ゲーム開発のための プロジェクト管理

－　開発フェーズに合わせた柔軟で効率的な管理手法　－

2014年3月13日　板垣 衛

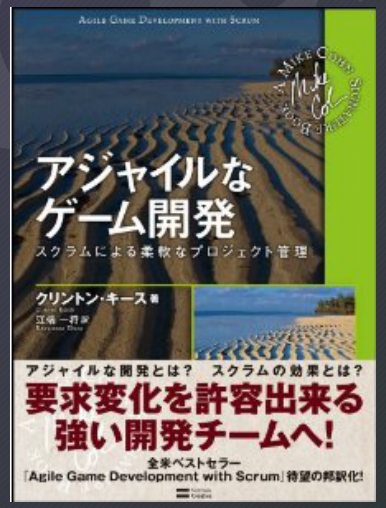
参考書籍

- 本書は、下記の書籍の内容に大いに基づいた上で、更なる柔軟性の追求と単純化を図る

アジャイルなゲーム開発

スクラムによる柔軟なプロジェクト管理

著者：クリントン・キース
訳者：江端一将
発行所：ソフトバンククリエイティブ



本書の趣旨

- 全スタッフへの説明
- プロジェクト管理とその用語に対する認識の統一
- プロジェクト管理手法の理解

※ゲーム制作現場へのスクラムの浸透が広がっていることを背景に、
あえてスクラムの用語を積極的に使用する方針とする

全員参加型のプロジェクト管理

- なぜ全員が理解する必要があるのか？
  - 誰かに管理してもらうプロジェクト進行ではない
  - 全員参加型のプロジェクト管理を実践

開発フェーズの認識

- コンセプトフェーズ
- プリプロダクションフェーズ
- プロダクションフェーズ
- ポストプロダクションフェーズ

【注】この図の通り、大局的な区切りで開発を行うことを規定するのではなく、まずは概念的な説明

開発フェーズの認識：コンセプトフェーズ

- アイディアを創出する
- プロトタイプを作る
  - アイディアが理解でき、完成形が想像できるもの
  - 動作するプログラムとは限らない

開発フェーズの認識：プリプロダクションフェーズ

- ゲームの「面白さ」を明確にする
- ダメならダメと判断する
- 量産可能な状態にする

開発フェーズの認識：プロダクションフェーズ

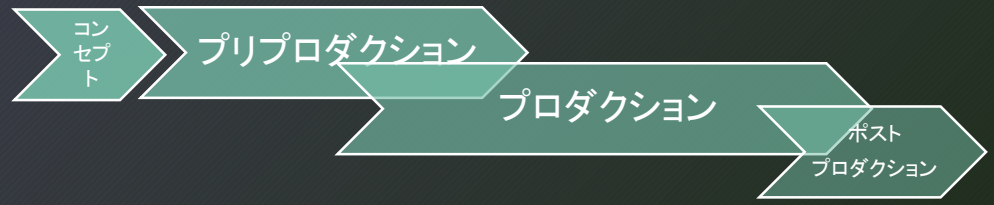
- コンテンツを量産する
  - 本当に「量産可能」になるまでこのフェーズには入らない
  - 量産が進んでいるものの大量手直しはあってはならない

開発フェーズの認識：ポストプロダクションフェーズ

- デバッグ
- 品質向上
  - 基本的に、機能追加は行わない

開発フェーズの認識：実際のフェーズ進行

- 要件ごとの開発フェーズが重なり合って進行
  - 例えば、「フィールドマップ」、「戦闘システム」、「ワールドマップ」、「メインメニュー」などの要件
- 隣接するフェーズも重なり合う



開発フェーズの認識：遵守すべき基本ルール

- プリプロダクションを重視
- ゲームの「面白さ」が確定するまで、プロダクションには絶対に入らない

※あくまでも基本ルール
※「変更要件」に対する柔軟な対応方法は、以降で説明

本書が目指す管理手法

- コンセプトフェーズ ～ プリプロダクションフェーズ
  - スクラム
- プロダクションフェーズ
  - 量産パイプライン管理システム
- ポストプロダクションフェーズ
  - BTS（Bug Tracking System）

