

コミュニケーション履歴を用いたプロジェクト管理における 問題の早期検知に向けて

赤木 里騎¹ 戸田 航史¹ 吉田 則裕² 伏田 享平³ 滝本 雅之³

概要: IT システムの開発においてプロジェクトマネージャの役割は多岐に渡る。その中で、プロジェクトマネージャがプロジェクトで発生するやり取り（コミュニケーション）全てについて詳細を把握することは困難である。本研究ではコミュニケーションの履歴を用いてプロジェクト問題化につながる要因を分析した。コミュニケーション履歴としては QA 票を用い、その分析にあたっては、問題化しなかった 1 プロジェクトと、問題化した 1 プロジェクトの計 2 プロジェクトを対象とした。問題化しなかったプロジェクトに対して時間帯と QA 票の状態に着目しメトリクス分析を行った結果、個人の行動は問題化にさほど影響を与えていないことがわかった。問題化したプロジェクトに対しては、データの各メトリクスを目的変数として決定木分析を行った。その結果、人的な要因と工程要因が目的変数として採用したメトリクスに影響を与えていることがわかった。

A communication log analysis for early phase detection of project issues

RIKI AKAGI¹ KOJI TODA¹ NORIHIRO YOSHIDA² KYOHEI FUSHIDA³ MASAYUKI TAKIMOTO³

1. はじめに

IT システムの開発において、プロジェクトマネージャ（以下、PM と略す）の役割は多岐にわたる。プロジェクト立ち上げ時の見積もり、計画策定から、実行時の品質、リスク、コミュニケーションのコントロールなど、マネジメントすべき領域は広範囲にわたる。特に近年ではシステム開発は複雑化し、場合によっては同時に複数のプロジェクト管理が求められるなど、PM によるプロジェクト管理は困難さを増している。そのため、PM がプロジェクトを細部にわたって詳細に把握することは、その役割の多さから困難であるため、問題化するプロジェクトを早期に発見するための枠組みが必要である。

本研究ではプロジェクトが問題化する兆候を早期に検知するため、プロジェクト上流工程におけるコミュニケー

ションに着目する。プロジェクトの上流工程では、システムの要件や仕様を確定するため、受発注者といったステークホルダーの間で詳細な情報の要求や確認といったコミュニケーションが発生する。このようなプロジェクト実施中に発生するコミュニケーションがどのように行われているかを分析することで、ステークホルダー間の認識齟齬や仕様検討漏れといった、問題化の兆候につながる事象を検知できると考えた。さらに、上流工程のようなプロジェクトの初期段階で問題化の予兆を PM に対して提供することで、プロジェクトの問題化を未然に防ぐことができると考えられる。

本研究では、コミュニケーションの履歴を用いてプロジェクト問題化につながる要因を分析した。分析にあたっては、実際の商用開発プロジェクト 2 件（うち、1 件は実際に問題化したプロジェクトである）で記録されたコミュニケーション履歴を対象として、コミュニケーションの状況を把握するメトリクスを定義し抽出した。次に、問題化の要因となりうると考えられる、コミュニケーションが発生している時間帯ややり取りの回数に着目して分析を実施

¹ 福岡工業大学
Fukuoka Institute of Technology

² 名古屋大学
Nagoya University

³ 株式会社 NTT データ
NTT DATA Corporation

表 1 A 案件の QA 票に記載されている項目

項目名	概要
ID	QA 票の管理番号
タイトル	QA 票のタイトル
作成者	QA 票の最初の作成者
担当者	QA 票の担当者
変更者	QA 票を変更した者
変更日	QA 票を変更した日時情報
状態	QA 票の対応状況
詳細情報	自然言語で記述された対応内容

した。また自然言語処理に基づきコミュニケーションの内容を定量化したメトリクスを加え、問題化に繋がる要因を探るために決定木分析を行った。

以降、2 節で分析に用いたデータセットおよびメトリクスについて述べる。3 節で分析結果について述べ、4 節で考察する。5 節で関連研究を示した後、6 節でまとめと今後の展望について述べる。

