
アジャイル開発手法スクラムの基礎と ゲーム開発導入事例

CEDEC 2010

アジェンダ

スクラムの基礎

スクラム導入事例

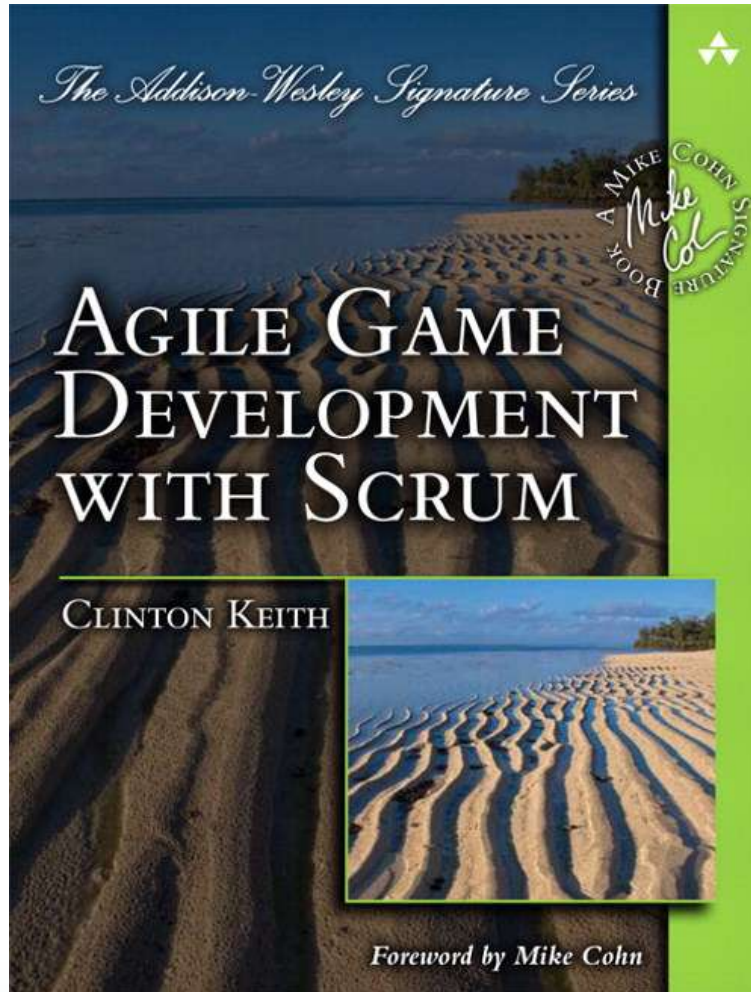
まとめ

さらに学びたい方へ

質疑応答

第1章

スクラムの基礎



この本をベースに
資料を作成
(2010年6月発売)