※開発フェーズに最適化した管理手法で効率化を図る
※「量産パイプラインシステム」は、本書が提案する独自システム

アジャイル開発

本書は、アジャイル開発の実践を目的とする

開発の柔軟性を追求：
アジャイル開発

- アジャイル（Agile）＝「迅速」
- 迅速かつ軽量な開発手法のこと
- 「スクラム」は、アジャイル開発方法論の一つ

※本書は、ゲーム開発に「スクラム」を導入することを促す

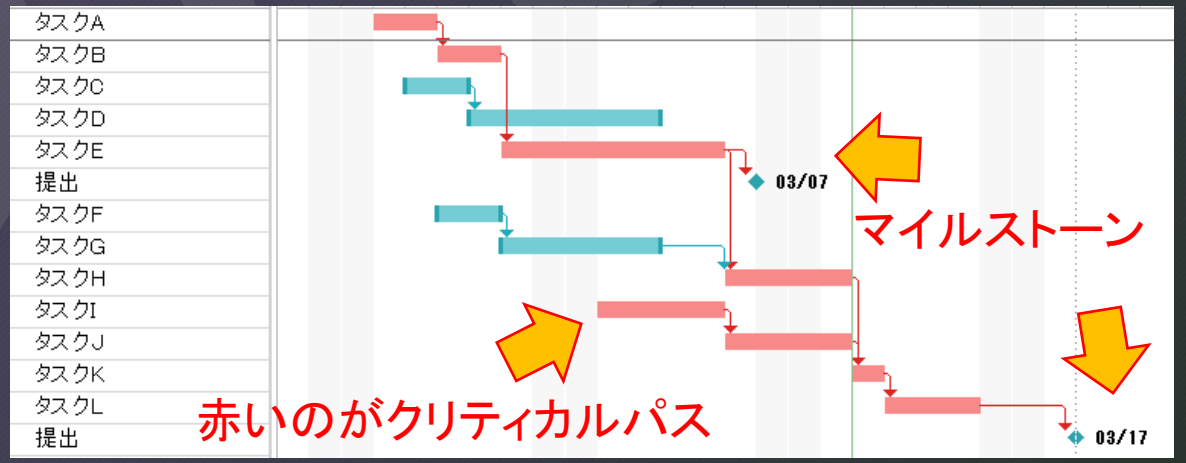
アジャイルマニフェスト

- アジャイル開発が「価値」とすること
  - プロセスやツールよりも個人との対話を、
  - 包括的なドキュメントよりも動くソフトウェアを、
  - 契約交渉よりも顧客との協調を、
  - 計画に従うことよりも変化への対応を、

