

品質要求を記述したユースケースによるソフトウェア開発プロジェクトの進捗把握手法

西野文雄 東基衛

早稲田大学大学院 理工学研究科 経営システム工学専攻

1 はじめに

プロジェクトマネジメントの知識体系(PMBOK: A Guide to the Project Management Body of Knowledge)では利用者(顧客, ユーザと同義とする)に対する進捗報告の重要性が述べられており, 利用者の視点を取り入れたソフトウェア開発の進捗把握手法の実現が求められている. 利用者の視点からの進捗把握としてユースケースを用いた手法がある. ユースケースは機能要求を記述したものであるため, ユースケースによる進捗把握では品質要求の進捗を把握することは難しい. 本研究では,

- (a) 利用者の視点から進捗を把握する
- (b) 品質要求の進捗を把握する

上記2つを目的とし, ISO/IEC 9126(JIS X 0129)のソフトウェア製品の品質モデルに基づいた品質要求をユースケースに記述し, そのユースケースを使用して機能要求と品質要求の進捗把握を行う手法を提案する.

2 提案手法の概要

利用者の視点の進捗把握及び品質要求の進捗把握を実現するために以下のアプローチを採る.

- (a) ユースケース単位で進捗を把握する  
 利用者の視点から進捗を把握するために, 機能要求の進捗をユースケース単位で把握する. そのために, ユースケースとクラスの関連付けを行い, クラスの進捗度からユースケースの進捗度を算出する.
- (b) 品質要求を詳細レベルの機能へ落とし込み, 詳細レベルのユースケースとして扱う  
 品質要求はゴール指向分析によって, 詳細レベル機能に落としこみ, 詳細レベルの機能に関するユースケースを作成する. 詳細レベルのユースケースの進捗度から品質要求の進捗度を算出する.
- (c) 品質要求をアクティビティへ展開する  
 品質要求の中には, 詳細レベルの機能にならないものもある. たとえば, 保守性は, 詳細設計において, プログラムの複雑性を減らす洗練されたコーディング技法によるアクティビティなどによって実現される. このような品質要求の進捗度は, クラスのアクティビティの進捗度から把握する.

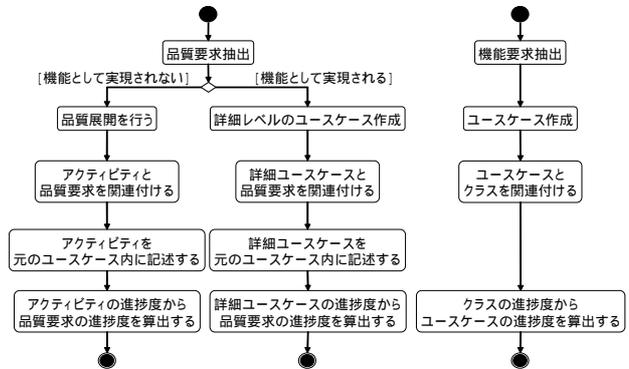


図 1 提案手法の概要図

提案手法の概要を図 1 に示す. 本研究では機能要求はユースケースに記述し, 品質要求は抽出後にユースケースのタグ付き値の特殊要件としてユースケース内に記述する. 図 2 に本研究で扱うユースケースの例を示す.

「現金を引き出す」ユースケース 1 「銀行顧客」は, 自分が誰であることを証明する 2 システムは, 銀行顧客の認証を行う 3 「銀行顧客」は, 現金を引き出す口座を選択し, その金額を指定する 4 システムは, 金額を口座から引きおろし, 現金を払い出す
<b>品質要求</b> 理解性「現金引き出しの操作は, 利用者の労力に影響を与えてはならない」

図 2 本研究で扱うユースケースの例

3 機能要求の進捗把握

本研究では, ユースケースに基づいて進捗把握を行うため, 機能要求と関係性が高いユースケースポイント(UCP: Use Case Point)法を利用する. また, 工数を見積もるために, Karner の UCP 法を基にして Schneider が提案した UCP から工数を見積もる手法を適用する.

UCP 法によって算出された工数を, COCOMO (COConstructive COst Model)<sup>[2]</sup>の工数配分の推奨比率をもとにして, 各フェーズへ配分する. また, 品質要求を考慮した工数を見積もるために, COCOMO の工数補正因子を参考にして求めた品質要求の工数補正因子を乗じて工数を補正する.

以下にユースケース  $i$  とクラス  $j$  の関連度  $R_{ij}$  を求め, クラスの進捗度からユースケースの進捗度を求める手順を示す.

- (1) ユースケース  $i$  の機能規模  $UCP_i$  を測定する

A Methodology for Software Project Progress Visualization by Using Use Case with Quality Requirements  
 Fumio NISHINO, Motoei AZUMA  
 Dept. of Industrial & Management Systems Eng.,  
 Graduate School of Science & Eng., WASEDA Univ.

