

ユーザとベンダ間の協調による 要求品質確保のための定量化事例

松村 知子ⁱ, 松本 健一ⁱ

概要

要件定義・管理はソフトウェア開発の成否を左右する重要な作業であり、システム要求を行う「ユーザ」と開発を担当する「ベンダ」の協力が欠かせない。しかし、CMMI や PMBOK など多くのプロジェクト管理手法やプロセス改善手法では、ベンダ作業に重点が置かれ、ユーザについてはベンダ作業の一部の管理対象としての取り上げ方が多い。本レポートでは、要件定義や要件管理などはユーザ・ベンダの協調作業によって進められるべきであると考え、ユーザ・ベンダ各々の作業の責任分担を調査し、各々の作業が十分に行われているかを定量的に判断する要求品質管理手法を提案する。提案する管理手法を用いることによって、ユーザ・ベンダのシステム開発に対する責任分担が明確になり、ユーザ・ベンダが相互に開発作業を監視することができる。その結果、問題を早期に発見し、ユーザ・ベンダのいずれかもしくは双方で適切な処置を行うことで、開発の破たんや遅延、コスト超過を未然に防ぐことができる。本レポートでは、要求品質管理の具体的な定量化モデルを提案し、いくつかについて金融取引システムの実開発プロジェクトのデータを用いて実際の計測を行い、実現可能性の評価を行う。

ⁱ 奈良先端技術大学院大学 情報科学研究科

1. はじめに

1.1. 背景

近年、ソフトウェア開発プロジェクトにおいて、ソフトウェア発注者（ユーザ¹）の参画が重要視されてきている。従来、ソフトウェア発注者（ユーザ）は要件定義などからソフトウェア受注者（ベンダ）まかせであるケースが多かった。しかし、ユーザ不在の開発では、要件のあいまいさから設計ミスや頻繁な仕様変更が発生し、品質低下、重大な不具合によるシステムダウン、コスト増大、納期遅れなどの原因となっている[5]。また、多重請負・オフショア開発などの複雑な開発体制や多重の受発注関係では、潜在的発注者（ユーザ）が多く存在し、統制されたプロジェクトの管理を困難にし、管理コストを増大させている[12]。

ソフトウェア開発において、ユーザ要件の抽出、定義、変更はプロジェクトの成否に大きく影響することはベンダ企業へのアンケート[4]などで指摘されていて、産業界と経済産業省では企画から要件定義、システム設計など超上流・上流工程に対して、集中的な改善の取り組みを行っている。経済産業省では、モデル契約書[7]で、「要件定義工程を含めた超上流工程はユーザが責任を負うべきフェーズであることを明記し」、上流工程の準委任契約などを推奨している。これは、ベンダの立場から、システム企画や要件定義の負担やリスクを低減し、公正な取引（契約）によってベンダの利益を保護する目的がある。共通フレーム 2007[3]では、ISO12207 規格に対して、企画プロセス、要件定義プロセス、契約の変更管理プロセスなどを追加することで、ユーザの役割や責任を明確にするための枠組みや手順を示した。

これらは、ガイドラインとしてユーザ及びベンダにとって有用であるが、双方が担う役割や責任の遂行を管理する方法までは示していない。そのため、どこまで要件定義工程で行うべきか、要件変更はどの程度許容できるかなど具体的な運用は、各プロジェクトの現場で当事者による状況に応じた判断が行われている。またこれらの作業は、定量的に管理されていないため、過去のデータなどから抽出される基準となる値がなく、客観的あるいは統一的な判断が困難である[13]。

従来、ソフトウェア開発における定量的データによる管理は、開発ベンダ組織内部でのプロセス改善やプロジェクト内部の進捗・品質管理を目的とし、組織間共有やユーザの視点に立って行われてこなかった。そのため、ユーザには定量的データによる管理に対する知識や技術、データの蓄積がなく、開発管理に参画することが困難な状況であった。しかし、ソフトウェアの問題の社会的な影響度の拡大や、マルチベンダ開発やオフショア開発など開発組織の複雑化に対して、定量的管理のより幅広い活用が求められてきている[12]。ベンダからの情報提供を求めるユーザ（発注者）も増加してきている。ユーザ側から見た