  価値とする。

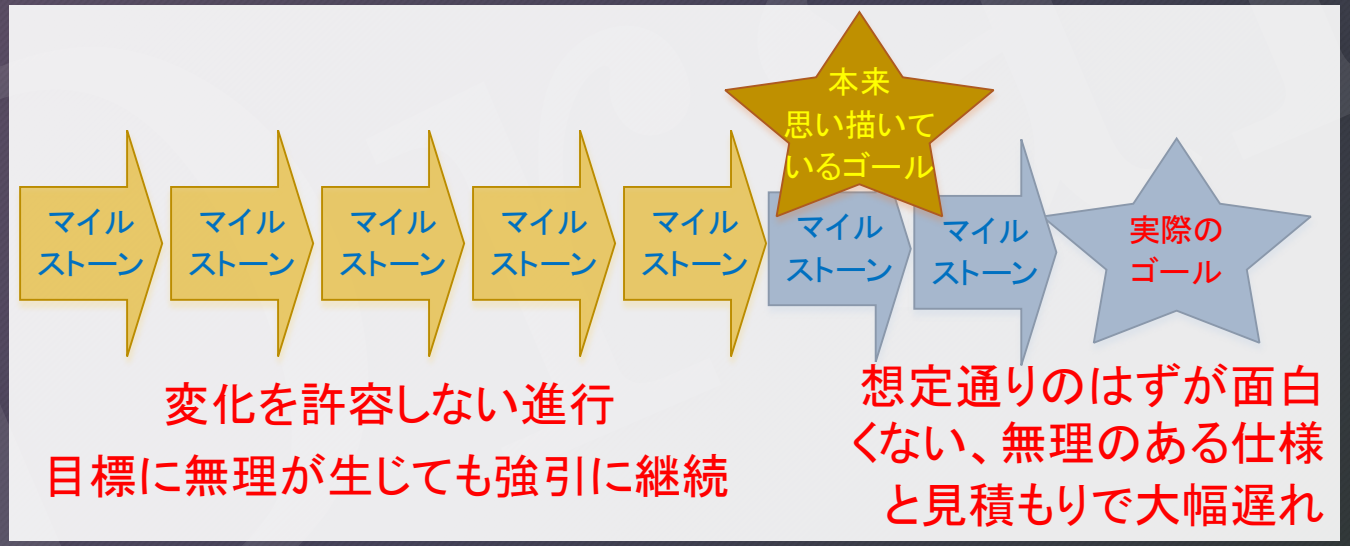
プロジェクト管理の予備知識：マイルストーンとクリティカルパス

- マイルストーン
  - プロジェクトの主要な目標
- クリティカルパス
  - プロジェクトの進行を左右する重要なタスク



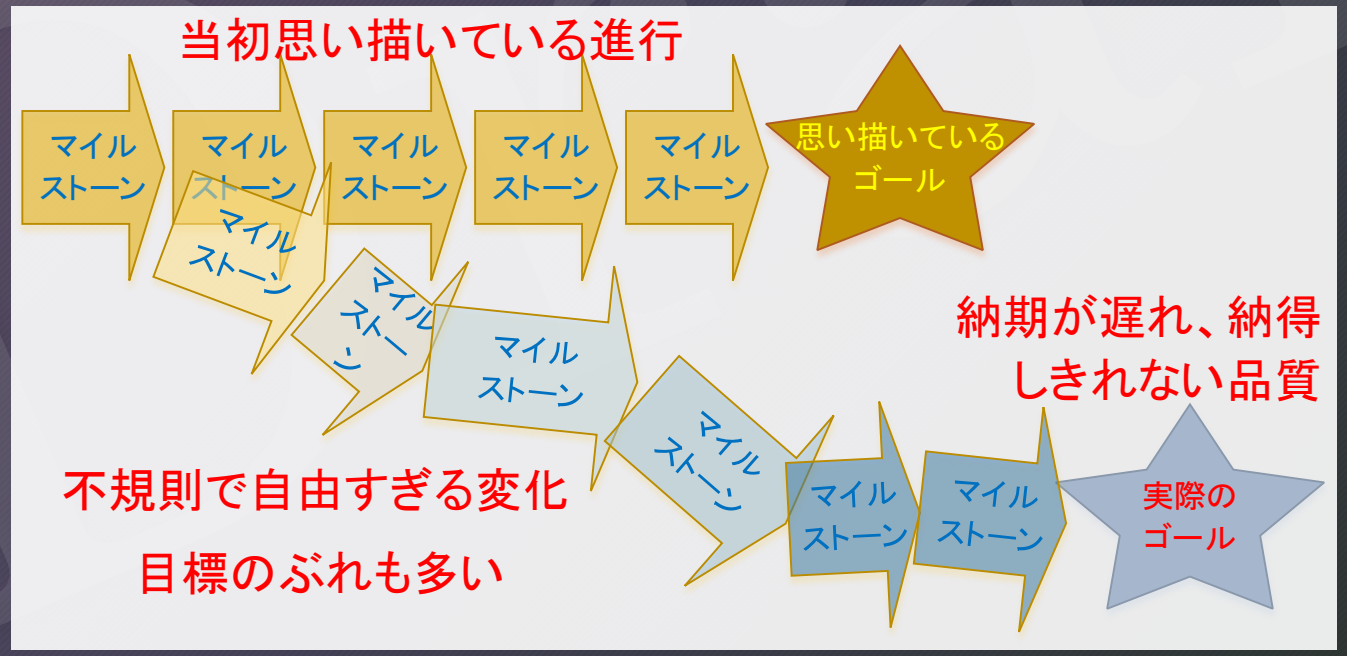
アジャイル開発の目標：変化への対応①

- 変化に対応できないと、ゲーム開発はよい結果が得られない



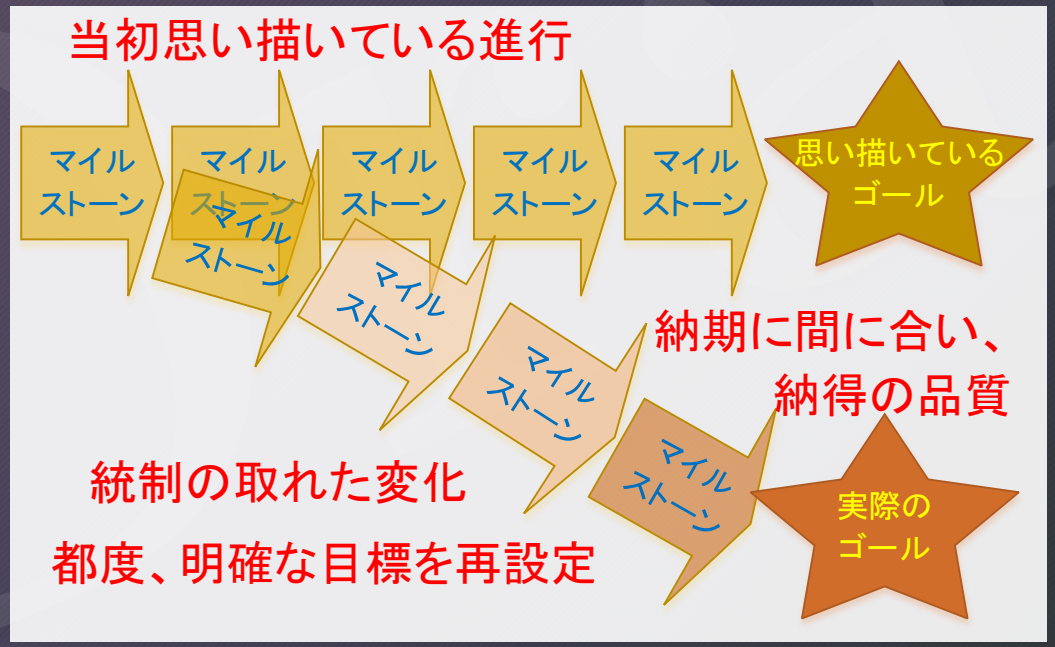
アジャイル開発の目標：変化への対応②

- ゲーム開発では、当初の想定と違うゴールに行き着くのは当然
- しかし、自由に変化に対応すればよいというわけではない



アジャイル開発の目標:
変化への対応③【目指すべき進行】

- 統制が取れ、見通しが可能な変化を



イテレーション

- イテレーション（Iteration）＝「繰り返し」
- 変化に対応する手法
- 短い開発を繰り返し行う
- イテレーションの節目にレビューし、次の目標を定め直す
- 「スクラム」はイテレーションのための方法論