2. 分析に用いたデータ

本研究では分析対象のコミュニケーション履歴として、株式会社 NTT データで実施した IT システム開発プロジェクトにおける QA 票のデータを用いた。仕様の確認や開発環境設定の不備など、受発注者間において確認が必要な事項が発生した際、プロジェクトにおける公式な記録として QA 票が起票される。QA 票に記載される項目はプロジェクトにより異なるが、多くの場合確認事項とそれが発生した日付、確認事項に対する回答内容とその回答日が記録されている。今回の分析では、問題化の定義を「プロジェクト計画時の品質、コスト、納期を達成できなかった」とする。その上で、QA 票から問題化するプロジェクトの特徴を分析するために、問題化しなかったプロジェクト（A 案件）と問題化したプロジェクト（B 案件）、計 2 プロジェクトのデータを使用した。

2.1 QA 票に記載されている項目

A 案件のデータは 138 件の上流工程における QA 票をまとめたものである。分析対象期間は、受発注者間のやりとりが開始してからの 1 ヶ月間とした。QA 票に含まれるデータ一覧を表 1 に示す。QA 票には、ある作業に対して受注側と発注側でのやり取りに関するテキスト情報、記入日・回答日などの日付情報、QA 票の状態が含まれている。QA 票の状態は、作成された時点を「新規」として、対応者が割り当てられ次第「対応中」に変更され、対応が完了すると「完了」に変更される。A 案件ではタイトルや状態、詳細情報など、項目が更新されるたびに新しいレコードとして記録される。よって、同一 ID に対するレコードの数を数え上げることで、1 件の QA 票で発生したやり取りの回数を抽出できる。

表 2 B 案件の QA 票に記載されている項目

項目名	概要
発生起因工程	質問・確認対象の事象が発生した工程
サブシステム	質問・確認対象のサブシステム
サブ業務	質問・確認対象のシステムが扱う業務
機能名	質問・確認対象の業務における機能
質問・確認内容	発注者側への具体的な質問内容。受注者側が記載する
記入者	質問・確認内容の記入者
記入グループ	記入者が属するグループ
記入日	質問・確認内容の記入日。複数回やりとりした場合は、最終更新日を記入する
回答者	質問・確認項目への回答者
回答内容	質問・確認項目に対する具体的な回答内容
回答日	回答内容の記入日。複数回やりとりした場合は、最終回答日を記入する
発見事象	確認結果が不具合であった場合の具体的な事象
仕様変更番号	確認の結果仕様変更が発生した場合に採番される管理番号
備考欄	質問・確認内容に関連する補足事項

表 3 B 案件の QA 票より抽出したメトリクス

項目名	概要
作業期間	記入日と回答日の差分
仕様変更あり	仕様の変更あり
仕様変更なし	仕様の変更なし
応答回数	受発注者間のやりとりの回数
NE 回数	質問・確認内容に含まれる固有表現の回数
確認メトリック	質問・確認内容に含まれる確認トピック数
拒否メトリック	質問・確認内容に含まれる拒否トピック数
謝罪メトリック	質問・確認内容に含まれる謝罪トピック数
要求メトリック	質問・確認内容に含まれる要求トピック数

B 案件のデータには 1780 件の QA 票が記録されている。B 案件の QA 票に記載されている項目を表 2 に示す。B 案件の QA 票には QA でのやり取りに加えて、QA の結果不具合と判断された場合の発生原因や工程が記録されている。B 案件では複数回のやり取りがあった場合には、備考欄にその履歴が追記されていく。そのため、1 件の QA 票で発生したやりとりの回数は、備考欄の内容を確認することで抽出できる。

2.2 自然言語処理によるコミュニケーション内容の定量化

B 案件は記録されている QA 票の数が多いことから、自然言語処理によりコミュニケーション内容の定量化を試みた。入手した QA 票に対して下記の処理を行い、表 3 に示すメトリクスを追加した。

