Journal of International Association of P2M
Vol.16 No.2, pp.109-127, 2022
研究論文

機械学習を用いたプロジェクト異常予測の実践
システム開発プロジェクトの失敗に影響するトリガー事象の抽出
A Practice of Project Failure Prediction Using Machine Learning
Extraction of Risk Triggers that Affect the Failure of System Development Projects

新田　佳菜　Kana NITTA†
森　俊樹　　Toshiki MORI†

P2M プログラムマネジメントでは、同時並行的に進行する多数のプロジェクトを管理する必要がある。しかし、従来の人手に頼ったマネジメント手法だけでは、全てのプロジェクトに十分に目が行き届かないという課題があった。そこで、機械学習を用いたプロジェクト異常予測をプログラムマネジメントに適用し、重大なプロジェクトリスクを自動で検知することを試みている。システム開発プロジェクトを対象とした試行により、精度面では実運用可能なレベルになったことを確認したが、その一方で、高リスクプロジェクトを検知するだけでは、プログラムに影響を及ぼすリスクとして適切に判断し、対応することができないという課題も明らかになった。本研究は、プログラムマネジャーがリスクを検知し対処するまでの意思決定の支援を目指し、システム開発プロジェクトの失敗の予兆となるリスク事象を体系的に抽出し、実事例と比較してその妥当性を確認した。
キーワード:　プログラムマネジメント、リスクマネジメント、プロジェクト異常予測、機械学習

In P2M program management, managers are forced to control multiple projects which progress at the same time. However, it is difficult for them to manage all the projects by their hands. We tried to apply machine learning to program management so that we can get predictions of project failure. Due to the trial, we confirm the accuracy has reached a level where it can be used in actual operations. On the other hand, it turns out that predicting failure is not enough to decide the risk will affect program's failure and take some actions for its early recovery. This study aims at supporting program managers to deal with program risks. We systematically extract risk triggers which lead system development projects into trouble, and validate them by comparing several case studies.
Keywords: Program Management, Risk Management, Project Failure Prediction, Machine Learning

1. はじめに

企業の製品開発部門では、顧客に対して多様な製品やサービスを提供するために、多数のプロジェクトが同時並行的に進行しており、組織としてこれらを統合的にマネジメントするP2M（プログラム＆プロジェクトマネジメント）[1]の概念が極めて重要となっている。P2M

†　（株）東芝　ソフトウェア技術センター

では、プログラムを「特定使命を実現する複数プロジェクトが有機的に結合された活動」と定義し、プログラム活動をスキームモデル（構想）、システムモデル（構築）、サービスモデル（運用）に分割し、それぞれの相互関係を一体化した統合マネジメントの概念を提供している [2]。P2M のリスクマネジメントでは、このプログラムのライフサイクルに沿って、プログラムとプロジェクトの役割と責任範囲を定義したうえで管理する必要があり、下位のプロジェクトリスクがプログラム実行に影響を及ぼさないよう、各プロジェクトで発生するリスクを監視する必要がある [3]。そのため、プログラム実行の過程において、プログラムマネジャーは常にプロジェクトリスクの見極めと特定使命の達成を念頭に置いたリスクへの対処が求められる。しかし、従来の経験者の知見に基づいたマネジメントだけでは、同時に進行する多数のプロジェクトに対して、プロジェクトの状況を正確に把握できず、リスクへの対応が後手に回るなどの問題がある。こうした問題に対して、定量的なデータに基づく客観的な各プロジェクトの状況把握、及び将来のリスク予測に基づく先回りの管理をすることで、プログラムマネジャーの意思決定を支援し、ひいてはプログラムの成功に貢献することが期待される。近年では、プロジェクト管理支援ツール及び、開発環境の発達により、組織内のデータベースに十分な量のプロジェクトデータが蓄積されるようになり、プログラムマネジメントへの機械学習の応用が可能になった。そこで、我々は機械学習モデルの 1 つであるナイーブベイズ識別器によるプロジェクトの異常予測 [4]を組織のプログラムマネジメントに適用し、プログラムの成否に影響を与えるプロジェクトリスクを自動で検知することを試みている。実開発部門でのシステム開発プロジェクトを対象とした試行により、予測精度の面において実適用可能なレベルに到達したことを確認した。

しかしながら、現状、リスクが高いプロジェクトを予測できても、予測結果に基づくリスク対処までには至っていない。これは、「危ない」プロジェクトを検知できても、実際にどのような対応をとるべきかがわからないという課題に起因する。本課題の解決には、プロジェクトが高リスクとなった原因を特定し、それに紐付いた対応策を実行する必要がある。

そこで、本研究では、システム開発プロジェクトで発生するリスク事象の原因分析に役立つ 25 項目のトリガー事象を提案する。これらのトリガー事象を用いることで、リスク事象の分析が容易となり、プログラムマネジャーの意思決定を支援できる。25 項目のトリガー事象は、IPA/SEC「IT プロジェクトの『見える化』上流/中流/下流工程編」 [5] [6] [7]に記載されたプロジェクト失敗事例の体系的な整理により抽出した。そして、実際に異常予測で高リスクとなった過去のプロジェクト事例 7 件を分析し、これらのトリガー事象と比較することで、その妥当性を確認した。

本論文の構成を以下に示す。まず、2 章では関連研究を紹介する。続く 3 章では現在の取り組みと課題について説明する。4 章では IPA/SEC「IT プロジェクトの『見える化』上流/中流/下流工程編」 [5] [6] [7]のプロジェクト失敗事例の体系的な整理により抽出した 25 項目のトリガー事象を示す。5 章では過去のプロジェクト事例 7 件の分析結果を示す。6 章ではリスク対応策検討のためのトリガー事象の活用方法について提案する。最後に、7 章でまとめを述べる。

2. 関連研究

リスクは「何らかの事業あるいは行動を行う上での不確実性」と定義される。P2M Version 2.0 におけるリスクマネジメントでは、経営環境の変化への対応などの「コーポレートリスク」、プログラムの価値予測誤りに関係する「プログラムオペレーションリスク」、プロジェクト遂行におけるエンジニアリング要素を持つ「プロジェクトリスク」の 3 つに分類されている[2]。武富は、P2M Version 2.0 のリスクマネジメントの定義に対し、ビジネス、プログラム、プロジェクトの 3 つのレベルにおいてそれぞれが対象とするリスクを定式化し、プログラムのライフサイクルに沿って各レベルでのリスクマネジメントのアプローチを明確にした [3]。佐藤らは、P2M のリスクマネジメントにバランス・スコアカードを導入し、プログラムオペレーションリスクとプロジェクトリスクをリスクの影響範囲によって分類・整理し、可視化する統合リスクマネジメントの手法を示した [8]。岩崎らは、リスク抑制を目的とした従来の P2M リスクマネジメントに対して、価値創造に向けた能動的なリスク許容によるリスクマネジメント手法である Risk Appetite Framework の導入を提唱した [9]。越島は P2M リスクマネジメントにおける「予測や制御のできない外的事由」への対応の必要性を主張し、サイバーセキュリティマネジメント手法（NIST-CSF、CCE）の P2M への適用を考察した [10]。プログラムにおけるリスク事象の分析についても研究がおこなわれており、清水は、プログラムリスクを時間軸で分類し、上流のプログラム設計段階までのリスクを戦略レベルのリスクとして、主要なリスク事象とその対策例をまとめた [11]。一方、本研究では、下流のプログラム実行段階におけるリスクにフォーカスし、システム開発プロジェクトで発生する主要なリスク事象を洗い出した。さらに、機械学習を用いることでプロジェクトリスクを自動で検知し、プログラムの成功に影響するリスクを早期に把握できる、新しい P2M リスクマネジメントの枠組みを示している。

