

# 見積もり概論

2022/04/09

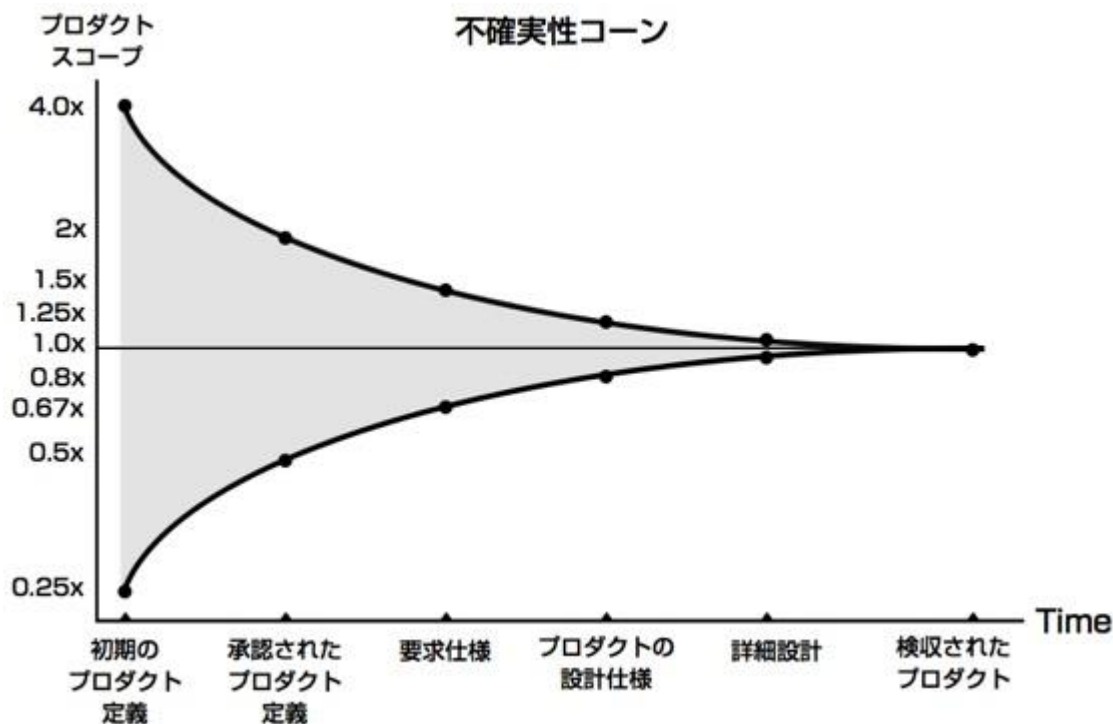
伊東 輝

# 1. 見積もりとは

- ある開発について、どれくらいの工数（人数×期間）で終わるのかを予想する作業
- 計画を立てる上で重要（どのような人をいつ何人入れれば予定した期日に開発が終わるのか、今の人員で作業した場合にいつ開発が終わるのか）
- QCD（品質・コスト・価値提供）を満たせる計画を立てる上で重要な作業

## 2. 見積もり作業の難しさ

- ソフトウェア開発には多かれ少なかれ不確実性がつきもの（過去と全く同じ開発作業はない）
- 前工程であればあるほど誤差が大きくなる
- 最も初期の段階では±4倍の誤差が発生する