- 「仕様変更番号」項目に記載ある QA 票を「仕様変更あり」、記載のない QA 票を「仕様変更なし」に分類した。
- 回答日と記入日の差分から作業期間を算出した。
- 「備考欄」に記載されている受発注者の応答履歴から

応答回数を算出した。

- 「質問・確認内容」、「回答内容」に対して自然言語処理を行い、「NE回数」「確認メトリック」「拒否メトリック」「謝罪メトリック」「要求メトリック」を算出した。

B 案件では開発当初に受発注者間で仕様についての確認が不十分であったため、仕様確認に関する QA 票でのやり取りを通して設計不足や設計漏れが検知された。当初定義していた仕様に追加、変更が必要と判断された場合には、QA 票に「仕様変更番号」を記載し QA 票と仕様変更内容を一貫して管理できるようにしている。応答回数については、受注側から仕様について発注者側に質問をすると応答回数は 1 回となり、発注者側が受注者側に対して質問に回答すると応答回数は 2 回となる。以降、同一の QA 票についてさらに質問が発生した場合には、応答回数は増加することとなる。「質問・確認内容」と「回答内容」の 2 つの項目から得られるデータについては 2.3 にて詳細を説明する。

2.3 自然言語処理

B 案件の「質問・確認内容」、「回答内容」の項目に記述される受注側と発注側のやり取りのテキストデータに対してトピック抽出を行うことで、「確認」や「要求」、「拒否」、「謝罪」メトリックを計測した。トピックとは、文書中出现する単語の確率分布であり、それぞれのメトリクスは、「仕様確認」、「相手に対する作業の要求」、「要求に対する拒否」、「謝罪」に対応するトピックが生起する確率である。例えば「相手に対する作業の要求」に対応するトピックとして「願い、ください」が出現する確率分布、「仕様確認」に対応するトピックとして「教授、教示」が出現する確率分布がそれぞれ挙げられる。各文書に含まれるトピックの抽出については、米国マサチューセッツ大学アモース校が開発した自然言語解析ツール MALLET [1] を使用した。

NE 回数はやり取りの文章中において固有表現が出現する回数を表す。固有表現とは地名や組織名等の固有名詞や、時間や割合等の時間表現や数値表現を表す。固有表現の出現数は対象文書の専門性や具体性の度合いを測る尺度に使用できるとされている [3]。固有表現の認識はコーパス情報を教師データとする機械学習技術（サポートベクタマシン [2] など）及び構文間の関係を解析する技術（係り受け解析）の組み合わせにより行われる。本研究では京都大学を中心に開発されている日本語構文・格・照応解析システム KNP の固有表現認識機能を用いて、固有表現の出現回数を計測した。笹野らの研究 [3] において KNP の固有表現機能は、既存の代表的な 7 つの手法と比較して優れていることが示されているため、本研究において採用した。

3. QA 票に対する分析

2 節で示した QA 票に対してメトリクス分析を行った。プロジェクトが問題化する際、受発注者間において確認が

必要な事項が発生する時間帯や確認事項に対する回答が遅延すると考えられる。A 案件については、QA が行われている時間帯や、QA 票の状態に着目して分析した。B 案件については、仕様変更などの問題化につながる事象の要因を探るため、決定木分析を行った。

