


アジャイル開発チームと共に歩む ～発注側から観るアジャイル開発～



KDDI株式会社 サービス企画本部
クラウドサービス企画開発部
荒本、山田、川上



はじめに

POってこんなに大変

開発マネージャーの苦勞

最後に

はじめに

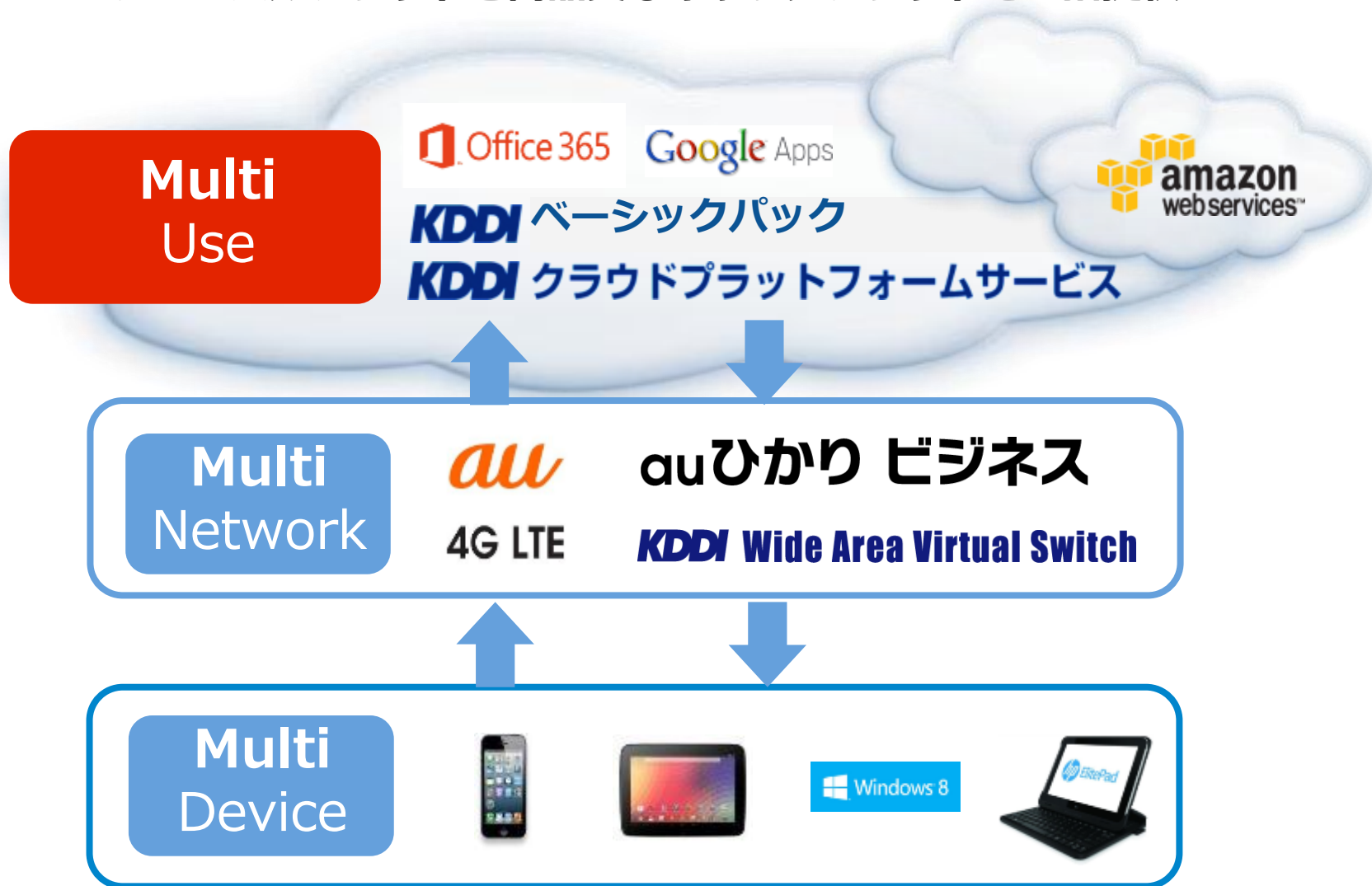




プロジェクトの概要など

KDDIのクラウド戦略

グローバルクラウドと高品質なキャリアクラウドを一括提供



なぜキャリアがクラウドなのか？

1) スケール Scale

- クラウドは先行投資型で、規模の経済が働くモデル
- 通信キャリアが従来やってきたビジネスモデルそのもの
- コンシューマサービス基盤としても活用



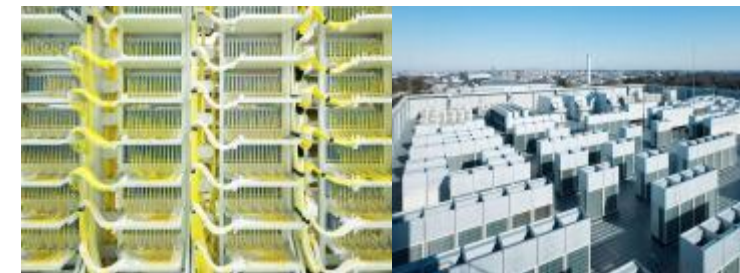
2) クオリティ Quality

- クラウドは電気、ガス、水道と同じユーティリティサービス
- 社会基盤を提供し続けてきたキャリアオペレーション
- 24時間365日サービスを提供し続ける万全の保守体制

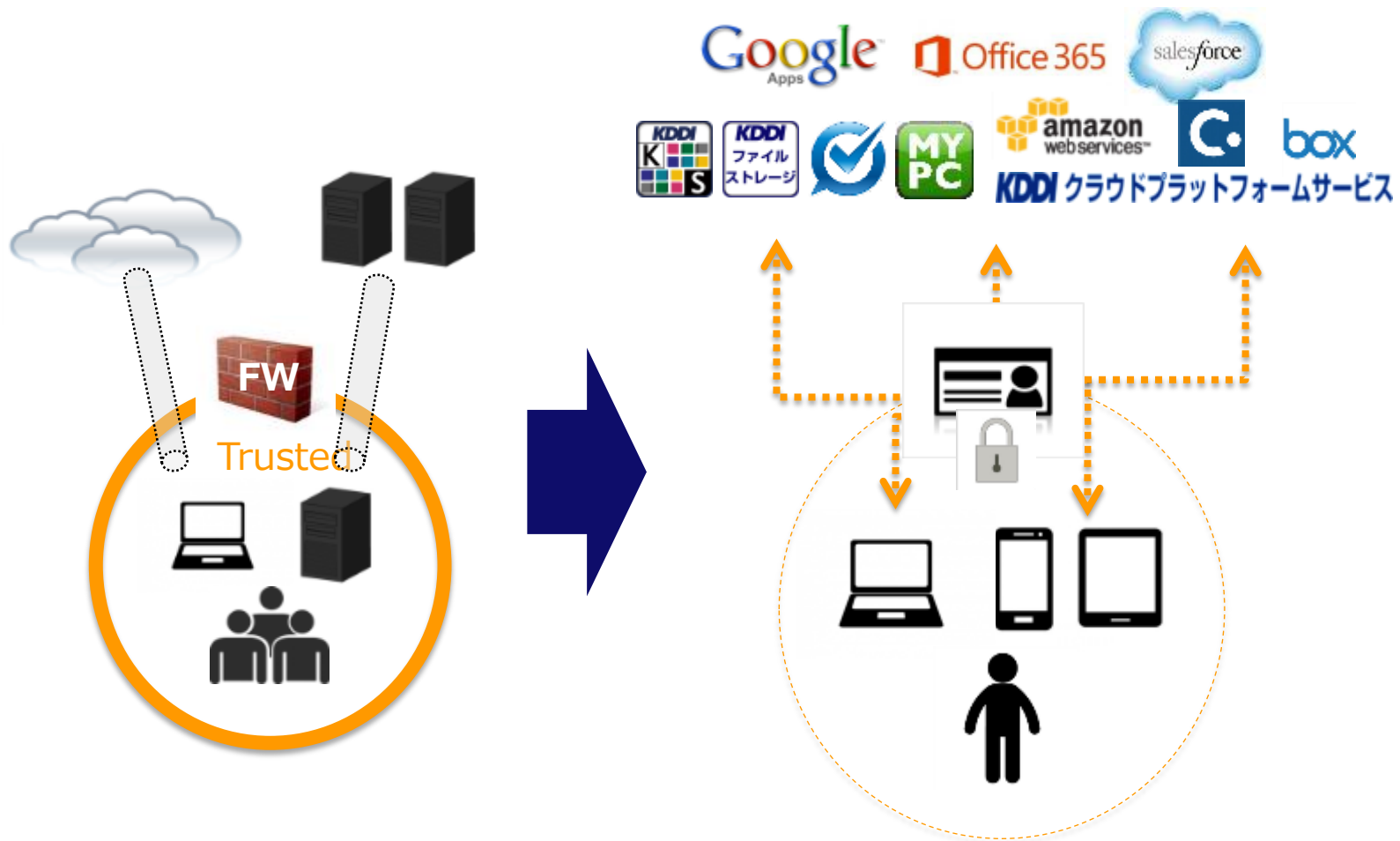


3) ワンストップ OneStop

- クラウドは安定したネットワークの確保が前提
- データセンターから、クラウド、ネットワーク、デバイスまで提供
- 障害の切り分け、管理者エンドユーザまで、幅広くサポート

