

# Use Case Point 法による工数見積ワークシート

1999,11/17初版

変更履歴1：技術要因T1とT2を交換（誤記）、環境要因E7:「アルバイト」を「兼任スタッフ」に（2003,3/11改訂1版）  
(株)永和システムマネジメント  
平鍋健児

システム名

このシートは、開発初期の Use Case を利用してプロジェクトの工数を見積もるためのワークシートです。  
主に表計算で算出しますので、色付けされたセルに数字を埋めていくだけで工数を見積もることができます。  
多くのセルにはコメントが付けてありますので、参照しながら入力を進めてください。  
すでに Use Case 図が書かれていれば、30分程度の時間で見積が出せるでしょう。

まず、このページの先頭の「システム名」に今から見積を行うシステム、あるいはプロジェクトの名前を記入してください。

作業手順は次のようになります。

- 1 Use Case 図作成
- 2 Actor の重み付け
- 3 Use Case の重み付け
- 4 UUCPの算出
- 5 技術要因の重み付け ( TCF )
- 6 環境要因の重み付け(EF)
- 7 UCP(Use Case Point)の算出 (UCP=UUCP × TCF × EF)
- 8 工数の算出

1,2,3,4でシステムの機能量を見積もり、5,6 でシステムの技術的な複雑性および環境的な難易度を見積もります。  
7 で Use Case Point を算出します。8 でそれを必要工数に変換します。

もともとオブジェクト指向型開発に適用されるものですが、Use Case という誰にでも理解できる記述から出発するため、構造化設計とウォーターフローによる開発にも十分適用可能と考えられます。

Use Case Point 法は、書籍「Applying Use Cases」の中に紹介された Gustav Karner によるものです。

## 1 Use Case 図作成

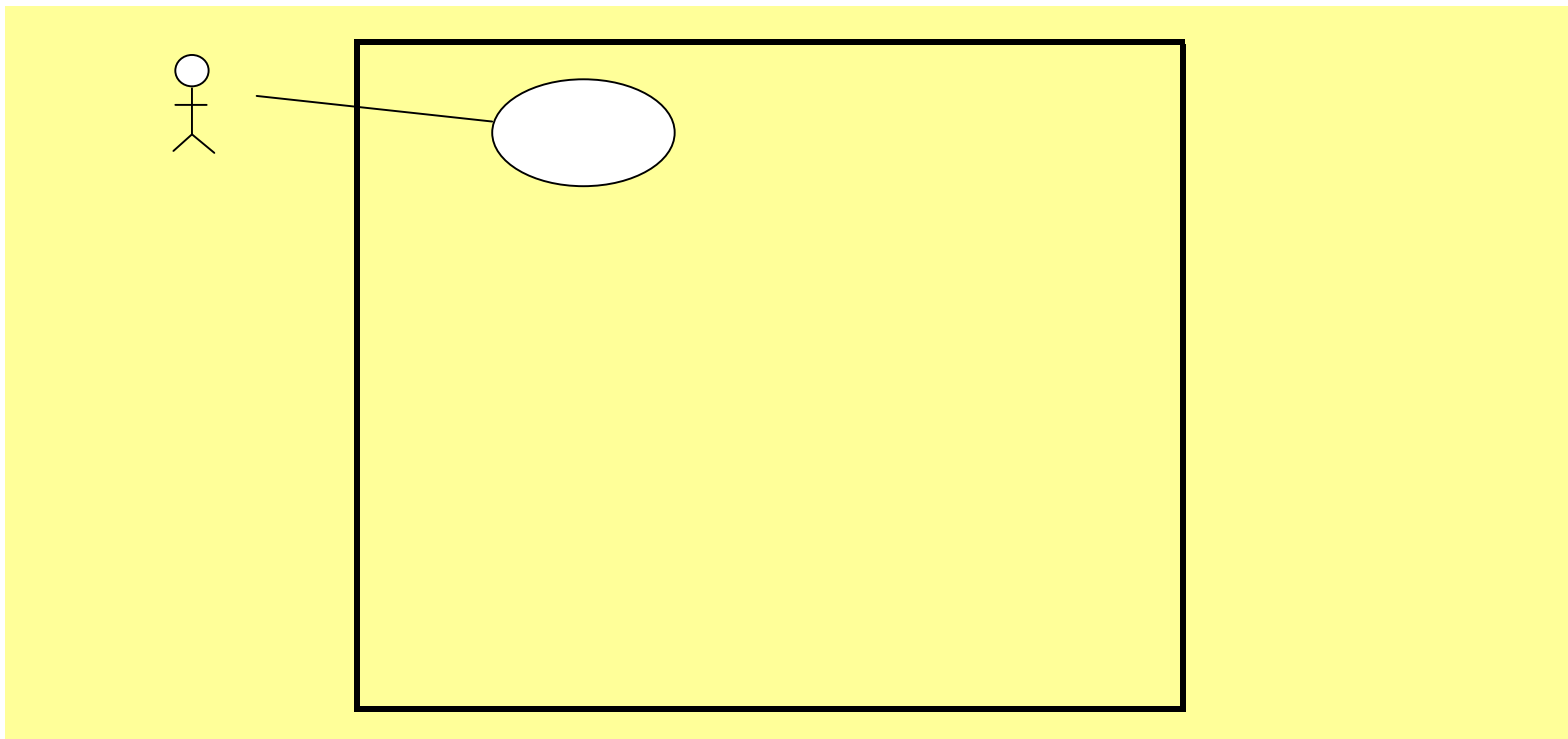
Use Case 図を書き，Actor と Use Case を抽出してください。