スクラム

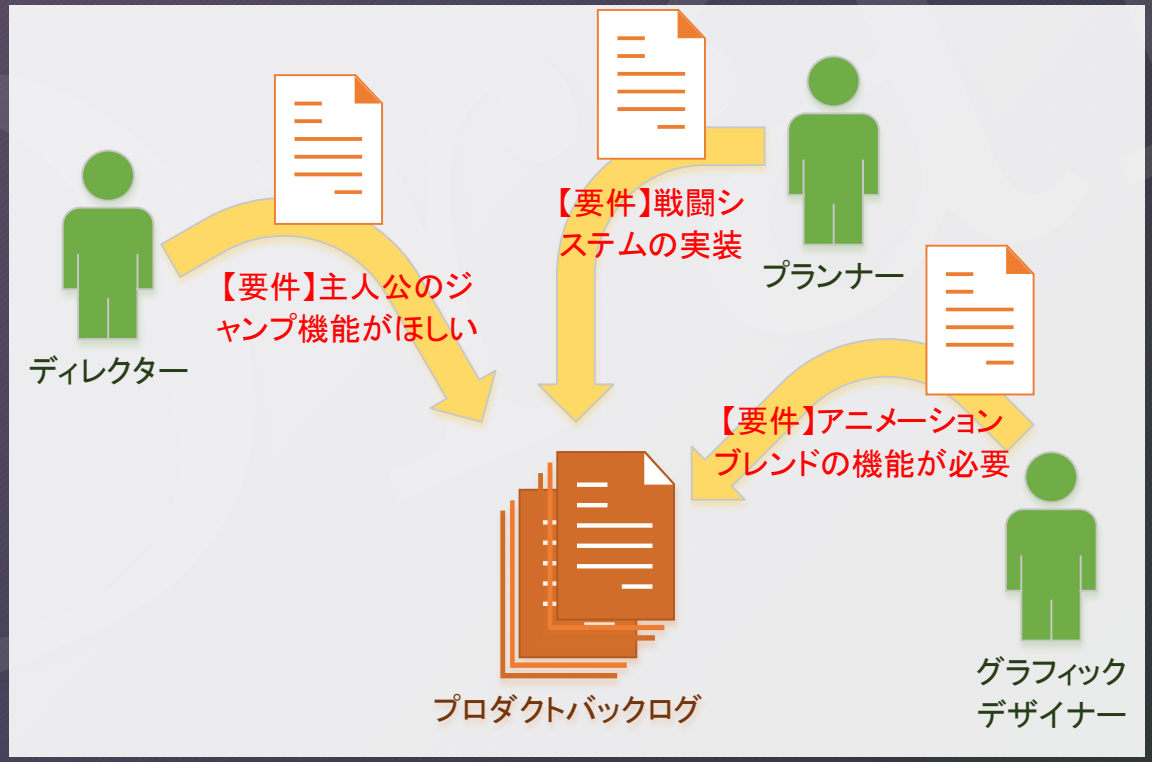
本書は、アジャイル開発方法論の一つである「スクラム」の実践を目的とする
本書では、細かい説明は抜きにして、実践すべきプラクティスのみを提示する

