

進化したアジャイル手法で開発した受発注 業務のクラウド型次世代EDIサービスと 電力会社向けシステムで「コスト負担の 削減」「業務効率アップ」に貢献する事例紹介

2019年03月22日

株式会社グローバルワイズ



- 1) アジャイルとは？
アジャイル開発の簡単な説明
- 2) 事例 1 EDI 受発注システム
「基本系のスクラムで開発」
- 3) 事例 2 電力関連会社システム
「ウォーターフォールとアジャイルの
ハイブリッドで開発できるのかの挑戦」

1) アジャイル開発とは？

1) アジャイル開発とは？

アジャイルの意味

- ・ [名・形動] 俊敏であるさま。機敏な。敏捷な。

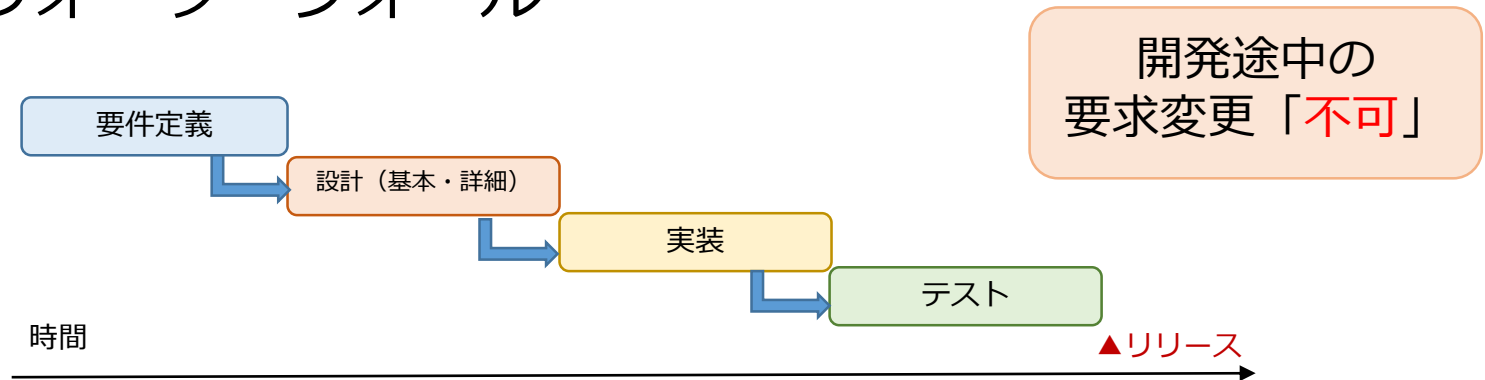
アジャイル開発とは？

- ・ ソフトウェア開発手法の一つ
- ・ 要求仕様の変更などに対して、機敏かつ柔軟に対応するためのソフトウェア開発手法

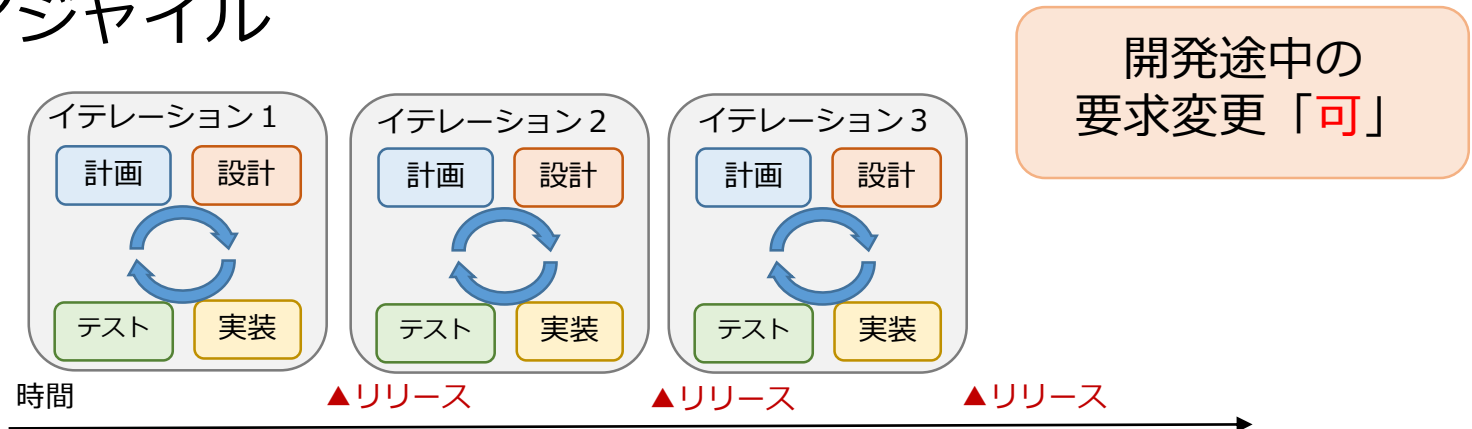
1) アジャイル開発とは？

従来型の開発との違いは？

- ・ウォーターフォール



- ・アジャイル



1) アジャイル開発とは？

何が適している？

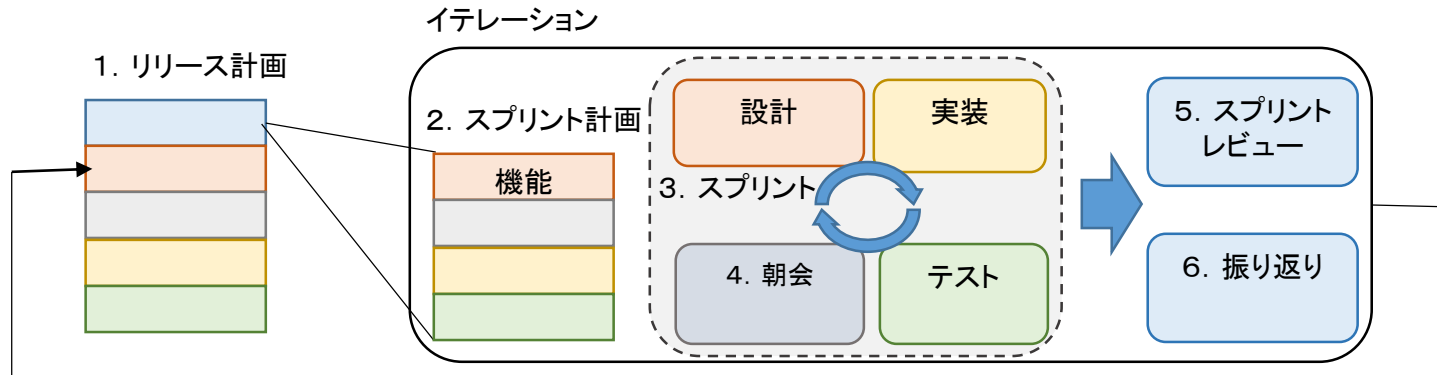
- 要求のゴールは見えているがぼんやりしている。現時点で要求を全てを決められない。
- 時間の変化で要求の変更の可能性がある。
- 顧客が情報システム担当者ではなくエンドユーザー仕様書を見ただけでは、判断、決定が出来ない・・・

要求の詳細が決まっている場合は、
ウォーターフォールの方が効率が良い。

1) アジャイル開発とは？

開発の進め方