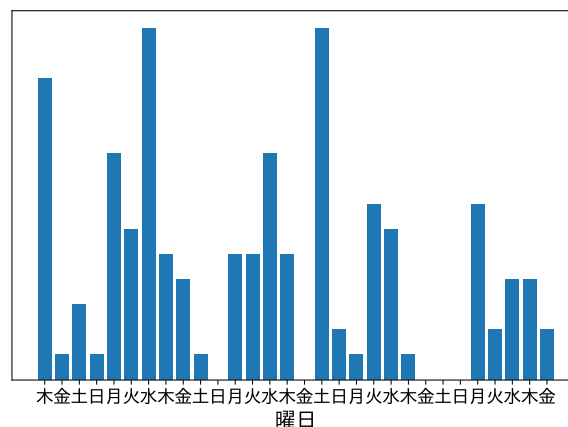


図 1 曜日ごとの QA 票変更回数

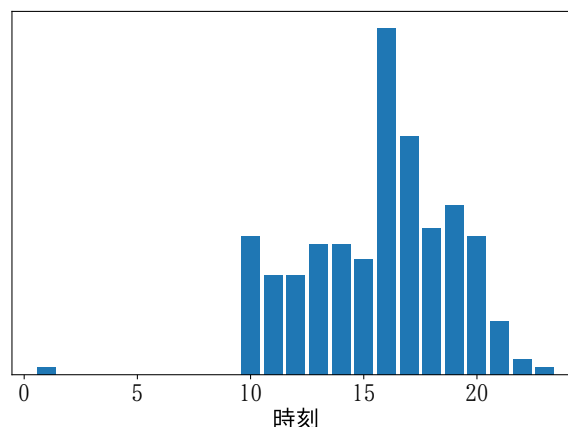


図 2 時間帯ごとの QA 票変更回数

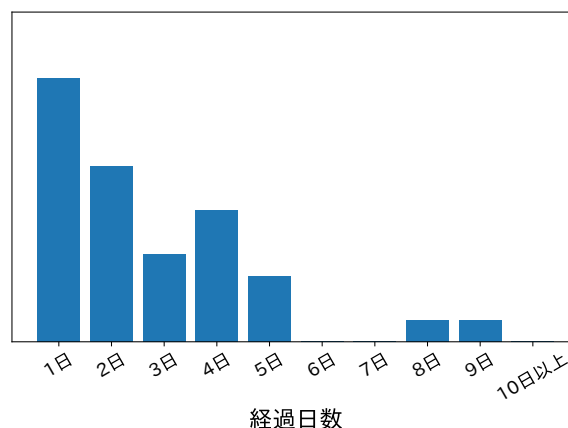


図 3 起票から対応中になるまでの日数

3.1 A 案件の分析詳細と結果

表 1 のメトリクスに対して、作業の状態が変更された時間帯や状態の変更回数、状態の変更にかかる日数などのヒストグラムを作成した。図 1 は分析対象期間中の QA 票の変更回数割合を示している*1。縦軸はある期間中の変更回数の割合を示し、横軸は各日（1 日単位）である。ここで変更回数とは、QA 票の状態や詳細情報が更新された回数である。A 案件では QA 票に変更が発生するたびにその変更日が記録されるため、記録されている回数を集計することで変更回数を集計できる。変更回数割合は、1 日の変更回数を分析対象の期間の全ての変更回数で割ることで算出した。図 2 は時間帯ごとの QA 票変更回数の割合を示している。縦軸はある期間中の変更回数、横軸は 0 時から 23 時までを 1 時間単位で区切っている。図 3 は新規から対応中になるまでの日数、図 4 は対応中から完了に至るまでの日数を示している。図 3、図 4 いずれも縦軸は分析期間中に発生した QA 票の状態変更回数の割合、横軸は状態が変更されるまでに要した日数を示している。

図 1 から、1 週間のうち週初めや週終わりよりも、中日に当たる火、水、木曜日に変更が増える傾向にある。図 2 から、QA 票の変更は 10 時から 20 時までに集中しており、特に 16、17 時に多く発生している。図 3 から、QA 票の多くは起票から 5 日以内には対応中（担当者が割り当てられている）に変更されている。図 4 から、QA 票の多くは対応中になってから 1 日以内に完了している。

3.2 B 案件の分析詳細と結果

B 案件に対して、問題化につながる要因を定量的に検出することを目的として分析した。分析にあたり、自然言語情報を利用したメトリクスも用いて決定木分析を行い、各目的変数について影響の大きいメトリクスを検出した。決定木分析を採用した理由は、要因が重要度によって階層的

