

開発モデルとは？

 株式会社コンピュータ技研

開発系の工程管理をする上で、基礎となるモデルがいくつか存在します。規模や要件に応じて、モデルごとにメリットとデメリットがあるため、プロジェクトにおいてモデルの選定も大切な要素の一つとなります。

開発モデル

1. ウォータフォール開発モデル

- ・V字モデル
- ・W字モデル

2. スパイラル開発モデル

- ・インクリメンタル・モデル
- ・イテレーティブ・モデル

3. アジャイル開発モデル

- ・スクラム[Scrum]
- ・エクストリーム・プログラム[XP]
- ・機能駆動開発[FDD]

4. プロトタイピング開発モデル

- ・ラビッドプロトタイピング
- ・進化的プロトタイピング

1. ウォーターフォール開発モデル

ほぼ全ての構築・開発の分野で利用されていて、工程を上流から順番に下流まで行う大規模プロジェクト向けのモデル。



メリット

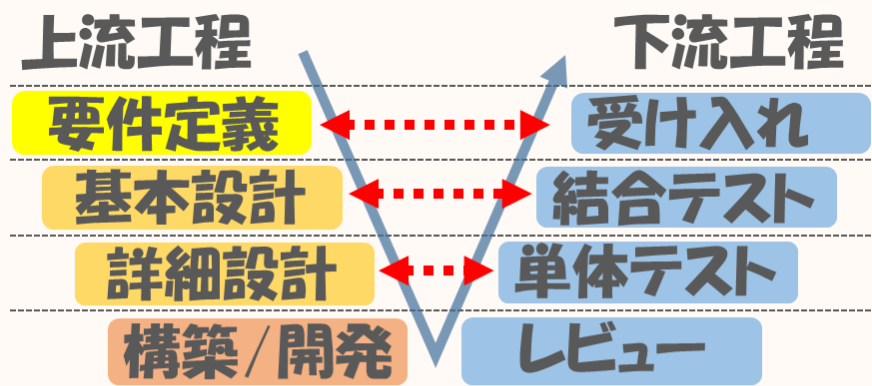
工程ごとに期限を設けて進めていくため、管理(品質・コスト・納期)と確認が容易。

デメリット

手戻りが発生しない前提のモデルのため、後半で戻りが発生したり、途中で仕様変更への対応が難しい。

1. ウォータフォール開発モデル(V字モデル)

ウォータフォール開発モデルにおける上流工程と下流工程を対応させる事により、各工程での過不足を発見しやすくしたモデル。



→ 工程の流れ
⇔ 対応関係

システム開発/構築のみならず、いろいろな案件でベースとして利用されている工程管理モデル。スケジュールや見積もりを算出しやすい。

メリット

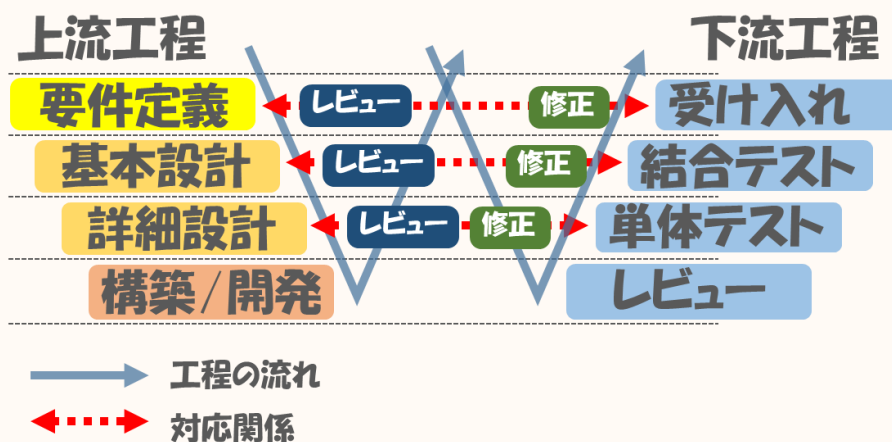
上流工程と下流工程で対応関係が明確になり、工程ごとの対応内容がチェックしやすいため、品質の向上が期待できる。