スクラム

- 「スクラム」とは、アジャイル開発方法論の一つ
- 透明性と柔軟性のための開発方法論
- その構成要素は「プロダクトバックログ」と「スプリント」

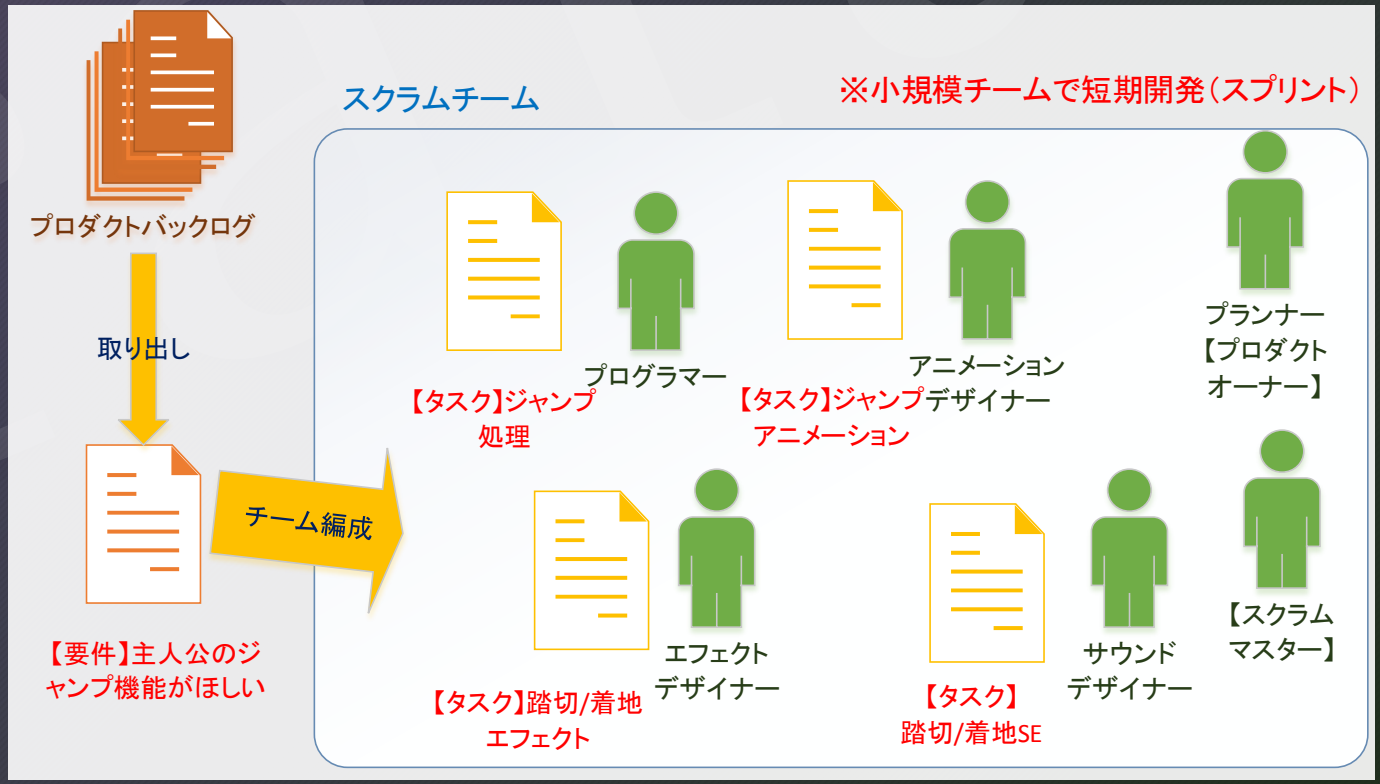
スクラムの構成要素：プロダクトバックログ

- ゲーム要件を溜め込む先が「プロダクトバックログ」



スクラムの構成要素：スプリント

- ゲーム要件に取り組むことが「スプリント」
- 小規模チームを編成して取り組む
- スプリントの期間は1～4週間程度
- スクラムチームは10人未満

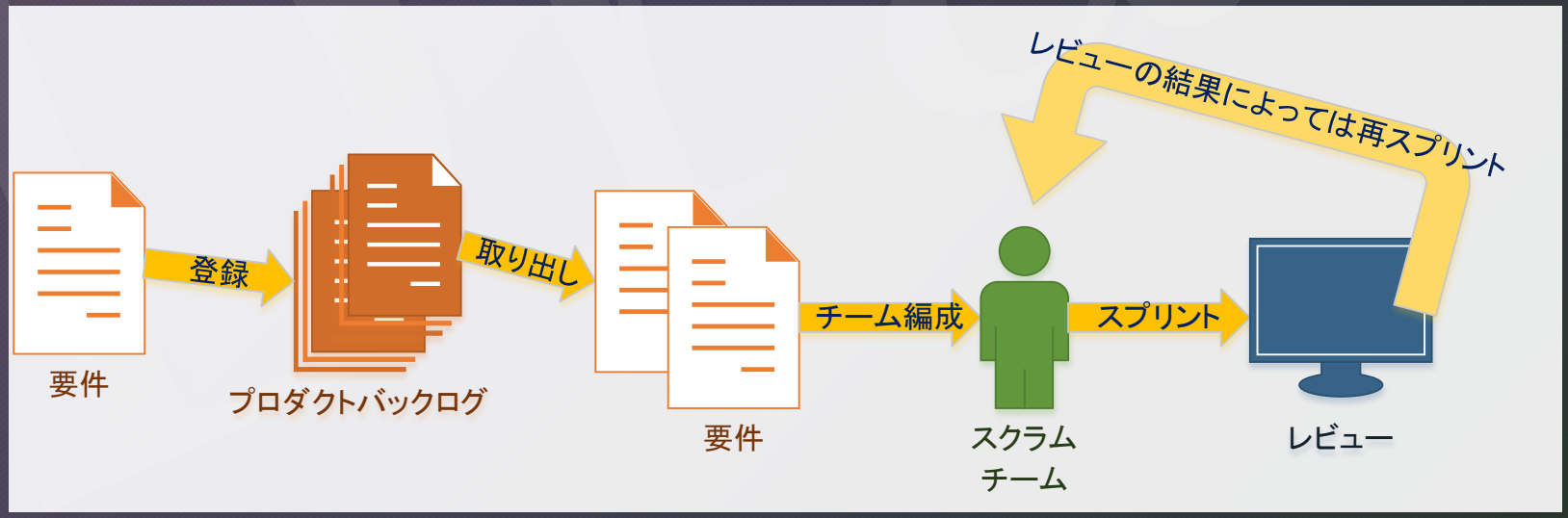


スクラムの適用対象

- コンセプトフェーズとプリプロダクションフェーズに適用
- プロダクションフェーズ、ポストプロダクションフェーズには適用しない
- ゲームの面白さの確立のために
  - プリプロダクションを柔軟に進行することが大きな目的

