

要求獲得におけるステークホルダ 識別手法の実適用評価

位野木 万里^{†1}

^{†1} 東芝ソリューション（株）

情報システムの要求獲得プロセスにおいて、ステークホルダ識別は、要求の源泉を特定する上で重要なタスクである。ステークホルダの識別が不十分であると、突発的な要求変更、開発工程での作業の手戻り、使いにくいシステムの開発などの問題が発生するリスクが高まる。こうした問題を解決するために、本稿ではステークホルダ図やステークホルダマトリクスを活用したステークホルダ識別手法を考案し、その有効性について考察した。

1. はじめに

情報システムの開発において、要求定義は、システム化の対象となる領域が抱える問題を特定し、その問題の解決策を考案し、情報システムを用いて解決策を実現するための方法を定義する重要なプロセスである。費用や納期、取り巻く環境などの条件を加味した上で、問題の特定からシステム開発に至る要求を、合理的かつ最適に導き出すために、要求工学のさまざまな標準や知識体系が提供されている [1],[2],[3],[4],[5],[6]。

筆者の所属する組織においても、そのような標準や知識体系に基づき要求定義を進めることを実践している。たとえば、筆者らは要求定義のプロセスを詳細化し、入力・処理・出力のプロセスからなる作業標準、成果物を記述する仕様記述の雛形、出力された成果物を検証するチェックリストを共有するなどの取り組みをしている。しかし、実際の要求定義は、システム化の対象となる領域、組織が直面する課題、市場・技術・環境などの条件に応じた工夫が必要となる。そのため、初級のアナリストや技術者が要求定義を効率的に進めることは困難である。

筆者らは、要求定義にはベテランの技術者やアナリストらのノウハウが必要であるとして、彼らの持つ暗黙知を形式化し共有する手法を考案し実践してきた [7],[8]。この取り組みの一環で、要求定義の初期の段階で獲得するステークホルダ要求 [6] に着目したところ、ステークホルダの識別にもさまざまなノウハウが必要であることが分かった。ここで、ステークホルダとは、要求、あるいは、要求が定めるビジネスや情報システムに関与する個人、あるいは組織を指す [5]。ステークホル

ダ識別は、要求定義の出発点であり、要求を獲得する源泉であるステークホルダの識別に失敗すれば、獲得できた要求の品質も安定しないことになる。筆者らは、ステークホルダ識別が不十分であったことが要因で、開発での手戻りが発生する問題に直面した。その失敗経験に基づき、ステークホルダ図、ステークホルダマトリクスによるステークホルダ識別の手法を具体化し、その効果を確認したので、本稿で述べる。

以下、本稿は次のように構成する。第2章では、本稿で取り上げるステークホルダ識別の位置づけを示し、筆者らの組織が直面した問題の例を説明する。第3章では、実際にステークホルダ識別を実施するにあたり、筆者らが取り組んだ手法を紹介する。第4章では、第2章で述べた問題に対して、取り組み中のステークホルダ識別手法を適用した結果を示し、本手法の効果を考察する。第5章では、このような取り組みで得られた教訓を整理し、第6章で本稿をまとめる。

2. ステークホルダ識別の位置づけと直面した問題

2.1 ステークホルダ識別の位置づけ

REBOK [5] において、情報システム開発の上流工程で実施される要求開発は、要求獲得、要求分析、要求仕様化、要求の検証・妥当性確認・評価のプロセスで構成されるとしている。このうち、要求の源泉となるステークホルダとのやりとりは、要求獲得のプロセスにおいて実施される。要求獲得プロセスおよびステークホルダ識別について、REBOKでは次のように整理されている。

- 要求獲得プロセスは、ステークホルダ識別、現状

システムの理解、現状システムのモデル化、課題の抽出と原因分析、課題解決に向けたゴールの抽出、ゴールを達成する手段の抽出、実現すべき将来システムのモデル化、要求の記述と詳細化によって構成される。要求獲得プロセスの最初に、要求の源泉となるステークホルダを識別する。

- 要求獲得活動において、ステークホルダは要求アナリストがインタビューを行う対象者となる。必要に応じ、インタビューを通して、インタビューしたステークホルダの関係者を新たなステークホルダとして特定する。

獲得する要求に関係するステークホルダを漏れなく識別することが重要であり、ステークホルダを識別するための分析手法が検討されている[5]。しかし、これらの手法は、実際の組織の標準に直ちに取り入れられるほどの具体化には至っていない。

2.2 ステークホルダ識別に関して筆者らの組織が直面した問題

筆者らの組織は、従来ステークホルダ識別を実施していなかった。その結果、発生した問題を以下に示す。

2.2.1 ステークホルダからの要求変更により手戻りが発生した問題

これは、あるシリーズ製品の次期モデルの開発時において発生した問題である。開発者は、次期モデルの開発において、既存モデルと同様のステークホルダから要求を獲得していた。ところが、基本設計に着手した際、想定していなかった関係者から、要求追加が発生した。ステークホルダ間での協議の結果、追加された要求は優先度の高い要求であるとの判断がされた。そこで、開発者は設計業務を中断し、追加要求に対応した。開発プロジェクトとしては、要求を再定義する手戻りが発生した。