デメリット

こちらも手戻りが発生しない前提のモデルのため、後半で戻りが発生したり、途中での仕様変更への対応が難しい。

1. ウォータフォール開発モデル(W字モデル)

V字モデルの上流工程からテストエンジニアが関わることによって、要件や設計の矛盾や問題を早期に発見し、手戻りを抑制できるモデル。



要件定義・設計のそれぞれの工程で、テストエンジニアがレビューを行うため、下流工程で発生しやすい問題点や、設計上の矛盾を早期発見しやすくなる。

メリット

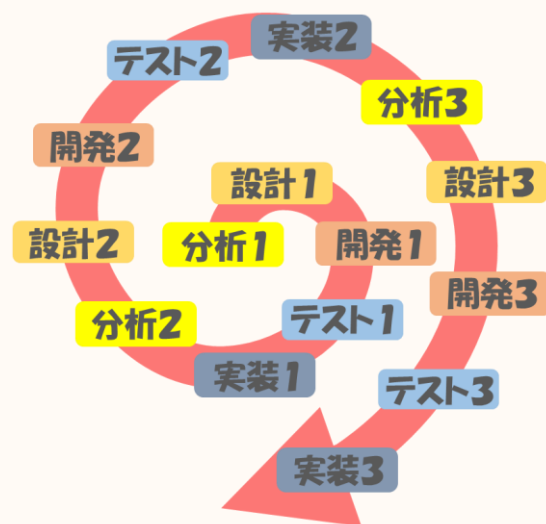
上流工程からテストエンジニアが入り、矛盾や問題点を早期発見することが出来るため、手戻りの抑止になる。