1. リリース計画を立てる。
機能を優先順位別にどれくらいの期間で実装するかを検討する。
2. スプリント計画を立てる。
スプリント内で実装する機能を検討する。
3. スプリント
機能の設計・実装を行う。
4. 朝会の実施
進捗報告、問題点の共有を行う。
5. スプリントレビュー
イテレーションでの成果物のレビューを行う。
6. 振り返り
スプリントの問題点、次回の挑戦等の議題を討議する。



1) アジャイル開発とは？

開発を進める上での登場人物

- ・ プロダクトオーナー

プロダクトに対しての最終決定権と責任を持つ。（主に顧客）
リリース計画の優先順位を決定する。

- ・ スクラムマスター

プロダクトオーナーとチームの仲介役。
チームの効率を最適化する役目。

- ・ 開発チーム

設計者、実装者を区別することなく開発に関わる人。

以上を踏まえ、事例へ・・・

2) 事例 1 EDI 受発注システム

2) 事例 1 EDI 受発注システム

顧客（プロダクトオーナー）

- ・ 部品メーカー + コンサルタント

要件

- ・ 仕入先への部品の受発注をEDIで行いたい。
（仕入先での看板発行で発注リードタイムを短縮したい）
- ・ 部品の受発注の実績をEDIの情報から取得したい。
- ・ 自社でサーバーを持つのではなく、クラウドを活用したい。
- ・ 最終的には、他の部品メーカーに利用してもらおうことも想定し、大規模な接続にも耐えうる機構にしてほしい。

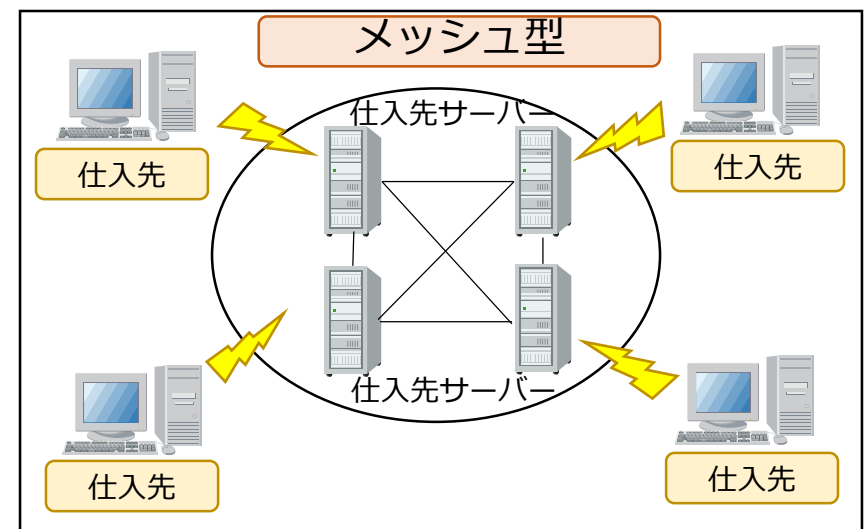
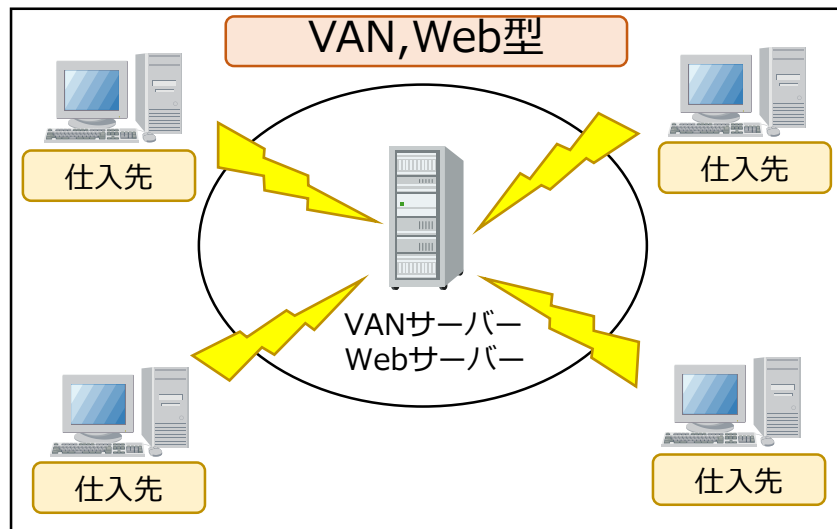
⇒ 後のトヨタWG共通EDIのベースとなる仕組み