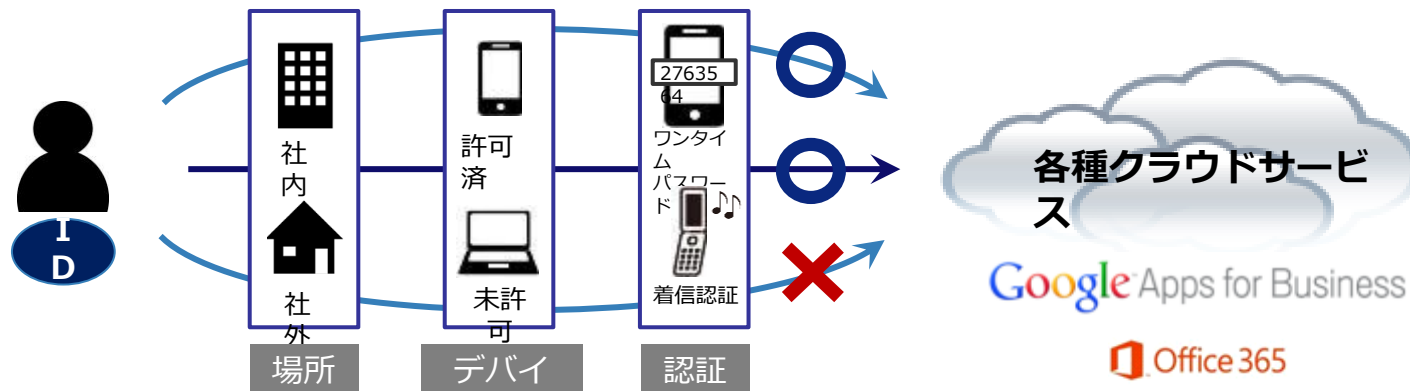


KDDI Business ID~IDaaSをスタート



アイデンティティ管理は、これからの時代のパーソナルなFW

KDDI Business ID



クラウドサービスのご利用を
より安全・安心・簡単にするために

プロジェクトメンバー（今日のスピーカー） 3人の紹介

	直前の案件 Waterfall	KDDI Business ID Agile(Scrum)
	企画担当	PO
	開発部マネージャー	開発マネージャー
	プロジェクトリーダー	PM/SCM

3人の関係性-直前のプロジェクト-

ベンダが出来るって言うのに、なんで出来ないって言うの (A->K)

現場の事分かってないのに、適当な事言わないで下さい (K->A)

サービスの仕様として無いと駄目な事分かるでしょ (Y->K)

仕様書に書いてないこと作れるわけじゃないじゃん (K->Y)

「山田と川上は相性最悪」

「荒本と川上は水と油だから合わない」

と某部長に言わせた関係。

一緒にセミナーに出て話をする関係

今日の話題

1年前は最悪だった相性が、1年間で何故変化したのか？

そこにアジャイル開発で、POと開発Tの一体感を高めるポイントがあるのでは？

以降、荒本と山田から具体的な話をさせていただきます。

POって大変





Agileのきっかけ

私のミッション

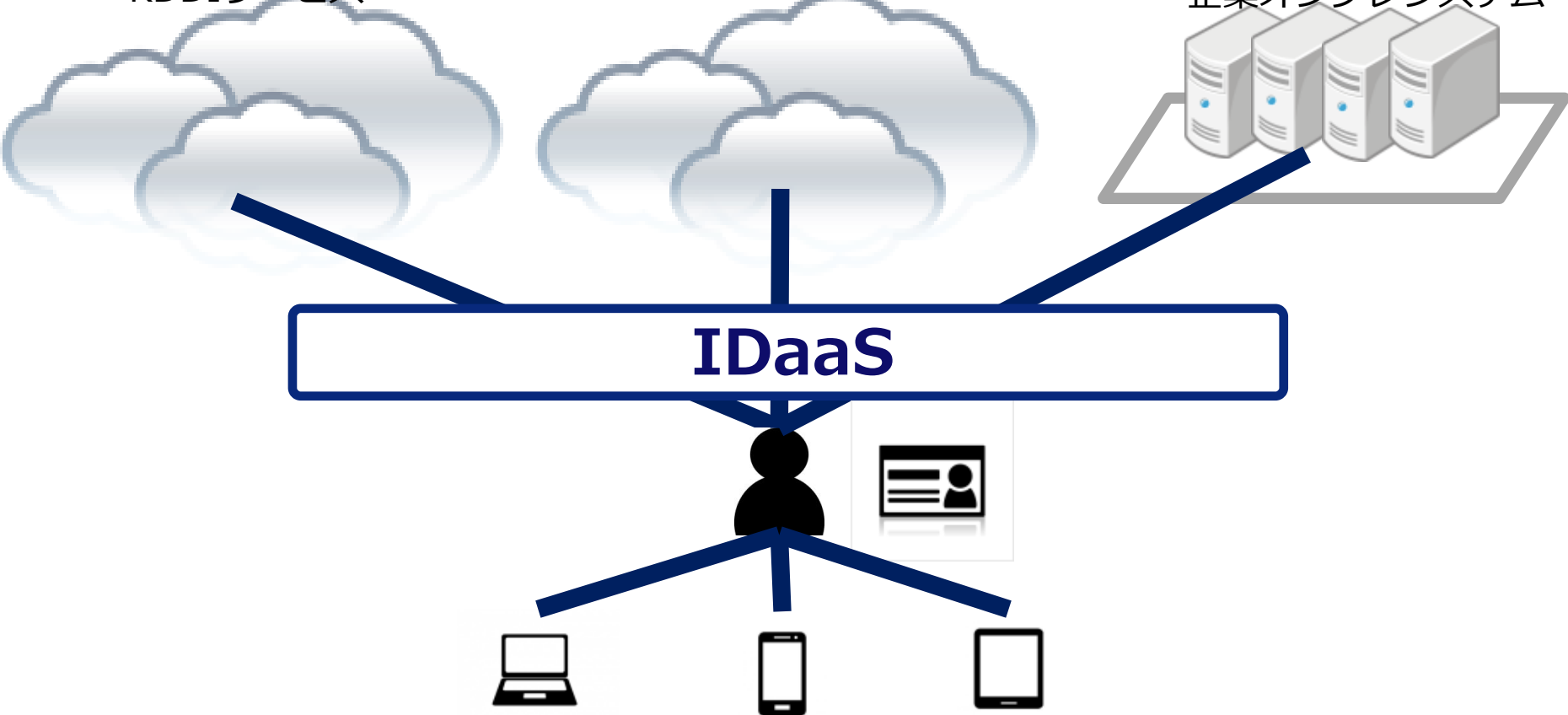
(Enterprise向けクラウドサービス企画)