IT プロジェクトにおけるリスク事象の分析については、過去にさまざまな研究がおこなわれてきた。Boehm はプロジェクト経験者を対象とした調査を実施し、ソフトウェア開発におけるリスク事象について上位 10 項目のリストを提示した [12]。Ropponen らは Boehm のリスト内容に偏りがあることを指摘し、Boehm のリストを拡張した内容をアンケートによって評価し、主成分分析を用いてプロジェクトのリスク事象を 6 項目にまとめた [13]。Chaos Report（1995）でもプロジェクト経験者への調査結果から、上位 10 項目のリスク事象を提示している [14]。Kasser らは「プロジェクトの失敗は技術的要因よりも人的要因に問題がある」と主張し、予め選定したリスク事象の候補をソフトウェア開発従事者へのアンケートによって評価し、Chaos Report などの結果と照合することで、9 つの重要なリスク事象を洗い出した [15]。IPA/SEC「IT プロジェクトのリスク予防への実践的アプローチ」では、ユーザー企業とベンダー企業双方の実務経験者による検討により、要件定義工程でのリスクを中心とした 24 項目のリスク事象ドライバーを洗い出した [16]。

また、リスク事象を単に分類するだけでなく、そうした情報を使ってプロジェクトの失敗を予測する研究も行われている。水野らはプロジェクトのリスクを評価するアンケートを実

施し、その結果からベイズ識別器を用いた予測モデルを構築して、プロジェクトの最終結果を予測した [17]。浜野らは失敗が予測されたプロジェクトの改善を目標に掲げ、水野らが予測に使用したアンケートとプロジェクトの最終結果の関係を相関ルールマイニングによって分析・評価し、重要なリスク事象を考察した [18]。森らは定量メトリクスを用いたプロジェクトの異常予測モデルを構築し、プロジェクトの途中段階で得られた新たなデータをもとに、逐次的な予測をおこなった [4] [19]。大島らはプロジェクトマネジメントに必要とされる知識を整理し、AI の適用可能性について検討した [20]。これらは、多数のプロジェクトが同時並行的に進行するプログラムマネジメントの環境において、より一層重要となる。

本論文では、リスク事象のうち、その発生原因となる事象を“トリガー事象”と呼ぶことにする。IPA/SEC「IT プロジェクトの『見える化』上流/中流/下流工程編」 [5] [6] [7]に記載されたプロジェクト失敗事例の体系的な整理により、システム開発プロジェクトで発生する 25 項目の典型的なトリガー事象を洗い出した（4.1 節）。これらのトリガー事象に対して、複数の文献 [12] [13] [14] [15] [16] [17]と比較することで網羅性を確認するとともに（4.2 節）、実際の異常予測において高リスクとなった実プロジェクト事例への当てはまりを評価した（5 章）。

3. プロジェクト異常予測

プロジェクト異常予測は、機械学習を用いて過去のプロジェクトの失敗傾向から、進行中のプロジェクトリスクを自動的に検知するものである。プロジェクト異常予測をプログラムマネジメントに適用する場合、スキームモデル段階で設計したプログラムの価値（V）や、プログラム成果の QCD が、システムモデルやサービスモデルで未達となるリスクを検知できるよう予測モデルを構築する必要がある。

3.1. プロジェクト異常予測の仕組み

プロジェクト異常予測モデルにはナイーブベイズ識別器を採用している [4] [21]。ナイーブベイズ識別器は予測に大量のデータを必要とせず、データの欠損や外れ値による影響を受けにくいという特徴をもつため、欠損やノイズを含む可能性のあるプロジェクトデータの分析に適している。

ナイーブベイズ識別器はベイズ統計を応用した機械学習モデルの一種である。ベイズ更新の仕組にしたがって事前確率を逐次的に更新し、最終的な事後確率（予測結果）を計算する。ベイズ更新はベイズの定理に従い、事前確率を$P(F)$としたとき、事後確率$P(F|D)$は次式によって計算される：

$$P(F|D) = P(F) \times \frac{P(D|F)}{P(D)}$$

このとき、新たな情報の影響度として事前確率に掛けられる数値 $P(D|F)/P(D)$ をリフト値と呼ぶ。リフト値は説明変数の値から統計的手法を用いて計算する。

異常予測モデルでは、予め設定したプロジェクトの失敗を示すメトリクスが閾値を上回る（または、下回る）確率を予測する。そのため、事前に、プロジェクトの失敗を定量的に評価できる形式で定義する。失敗の定義として、P2M におけるプログラムとプロジェクト間の責務を示すリスクの閾値を設定 [3]し、「プロジェクトマネジメントで対応すべきリスクの閾値を超えた場合とする」などが考えられる。また、予測モデルに利用する説明変数も統計的分析や経験者による知見から決定する。図３－１にプロジェクト異常予測モデルの適用イメージを示す。

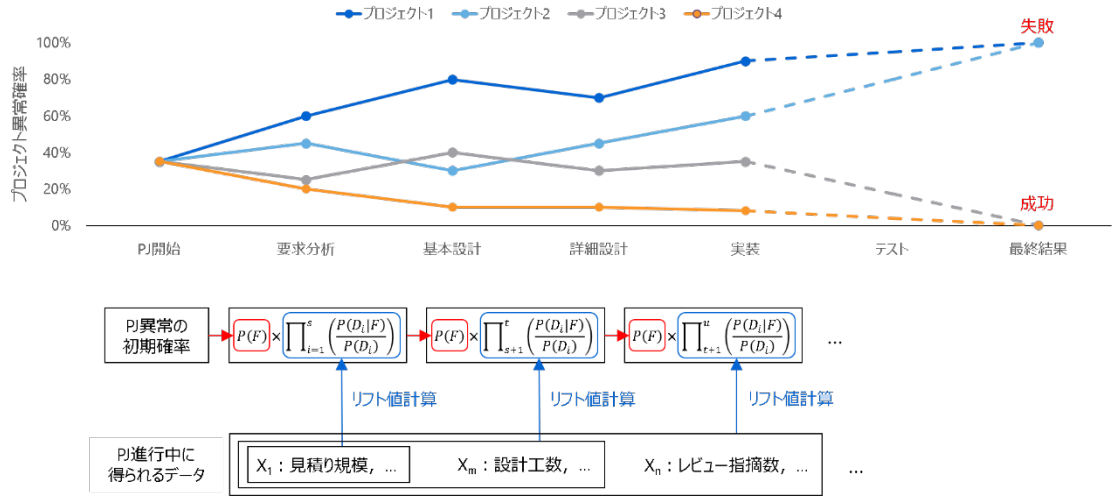


図３－１　ナイーブベイズ識別器を用いたプロジェクト異常予測

3.2. 実開発部門への適用

適用対象部門では事業戦略に基づいて複数のプロジェクトを並行して実施している。また、これらのプロジェクトは特定顧客との中長期的な関係構築を目指して戦略的に展開されており、広義のプログラム活動に相当する。この組織のプログラムマネジメントに上記プロジェクト異常予測を適用し、実開発部門において同時進行する多数のシステム開発プロジェクトを対象とした異常予測を試行している。進行中のプロジェクトに対する異常予測はこれまでに複数回実施しており、データ集計・分析・報告の手順は概ね確立し、予測モデルの精度も実適用可能なレベルに達している。プロジェクト異常予測の実施手順の概要を図３－２に示す。

プロジェクトの失敗の定義にはさまざまなパターンがあり得るが、本取り組みでは「プロジェクトの製造損益がある一定の値を下回った場合」を失敗と定義し、経験者の知見等に基づいて約 20 個の説明変数を採用した。図３－３は採用した変数の一部を示している。組織内に蓄積された過去 10 年分の約 500 件のプロジェクトデータを学習データとしてプロジェクト異常予測モデルを構築し、交差検証などを用いて予測精度の妥当性を確認した。その上で、進行中のプロジェクトの予測に適用し、新たなデータに対しても予測精度が概ね良好である