2) 事例1 EDI受発注システム

EDI受発注システムの設計方針(1)

クラウドが前提条件。

- ・ Web型、VAN型の集中管理ではなく、ピアツーピア型のメッシュ型EDIとして設計。
- ・ 各企業が独立したクラウドのサーバーを使用することにより、他企業のデータが混在しない構造とする。



2) 事例 1 EDI 受発注システム

EDI 受発注システムの設計方針 (2)

受注側 (仕入先)

- ・仕入先でかんばんを発行する機能。
- ・画面から簡単なオペレーションで出荷情報を送信する画面が必要。
(単なるデータ交換の仕組みではなく、操作画面が必要)

発注側

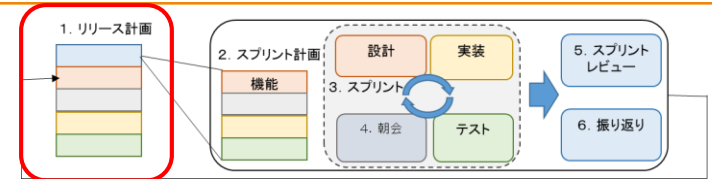
- ・得意先内示から部品所要量展開を行う機能が必要。
(現状、EXCELで手間をかけて発注データを作成している)
- ・部品所要量展開の結果を簡易的な操作で仕入先に発注する画面が必要。
- ・受発注の結果を自社のホストで取り込む機構。

⇒ 仕入先にアプリケーションを配布してしまうとバージョンアップが大変なので、Webアプリケーションとして作成。
自社のホスト側からEDIサーバーに実績をPULLで取得するアプリを作成。

⇒ アジャイル開発へ

2) 事例1 EDI受発注システム

リリース計画の作成



プロダクトオーナー（顧客）と打ち合わせを行い、必要な機能の優先順位と目標期間を設定する。

チームの実力から約2週間程で実装可能な機能に切り分ける。

■ リリース計画

1. 通信プロトコルの設計及び実装

2. 簡易メッセージによるサーバー間でのデータの送受信

3. 発注画面、受注画面の実装 + 最小限データ通信

...

7. 仕入先ごとの通信先変更設定

...

10. 仕入先でのかんばん印刷機能作成

...

2) 事例 1 EDI 受発注システム

アジャイル開発を行う為の準備 (1)

プロジェクトの規模から 4 人構成とした。

チームメンバーの選定

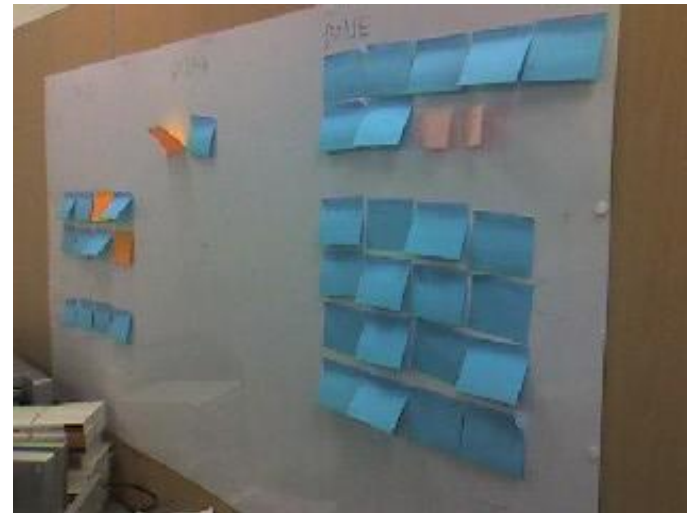
- ・ 設計 + コーディングが両方できる人を 1 人。
- ・ コーディングに長けた人を 1 人。
- ・ 後は、向上心を持って新しいことに意欲を感じられる人。
経験は薄くても可。

2) 事例1 EDI受発注システム

アジャイル開発を行う為の準備 (2)

物理的な準備 (1)

- ・いつでもチームが自由に使えるホワイトボード
- ・大量の付箋紙とマジックペン
チーム内のタスクを付箋紙で管理する為。



2) 事例1 EDI受発注システム

アジャイル開発を行う為の準備 (3)

物理的な準備 (2)

- ・ チームが開発に専念できる~~監禁部屋~~プロジェクトルーム



チーム内で頻繁にコミュニケーションをとる必要が為、
専用の部屋があると便利

2) 事例1 EDI受発注システム

イテレーションのスケジュール

本開発では、イテレーションの期間を2週間で設定した。

1週目

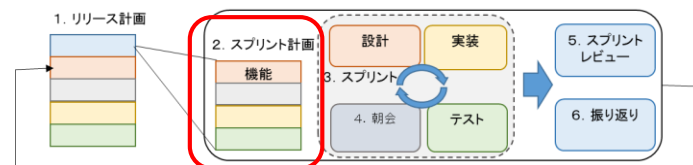
	月	火	水	木	金	土	日
AM	●スプリント 計画	●朝会	●朝会	●朝会	●朝会		
PM							

