TS COMMUNICATION
TS Communication Co.,Ltd.

プロジェクト管理計画書での「4．開発方式」

２００８年　１２月

ＴＳコミュニケーション株式会社

TS COMMUNICATION
TS Communication Co.,Ltd.

概　要

●プロジェクト管理計画書での「4．開発方式」

開発方式を決めるに当たっては、種々データ根拠がない仮説や錯覚にまどわされないことが、重要だと思います。今回の開発に当たっての条件、制約、ステークホルダーの関与等の情報を収集してデータ根拠を作成して決定していくべきです。
例えば、ある開発方式を用いて成功すると、これがどんなプロジェクトでも「OK」というような勘違いや錯覚をおこすエンジニアは少なくないし、大規模開発での開発が「少人数に適する開発方式」にも当てはまる、と早合点するケースも多いと思います。

開発方式を決めるポイントを次に示します。
①プロジェクトの条件、制約を列挙してまとめ、把握する。
②ステークホルダー関連図をまとめ、テストへの参画等プロジェクト関与を把握する。
③プラットホームの種類、開発ツールを検討する。
④開発規模とプロジェクト体制を検討する。
⑤適応パッケージを検討する。
⑥上記①～⑤を十分検討して、生産性向上施策を検討する。

開発方式は、上記のポイントを十分に検討して選択する必要があります。
ポイントについてまとめていきますので、皆様のご参考になれば幸いです。

TS COMMUNICATION
TS Communication Co.,Ltd.

プロジェクト管理計画書での「4. 開発方式」のポイントとして、概要を次に示します。
＜＜＜「プロジェクト管理計画書」を作成して「プロジェクトの 見える化」を実践しましょう！＞＞＞

４．開発方式

開発手順/技法、支援ツール、生産性向上施策の「見える化」です。

前ページに示した①～⑥のポイントを十分検討して開発方式を決定、そして内容をまとめて、開発方式の 見える化を実践していきましょう。

実は・・・、わたしも失敗は何度も・・、ここでの判断/決定ミスは修復が困難ですよ！

十分に検討しないと、どうなるか。

【例えば・・】

一般論としていえば、大規模開発にはウォーターフォール型が向いていると言われています。本来ウォーターフォール型にすべきところを非ウォーターフォール型にして実行した場合、必ず開発コストは膨れ上がります。
非ウォーターフォール型のＲＡＤやインクリメンタルによる開発だと、後になってシステム全体のインタフェース不整合で、出戻り作業が多発してしまいます。

このようにならないように、プロジェクトの失敗に繋がらないために、十分な検討、識者を含めてのレビュー等にて決定していきましょう！

開発方式を決めるためのイメージを次に示します。

TS COMMUNICATION
TS Communication Co.,Ltd.

ウォーターフォール型

開発手順／技法として最初にモデル化されたものです。汎用機での開発の時代から大きく変わらない手法で、大規模システム開発など多くのプロジェクトで採用されています。開発手順／技法の基本中の基本です。

プロトタイピング

開発に入る前にシステムのプロトタイプ（試作品）を作り、ユーザーの評価をもとに、
これを修正しながら開発を進める方法。
もともとは手法を指す言葉だが、開発工程のタイプととらえることも多い。
「要件定義」や「外部設計」といった上流工程にプロトタイピングを組み込んで、
「プロトタイピングを伴うウォーターフォール型」として推進していくこともできる。
プロトタイピングは、比較的小さなシステム開発において、イタラティブやＲＡＤと
ともに推進していくことも検討できる。

TS COMMUNICATION
TS Communication Co.,Ltd.

スパイラル（ 渦巻型）

もともとは，ウォーターフォールとプロトタイピングの両方を組み合わせた方法。
「計画」や「要件定義」、「外部設計」といった上流工程でプロトタイプを作成し、
有効性やリスクを検証・確認。十分リスクが小さくなったと判断した段階で、
ウォーターフォールに入る。

インクリメンタル（ 増殖型）

システム開発の対象をサブシステムやコンポーネントに分割し，
優先順位の高い部分から順次完成させていく方法。

イタラティブ（ 繰り返し型）

開発対象を小部分に分割し，各部分ごとに複数の開発サイクルを
繰り返して修正や改良を加えながら完成させていく方法。

TS COMMUNICATION
TS Communication Co.,Ltd.

RAD(Rapid Application Development)

短期間・高品質・低コストを目指して提唱された開発方式のことだが、
開発工程の種類としてこの言葉を使う場合もある。
システムの目的や範囲が明確で、リスクの少ない比較的小規模のプロジェクトを対象としており、少数精鋭チームでの開発、エンドユーザーの十分な参画が必須。
小規模な新規開発プロジェクトおよび、既に稼動しているシステムに対して機能追加していく期間内開発を優先させるプロジェクトに向いている。