1つのIDであらゆるサービス、システムへ
いつでもどこでも安全に繋がる世界へ

KDDIサービス

グローバルサービス

企業オンプレシステム



Moving Target

しかし、Cloud & Mobileの市場は
先読みが容易ではない



Agileの決断

先読みが難しい
⇒変化への抱擁、継続的デリバリー

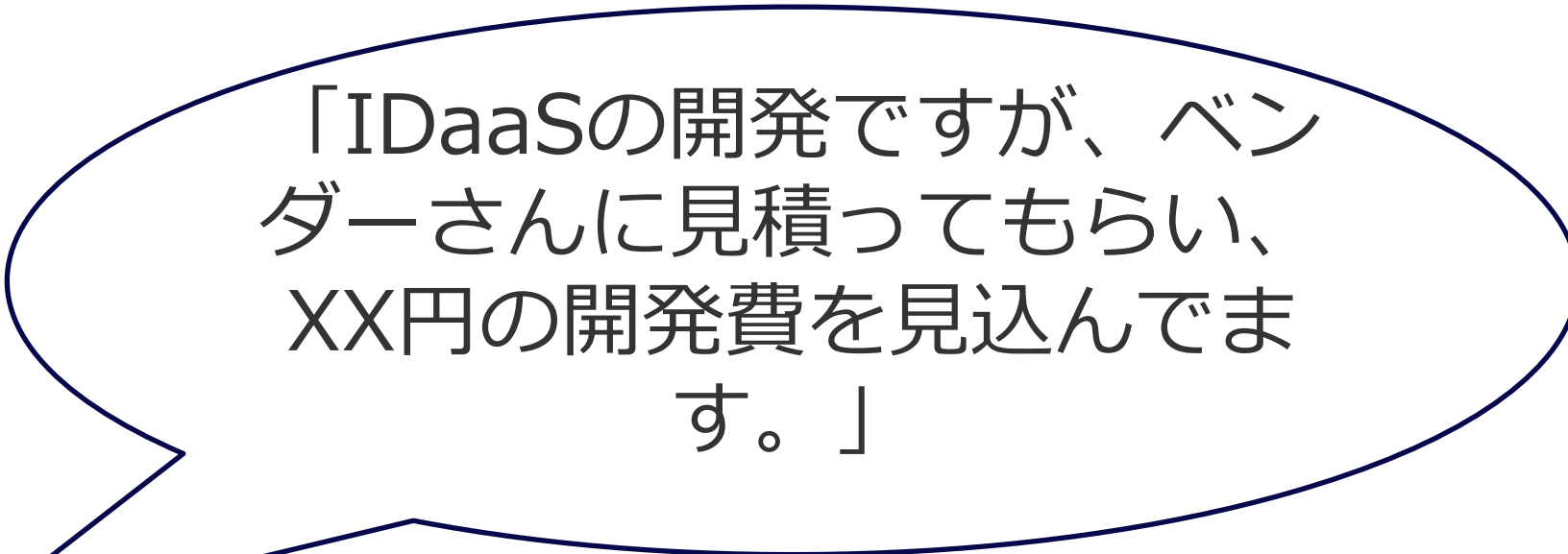
KDDIサービス

グローバルサービス

企業オンプレシステム

Agile

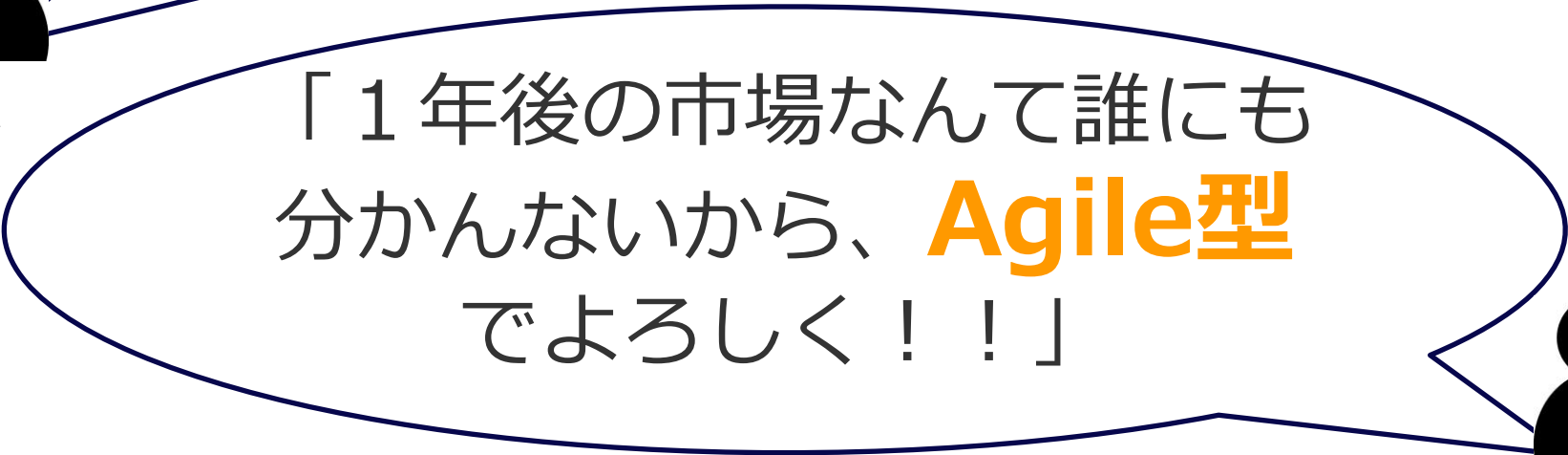




「IDaaSの開発ですが、ベンダーさんに見積ってもらい、XX円の開発費を見込んでます。」



私



「1年後の市場なんて誰にも分かんないから、**Agile型**でよろしく!!!」



上長

余談：サービス企画の立場から

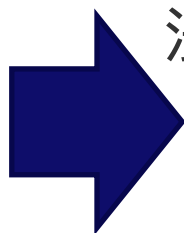
これまでのWF型

一度出した要件を変更することは、開発費も増え、スケジュールも遅延を招く
「やってはいけない行為」



Agileへの期待

変化を抱擁しながらいいものが作れる、しかも早くて安い。なんていい手法なんだ！



ポイント①

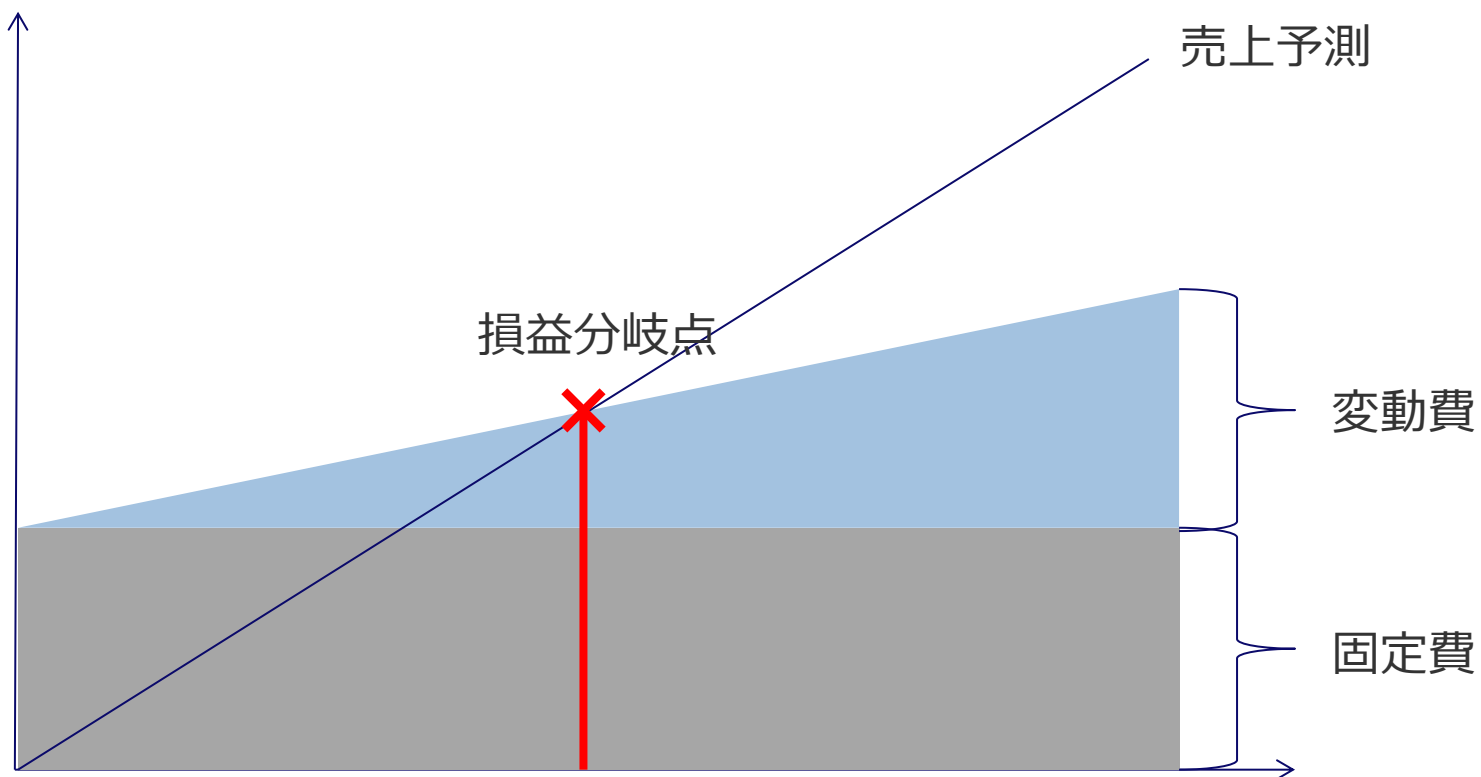
- **環境変化**への追従からAgileは必然だった
- 意思決定者の1人に**Agile推進者**がいたこと