2週目

	月	火	水	木	金	土	日
AM	●朝会	●朝会	●朝会	●朝会			
PM					●スプリント レビュー ●振り返り		

2) 事例1 EDI受発注システム

スプリント計画 (1)



- ・リリース計画の1コマを小さい機能に振り分け、付箋紙に記載
スクラムマスターが決めるのではなく、チームで決める。

ログイン機能

- ・機能の見積を日単位で行う (1日、2日、3日・・・)

見積ゲーム (みんなで一斉に工数の札を挙げて多い工数を記載)

ログイン機能 工数:2日

- ・誰が機能を作成するかをチーム内で決める。

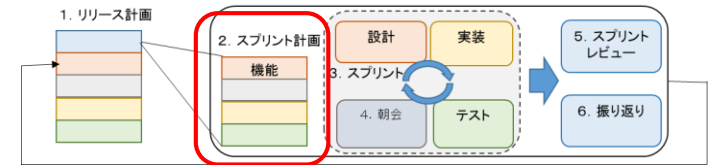
誰が作成した方が良いかを、作業効率、他の機能の兼ね合いで
チームで話し合いにより決定する。

チームメンバ個人の自主性を尊重する。
個人に押し付けたタスクとしない。

ログイン機能 工数:2日
作成者 ○○君

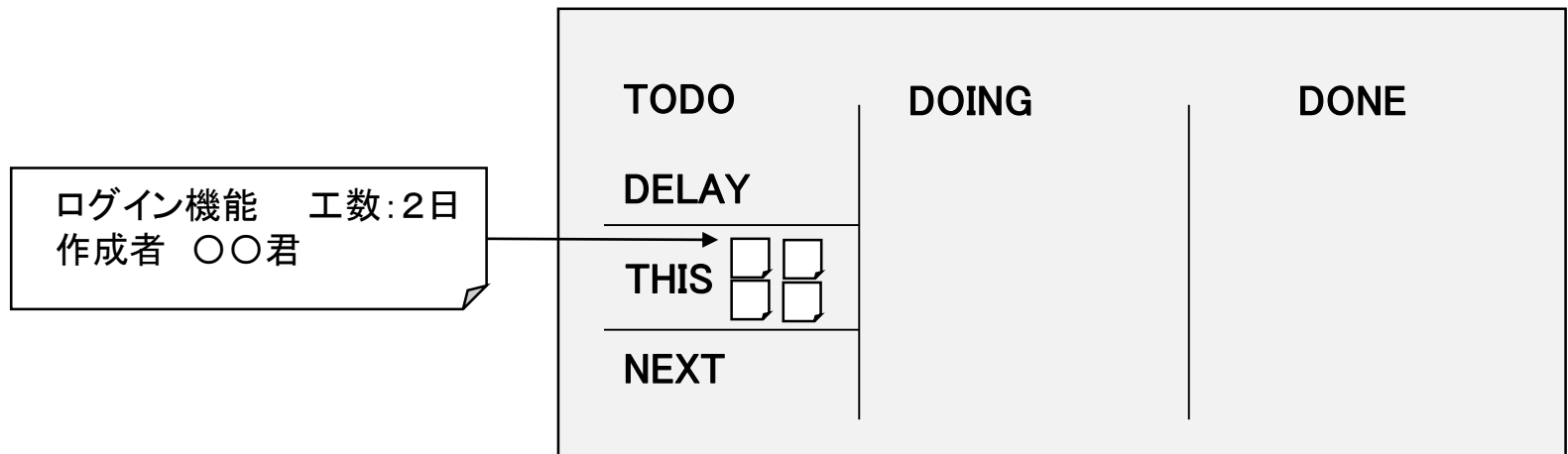
2) 事例1 EDI受発注システム

スプリント計画 (2)



決まった機能をタスクかんばんに掲示する

タスクかんばん



TODO DELAY

TODO THIS

TODO NEXT

DOING

DONE

・・・ 前イテレーションで間に合わなかったタスク

・・・ 本イテレーションのタスク

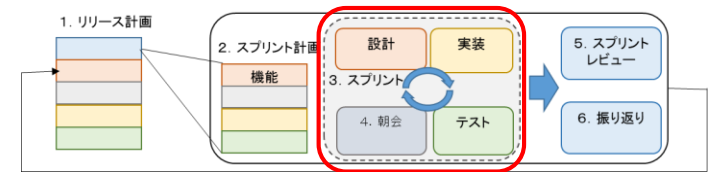
・・・ 次回イテレーションで行うべきタスク

・・・ 現在着手しているタスク

・・・ 完了したタスク

2) 事例1 EDI受発注システム

スプリント (1)