¹ 本レポートでは、ソフトウェア開発関係者として、主に「ユーザ」「ベンダ」という2つの言葉を用いる。特記しない限り、「ユーザ」はソフトウェア発注者、ソフトウェア利用者（エンドユーザ）などを含み、「ベンダ」はソフトウェア受注者、ソフトウェア開発者などソフトウェアを供給する人・組織を示す。

見積もりの変更の要因が、ベンダの分析不足や契約後の予想外の要件変更発生が 50%を超える（付録 図 1 参照）ことから、要件定義工程でのベンダの役割と同様に要件定義後の工程でのユーザの役割が重要であるとわかる。そのため、請負契約の開発では従来ベンダ任せになっていた設計、テスト情報、品質、計画・管理情報などの開示を要望するユーザが多い（付録 図 2 参照）と考えられる。

1.2. 本研究の目的と期待される効果

StagE プロジェクト²では、ユーザ・ベンダの双方の立場からソフトウェア開発プロジェクトの成功を支援するため、「ソフトウェアタグ」技術の研究・開発を行っている。ソフトウェアタグは、プロジェクトやプロセス、プロダクトの品質を定量的に判定するための統一規格となるべきデータ項目を定義する[1]。本研究では、ソフトウェアタグ研究の一環として、要求品質を確保するために、ユーザ・ベンダ各々の要件に関する作業を定義し、それぞれの作業が十分に行われているか判断するための定量的なデータに基づく管理手法の開発を目的としている。

本レポートでは、いくつかの具体的な要求品質各のための定量的管理モデルを提案し、実プロジェクトのデータを用いて実計測した結果について報告する。提案する定量的な管理方法は、ユーザ・ベンダ間の開発作業の責任分担を明確にして、双方の合意によって重要な管理対象の作業を選択し、両者の共有データとコミュニケーションの定量的分析によって、問題の有無を相互確認し是正処置を行うことを支援する。なお、以降用いる「ユーザ」「ベンダ」は、特記しない限り、受発注関係にある「発注者」「受注者」と同等する。

提案する定量的な管理方法は、ソフトウェア開発のユーザ・ベンダの各々の観点からデータ計測や分析を行う点が、これまでのプロジェクト管理と異なる点である。従来のソフトウェア工学におけるプロジェクト管理[3][8][9][11]・要求工学[2][5]などと異なる点を、以下にあげる。

1. ユーザ・ベンダの責任分担(作業)を明確に定義し、計測するデータは基本的にユーザ・ベンダ間で共有・授受されているデータを用いる。
2. 従来の「要件定義＝ユーザ責任」「要件変更管理＝ベンダ責任」と言った暗黙の工程別の責任分担を排除し、作業単位でのユーザ・ベンダ双方の責任分担とその作業実績を明確にする³。
3. ユーザ・ベンダ各々の作業、およびユーザ・ベンダ協働作業について、個別に計測を行う。個別の定量化により、ユーザ・ベンダ双方が独自に各作業の統計的管理が可能になる。

² Software Traceability and Accountability of Global Software Engineering : 文部科学省のプロジェクト次世代 IT 基盤構築のための研究開発「エンピリカルデータに基づくソフトウェアタグ技術の開発と普及」の略称

³ 例えば、検収（ユーザによる要件確認）工程では、ユーザは十分なテストを行い、要件を確認する責任があるが、ベンダは変更要求やバグに対する調査・デバッグ・修正といった作業に責任があるとし、それぞれを別の作業と位置づける。