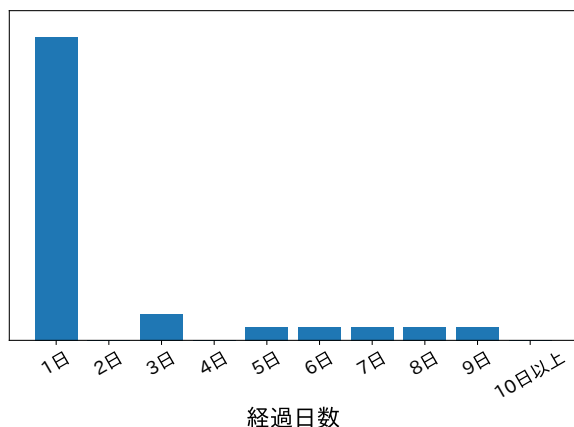


図 4 対応中から終了するまでの日数

*1 具体的な変更回数は開示できないため、以降図中に現れる縦軸の値は全て相対値として表す。

表 4 決定木の目的変数と重要要因

目的変数	重要要因
発生起因工程	記入者, サブシステム, 発見事象, 作業期間
サブシステム	記入者, 記入グループ, 機能名, 回答者
サブ業務	機能名, 記入者, 記入グループ, サブシステム
機能名	サブ業務, 記入者, 発生起因工程
作業期間	記入者, サブ業務, 発生起因工程, 回答者
発見事象	発生起因工程, 記入者, 回答者
仕様変更あり	確認メトリック, 記入者, 回答者
仕様変更なし	記入者, 発生起因工程, 応答回数, 確認メトリック
応答回数	確認メトリック, 記入者, 回答者, サブ業務
NE 回数	記入者, 発生起因工程, 回答者, 確認メトリック
確認メトリック	発生起因工程, 記入者, 応答回数, サブ業務
拒否メトリック	記入者, 作業期間, 確認メトリック, 発生起因工程
謝罪メトリック	応答回数, 記入者, 発生起因工程
要求メトリック	サブ業務, 応答回数, 記入者, 発生起因工程

表 5 決定木中のノードとして出現した説明変数の出現回数

説明変数	出現回数
記入者	14
発生起因工程	9
回答者	6
サブ業務	5
確認メトリック	5
応答回数	4
記入グループ	2
機能名	2
サブシステム	2
作業期間	2
発見事象	1

に得られるため結果の解釈が容易であり、問題化する原因の把握に有用と判断したためである。決定木分析では、QA 票の項目のうちある 1 つの項目を目的変数、その他の項目を説明変数として決定木を作成した。目的変数としては表 2、表 3 に示したメトリクスから記入者、記入グループ、回答者、質問・確認内容、回答内容、記入日、回答日、仕様変更番号、備考欄を取り除いた、計 14 メトリクスを採用した。

決定木分析の結果得られた決定木の例として、目的変数として拒否メトリックを採用したものを図 5 に示す。決定木は階層構造で示され、上部に現れるノード（楕円で囲まれた説明変数）がより目的変数に寄与する説明変数と考える*2。図 5 の最下部に示した箱ひげ図は、各終端ノードにおける拒否メトリックの分布およびデータの個数を示す。図 5 からは、拒否メトリックの多寡に影響する主要因は記入者であることがわかる。また、作業期間（回答にかかった時間）や確認メトリック、発生起因工程も拒否メトリックの多寡に影響を与えることがわかる。このように決定木を用いることでそれぞれの要因がどのようなルールに従って分類されているかを把握できる。