デメリット

要件や設計段階に入るテストエンジニアとして**高い知識と経験持つ人材が必要**。要員がV字より多めになりやすい。

2. スパイラル開発モデル

1つのシステムをいくつかのサブシステム単位でお客様確認とリリースがあり、段階的にシステムの完成度を高めていくモデル。



システム全体の納品が終わるまではサブシステムのリリースの複数回あるため、前回リリースしたサブシステムの改修が飛び込みやすい。

メリット

システム完成までは、仕様変更に柔軟な対応が可能で、お客様もシステムの進捗が確認できるため、満足度も高くなる。

デメリット

プロジェクト開始時にフェーズ数が不明なため、スケジュールの見積もりが難しい。リスクの管理と分析の比重が高くなる。

2. スパイラル開発モデル

インクリメンタル・モデル

大枠はスパイラル開発モデルですが、**独立性の高いモジュール(機能)**で分割し、「設計」→「開発」→「テスト」→「リリース」を行っていきます。

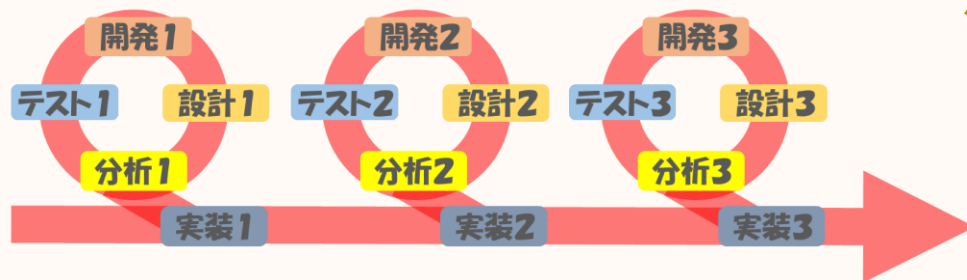
イテレーティブ・モデル

大枠はスパイラル開発モデルですが、**機能の優先度や重要度の高さ**で分割し、「設計」→「開発」→「テスト」→「リリース」を行っていきます。

3. アジャイル開発モデル

ユーザ要求を優先し、仕様変更を前提に開発を進め、小規模なテストを繰り返し、短期間でリリースを目指すモデル。

1サイクルをイテレーションと呼びます。



システムや機能の単位で複数チームを構築して1つのシステムを作ることもあり、上記の図が平行にいくつも並ぶことがある。

スパイラル開発モデルとの違いは、リリースされた後も、要望が満たされるまでは、イテレーション内で改修を繰り返します。

メリット

機能単体で分析～テストを繰り返しているため、リリースまでが短く不具合への対応も早く、工数も少ない。

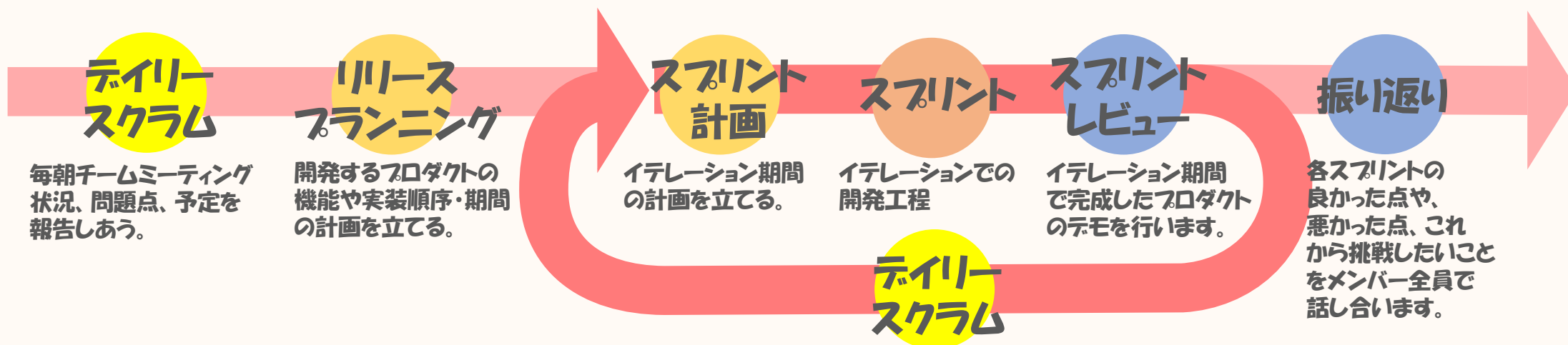
デメリット

プロジェクト全体の納期とは別に、仕様や要件ごとにスケジュールを設定して開発するため、コントロールが難しい。

3. アジャイル開発モデル(スクラム[Scrum])

開発チームの作業とプロダクトに責任を持つ「**プロダクトオーナー**」と、プロジェクトを円滑に進めることに責任を持つ「**スクラムマスター**」がチーム全体を支援し、**開発チーム(3～9人程度)**で開発を行います。

スクラムでは、以下の6つのプロセスを定義、段階的な管理と反復をしながら開発を進めていきます。



3. アジャイル開発モデル(エクストリーム・プログラム[Extreme Programming])

プログラマー中心の開発手法で、常に2人1組でプログラミングを行う。細やかなチェックを行う「**ペアプログラミング**」、チーム内で共通したコード記法などを定める「**コーディング規約**」などの**プラクティス(原則)**のもと、開発を行っていきます。

エクストリーム・プログラミングには以下の5つのポイントがあります。

コミュニケーション

プロジェクトが失敗する原因の多くはコミュニケーション不足です。エクストリームプログラミングでは開発チーム内だけでなく、顧客とのコミュニケーションも重視します。

シンプル

最初の設計を極力シンプルにします。基本的な機能だけを盛り込み、そのほかの機能が必要になれば、その都度対応します。

フィードバック

ソフトウェア開発において、不要な機能を盛り込むのは大きな無駄となります。顧客からのフィードバックを得て、必要な機能を洗い出すことが大切です。

勇気

最初に綿密な計画を立てないという特性上、途中で大胆な変更が求められる場合があります。

尊重

チームで開発する以上、ほかのメンバーを尊重する姿勢は欠かせません。

3. アジャイル開発モデル(機能駆動開発[Feature Driven Development])

ユーザにとっての機能価値(Feature)を重視した開発手法です。
ユーザのビジネスを「見える化」して必要な機能を洗い出し、適切な間隔で反復的にソフトウェアの開発を繰り返すのが特徴です。



4. プロトタイプ開発モデル

要件定義後、プロトタイプを作成し、ユーザー側が確認、要件を満たすまで修正します。承認後に各種設計へ入り、本番の開発工程に入っていくモデル。

上流工程	分析	1. 要求分析 2. 要件定義
		3. プロトタイプ作成
	設計	4. 基本設計/外部設計 5. 詳細設計/内部設計
	開発	6. 構築/開発
下流工程	テスト	7. パラメータチェック/コードレビュー 8. 単体テスト 9. 結合テスト 10. 受入テスト
	実装	11. リリース/納品