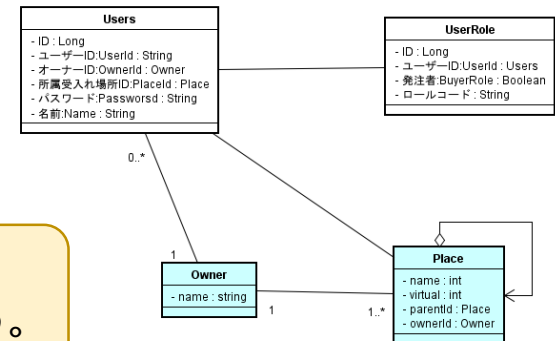
設計

基本設計

- ・ 簡易的な画面設計、データフローを記述する。

詳細設計

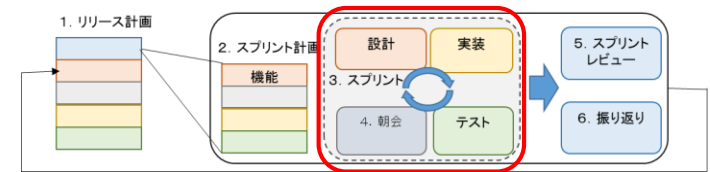
- ・ 骨格となるクラス図、シーケンス図のみを作成する。



チームが意思統一を図る為にも最低限の設計は必要。
設計が何もないと個々が勝手に自分の解釈で実装してしまう。

2) 事例1 EDI受発注システム

スプリント (2)



実装

- ・実装者同士で干渉しそうな機能に関しては、お互いどのように実装すべきかを検討してから実装する。

テスト

- ・JUNIT等自動テストを活用する。
- ・カバレッジは常に80%以上をキープすること。
- ・全件テストが通るまで絶対にコミットしてはいけない。

テストが通らない状態でコミットしたら100円罰金！！
(次回の飲み会の貯金になります・・・)

リファクタリング

- ・プログラムの外部から見た動作を変えずにソースコードの内部構造を整理する。(プログラムの改修を容易な状態に保つ)

アジャイルは反復型開発である為、次期イテレーションで改修した場合の、他のソースへの影響を即座に知る必要がある。その為、テストが重要視される。

2) 事例1 EDI受発注システム

朝会



タスクかんばんを通じて進捗の確認を行う。

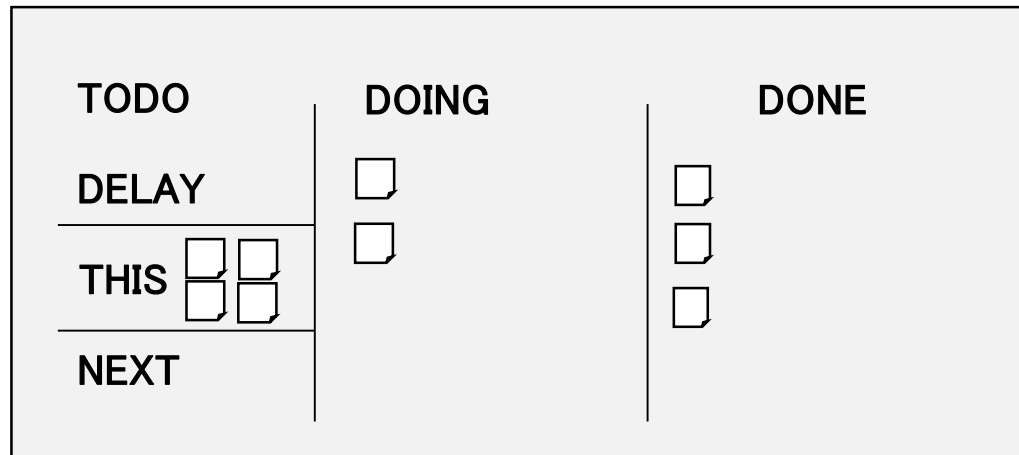
チームメンバー個々に以下の内容を確認する。

- ・ 昨日行ったタスク
- ・ 本日举行予定のタスク
- ・ 実装を行うことでの問題点、障害

問題、障害がある場合は、解決する方法を模索する。

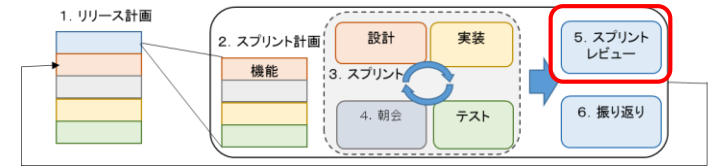
タスクかんばん

朝会は簡潔に行い、問題点、障害を取り去ることに尽力する。
時間効率を図る為、個々の諸問題に関しては、朝会以外で個別に行う。



2) 事例1 EDI受発注システム

スプリントレビュー



スプリント終了時にリリース可能な状態にし、
プロダクトオーナーに対して完成した機能のデモを行う。

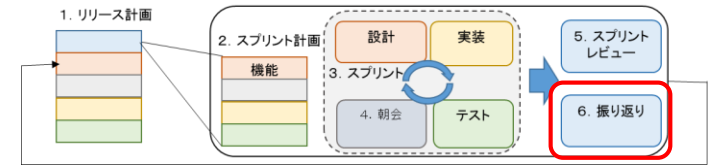
プロダクトオーナーは、機能に対しての評価を行い、
改修すべき箇所がある場合には指摘を行う。

フィードバック内容とリリース計画を考慮し、
次期イテレーションの内容を決定する。

プロダクトオーナーの指摘、要件の変更は、
スプリントレビューを行うことで反映され、
次期イテレーション以降で改善されて行く。

2) 事例 1 EDI 受発注システム

振り返り



スプリント終了時にKPTで振り返りを行う。
振り返りの内容を次期イテレーションに反映する。

KEEP <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	TRY <input type="checkbox"/> <input type="checkbox"/>
PROBLEM <input type="checkbox"/>	