*2 各ノードに現れる閾値は開示できないためマスクしている。

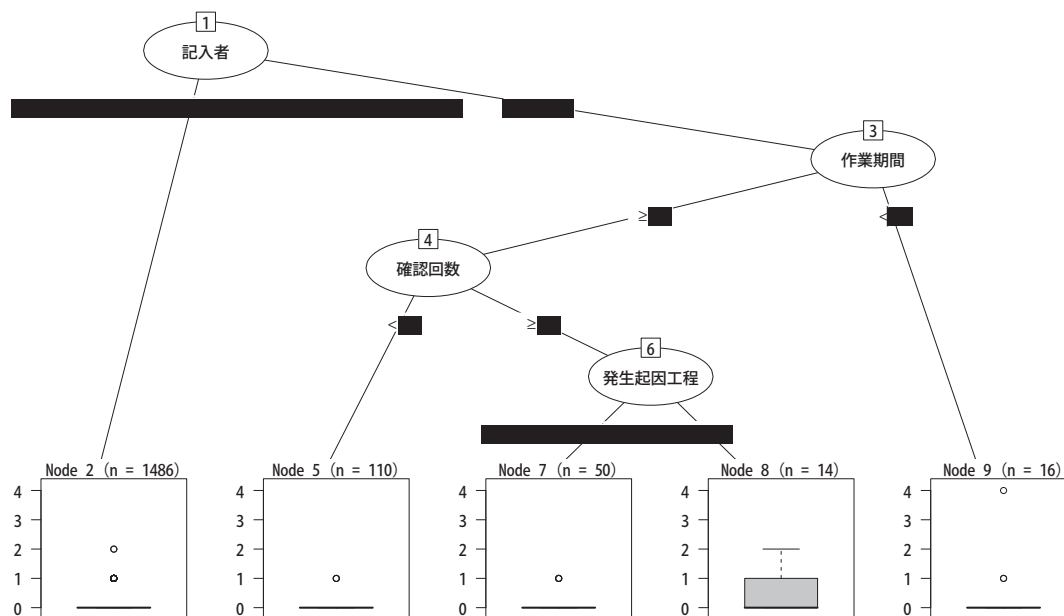


図 5 目的変数が拒否メトリックの場合の決定木

各目的変数に対して得られた決定木のサマリを表 4、表 5 に示す。表 4 は各目的変数に対して得られた決定木に現れるノードのうち、上位のものをまとめたものである。すなわち、目的変数に対して影響を与える説明変数を示している。また、表 5 は各説明変数が決定木中出现した回数を集計したものである。決定木は分析対象とした目的変数の数、すなわち計 14 種作成される。表 5 から、記入者は全ての決定木中出现していることが分かる。また、発生起因工程は 9 回、回答者は 6 回と、半数近くの決定木中出现していた。これらの結果より問題化したプロジェクトにおいては、人的要因と工程要因が目的変数に影響していることが分かった。

4. 考察

QA 票に関連する作業時間について、図 1 に示したとおり 1 週間のうち火、水、木曜に QA 票変更回数が増える傾向にある。これは週初めに 1 週間の計画をたてて開発を進めるために、週の中日に QA 票の変更回数が増える傾向にあると考えられる。また 1 日の中での作業時間帯は、図 2 より 16、17 時に変更回数が増えている。これは、標準的な業務時間が終了する（退勤する）前に、その日に行った作業内容を QA 票に反映するためと考えられる。

次に QA 票の対応期間に着目する。3.1 で示した結果より、QA 票の状態が変更されるまでの日数（対応にかかる時間）は問題化に影響があるとは言えない。ただし、図 3 に示したように、QA 票の状態が新規から対応中へ変更されるまでにかかる日数については、大多数が 5 日以内であったため、5 日を超えると問題化の原因になりうると考えられる。また図 4 に示したように、QA 票の状態が対応中か

ら解決済みに変更されるまでにかかる日数は、大多数が 1 日以内であったことから、2 日を超えると問題化の原因になりうる。対応中から解決済みへの変更にかかる日数が 1 日以内のものが多く理由として、QA 票の状態に関する運用が統一されていないケースも考えられる。これは、状態を解決済みに変更するタイミング、基準が明確になっていないため、PM が想定していない使い方をされていたことが考えられる。

B 案件の分析結果からは、人的および工程要因がプロジェクトの問題化に影響を与えることが示唆される。また確認メトリックや応答回数といった受発注者間のコミュニケーション内容を定量化したメトリクスは、同時にかつ多数出現している。これは事前に定義した仕様が曖昧であるため、受発注者間で確認が頻発していることを示唆している。このような傾向を今回提案するメトリクスを用いてプロジェクトの初期段階で検知することで、PM は早期に該当する課題について重点的に対処することが可能になると考えられる。

