

実例:ユーザー企業責任で 25サイトをアジャイルに開発

前田 圭一郎

株式会社リクルート

システム基盤推進室

アプリケーションソリューショングループ

はじめに

- **内製しない体制**
- **限られた納期**
- **ユーザー企画部門との成果物責任**

そんな環境で**アジャイル**?

どんな事例か？

- 完全内製ではないユーザ企業が、
- 自社システムのビジネス価値を高めるために、
- アジャイル開発の考え方を自分たちの現場に適用させる試みを組織的に、
- 年単位（3年）で取り組んできた



- 素晴らしい正解ではないかもしれない
（そもそもリクルートの中でも、いろいろなやり方があります）

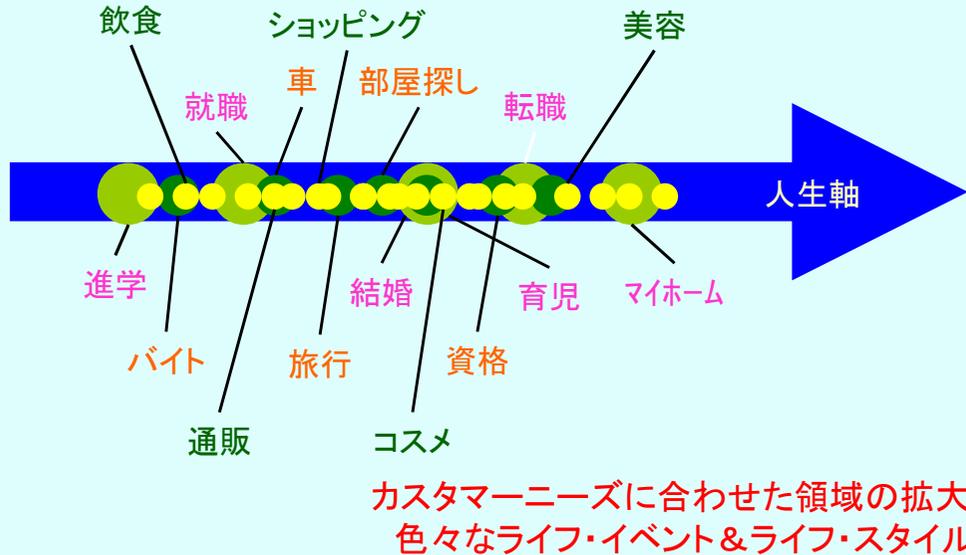
しかし、ビジネスの現場、開発の現場で、3年間、
「どうすればもっとよくなるか？」
を継続的に考えて実行してきた結果です。

特に、ヒトの考え方・習慣をポイントに、皆さんの考えるヒントになれば

リクルート システム部門の概要 (とりまく人々)

リクルートとは

◆リクルートの会社概要



ライフイベント(節目)領域

ライフスタイル(日常)領域

フリーペーパー
WEB・モバイル
その他

<p>人材領域</p>	<p>教育</p>	<p>住宅</p>	<p>販促</p>	<p>販促</p>	<p>新領域</p>
--------------------	------------------	------------------	------------------	------------------	-------------------

◆システム部門の役割



- MPと協働しながら、新メディアの開発と、既存メディアの改善
⇒ Webメディア・サービスを構築する役割
- メディアを開発する際に、ベンダーと協働しながらプロジェクトマネジメントをする役割

リクルートとは

◆システム部門の組織概要

●システム部門 各事業担当部門

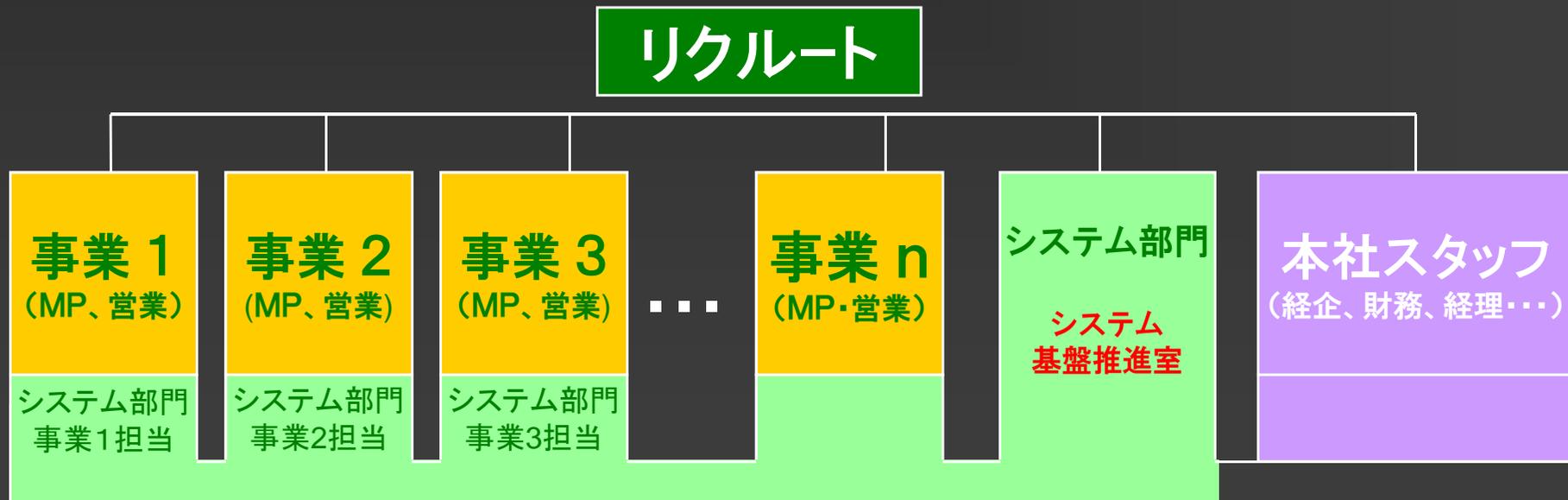
- ・各事業に密着し、MPと協力して、リクルート側PMとしてシステムの構築や保守を行なう。

●システム部門システム基盤推進室

(アプリケーションソリューションG、インフラソリューションG)

- ・全社的な視点から、共通性の高いサービスを提供。
- ・開発手法の検討・実践。
- ・リクルートの開発の統括業務(コスト、生産性)
- ・新製品の検証
- ・インフラの提供

本日のお話



Webメディアの サービス設計・開発の課題

背景

◆Webメディア・サービス設計・開発の リクルートの課題

●“情報誌”のメディア立上げが勝ちパターンの文化。

従来の“メディア立ち上げ型”:

- ・新たな情報誌カテゴリを生み出し、世の中に提供。
- ・スタート時に最大の効果を。
- ・ビジネスであり、失敗は許されない。

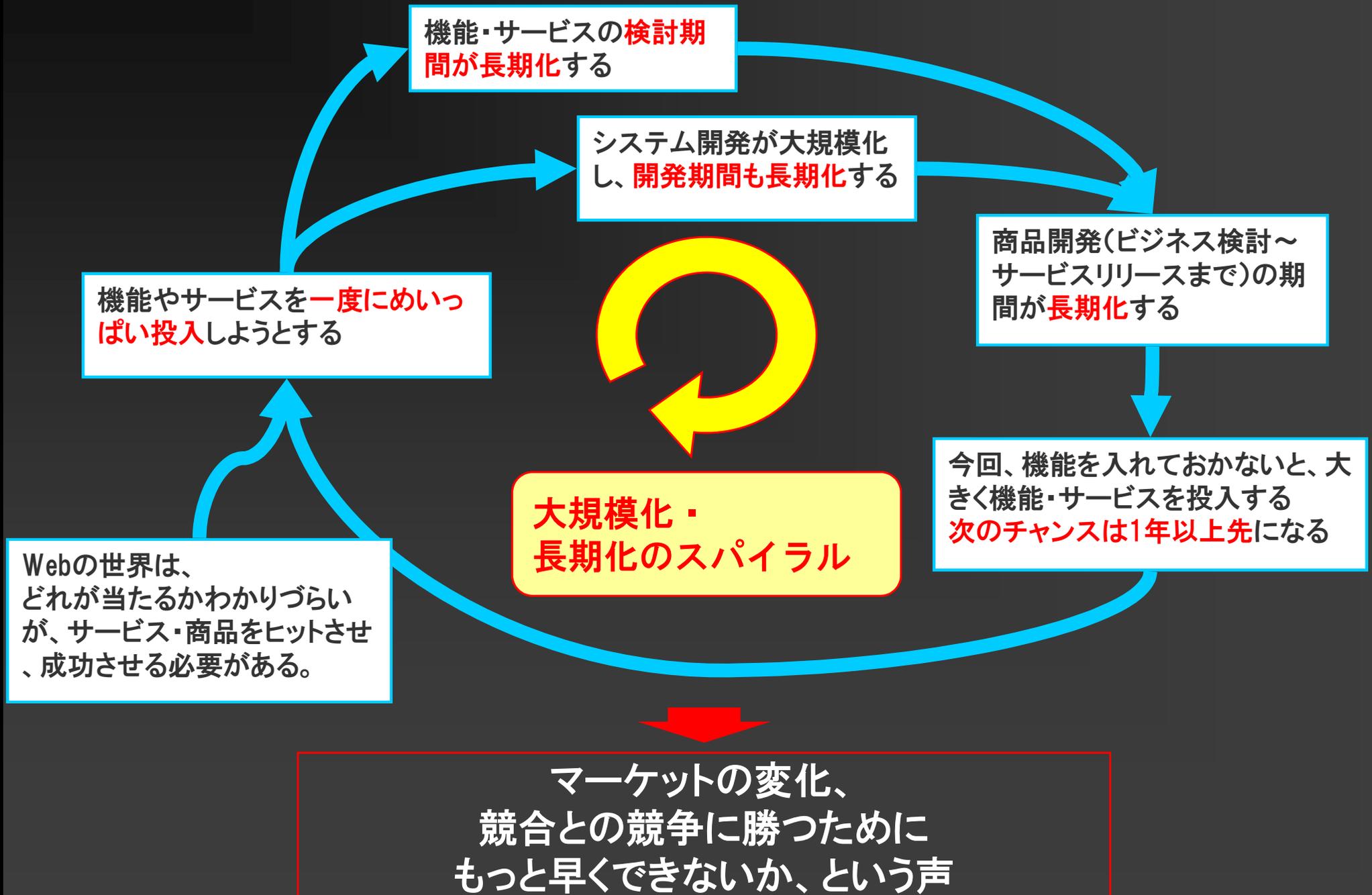
ところが

●変化が激しいWEBの世界では、ビジネスの環境が刻一刻と変わっていく。

『実際に世に出してみないとわからない』というのがWEBの世界。

背景

◆何が起きる？



背景

◆Webメディア・サービス設計・開発をとりまく状況 まとめ

- 変化が激しいWEBの世界では、ビジネスの環境が刻一刻と変わっていく。
『実際に世に出してみないとわからない』というのがWEBの世界。
- リクルートは、“情報誌”のメディア立上げの文化。

背景

- 仮説ベースのためリスク判断が難しい
- 失敗が許されないビジネス根幹のサービス

サービス・広告商品設計

- スタート時に漏れなく機能／要件を洗い出し
- スタート時に最大の効果を求める

システム開発

- 開発着手までに時間がかかるプロセス
- 開発品質追求

メディア／サービスのリリースまで時間がかかる構造

「もっと“スピード”を」の 要求

もっと“スピード”を早くするために

◆もっと“スピード”を早くするために

~~システム開発の、開発方法論や
アーキテクチャーの変革により、
劇的にスピードアップを図る~~



システム開発のみならず、
サービス開発の考え方・やり方自体を変え、
その上でそれに即したシステム開発の工夫
を行う必要がある。

スピードを求めて

◆考え方・やり方を変える

SWAT

Speedy Willing Alliance Team



「一致団結し意思をもってスピードを追求した開発チーム」
(MP、システム部門、開発パートナーが一つのチーム)