1.3. 本研究のアプローチ

以下に、本研究のアプローチを示す。

- ① 要件に関する作業をユーザ・ベンダ各々に抽出する
- ② プロジェクトの計画・実施過程で、ユーザ・ベンダ間で授受される共有データ及びコミュニケーションを抽出する
- ③ 3つの要素を組み合わせ、要求品質管理の単位となる「管理フェーズ」を構成する
- ④ 共有データやコミュニケーションの定量的な分析を行い、相互確認や、必要に応じて是正処置を行う

本レポートでは、①～③について、[3][5]などの文献を参照するとともに、ご協力いただいたユーザ・ベンダ企業の開発関係者と共に実プロジェクトでの状況を分析して、実施した結果について述べる。④については、具体的に3章に示すが、実プロジェクトデータの分析を行った上で有効と判断されたいくつかのパターンを抽出し、モデル化した。

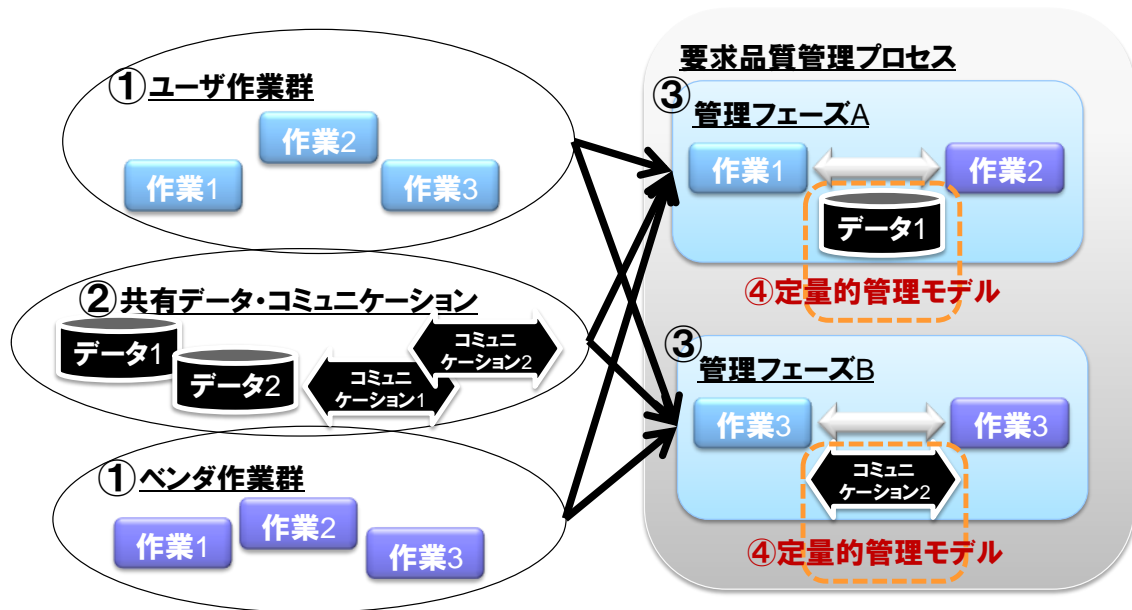


図 1-1 本研究のアプローチ概要

2. 要求品質の定量的管理

本章では、提案する要求品質管理方法の具体的な構築過程と各ステップでの成果について紹介する。

2.1. ユーザ・ベンダ別作業の定義

表 2-1 に、ユーザ・ベンダの要件に関する作業例⁴を示す。これらは、ユーザ・ベンダ(発注者・受注者)間で発生した問題の事例[6][10]や超上流工程のユーザ・ベンダの作業ガイドライン[5]を参考にして抽出した。「作業分担者」は、作業成果に対して責任を持つステークホルダ（ここでは、ユーザもしくはベンダ）を示す。例えば、“要件確認”の“検収試験計画”はユーザ・ベンダ双方で協力して作成するとしても、最終的な試験計画について責任はユーザにあるものと考え、ユーザの作業に入れている。「作業概要」は、それぞれの作業を作業の主体者の実施目的に応じて分類したものである。たとえば、「ユーザ」の“要件抽出”は、ユーザ（発注者のシステム部門など）が実際のシステム利用者（運用者、関連企業など）から要望をヒアリングし、優先度やリスクの分析を行い、最終的な要件を固めていく作業である。それに対して、システムの要件を記した RFI, RFP, 要件定義書の作成は、ベンダへの見積もりや開発のための要件伝達が目的なので、「要件伝達」に含める。