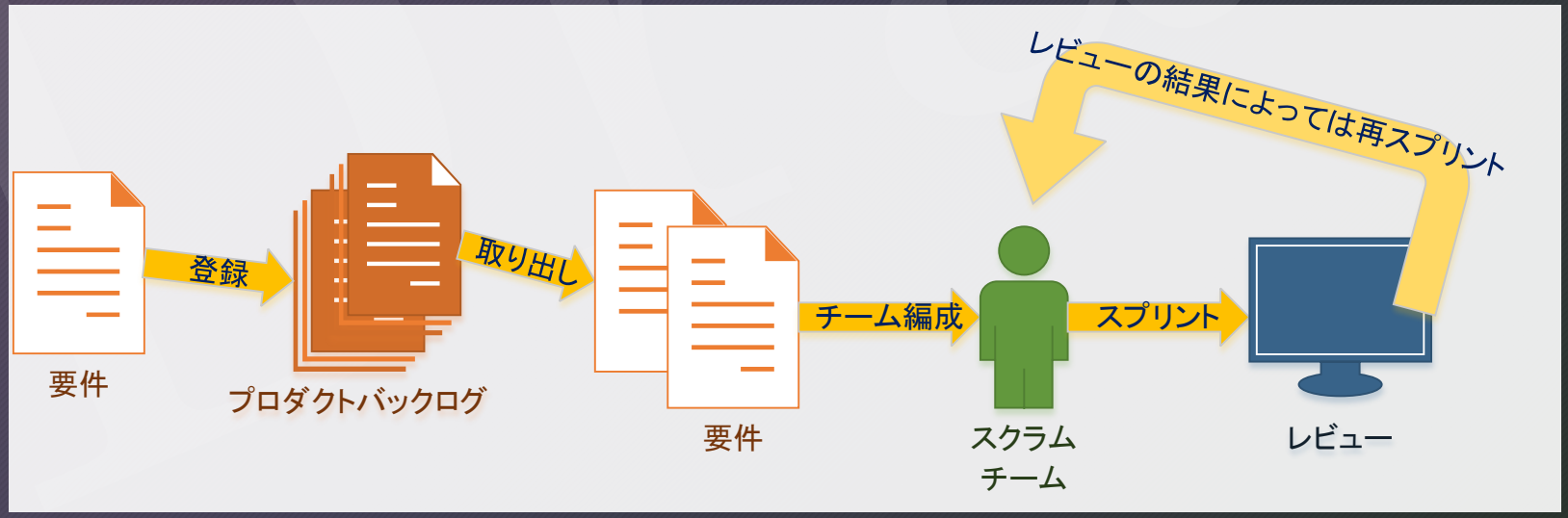
スクラムの流れ①

- ゲーム要件をプロダクトバックログに登録
- プロダクトバックログから幾つかの要件を取り出してスプリントを実施
- スプリントに参加するスクラムチームを編成



スクラムの流れ②

- タスクを割り出しスプリントの完了日を設定して作業
- 完了時にはチームスタッフ全員とディレクターでレビュー
- 必要に応じて再スプリント



ツールの活用

- Webツールによる管理を行う
  - ➢プロダクトバックログの管理
    要件登録／優先度管理／状態管理（未着手・作業中・完了・棄却）
  - ➢スプリントの管理
    スクラムチーム編成／タスク／進捗・達成度／レビュー結果
- ツール利用で手間の軽減とプロジェクトの透明化
- （できれば）協力会社の進行管理にも活用

※独自の内製ツールを想定

プロダクトバックログ

以降、スクラムのプロダクトバックログについて説明

プロダクトバックログの書き方：ユーザーストーリー

- プロダクトバックログには「ユーザーストーリー」を用いる
  - ユーザーストーリーとは、下記の形式で書かれた要件のこと

（ユーザーの役割）として、（ゴール）を達成したい。
それは、（理由）のためだ。

※書籍「アジャイルなゲーム開発」から、そのまま引用

プロダクトバックログの書き方：ユーザーストーリーの例

- 「プレイヤーとして、他のプレイヤーのボイスチャットをミューとするボタンがほしい。それは、他のオンラインプレイヤーに邪魔されるのを防ぐためだ。」
- 「アニメーターとして、ゲームをリスタートせずに直接ゲーム内のアニメーションを変更したい。それは、アニメーションの修正作業を効率化するためだ。」
- 「プレイヤーとして、自分の体力メーターを確認したい。」

※書籍「アジャイルなゲーム開発」から、そのまま引用

プロダクトバックログの作成方法：遠慮のない大きな要件

- まずはスプリントにかかる期間など度外視
- 大きな要件を遠慮なく書く
- 必要な要件、アイディアが失われないように、思いついたら形式にとらわれずに書く

プロダクトバックログの作成方法：トピックバックログ

- スプリントの際に要件を細分化
- 要件の中の小さな要件を「トピックバックログ」とする
- トピックバックログには「達成条件」も加える
- 必要があれば普段からもトピックバックログを追加
- 既存の要件の関係を組み替えてトピックバックログ化することも

※「トピックバックログ」は、本書のために設定した新造語
　由来は分散SCMの「トピックブランチ」から
※「達成条件」の設定も独自のスタイル

プロダクトバックログの作成方法：トピックバックログの例

- サンプル要件：「プランナーとして、戦闘の面白さを確立したい」
  - トピックバックログ：
    通常移動と近接攻撃の仕様を実装したい
    【達成条件】SE、メニューは現時点で不要

※現実的なスプリントのために、プロダクトバックログを細分化
※本質的に必要な要素に絞って素早くスプリントを実施

プロダクトバックログの作成方法：トピックバックログの目的

- 大きなプロダクトバックログを細分化することのほか、重要なプロダクトバックログを見失わないように階層化

プロダクトバックログの達成条件

- スプリントの完了によって達成
- 不満足な結果だった場合は再スプリント
- 良い方向につながらないと判断したら
  「失敗」要件として棄却
- 全てのトピックバックログを達成したら
  プロダクトバックログ達成

リリース計画と
プロダクトバックログの優先度

- リリース予定を設定し大きな計画を立てる
  - 例えば雑誌紹介やパブリッシャー提出などの予定日
- リリースまでに達成したいプロダクトバックログを選別
  - 複数のプロダクトバックログを束ねたものがリリース計画
- プロダクトバックログに対応優先度を設定

※スプリントの進行状況によっては、優先度の低いプロダクトバックログは、
最終的に製品から漏れる可能性もある

リリース計画によるプロジェクトの調整

- プロダクトバックログの優先度管理により、
  最悪の場合実装を諦めるものも決めておく

- しかし、そうならないためにも、時折のリリース
  計画により、全ての実装が間に合うように調整する

スプリント

以降、スクラムのスプリントについて説明

スプリントの方針①:
スプリントの実施

- スプリントの期間は1～4週間程度
- スプリントに参加するスクラムチームは10人未満程度
- スプリントの終了時は何かしらのレビューを実施
  - コンテンツ制作のための機能追加や修正の要件は
    なるべく単独で扱わず、他のゲーム要件のスプリントに
    組み込み、達成時にいっしょにレビューする

※書籍ではスプリントの期間は2～4週間としている
※書籍ではスクラムチームは6～10人、多くても12人としており、論文でも7～9人とされている
※書籍ではスプリント完了時の状態を「出荷可能な状態」と説明