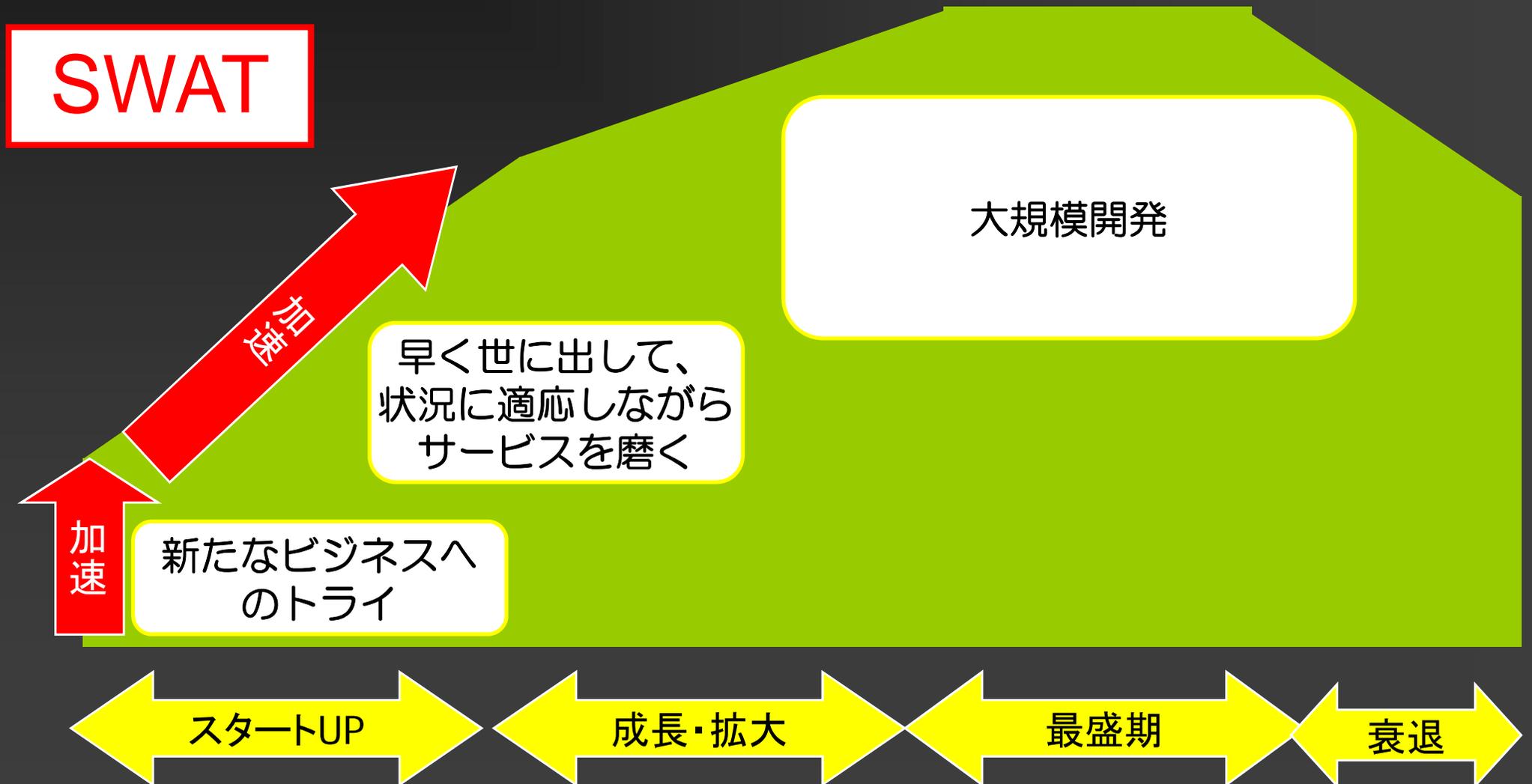
※本家SWAT(Special Weapons and Tactics)の意味も込めています。

新しいメディア/サービスにトライする際の「考え方や意思」を本気で変える

	従来のやり方	新しいやり方 SWAT
ビジネス検討手法を変える	C/O時点で100%の完成度を目指す 1 『中途半端なものは世に出せない』という意識があり、サイト立ち上げ時点で多くを盛り込む。	スモールスタート (1000FP~3000FP) スモールスタートし、カスタマやクライアントの反応を見ながら、段階的にサービスを成長・適合させていく
	ビジネスのコアおよびその周辺も漏れなく検討 2 ビジネスコアに直接リンクしない、よりリッチな機能やユーザビリティ、入稿機能、運用機能なども漏れなく検討し磨き上げていく。	ビジネスのコアに絞り込んで検討 ビジネスのコアに直結しない部分については、最低限の検討パワーに抑える。
	やりたいことを積み上げて予算をとる 3 ある程度時間をかけて積み上げたものに対して、予算とスケジュールを決める。	期間と投資上限をあらかじめ限定 ビジネスの見立てがついた時点で、このプロセスの適用判断を行い、ビジネス検討期間も含めて期間、投資上限を決定する。
	ステークホルダーと調整・確認しながら進める 4 検討の漏れや、混乱や不満が出ないように、関連部署や上長と確認・調整を繰り返していく。	小規模チームに権限委譲 ステークホルダーとの調整は最小限に抑え、少人数の体制で意思決定していく。
	ウォーターフォール型のシステム開発手法 5	独自のシステム開発手法
システム開発手法を変える		

スピードを求めて

◆ビジネス・サービスのライフサイクルイメージ と使い分け



SWATとは？

◆SWATの歴史(約3年)

2006年1月～ 短納期開発ソリューション「Rapid開発」の誕生

Rapid開発(システム開発手法)を考案し、フィジビリティースタディー1号プロジェクト

2007年4月～ RapidからSWATへ昇華

全社展開。

システム開発手法という枠組であった“Rapid”に、考え方・理念を加え、名称もSWATに変更。

2008年10月～ SWATからSWAT2.0へ

Rapid時代から含め、2008年末まで、計25サイトをC/O。

25サイトの経験を元に、全面的改善を実施(社内通称:SWAT2.0)

現在に至る

◆SWATの特徴

基本の特徴:スピード

リクルートで実施している従来型よりスピードが早い。
例として 1500FP規模 で要件定義～リリースまで **約3ヶ月**

特徴1:アジャイル的な開発プロセス

- ①開発工程
設計製造時点での動作確認
- ②タイムボックス
タイムボックスを利用。1週間単位。
- ③朝会、バーンダウンチャート、レトロスペクティブ 等

特徴2:コミュニケーション重視の開発スタイル

- ①敏速なユーザー確認
要件確認会で、MP、システム部門、開発パートナー3者が集まり、
動くアプリで、即座にユーザー確認
- ②コミュニケーション(会議体)のモデルを作成
要件確認会進行、要件確認会のタイミング、頻度などモデルを作成。

