

要件定義工程の進捗管理における構造化率の提案

A proposal of structured rate for requirements definition process in progress management

海野拳市[†] 太田修平[‡] 石井信明[‡]
Kenichi Umino[†] Shuhei Ota[‡] Nobuaki Ishii[‡]

[†] 神奈川大学院 工学専攻 [‡] 神奈川大学 工学部

[†] Graduate School of Engineering, Kanagawa University. [‡] Faculty of Engineering, Kanagawa University.

要旨

情報システム開発において要件定義は重要とされている。しかし現在要件定義を管理する手法として体系的なものではなく、担当者の経験に基づいて行われることが多い。本発表では、進捗管理にソフトウェアのバグ検出を予測する信頼度成長曲線を応用した構造化率を用い、要件定義工程の計画立案方法を提案する。

1. はじめに

情報システム開発において、工程の遅延、コスト増加、品質上の不具合がしばしば生じている。JUASの調査[1]によると、工期遅延理由の50%以上が要件定義の問題となっている。

情報システム開発は、要件定義、設計、プログラミング、テスト、運用・保守の流れで実施される。この中の要件定義は、ステークホルダーの曖昧な要求を抽出し、システムに必要な要件を整理し、実装可能な仕様に落とし込む重要な段階である。

設計以降の段階の進捗管理には、プロジェクトマネジメント手法が利用されている。しかし、現状のプロジェクトマネジメント手法は、作業が固まった後に有効な手法といえる。そのため、作業内容が不確定な要件定義段階の進捗管理に適用するには課題があり、新たな手法が必要である。

本研究では、要件定義段階の進捗管理手法の提案を目的とする。プロジェクトマネジメントにおける進捗管理手法のEVM(Earned Value Management)、要求工学の要件獲得手法、信頼性工学におけるソフトウェアテストの信頼度成長曲線などを活用し、要件定義における進捗管理手法の開発を目指す。

2. 先行研究

プロジェクト管理技術であるプロジェクトマネジメントには、EVMという進捗管理手法がある。EVMを活用するには作業内容が明らかである必要があり、作業内容、時間などを予め予測するのが難しい要件定義に適用するには工夫が必要となる。また要求工学では、要件定義を要求管理と要求開発に分けて行う方法[2]が示されているが、要件定義の進捗管理管は対象としていない[2]。信頼性工学では、ソフトウェアテストにおいて信頼度成長曲線[3][4]という、テストの進捗度合いに応じたバグの検出数を予測する手法があり、要件定義の要件抽出予測に応用可能と考えられる。

さらに要件定義における進捗管理の研究としてIshii, Muraki[5]は、ステークホルダー間のコミュニケーション量に着目した方法を提案している。要件の明確さを要件構造化率S, また要件管理工数をMMH, 要件定義開発工数をTMHという指標で表し、人的資源の配置管理方法を導く。しかし提案手法は、要件定義開始タイミングのアルゴリズムが厳密ではない点、構造化率Sが要件の収束状況を正しく反映できていないなどの課題がある。

3. 要件定義工程の進捗管理

図1に要件定義段階の進捗管理の概要を示す。要件定義段階は、計画、分析、要件定義活動、業績監視のフェーズに分かれており、これらにより目標の達成に向けた管理が行われる。これらの活動は一方方向に行われるのではなく、適宜修正が行われる。ここで計画とは、目標や進捗に応じて、要件定義作業

の計画を行うフェーズである。分析は、計画、要件定義進捗監視の結果から、進捗状態の分析を行うフェーズである。

なお本発表は、図1の計画フェーズを対象としている。すなわち、要件定義の進捗を表す要件構造化率の算出に信頼度成長曲線を応用し、要件定義の計画のモデル化、最適管理手法を探求する。

要件定義の特徴は、作業の特性から、図2のようにコミュニケーション重視の活動と文書化重視の活動の2つに分けることができる[2]。

文書化重視の活動は、1方向に行われるのではなく、繰り返し行われる。これらの活動は、単純ではないが、ある程度形式化が可能である。一方コミュニケーション重視の活動は、基本的にステークホルダー間のコミュニケーションに依存しており、計画・管理が難しい。本研究では後者のコミュニケーション重視の活動を対象とし、その進捗を管理する手法を提案する。