ことを確認した。また、予測結果の活用方法として、スタッフ間での組織を横断した情報共有やマネジメント層への情報提供なども実施した。

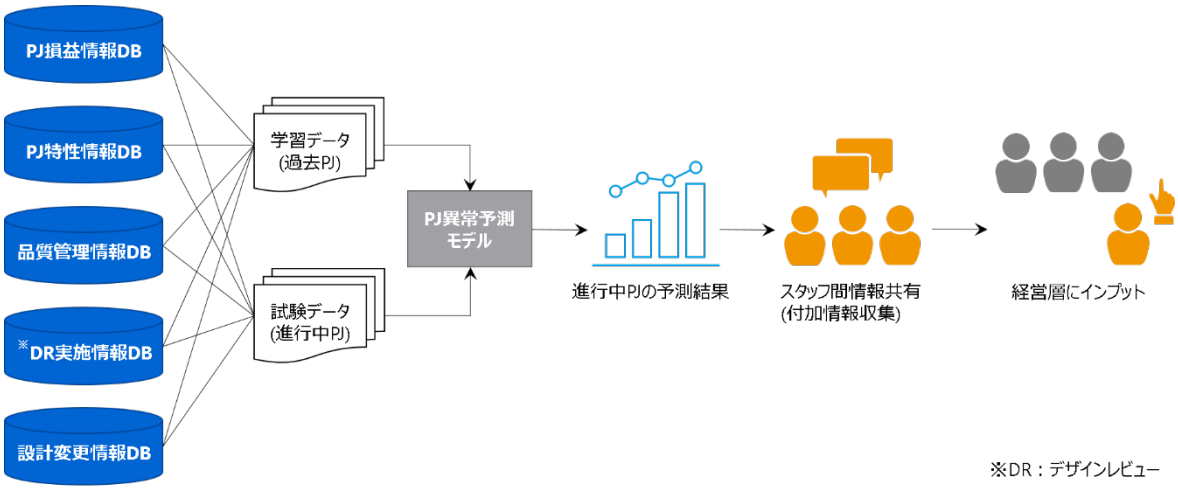
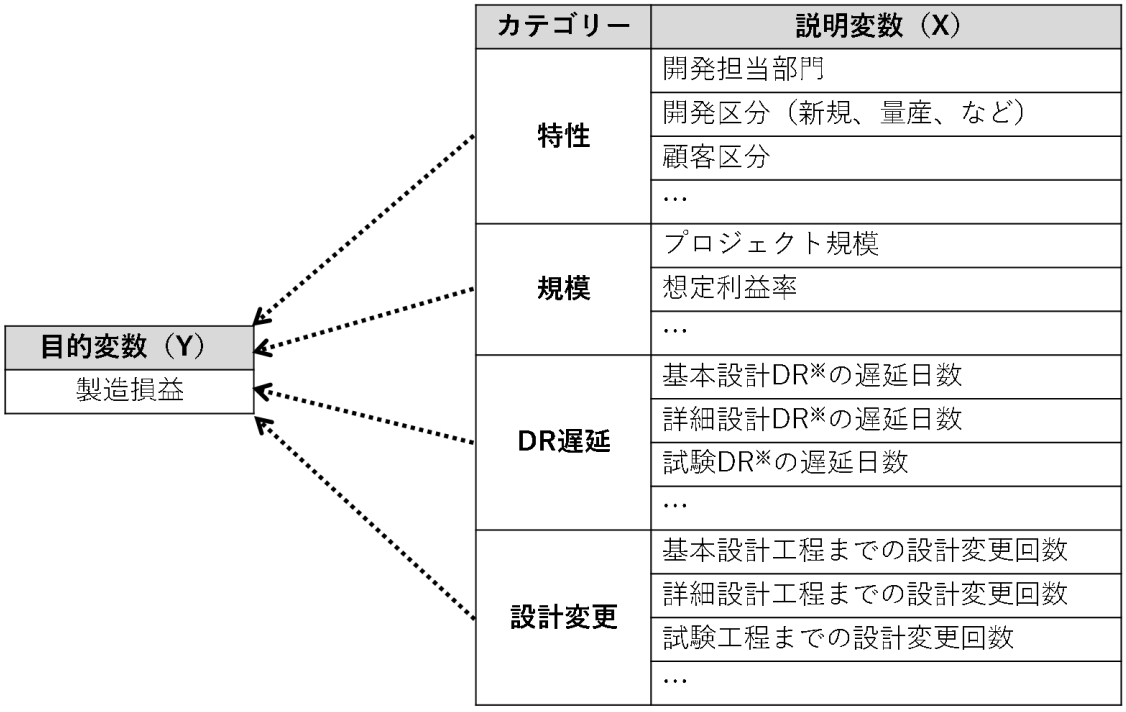


図３－２　プロジェクト異常予測の実施手順



カテゴリー	説明変数（X）
特性	開発担当部門
	開発区分（新規、量産、など）
	顧客区分
	…
規模	プロジェクト規模
	想定利益率
	…
DR遅延	基本設計DR※の遅延日数
	詳細設計DR※の遅延日数
	試験DR※の遅延日数
	…
設計変更	基本設計工程までの設計変更回数
	詳細設計工程までの設計変更回数
	試験工程までの設計変更回数
	…

※DR：デザインレビュー

図３－３　予測モデルに含まれる変数（一部抜粋）

試行により、予測モデルの精度が概ね良好であることは確認できたが、一方で課題として、予測結果に基づく早期のリスク対処までにはなかなかつながらないことも判明した。それは、異常予測によって同時進行する複数プロジェクトの中から「危ない」プロジェクトを検知できても、それをプログラム実行に影響を及ぼすリスクと判断し、適切な対応策を検討するには情報が不足しているためである。進行中のプロジェクトからリスクを検知し、その

対応にまで繋げるには、現在のプロジェクト異常予測の枠組みに加えて、リスク事象の原因分析と対策の立案を支援する新たな仕組みが必要である。

4. プロジェクト失敗のトリガー事象

本研究では、システム開発プロジェクトの失敗に影響する 25 項目のトリガー事象を洗い出した（表４－１）。これらのトリガー事象は、IPA/SEC「IT プロジェクトの『見える化』上流/中流/下流工程編」[5][6][7]の“プロジェクトにおける問題事象と対策事例集”に含まれる計 193 件のプロジェクト失敗事例（表４－２、表４－３、表４－４）に基づいて、KJ 法[22]を用いて問題事象を分類・整理することで抽出した。

表４－１　プロジェクト失敗のトリガー事象

大分類	中分類	トリガー事象
内部要因	体制・スキル	リソース不足
		役割・責任分担不明瞭
		要員の交代・離脱
		要員のスキル不足
	管理・環境	マネジメント不良
		コミュニケーション不全
		プロセス・ルールの不徹底
		作業環境の不整備・調整不足
	経営・営業・他PJ	組織とPJ間の隔たり
		営業主導の受注
		他PJとの並行作業
外部要因	顧客	顧客都合の変更・遅延
		顧客との認識の相違
	外注・調達先	安易な外注・調達先の選定
		外注・調達先の管理不十分
作業状況	見積・計画	受注前のリスク対策不足
		過少見積もり
		短納期
	要求仕様	業務要件の理解不足
		曖昧・不明確なスコープ
		不完全な要求仕様
		要求仕様の追加・変更発生
	設計・システム構築	利用技術の試行・検証不足
		技術難易度が高い
		設計不良