詳細なものを書く必要はありません。Actor および Actor と直接関係する Use Case が識別されていればOKです。

人間以外のActorも含めてください。作ろうとするシステムの機能(Use Case)とシステムの外側であって，Use Case とコミュニケーションするもの(Actor)を明確にしてください。

<<extends>> や <<uses>> の Use Case を書いてもよいですが，Actor と直接やり取りしないものは見積には関係しません。

複雑な機能を1 Use Case とするか複数の Use Case とするかは，Actor's Satisfaction すなわち，1つの Use Case が Actor の1つの要求を満足できるかどうかを基準にしてください。また，人 Actor とやり取りするGUI を伴う Use Case では，1画面を1 Use Case と考えてもかまいません。







#### 4 UUCPの算出

Actor と Use Case の重み付けの係数を総計し，UUCP(unadjusted Use Case Point)とします．

$$UUCP = \boxed{\phantom{0000}} \dots = +$$

これをもってシステムの機能量の見積とします．続いて，技術要因および環境要因について見積もります．

#### 5 技術要因の重み付け ( T C F )

以下の技術要因について，0-5の6段階にレベル付けしてください．(0=無関係，5=本質的, 3=平均的)  
コメントがあれば，理由欄に記入してください．

要因番号	意味	重み	係数	値	理由
T1	分散システム	2			
T2	レスポンスやスループットのパフォーマンスが重	1			
T3	オンラインでエンドユーザが効率的に使える	1			
T4	内部処理が複雑	1			
T5	コードが再利用可能である	1			
T6	インストールしやすい	0.5			
T7	使いやすい	0.5			
T8	移植性が高い	2			
T9	変更しやすい	1			
T10	並行処理を行う	1			
T11	セキュリティについての特別な配慮が必要	1			
T12	サードパーティから直接アクセスできる	1			
T13	ユーザの特別なトレーニングが必要	1			
合計	TFactor= 重み × 係数				...

TCF(Technical Complexity Factor)は，以下のように求めます．

$$TFactor = \text{重み} \times \text{係数} = \boxed{\phantom{000}} \dots$$

$$TCF = 0.6 + (0.01 \times TFactor) = \boxed{\phantom{000}} \dots$$

## 6 環境要因の重み付け(EF)

以下の環境要因について，0-6の5段階にレベル付けしてください．

- E1～E4 0=経験無し，5=エキスパート，3=平均的  
 E5 0=モチベーションなし，5=高いモチベーション  
 E6 0=極端に不安定，5=不変，3=平均  
 E7 0=全員プロジェクト専任スタッフ，5=全員他のプロジェクトと兼任  
 E8 0=非常に簡単な言語，5=非常に難しい言語，3=平均的

コメントがあれば，理由欄に記入してください．

要因番号	意味	重み	係数	値	理由
E1	採用するプロセス(工程)に慣れている(*1)	1.5			
E2	アプリケーション開発の経験がある	0.5			
E3	採用する方法論に慣れている(*2)	1			
E4	リーダーの能力	0.5			
E5	モチベーション	1			
E6	仕様の安定性	2			
E7	メンバに外注，アルバイトが含まれる(*3)	-1			
E8	プログラミング言語が難しい(*4)	-1			
合計	EFactor= 重み×係数				...

EF(Environmental Factor)は，以下のように求めます．

$$\text{EFactor} = \text{重み} \times \text{係数} = \boxed{\phantom{000}} \dots$$

$$\text{EF} = 1.4 + (-0.03 \times \text{EFactor}) = \boxed{\phantom{000}} \dots$$

(\*1)～(\*3)は，Karner のオリジナル表現を若干変更していますので注意してください．

(\*1) オリジナルは，「Ratinoal Unified Processに慣れている」

(\*2) オリジナルは，「オブジェクト指向に慣れている」

(\*3) オリジナルは，「Parttime worker が含まれるか」

(\*4) どのプログラミング言語にどの値を割り当てるかは，オリジナルにも記述がない  
 個人的には，2=Basic/COBOL, 3=C/Pascal(Delphi)/Smalltalk/Java/Eiffel, 4=C++

7 UCP(Use Case Point)の算出

最後に、これまで計算した、UUCP, TCF, EF から、UCP を求めます。

UCP = UUCP x TCF x EF = ...

8 工数の算出

平均的には、1 UCP を20人時間で見積もります。

工数 = UCP x 20 人時間 =  人時間
1日 8 時間とした場合 =  人日
1ヶ月160時間とした場合 =  人月

しかし、安全のために以下の数値を確かめてください。

危険指数 = E1-E6で2以下の要因数とE7-E8で4以上の要因数の和 =  ...

が3または4の場合、1 UCP を 28人時間で見積もります。

工数 = UCP x 28人時間 =  人時間
1日 8 時間とした場合 =  人日
1ヶ月160時間とした場合 =  人月

また、 が5を超えた場合、リスクが非常に高く成功が難しいプロジェクトです。
範囲内に納まるように、プロジェクトの環境要因を再調整してください。

から、このプロジェクト

以上で終わりです。「見積結果シート」に、簡潔に結果のみをまとめています。