著者は米国で
ゲーム用スクラムの
コーチをしている
Clinton Keith

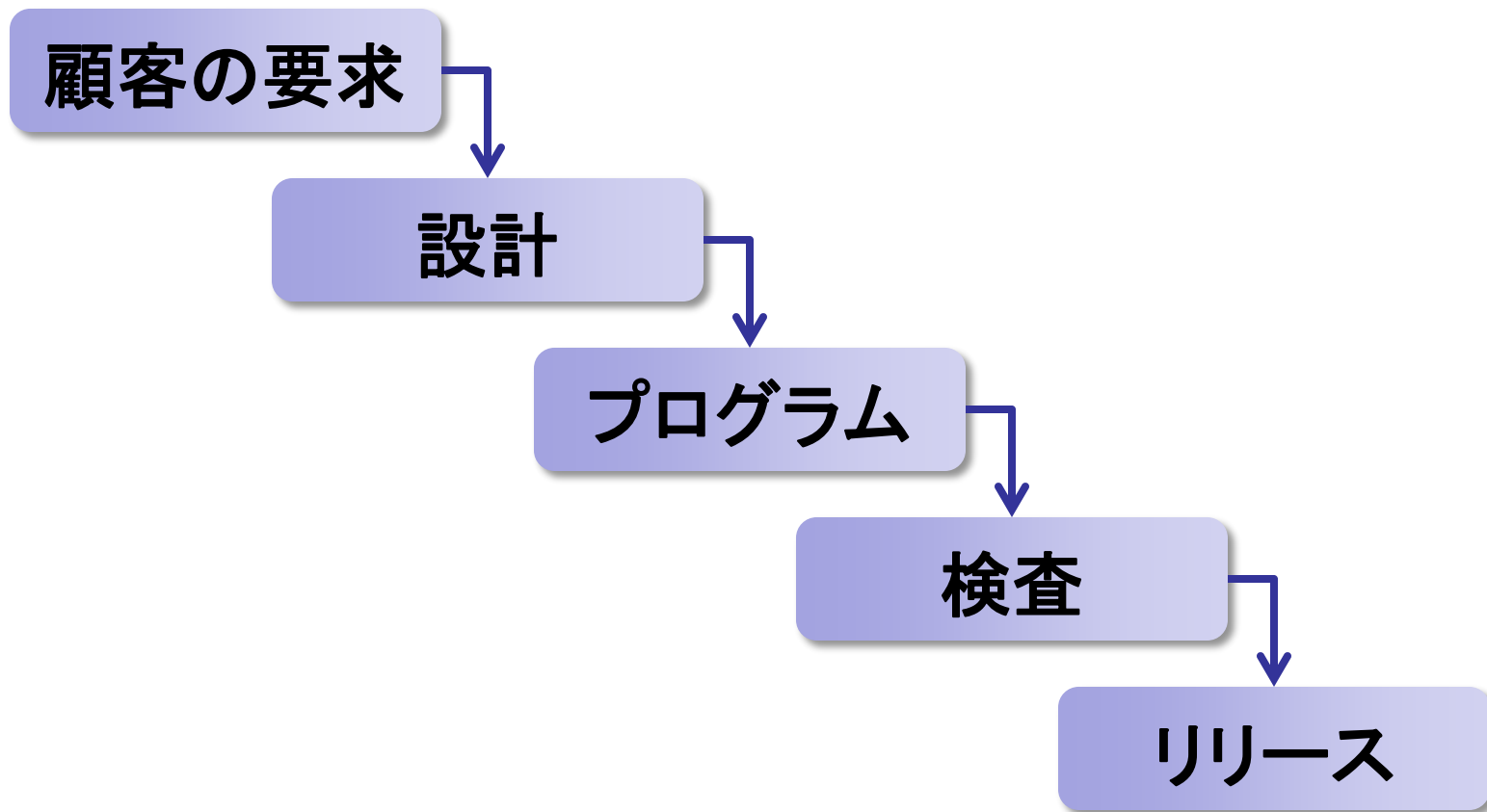
→ゲーム会社
High moon studiosのCTO

※若干本来のスクラムとは
違う部分があります

スクラム

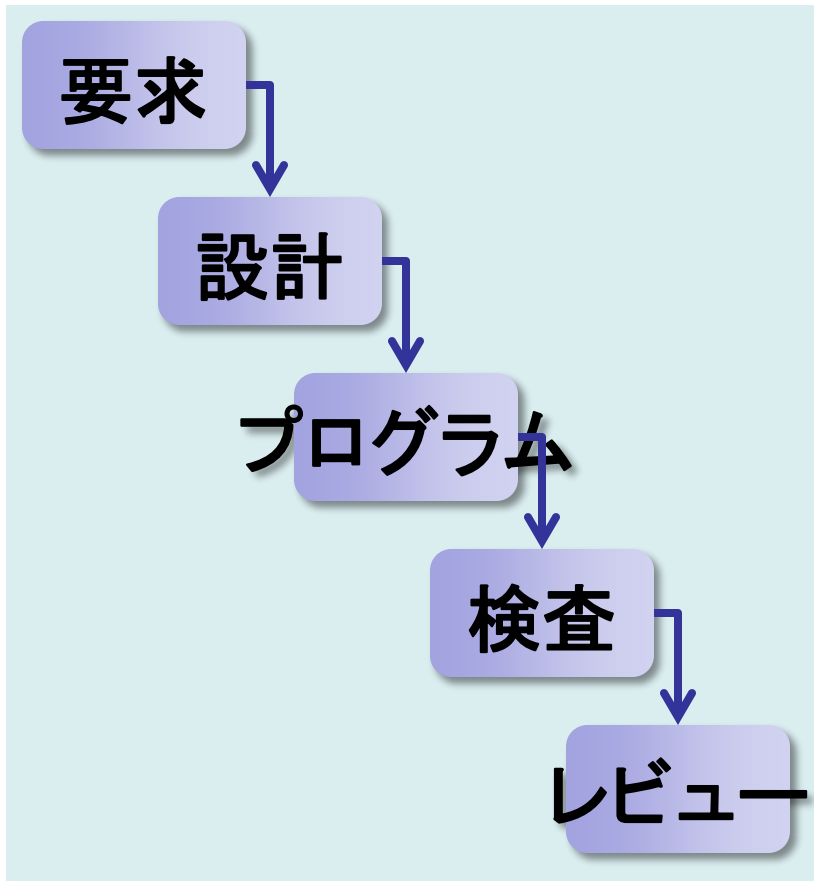
= **Agile**開発手法という
ソフトウェア開発手法の
中の1つ

ウォーターフォール型開発

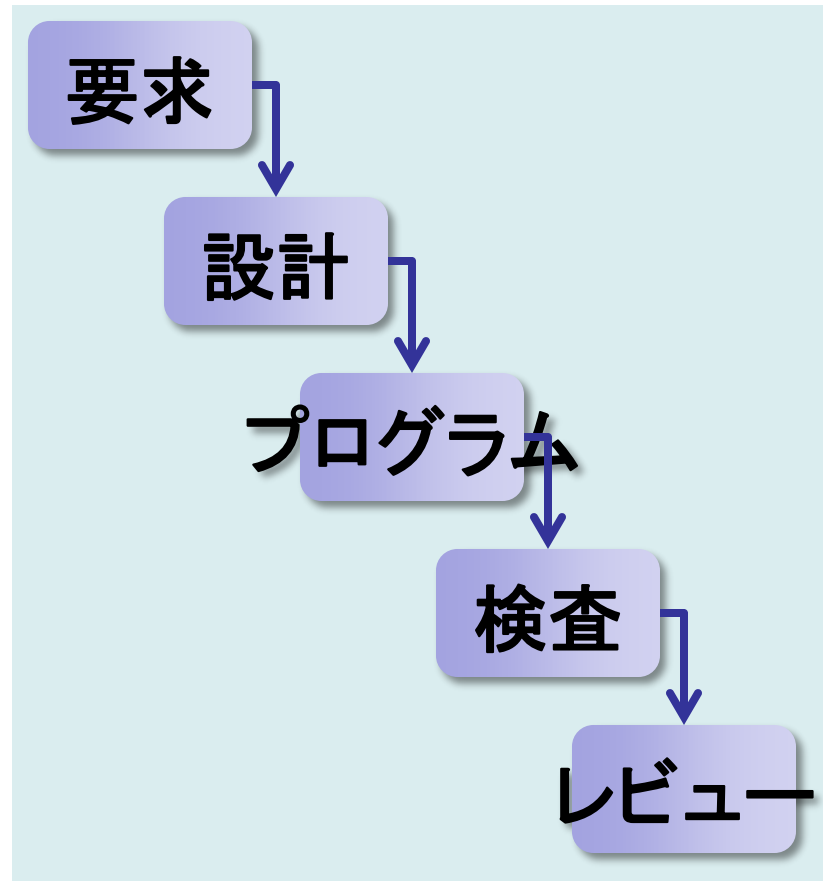


数カ月～数年

アジャイル型開発



繰り返し



繰り返し

質問

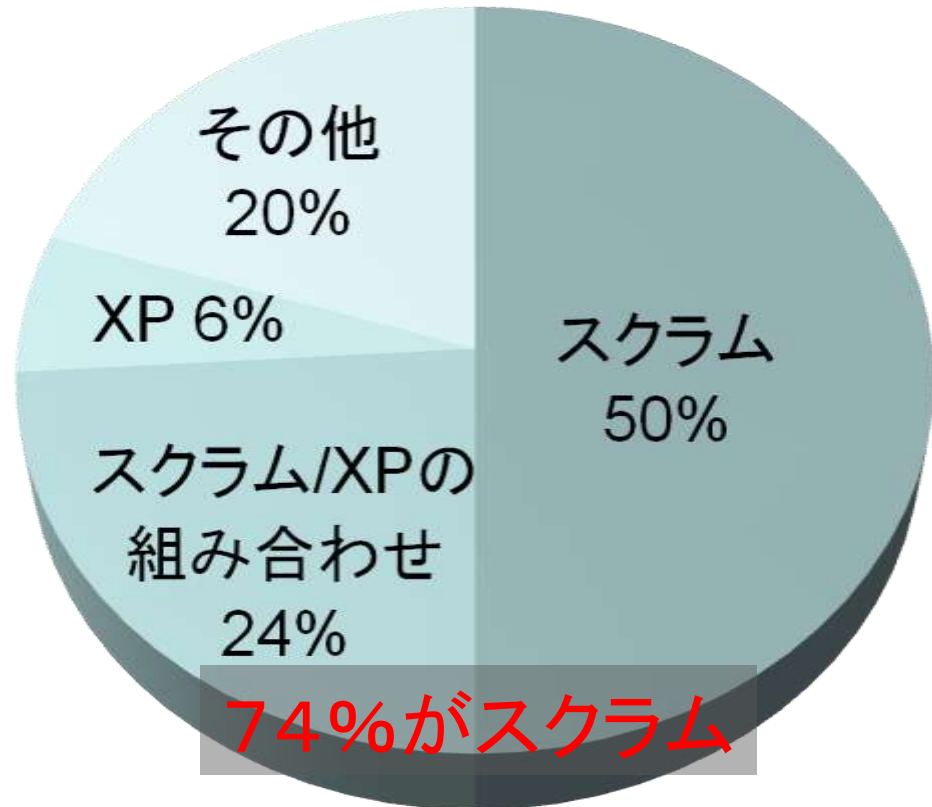
ゲーム開発は
「ウォーターフォール型開発」が
近いですか？

「アジャイル型開発」が
近いですか？

アジャイル開発手法の種類

- Evo
- Crystal
- DSDM
- **XP**
(eXtreme Programming)
- **スクラム**
- Lean Software Development

米国で利用されているアジャイル手法



米VersionOne

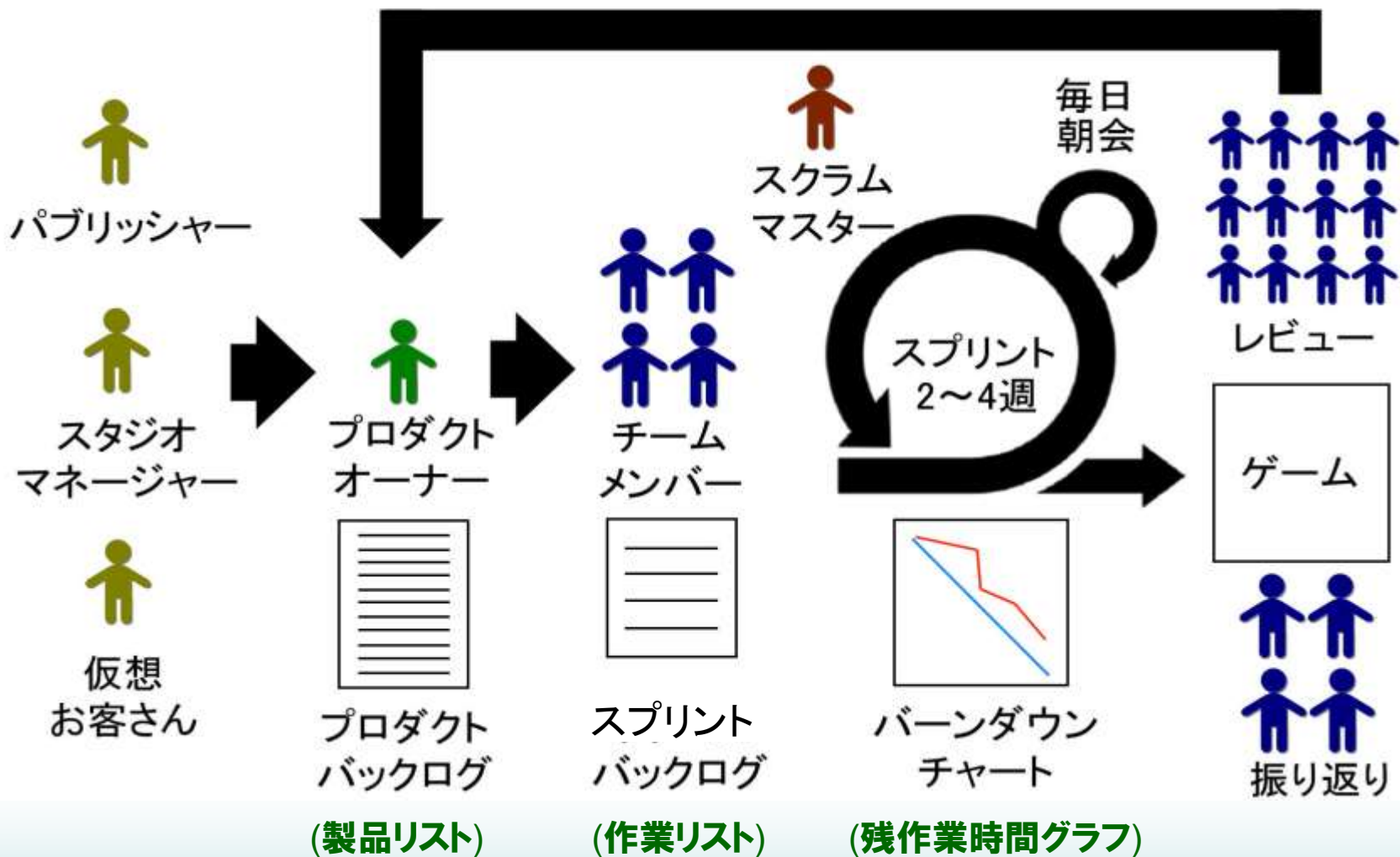
State of Agile Development Survey 2009

スクラムの技法

1. チーム分けと役割
2. リリース計画
 プロダクトバックログ(製品リスト)
3. スプリント計画(作業計画)
 スプリントバックログ(作業リスト)
4. スプリント(作業)
 バーンダウンチャート(残作業時間グラフ)
5. デイリースクラム(朝会)
6. スプリントレビュー(レビュー会)
7. ふりかえり