(2) ユースケース  $i$  が各クラスを呼び出す総数  $\sum_j C_{ij}$  を測定する

(3) ユースケース  $i$  がクラス  $j$  を呼び出す数  $C_{ij}$  を測定する

(4) ユースケース  $i$  でのクラス  $j$  の規模  $UCPcl_{ij}$  を求める

$$UCPcl_{ij} = UCP_i \times C_{ij} / \sum_j C_{ij}$$

(5) クラス  $j$  の規模  $UCPcl_j$  を求める

$$UCPcl_j = \sum_i UCPcl_{ij}$$

以上によって、ユースケースの規模からクラス単位の規模へ変換することができた。クラスの進捗度からユースケースの進捗度を表すために、ユースケース  $i$  とクラス  $j$  の関連度  $R_{ij}$  を求める

(6) ユースケース  $i$  とクラス  $j$  との関連度  $R_{ij}$  を求める

$$R_{ij} = UCPcl_{ij} / UCPcl_j$$

(7) クラスの進捗度が  $P_j$  のとき、ユースケース  $i$  の進捗度  $P_i$  を求める

$$P_i = P_j \times R_{ij}$$

#### 4 詳細な機能になる品質要求の進捗把握

Cysneiros はゴール指向分析で抽出した品質要求をユースケースへ統合する手法を提案した<sup>[3]</sup>。本研究では、Cysneiros の手法を利用してユースケースに記述された品質要求を詳細レベルの機能に落としこみ、その機能に関するユースケースを作成する。図 3 にその例を示す。詳細レベルのユースケース作成後は機能要求と同様に進捗を把握する。

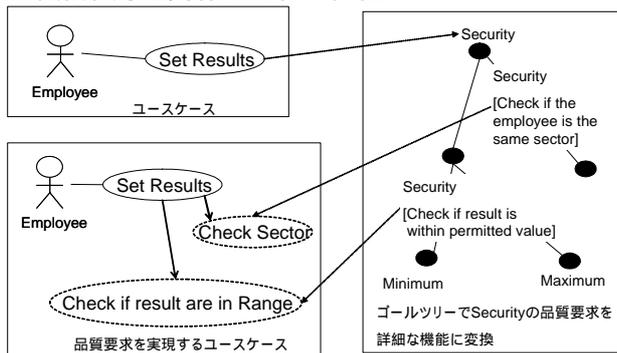


図 3 詳細なユースケースになる例

#### 5 アクティビティになる品質要求の進捗把握

櫻井らは、文献[4]において、ISO/IEC9126 のソフトウェア品質特性と、それを実現するためのアクティビティとの関連度を示している。本研究ではこの図 4 を用いて品質要求をアクティビティへ展開する。

アクティビティになる品質要求の工数に関しては、COCOMO の工数配分の推奨比率で各フェーズに

配分された工数から、各アクティビティの実績データを基にして算出する。

要求品質	工程 開発特性 分割要求品質	基本設計			詳細設計			
		構造化設計	機能テスト設計 充実化	メタ形式標準化 データ形式標準化 ユーザー標準化	用語の標準化 図表記載充実化 カスタマイズ追及化 資源仕様適正化	デザインレビュー 充実化	部品再利用 構造化設計	プログラムモジュール 単独立化 構造化設計 充実化
含目的性		2	4	3	3	7	2	7
正確性		3		5	5	7	3	7
標準適合性							2	
セキュリティ						2		2
機能性標準適合性				5		2		2
解析性					3	4		3
保守性						7		2
変更性						7		3
安定性								2
試験性				4			3	4
保守性標準適合性								4

図 4 品質要求とアクティビティとの関係(文献[3]より一部引用)

クラスのアクティビティの進捗度からユースケースの品質要求の進捗度を算出するために、クラスのアクティビティとユースケースの品質要求との関連度を求める。関連度の求め方、及びアクティビティから品質要求の進捗度を求める方法を以下に述べる。

(1) 各品質副特性の重要度及び要求水準に基づいて、ユースケースの品質要求  $k$  に重み  $QW_k$  を与える。

(2) 図 4 に示した品質要求  $k$ 、アクティビティ  $a$  との関連度  $W_{ak}$  を用いて、クラスのアクティビティ  $a$  とユースケースの品質要求  $k$  との関連度  $RA_{ak}$  を求める

$$RA_{ak} = QW_k W_{ak} / \sum_k QW_k W_{ak}$$

(3) クラスのアクティビティの進捗度が  $P_a$  のとき、ユースケースの品質要求  $k$  の進捗度  $P_k$  を求める

$$P_k = P_a \times RA_{ak}$$

#### 6 おわりに

本研究では、ISO/IEC 9126 の品質特性に関する品質要求を記述したユースケースを利用してソフトウェア開発プロジェクトの進捗把握を行う手法を提案した。今後の課題として、品質要求からアクティビティへ変換されるのか、機能に変換されるか、の条件分岐を系統的に定めることが挙げられる。

#### 参考文献

[1] Geri Schneider, J. P. Winters, 羽生田栄一[訳]: ユースケースの適用: 実践ガイド, ピアソン・エデュケーション (2000)  
 [2] B. W. Boehm, et al: SOFTWARE COST ESTIMATION WITH COCOMO, Prentice Hall PTR (2000)  
 [3] L.M. Cysneiros: Nonfunctional Requirements: From Elicitation to Conceptual Models, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, Vol. 30, No. 5, May 2004  
 [4] 櫻井通晴, 村上敬亮: ソフトウェア価格決定の理論と実務, 中央経済社 (1998)