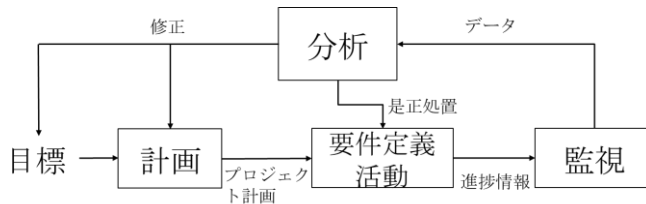


図1. 要件定義段階の進捗管理

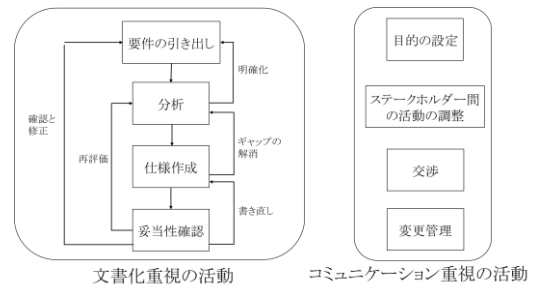


図2 要件定義の作業構造

3.1. 要件定義進捗モデルの検討

本研究では、モデルを、Ishii, Muraki[5]の研究で用いた指標、バグの検出を予測する信頼度成長曲線を組み合わせて作成する。

以下にモデルで使用する指標について説明する。

- ・ 要件管理工数(MMH)： 要件定義に関わる要件の交渉・調整・共有化、変更管理などの工数。
- ・ 要件定義開発工数(TMH)： 要件獲得、要件定義書の作成作業の工数。
- ・ 構造化率 S: 要件の明確さ、ステークホルダー間の共有化の進展を表す指標として用いる。Sが1の場合は、要件定義終了状態。
- ・ CT: 要件管理においてステークホルダー間の必要なコミュニケーションの時間。この時間が減少することは要件定義が進んだと想定。

ここで第*i*期の要件*j*ステークホルダー*k*の MMH, TMH は、(1), (2)式で求める。

$$MMH_{j,k}^{(i)} = CT_j \times r \times m_j \quad (0 < r < 1) \quad (1)$$

$$TMH_{j,k}^{(i)} = (1 - MMH_{j,k}^{(i)}) \times R_Ratio_{j,k}^{(i)} \quad (2)$$

(1)式の m_j は、要件ごとのステークホルダーの人数の合計を指す。 r は、コミュニケーション量が期を進むごとに減ることを表している。(2)式の $R_Ratio_{j,k}^{(i)}$ は、要件ごとの仕事量の配分を指す。配分の合計は1になる。また TMH の要件*j*の*i*期までの累積工数を $TTMH_j^i$ とする。 i は、第*i*期を示す。実験では、(3)式、(4)式内の a を1、 b を1としている。また r は、0.9としている。

次に構造化率 S を、(3), (4)式に示す。信頼度成長曲線[3][4]の Goel-Okumoto モデルと、遅延 S 字モデルを利用する。これらのモデルを用いる根拠として、要件の抽出とバグの検出は、どちらも初期に多く検出され、最終的に収束するという類似性がみられることである。

なおモデルの特徴として、Goel-Okumoto モデルは放物線を描き、その後収束する形をとる。対して遅延 S 字モデルは、ある期間までは緩やかに推移し、その後大きく伸び、最後は収束するという S 字型を描く。Goel-Okumoto モデルでは、バグを見つける工程を1つと仮定している。一方遅延 S 字モデルでは、バグの発見、修正の2段階の工程を経ると仮定している[4]。これらを要件定義に当てはめると要件定義

を 1 つの工程と見るならば Goel-Okumoto モデル, 要件定義工程を要件の抽出, 整理の 2 段階を経るとするならば遅延 S 字モデルを用いることになる。

またステークホルダーごとの S は, ステークホルダーがかかわる要件ごとの S の平均値とし, (5) 式で求める。ここで p は要件数を表しており, また l_k は, ステークホルダーごとの要件数を表している。