※()の中は適当翻訳です

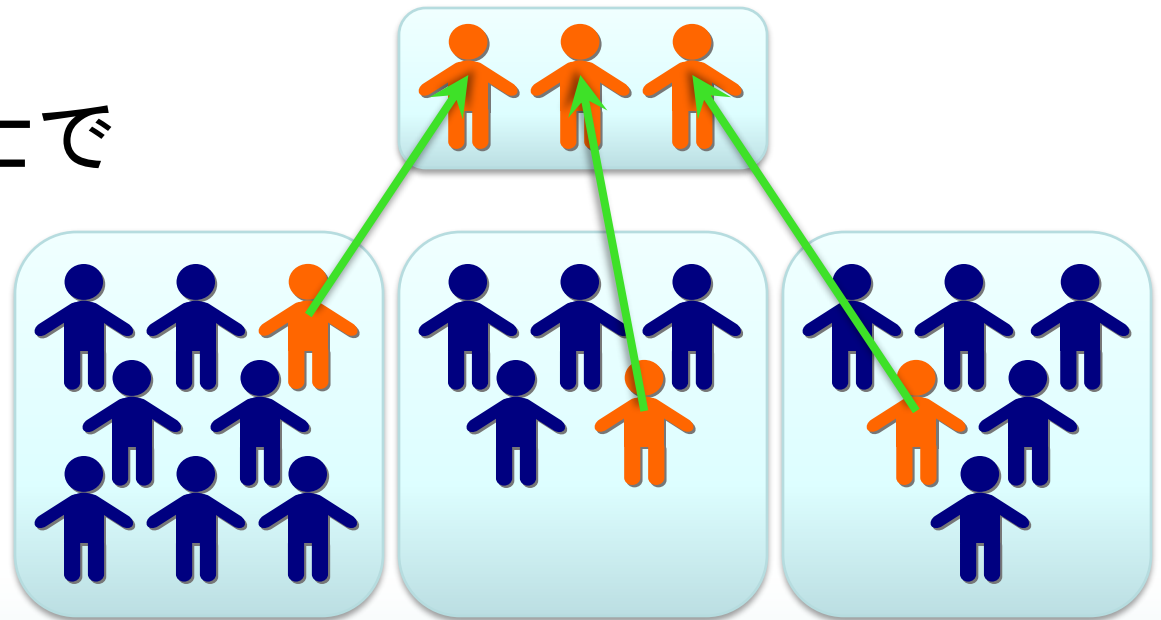
スクラムの流れ



1. チーム分けと役割

- チームの人数は5～9人が理想的
 - ▶ 一体感、情報伝達速度などで理想的な人数

- さらに代表同士でチームを組む(階層化)



チーム分け

● 特徴ごとに職種混同でチームを作成する
→ チーム単体で成果が上げられる事

▶ 例) 乗り物パートチーム

- 企画 2名

 - 車の挙動バランス調整、テストレベルの作成

- デザイナー 2名

 - 乗り物、小物、マップジオメトリの作成

- プログラマー 3名

 - 乗り物制御プログラム、物理、敵AI

- エフェクト 0.25名 ← 複数チームに所属している

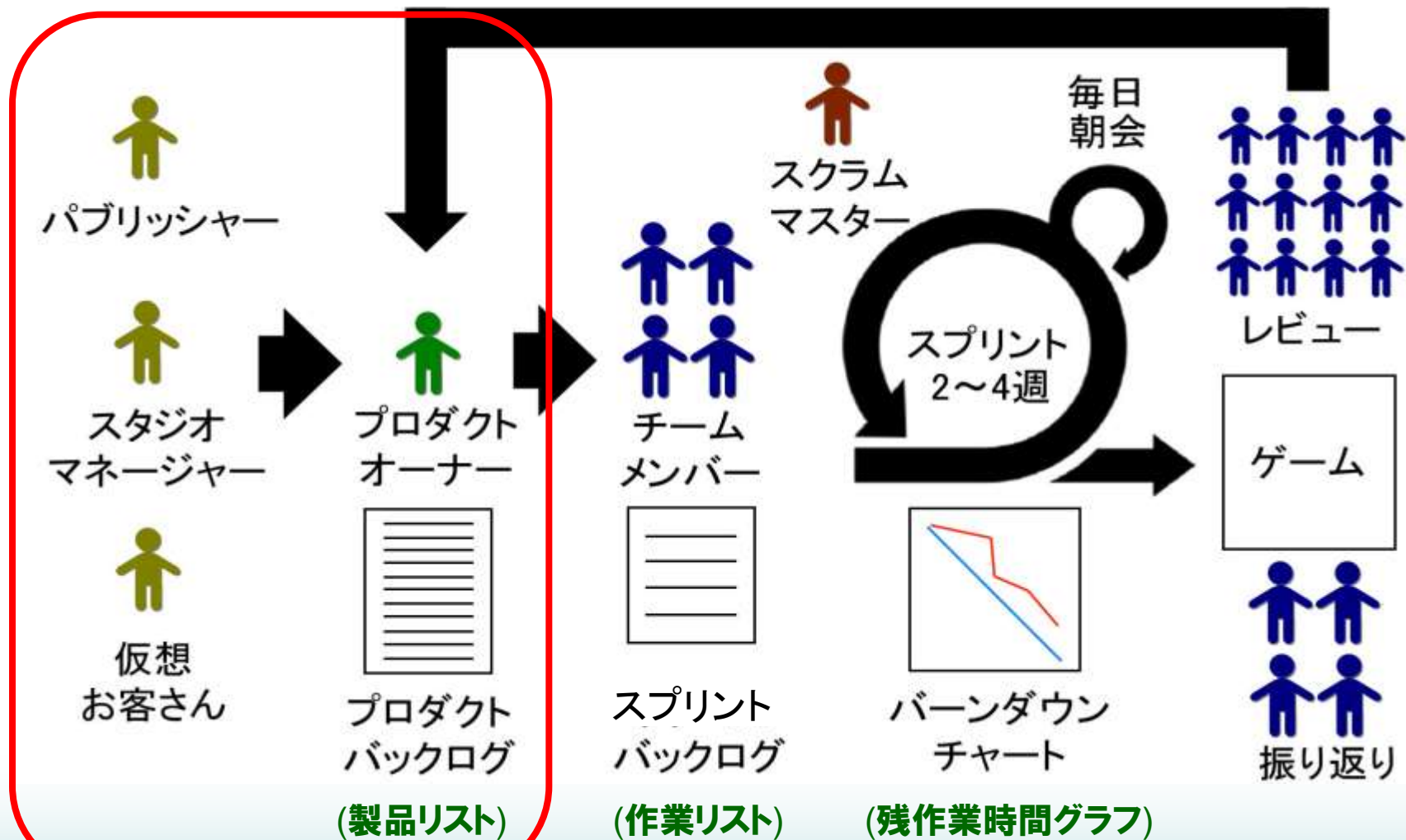
役割 プロダクトオーナー

- チームに所属し、以下の責任を負う人
 - ▶ 黒字になる責任
 - ▶ 顧客と開発者の間に立ち共有されるビジョンを提示
 - ▶ どんな要素をどんな優先順位でゲームに組み込むかの決定
 - ▶ リリース計画、及び受渡日の確立
- 一般的にはプロデューサーかディレクター

役割 スクラムマスター

- 2～4のチームに所属し、以下の役割を担当
 - ▶ 進捗をチェックし、障害を取り除く
 - ▶ 改善促進立案
 - ▶ スプリントレビュー及びふりかえりの準備
 - ▶ コミュニケーション支援
- 管理者ではない。促進者である
- 出来れば開発者と兼任しない方が良い。
 - ▶ 自分の開発作業を優先しがちなため
- プロダクトオーナーと兼任もダメ
 - ▶ プロダクトオーナーにダメ出しなども行うため