図 2-1 は、「作業概要」の実施工程の参考とするため、共通フレーム 2007[3]の主ライフサイクルプロセスにマッピングした例である。あくまで一例であって、すべてのプロジェクトですべての作業が行われるわけではない。

表 2-1 ユーザ・ベンダ作業具体例

作業分担者	作業概要	作業内容（具体例）
ユーザ	要件抽出	企業戦略
		ビジネス分析
		ステークホルダとの合意
		リスク把握
	要件伝達	RFI,RFP 作成
		要件定義書作成・レビュー
		システム設計(基本設計)レビュー
		Q&A
	要件確認	検収試験計画(要件定義時)
		検収試験の実施・確認
	要件変更・改善	要件変更・改善の抽出/調査/依頼
		リスク管理(環境, 政策, 市場動向の監視)

⁴ これがすべての作業、あるいはすべてのプロジェクトでこれらの作業が行われるというものではない。

ベンダ	要件抽出支援	機能要件・非機能要件の確認
		Q&A
	要件理解	システム設計(基本設計)作成・レビュー
		Q&A
	要件実現	要件トレーサビリティの確保
		Q&A
	要件変更対応	要件変更・追加ルール策定・管理
		リスク管理(環境, 政策, 市場動向の監視)
		軽微な変更への準備(変更容易性)

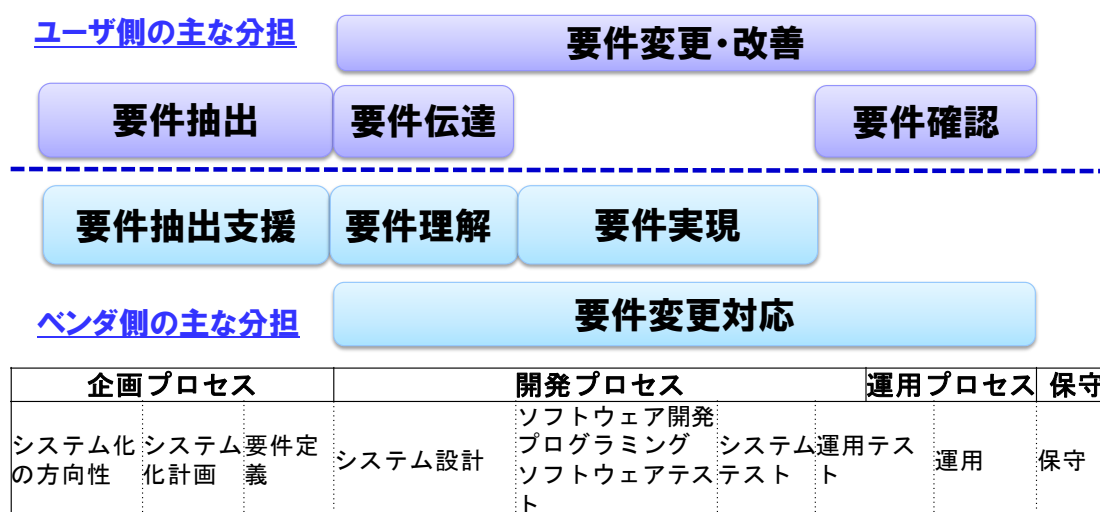


図 2-1 ユーザ・ベンダの要件関連作業と主ライフサイクルプロセス[3]