本問題の要因は、開発者が対象となる製品の市場等の変化を認識せず、いつもの決まったステークホルダから要求を獲得していたことにあると考えられる。

なお、本問題は第4章にて再度取り上げて説明し、発生した問題の解決策とその適用評価について述べる。

2.2.2 オペレーション担当の要求が漏れていた問題

これは、基幹系メインフレームシステムのメインテナンスコスト削減のため、オープンプラットフォームへのリプレースに対して発生した問題である。開発者側は、Webトップによるアプリケーションに置き換えることを提案し、発注者側の合意を得て、リプレースを実施した。Webトップの画面にアプリケーションを置き換えた

ため、従来とは異なる画面操作となった。発注者側の受け入れテストにおいて、実際にオペレーションをする担当者から、従来通りの業務ができず、業務運用に支障をきたすとの意見があがり、ユーザインタフェースの修正要求が発生し、開発作業をやり直すことになった。

本問題の発生要因として、要求獲得の段階で、ステークホルダとして、実際に業務を行う担当者を含めて、合意形成を行っていなかったことが考えられる。

2.2.3 保守に関連する関連機器メーカーの要求が漏れていた問題

これは、ある機器を用いた情報システムの運用に関する問題である。当該情報システムの保守運用では、その機器向けの消耗品の交換メッセージが画面に表示されると、機器メーカーの保守サービス担当に交換を依頼する運用ルールがあった。しかし、当該情報システムの保守担当者には運用ルールの知識はなく、消耗品の交換メッセージが表示されるつど、開発部門に問い合わせ、対応の指示待ちを繰り返していた。その間、情報システムは停止となり、利用者は情報システムを用いた業務が実行できず、業務効率が悪化する事態が発生していた。

本問題の要因の1つは、機器メーカーの保守サービス担当がステークホルダから漏れていた点にある。その結果、消耗品交換に対する保守運用の要求が仕様化されず、このような問題が発生したと考えられる。

3. ステークホルダ識別の具体化による問題解決

上述した2.2.1～2.2.3の問題発生の要因は、要求獲得時にステークホルダの分析が不十分であった点にあると考えられる。こうした問題の解決には、要求を獲得する源泉となるステークホルダの識別方法を、開発標準中に取り入れ、分析した結果を可視化し、それらを元に関係者で合意をし、合意されたステークホルダを中心に要求定義を実施することが重要である。

筆者らは、要求定義ではベテランの技術者やアナリストらのノウハウが必要であるとして、暗黙知を形式知化し共有する手法を考案し実践してきた[7],[8]。文献[7],[8]では要求定義のプロセス全体や品質検証の方法を形式知化したものの、ステークホルダ識別に特化したノウハウは形式知化できていなかった。そこで、本稿ではステークホルダ識別の実施手法の形式知化に取り組んだ。本稿で考案した手法は、Sharpらが提案したステークホルダ識別の技術である、(1) ベースラインと関連ステークホルダによるステークホルダ識別[9],[10]および、

(2) ステークホルダマトリクスによる利害関係予測 [5] により構成する。本稿で提案する手法と既存手法との違いは、成果物の記述の雛形を定義し、その雛形に基づき (1) および (2) の実施手順を具体化した点である。本手法により、2.2.1 ~ 2.2.3 の問題を含む、ステークホルダの漏れの防止や、想定外のステークホルダからの要求追加による開発計画の変更などのリスクを回避することを目指している。

以下、それぞれの手段の詳細について解説する。

3.1 ベースラインと関連ステークホルダによるステークホルダ識別

ここで提案する手法はSharpらが提案したステークホルダ識別 [9],[10] に基づく。Sharpらはベースラインステークホルダとして次の人や組織を列挙している。

- ・ ユーザ：システムの直接の利用者。ユーザはシステムの利用頻度、利用経験、期待する目標、組織内の地位、組織の内部か外部かによって区別できる。ここには操作員も含まれる。
- ・ 開発者：分析者、設計者、プログラマ、テスト、品質保証、保守員、教育担当者、プロジェクト管理者。ツール開発者など二次的な支援者も含む。
- ・ 意思決定者：システム開発に責任を持つ組織内のステークホルダ。開発者と発注者の両方を含む。

Sharpらはベースラインステークホルダへの情報のやりとりのパターンにより、関連するステークホルダを次の3つのステークホルダに分類している。

- ・ サテライト：ベースラインステークホルダとさまざまな方法により相互作用を行うステークホルダ。
- ・ サプライヤ：ベースラインステークホルダに情報

を提供し、その仕事を支援するステークホルダ。

- ・ クライアント：ベースラインステークホルダが開発した製品を処理/検査するステークホルダ。

Sharpらは実開発で本手法を実施するための成果物の統一化などの方法は示していなかった。そこで、筆者らは、ステークホルダ間の関係を記述する成果物の記述の雛形と本雛形を用いたステークホルダの識別手順を定義した。定義した手順を以下に示す。