2.リリース計画



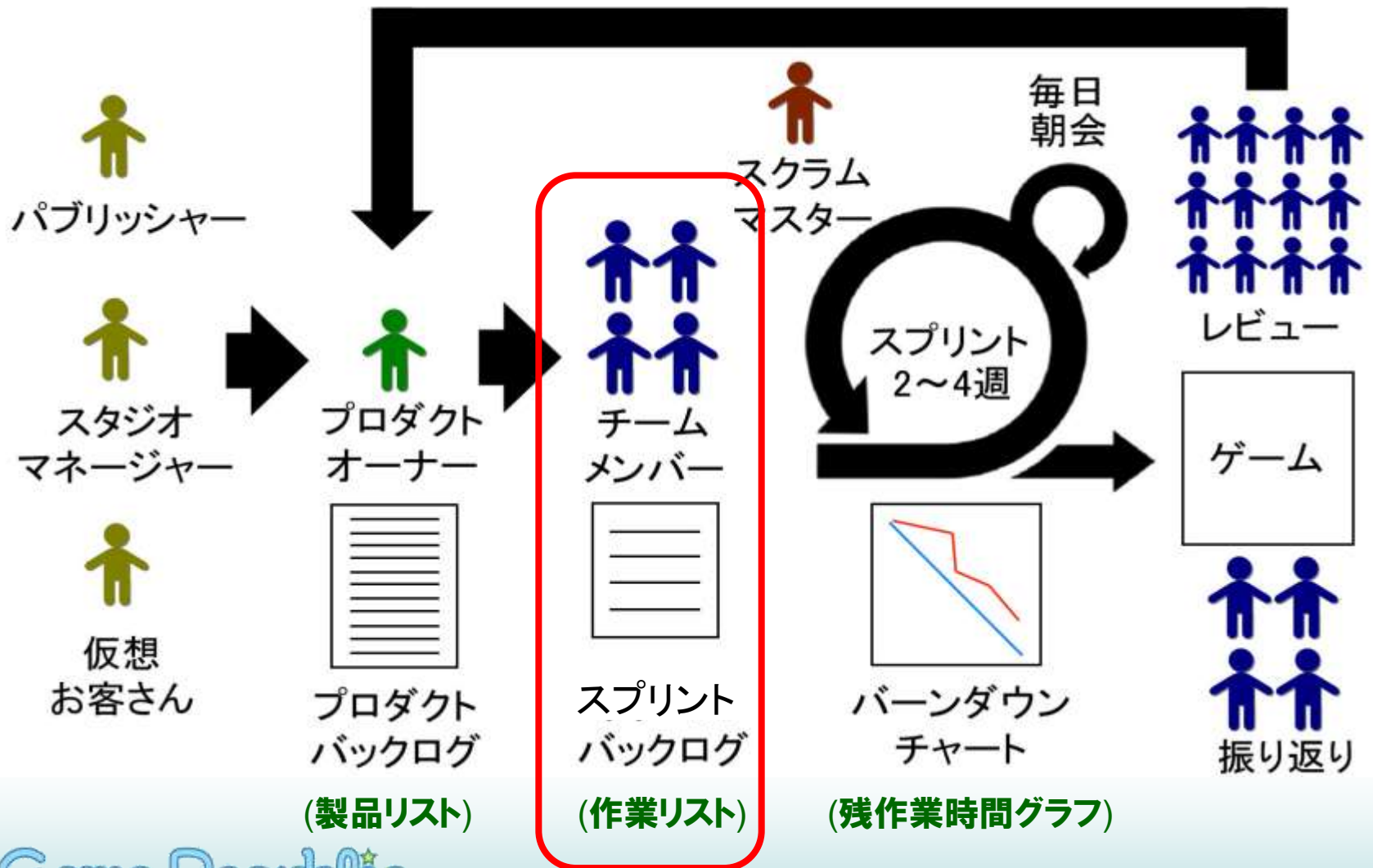
プロダクトバックログ (製品リスト)

内容	重要度	工数	PG	モデル	モーション	音響	エフェクト	企画
車が24車種ほしい。何故ならライバルゲームが19車種だから	800	300	40	70	160	0	20	10
ヘッドライトが付いて欲しい。何故なら車の接近に気づかないから。	1000	20						
テールランプが付いて欲しい。何故なら車の接近に気づかないから。	200	20						
アクセルとブレーキが欲しい。ギアはATだけで大丈夫	1500	300						
車のパラメータをゲーム機上で調整したい。何故なら車種数が多いのでトライ&エラーが大変だから	600	100						

2. リリース計画

- ゲームに必要な機能をまとめたリストを作る
ミーティング
 - ▶ プロダクトバックログ(製品リスト)
- 機能、理由(何故なら)、重要度、工数を記載
- 後で変更、追加可能。ただし追加出来るのは
プロダクトオーナーのみ
- 工数はまずはざっくり。適宜更新
- チーム全員が常に見れる場所に置く

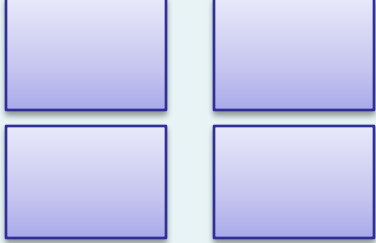
3. スプリント計画(作業計画)



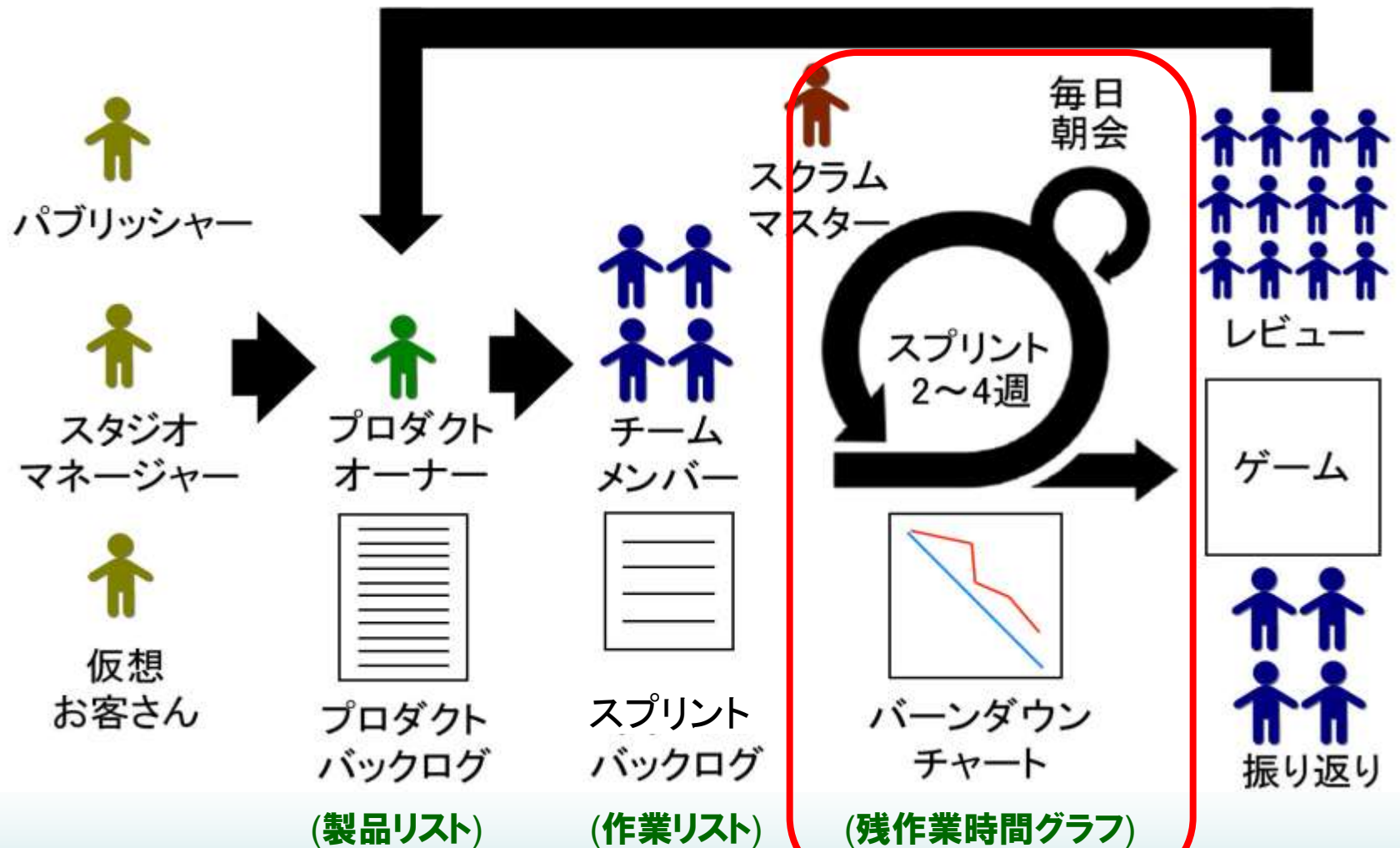
3. スプリント計画(作業計画)