社内決裁

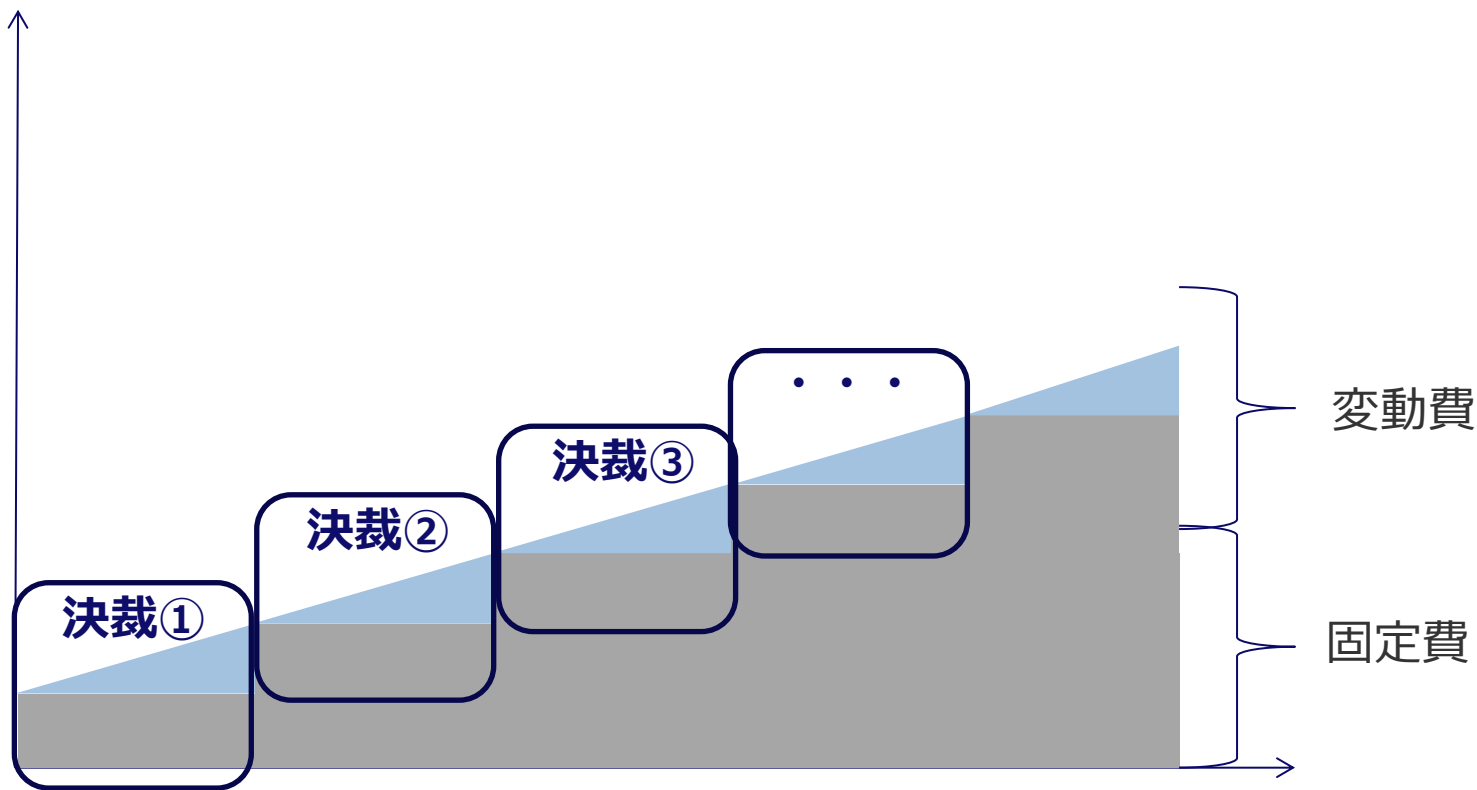
社内決裁

当社の決裁フローは、
ウォーターフォール型を前提としたもの



Small Start

3カ月おきに、サービス/機能追加を投入し、
都度決裁をとるスタイルとした



注力したこと

定期的に経営層にデモを行い、
Agileアピールを続けた



ポイント②

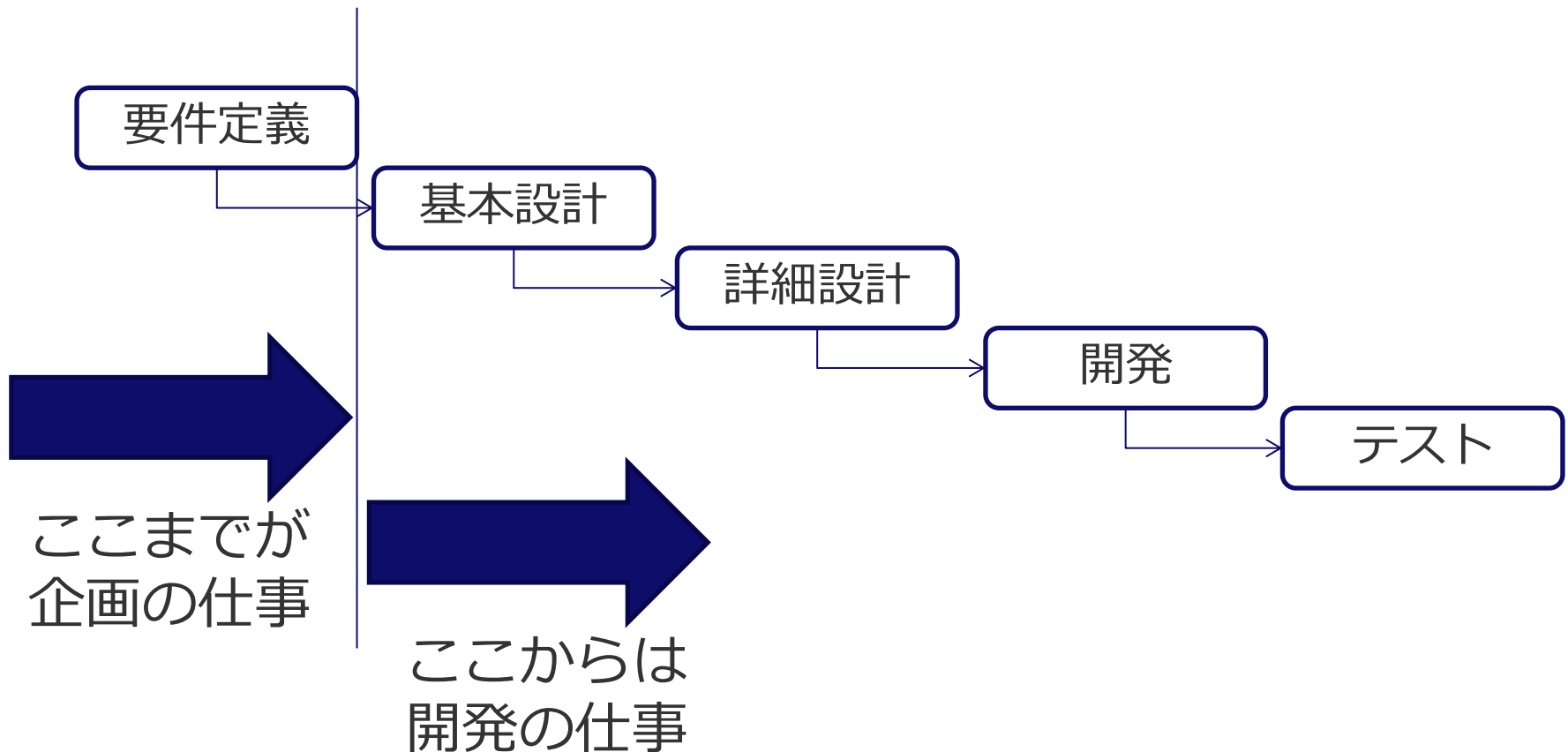
- 環境変化への追従からAgileは必然だった
- 意思決定者の1人にAgile推進者がいたこと
- Agile型での決済に無理に拘らないこと
- 経営層にAgileアピールをし続けること



要件定義 & 開発の推進

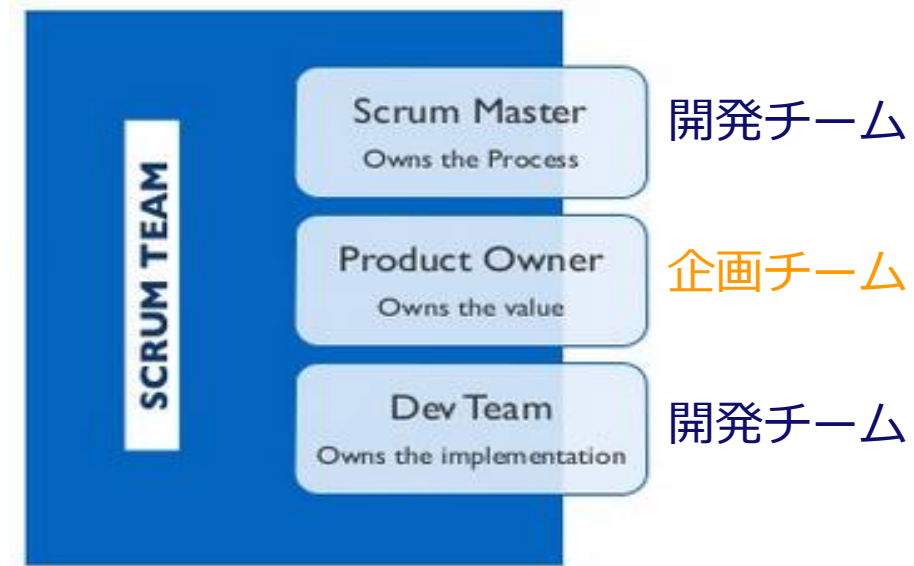
プロジェクトはスタートしたものの、

メンバーはウォーターフォール型しか
経験したことがない



始めに取り組んだこと

① Scrumによる チーム作り



② Agile トレーニング

MVP
Team Building
Less Mass
TDD

• • •

特に注力したこと 「Less Mass」

PO

出来るだけ多くのストーリーを
リリースしたい



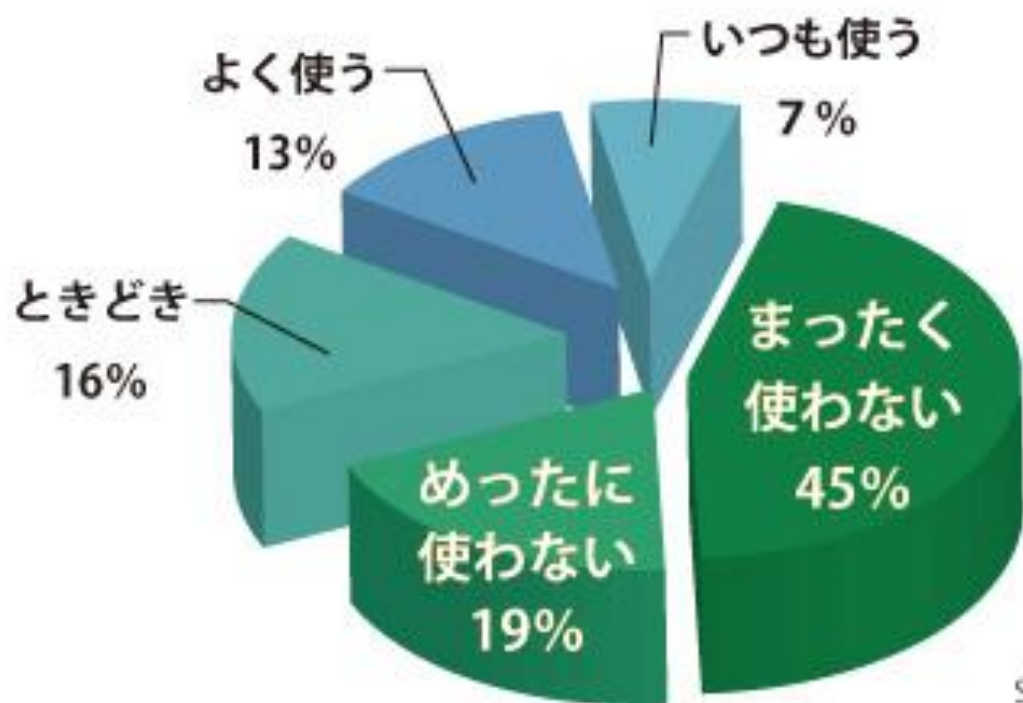
開発
チーム

タイムボックス



特に注力したこと 「Less Mass」

いかにして作りこまないかを追求しない限り、早くて安くていいものは作れない



まったく使わない
めったに使わない

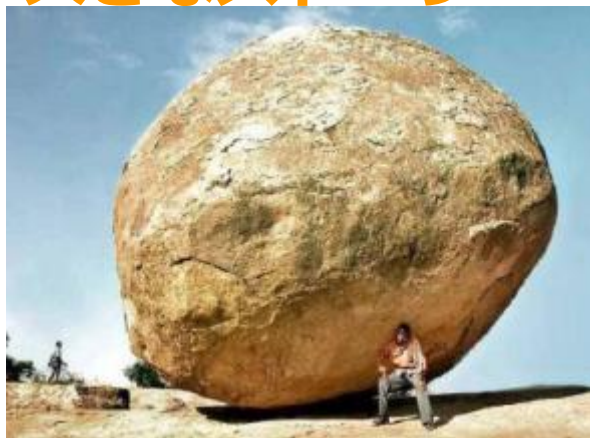


Standish Group Study Reported at XP2002 by Jim Johnson,
一般的なシステムにおけるフィーチャーや機能の使用頻度