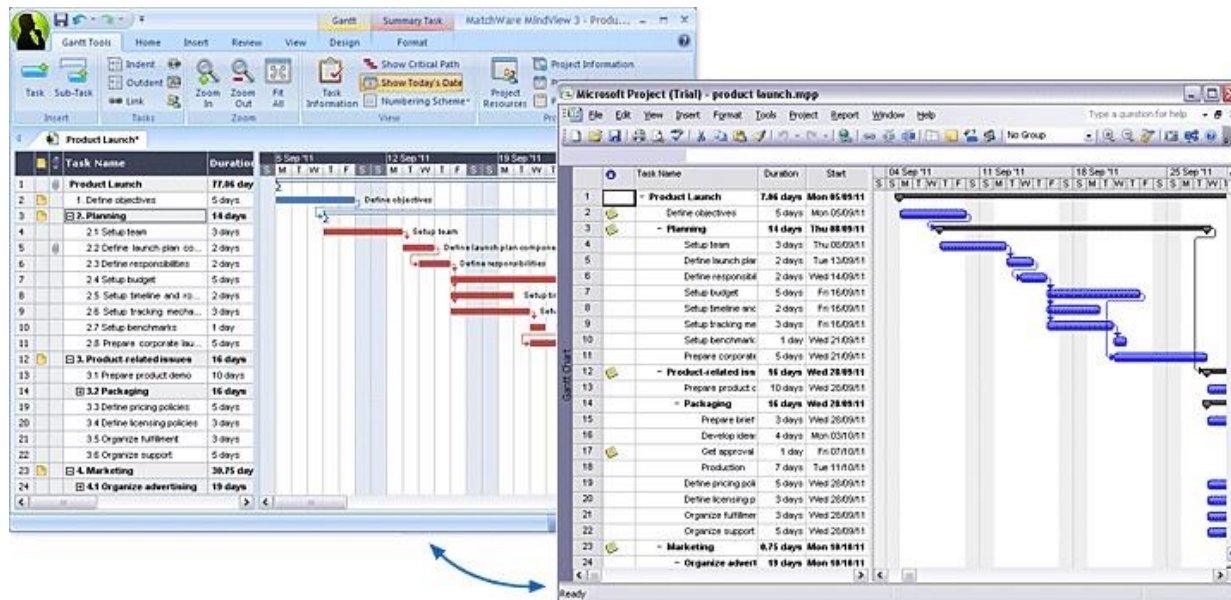
オルタナティブ・ブログ >  
芝本秀徳の『プロジェクトマネジメントの守破離』 >  
プロジェクトの本質とは何か？ プロジェクトマネジメントの基本1  
(<https://blogs.itmedia.co.jp/hideshibamoto/2013/01/post-7a0e.html>)

### 3. 見積もり手法

- 見積もり手法は、「ボトムアップ見積もり」「類推見積もり」「三点見積もり」「直感による見積もり」「パラメトリック見積もり」に大別できる。

### 3. (1) ボトムアップ見積もり (標準見積もり)

- ソフトウェア開発を一つ一つの作業に分解し、一つ一つの作業の工数を見積もり、それを合算する手法
- 最も標準的な見積もり手法だが、作業内容が具体的に見えるまでは精度が低くなる
- 作業単位に分解する際にはWBSが用いられる



MS Project WBS  
(<https://www.msprojectwbs.com/>)

## 3. (2) 類推見積もり

- 類似する案件の工数を参考にして工数を見積もる手法
- 過去に似た案件を実施したことがある場合に有効

### 3. (3) 三点見積もり

- 楽観値、悲観値、最可能値（最も生起確率が高い値）の三点を算出する手法
- 他の手法と組み合わせ、誤差の大きさを表現したい場合に用いる

### 3. (4) 直感による見積もり

- 熟練度が高い複数人の有識者により、直感的に見積もりを行う手法群
- 新規性が高い案件において初期段階から迅速に見積もりを行うことができる反面、正確性は熟練度に依存する