- 今回のスプリントで行う内容を決めるミーティング
 - ▶ スプリントバックログ(作業リスト)
- 内容はプロダクトバックログの中からプロダクトオーナーが決める
- その内容がスプリント内で実装できるか、チームの中で実業者達が作業単位(タスク)に分割して時間を見積る

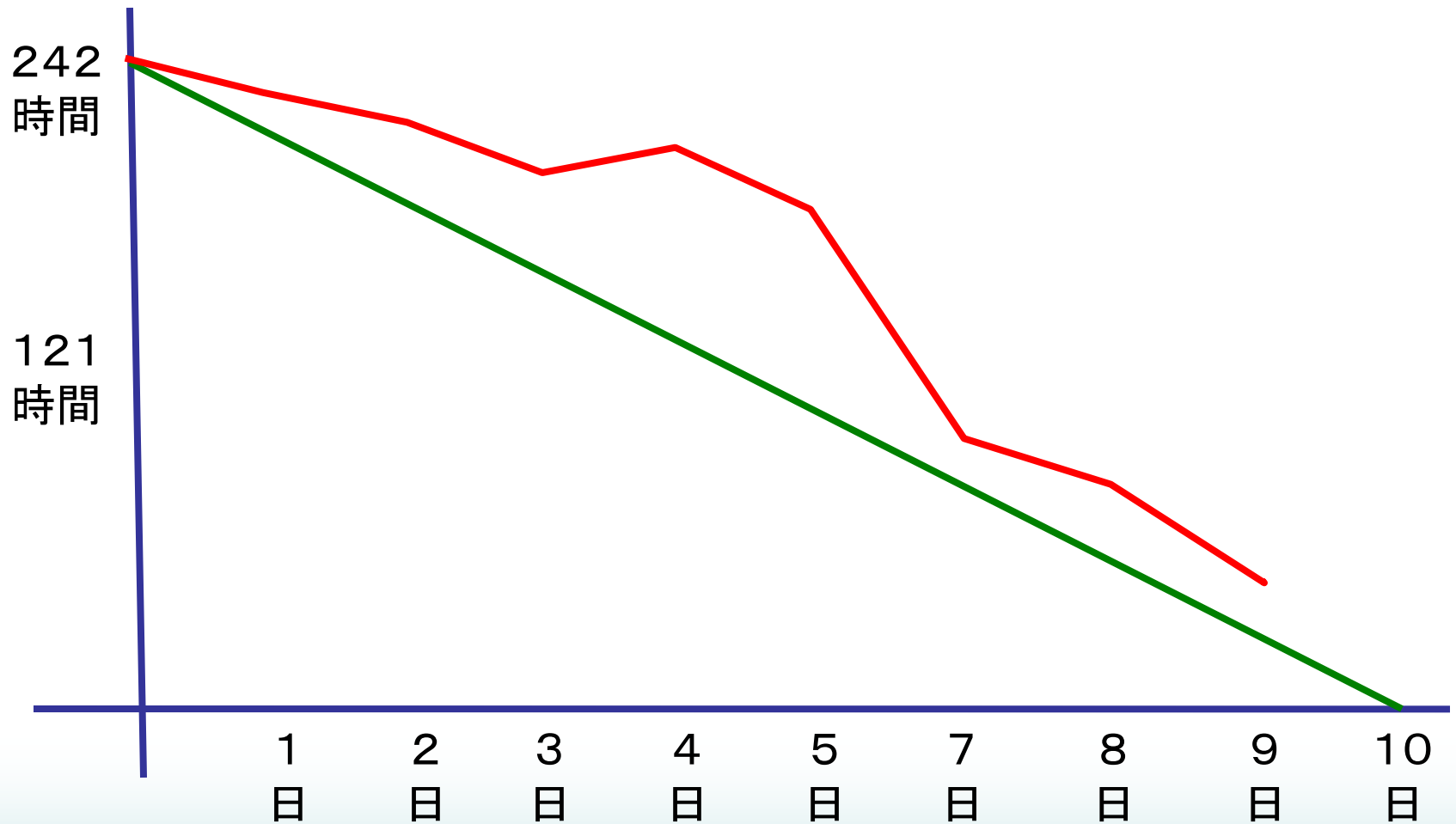
スプリントバックログ

プロダクト バックログ	タスク	作業中	完了
			
			
			

4. スプリント(作業)



バーンダウンチャート(残作業時間グラフ)



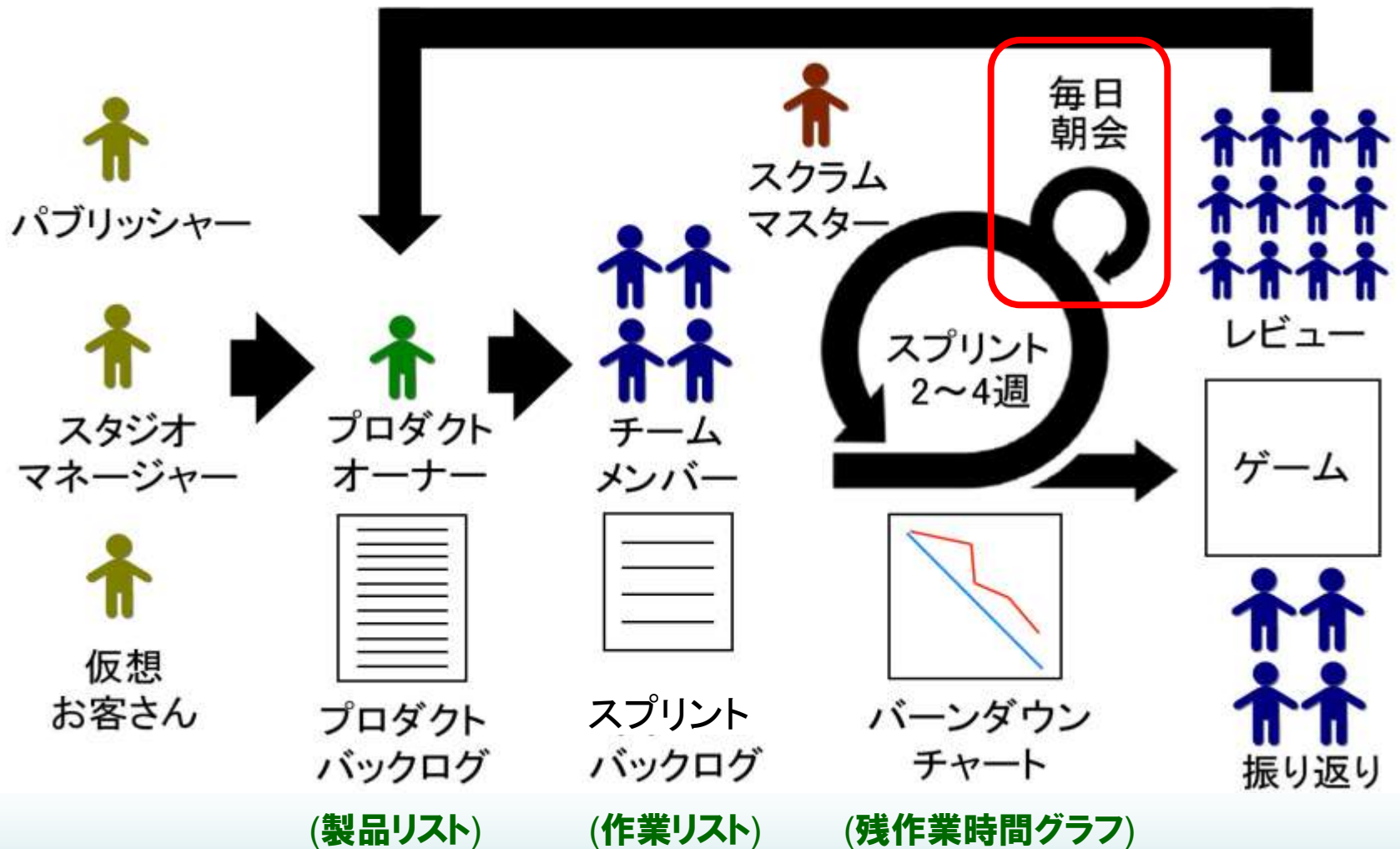
4. スプリント(作業)

- スプリント計画で決めた作業を消化していく
- 期間は2週間～1ヶ月
→ゲーム業界は変化が大きいので
2週間くらいが良いらしい？
- 付箋とバーンダウンチャートを使用して
進捗を更新する
 - ▮ 紙が良い。常に間接視野に入る事でメンバーに
スケジュールに対する意識が生まれる
- スプリント中の追加作業は避けるべき
 - ▮ チームの責任と自立性が失われる

