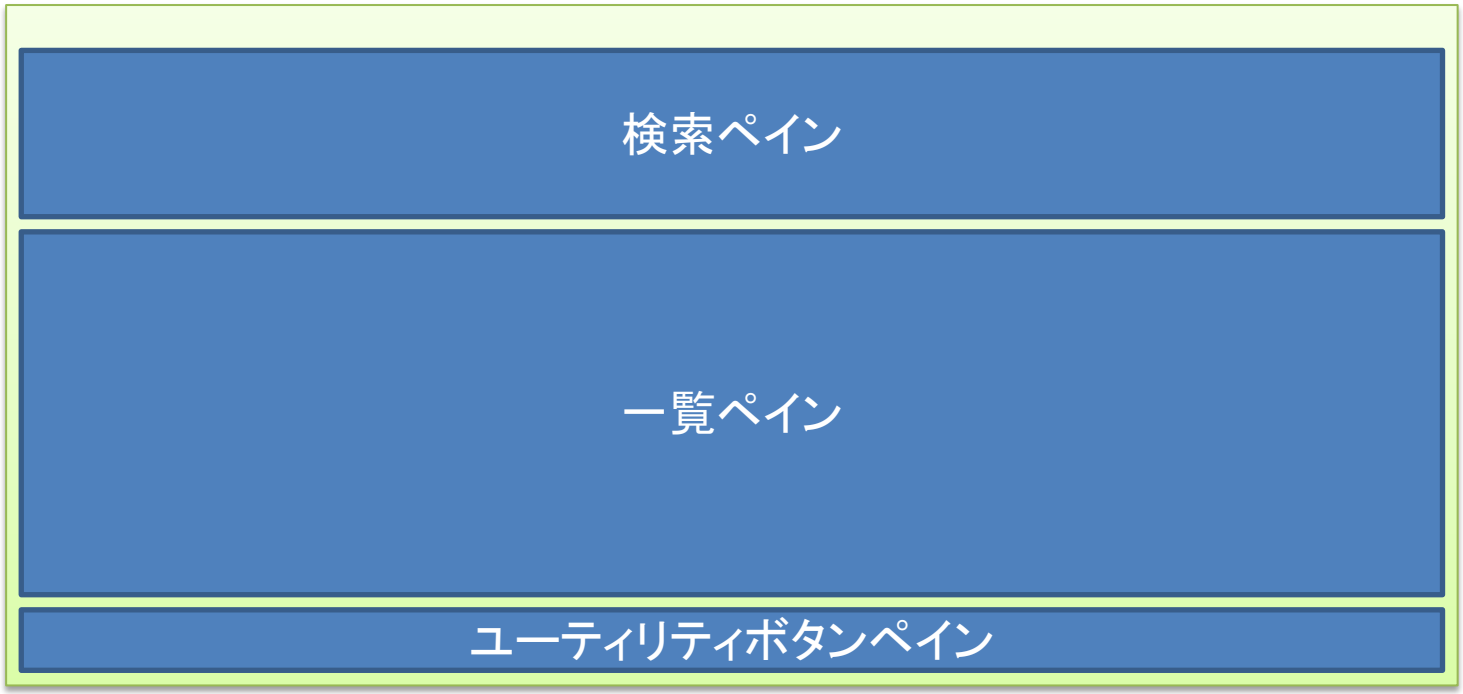
プラクティス体験ゲーム

配布物の確認

- ゲームの進行方法　４枚
- “Requirement”と書かれたカード
  - 30枚
- “Event”と書かれたカード
  - 10枚
- A3用紙　2枚 / A4用紙　3枚
- カラーペン　１組
- サイコロ（6面×2, 8面×2)
- 付箋

開発対象について

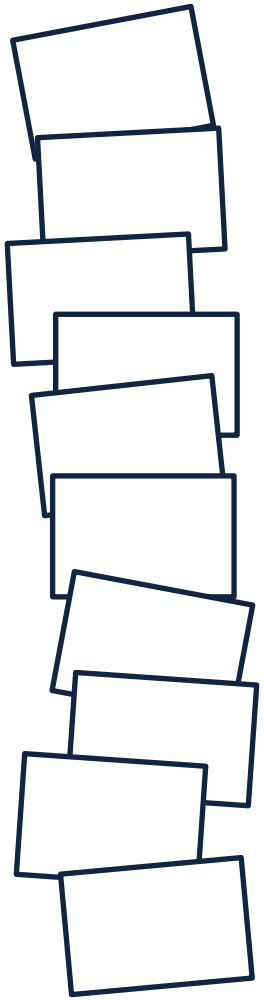
- 在庫管理システム
  - Windowsアプリケーション（下図はイメージ）



さぁ、“顧客役”の出番です！

- #01～#10と書かれたカードを手にとる
- テーブルの端に、縦１列に並べる
- 大事だと思う順に並べる
- “顧客役”１人で決定する

※“開発担当役”は進捗を表すグラフや
TODO管理表の準備をする



5分！

まずは、何が大切か考えよう

優先順位付け

“開発担当役”の出番です！

- 優先順位付けされたカード10枚を見積る
- 設計、実装、テスト総てを考慮し、どれくらいの時間がかかるか見積もる
- 見積りは、10枚のうちだいたい真ん中くらいの時間がかかりそうなカードを見つけるところから始める

※“顧客役”はタイムキーパーをお願いします

気付きをメモすると後で役に立ちます

20分！

どれだけかかる？　皆で話そう

見積り

プロジェクトが走り始めます

- 詳細はルールブックの3ページをご覧ください
- サイコロは誰から投げても構いません
- 時間は正確に計ります(延長はありません！)

※ ゲーム中に困ったら、教えてください

85分！

何が起きるか？　まずはやってみよう

作業

お疲れ様でした！

終了！

Keep, Plobrem and Try

ふりかえり

プラクティスのおしながき

- ユーザの言葉で書いた要件定義書
- 開発関係メンバー全員での工数見積もり
- 付箋に作業項目を書き出して貼り出す
- WBSのビジュアル化
- 2名態勢でのプログラミング
- 検証用プログラムを作ってから開発を始める
- プログラムの内部構造のみを綺麗にする
- いつでもリリース可能な状態にしておく

“KPT“でまとめてみましょう

- Keep
  - 良かったこと
  - このまま続けて行きたいこと
- Problem
  - 気付きや問題に思ったこと
  - 改善していきたいこと
- Try
  - チャレンジしてみようと思ったこと
  - Problemの具体的な解決案

まとめ

- 工数見積は、基準を全員で作ること
- 書き出した作業は、立候補制で実施すること
- 残作業量は、ひと目で分かるようにすること

本資料の作成に当たって参考にした文献

参考文献

アジャイル開発とスクラム

顧客・技術・経営をつなぐ協調的ソフトウェア開発マネジメント

平鍋 健児／野中 郁次郎著　　翔泳社

http://www.amazon.co.jp/dp/4798129704

気になること、気付いたことを教えて下さい

質疑･応答