本プロジェクトでは、チームメンバが真面目なのか日本の文化なのか
わからないが、「問題」の意見が出るが、「良かったこと」の意見が
少なかったです・・・
イテレーション中に何が起こったかがいつでも見れる状態が大事。

2) 事例 1 EDI 受発注システム

アジャイル開発を行ったチームメンバーの声

- ・テストをしっかり描くので、後戻りを恐れず、開発が出来る。
- ・仕様周りは、下りてきた物ではなく自発的に考えられる。
- ・余分なドキュメントを書かずにすむので直ぐにコーディングが出来る。
- ・顧客の意図を汲みながら作成できるので意欲が沸く。

2) 事例1 EDI受発注システム

アジャイル開発を行った効果

- ・プログラミングスキルの向上
反復開発である為、他人のソースを改修することが多い。
その為、スキルの高いプログラマのコードを参考にする機会が多く、
中級以下のプログラマのコーディング能力が向上した。
- ・設計能力の向上
顧客と直接接する機会が増えたため、顧客の意図を汲んで、
設計できるようになった。
- ・開発意欲の向上
設計から与えられた仕様書を元にプログラムを書くのではなく、
意見を出し合った仕様でプログラムを書くので、プロジェクトに
前向きに参加するようになった。

チームメンバー個々が上流工程のスキルを身に付けることが出来た。

3) 事例 2 電力関連会社システム

2) 事例2 電力関連会社システム

顧客（プロダクトオーナー）

- ・ 電力関連会社の主管部担当者
（情報システム担当者ではない）

要件

- ・ 大雑把にやりたいことは決まっている。
- ・ どのように実現するかは、ベンダー任せ。
- ・ 外部連携するシステムがあるが、並列同時開発でインターフェースが定まっていない。
- ・ 仕様書は納品物としてしっかり書かないといけない。

2) 事例2 電力関連会社システム

どのように開発するかを社内で検討

- ・顧客が情報システムではなく、エンドユーザーである為、設計書で合意しても、製造物に対して、意思の乖離が発生してしまうのではないかと？
⇒ 顧客満足度を得るにはどうしたらよいか？



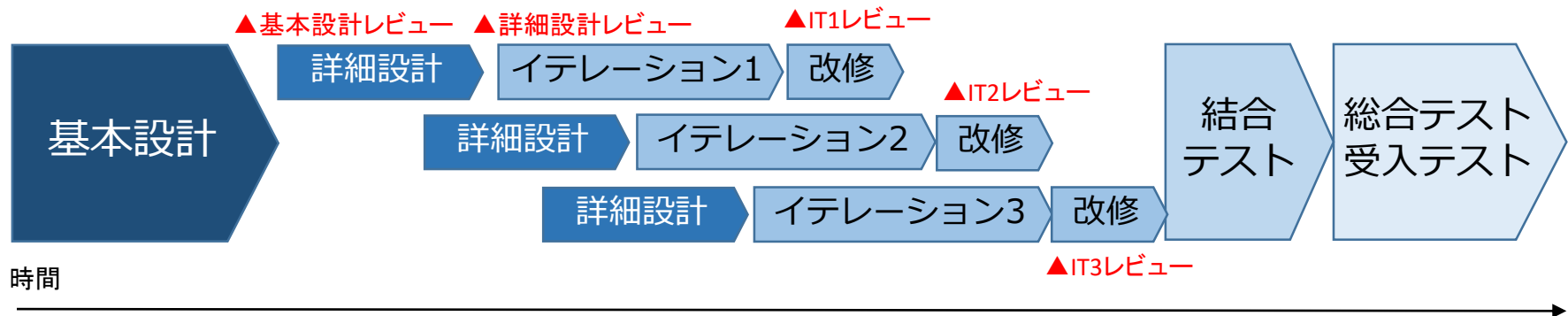
- ・意思の乖離を極力減らすために、できた製造物から順に顧客にレビューしてもらい、改善することで顧客満足度が得られるのではないかと？



- ・完全にアジャイルでは開発できそうもないので、ウォーターフォールとアジャイルのハイブリッドで開発してみよう！！

2) 事例2 電力関連会社システム

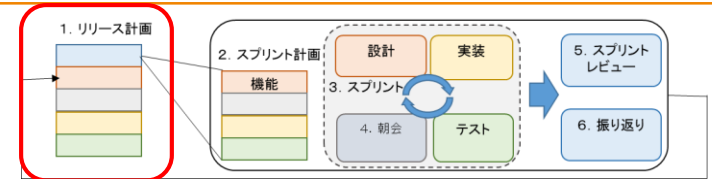
顧客に業務実施計画を提案



- 基本設計にて、大枠の機能の承認を得る。
- イテレーション前に詳細設計の承認を得る。
- 開発を3つの機能をグループ分けし、機能グループ毎にレビューを行い、レビュー後、意思と乖離する箇所を改修・改善する。