スプリントバックログ

プロダクト バックログ	タスク	作業中	完了
	 		
			 
		 	

5. デイリースクラム(朝会)

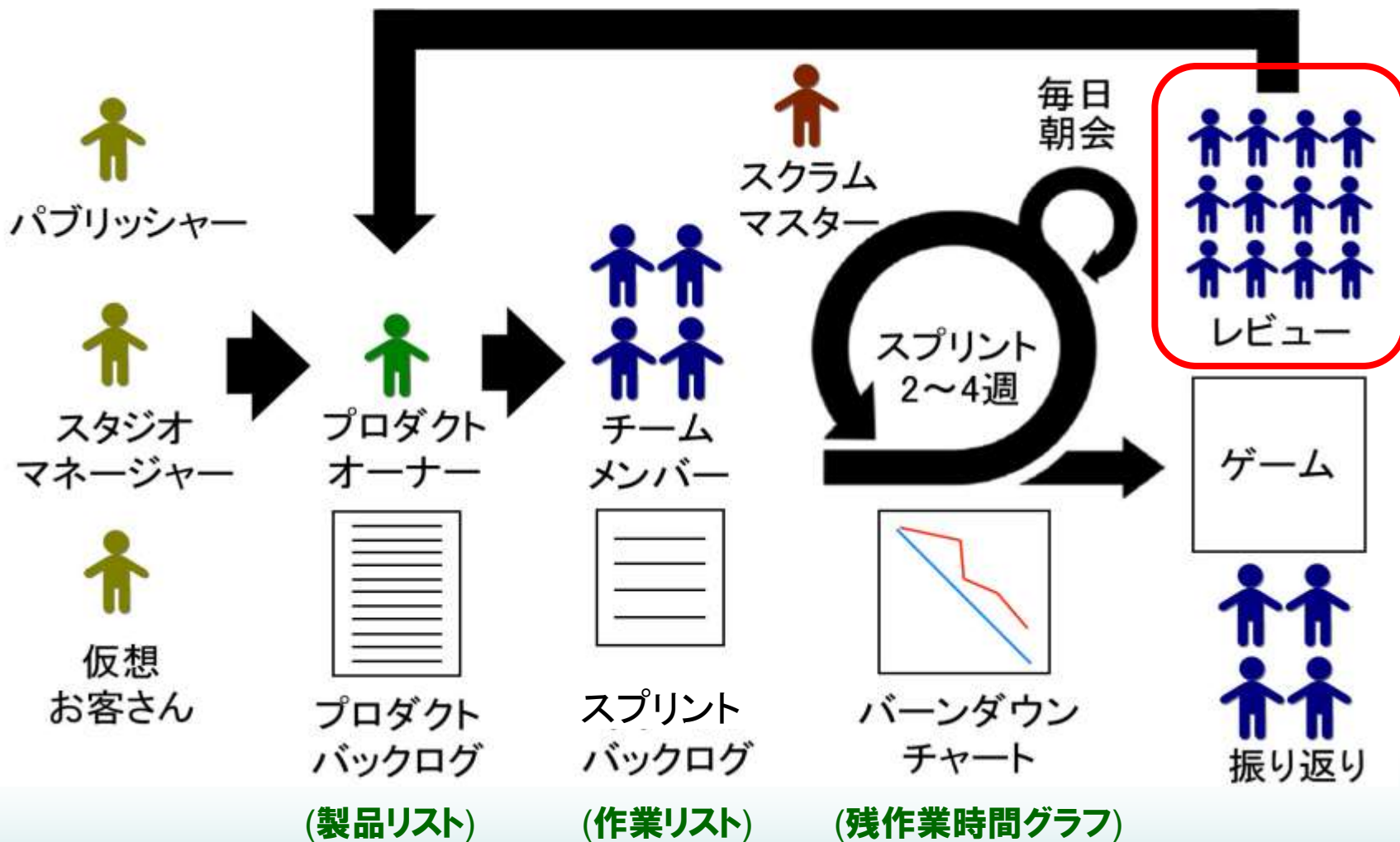


5. デイリースクラム (朝会)

- 平日の決まった時間に行く(毎朝10時とか)
- 15分前後で完了させる
 - ▶ 長くなりそうな案件は、別途ミーティング開催
- 全員立って行う
- 各自が以下を報告
 - ▶ 昨日何をしたか
 - ▶ 今日何をするか
 - ▶ 問題はあるか



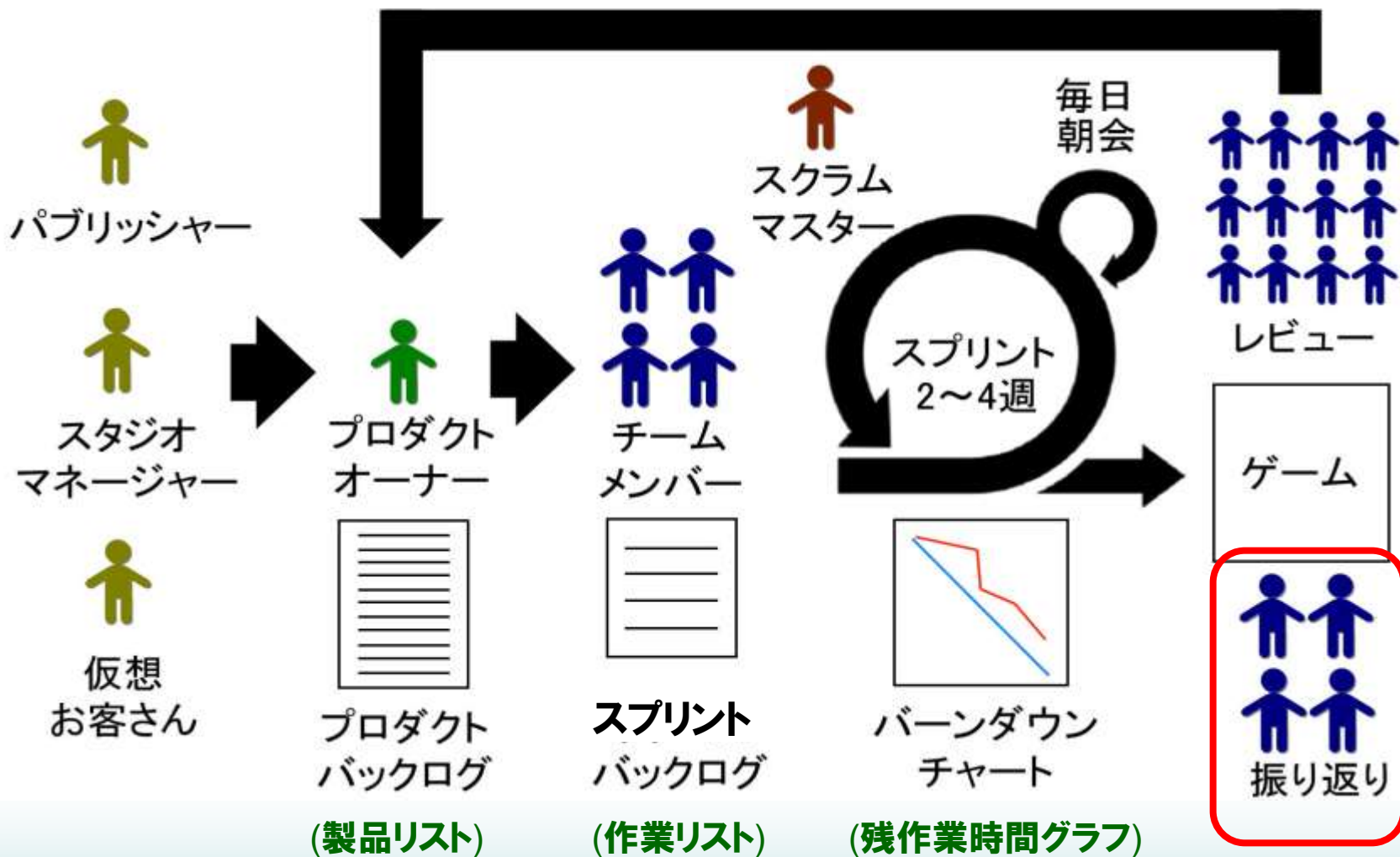
6. スプリントレビュー(レビュー会)



6. スプリントレビュー (レビュー会)

- 成果物の確認会
- スプリントの終了の度に行う
(2~4週に1回)
- スタッフ全員と、ディレクター
プロデューサーに参加して貰う
- パワポなどは使わずに、開発して動いている
状態のものを見ることを中心に進める
- 現状確認、問題、アイデア等を出し合う

7. ふりかえり



7. ふりかえり

- 各チームでそれぞれ今回のスプリントの内容をふり返って改善点を洗い出す
(2～4週に1回)
- 反省会では無く、**改善**する会議
- よくKPT法が使われる