2.2. 定量的管理フェーズとソフトウェアタグ項目

定量的に管理する作業は、表 2-1 の作業から選択できる。しかし、実際には作業内容の重要度、優先度、管理の効果などからユーザ・ベンダ双方の協議により選択されるべきである。図 2-2 は、定量的に管理する作業（以降、フェーズと呼ぶ）の例を示している。フェーズ選択のポイントは、以下のとおりである。

- (1) ユーザ・ベンダ各々に責任のある作業が存在する。もしくは、要求品質確保上、重要な作業が存在する。
- (2) ユーザ・ベンダ間で何らかのコミュニケーション・データの授受が存在するか、共有可能なデータが収集でき、定量的に計測できる⁵。
- (3) ユーザ・ベンダ相互の確認・合意によって、ユーザ・ベンダの双方もしくは一方が

⁵ Q&A は、明文化された質疑応答記録が残ることを前提としている。例えば、ユーザとベンダが同じ作業場所において、仕様確認が口頭で行われる場合には管理することが困難である。逆に、管理を行うためには、あらかじめ仕様確認を記録に残るようにコミュニケーションプロセスを計画しなければならない。

是正処置をとることが可能である。ただし、プロジェクト進行中や直後に是正処置がとれないような場合でも、プロジェクトの事後検証や次プロジェクトへの改善を目的とした管理も対象とする(3.9節参照)。

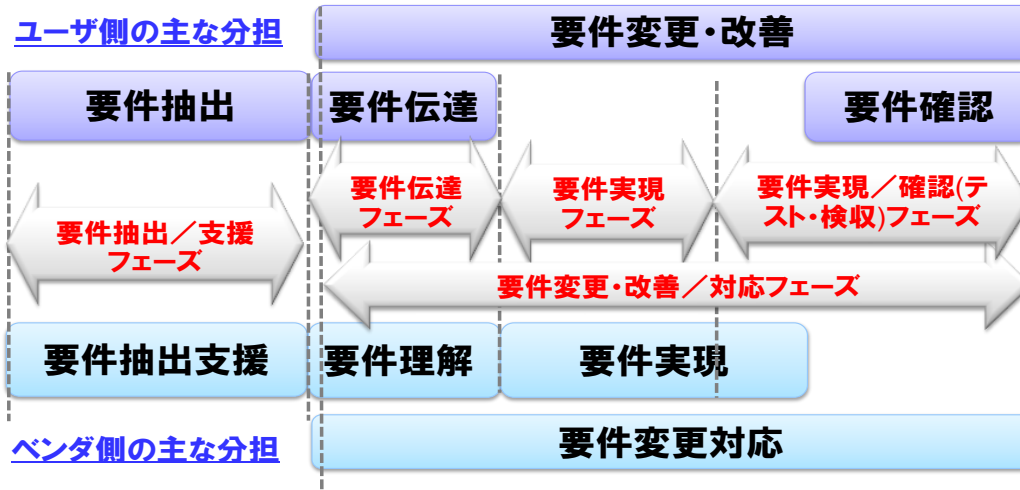


図 2-2 定量的管理を適用するフェーズ例

表 2-2 は、「ソフトウェアタグ規格 Ver1.0」[1]の一般的に定量管理に用いられるデータ項目を、図 2-2 のいくつかのフェーズに対応させた表である。[1]のデータ(タグ項目)には、過去の統計分析や時系列分析など企業での実績や研究実績が多いため、定量的な管理を行う際の有用性が高いと言える。

表 2-2 要件管理の対象フェーズと活用可能なタグ項目

対象フェーズ	タグ項目表 (タグ規格 Ver. 1.0[1]から抜粋)				
	分類	タグ項目	説明	具体化例	実証データ例
要件抽出/支援フェーズ	要件定義	ユーザヒアリング情報	要件に関してユーザに行ったヒアリングに関する情報	ユーザヒアリング実施件数(回), ユーザヒアリング項目数(件), ユーザヒアリング回答率(ユーザヒアリング回答数÷ユーザヒアリング項目数)など	ユーザヒアリング議事録, ユーザヒアリング質問票など
		規模[推移]	開発側で作成した要件数	画面、機能項目、ユースケース、アクター、顧客要件、機能、FP など	要件定義書など
		変更[推移]	変更された要件数	規模の計測単位に依存	要件定義書, 要件定義書の変更履歴など