特徴3:シンプルな開発ドキュメント

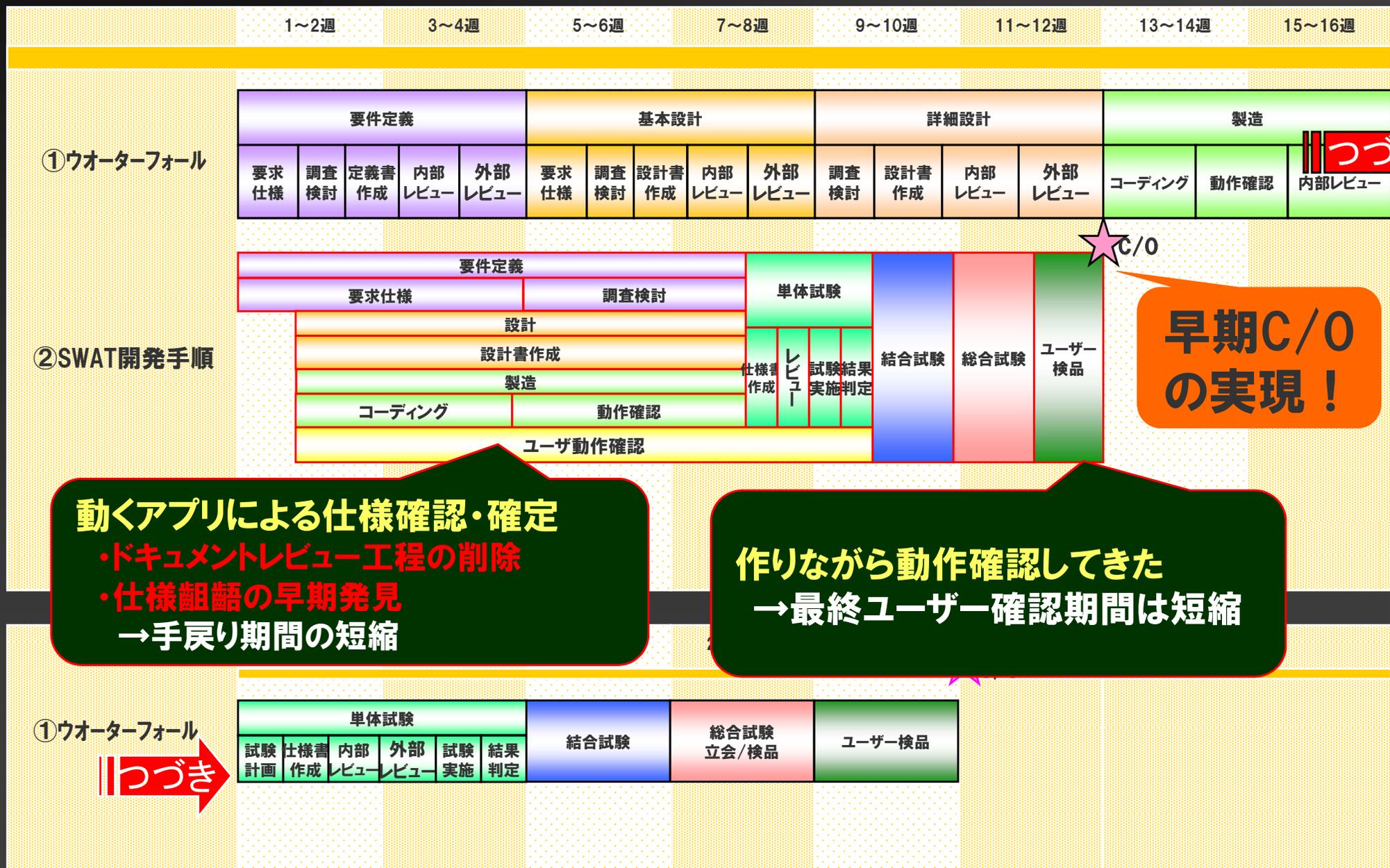
- ①レビューのためのドキュメントを極力減らし、開発者が必要なものだけ
簡易設計書、簡易設計書作成フローなど工程別作成ドキュメントを集約。

特徴1:アジャイル的な開発プロセス

SWATとは？

◆特徴 作業工程の変更による納期圧縮

ウォーターフォール開発とSWAT開発の作業工程の比較(イメージ)



◆特徴 作業工程の変更による納期圧縮



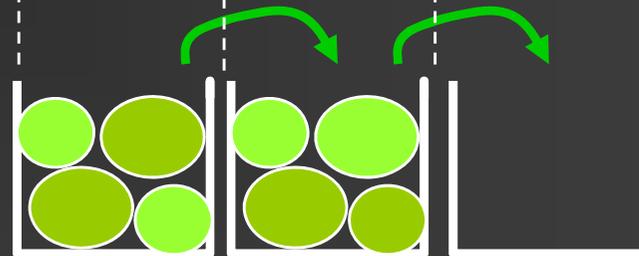
			概要
■ ビジネス検討	PH1		商品・サービス設計、機能検討。
	PH2		細部の検討。システム化範囲の仮決定。
■ 要件定義	PH1		要件定義PH2の準備期間。
	PH2	サイクル1	要件・仕様を詰めていく。
		サイクル2	サイクル1で作成したアプリの動作確認の結果、さらに要件を詳細に詰めていく。
■ 設計・製造・動作確認	サイクル1		画面を正常系として、一通り製造し、動作確認を実施する。 ※仕様確認しながら確定していくので、異常系はまだ、しなくても良い。
	サイクル2		サイクル1で作成したアプリを完成させ、動作確認を実施する。
■ ユーザ動作確認			製造されたアプリを随時、MP、システム部門が動作確認をする。
■ テスト			各種テストを実施。

SWATとは？

◆特徴 タイムボックス(週間サイクル)

		1週目	2週目	3週目	4週目	5週目	6週目	7週目	8週目
				▽ TB計画 チューニング					
			TB1	TB2	TB3	TB4	TB5	TB6	TB7
ビジネス検討 PH1/PH2	要件定義 PH1	要件定義 PH2							
		サイクル1				サイクル2			
		設計・製造・動作確認							
		サイクル1				サイクル2			
TB計画		製造	製造	製造	製造	製造	製造	製造	単体テスト
		単体動確	単体動確	単体動確	単体動確	単体動確	単体動確	単体動確	
		結合動確	結合動確	結合動確	結合動確	結合動確	結合動確	結合動確	
		ユーザ動確	ユーザ動確	ユーザ動確	ユーザ動確	ユーザ動確	ユーザ動確	ユーザ動確	ユーザ動確