Goel-Okumoto モデル:

$$S_{j,k}^{(i)} = a \left(1 - e^{-b TTMH_{j,k}^{(i)}} \right) \quad (3)$$

遅延 S 字モデル:

$$S_{j,k}^{(i)} = a \left(1 - (1 + b TTMH) e^{-b TTMH_{j,k}^{(i)}} \right) \quad (4)$$

$$S_k^{(i)} = \frac{\sum_{j=1,p} S_{j,k}^{(i)}}{l_k} \quad (5)$$

3.2. 数値計算例

単純な要件定義のケースを用いて, $R_Ratio_{j,k}^{(i)}$ の配分の最適化を目指す。目的関数と制約条件は(6), (7), (8)式に示す。ただし, m はステークホルダー数を表している。

$$\text{Maximize} \quad \sum_{k=1,m} S_k^{(n)} \quad (6)$$

$$\text{s. t} \quad \sum_{j=1,n} R_Ratio_{j,k}^{(i)} \leq 1 \quad (i = 1, 2, \dots, n) \quad (7)$$

$$R_Ratio_{j,k}^{(i)} \geq 0 \quad (8)$$

目的関数(6)式は, (5)式の n 期におけるステークホルダーごとの S の和の最大化し, $n = 15$ とする。また制約条件(7)式は 1~ n 期までの, $R_Ratio_{j,k}^{(i)}$ の合計を 1 以下にする制約であり, (8)式はそれにかかる非負条件である。この仕事量の配分の最適化問題を, ソルバーを用いて解いた結果と, 1~5 期に RD1, 6~10 期に RD2, 11~15 期に RD3 の要件定義を行う割合を 1, そのほかを 0 とした, 集中型と呼称する 2 パターンで実験した。

問題設定について表 1 を用いて説明する。RD1, 2, 3 は, 要件定義作業を示す。次に A, B, C はステークホルダーを示す。また, 1 はステークホルダーがその要件に携わることを示す。初期 CT は, ステークホルダー間で初めに必要なコミュニケーション量を示す。例えば RD1 のステークホルダーは A および C の 2 名であり, その初期 CT は 0.15 である。

表 1 実験の設定

	関係者 A	関係者 B	関係者 C	関係者数	初期 CT
RD1	1	0	1	2	0.15
RD2	1	1	1	3	0.1
RD3	1	1	0	2	0.05

図 3 は Goel-Okumoto モデル, 図 4 は遅延 S 字モデルを用いて集中型で構造化率 S を計算した結果である。また, 図 5 は Goel-Okumoto モデル, 図 6 は遅延 S 字モデルを用いて, ソルバーで構造化率 S を計算した結果である。また, それぞれのモデル, スケジュールに対する, 目的関数であるステークホルダーごとの構造化率の合計も図内に示している。集中型の 1~5 期のステークホルダー B, 10~15 期のステークホルダー C に作業がなく, その間 S が全く変化していないことがわかる。図 3 では, 要件を新たに始める 1, 6, 11 期に構造化率 S が大きく上がり, そこから少しずつ収束しているのが見られる。これ

は Goel-Okumoto モデルの放物線状を描き収束する特徴がでた結果といえる. 図4では, 遅延 S 字モデルに特徴的な S 字が Goel-Okumoto モデルと同様に新しく作業を始めるタイミングで見られる.

一方ソルバーによる解は, 図5, 図6のように, Goel-Okumoto モデルと遅延 S 字モデル両者ともステークホルダーごとの S の推移に違いがあまりない. 図5では, 構造化率 S が, 大きく伸び, 収束している様子が見られる. しかし図6では, 特徴である S 字の収束部分が見られない. これは, 作業期間が足りず, 設定した期間内では収束できていないことを示している.

目的関数の結果を見ると, どちらのモデルでも集中型のほうがよいという結果になっているが, 最適性については, 詳細な分析を要する.