本研究では IT システム開発におけるコミュニケーション履歴を対象に分析した。受注側と発注側とのやりとりから自然言語処理を行い、各メトリクスから人的な要因が B 案件の問題化に繋がる傾向がみられた。ただし、問題化の要因はデータセットの所有者（会社）に依存すると考えられる。例えば、今回分析対象とした QA 票ではプロジェクトで発生した確認項目のみを扱っているが、組織によっては不具合（バグ）や新規要件に関する情報も同一の QA 票で扱う場合があると考えられる。また、問題化の定義や QA 票の運用プロセスによっても結果は異なってくると考えられる。今後、他のデータセットに対しても同様の検証

を行い、その妥当性を検証することが必要である。

5. 関連研究

本研究ではプロジェクト実施上の課題の早期検知を目的として、開発の過程で得られる QA 票を対象に分析を行った。プロジェクトの問題の検知に関する既存研究としては、玉田らのスケジュールの遅延リスク検出についての研究が挙げられる [4]。スケジュールの遅延は問題化したプロジェクトに発生する現象の 1 つであり、遅延のリスク検出を目的としたソフトウェア開発プロジェクトからのデータの自動収集と、そのデータを時系列に基づいて可視化するツールを開発している。収集・可視化ツールの利用により、スケジュール遅延リスクの検出が可能であることが示された。松村らは玉田らのツールを利用し、プロジェクトの問題の早期発見、改善手法を提案している [5]。実際のプロジェクトへの適用結果を通して、提案する手法が有用であることを示された。ただし、これらの手法の導入にはプロジェクト全体へのデータの自動収集ツールの導入コストが発生する。

Mizuno ら [6] は問題化したプロジェクトに発生する、プロジェクトの混乱の予測（早期検知）を目的とした研究を行っている。Mizuno らはプロジェクトの早期段階でリスクに関する調査アンケートを利用し、プロジェクトが最終的に混乱状態に陥るかどうかをベイズ識別器を用いて判定している。同様に浜野ら [7] は Mizuno らの研究で得られたアンケートに対し相関ルールマイニングを適用することで、要求仕様の定義開発計画などに関するプロジェクトのリスク要因を分析している。分析の結果から、要求仕様に関するルールが複数存在することを示している。ただし、分析に必要なアンケートデータは、その取得・分析に時間を要するため、多くのプロジェクトにおいて、開発の初期段階でこのようなアンケートを実施することは難しい。本研究はプロジェクトの開発過程で必然的に作成される QA 票を用いるため、個別にツールの導入やアンケートの実施が必要なこれらの既存研究とは異なる。

本研究で扱ったプロジェクト実施上の課題に影響を与える要因の分析に関する既存研究としては、古山ら [8] の研究が挙げられる。プロジェクト遂行に及ぼす影響要因の特定を目的として、国内のソフトウェア開発企業から収集したデータを対象とした分析を行っている。データに含まれる、各プロジェクトが計画通りにプロジェクトを進めることができたかどうかを判断した「実績の評価」を対象として、実績の評価に影響を与える要因を調査し、開発において仕様の曖昧さや妥当性が納期遅延に影響することを明らかにした。ただし、分析に採用しているメトリクスが要求仕様の明確度合いや主開発言語といったプロジェクトの性質のような荒い粒度のものであり、具体的な個々の QA 票の記述内容にまで踏み込んで分析を行った本研究課題とは

分析対象の粒度が異なる。

要件定義工程における自然言語によるコミュニケーションを対象とした分析に関する既存研究としては松村ら [9] の研究が挙げられる。松村らはソフトウェア開発における混乱の原因の 1 つである要件定義工程の定義漏れの回避を目的として、ソフトウェア開発の上流工程である要件定義工程のレビュー議事録に着目し、分析を行っている。具体的には要件定義工程におけるレビュー議事録中の発注側・受注側の発言をその意図に応じていくつか分類した上で、発言分類と下流工程の不具合票の関連を調査した分析を行い、ソフトウェア開発において上流工程でのコミュニケーションが下流工程に大きく影響を及ぼすことを示している。さらに松村らの分類をもとに、レビュー議事録中の発言を自動分類する研究 [10] も行われており、ある程度の精度で発言意図の自動分類が可能な事も示している。自然言語を対象とした分析という点は本研究と共通するが、当該研究では発言意図の分類が手動で行われており、トピック抽出を自動化し、作業コストの低減を図った本研究とは異なる。