表４－２　プロジェクトにおける問題事象と対策の事例集タイトル一覧（その１）
～IT プロジェクトの『見える化』上流工程編～

事例番号	事例タイトル	事例番号	事例タイトル
1	顧客側担当者が異動になり約束事が反故に	30	未知のパッケージを協力会社から提案してもらったが、そのパッケージに品質問題が発生
2	本当のユーザーを見誤ってしまう	31	経験の少ないオープン系システムで、プロジェクト・マネージャがコントロール不能に
3	新サービスと非合理な納期	32	複数会社でのプロジェクトで旗振り役が不在
4	仕切る気がないとりまとめ役	33	顧客指定のフレームワークで受注し、大幅コスト増
5	発注側から要件が小出しにされ、マネジメントできなくなる	34	本番データに現場部門のパッチが当たっており、論理的不整合が多く移行コスト増大
6	体制の事前確保が仇に	35	現行システムを再利用した新システムを導入しようとして例外処理に苦しむ
7	営業がプロジェクト・マネージャの承認のないまま概算見積もりで受注契約を結んでしまった	36	顧客側からの要件出しが細切れで、なかなか要件を確定してくれない
8	基本設計の承認なしで詳細設計に着手	37	既存資産が流用できると思っていたが、流用できないことが分かりコストオーバー
9	評価が終わっていない基本設計で協力会社に着手依頼	38	顧客との共同開発という名目で、イニシアチブが取れずに要件がなかなか確定できない
10	性急な立ち上げでの体制確保のため、要員調達でコスト高に	39	何でも言うことを聞くという応札条件の案件
11	業務経験・業務知識の乏しい要員で要件定義工程を実施	40	パッケージの品質と機能は十分に確認しよう
12	コーディングの標準化を実施する時間を惜しんだためにかえって非効率に	41	仕様変更に関する扱いは、顧客と意識合わせをしておくこと
13	法務部同士で合意に至らず契約が破談	42	大幅な再利用を前提とした開発は要注意
14	初物プログラム言語での開発で十分な性能が出ない	43	作りながら仕様を確認
15	現行システムと新システムの並行開発	44	知らないもので決まっていないものを作る
16	運用テスト入ってから現場固有の業務要件が多数発覚	45	無理な見積もりを通すと作り手か使い手のどちらかが破綻する
17	標準仕様に先駆けてシステム構築したが、標準化への対応で品質問題が発生	46	個別に検討したものを繋げてみたら、繋がらなかった
18	仕様検討に時間を要し、十分なテストができないまま受入テスト(1)	47	顧客側の体制、顧客との役割分担、作業分担を明確にすること
19	仕様検討に時間を要し、十分なテストができないまま受入テスト(2)	48	オフショア開発では、ドキュメントの記述は詳細に、プロジェクト管理はしっかりと
20	顧客側メンバーの体制が弱く、仕様が詰められない	49	顧客の体制が弱いときは、仕様追加、変更が多発する可能性が大
21	不条理な予算圧迫要請で開発に着手	50	規模が大きくなくても、協力会社に丸投げは危険
22	さほど大きな影響はないだろうと思い、既存システムのプログラムを修正	51	信頼関係だけでは適切な課題管理はできない
23	時間のかかる顧客承認プロセスを見切って、承認前に作業着手	52	インパクトが大きい要件への対処は先送りしない
24	契約金が未決定ながら進めたプロジェクトだったが、契約時に予算額が大幅に減額	53	フィット&ギャップ分析をせずにパッケージ導入の決断はしない
25	保守フェースにおける作業慣習で、仕様変更の見積りとともに作業を着手してしまった	54	過去データ移行の難易度はプロジェクト・コストにインパクトあり
26	最新技術てんこ盛りで、納期も短いシステム構築で手戻り多発	55	中核と思われる機能以外に相応の開発規模が隠れていることも
27	DBMSが詳細設計後に決まって手戻り	56	ドキュメントの成果レベルは事前調整が不可欠
28	開発効率向上のためのフレームワーク作りがボトルネックになってしまった	57	RFPとの差異が明確になっているか
29	プロジェクト・マネージャに大規模プロジェクトの経験がなく、結合テストの頃から調整不足が露呈	58	パッケージ開発では業務要件とのギャップが重要な見極めのポイント

表４－３　プロジェクトにおける問題事象と対策の事例集タイトル一覧（その２）
～IT プロジェクトの『見える化』中流工程編～

事例番号	事例タイトル	事例番号	事例タイトル
1	仕様変更による「品質問題」	30	性能の異なる新旧資源混在環境でのトラブル
2	仕様変更による「予算超過」	31	異常時挙動を想定しない設備設計
3	仕様変更による「納期遅延」	32	基本設計書の品質不足の問題
4	仕様変更による「システム老化」	33	仕様確定遅れによる問題
5	営業が中流工程以降身を引き仕様変更で「不採算」に	34	アプリケーションプログラムの単体品質にばかりとらわれ、システム全体の納期・品質が守れず
6	仕様変更の連続で出口が見えないプロジェクト、メンバーが疲弊	35	業務アプリケーション・プログラム性能にとらわれシステム性能問題を見落とす
7	変化即応型の開発をScope変更ルールが無く詰け負った	36	仕様通りに開発したが運用障害で事業が長時間停止
8	変更要求問題で、PMの交渉力、報連相が不足	37	システム移行設計不良
9	開発中のリスクについてPMが上司にエスカレーションをあきらめた	38	製品間インターフェース保証の見落とし
10	PM・顧客間不信頼	39	有名ソフトを未検証のまま使用し不具合発生
11	仕様変更を一切受け付けないことによる問題	40	営業主導で決めたソフトで開発し要員不足、クレーム、不具合発生
12	複雑な顧客ステークホルダー組織で仕変が多発	41	パッケージ採用と言いながらカスタマイズ量過多（実質固有ソフト）
13	仕様凍結確認せず中流工程に着手	42	パッケージ機能の認識、期待感のギャップ大。規模、工数、費用、納期大幅ズレ
14	マイグレーションの要件、実質はシステムレベルアップ。仕様追加極大	43	特定顧客で未完パッケージを完成させる計画が頓挫
15	世代交代で顧客、SIベンダーともに熟知した要因不在。マイグレーション仕様固まらず	44	システム全体としての信頼性設計の不備
16	正式要求仕様書無く作業。仕様曖昧により変更多発	45	過負荷・限界系の見落とし
17	当初納期の大幅前倒し要求を受け付けQCD問題に	46	方式性能ボトルネックの見落とし
18	当初価格の大幅削減要求を受け付けQCD問題に	47	最新ソフト、バージョンを検証不足のまま採用。不具合多発
19	過剰な責任所在文化による契約締結遅れ、物事のあいまいさ	48	自前ソフト以外の検証、確認認識薄く、後工程で問題発生
20	顧客責任部門、責任者不明	49	アプリケーション・プログラムの生産性で言語を選択したが、性能問題がネックとなった
21	変えられない既存業務プロセスにパッケージを導入	50	開発ルール未確定のため保守性、流用性が悪い
22	システム共同化、再構築で自社独自のテスト不十分。重大障害発生	51	新しい言語の熟練者不足でプログラム完成度が低い
23	システム共同化を目指すが、先行ユーザーのサービスインの遅れで納期遅延	52	客観的な生産指標が少なく、工数、費用にインパクト大
24	ハードウェア、OS更改でテスト必要認識不足。重大障害発生	53	担当者にシステム的認識欠落、障害時の切り分け、情報採取の仕組不足
25	スケジュールが遅れ、業務主体のテストとなり、基盤・運用系不安定	54	オフショアで品質と納期遅延の問題（発注側問題）
26	障害の数の定量的管理にとらわれ過ぎ，品質を誤評価	55	オフショアで品質と納期遅延の問題（受注側問題）
27	システム統合の場合、企業文化の相違によるコンティンジェンシー・プラン不備	56	外注会社責任者は担当者に任せきり
28	外部委託のSLA不備	57	特定ノウハウについて外注依存。SIベンダーのコントロール不可
29	ソフト開発量の膨張で納期大幅延伸	58	行動規範順守が困難。機密情報漏えいの問題