- ・ ステークホルダを洗い出しベースラインステークホルダとする。
- ・ 抽出した各ベースラインステークホルダに対して情報のやりとりを考慮し関係者を洗い出す。
- ・ 洗い出した関係者をサプライヤ/クライアント/サテライトステークホルダに分類する。
- ・ 抽出できたステークホルダを共通フォーマットを用いてステークホルダ図に記述する。
- ・ 複数名の開発者によってステークホルダ図を記述し、記述結果を比較する。
- ・ 関係者間で上記の結果の差異を議論し、ステークホルダ図を再構成し、関係者全員で合意を得る。

上記の具体例を図1を用いて説明する。図1は、共通フォーマット上に勤務管理システムを題材にしたステークホルダの関係を示した例である。勤務管理システムは、組織に所属する従業員の勤務時間などを管理するシステムである。勤務管理システムに蓄積管理されるデータに基づいて、給与計算等が行われる。

ベースラインステークホルダには、開発者や仕様決定者のほか、利用者として従業員、事務担当、管理者、総務担当、経理担当を定義している。また、勤務管理システムの情報に基づいて給与計算を行うことから、人事給

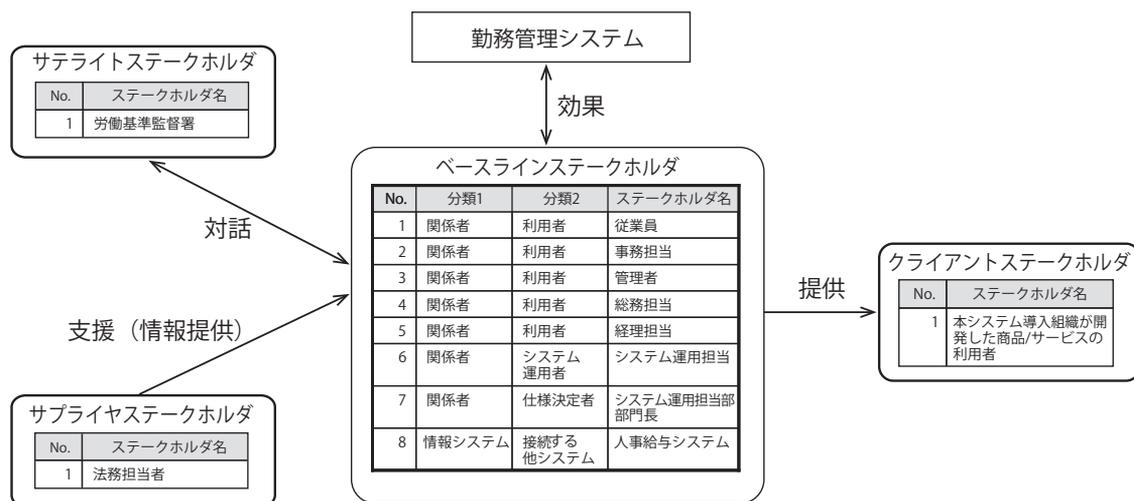


図1 ベースラインおよび関連ステークホルダ間の関係図

与システムをベースラインステークホルダとしている。

ベースラインステークホルダである総務担当は、法律上の問題点や法令の変更情報などを常に確認しておかなければならない。労働基準監督署は、従業員の勤務情報を総務担当から入手し、従業員の勤務実態の適切さを監視する。労働基準監督署からの通達等により、勤務管理の業務ルールの仕様に影響が発生する可能性がある。このような場合に、労働基準監督署をサテライトステークホルダとして抽出できる。

従業員の労働時間の上限などは法律に基づき決定されるため総務担当は法令の変更情報などを得る必要がある。よって、総務担当に情報を提供するサプライヤステークホルダとして法務担当者を抽出する。

従業員の本来業務は、顧客に商品やサービスを提供することである。本勤務管理システムを導入した組織が開発した商品／サービスを提供する顧客を、クライアントステークホルダとして定義する。

以上、ベースラインステークホルダを基点とし、サテライト、サプライヤ、クライアントの観点に拡張してステークホルダを洗い出すことにより、ステークホルダを漏れなく洗い出すことが期待できる。

3.2 ステークホルダマトリクスによる利害関係予測

3.1に従いステークホルダの抽出後、ステークホルダマトリクスを用いてステークホルダ特性を分析し、利害関係予測やリスク抽出を行う。ステークホルダの特性分析では、ステークホルダと要求の利害関係の度合いを、影響度や重要度として評価する。影響度とは、その要求を開発の範囲とするかどうかの意思決定に対する決定力や発言力など、あるステークホルダが及ぼす影響の大き

さである。重要度とは、定義される要求の必要性の度合いである。

筆者らの手法では、ベースライン、サテライト、サプライヤ、クライアントの観点で洗い出したステークホルダに対して、主な要求ごとに、影響度と重要度の観点で、ステークホルダ間の関係を分析する。ステークホルダごとの影響度と重要度の特性に対する位置づけは、後述するステークホルダマトリクスを用いて可視化し、関係者レビュー、合意形成、知識継承に用いる。影響度と重要度は、それぞれ、高い／低いの2値を設定する。ステークホルダの観察やインタビューから、重要度や影響度の考え方に差があり、要求の優先度にばらつきが生じた場合、影響度、優先度の値を、-5～5までの10段階に詳細化し、ステークホルダマトリクス上での配置を決める。10段階の値の考え方を表1に示す。