特に注力したこと 「Less Mass」

頭では理解したつもりだが立ち上がりは

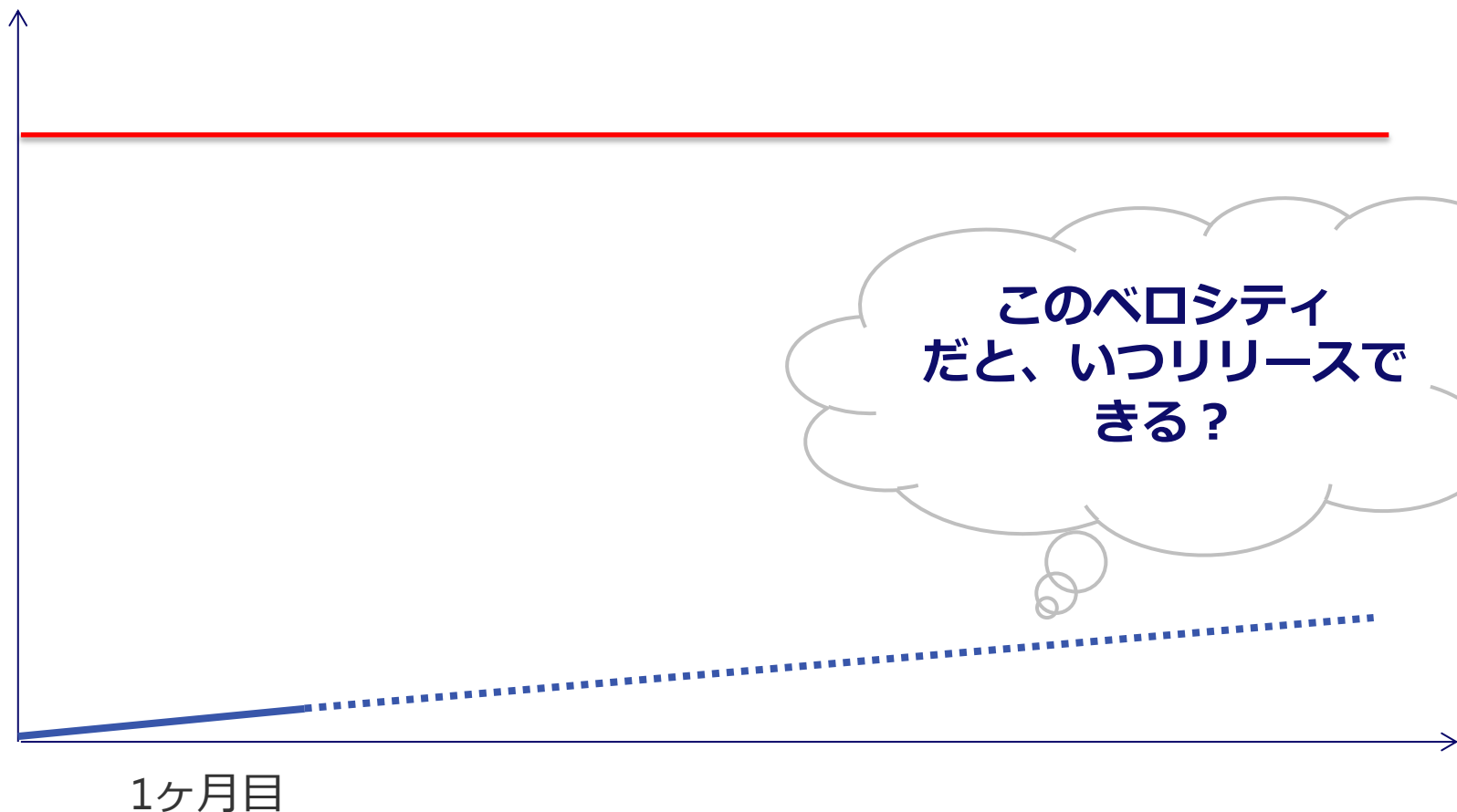
大きなストーリー



ストーリーイメージ：
Aサービスへのログインが
人毎に場所、デバイスによって、
制御されること

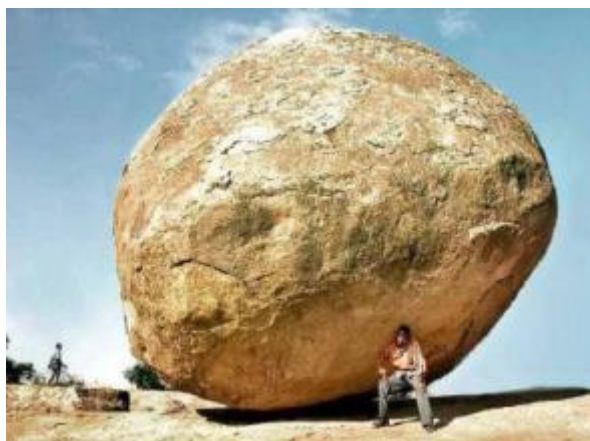
特に注力したこと 「Less Mass」

初期は低空飛行のベロシティに



特に注力したこと 「Less Mass」

ストーリー粒度を細かくし、
ストーリー単位で実施有無/優先順位を判断



ストーリーイメージ：
Aサービスへのログインが
人毎に場所、デバイスによって、
制御されること



- ① Aサービスへのログインができる
- ② 認めれた場所からのログイン可能
- ③ 認められてない場所からのログインは、エラーを出力する
- ④ 認められたデバイスからのログイン可能
- ⑤ 認められていないデバイスからのログインはエラーを出力する