表４－４　プロジェクトにおける問題事象と対策の事例集タイトル一覧（その３）
～IT プロジェクトの『見える化』下流工程編～

事例番号	事例タイトル	事例番号	事例タイトル
1	中核SEの離脱	40	大幅な工数増加と工程遅延
2	仕様凍結の遅延	41	業務担当リーダーの離脱
3	特定機能の進捗遅れ、品質劣化	42	協力会社のリーダー、中核要員がくるくる変わる
4	検収時期に一括納品できず	43	協力会社のスキルが低すぎて全く役に立たない
5	パッケージとカスタマイズのソースコード範囲が不明確	44	プロジェクト・マネージャ、リーダーと実務担当者の関係が悪化してコミュニケーションも作業も進まない
6	テスト工程で設計仕様変更が多発	45	役割分担が不明確
7	開発側の運用イメージ不足と仕様説明不足	46	複雑な開発組織内で、口頭による依頼、周知が多用され混乱
8	顧客の機能要件説明能力不足と開発側の顧客ニーズ把握ミス	47	事務方との連携が悪く、開発グループが客先で孤立
9	業務設計の中心となる設計リーダーが倒れて復帰できない	48	総合テストが品質不良で進まない
10	大幅な工数増加と工程遅延	49	仕様書と実装の差異に顧客がクレーム
11	大幅な工数増加による採算悪化	50	顧客側のプロジェクト・マネージャが交代してしまった
12	大幅な工数増加と工程遅延	51	協力会社が設計を完了した時点で次工程の開発を断ってきた
13	納期に間に合わない	52	業務有識者不在での総合テスト実施
14	仕様があいまいなままシステムを提供	53	契約前に作業完了
15	プロジェクト・マネージャが多忙	54	ベンダー提供の業務パッケージの品質不良による運用の中止
16	共同受注案件での開発分担不明確による混乱	55	営業時に営業SEが決めたブラウザが実用に耐えなかった
17	要件定義が不十分で、テスト工程にになってリリース必須機能の漏れが発覚	56	営業時に営業SEが決めたRDB製品の最新版を自社ミドルウェア製品がサポートしていなかった
18	テストの進め方が分からず、テスト計画もできない	57	パッケージ・ソフトの利用によりブラックボックス化が問題に
19	稼働１ヶ月半前でも結合・総合テスト計画が不十分	58	結合テストで重複障害やデグレードが多発
20	大幅な工数増加による採算悪化	59	自社のプロジェクト・マネージャが体調不良で倒れた
21	共通ライブラリのスケジュール遅れ、品質不良	60	顧客側の担当差が能力・経験不足でまったく役に立たない
22	見積もり時の前提想定不足	61	テスト工数を小さく見積もってしまった
23	メーカー提供プロダクトの不具合でシステムが動かない	62	開発チームへの指示系統が複数
24	システムの品質が悪い	63	個別チームの状況を把握していないため、総合テストに入っていいのか分からない
25	システムの品質が悪い	64	キーパーソンが疲弊している
26	大量印刷時間が長い	65	協力会社が多階層になり責任範囲があいまいに
27	ユーザーの研修で急に処理が遅くなる	66	実質的なマネジメントを協力会社の開発グループが実施
28	発注時の丸投げによって失敗	67	総合テストの進捗が悪い
29	パッケージのバグフィックスに時間がかかる	68	テスト作業に無駄が多い
30	大量データの処理方式の設計ミス	69	サービス開始の可否が判断できない
31	あいまいな仕様のまま実装し、テスト工程で要件不一致が多発	70	総合線表がない
32	外字印字の不良	71	顧客側と開発側のプロジェクトの一体感欠如
33	運用中の業務停止	72	オープンシステムのプロジェクト経験がない
34	指揮・命令系統が不明確	73	要求仕様が固まらないまま開発に突入
35	プロジェクト編成の中でプロジェクト・マネージャ的な人材が２人存在する形になり混乱	74	プロジェクト全体に疲弊感が漂っている
36	他業種から転身した元請け会社が未熟で混乱	75	外部調達のパッケージ製品の品質が悪くて使えない
37	DBMSの技術的な問題で対応できない	76	プログラムの品質が悪い
38	結合テストで品質不良が発生	77	パフォーマンス設計の不備
39	受注後の要件詳細化、基本設計体制に不備		

4.1. トリガー事象の抽出

「IT プロジェクトの『見える化』上流/中流/下流工程編」[5][6][7]における“プロジェクトにおける問題事象と対策の事例集”の各事例には (1) 問題事象名、(2) 問題事象の詳細、(3) 問題原因、(4) 実際に行われた対策、(5) 再発防止策の 5 項目が記述されている。本研究では、主に、(1) 、(2) 、(3) の記述に着目して事象の分類・整理を行った。図４－１に、実際の問題事象および KJ 法で作成したカードの一例を示す。

同様の方法で全 193 件の問題事象から約 400 件のカードを作成し、分類・整理を繰り返すことで、表４－１に示す 25 項目のトリガー事象を洗い出した。カードの作成や分類・整理においては、問題事象間の記述項目の違いや、書き手による内容のバラつきを吸収するために、なるべく一貫した表記・粒度になるように留意して作業を進めたことも付け加えておく。

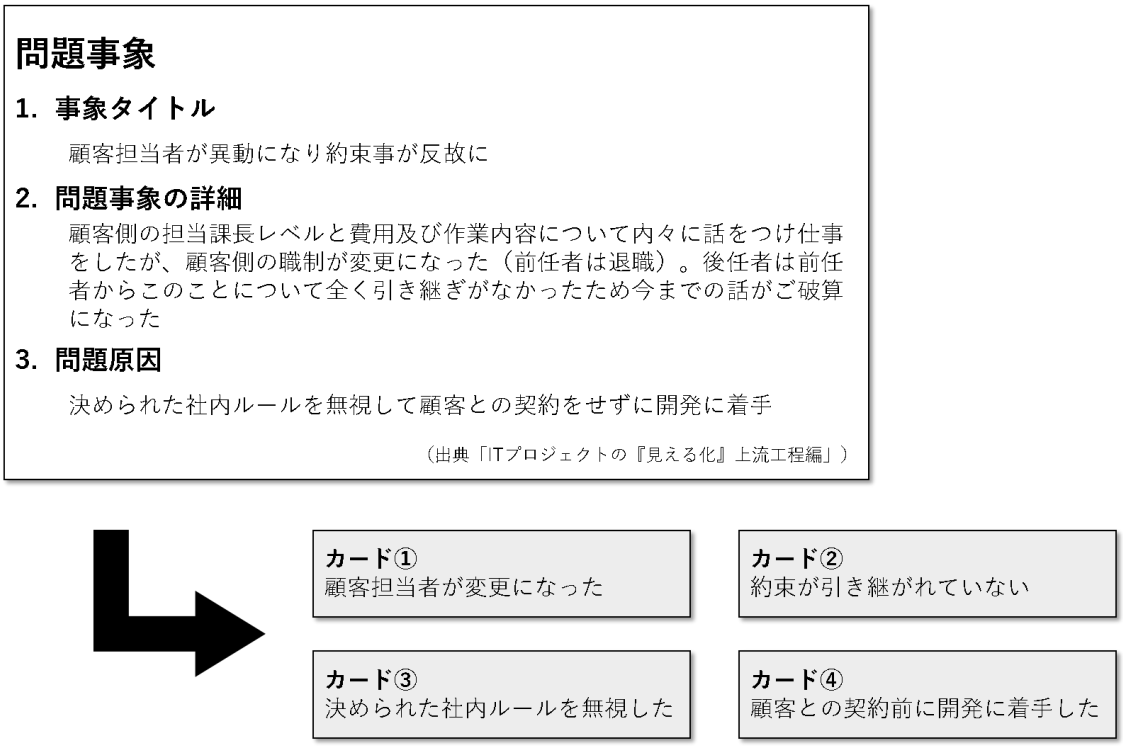


図４－１　「IT プロジェクトの『見える化』」問題事象と作成カード例

4.2. 他文献との比較

これまで、IT プロジェクトにおけるリスク事象の分類に関して、さまざまな研究が行われてきた。表４－５は、本研究のトリガー事象と他の代表的な文献[12][13][14][15][16][17]のリスク事象を比較した結果である。表中の〇は、他文献が本研究のトリガー事象と同様の内容を含んでいることを意味している。