図2は「事務業務Xをシステム化し自動処理したい」という要求へのステークホルダA、Bの重要度と影響度を表す。図2の配置は次の①②により決める。

- ①各ステークホルダの重要度を高い／低いの2値で分類する。図2の場合はAが高く、Bが低くなった。次に、高い／低い値を表1の5段階に詳細化し関係者間で議論する。図2の場合、AとBの重要度を10段階（-5～5）評価でそれぞれ3と-3とした。
- ②各ステークホルダの影響度を高い／低いの2値で設定する。図2の場合はA、Bのどちらも高くなった。そこで、重要度と同様に「高い」の値を表1の5段階に詳細化し関係者で議論した。Aは本システムの投資実行への権限を持つ1人であり、AはBの意見を考慮する関係にあるので、影響度を10段階（-5～5）評価でそれぞれ4と2と定義した。

表1 「影響度」、「重要度」の値の考え方

値	影響度	重要度
+5	意思決定の最終決定権あり	当該要求は絶対に必要
+4	意思決定者の1人	当該要求は必要
+3	意思決定者への発言力あり	当該要求はほぼ必要
+2	意思決定者が意見を重視する	当該要求はある方が良い
+1	意思決定者が意見を考慮することがある	当該要求はある方が良いがなくても良い
-1	意思決定者が意見を考慮することはほぼない	当該要求の要／不要に関心なし
-2	意思決定者が意見を考慮することはない	当該要求の存在を認識していない
-3	意思決定者はその存在をほぼ考慮しない	当該要求はほぼ不要
-4	意思決定者はその存在を考慮しない	当該要求は不要
-5	意思決定者はその存在をまったく考慮しない	当該要求は絶対に不要

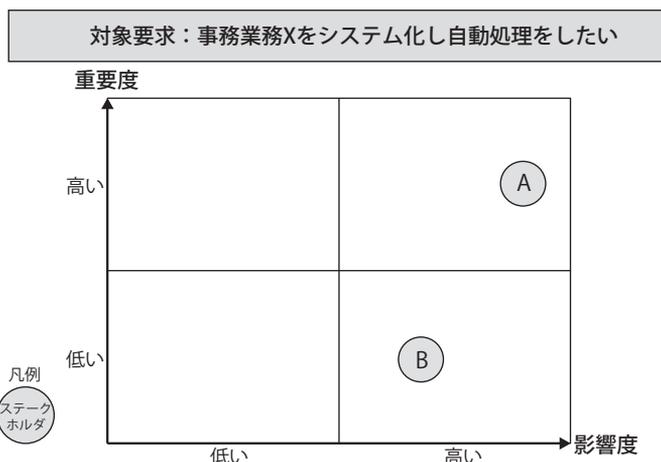


図2 ステークホルダマトリクスの記述雛形

たとえば、文献[6]が示すステークホルダ要求の仕様化方法で示された管理や運用などの主要な要求ごとに図2のようなステークホルダマトリクスを定義することで、ステークホルダごとの重要度と影響度の違いを明確にできる。BがAよりも高い影響力を持った重要度の高い要求（図2とは別の要求）のステークホルダマトリクスがあったとする。全体の投資額が限定され、要求の絞り込みが必要な状況では、各ステークホルダマトリクスを比較することで、AとBが衝突するリスクがあることを事前に把握できる。このようなリスクが事前に把握できれば、ステークホルダ間でのすり合わせの期間をあらかじめ計画に組み込むなど、余裕を持った対応が可能になる。

4. 適用評価・考察

第3章で述べたステークホルダ識別の手法を、2.2.1～2.2.3に示した問題に対して適用し、結果を分析し、本手法の有効性を考察する。

4.1 事例適用

ここでは2.2.1の問題に対して、提案するステークホルダ識別の手法を適用した結果を示す。

4.1.1 ステークホルダ図によるステークホルダの抽出

2.2.1の問題で示したシステム開発に関するステークホルダを洗い出し、ステークホルダ図としてまとめたものを図3に示す。

2.2.1にて述べたように、本件はシリーズ製品の次期モデルの開発案件である。前モデル開発までのステークホルダは、発注者である組織A、その製品の利用者となる組織B、製品の開発を担当する組織CとD、製品開発への投資実行を担当する経営部門Eである。そこで第3章で考案した手法を適用し、ベースラインステークホルダに対して情報のやりとりがある関係者を整理すると、上位事業部門X、研究開発部門Y、クライアント製造メーカーZを洗い出すことができた。

上位事業部門Xは、ここで開発した製品の販売に関心を示し、経営部門Eと連携している。よって本部門Xは、サテライトステークホルダとなる。研究開発部門Yは最新技術についての情報を経営部門や開発者にもたらす。よってこれをサプライヤステークホルダと位置づけた。開発者が開発した製品を提供する予定のクライアントは製造メーカーZである。よって、Zをクライアントステークホルダとして定義した。