7. ふりかえり KPT法

Keep

今回のボス戦は
今までで
一番面白い

プラグインで
作業効率が
上がった

Try

アップする前に
必ずチェック
する事

モーション出力
プラグインに
チェック機能の
追加を依頼する

Problem

モーションの
トラブルが
多かった

マップの
ロード時間が
長くなった

何故マップのロード
時間が長くなった
原因を調べる

他のボス戦でも同じ要素が
入れられないか確認する

スクラムを導入すると

リソースの破棄が減る ⇨ コミュニケーションの強化

プロダクトバックログ(製品リスト)、デイリースクラム(朝会)
スプリントレビュー(レビュー会)、ふりかえり、機能チーム

管理コストが減る ⇨ 自己管理

チーム単独で成果が上げられる
スケジュールの見える化、自分自身の作業を自分でコミット

スケジュールが明確になる ⇨ 透明化

プロダクトバックログ(製品リスト)、バーンダウンチャート



開発効率が向上する

第2章

スクラム導入事例

プロジェクト紹介

- 2009年末頃に始まった新規プロジェクト
- メンバーは20名～30名
- 過去同じチームでタイトルを完成させた経験あり
- スクラムはもちろん初めて

最初のステップ

プロダクトオーナー
(ディレクター)に
賛同してもらおう

説明する時のポイント

● スクラムの実績を伝える

- ▶ BioWareやValveなどで採用

● 味方になってくれそうな人を見つけておく

- ▶ リードプログラマー、リードデザイナー

● 正しく理解して貰うため 専門用語はなるべく使わない

- ▶ 特に「スプリント」と「バックログ」は馴染みが無い

ディレクター
「確かに良いかも。やってみますか」



チーム編成

● スクラムチームは3つ

- ▶ プログラマー 4名で「プログラマーチーム」
- ▶ プランナー 4名で「プランナーチーム」
- ▶ デザイナー 9名で「デザイナーチーム」
- ▶ プロジェクトアシスタント 1名

職種でチームを分けてる？

職種でチームを分けた理由

● 機能でチームを分けるほど人数が居なかった

▸ 特にプログラマー

● 大きな変化を避けた

今回取り入れたスクラムの技法

技法	採用	以前から
機能でチーム分け	×	×
スクラムマスター	△	×
プロダクトバックログ(製品リスト)	○	○
スプリントバックログ(作業リスト)	○	△
デイリースクラム(朝会)	○	×
バーンダウンチャート(残作業時間グラフ)	△	×
スプリントレビュー(レビュー会)	○	△
ふりかえり	×	×

× スクラムをやりたい

**開発効率を上げる
ために
スクラムを導入したい**

**いきなりスクラムの
全ての技法を導入しない**

→変化が大きすぎて逆に危険

チーム編成

● プロダクトオーナー

- ▶ ディレクターが担当

● スクラムマスター

- ▶ プロダクトオーナーが兼任

- 会社の枠組みとして、そういう職種を用意するのは非常に困難だと感じている
- 自分自身が有効性を感じられないと、会社に対して説得しづらい

リリース計画

- プロダクトバックログ(製品リスト)はエクセルで管理
- 以前から似た感じの事を行っていたがスプリントレビュー(レビュー会)の情報が多く入るようになったので精度が上昇

スプリント計画(作業計画)

- スプリントバックログ(作業リスト)を
Redmineで管理

検索:

プロジェクト

チケット

スプリントバックログ

フィルタ

ステータス

未完了

適用 クリア 保存

フィルタ追加:

チケット

- チケットを全て見る
- サマリ
- 変更記録

計画

カレンダー | ガントチャート

カスタムクエリ

- 未修正バグ
- 未完了タスク
- 未完了要望
- 自分の全チケット
- 自分の未完了チケット

#	トラッカー	ステータス	優先度	題名	起票者	担当者	更新日	開始日	期限日	進捗 %
1161	要望	【新規】未着手	通常	2010年03月18日 12:00 PM	2010年03月18日		
1160	要望	【新規】未着手	通常	2010年03月18日 11:57 AM	2010年03月18日		
1159	要望	作業中	通常	2010年03月18日 11:24 AM	2010年03月18日		
1158	要望	作業中	通常	2010年03月18日 11:20 AM	2010年03月18日		
1157	要望	【新規】未着手	通常	2010年03月18日 11:15 AM	2010年03月18日		
1156	要望	【新規】未着手	通常	2010年03月18日 11:13 AM	2010年03月18日		
1153	タスク	作業中	通常	2010年03月18日 18:32 PM	2010年03月18日		
1152	タスク	作業中	通常	2010年03月18日 18:32 PM	2010年03月17日		
1150	タスク	作業中	通常	2010年03月18日 19:07 PM	2010年03月17日	2010年03月24日	
1147	タスク	作業完了	高め	2010年03月18日 11:16 AM	2010年03月17日		
1146	タスク	【新規】未着手	通常	2010年03月17日 17:37 PM	2010年03月17日		
1139	タスク	作業中	通常	2010年03月16日 11:52 AM	2010年03月16日		
1134	要望	作業中	通常	2010年03月16日 11:44 AM	2010年03月15日		
1133	バグ修正	【新規】未着手	通常	2010年03月15日 16:56 PM	2010年03月15日		
1132	要望	【新規】未着手	通常	2010年03月15日 16:49 PM	2010年05月06日	2010年05月07日	
1131	要望	【新規】未着手	通常	2010年03月15日 16:44 PM	2010年05月19日	2010年05月20日	
1130	要望	【新規】未着手	通常	2010年03月15日 16:43 PM	2010年05月17日	2010年05月18日	
1129	要望	【新規】未着手	通常	2010年03月15日 16:29 PM	2010年04月01日	2010年04月02日	
1127	タスク	作業中	通常	2010年03月15日 18:55 PM	2010年03月12日		
1124	タスク	作業完了	通常	2010年03月12日 13:29 PM	2010年03月12日		
1123	要望	作業中	通常	2010年03月16日 11:45 AM	2010年03月12日		
1122	タスク	作業中	通常	2010年03月12日 11:44 AM	2010年03月12日		

1stプレイアブル

期日まで 13日 (2010年03月31日)

予定済みの項目で作業中



24 終了 (35%) 44 未完了 (65%)

概要

- 概要
- スケジュール
- ガントチャート
- チケット
- 新しいチケット
- チャート
- ニュース
- Wiki
- フォーラム
- ファイル
- リポジットリ
- 設定

関連するチケット

- 要望 #931: ...
- 要望 #937: ...
- 要望 #938: ...
- 要望 #939: ...
- 要望 #940: ...
- 要望 #941: ...
- 要望 #1012: ...
- 要望 #1014: ...
- 要望 #1016: ...
- 要望 #1017: ...
- 要望 #1019: ...
- 要望 #1072: ...
- 要望 #1100: ...
- 要望 #1123: ...
- 要望 #1134: ...
- 要望 #1140: ...
- 要望 #1156: ...
- 要望 #1157: ...
- 要望 #1159: ...
- 要望 #1160: ...
- 要望 #1161: ...
- タスク #648: ...
- タスク #813: ...
- タスク #814: ...
- タスク #851: ...
- タスク #887: ...
- タスク #923: ...
- タスク #932: ...
- タスク #1004: ...
- タスク #1005: ...
- タスク #1011: ...
- タスク #1020: ...
- タスク #1038: ...
- タスク #1044: ...
- タスク #1073: ...