特に注力したこと 「Less Mass」

開発メンバーにもストーリー実施可否の
権限を与えた

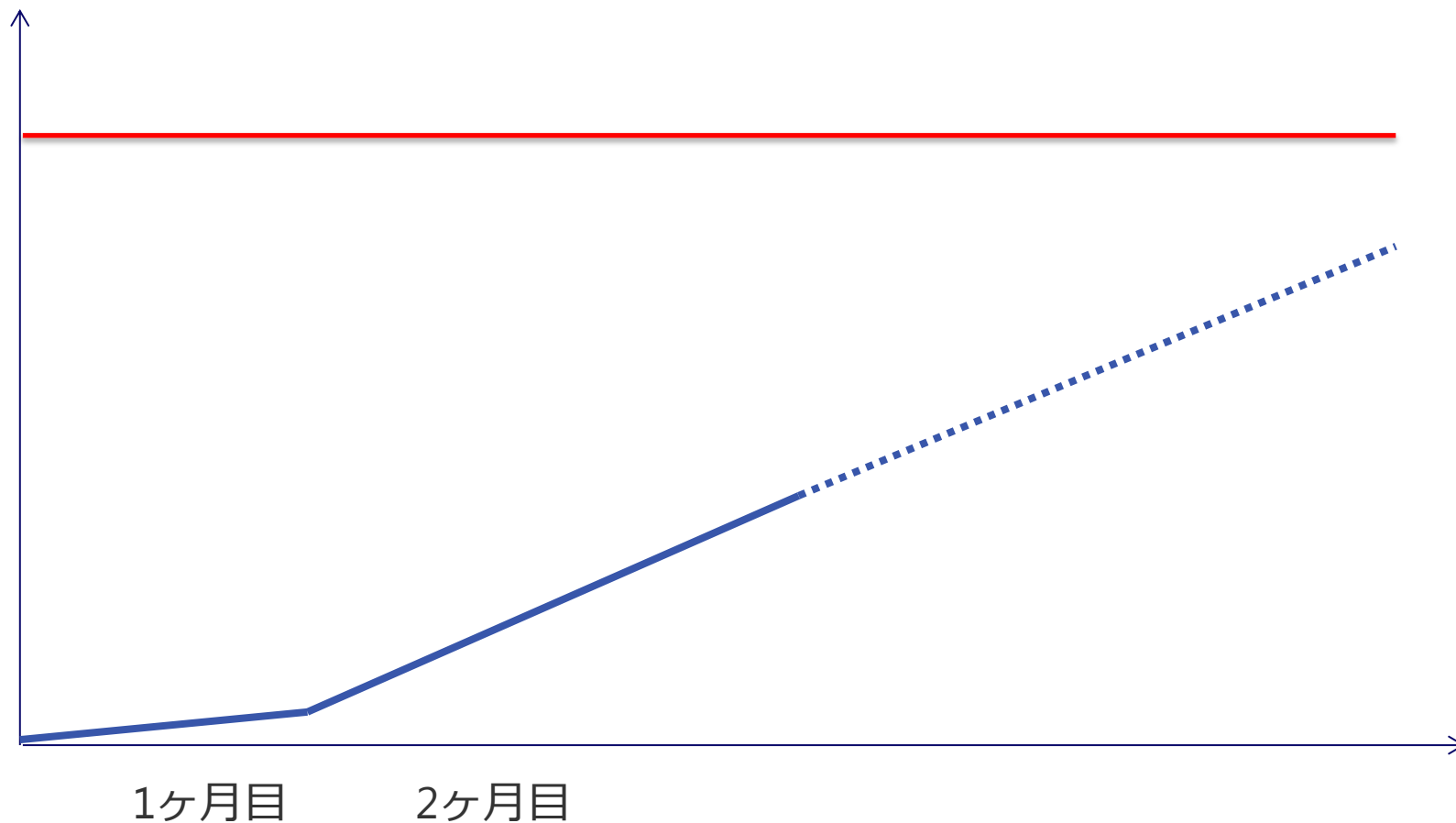


×：要件を定義されたから作る
○：必要性を理解したから作る

※但し、最終決定権はPOが持つ

特に注力したこと 「Less Mass」

徐々にベロシティが上がってきた



特に注力したこと 「Team Building」

週に1回KPTを繰り返し、
改善活動を全員に浸透

KEEP

やってよかったこと。
次回もやりたいこと。
継続したいこと。

TRY

次回新たにやってみ
たいこと。

例：
超過することが
分かった時点で
チーム内でミーティング

PROBLEM

課題だったこと。
次回改善したいこと。

例：
開発タスクの消化が見積時間を大幅に超過

特に注力したこと 「Team Building」

Scrumチームは崩さず

KPTを繰り返す

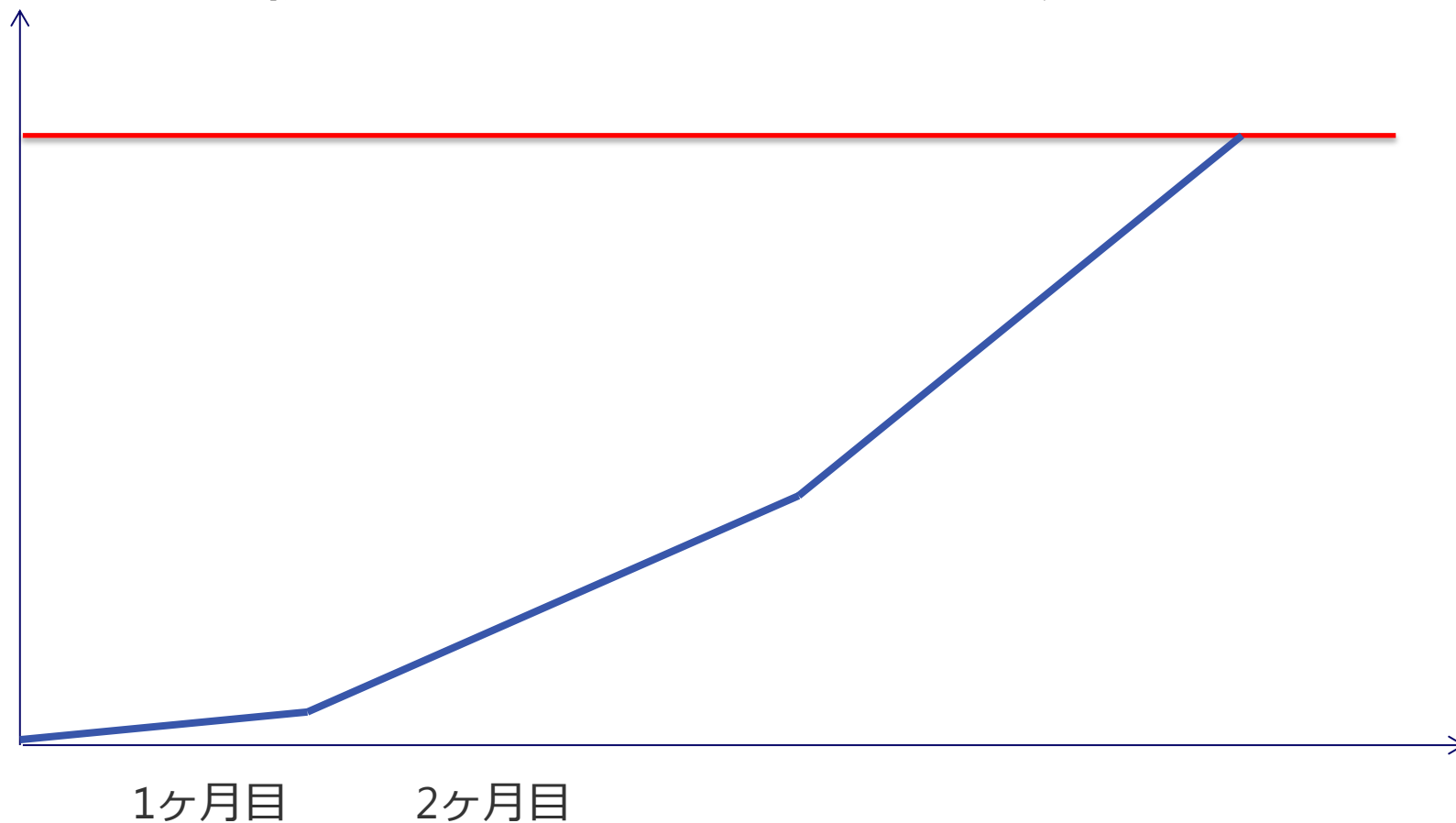
あとは、

**communicate,
communicate,
communicate.**

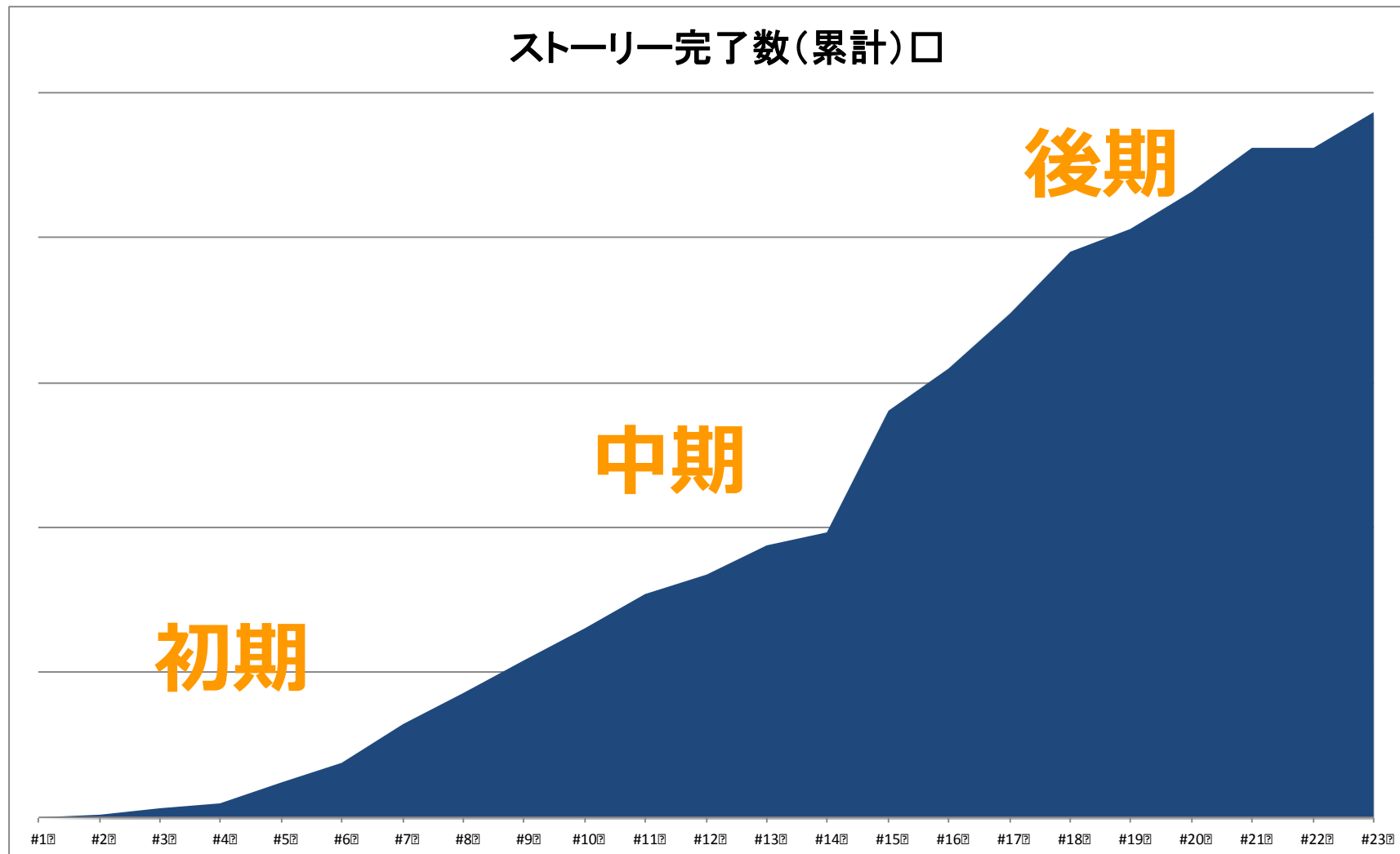


結果として、

変化も抱擁しながら、
当初計画通りにリリース完遂



実際のストーリー消化



ポイント③

- 環境変化への追従からAgileは必然だった
- 意思決定者の1人にAgile推進者がいたこと
- Agile型での決済に無理に拘らないこと
- DemoによるAgileアピールをし続けること
- Agileは**魔法の杖ではない**ことを理解する
- 要件定義の**マインドチェンジ**を徹底すること
- メンバー全員が**ジブンゴト化**する風土を作ること



開発マネージャーの苦勞



アジャイル開発との出会い

W/F開発していた時

アジャイル開発とは

表面的には知ってはいたが・・・

W/F開発していた時

アジャイル開発とは

表面的には知ってはいたが . . .

**「うちの会社では
無理だな。」**

なぜ「うちの会社では無理」と思ったいたか



**自社で開発できる
人達の開発手法だよ**

なぜ「うちの会社では無理」と思ったいたか



**自社で開発できる
人達の開発手法だよ**

当社では

- 社内に**プログラマ**という職種の人はいない
- 仕様を書いて実装は**SIerに委託**するケースが多い

なぜ「うちの会社では無理」と思ったいたか



**開発することを走りながら
決めるなんて、どうやって
会社の承認を取るのだ？**

なぜ「うちの会社では無理」と思ったいたか



**開発することを走りながら
決めるなんて、どうやって
会社の承認を取るのだ？**

当社では

- **開発内容**
- **スケジュール**
- **コスト**

をコミットして開発承認される仕組み

なぜ「うちの会社では無理」と思ったいたか



**成果物を定義しない
発注なんてできない！**

なぜ「うちの会社では無理」と思ったいたか



**成果物を定義しない
発注なんてできない！**

当社では、

- 開発部門が作成した仕様書に基づき、
発注先を**調達部門が決定**する仕組み

しかし、課題も抱えていた

- 1 長いリリースサイクルにより
サービス競争力の低下
- 2 ベンダ丸投げにより技術力の低下
技術の目利きが出来ないための
品質の低下
- 3 対ベンダ、対部門間の牽制により
角が取れた丸いサービス

A decorative network pattern of blue lines and dots is visible in the top right corner of the slide.

Let's try anyway.

- ① **開発チームの在り方**
- ② **受発注の関係**
- ③ **社内部門との関係**



変えたこと その1 開発チームの在り方を変える

これまでの開発チーム

RFP作成と受入試験

これまでの開発チーム

RFP作成と受入試験



アジャイル開発を始めるには

設計・実装まで踏み込む

- SIerだけにアジャイル開発してもらっても、抱えている課題解決にならない
- スプリントを回すために実装まで理解が必要



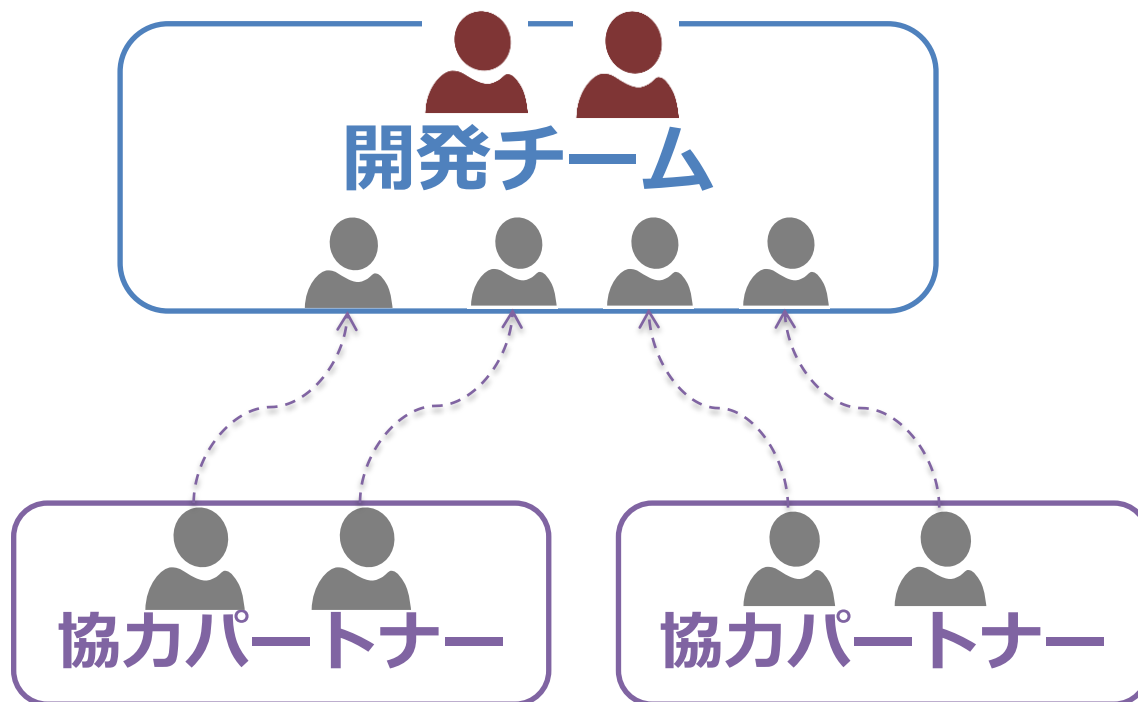
自社でプログラミングする 内製を開発方針とする

自社でプログラミングする 内製を開発方針とする



しかし、
時間が掛かる

外部のパートナーを 社内に置き、社員と**共同開発** することでスキルを吸収



トレーニングより**実践**（**ペアプロ**） が圧倒的に速い

- **アーキテクチャー設計・ミドルウェア製品選定を自らが決定**
- **プログラミング未経験者が、6ヶ月間のプロジェクト期間で商用コード実装**





変えたこと その2 受発注の関係を变える

受発注の壁

KDDI
(発注側)