6. まとめ

本研究では 2 プロジェクトのコミュニケーション履歴を対象に、日時情報を考慮した可視化やプロジェクトの問題化に繋がる要因の分析を行った。A 案件の分析結果から、作業期間や作業時間帯によるプロジェクト問題化の傾向は観測されなかった。B 案件のデータを対象に行った決定木分析の結果からは、人的および工程要因がプロジェクトの問題化に影響を与えることが示唆された。

今後の課題として、問題化の兆候を早期に検知し、PM に対して提示できる仕組みを提供することが挙げられる。また今回は決定木分析を用いて、問題化につながる要因を調査した。今後はランダムフォレストやサポートベクタマシン等のクラス分類手法を用いることで、より詳細に分析することが挙げられる。ただし、決定木分析の結果からは、問題化につながる要因が階層的に得られるという特徴があるため、複数の手法の結果を統合して、プロジェクトが問題化する要因を特定する方法も有用と考えられる。

謝辞 貴重な QA 票データを提供いただき、分析結果に関する議論に参加いただいた A 案件および B 案件の関係者に謝意を示す。本研究の一部は科研費（課題番号: 17H00731）の助成を得た。

参考文献

- [1] McCallum, A. K.: Mallet: A machine learning for language toolkit (2002).
- [2] Cortes, C. and Vapnik, V.: Support-vector networks, *Machine learning*, Vol. 20, No. 3, pp. 273–297 (1995).
- [3] 笹野遼平, 黒橋禎夫: 大域的情報を用いた日本語固有表現認識, 情報処理学会論文誌, Vol. 49, No. 11, pp. 3765–3776

- (2008).
- [4] 玉田春昭, 松村知子, 森崎修司, 松本健一: プロジェクト遅延リスク検出を目的とするソフトウェア開発プロセス可視化ツール: ProStar, Information Science Technical Report NAIST-IS-TR2007002, ISSN 0919-9527, Graduate School of Information Science Nara Institute of Science and Technology (2007).
 - [5] 松村知子, 森崎修司, 勝又敏次, 玉田春昭, 吉田則裕, 楠本真二, 松本健一: 問題の早期発見・改善を支援するインプロセスプロジェクト管理手法の実プロジェクトへの適用, 電子情報通信学会論文誌, Vol. 92-D, No. 11, pp. 1974–1986 (2009).
 - [6] Mizuno, O., Hamasaki, T., Takagi, Y. and Kikuno, T.: An Empirical Evaluation of Predicting Runaway Software Projects Using Bayesian Classification, *Proceedings of 5th International Conference on Product Focused Software Process Improvement (PROFES2004)*, Vol. LNCS 3009, pp. 263–273 (2004).
 - [7] 浜野康裕, 天寄聡介, 水野 修, 菊池 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).
 - [8] 古山恒夫, 菊地奈穂美, 安田 守, 鶴保征城: ソフトウェア開発プロジェクトの遂行に影響を与える要因の分析, 情報処理学会論文誌, Vol. 48, No. 8, pp. 2608–2619 (2007).
 - [9] 松村知子, 吉田 誠, 井手直子, 森崎修司, 戸田航史, 松本健一: ソフトウェア開発の要件定義工程におけるユーザ・ベンダ間のコミュニケーション分析と活用方法, プロジェクトマネジメント学会 2011 年度 春季研究発表大会予稿集, pp. 427–432 (2011).
 - [10] 戸田航史, 松本健一, 吉田 誠, 井手直子: ユーザ・ベンダによるソフトウェア要件合意形成プロセスにおけるユーザ発言の自動分類, ソフトウェア工学の基礎 XVIII FOSE2011, pp. 225–230 (2011).