### 3. (4) ①ストーリーポイント

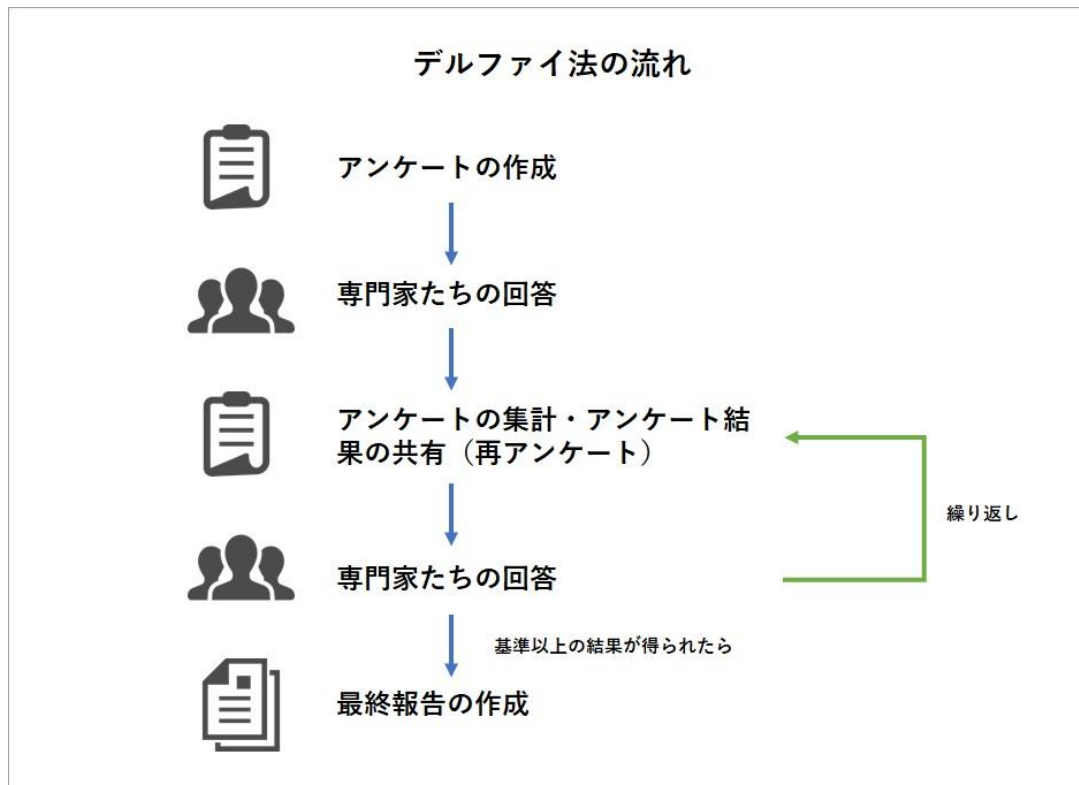
- 基準となる案件に対する見積もり対象の案件の大きさを、各々の有識者が見積もる
- 案件の大きさは原則としてフィボナッチ数列で表現する  
(…1, 2, 3, 5, 8, 13, 21, 34…)
- プランニングポーカーと呼ばれる道具を使って見積もりを行うことも多い



株式会社ハートクエイク >  
チームで優先順位を決めるプランニングポーカーのやり方  
(<https://heart-quake.com/article.php?p=7070>)

### 3. (4) ②Delphi法

- 「多数の専門家に見積もり依頼を出す→見積もり結果の集約→再度見積もり依頼を出す…」を繰り返すことで、見積もり結果の収束を行うという手法



PROMAPEDIA >

デルファイ法とは何か？予測技法の紹介とやり方を解説  
(<https://ssaits.jp/promapedia/method/delphi-method.html#>)

### 3. (5) パラメトリック見積もり

- 統計的な数値を用いて見積もりを行う手法群
- 数値さえ集まれば、客観的な基準による見積もりが可能
- 手法によっては、新規性が高い案件での初期段階でも適用が可能

# 3. (5) ①ファンクションポイント法

- 機能ごとに複雑さに応じたポイントを与えることで、規模を見積もる手法

種類	例	複雑さ		
		低	中	難
外部インスト	入力画面 (低=1~4項目、中=5~9、高=10項目以上)	3	4	6
外部アウトスト	集計や分析、フォーマティングを行った出力画面、振票	4	5	7
内部論理ファイル	データベース (低=1~9項目、中=10~20、高=20項目以上)	4	10	15
外部インターフェイス	オンライン処理など	5	7	10
そのほか	ロジックなど	3	5	8

ITmedia エンタープライズ >  
 見積もり仮想体験の旅へ、いざ出発！ (1-3)  
<https://www.itmedia.co.jp/im/articles/0901/15/news130.html>

機能名称	外部インスト			外部アウトスト			データベース			外部インターフェイス			そのほか			FP
	低	中	難	低	中	難	低	中	難	低	中	難	低	中	難	
端末画面IC情報の読み取り	0	0	1	0	0	0	2	0	1	0	1	0	0	0	1	44
端末設定画面・IC端末設定機能	2	0	1	0	0	0	1	1	0	0	0	0	0	0	1	34
端末設定画面・ネットワーク設定機能	0	2	0	0	0	0	0	0	0	0	0	0	0	1	0	13
端末設定画面・ICデバイス設定機能	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	9
端末設定画面・表示画面・文言設定機能	6	0	0	0	0	0	0	0	0	0	0	0	1	0	0	21
Web管理画面・ID・パスワード認証機能	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	10

### 3. (5) ②ハルステッドモデル

- プログラムステップ数から規模を見積もる手法  
※正確に言うと、プログラムの中に出現する演算子と被演算子の数を数えることで規模を見積もる手法

### 3. (5) ③COCOMO

- プログラムステップ数をベースに、チームの規模や技術力や案件の制約に応じた補正係数をかけ合わせて規模を算出手法
- モデルは工程ごとに3つ存在し、それぞれのモデルは1981年にTRW社のバリー・ボーム氏が63件の案件を分析して作成したものである