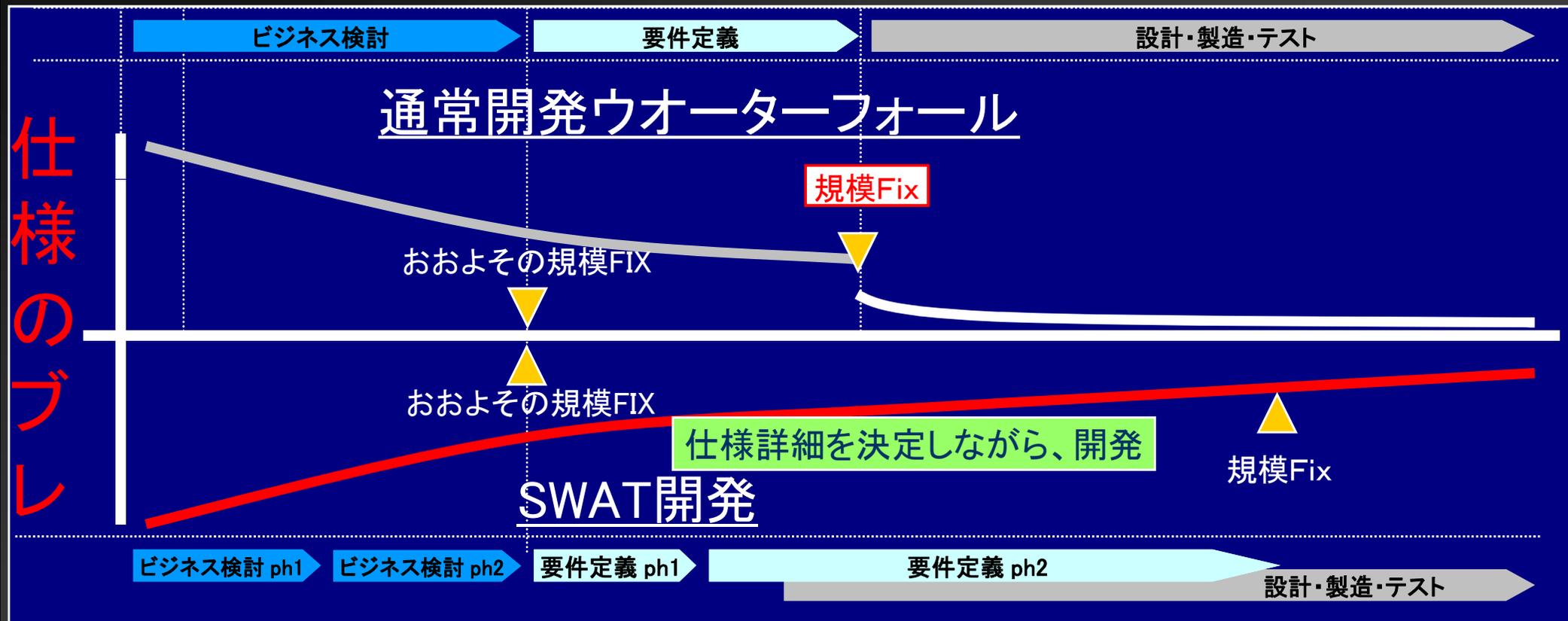
9週目	10週目	11週目	12週目
▽ TB計画 チューニング			C/O
TB8	結合テスト		テストBuffer
Buffer	セキュリティ試験	システム総合テスト	
		性能試験	
		ユーザー総合テスト	
	リリース準備		



計画を後ろ倒ししていくと、バッファとして用意したタイムボックスが埋まっていきます。このタイムボックスの空き具合がバッファ残量を示します。

◆リスクの考え方

・ 予め高い精度で予測する事が難しい。最後まで、開発規模・仕様が確定されない。



リリース日は予め決めた中で、プロジェクトの後半まで、量や難易度に応じて調整していくので、開発規模や仕様の範囲は事前には予測できない。

→リクルートが責任を持つ

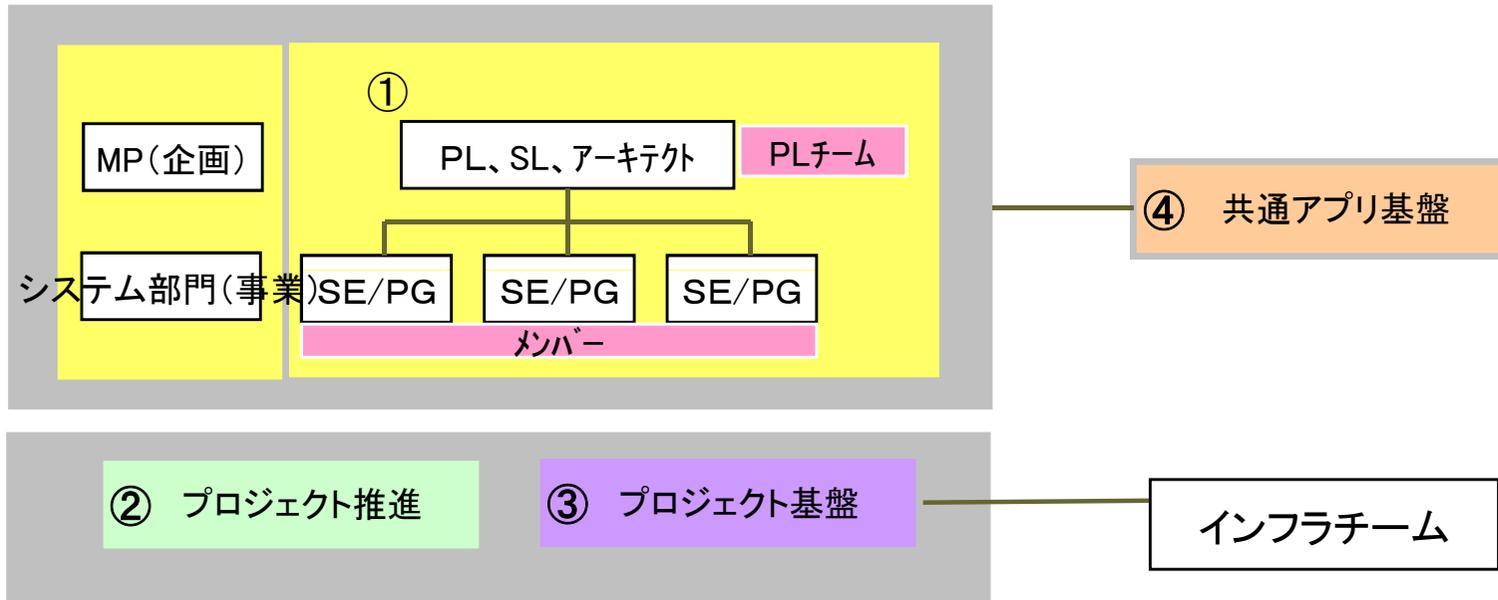
MP(ユーザー部門)、システム部門(システム部門)、開発者の3者が、密なコミュニケーションを取りながら、開発をコントロールすることが、非常に大切。