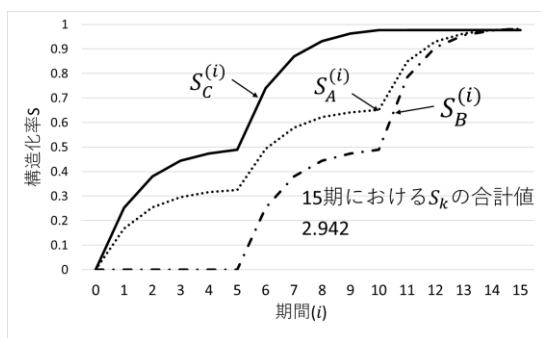


図3 集中型による Goel-Okumoto モデルの結果

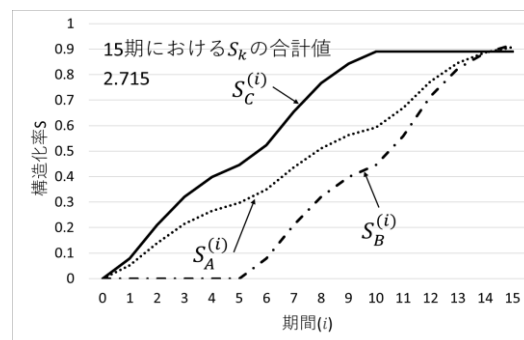


図4 集中型による遅延 S 字モデルの結果

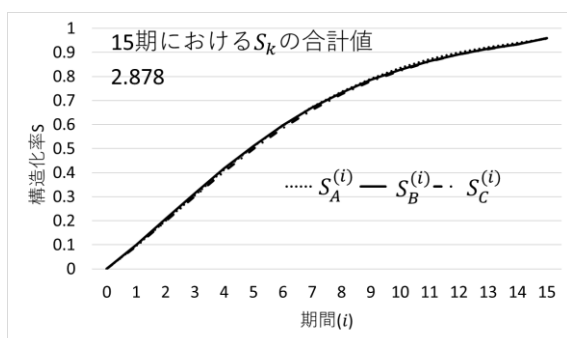


図5 ソルバーによる Goel-Okumoto モデルの結果

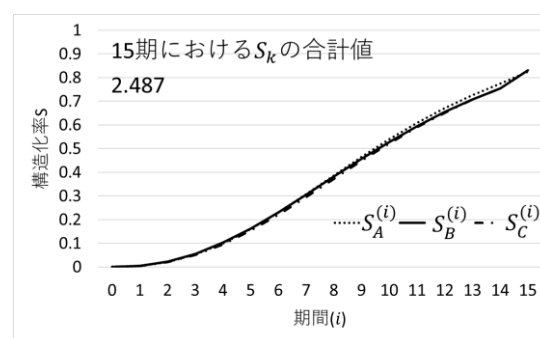


図6 ソルバーによる遅延 S 字モデルの結果

4. 今後の課題

最適化計算において, 最適解を安定して求める方法を検討する. また, 進捗管理に適した成長曲線のパターンを探求し, 要件定義の計画立案に使用する. さらに, 要求ごとの難易度を反映した構造化率 S の算出方法を求める必要がある. ステークホルダーの能力を設定することも, 検討が必要である. さらに, 現状は進捗管理のなかの計画機能のみを対象としているが, 実測値を追加し, 要件定義の進捗状況により, 計画を逐次変更する管理手法を検討する.

参考文献

- [1] 日本情報システム・ユーザー協会 (JUAS), ソフトウェアメトリックス調査 2016, 2016.
- [2] Wiegers, K., Beatty, J., ソフトウェア要求 第3版, 日経 BP 社, 2014.
- [3] Zhang, X., Teng, X., Pham, H., Considering Fault Removal Efficiency in Software Reliability Assessment, IEEE Trans. Systems, Man, and Cybernetics - Part A: Systems and Humans, Vol.33, No.1, 2003, pp.114-120.
- [4] 山田茂, ソフトウェアの信頼性/安全性評価, 安全工学 33 卷 6 号, 1994, pp.432-441.
- [5] Ishii, N., Muraki, M., Communication-Centered Project Management for Requirements Definition Phase, Industrial Engineering & Management Systems, Vol.11, No.1, 2012, pp.39-47.