表４－５　本研究のトリガー事象と他文献の比較結果

大分類	中分類	トリガー事象	文献比較					
			Bo [12]	Ro [13]	CR [14]	Ka [15]	IPA [16]	Mi [17]
内部要因	体制・スキル	リソース不足		○	○	○		
		役割・責任分担不明瞭						○
		要員の交代・離脱	○					
		要員のスキル不足	○	○	○			○
	管理・環境	マネジメント不良						○
		コミュニケーション不全					○	
		プロセス・ルールの不徹底				○		○
		作業環境の不整備・調整不足						
	経営・営業・他PJ	組織とPJ間の隔たり			○	○	○	○
		営業主導の受注				○		○
		他PJとの並行作業						○
外部要因	顧客	顧客都合の変更・遅延						
		顧客との認識の相違			○	○	○	○
	外注・調達先	安易な外注・調達先の選定		○			○	
		外注・調達先の管理不十分	○	○				
作業状況	見積・計画	受注前のリスク対策不足						○
		過少見積もり	○	○		○		○
		短納期	○	○	○	○		
	要求仕様	業務要件の理解不足					○	
		曖昧・不明確なスコープ					○	
		不完全な要求仕様			○	○	○	
		要求仕様の追加・変更発生	○	○	○	○	○	○
	設計・システム構築	利用技術の試行・検証不足						○
		技術難易度が高い	○	○	○			
		設計不良	○	○				

Bo：Boehm (1991) [12]、Ro：Ropponen (2000) [13]、CR：Chaos Report (1995) [14]、
Ka：Kasser (1998) [15]、IPA：IPA (2013) [16]、Mi：水野 (2005) [17]

比較の結果、過去文献に記載されたリスク事象は、文献ごとに粒度が異なり、項目的にもかなりばらつきがあることがわかった。一方、本研究のトリガー事象は比較した文献のリスク事象の全てをカバーする内容を含んでおり、システム開発プロジェクトの失敗に影響するリスク項目としてある程度の網羅性をもっていることが確認できる。また、他文献にない項目として、「作業環境の不整備・調整不足」「顧客都合の遅延・変更」の 2 つを追加した。後者は、他文献では「要求仕様の変更」などの項目に暗黙的に包含されているが、IPA/SEC「IT プロジェクトの『見える化』上流/中流/下流工程編」 [5] [6] [7]では、顧客による一方的な契

約内容の変更や、顧客都合による意思決定の遅れなどの問題が数多く見られたことから、今回、独立した項目として分離することとした。

5. 実プロジェクト事例による検証

実開発部門で異常予測を試行した多数のプロジェクトの中から、高リスクと判定された過去のプロジェクト事例 7 件を抽出し、それらで実際に発生したリスク事象と本研究の 25 項目のトリガー事象との対応関係について考察した。

5.1. 過去事例の分析方法

過去事例のリスク事象分析には、プロジェクトの振り返り会議用の資料（報告資料、議事録、など）を用いた。資料には、全体総括、コスト状況、不具合状況、個別振り返りなどの内容が含まれている。これらの情報に対して、(1) リスク事象の抽出、(2) リスク事象の分類、(3) 重要リスク事象の選別、という流れで作業を実施した。図５－１にリスク事象分析の例を示す。以下、分析手順について説明する。

② リスク事象の分類　　① リスク事象の抽出　　③ 重要事象の選別

ID	分類	リスク事象	出現回数	原因事象	重要事象
1	内部要因：管理・環境	使用器具の不足	1		○
2	内部要因：管理・環境	試験手順の属人化	1		○
3	外部要因：外注・調達先	外注依存の開発	1		
4	外部要因：外注・調達先	調達先への指示が不明瞭	1		
5	作業状況：見積・計画	リスク対策費不足	1		
6	作業状況：見積・計画	仕様曖昧のまま費用調整	1		○
7	作業状況：見積・計画	過少見積もり	2	6	○
8	作業状況：見積・計画	リスク考慮不足の試験計画	1		
9	作業状況：要求仕様	追加要求発生	1		
10	作業状況：設計・システム構築	未経験技術の使用	1		○
11	作業状況：設計・システム構築	新規設計作業が難航	1		○
12	顕在化した問題：不具合	不具合発生	3	10, 11	○
13	顕在化した問題：支出	不具合対応費増加	1	12	○
14	顕在化した問題：進捗	設計遅延	1		
15	顕在化した問題：進捗	不具合による後戻り作業発生	1	12	○
16	顕在化した問題：進捗	試験遅延	2	1, 2, 15	○
17	最終結果：コスト（C）	マイナス損益	1	13	○
18	最終結果：納期（D）	納期遅延	1	16	○
19	最終結果：納期（D）	顧客納期遅れ	1	18	○

図５－１　リスク事象分析の例

はじめに、プロジェクトの振り返り会議用の資料から可能性のある全てのリスク事象を抽出する（図５－１ ①リスク事象の抽出）。今回の分析では、1 プロジェクト当たりおよそ 30~50 個のリスク事象を抽出した。リスク事象抽出の際には、各リスク事象に関して、その内容だけでなく、リスク事象間の依存関係や資料中の出現頻度についても併せて記録する。また、リスク事象の名称としては、なるべくプロジェクト固有の表現を避け、一般的な用語を使用する。

つぎに、抽出したリスク事象を、カテゴリー分けする（図５－１ ②リスク事象の分類）。今回の分析では、表４－１の中分類 8 項目に、QCD 視点のプロジェクト中に顕在化した問題や最終結果を表す 6 項目と、“その他”の項目を追加した全 15 項目のカテゴリーを使用した。このカテゴリー分けにより、リスク事象の偏りや抜け漏れのチェック、トリガー事象との対応関係の確認などが可能となる。

さいごに、プロジェクト失敗に直接的に関連する重要リスク事象を選別する（図５－１ ③重要事象の選別）。上流のリスク事象と最終結果（プロジェクト失敗）のつながりに注意し、プロジェクト全体の状況を表現するのに必要十分なリスク事象を選ぶ。その際、出現回数の多いリスク事象や資料の全体総括で強調されているリスク事象を中心に、その前後関係にあたるものを選択するとよい。

図５－２は、図５－１の分析結果から作成したリスク事象ネットワーク図である。リスク事象ネットワーク図では、1 つのリスク事象を 1 つの四角で記載し、リスク事象間の因果関係を矢印で結んで表現している。こうした図を用いて全体を俯瞰することで、プロジェクト失敗の根本原因やリカバリーのタイミング、適切なリスク対応策などの検討が容易になることが期待できる。