特徴2:コミュニケーション重視 の開発スタイル

SWATとは？

◆SWAT開発体制

●プロジェクト体制



① アプリ開発チーム

☆PL、SL、アーキテクト、開発メンバー

- ・ アプリ全般を担当。
- ・ 事業側と要件定義、アプリの製造、テスト

② プロジェクト推進

☆SWATスキームの運営

- ・ SWATスキーム・考え方の教育・指導
- ・ 社内各種調整

☆プロジェクトモニタリング

- ・ プロジェクトの各種会議の参加、チェックポイントでの確認会の運営

③ プロジェクト基盤

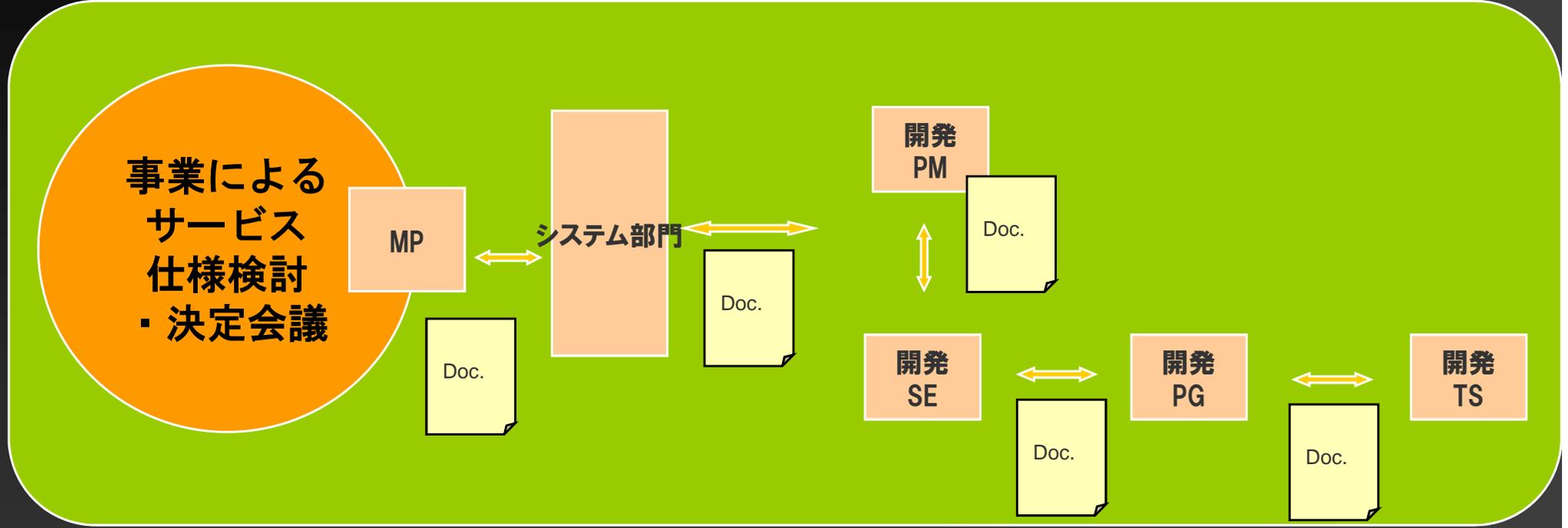
- ・ 環境構築、技術検証、負荷試験サポート等のインフラチームとの窓口
- ・ 高度なロジックやインフラ面も考慮した設計が必要な部分の実装サポート・確認

④ 共通アプリ基盤

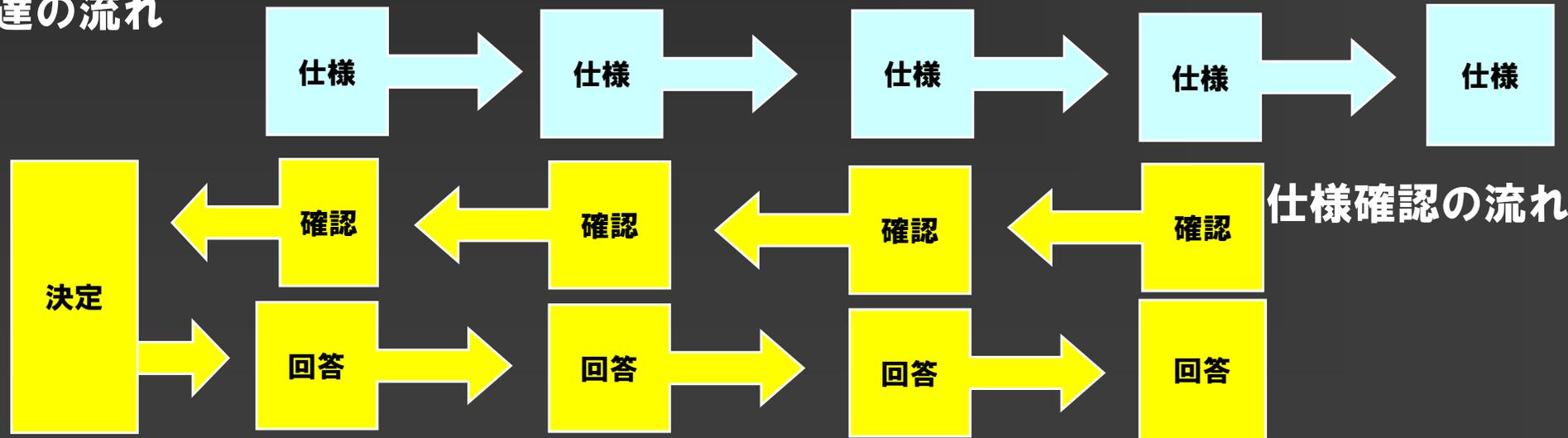
☆共通フレームワークの保守・教育・拡張

SWATとは？

◆要件・仕様の伝達・決定(旧来のやり方のイメージ)