4.1.2 ステークホルダマトリクスによる重要度と影響度の把握

対象となるシリーズ製品開発の次期モデルは、「ある条件下の処理時間は10分とする」という時間効率性要求と「誤動作を誘発しにくいGUI」という操作性要求に対して要求変更が発生した。ベースラインステークホルダは時間効率性要求を優先し、操作性要求を開発範囲外と決定した。しかし、開発の途中で要求変更が発生し、操作性要求へ急遽対応することになった。

時間効率性要求と操作性要求に対するステークホルダ

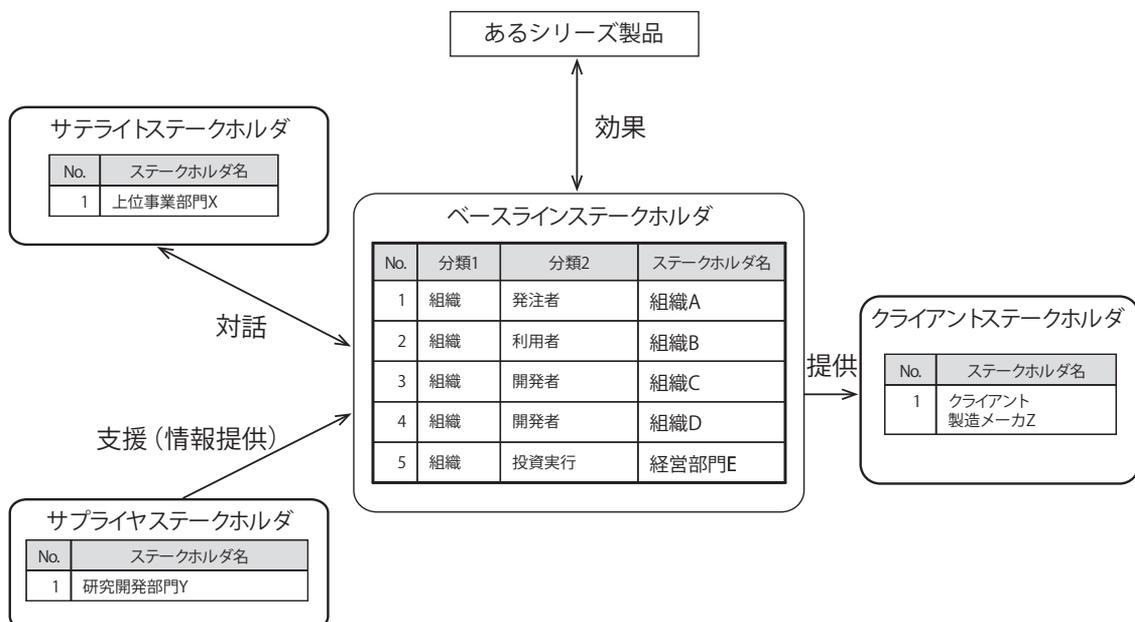


図3 あるシリーズ製品開発におけるステークホルダ図

マトリクスを、それぞれ図4および図5に示す。図中の記号A～Zは、図3のステークホルダ名の末尾の記号と一致させた。ステークホルダマトリクスは10段階評価（-5～5）で詳細化した。

●時間効率性要求のステークホルダマトリクス（図4）

発注者組織A，利用者組織B，開発者組織C，Dはこれまでの経験が共有され，時間効率性要求は最重要事項と認識されていたため，重要度を5とした。Aには開発範囲の意思決定権があり影響度は最も高く，影響度を5とした。Bは開発範囲の意思決定権はないもののAが利用者であるBの意見を考慮する関係にあるため，Bの影響度を2とした。DはCの下部組織であり，それぞれ発注者の依頼で開発するため，影響度は低く，C，Dの影響度は，それぞれ-2，-3とした。

投資実行を決定する経営部門Eは，本要求を開発範囲とするかどうかの意思決定をAに委譲していたため影響度は-5とした。また，上位事業部門X，研究開発部門Y，クライアント製造メーカZは，実際の開発範囲への決定権はないため影響度を-5とした。E，Xは時間効率性は追求してもよいと考えている程度で，さらにXの関心がより低いため，各重要度は2，1とした。Zの本要求への関心は低く，Yは要求の存在すら認識していないと想定したものの，絶対的不要である-5とできる根拠はないため，それぞれ，-1，-2程度とした。

●操作性要求のステークホルダマトリクス（図5）

Zが本製品を実際に利用する可能性が高くなり，操作性要求を自らの問題として捉えたことから，重要度は5となった。また，最終顧客であるZの要望に応えるため事業方針が変更され，XおよびEの本要求の重要度が5となった。そして，EはAへの権限委譲を解消し，Xは上位事業部門としてEへの影響力を高めた。よって，XとEの本要求への影響度は，それぞれ，5，4とした。なお，Yの影響度は従来通り-5で，操作性要求への関心は低く重要度は-3である。