2) 事例 1 EDI 受発注システム

リリース計画の作成



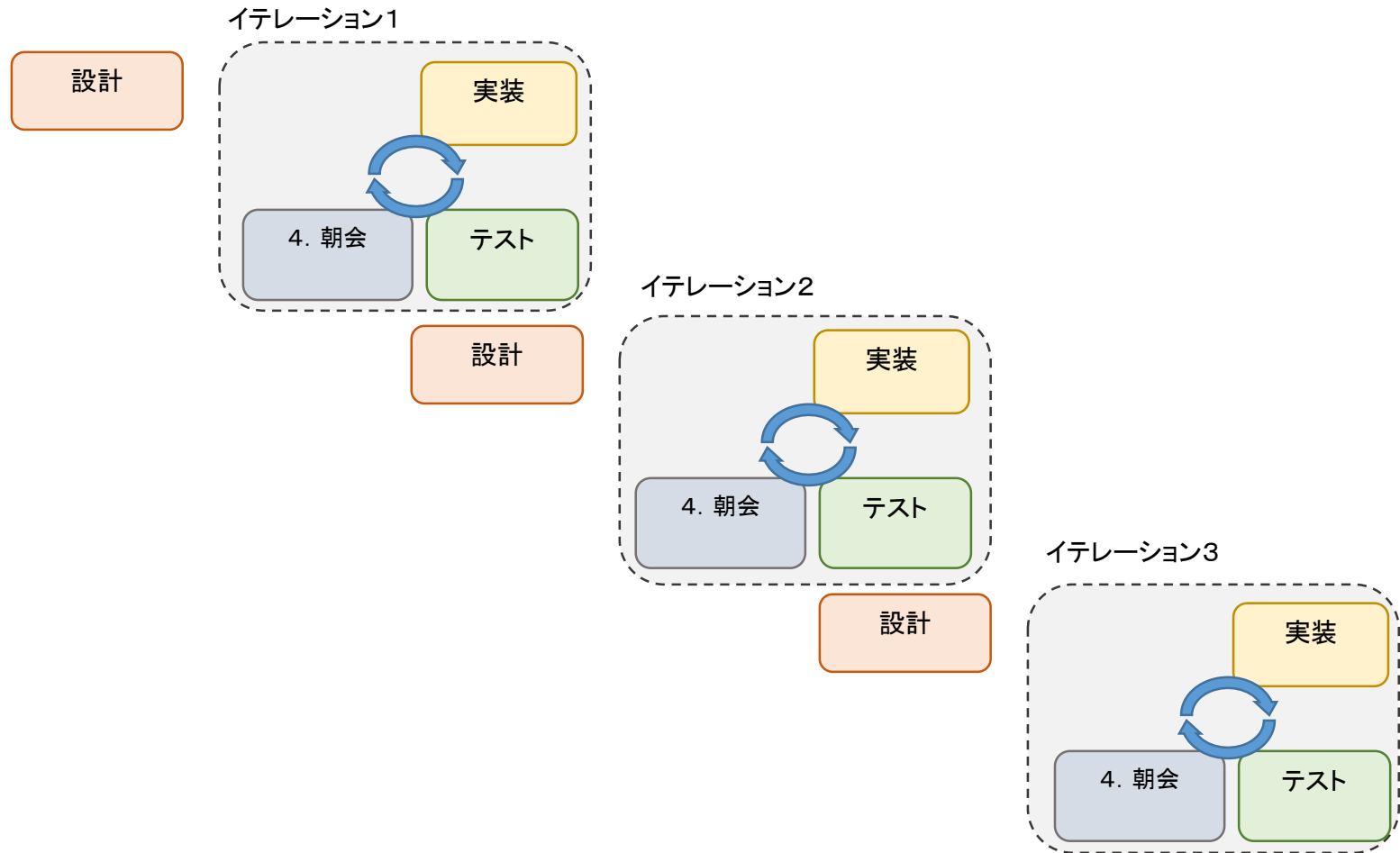
プロダクトオーナー（顧客）と打ち合わせを行い、必要な機能の優先順位と目標期間を設定する。
納期の期間から3分割で実装可能な機能に切り分ける。