バーンダウンチャートの代わりロードマップ

ガントチャート

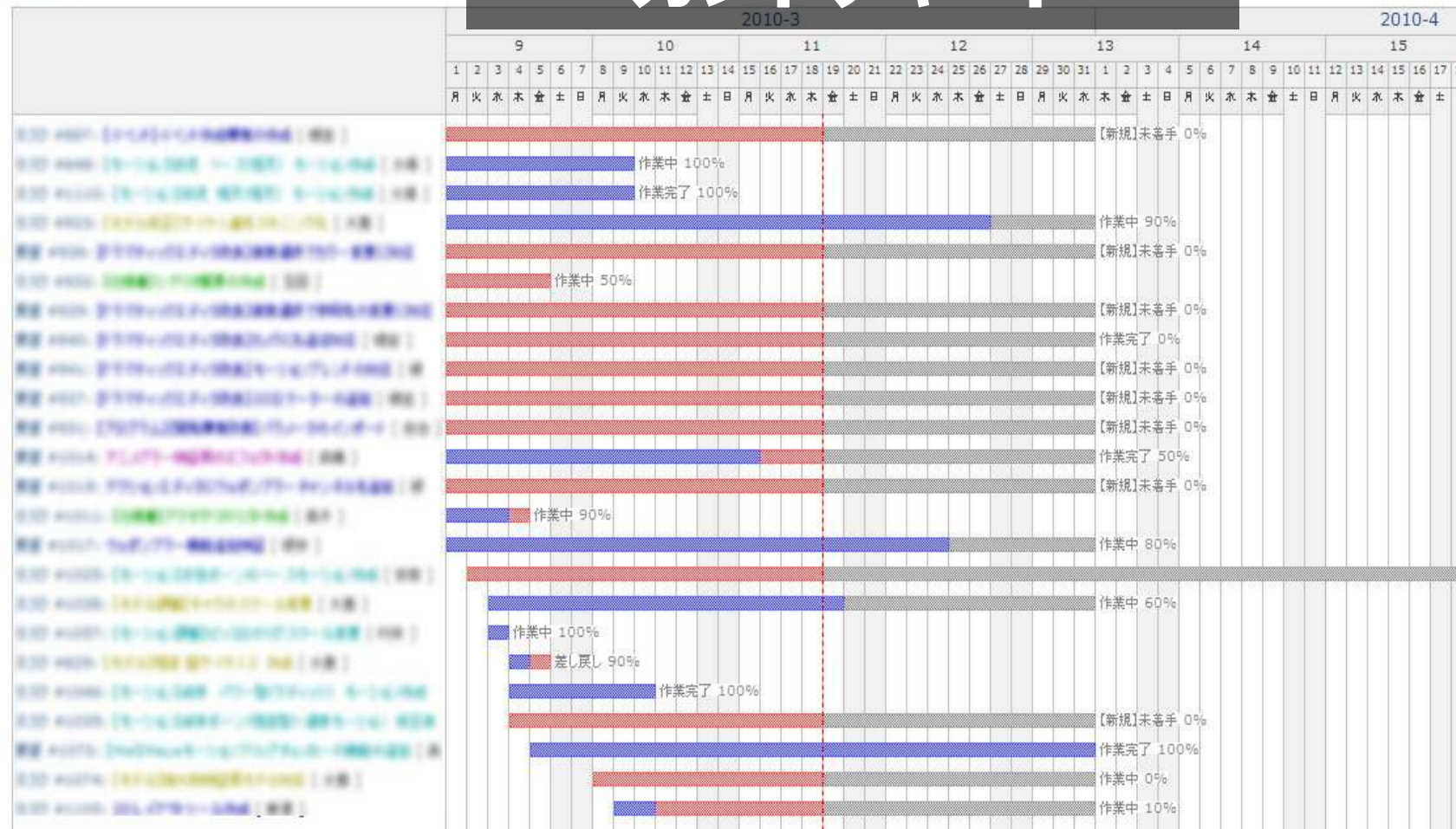
バーンダウンチャートの代わり ガントチャート

フィルタ
 ステータス 未完了 ▼

フィルタ追加: ▼

日付の範囲
6 ヶ月 from 3月 ▼ 2010 ▼

適用 クリア



チケット

チケットを全て見る
サマリ
変更記録

計画

カレンダー | ガントチャート

カスタムクエリ

未修正バグ
未完了タスク
未完了要望
自分の全チケット
自分の未完了チケット

スプリント計画(作業計画)

- スプリントバックログはRedmineで管理
 - ▶ 紙は流石に面倒なのと、チーム的には以前からTracを使っていたので導入に抵抗など無し
 - ▶ ガントチャート形式なので、変化の少ないデザインチームでは使いやすいが変化の多いプログラマーチームでは少々使いにくい
 - ▶ 作業を始める時と作業の終了時以外は見ない
 - 紙との併用が良いかも？

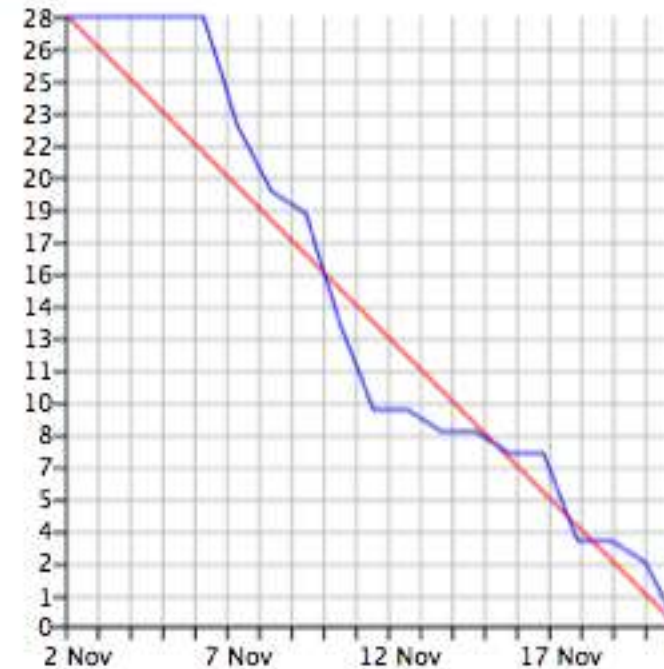
余談 Pivotal Tracker

The screenshot shows the Pivotal Tracker interface for a project named "sakuratweets". The top navigation bar includes "Projects", "PIVOTAL TRACKER", and "Welcome, Jun Ohwada". Below the navigation bar, there are tabs for "Done", "Current", "Backlog", and "Icebox". The "Current" tab is active, showing a list of stories with "Accept" and "Reject" buttons. The "Backlog" tab is also visible, showing a list of stories with "Start" and "Finish" buttons. The "Icebox" tab is also visible, showing a list of stories with "Start" buttons. The interface is in Japanese.

© 2010 Pivotal Labs / Policies / Feedback

Release: XPI 0.3.0 - working! (done)

Points remaining



スプリント(作業)

● 期間を4週間に設定

- ▶ パブリッシャーへの提出が1ヶ月単位のため
- ▶ 今まで1ヶ月単位だったため

● 4週間だとやはり長い

- ▶ 1年のプロジェクトだとレビューが8回くらいでα版になってしまう
- ▶ スプリント中に**変更**が入ってしまう
- ▶ 次は**2週間**で出来ないか検討中

スプリントレビュー(レビュー会)

- 会議室にチームメンバー全員(20名)集めた。
- 時間は2時間を設定
- プロダクトオーナーが最初に説明して、その後自由にプレイ



スプリントレビュー(レビュー会)

● 酸欠

● 意見を出す人は数名
それ以外の人にはボンヤリ

● 後半ダレた

● 少しではあるが、良い意見と悪い意見が
ヒアリング出来た

● 全員がゲームの進行状況を確認出来た

スプリントレビュー(レビュー会)

- 時間は1回30分x8セット交代制
- 出席者は最大6人。
ディレクターとプロアシは常に参加
- 付箋に
「良かった所」
「改善したい所」
を書いて
ホワイトボードに貼る
- 机にお菓子
(アメちゃん)



スプリントレビュー(レビュー会)

- 酸欠にならずに済んだ
- 沢山の良い意見と非常に沢山の悪い意見がヒアリング出来た
- ダレずに進行出来た
- 全員がゲームの進行状況を確認出来た

- 議論の情報共有が出来なくなった

スプリントレビュー(レビュー会)・議論

- レビューで出た「良かった所」「直したい所」を全てExcelに写し、全意見に対してリーダー達で**議論**を行う。
- 議論は**開発現場**で行い、スタッフ全員が内容を把握できるようにする。
- その意見を元にプロダクトバックログを修正する。
- 全「直したい所」に対して返答を書いたExcelを**全員にメール**する

ふりかえり

- 会社に全くない習慣だったのと
当時は効果が不鮮明だったため
導入を見送った。
- ゲーム改善の機会は
レビュー会で対応出来るが
制作パイプライン改善の機会がない
- 絶対導入するべき

第3章

まとめ

スクラムの効果

- メンバーの意見が多く拾えるようになり
より**ビジョン**が**共有**出来るようになった
- リソースの破棄は減った…かな？
ゲームのジャンルすら違うので不明
- スケジュールは**明確**になった
- 自己管理化はまだ
→不足: チーム分け、ふりかえり

チームが生き生き と 仕事をしている

→ビジョンが末端のスタッフにまで
行き届いている

会社にも変化が…

朝の風景



スクラムを10ヶ月続けてみて

- スクラムをやっていないチームでも自主的に朝会を始めた
- Redmineを使用しているプロジェクトが7つに増えた

良い習慣が勝手に広がった

第4章

さらに学びたい方へ

アジャイル参考資料

本

- ▶ Agile Game Development with Scrum
- ▶ アジャイルな見積りと計画づくり
- ▶ アジャイルソフトウェア開発 スクラム

Web (以下の2つは特にオススメ)

- ▶ ScrumGuide_ja.pdf
- ▶ THE_SCRUM_PRIMER_ja.pdf

■ GamePM 勉強会

→佐々木さん主催の
主にゲーム業界用プロマネ勉強会

■ すくすくスクラム

→IT系。なんと参加費無料

■ Agile系カンファレンス

→ IT系。数多くの導入事例が聞ける

ご清聴ありがとうございました