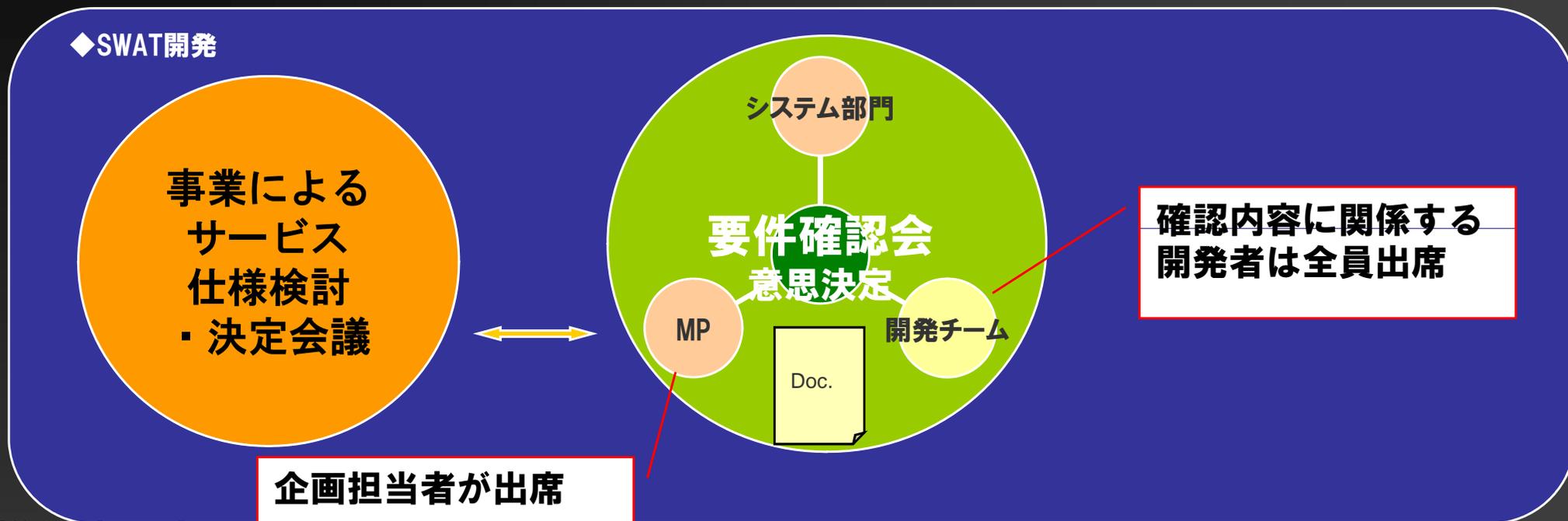
仕様伝達の流れ



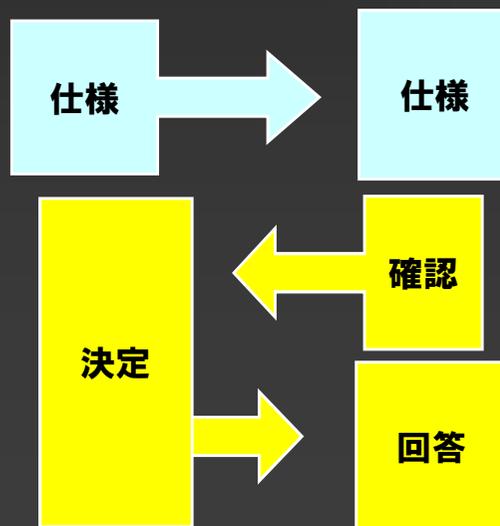
- ・決定・コミュニケーションの遅れが最も、開発現場を遅らせる。
- ・ドキュメントベースのやりとり、が非常に重い

SWATとは？

◆要件・仕様の伝達・決定



仕様伝達の流れ



仕様確認の流れ

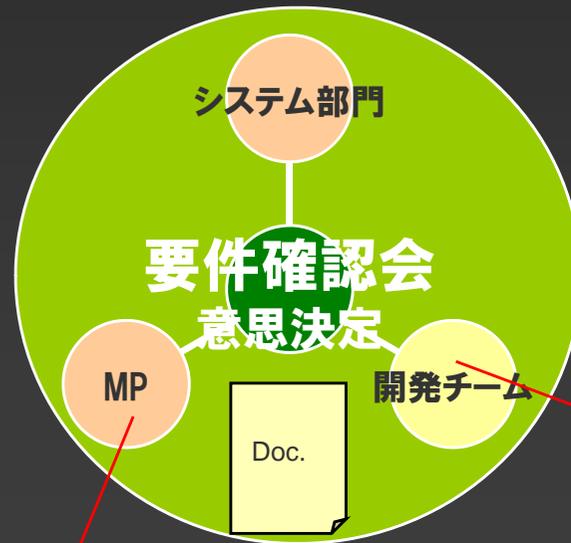
・MP(企画担当者)・システム部門・開発者が対面で要求・仕様を伝達・確認。

SWATとは？

◆要件確認会の内容

■毎週、定例でシステム機能毎の具体的な要件・使用を確認・決定する。

- ・ドキュメントではなく、プロジェクター上にて動くアプリと簡易設計書にまとめた確認事項を映し、仕様確認及びアプリのお披露目を行う。
- ・確認した事項はプロジェクタ上で簡易設計書に随時記述し、確認会参加者で共通認識する



企画担当者が出席

確認内容に関する開発者は全員出席



SWATとは？

◆要件確認会のコミュニケーション原則

- ・ 確認する画面の**実装者自身が発表を行い、確認**を行う。
- ・ 他機能への影響をチェック・共有するため、**周辺機能のメンバーは全員、出席**する。
- ・ 要件確認会以外の場で担当者のみで要件内容を変更すると、仕様齟齬が生じたり、他機能への影響に気づかない可能性があるため、**必ず要件については、要件確認会で決定**する。
- ・ **10分以上議論しても結論が出ない議案は、プロジェクト開発定例へエスカレーション**し、意思決定できるメンバーで結論を出す。
- ・ 事業側、開発側共に**“80%ルール（仮決め）”**により意思決定を素早く行う。

※80%ルールについて

- ・ MP・システム部門・開発パートナーとも80%の精度でコミュニケーションをとり、間違いがあった場合はその都度、三者でフォローするようにする。
 - ・ 判断に必要な情報を100%そろえて判断するのではなく、80%の情報で判断をして前に進む。
- あいまいな状況の中で判断をして前にすすみ、間違えていたらその都度軌道を修正。

SWAT 企画・事業担当者 向け心得(抜粋)

◆ケース 1

一度決めた内容が、難易度が高いことが判明した。

要件定義 ph1

要件定義 ph2

設計・製造・動作確認

✖ BADケース

② えっ！困ります。
できるっていったじゃないで
すか。要件定義書にもこう
書いてありますし。なんとか
頑張ってくださいませんか。

企画部門

システム
担当

① この検索が複雑で、
実装してみると、性能
上、問題がありそうで
す。

開発

③ わっ、わかりました…

まいったなあ、手戻り
も大きいし。次からは
ちゃんと時間をかけて
調査しないと。

これが積み重なると、
お互い、ジャッジが慎重になり、
どんどん効率が落ちていきます。
「持ち帰って検討します…」

SWAT 企画・事業担当者 向け心得(抜粋)

◆ケース 1

一度決めた内容が、難易度が高いことが判明した。

要件定義 ph1

要件定義 ph2

設計・製造・動作確認

企画部門

システム
担当

開発

② そうですか…
それでは別の仕様で目的を
果たす方法を考えましょう
!

