

ソフトウェアプロジェクトリスク管理のための Goal-Question-Metric モデリング

門 田 暁 人^{†1}

本稿では、プロジェクト遅延の兆候を検出することを目的として GQM モデリングを行った際に、筆者らが直面した課題とその解決方法を紹介する。

Goal-Question-Metric Modeling for Software Project Risk Management

AKITO MONDEN^{†1}

This paper introduces a case study and improvements of GQM modeling for project slippage detection.

1. はじめに

ソフトウェア開発プロジェクト管理では、スケジュール通りにプロジェクトを完遂させることが至上命題となる。しかし、現実には、プロジェクトをとりまく様々な問題により作業遅延が発生し、コスト増大や納期超過を招くことも少なくない。筆者が EASE プロジェクト¹⁾において、ソフトウェア開発企業 12 社 20 名の開発経験者にアンケートを実施したところ、2 名を除いてプロジェクトの大きな遅延を経験したことがあると回答し、その発生工程 (複数回答) は図 1 の通りであった (n.s. は未回答を示す)。図のように、プロジェクトの遅延は、要求分析からシステムテストまでに至るまで、幅広い工程で発生している。特に、「詳細設計」と「結合試験」で遅延が数多く発生しており、上流工程での問題がこれら工程で発覚し、遅延を引き起こしたケースが多いと考えられる。

筆者はこれまでに、プロジェクト遅延の兆候を可能な限り早期に、かつ、定量的に検出することを目的として、開発プロセスからリアルタイムにデータを計測・分析し、開発現場へとフィードバックすることに取り組んできた。データ計測に際しては、やみくもに計測を行っても良い成果は期待できないため、ゴール指向の計測フレームワークとして、Goal Question Metric

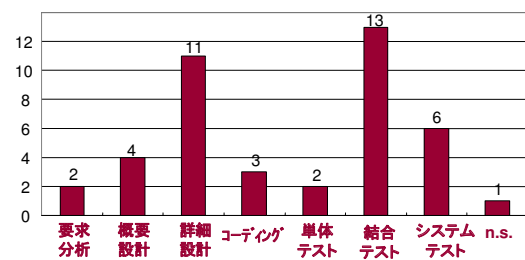


図 1 プロジェクト遅延発生工程

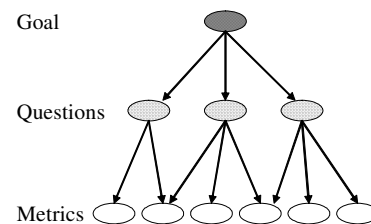


図 2 GQM モデル

(GQM) パラダイム²⁾ を利用した。本稿では、GQM の適用に際して筆者が直面した課題、および、その解決方法を述べる。

2. GQM モデルとその課題

2.1 GQM モデル

GQM モデルは、図 2 のように、計測のゴール (Goal) と、そのゴールを達成するために答えなければ

^{†1} 奈良先端化学技術大学院大学情報科学研究科
Graduate School of Information Science, Nara Institute
of Science and Technology

ばならない質問 (Question), および, 質問に答えるためのメトリクス (Metrics) の 3 階層から構成される。また, メトリクス値の解釈方法 (ゴールにどう結びつくかの記述) が狭義の “Model” として別途記述される。例えば, 「設計における問題を明らかにする」という Goal に対して, 「設計品質は?」という Question と 「設計バグ密度 X」という Metrics が定義された場合, Model は, 「 $X > 3$ 個/ページ数 \Rightarrow 設計品質に問題あり」などと記述される。

2.2 GQM モデルの課題

従来, ソフトウェア開発管理に関して, 数多くの GQM モデルが作成されてきたが³⁾, プロジェクトのリアルタイム計測 (コーディング/単体テスト工程) に関するモデルの構築事例は報告されておらず, 次のような問題があった。

- (1) リアルタイム計測のような時系列で変化するメトリクスに関する GQM モデルの構築事例はほとんどなく, どのような G, Q, M を定めるべきか明らかでない。
- (2) 前記の 「 $X > 3$ 個/ページ数」のようなメトリクス値に関する基準値 (ベースライン値) を決めるには, 当該開発組織における計測の積み重ねが必要となり, モデル構築の敷居が極めて高い。
- (3) 与えられた Goal に対し, Q, M, Model の定義した際に, その根拠や仮説がモデル中に明記されず, 不明確である。

3. GQM モデルの構築

筆者らは, 2.2 節で述べた課題を踏まえ, GQM モデルを拡張し, プロジェクトの遅延の兆候を検出するための GQM モデルを作成した。

まず, G や Q の定義に役立てるために, 開発経験者へのアンケートにより, 「プロジェクト遅延が発覚する前に, どのような兆候があったか?」を調査した (表 1)。アンケート対象者は図 1 と同じである。表 1 に示されるように, 不具合の発生/対応, 設計変更, 命令系統の問題などが大きな割合を占めた。このことから, 筆者らは, G の候補として, 不安定な要求, 不完全な設計, 人的資源の割り当ての不備, 人の劣悪なソースコード品質などの検出を取り上げた。

次に, M の候補を決めるにあたって, 開発現場におけるメトリクス計測の負荷を低く抑えるために, Empirical Project Monitor (EPM)⁴⁾ の利用を前提とすることにした。EPM は, バージョン管理システムや, 障害管理システムなどから時系列データを自動計測で

表 1 プロジェクト遅延の兆候

兆候	回答数
多すぎる不具合 (バグ) の発生	4
設計変更の増加	3
スケジュール通りに作業を進めるのが次第に苦しくなる。	3
仕様書レビューが不十分なまま作業が進む。	2
工数が増大するが成果物ができない。	2
命令系統の混乱	2
「完了」との報告があるもエビデンスなし。	1
見積もりをするたびに規模が増える。	1
設計書のページ数が少なすぎる。	1
バグ対応が遅すぎる。	1

きる。その半面, 計測できるデータに制限があるため, トップダウン/ボトムアップの両面からモデルを構築することとなった。

また, 次の 2 点について GQM モデルを拡張した。

- (1) Q の定義に関する仮説を明記するための 「Hypothesis レイヤー」を設けた。
- (2) メトリクスのベースライン値を持った厳密な Model を記述することを諦めた。その代わりに, メトリクス値の 「変化」に着目することとし, 値の変化をどう解釈すべきか記述する 「Interpretation レイヤー」を設けた。

以上の方針に基づいて, EASE プロジェクトにおいて 4 つの GQM モデルを作成し, 先進ソフトウェア開発プロジェクト⁵⁾ に適用し, その有用性が確認された。

4. おわりに

本稿では, プロジェクト遅延の兆候を検出することを目的として GQM モデリングを行った際に, 筆者らが直面した課題とその解決方法を紹介した。

参 考 文 献

- 1) EASE プロジェクト <http://www.empirical.jp/>
- 2) Basili, V. R. and Weiss, D. M., “A methodology for collecting valid software engineering data,” IEEE Trans. on Software Engineering, Vol.SE-10, No.6, pp.728-838, 1984.
- 3) Solingen, R. and Berghout, E., “The Goal / Question / Metric method - A practical guide for quality improvement of software development,” McGraw-Hill, 1999.
- 4) 大平, 横森, 阪井, 岩村, 小野, 新海, 横川, “ソフトウェア開発プロジェクトのリアルタイム管理を目的とした支援システム,” 信学論 D-I, Vol.J88-D-I, No.2, pp.228-239, Feb. 2005.
- 5) 先進ソフトウェア開発プロジェクト <http://sec.ipa.go.jp/std/ad.php>