■ リリース計画

1. 機能グループ 1
2. 機能グループ 2
3. 機能グループ 3

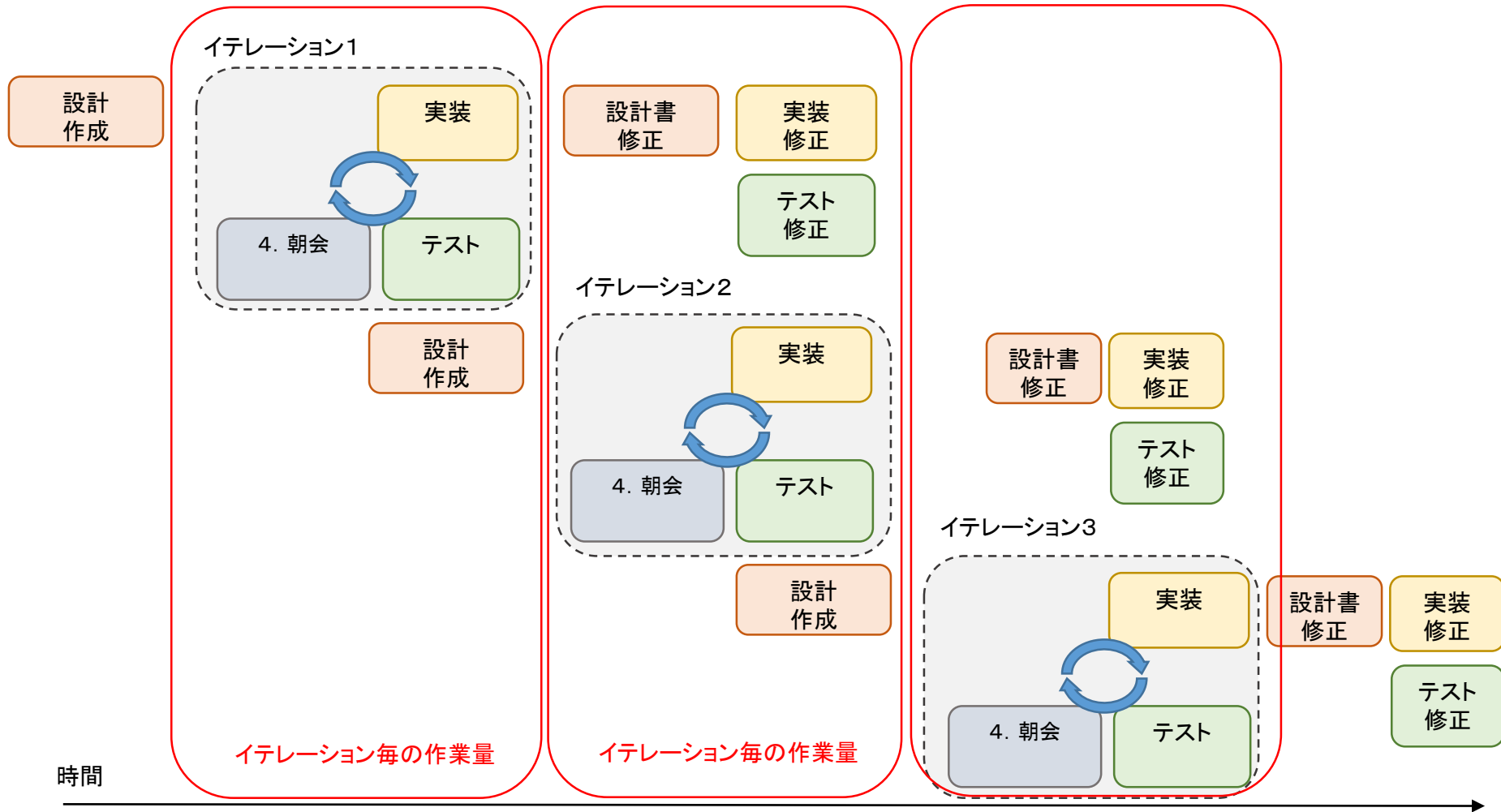
2) 事例2 電力関連会社システム

詳細設計のレビューが必要である為、イテレーションから詳細設計を外だしに・・・



2) 事例2 電力関連会社システム

イテレーション後の改修が誤算・・・



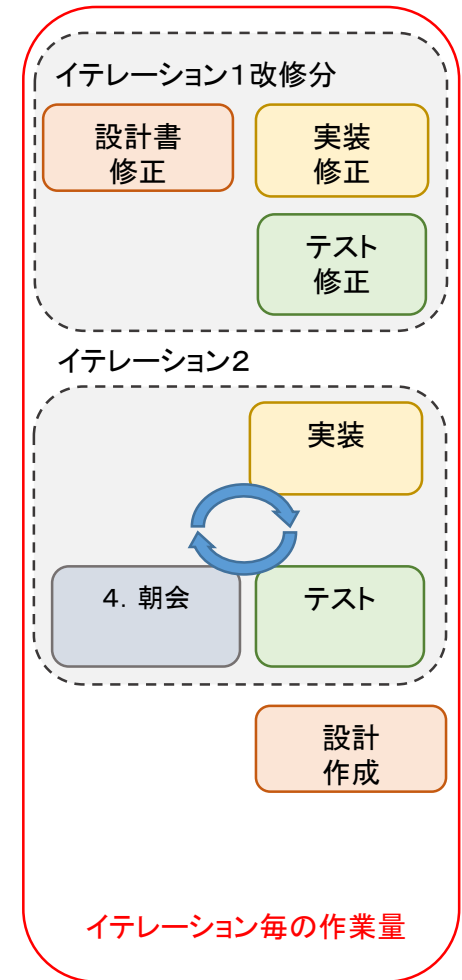
2) 事例2 電力関連会社システム

イテレーション後の改修が誤算・・・

- ・イテレーション1は問題なくスケジュール通り。これならいけると思いきや・・・
- ・イテレーション2から、イテレーション1の改修が思いの外重く、作業量が増えた。
- ・改修の人員を予算の関係上、別途割り当てていなかったため、そのままのチームで改修を行いながらイテレーションを続行せざるを得ない・・・



チームメンバーを増員できない事情があり、残業の嵐・・・



2) 事例2 電力関連会社システム

最終的に・・・

イテレーション後の改修を行うことにより、顧客との意思の乖離をなくし、ある程度顧客満足度は得たが・・・



プロジェクト的には失敗かも・・・

イテレーション後の改修で顧客満足度の向上を狙ったが、その分のコストを積んでいなかった為、結果的にチームメンバに負担が掛かった。

但し、見積りに改修分を積むと他社との競争に負け、受注できない可能性がある。結局、ウォーターフォールで最小の見積りを行い、改修要望は別途で改修するしかないのか・・・（大手ベンダーの儲けの手口と変わらない）

これに懲りず、グローバルワイズは、最小限の予算で顧客満足度を得られる工夫と努力を模索し続けます！！

お問い合わせ先



株式会社グローバルワイズ

URL : <http://www.g-wise.co.jp>

<本社>

愛知県名古屋市中村区名駅南2-14-19 住友生命名古屋ビル21F

TEL : 052-581-2600 FAX : 052-533-3611

<東京事業所>

〒101-0021 東京都千代田区外神田4-5-5 アキバ三滝館9F

TEL : 03-6206-4828 FAX : 03-6206-4829

<青森事業所>

青森県青森市長島2-13-1 AQUA青森スクエアビル7階

TEL : 017-764-0711 FAX : 017-764-0712