スプリントの方針②:
スプリント中

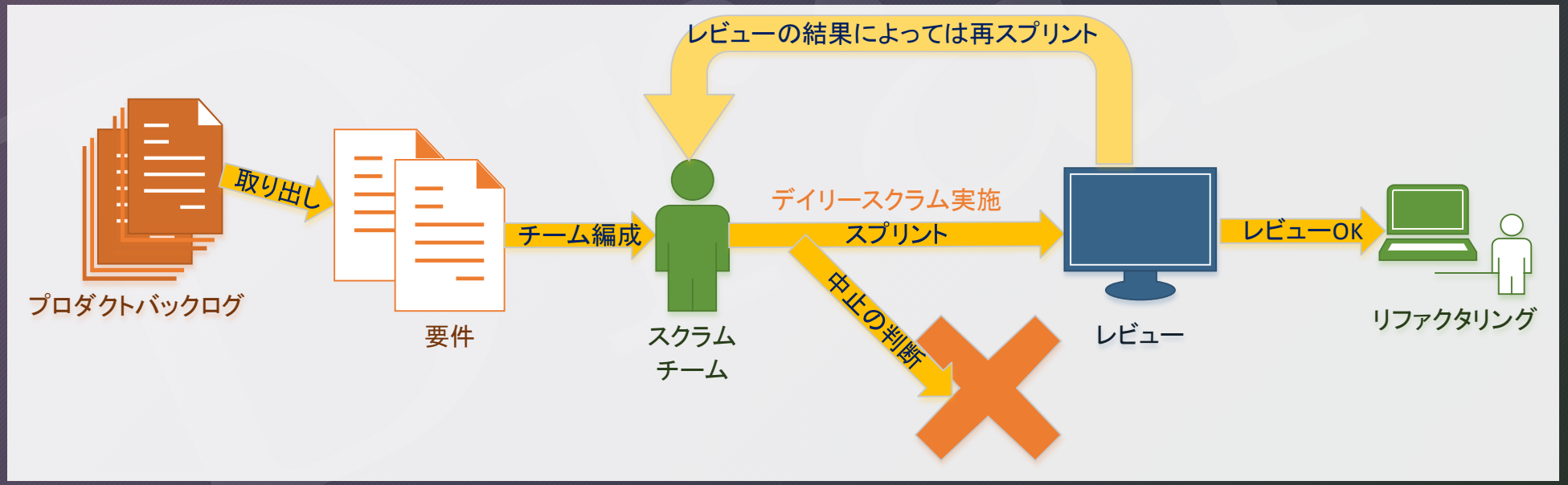
- スプリント中は、チーム外からのタスク追加・変更要求を受け付けない
  - 本当にそれが必要な状況はスプリントが失敗している時
- 途中で失敗と判明したスプリントは無理に継続せず中止

スプリントの方針③:
スプリント後

- スプリント後は必ずリファクタリングを実施
  - スプリントの目標はゲームの面白さの確立にあるため、スプリント中は暫定処理も辞さない
  - スプリント中は少しでも早くレビューを実施してゲームの面白さを確認することを重視
  - スプリント完了後は、正式に認められた実装のため、量産のためにリファクタリングを行う

スプリントの流れ

- スプリントの方針に基づきスプリントの流れを説明



スプリントの流れ①:
プロダクトバックログの選別

- プロダクトバックログの選別
  - ディレクターおよび各リーダースタッフにて、取り組むべき要件を協議
  - 複数のプロダクトバックログを1回のスプリントに含む
  - 必要に応じてプロダクトバックログを細分化
  - 10人未満のチーム、1～4週間程度の所要日数が選別の目安

スプリントの流れ②:
スクラムチームの編成／スプリント計画

- スクラムチームの編成
  - ➢各セクションのリーダーが適切なスタッフをアサイン
- チームメンバーで作業方針を協議
- 全タスクを洗い出してスプリントを計画
- スプリントの完了日を設定

※割り当てた「プロダクトバックログ」を、スプリント内でさらに「スプリントバックログ」に細分化する方法もあるが、それはせず、単純にタスクのみを扱う

スプリントの流れ③:
日々の業務

- 15分のデイリースクラムで日々の進捗確認
  - 毎朝ショートミーティングを実施
  - 簡潔な作業内容と問題の有無の報告のみ
  - 問題の詳細は別途必要最小限のスタッフでミーティング
  - チームリーダーが進行する

※書籍ではデイリースクラムの進行担当は「スクラムマスター」
※スクラムマスターについては後述

スプリントの流れ④:
スプリントの完了

- スプリントが完了したらレビューを実施
  - 参加メンバーはチームメンバー全員とディレクター、その他利害関係者
- レビューの結果、変更・修正が発生したら再スプリント
  - 改めてチームメンバーの調整、

    数日～1週間程度の再スケジューリングを行う

スプリントの流れ⑤:
スプリント後

- スプリント後は期間を設定してリファクタリングを実施
- チームメンバーも必要最小限に
- リファクタリングを終えて初めて本当のスプリント完了

スプリントの流れ⑥:
スプリントの中止

- スプリント中にうまくいかいと判断したら早めに中止
- 判断のための中間レビューも必要に応じて実施
  - ディレクター判断を求める場合など

スプリント期間の見積もり

- バッファの設定
  - スプリントは十分見積もり可能な小規模な開発だが、それでも全体のバッファ（予備日）を2～3日程度設ける
- 再スプリント・リファクタリングの期間はその時見積もる
  - ただし、その後予定していたスプリントに影響が出ることは十分に考慮する
  - リファクタリングの完了がバッファ内に収まることを想定してあらかじめ見積もるのもよい