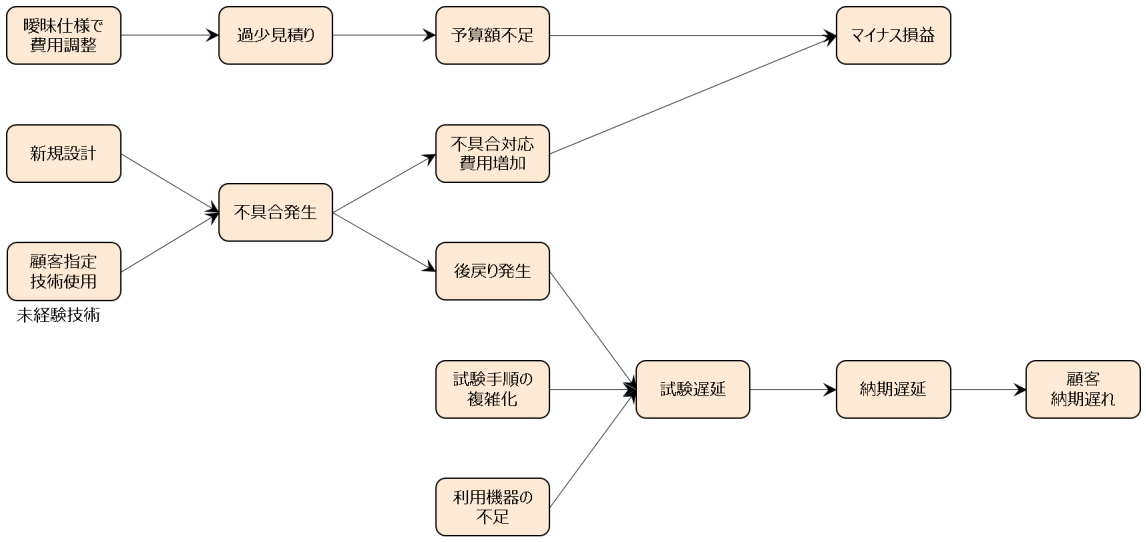


図５－２　リスク事象ネットワーク図

5.2. トリガー事象の事例適用結果

異常予測で高リスクとなった過去事例 7 件に対して、25 項目のトリガー事象の当てはまりを検証した結果を表５－１に示す。表５－１において、各事例から抽出したリスク事象がトリガー事象と一致している場合には〇で示し、特に重要リスク事象と一致している場合には◎を記入している。

表５－１の結果から、25 項目のトリガー事象の内、23 項目が実際のプロジェクト事例でリスク事象として抽出されていることを確認した。このことから、本研究のトリガー事象には

実際のプロジェクトで発生するリスク事象においてある程度の網羅性があり、実用的な内容であると考察できる。また、今回の事例 7 件の特徴として、「コミュニケーション不全」と「他 PJ との並行作業」に起因する問題が多いことが読み取れる。詳細を確認したところ、「コミュニケーション不全」については、ハードウェアとソフトウェア担当者間の認識の齟齬によるトラブルが複数あった。「他 PJ との並行作業」については、派生開発プロジェクトにおいて派生元製品の不具合が派生先の開発に影響を及ぼしているケースが確認された。一方、今回の事例には、「体制・スキル」に関連するリスク事象や、「マネジメント不良」、「組織と PJ 間の隔たり」、「営業主導の受注」、「曖昧・不明確なスコープ」などはあまり含まれていなかった。これらのトリガー事象が少なかった原因として、今回のインプットデータはプロジェクトメンバーが自らのプロジェクトに対して省察した結果であり、個々のプロジェクトの範囲を超えた組織レベルの問題等については、プロジェクト内部からリスクとして挙げにくかった可能性があると考えられる。このように、トリガー事象の発生状況や分布を確認することで、組織におけるリスクの傾向や特徴などの分析・考察が可能となる。

表５－１　実プロジェクト事例とトリガー事象との対応関係

大分類	中分類	リスク要因	PJ1	PJ2	PJ3	PJ4	PJ5	PJ6	PJ7
内部要因	体制・スキル	リソース不足		◎				○	
		役割・責任分担不明瞭				○		○	
		要員の交代・離脱							
		要員のスキル不足				◎			
	管理・環境	マネジメント不良				○			
		コミュニケーション不全			◎	◎	○	◎	○
		プロセス・ルールの不徹底		○			○	○	○
		作業環境の不整備・調整不足	◎	○			○	○	
	経営・営業・他PJ	組織とPJ間の隔たり							
		営業主導の受注					○		
		他PJとの並行作業		◎	◎			◎	◎
外部要因	顧客	顧客都合の変更・遅延	○			○	◎	◎	◎
		顧客との認識の相違				◎			
	外注・調達先	安易な外注・調達先の選定		◎		◎	◎	○	
		外注・調達先の管理不十分	○	○		◎			
作業状況	見積・計画	受注前のリスク対策不足	◎			○	○		
		過少見積もり	◎			○			○
		短納期		◎					
	要求仕様	業務要件の理解不足				○			
		曖昧・不明確なスコープ							
		不完全な要求仕様	◎			◎			
		要求仕様の追加・変更発生	○		○			◎	
	設計・システム構築	利用技術の試行・検証不足	◎	◎					
		技術難易度が高い	○		◎		○		
		設計不良	◎		○		◎	◎	○

さらに、リスク事象間の因果関係を図化したリスク事象ネットワーク図を用いることにより、リスクに関する経験・ノウハウの効率的な知識継承が期待できる。すなわち、因果関係を図化することで、失敗事例におけるリスク事象の流れを追うことが容易になり、失敗事例を疑似体験しやすくなる。組織内でリスク事象ネットワーク図を蓄積し、プロジェクトマネジメント教育に取り入れることにより、失敗事例の疑似体験を通じて、より定着度の高い教育コンテンツを提供できる可能性がある。また、今回は失敗事例を対象としたが、成功事例についても同様のやり方で分析することで、過去の成功事例を参照して現在のプロジェクトに取り入れることが容易になると期待される。

今回の検証により、リスク事象分析、および、リスク事象ネットワーク図の有効性を確認することができたが、分析の方法や手順についてはまだ課題も残っている。今回の分析作業は筆者を含む 4 名で実施したが、同じ事例であっても異なる作業者が抽出したリスク事象にはかなりバラつきがあった。分析の客観性を確保するためには、用語や概念を明確に定義したうえでリスク事象の表現や抽象度をそろえ、第三者によるレビューを実施するなど、人に依存する部分をなるべく減らす工夫が必要である。

6. トリガー事象の活用方法

プロジェクト異常予測の結果からプログラム実行に影響を及ぼすリスクを検知し、その対応まで繋げるには、「危ない」プロジェクトを検知した際に、高リスクとなった原因の特定と、具体的に「何をすべきか」を提案する仕組みが必要である。本研究のトリガー事象を異常予測の枠組みの中で活用する方法の１つとして、類似プロジェクト検索によるリスク対策の推薦機能が考えられる。

類似プロジェクト検索によるリスク対策の推薦機能は、異常予測の結果とリスク対策に関する組織的に蓄積してきたノウハウを、本研究のトリガー事象を用いて対応付けることで実現できる。図６－１に類似プロジェクト検索によるリスク対策の推薦機能を追加したプロジェクト異常予測のイメージを示す。高リスクが予測された進行中のプロジェクトに対して、ナイーブベイズ識別器の中間データのコサイン類似度を求めることにより過去の類似プロジェクトの検索が可能である。過去の代表的なプロジェクト事例について本研究の方法で分析し、各トリガー事象の対応策を事前にリスト化しておくことで、進行中のプロジェクトのリスク要因を推測し、妥当なリスク対応策を提示できる可能性がある。異常予測の結果にこれらの情報が併せて提示されることで、プログラムマネジャーがプログラム実行に影響を及ぼすリスクを検知し、適切な対応策を施すための意思決定を支援することが期待できる。

また、高リスクプロジェクトの原因分析を効率的に行うには、予測モデルで使用する説明変数が本研究のトリガー事象とある程度紐づいていることが望ましい。そのためには、組織として継続的に異常予測モデルを見直し、データ収集コストと予測への寄与度のバランスに考慮しつつ、トリガー事象に関連するデータを少しずつデータベースに取り込んでいくことが重要である。