メリット

上流工程の段階でユーザーがプロトタイプを確認するため、仕様固めの精度が高くなり、手戻りが発生しにくい。

デメリット

プロトタイプ作成・確認段階で、工数が多くかかってしまう傾向があります。
機能追加のリクエストが発生しやすい。

4. プロトタイピング開発モデル

ラビットプロトタイピング

「時間と費用を最小に抑えること」を最大の目標とし、要求仕様が確定した段階で 開発したプロトタイプは破棄するプロトタイプ型開発の一種です。開発したプロトタイプはシステム開発には流用しません。

進化型プロトタイピング

進化的プロトタイピングは開発したプロトタイプを流用し、改良を重ねながらシステムを完成させるプロトタイプ型開発です。要求仕様を完全に把握する必要はなく、仕様が明確となっている部分から開発に着手します。

どの開発モデルを使えばいいの？

開発モデルごとに得意不得意がありますので、品質やコスト、納期、案件の性質によってふさわしいものを選ぶことが大切です。

システム開発/構築・etc…
(長期・コスト大)

内製化・アプリ開発・商品開発
(中長期・コスト中)

業務改善・効率化ツール
(短期・コスト低)

ウォーターフォール開発モデル

工程ごとに期限を設けて進めていくため、管理(品質・コスト・納期)と確認が容易。

スパイラル開発モデル

全システムの完成までは、仕様変更に対応が可能で、お客様もシステムの進捗が確認できるため、満足度も高くなる。

アジャイル開発モデル

機能単体で設計・計画・実装・開発・テストを繰り返しているため、リリースまでが短く、不具合への対応も早く、工数も少ない。

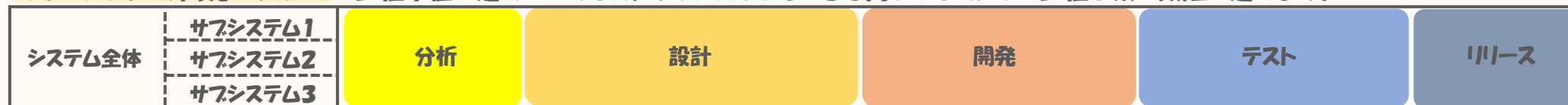
プロトタイピング開発モデル

上流工程の段階でユーザーがプロトタイプを確認するため、仕様固めの精度が高くなり、手戻りが発生しにくい。

モデルごとの作業工程イメージ

モデルごとに、工程の進め方から実装までのサイクルに違いがあります。

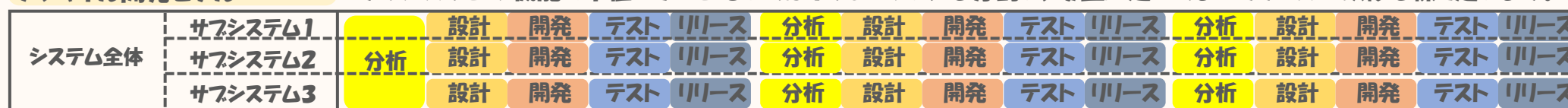
ウォーターフォール開発モデル 工程単位で進めていくため、サブシステム1～3も同じタイミングで工程が次の段階に進みます。



スパイラル開発モデル サブシステム単位で工程を進め、実装が完了した時点で、次のサブシステムの工程に進みます。



アジャイル開発モデル サブシステムや機能の単位でチームもしくはイテレーションを分割し、要望に近づくまでリリース→改修を繰り返します。



プロトタイプ開発モデル 要件定義後はプロトタイプの作成に重点があり、承認されれば工程が次の段階に進みます。



※代表的なモデルの作業工程イメージのため、実際と異なる場合もあります。

最後に

工程管理についての説明は以上になります。

受け持った仕事は全体のドコの工程なんだろうか？とか、これはどのモデルを使って進めているのだろうか？とか、興味をもって「品質・コスト・納期」を守るプロジェクトの一員になられることを願っております。

おつかれさまでした！！