ところで，A～Dの本要求への重要度は低い点で共通しており，-3とした。また，A～Dの影響度は，X，Eと比較し，それぞれ，3，1，-2，-3とした。

図5では，影響度が高いと設定しているス

テークホルダ間で，要求の重要度が衝突していることが明らかになる。また，さらに図4と図5を比較すると，影響度が高いAとBは，図4，図5の2つの要求に対して，異なる優先度を設定している。こうしたことから，投資が限定され，要求を絞り込む際に，効率性と操作性の一方を開発することになれば，その選択に至る過程で衝突が発生するリスクがあることが事前に分かる。

4.2 ステークホルダからの要求変更により手戻りが発生した問題への適用効果

4.2.1 ベースラインと関連ステークホルダによるステークホルダ識別の効果

図3のステークホルダ図を用いることで，ベースラインステークホルダと暗黙的に連携していた上位事業部門X，研究開発部門Y，今後の製品提供先となる製造メーカZなど，要求獲得の源泉となる可能性のある組織を洗

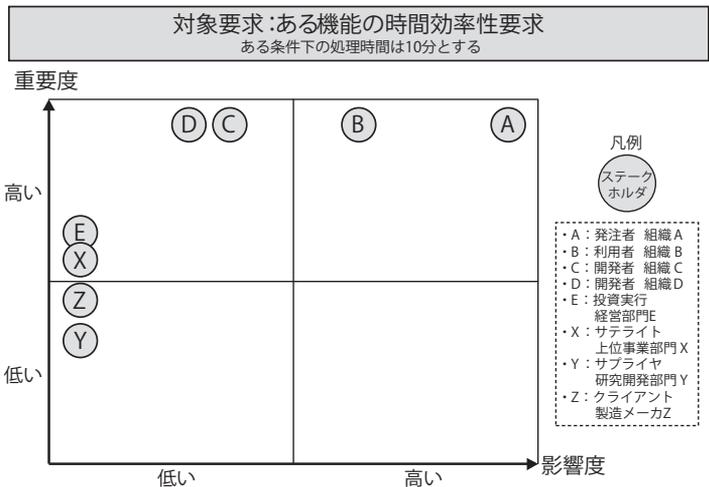


図4 時間効率性要求に対するステークホルダマトリクス

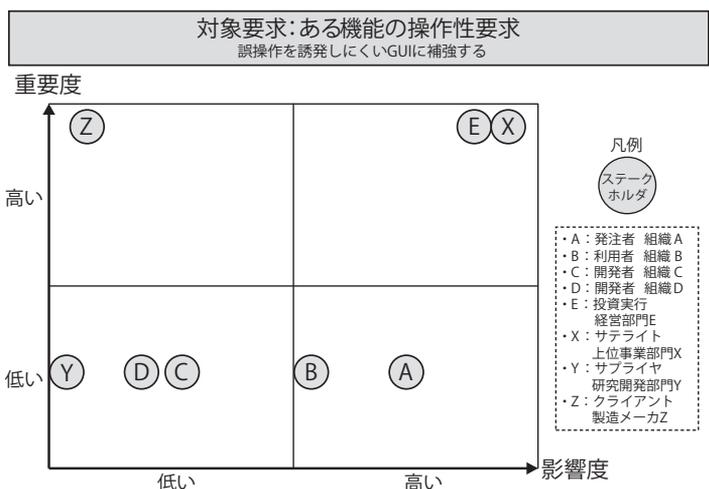


図5 操作性要求に対するステークホルダマトリクス

い出し、関係者間で認識することができた。通常の開発では、ベースラインステークホルダを優先して要求を獲得する傾向があるが、関連ステークホルダの影響を受けて要求への重要度や影響度が変化する可能性を意識することで、要求変更へのリスクを事前に洗い出すことができる。

4.2.2 ステークホルダマトリクスによる利害関係予測の効果

問題があった2.2.1のケースでは、「誤動作を誘発しにくいGUI」要求に対しては、重要度は低いとして開発範囲には取り入れず、「ある条件下の処理時間は10分とする」といった時間効率性要求についての対応を優先した。図4、図5のステークホルダマトリクスを用いて、各要求に対するステークホルダ間の重要度や影響度の関係を可視化することで、重要度が逆転している要求の存在が明らかになった。加えて、重要度が逆転するケースで、要求の優先度について合意を得ずに進めた場合、影響度の高い組織から手戻り要求が発生するリスクがあることも認識できた。ステークホルダマトリクスは、要求とステークホルダの違いによって、重要度と影響度が異なる状況を可視化し、関係者間で状況を共有しやすくし、問題発見と対策立案に早期に着手することに役立つ。

性能や移植性の改善要求は、開発者からの要求であれば、重要度が高くなり、操作性や機能性などの要求は、関連ステークホルダからの重要度が高くなる傾向にある。また、新たな法令や法改正などの発生によって、重要度が変化するパターンも考えられる。図4、図5のようなステークホルダマトリクスにより統一した記述フォーマットで状況を蓄積しておけば、重要度と影響度の関係のパターン化と共有がしやすくと考えられる。このようなパターンは要求獲得におけるノウハウを形式知化したものであり、パターンを再利用して、要求獲得計画を立案することで、突発的な要求の発生による開発作業の手戻りや開発作業の混乱を防止することが期待できる。