スクラムチーム

以降、スクラムのスクラムチームについて説明

スクラムチームの編成

- 役割横断型チーム
- チームリーダー
- プロダクトオーナー
- スクラムマスター

※「チームリーダー」は本書が提案する規定

スクラムチームの編成：役割横断型チーム

- 各セクションからスプリントに参加するメンバーを集めそれぞれに役割を設定する
  - グラフィックプログラマー、AIプログラマー、アニメーションデザイナー、レベルプランナーなど

スクラムチームの編成：
チームリーダー

- チームメンバーから選出
- プロダクトオーナーとともに、チームの開発進行に責任を持つ
- なるべく多くのスタッフに責任ある立場を経験させることも狙いの一つ
- デイリースクラムの進行を担当

※書籍ではデイリースクラムの進行担当はスクラムマスター

スクラムチームの編成：プロダクトオーナー

- チームメンバーにゲームのビジョンを伝え、タスクの優先度を設定する役割
- 主にはディレクターやプランナーが努める
- スプリント中の変更や調整を判断する
  - スプリント中は、チームの外部からの変更や追加を受け付けない
- 複数のチームに渡って管理
  - チームのメンバーとして直接開発を行うものではない

スクラムチームの編成：スクラムマスター

- 「スクラム」を成功させるための指導担当
- 複数のチーム、複数のプロジェクトに渡って担当する
- デイリースクラムに参加し、スクラムの進め方が適正かをチェックし、指導する
- 進捗を確認し、チームが度重なる変更を受け入れてスプリントが間延びするような事態を防ぐ
- 外注管理として外注先にスクラムの実践を促す場合、一時的にスクラムマスターとして赴任する

複数のスプリントに参加

- スタッフは複数のチームに同時に所属し、複数のスプリントに同時に参加する
  - 一つのスプリントの最初から最後までフルにタスクがつまっているスタッフはそう多くない
- 自身の全スプリントの全タスクを優先度順に並べて自己管理する
  - 内製Webツールが自己管理をサポートする
  - タスクの優先度と作業日数を割り振って作業日を自動計算できる
  - その結果、目標達成に影響のあるスプリントがあれば警告

タイムボックス化

- 時間に制限を設けることを「タイムボックス化」という
- スクラムはタイムボックスの遵守を原則とする
- チームスタッフは相互に時間に厳しく活動する
- スプリントは1～4週間のタイムボックス
- デイリースクラムは15分のタイムボックス
- ミーティングも基本的に1時間などでタイムボックス化する
- ミーティングは進行役と別にタイムキーパーを設定してリミットの通知を担う
- リミットに達したら、たとえ途中であっても打ち切る
  - スプリントはそういうわけにもいかないので適正なバッファ管理を行う
- 必要があればミーティングを再設定

スプリントとスクラムチームの要約

繰り返しと言い直しになるが、
スプリントとスクラムチームに関する要約を説明

スプリントとスクラムチームの要約①

- スプリントのイメージは旧来のタスク管理
- 十分全体管理できる程度の期間と規模
- タスクの優先度を決め進行方針を調整するのが
  プロダクトオーナー
- チームの作業進行に責任を持つのが
  チームリーダー
- スプリントの進行を監督しサポートするのが
  スクラムマスター

スプリントとスクラムチームの要約②

- チーム内では状況に合わせて柔軟にタスクを調整
- しかし目標がぶれてはいけない
  - 大きく目標を変える必要が生じたらスプリントをやり直す
- スプリントが終わるまでチーム外からの干渉は受け付けない

スプリントとスクラムチームの要約③

- プロダクトオーナーが
  「スプリントの結果が失敗に終わる」と判断したら
  その時点でディレクターと相談してスプリントを中止
- チームリーダーが
  「想定通りの進行ができない」と判断したら
  プロダクトオーナーと相談してタスク調整
  もしくはスプリントを中止

スプリントとスクラムチームの要約④

- スプリントの結果によっては
  再スプリントを計画して
  最善の結果を何度でも求める
- スプリントの終わりには
  必ずリファクタリング

量産パイプライン管理システム

プロダクションフェーズではスクラムの手間さえ省き、
ゲームコンテンツの量産に特化した管理を行う

プロダクションフェーズの管理

- プロダクションフェーズにはスクラムを用いない
- 内製ツールの量産パイプライン管理システム
  のみを使用して管理する

量産パイプライン管理システム

- 内製Webツールによる管理を行う
- 量産コンテンツに対して各セクションが作業をおえたかどうかを管理するだけの単純なシステム
  - 例えば、戦闘キャラならモデル、テクスチャ、モーション、SE、エフェクト、戦闘スクリプト、AIの実装が済んだかどうかをチェックする

量産パイプライン管理システム：進捗管理

- 量産コンテンツの実装率を表示する
- 日々の実績を記録し、量産コンテンツの実装ペースをグラフ表示し、終了予測する

量産パイプライン管理システム：パイプライン管理

- 工程の前後関係をチェックする仕組みをサポート
  - 例えば、モーションが実装されたらSE,エフェクトの作成が可能
  - 修正の発生も記録し、例えば、モーションの修正の影響を受けてSE,エフェクトの確認と調整が必要なことが促される
- 作業が可能なコンテンツの絞り込み表示をサポート
  - 自身が作業可能なコンテンツ、作業しなければならないコンテンツを素早くリストアップ

BTS（Bug Tracking System）

ポストプロダクションフェーズは旧来からの手法でBTSで管理

スクラムの併用を考えず、BTSのみに集中する

ポストプロダクションフェーズの管理

- ポストプロダクションフェーズはスクラムを用いない
- BTSのみに集中して管理

以上