1.2　アジャイルの各種プラクティス

◇ リリース計画

どのストーリーをどのイテレーションで実現するか、顧客が主体となって計画を立てる。

◇ 短期リリース

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する。

◇ 共通の用語

用語集を作成し、チーム全員の使用する用語とその概念の不一致を解消する。

◇ シンプル設計

今必要とされている機能だけのシンプルな設計・実装を行う。将来的に使うかもしれない、という機能の設計は行わない。

◇ テスト駆動開発（TDD）

実装よりも先にテストを作成する。実装は、そのテストをパスするようにシンプルに作成する。テストはテストコードとしてプログラミングし自動化することで、変更コストを抑制することができる。

◇ リファクタリング

完成済みのコードでも、随時、改善処置を行う。この際、外部から見た動作は変更させずに、内部構造をより見通し良く優れたものになるようにする。

◇ ペアプログラミング

プログラミングを、二人一組で行う。一人がコードを書き、もう一人はそれをチェックしながら、仕様を確認したり全体構造を考えたりするなど、ナビゲートを行う。二人は、この役割を定期的に交代しながら仕事を進める。

◇ ソースコードの共同所有

チームのすべてのメンバが、いつでもどのコードでも変更することができる。

◇ 継続的インテグレーション

単体テストをパスするコードが追加されるたび、全てのコードがテストされ、全体として正常な状態を維持していく。

◇ 最適なペース

知的作業には、週40時間の労働時間が最適である。心身ともに健全な状態を保つ必要があり、それ以上の労働は適切ではない。

◇ 全員同席（オンサイト顧客）

開発者だけでなく、顧客、マネージャも含めプロジェクトに関係するメンバーが全員、一つのチームとして一ヶ所に集まり作業をする。

◇ コーディング標準

コーディング標準をチームで定義し、それに従って一貫性のあるコードにする。

◇ 受け入れテスト

イテレーションごとに顧客の立場からテストを行い、ストーリーが実現できているか、望むシステムになっているか確認する。

プロダクトオーナー

スクラムマスター

開発チーム（6±3人）

ITEMS
1
2
3

プロダクトバックログ

1 2
3 4

プランニング
ポーカー

スプリント
バックログ

WORKS

タスクボード

バーンダウン
チャート

デイリースクラム
（毎朝）

スプリント

（1〜4週間）

動く
ソフトウェア

スプリントデモ

振り返り

スプリント
プランニング

アジャイルソフトウェア開発に有効な様々なプラクティス

- スタンドアップミーティング
- タスクかんばん（タスクボード）
- バーンダウンチャート
- ベロシティ計測
- プランニングポーカー
- インセプションデッキ
- ユーザーストーリーマッピング
- ふりかえり

アジャイルは改善のプロセス

失敗から学び成長するために、定期的にふりかえる

◇ふりかえりの代表的な手法

・KPT（Keep, Problem, Try）

・YWT（やったこと, わかったこと, 次にやること）

など

1.2.5　KPT

Keep

良かったこと
続けたいこと

Problem

うまくいかなかったこと
問題点

Try

問題に対する改善策
次に試したいこと