4.2.3 生産性効果

表2に2.2.1で対象とした次期モデル開発の計画時の開発工程表を示す。開発期間は6カ月である。開発人員は通常は3名で、テスト工程では8名を予定していた。合計のコスト見積りは11人月である。事前調査では、従前より利用者からの要望としてあがっていた

処理時間性能の短縮のため、処理のメカニズムの調査や新アルゴリズムの実現可能性の分析を計画し、当初の要求定義に2人月の工数を予定していた。

実際には、従来のステークホルダではない別のステークホルダからの要求追加が基本設計時に発生し、その結果、基本設計を止め、要求定義を改めた。実際の実績工程表を表3に示す。要求定義には4.5人月を必要とした。4.5人月の内訳は、当初予定の要求定義：2人月、追加要求に起因した要求のスコープ再定義：0.7人月、追加部分の要求定義：0.55人月、当初予定の要求定義結果の見直しとドキュメント修正：1.25人月である。なお、基本設計からシステムテストの工数が増加したのは、開発の全体規模が増加したためである。

あらかじめステークホルダ識別を行い、ベースラインステークホルダと関連ステークホルダとの関係も踏まえた要求獲得を行うと、従来よりも要求定義のコストは増加する。ステークホルダが増加すると事前準備や調整など次のコストが必要になると考えられる。

- ・ ヒアリング/レビュー：3H * 3人 * 3回 = 27人H.
- ・ 要求仕様化：ヒアリングごとに準備とまとめで合計8人H. 3回分として、8人H * 3回 = 24人H.
- ・ 開発者内部レビュー：4人H * 3人 * 3回 = 36人H.

上記の合計による増加コストは87人Hで、1人月を150人Hとすると約0.58人月となる。上記見積りを考慮し、本手法による開発工程表を表4に示す。表4では

表2 予定工程表および工数

No.	タスク	工数 (人月)	1カ月目	2カ月目	3カ月目	4カ月目	5カ月目	6カ月目
1	事前調査	1	→					
2	要求定義	2		→				
3	基本設計	3			→			
4	詳細設計/開発/単体テスト	3				→		
5	総合テスト/システムテスト	2						→
合計		11	1	2	2	2	2	2

表3 実績工程表および工数

No.	タスク	工数 (人月)	1カ月目	2カ月目	3カ月目	4カ月目	5カ月目	6カ月目
1	事前調査	1	→					
2	要求定義	4.5		→	→			
3	基本設計	3.25			→	→		
4	詳細設計/開発/単体テスト	3.5					→	
5	総合テスト/システムテスト	2						→
合計		15.75	1	2	2.75	3	3.5	3.5

表4 理想プロセスおよび工数

No.	タスク	工数 (人月)	1カ月目	2カ月目	3カ月目	4カ月目	5カ月目	6カ月目
1	事前調査	1	→					
2	要求定義	3.38		→				
3	基本設計	3.25			→			
4	詳細設計／開発／ 単体テスト	3.5				→		
5	総合テスト／ システムテスト	3.5						→
	合計	14.63	1	2.6	2.53	2.5	3	3

開発期間は6カ月、詳細設計～システムテスト（表4のNo.4と5）の開発期間は表2と同一とした。

ステークホルダを識別し仕様化する工数は0.25人月である。追加要求に関する要求定義は0.55人月である。したがって $0.58 + 0.25 + 0.55 = 1.38$ 人月が計画に対する要求定義の追加コストとなる。よって、理想とする要求定義のコストは $2 + 1.38 = 3.38$ 人月となる。なお、ここには要求獲得結果の説明コストは織り込んでいるため、ベースラインステークホルダである経営者側とのレビューコストは追加コストにはしない。

筆者らの提案する手法の導入により、要求定義の実績コストが4.5人月から3.38人月へと減少するので、 $4.5 - 3.38 = 1.12$ 人月のコストを削減できる。さらに、工数だけではなく、基本設計などの段階でリードタイムの余裕が生まれるので、新たな人員を獲得することが必要なくなり、開発リスクの回避につながる。

4.3 実際の実運用担当者への要求が漏れていた問題への適用

2.2.2にて述べた問題の要因は、ステークホルダとして実際の実運用担当者が漏れていた点にある。ステークホルダ図として実際の実運用担当者を洗い出すとともに、発注者と実運用担当者に対して、図4、図5のようなステークホルダマトリクスを記述すると、「従来と同様の操作性」といった操作性要求に対する重要度にギャップがあることが認識できる。そして、その要求を開発範囲に入れないことによりどのような影響があるか、事前検討の余裕が生まれ、今回の問題発生を防止することが期待できる。

4.4 保守に関連する関連機器メーカーの要求が漏れていた問題への適用

2.2.3の問題の要因は、ステークホルダ識別において、システムの直接の利用者、管理者だけではなく、メイン

テナンスにかかわる組織や部門を考慮に入れておかなかった点にある。ステークホルダ図により、保守担当者ややりとりをする関連機器メーカーをサテライトステークホルダとして洗い出せた。

ステークホルダマトリクスを用いれば、「業務停止を防止するための保守運用の手順のマニュアル化」という要求の重要度が高いことが容易に把握できる。ステークホルダ図とステークホルダマトリクスを相互に参照・確認