壁

Sier
(受注側)

受発注の壁

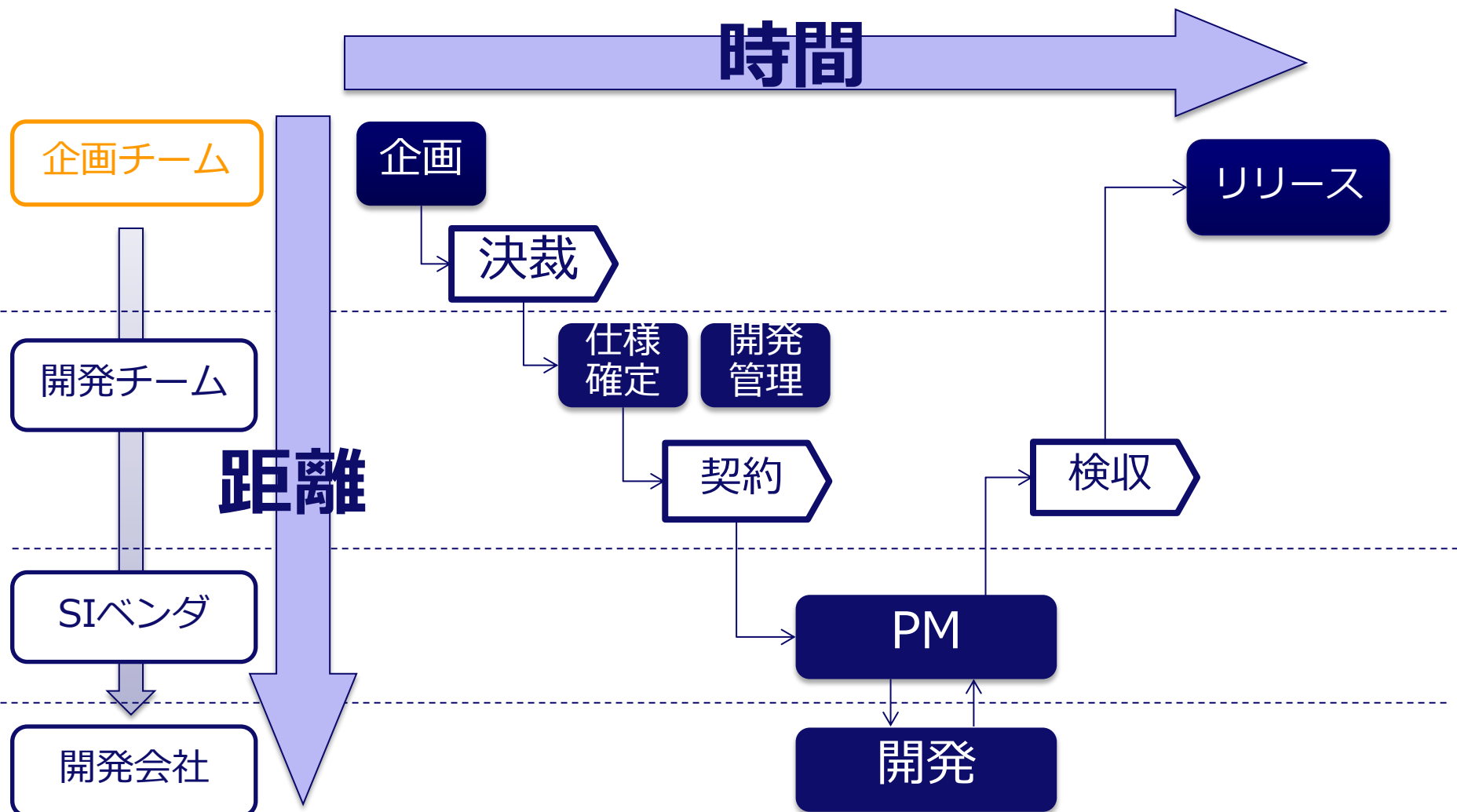


イノベーションが起こりにくい

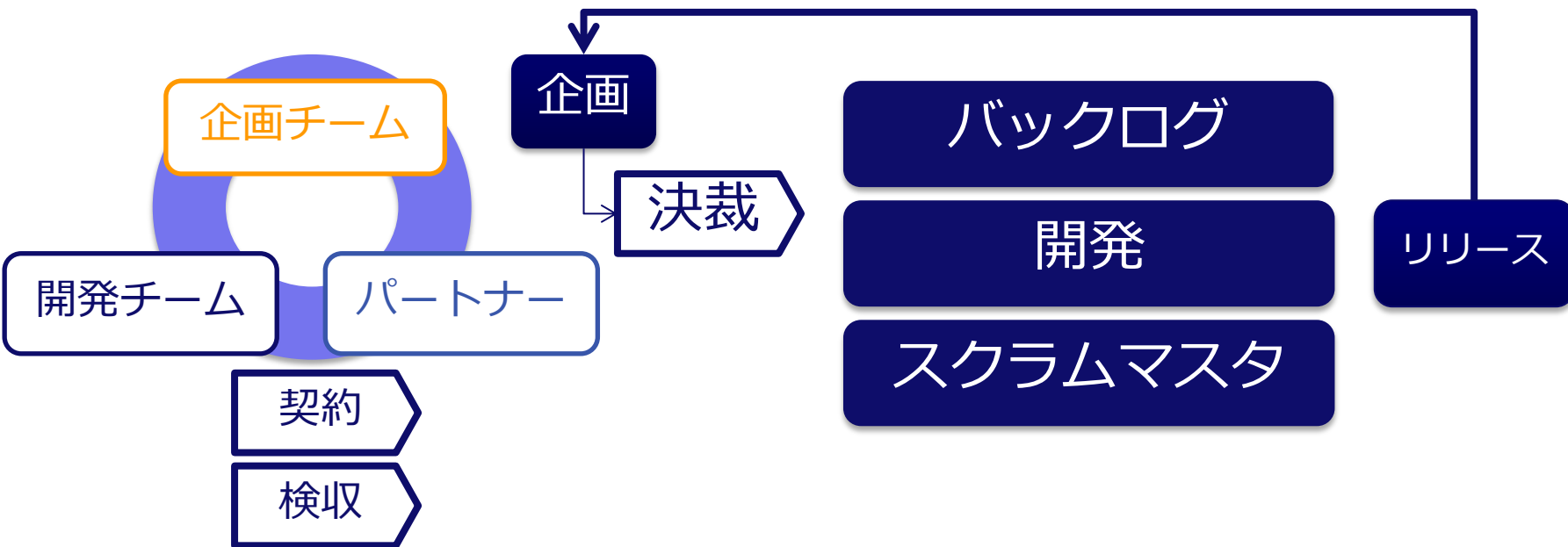
ユーザ企業と
SIベンダを
縦の関係から
横の関係へ

縦の関係から

企画と開発者の距離が遠く、見通せない



企画部門と開発者が一つのチーム



そのためには契約形態を変える

アジャイル開発では

**契約時に成果物が
定められない**



請負契約は向かない

契約を変える

準委任契約

準委任契約



発注側

成果物の責任を負う覚悟

受注側

言われた通り実装するだけではなく、成功のために個人の技術スキルを発揮



変えたこと その3 社内部門との関係を変える

調達部門との関係を変える

これまで

成果物に対する調達

- 開発部門は仕様書（RFP）を定義
- 調達部門は仕様を満たすものを如何に安価に購入するか
- 競争入札が原則

RFP作成

競争入札

技術評価

選定
発注

システム要求仕様

開発提案書

請負契約

調達部門との関係を変える

アジャイルでは

**エンジニアチームの工数
を調達**

**パートナーのエンジニア
スキルセットで競争入札
して評価**

開発に必要な技術スキルセットでRFP・選定

要求スキルセットを提示し、エンジニアチームの提案を依頼

スキルセット例：

アジャイル開発の経験

テスト駆動開発

フロントエンドMVC

など

求める必須レベルとのマトリクスで提案依頼

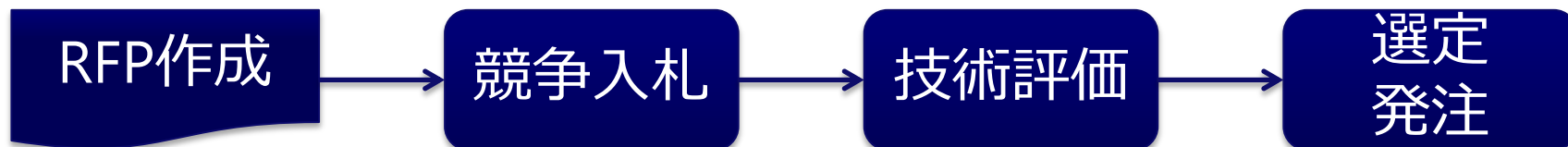
提案内容をポイント化し定量評価

2社を選定した混成のチーム

調達方法の従来との比較

業務フローは変えない

通常

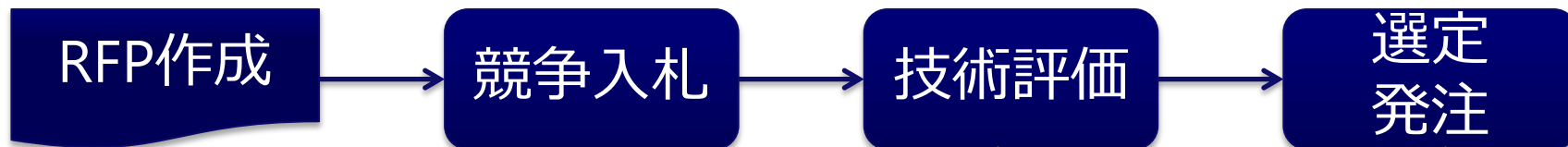


システム要求仕様

開発提案書

請負契約

アジャイル



求められる
スキルセット

エンジニアチーム
提案書

準委任契約

電気通信事業者として システムの品質が最も大切



総務省

Ministry of Internal Affairs
and Communications

▶ご意見・ご提案 ▶English

Google™ カスタム検索

● サイト内

ここに検索語句を入力



[総務省トップ](#) > [広報・報道](#) > [報道資料一覧](#) > [KDDI株式会社に対する事故防止に係る指導](#)