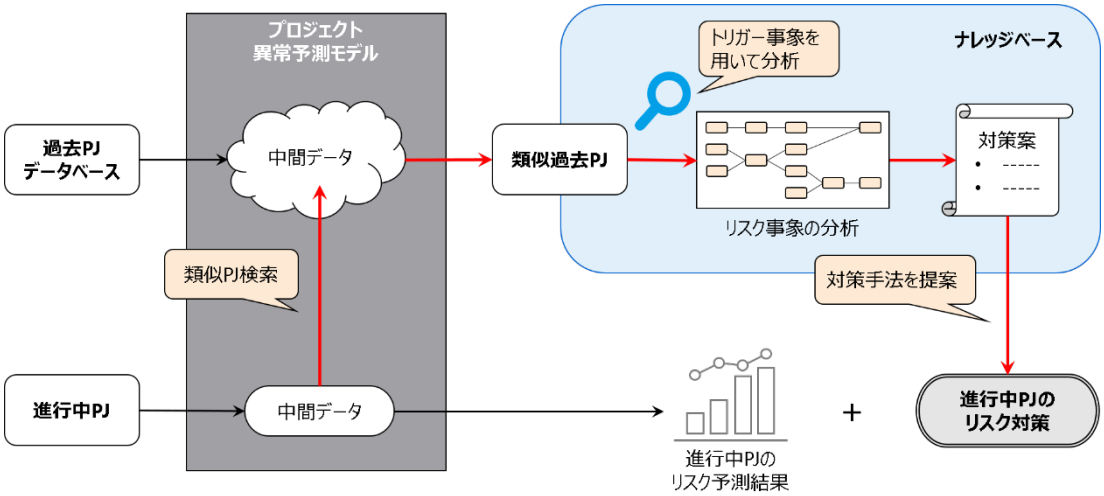


図６－１　類似プロジェクト検索によるリスク対策の推薦機能

7. おわりに

プログラムマネジメントでは、同時並行で進行する多数のプロジェクトを管理する必要があるが、従来の人手に頼ったマネジメント手法だけでは、全てのプロジェクトに十分に目が行き届かず、リスクへの対応が後手に回るという課題があった。本課題に対し、機械学習を用いたプロジェクト異常予測をプログラムマネジメントに適用し、同時進行する多数のプロジェクトから、プログラムの成否に影響を与えるリスクを内包するプロジェクトを自動で検知することを試みている。実開発部門でのシステム開発プロジェクトに対して異常予測を試行した結果、各プロジェクトのリスクの大小は高精度で予測できるようになった。しかし、現在のプロジェクト異常予測の枠組みでは、実際にプログラムに影響を及ぼすリスクとして判断し、対処するための情報が不足しており、予測結果に基づくリスク対処までにはつながりにくいという課題が新たに浮上した。そこで、本研究では、システム開発プロジェクトの失敗の主要因となる 25 項目のトリガー事象を洗い出し、これらのトリガー事象に対して、プロジェクトリスクに関するソフトウェアエンジニアリングの代表的な文献と比較し、項目としてある程度の網羅性があることを確認した。また、プロジェクト異常予測で実際に高リスクとなった実プロジェクト事例への当てはまりを評価し、トリガー事象の大部分の項目が実際のプロジェクトで発生していることも確認した。この結果から、本研究のトリガー事象が、プロジェクト異常予測で高リスクとなったプロジェクトのリスク事象分析に有効であることがわかった。

トリガー事象の一部はシステム開発プロジェクトに限定される項目もあるが、その他一般的なプロジェクトに共通する項目も多く含まれている。また、プロジェクト異常予測はシステム開発プロジェクトに限らず適用可能であり、今後は、プログラムマネジメントで扱う幅広いプロジェクトに対して異常予測を適用し、トリガー事象も合わせて一般化していきたいと考えている。

参考文献

[1] 日本プロジェクトマネジメント協会「改訂 3 版 P2M プログラム&プロジェクトマネジメント標準ガイドブック」、日本能率協会マネジメントセンター、2014

[2] 国際 P2M 学会ウェッブサイト http://www.iap2m.org/pdf/p2mconcept200906.pdf

[3] 武富為嗣「P2M によるリスクマネジメントのフレームワーク」、国際 P2M 学会誌、Vol.4、No.2、pp.93-102、2010

[4] 森　俊樹、覚井真吾、田村朱麗、藤巻　昇「プロジェクト失敗リスク予測モデルの構築」、プロジェクトマネジメント学会誌、Vol.15、No.4、pp.3-8、2013

[5] 独立行政法人情報処理推進機構（IPA）ソフトウェア・エンジニアリング・センター（SEC）「IT プロジェクトの『見える化』～上流工程編～」、日経ＢＰ社、2007

[6] 独立行政法人情報処理推進機構（IPA）ソフトウェア・エンジニアリング・センター（SEC）「IT プロジェクトの『見える化』～中流工程編～」、日経ＢＰ社、2008

[7] 独立行政法人情報処理推進機構（IPA）ソフトウェア・エンジニアリング・センター（SEC）「IT プロジェクトの『見える化』～下流工程編～」、日経ＢＰ社、2006

[8] 佐藤達男、亀山英雄「複雑な都市インフラ開発における統合リスクマネジメントの有効性」、国際 P2M 学会誌、Vol.9、No.2、pp.123-135、2015

[9] 岩崎祐子、渡辺研司「プロジェクトファイナンスにおけるリスクマネジメントに関する考察」、国際 P2M 学会誌、Vol.12、No.2、pp.103-118、2018

[10] 越島一郎「IMHO：P2M のためのリスクマネジメント手法」、P2M マガジン、No.9、pp.56-62、2020

[11] 清水基夫「P2M における戦略とリスクのマネジメントに関する一考察」、国際 P2M 学会誌、Vol.5、No.1、pp.129-138、2010

[12] B. W. Boehm. “Software Risk Management: Principles and Practices”，IEEE Software，Vol.8，pp.32-41，1991

[13] J. Ropponen and K. Lyytinen. “Components of Software Development Risk: How to Address Them? A Project Manager Survey”，IEEE Transactions on Software Engineering，Vol.26，No.2，pp.98-112，2000

[14] The Standish Group International, Inc. “The CHAOS Report (1994)”，1995 https://www.standishgroup.com/sample_research_files/chaos_report_1994.pdf

[15] J. Kasser and V. R. Williams. “What do you mean you can't tell if my project is in trouble?”，DoD Software Tech News，Vol.2，No.2，1998

[16] 独立行政法人情報処理推進機構（IPA）ソフトウェア・エンジニアリング・センター（SEC）「IT プロジェクトのリスク予防への実践的アプローチ」2013 https://www.ipa.go.jp/sec/softwareengineering/reports/20130327_3.html

[17] 水野　修、阿部誠也、菊野　亨「プロジェクト混乱予測システムのベイズ識別器を利用した開発」、SEC journal、Vol.1、No.4、pp.24-35、2005

[18] 浜野康裕、天嵜聡介、水野　修、菊野　亨「相関ルールマイニングによるソフトウェア開発プロジェクト中のリスク要因の分析」、コンピュータソフトウェア、Vol.24、No.2、pp.79-87、2007

[19] T. Mori, S. Tamura and S. Kakui. “Incremental Estimation of Project Failure Risk with Naive Bayes Classifier”, In 2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, pp.283-286, 2013

[20] 大島丈史、内平直志「プロジェクトマネジメントへの AI 活用の知識分類モデル　IT 企業における AI 適用方策の研究」、国際 P2M 学会誌、Vol.13、No.1、pp.121-141、2018

[21] D. D. Lewis. “Naive (Bayse) at forty: the independence assumption in information retrieval”, In Proc. of the European Conference on Machine Learning (ECML), pp.4-15, 1998

[22] 川喜田二郎「発想法　改版 - 創造性開発のために」、中央公論新社、2017

査読　　2021 年 12 月 24 日

受理　　2022 年 3 月 15 日