することで、重要な要求に対して、要求を詳細化する際に考慮すべきステークホルダの定義漏れを防止できる。

5. 得られたプラクティス

以上の適用評価を通して、要求獲得において組織が考慮すべきプラクティスを整理する。

- ベースラインステークホルダと関連ステークホルダを洗い出し、これらのステークホルダから要求獲得を行うべきである。これまでの習慣で固定化されたステークホルダからのみ要求獲得をすると、市場、環境、技術などの変化に柔軟に対応できないリスクがある。
- 要求の重要度がステークホルダ間の衝突なく決まると要求獲得が成功したように思えるものの、図4と図5から読み取れたように、要求間のバランスが崩れるリスクが内在していることを意識すべきである。操作性要求と性能要求は、ステークホルダの違いにより重要度が異なる。あらかじめ、誰と誰の要求が対立するかを予測し、すり合わせ作業を事前に計画することが重要である。
- ステークホルダは一見関係ないと思われる範囲まで幅広く集め認識しておく。ステークホルダ図とステークホルダマトリクスを相互に確認し、開発や経営に携わる組織、納入するハードウェア、ソフトウェアの販売／製造／保守業者、保守・運用、国・地域の法令や業務ルールに影響する組織まで、広くステークホルダを認識し、要求に反映させる必要がある。

6. おわりに

本稿では、ステークホルダ図、ステークホルダマトリクスによるステークホルダ識別の実践方法を考案し、そ

の適用効果を示した。

ステークホルダ識別は、要求獲得の出発点であり、要求を獲得する源泉であるステークホルダの識別に失敗すれば、その後、獲得できた要求の品質も安定しない。本稿で述べた手法のように、ベースラインステークホルダに加えて関連ステークホルダを考慮することや、ステークホルダの各要求に対する影響度と重要度をステークホルダマトリクスを用いて分析することは、開発途中での突発的な要求変更を防止し、余裕を持った開発に取り組むことを可能にする。

要求獲得に関するすべての問題が、今回提案したステークホルダ識別の手法のみで、解決できるわけではない。しかし、統一した手法、考え方、成果物等の記述雛形を用いることで、ノウハウが統一した形式で蓄積できる。そして、蓄積されたノウハウは組織の資産となり、資産の再利用により、開発生産性の向上、品質の安定化、リードタイム短縮に有効であると期待できる。今後も、取得できたノウハウを組織の資産として蓄積し、共有・再利用することに取り組んでいく。

参考文献

- 1) IEEE: IEEE std. 830-1998, IEEE Recommended Practice for Software Requirements Specifications (1998).
- 2) International Institute of Business Analysis: A Guide to the Business Analysis Body of Knowledge® (BABOK® Guide) Version 2.0 (2009).
- 3) (独) 情報処理推進機構, ソフトウェア・エンジニアリング・センター: 機能要件の合意形成ガイド, <http://sec.ipa.go.jp/repor>

ts/20100331.html (2013年3月15日現在)

- 4) (独) 情報処理推進機構, ソフトウェア・エンジニアリング・センター: 非機能要求グレード, <http://sec.ipa.go.jp/reports/20100416.html> (2013年3月15日現在)
- 5) 情報サービス産業協会, REBOK 企画 WG: 要求工学知識体系, 近代科学社 (2011).
- 6) ISO/IEC/IEEE: ISO/IEC/IEEE 29148 Systems and Software Engineering — Life Cycle Processes — Requirements Engineering (2011).
- 7) 北川貴之, 橋本憲幸, 吉田和樹, 位野木万里: 要求定義における暗黙知の形式知化手法, コンピュータソフトウェア, Vol.27, No.3, pp.93-98 (Aug. 2010).
- 8) 位野木万里: 高品質な要件定義のための暗黙知の形式知化と共有手法, 東芝レビュー, Vol.66, No.9, pp.64-65 (2010).
- 9) Sharp, H., Finkelstein, A. and Galal, G.: Stakeholder Identification in the Requirements Engineering Process, Proceedings of 10th International Workshop on Database & Expert Systems Applications (DEXA), pp.387-391 (1999).
- 10) 妻木俊彦, 白銀純子: 要求工学概論, 近代科学社 (2009).

位野木 万里 (正会員) inoki.mari@toshiba-sol.co.jp
 東芝ソリューション (株) IT 技術研究所 研究開発部 研究主幹。
 1991 年早稲田大学大学院理工学研究科数学専攻修士課程修了,
 同年 (株) 東芝入社。2012 年 7 月より現職。2008 年早稲田大学大学院理工学研究科情報・ネットワーク専攻博士課程修了,
 博士 (工学)。ソフトウェア資産および経験的に得られた暗黙知を形式知化し, それらを再利用する手法の考案, 実開発での適用に従事。本会理事 (2012 年~), 日本工学会理事 (2012 年~), 国立情報学研究所特任教授 (2012 年~)。

投稿受付: 2012 年 8 月 9 日

採録決定: 2013 年 3 月 13 日

編集担当: 中野美由紀 (東京大学)