※詳細

SE学院 >

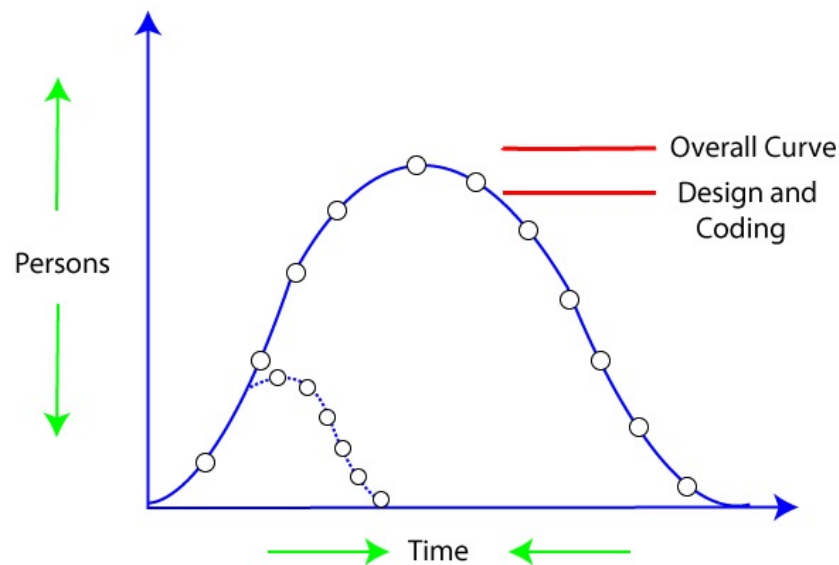
プロジェクト管理 >

COCOMOによるプロジェクト工数の見積もり

(<http://itref.fc2web.com/management/cocomo.html>)

### 3. (5) ④putnamモデル

- 開発工数に関するマンパワー曲線（どの工程でどの程度人数をかけるのかを示した曲線）の概念を適用した手法
- 運用・保守工程を含めたシステムライフサイクル全般をカバーできるのが特徴であり、総工数から各工程の工数を見積もることができる



java T point >  
Putnam Resource Allocation Model  
(<https://www.javatpoint.com/putnam-resource-allocation-model>)

### 3. (5) ⑤Dotyモデル

- 合計数万ステップ程度までの比較的小規模な案件に適した分析手法
- 開発工数をステップ数の指数乗に比例することをベースに推定する
- 開発規模から開発工数を決定できる統計的な関係式を提供するベースライン型モデルと、開発環境の要員も考慮できる多変量型モデルが存在する



## 4. 見積もりを行う際のポイント

- 見積もりやその後の作業計画の失敗からデスマーチを招く典型的なパターンがいくつか存在する
- ここでは、見積もり担当者ができる範囲で、デスマーチ（終わりが見えない高稼働）を避けるための対策を紹介する

## 4. ①不確実性は先んじて排除する

- 要件の明確性や技術面での実現性に問題がある場合は、要件調整や技術検証をなるべく早く行うこと
- これらは不確実性であるため、不確実性を放置したまま見積もりを行うと、誤差が大きくなる
- 誤差は後工程に行けば行くほど小さくなるので、後工程での再見積もりも有効

## 4. ②「現行踏襲」に危機感を感じる

- 「現行踏襲」という要件は存在しない
- 実際には、現行運用の分析や、現行運用と矛盾しない形での対応の検討が必要
- 「現行踏襲」という言葉のイメージに騙されて上記作業の過小見積もりを行う事例が後を絶たない
- 「詳細は過去の〇〇案件参照、と書かれている」「現行の画面と項目が類似している画面イメージが要件として提示される」等も「現行踏襲」と同じ意味

## 4. ③無茶な予定にはなるはやで対策する

- 法令対応や経営上の都合でリリース日が決まっている場合、無茶なスケジュールになりやすい
- 時間が経てば経つほど取れる対策が減っていくので、早めの対策が必要
- 初期段階であれば、スコープ見直し、（育成も考慮した）人員投入、といった教科書通りの対策が取れる
- 案件が走り始めてからは、（標準WBS通りではなくなる形での）作業の組み換え、社内プロセスの一部省略、といった属人的な対策にならざるを得ない
- 放置し続けると体力勝負に…リリース後にも響き得る

ご清聴ありがとうございました。