報道資料

平成25年7月16日

KDDI株式会社に対する事故防止に係る指導

総務省は、本日、KDDI株式会社に対し、同社が提供する携帯電話サービスにおいて生じたような事故が再発しないよう、設備等の総点検を実施し、その結果を取りまとめるとともに、個人情報の滅失を含む今後の事故の再発防止に向けた取組等について、報告するよう指導しました。

1 経緯

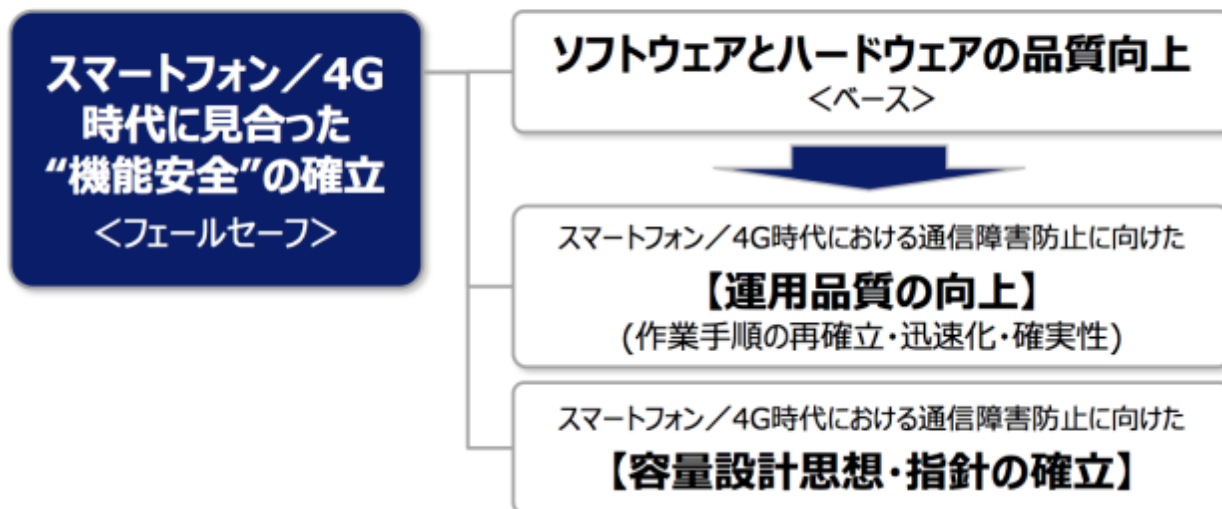
KDDI株式会社（以下「KDDI」といいます。）が提供する携帯電話サービスにおいて、平成24年12月31日、平成25年1月2日、同年4月16日から19日、27日、5月29日及び30日において重大な事故を含む大規模な通信障害が発生しました（概要は別紙¹のとおりです。）。

平成24年12月31日、平成25年1月2日、4月27日、5月29日及び30日の事故は、平成24年9月から開始されたサービス

重大障害を起こさないため 開発・運用プロセスを細かに 定めて品質を守っている

一連の通信障害の対策に関する基本方針

11



ウォーターフォール型が ベースの開発プロセスの中で アジャイルを行う場合の 差分を明確化

RFPの記載項目

運用要件

設計書レビュー

技術評価

運用部門との関係を変える

開発プロセスを逸脱しないことの理解を得る



運用部門からもスクラムイベントに参加
求められる運用要件をバックログ化



ユーザ企業でアジャイル開発を
成功させるために欠かせないこと

**SIerだけにアジャイルで
お願いしても大して変わらない**

**アジャイル開発とは開発部門に
おける開発手法の一つではない**

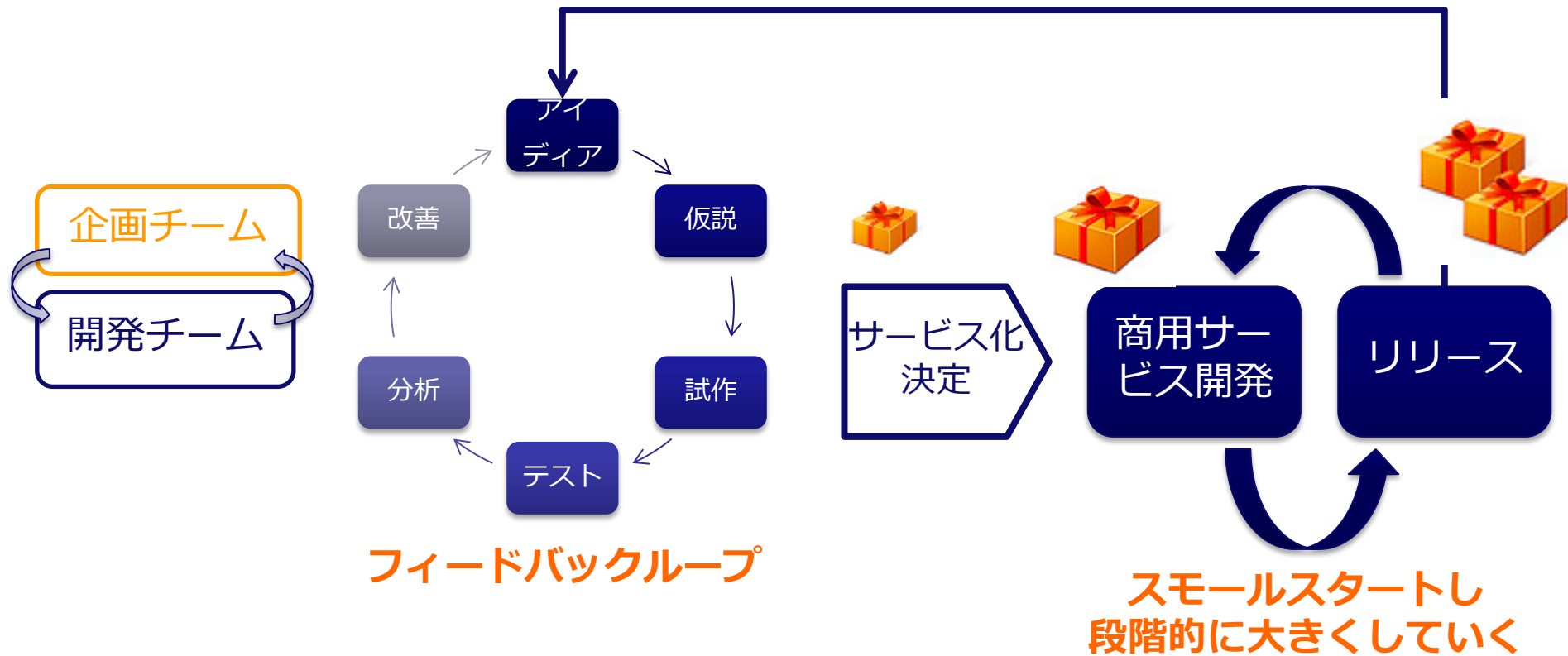
発注側のユーザ企業が
変わらなければ
アジャイルで変化は起きない



次への取り組み
イノベーションを生み出す開発チームへ

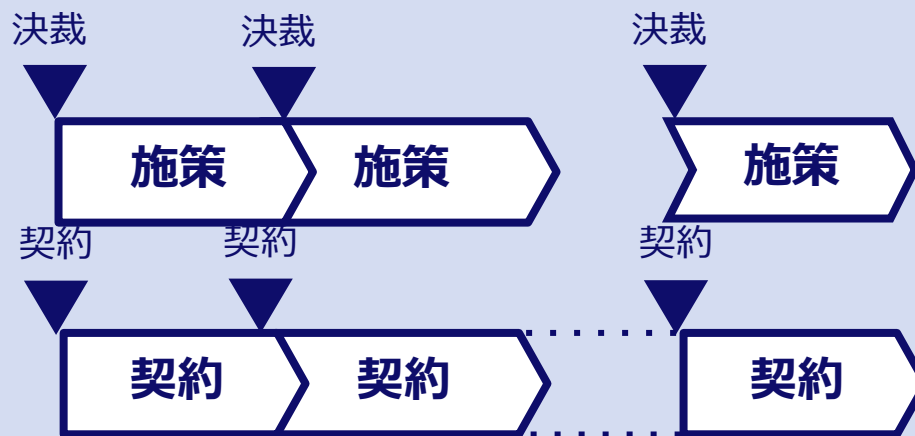
目指すサービス開発プロセス（3つのループ）

アイデアを**プロト開発**、価値のあるものをサービス化
目に見えるものをユーザ（社内/社外）に見てもらい短いサイクルの中で軌道修正

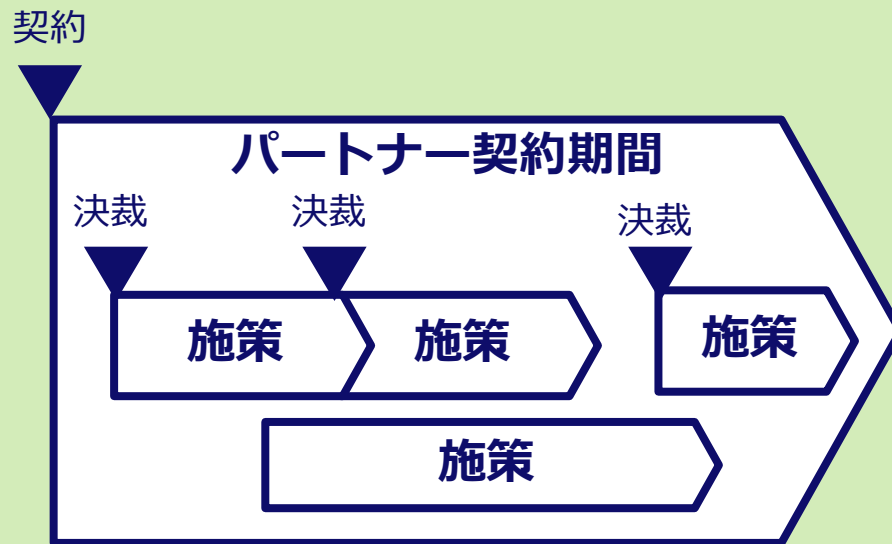


KDDIと開発パートナーが家族になる関係づくり

施策の決裁に
依存した契約



期間で定額契約




チームの一員としてイノベーションを
一緒に起こせる開発チームへ



キーパーソンとなる優秀なエンジニア
が集う開発現場・契約制度へ

最後に





組織やプロセスの壁を突破するには、
実務者同士の相互理解と
組織的なサポート、
上位者の理解
が必要。



Quality Cloud