① この検索が複雑で、
実装してみると、性能
上、問題がありそうで
す。

★ GOODケース

互いに100%のジャッジ精度を相手に求めたり、変更が効かないと時間がかかる。
80%正しそうならジャッジして先に進めていくのと、ジャッジが間違っていたら、お互い協力して、すぐにリカバリーしていくことが重要。(これが後からの“仕様変更あり”の意味。×「後で決めれば良い」 ○「ジャッジが間違っていたら正しく直せば良いので、素早くジャッジする」 =80%ルール)

SWAT 企画・事業担当者 向け心得(抜粋)

◆ケース 2

要件確認会にて、随時意思決定がなされない。

要件定義 ph1

要件定義 ph2

設計・製造・動作確認

✗ BADケース

企画部門

② う～ん、ここは私では
決められませんね。
次回の会議で聞いてみる
ので、2,3日待ってもらえま
す?

システム
担当

① この機能はこれで
いいですか?

開発

③ わかりました。2,3日なら。
(とりあえず、他の機能を進める
かー。けど、後の工程がパツパ
ツだな。)

これが積み重なると、納期通りにC/O
できない可能性があります。

SWAT 企画・事業担当者 向け心得(抜粋)

◆ケース 2

要件確認会にて、随時意思決定がなされない。

要件定義 ph1

要件定義 ph2

設計・製造・動作確認

企画部門

システム
担当

開発

①この機能はこれでいいですか？

②機能の目的・主旨は営業に確認済です。詳細な仕様は未確認ですが、ほぼ間違いないので、これで進めてください。後で念のため、営業に念押ししておきます。

★ GOODケース

※ SWAT開発では、要件確認会にて随時意思決定する必要があるため、会議前にステークホルダーに判断軸の確認しておきつつ、“80%ルール”で決定していく。

SWAT 企画・事業担当者 向け心得(抜粋)

◆ケース 3

開発中に規模が膨らみすぎていることが判明した。

要件定義 ph1

要件定義 ph2

設計・製造・動作確認

✖ BADケース

企画部門

② それでは機能毎の
工数一覧を出してください。
それで優先順位が付
けて機能を絞ります。

システム
担当

① 機能の絞込みをしないと
納期に間に合わないです
ね。

開発

③ わかりました。
工数一覧を作成します
ね

見積もり作業に手が取られ、
大きくロスしてしまいます。

SWAT 企画・事業担当者 向け心得(抜粋)

◆ケース 3

開発中に規模が膨らみすぎていることが判明した。

要件定義 ph1

要件定義 ph2

設計・製造・動作確認

企画部門

システム
担当

開発

① 機能の絞込みをしないと納期に間に合わないですね。

★ GOODケース

②わかりました。大きなくりで後回しでも良い機能を再検討します。逆に、どこを削れば想定規模に収まりそう、とか予測があれば、教えてください。

※ ウォーターフロー開発では、開発フェーズに入ってからでは、想定しないケース。

※ 開発フェーズ時に、調査や検討に時間をかけない。

SWAT 企画・事業担当者 向け心得(抜粋)

◆ケース 4

開発中に、影響の大きそうな仕様変更要望があがってきた。

要件定義 ph1

要件定義 ph2

設計・製造・動作確認

✖ BADケース

企画部門

① この画面に
〇〇の
機能を付けたい
ですね。

システム
担当

② それをやると、テーブル
に変更が入るので影響が大
きそう。持ち帰って影響範囲
を調査してもらえますか？

開発

③ わかりました。
影響範囲を確認します。

本来であれば、持ち返って影響調査
をするべきシーンかもしれませんが、
これが積み重なると、スピードを大き
く阻害して、調査などの工数もかさみ
ます。

SWAT 企画・事業担当者 向け心得(抜粋)

◆ケース 4

開発中に、影響の大きそうな仕様変更要望があがってきた。

要件定義 ph1

要件定義 ph2

設計・製造・動作確認

企画部門

システム
担当

開発

① この画面に
〇〇の
機能を付けたい
ですね。

②それをやると、テーブルに
変更が入るので影響が大きそ
う。それよりは、こちらの方法
なら、軽く済むと思うので、こ
ちらで行きませんか？

★ GOODケース

※ SWAT開発では要件確認会の場で、このようにリアルタイムで規模コントロールするためのコミュニケーションが求められる。

SWAT まとめ

SWATとは？

◆SWATの開発 まとめ

・80%の意思決定で行う。

◆特徴3

仮決め開発による意志決定のスピード化、納期圧縮

ビジネス検討
(事業戦略立案)

ビジネス検討
(事業要件の具体化)

要件定義
Phase1

要件定義
Phase2

C/O

開発工程完了

◆特徴1

開発パートナーのビジネス検討フェーズ参加

・開発当事者が入る事により、仕様のキャッチアップを早く行える。

設計・製造・動作確認

単体テスト

セキュリティ性能テスト

アプリケーション保守・改善

設計書記述

ユーザ確認(テスト)

全件テスト

業務運用設計
Phase1

業務運用設計
Phase2

運用テスト

◆特徴2

作業工程の変更による納期圧縮
(タイムボックス)

・要件定義をしながら、画面を作成していく。
・製造期間中からユーザー側で動作確認を行う。
・徐々に仕様細部を決定しながら製造していく

SWATとは？

◆実践して

●与えられた期間内に、仕様を決定していき、「決まった事」の量をふやしていくことができているならば、プロジェクトが進むにつれ、先の状況が予測できていくようになっていく。

反対に、課題が発見され続け、タスクの量が増えていくばかりで、「決まった事」が増えていかない、いわゆるバーンアウトな状態になると、当たり前だが、先の状況が予測できず、プロジェクトは火を噴く。

これを防ぐためには、

●事前の計画に力を入れ、不確実性が高すぎる要素がないかどうか、先に先につぶして行くダンドリを組むこと。

●人のコミュニケーション原則が大事。特に80%ルール。人は100%の責任・確実性を求められ、前言撤回を認められないで窮地に追い込まれる経験を重ねると、絶対に、持ち帰って、いろんな情報を集め、いろんな角度から問題ないか検討しようとする、これを各自が始めると、ずるずると遅れ始める。

何か大きなタスクが増えたわけではないので、気づくのが遅れやすい。

SWATとは？

◆実践して

- 特に人の意識を変えるのが大変。担当者の意識を変えるのも骨が折れるし、なかなか、言葉で伝えても腹に落ちず、行動に現れにくいもの。

旧来の、特にウォーターフォールでは正しい行動・言動が裏目に出てしまう。また、担当者がOKだとしても、プロジェクトチーム外に重要なステークホルダーがいる場合、それも複数名、というケースでは、そこを震源地として、プロジェクトが右往左往する可能性も高い。

- いずれにせよ、ユーザー企業側のあり方に大きく左右される。

ご清聴、ありがとうございました。