要件伝達フェーズ	設計	規模[推移]	設計成果物の規模 ※新規・改造・再利用(流用)毎に計測する	機能設計 (ページ数・帳票数・画面数・ファイル数・項目数・UML図の数、クラス数、バッチプログラム数、重要な機能数など) 構造設計 (データ項目数, DFDデータ数, DFDプロセス数, DBテーブル数 など) など	基本設計書, 機能設計書, 構造設計書, 詳細設計書など
		変更[推移]	変更された設計成果物の数, もしくは変更量	規模の計測単位に依存	基本設計書, 機能設計書, 構造設計書, 詳細設計書, 各設計書の変更履歴など
		要件の網羅率	要件定義で作成された要件の実装率	設計に取り入れられた要件数 ÷ 要件数	基本設計書, 機能設計書, 構造設計書, 詳細設計書など
	品質	レビュー状況	成果物(仕様書, 設計書, プログラムコード, テスト仕様書など)のレビューに関する情報	レビュー回数, 対象規模, 実績回数, ユーザ参画の有無, レビュー時間など	レビュー議事録など
		レビュー作業密度	レビュープロセスの品質, もしくはレビュー対象の品質	レビュー工数率, レビュー密度など	レビュー議事録 (ソースコード, レビュー対象各設計書)など
		レビュー指摘率[推移]	レビュープロセスの品質, もしくはレビュー対象の品質	レビュー指摘数, 指摘密度, 不具合指摘数推移など	レビュー議事録 (ソースコード, レビュー対象各設計書)など
要件実現/確認 (テスト・検収)フェーズ	品質	欠陥件数[推移]	テスト設計の品質とコード品質		テスト結果, 障害管理票, 品質管理表など
		欠陥対応件数	欠陥の対応進捗, 対応内容	不具合消化数, 障害滞留時間, 検出欠陥の分析実施回数, 類似欠陥調査実施回数など	テスト結果, 障害管理票, 品質管理表など
		欠陥密度	テスト設計の品質とコード品質	検出欠陥数をシステム規模で割る: 不良摘出件数 ÷ ステップ数など	テスト結果, 障害管理票, 品質管理表, ソースコードなど
		欠陥指摘率	テスト設計の品質	不良摘出件数 ÷ テストケース消化数	テスト計画書, テストケース, テスト結果, 障害管理票, 品質管理表など
要件変更・改善/対応フェーズ	要件定義	変更[推移]	変更された要件数	規模の計測単位に依存	要件定義書, 要件定義書の変更履歴など
	計画・管理	会議実施状況	ユーザ・ベンダ間, ベンダ間での情報共有状況を把握	各会議の時間, 参加人数, 資料量, 議事数, 議事録量, 情報共有者数など	議事録など
		累積リスク項目数	リスク認識が十分であったかを把握	進捗会議で挙げられたリスク項目数の累積, 軽微, 重大, その他の項目でわかる	議事録, リスク管理表など
		リスク項目	リスク対策が適切	リスク管理項目の最長, 平均	議事録, リスク管理

		の滞留時間	になされていたかを把握	滞留時間	表など
--	--	-------	-------------	------	-----

2.3. 定量的管理フェーズのモデル化

定量的管理フェーズの具体的内容は、現場への導入や事例の再利用を可能にするため、定量的管理モデルとして記述する。

定量的管理モデルには、以下の点を示す。

- 計測目的（ユーザ・ベンダ別）
- 監視と是正処置例
 - ▶ 尺度（計測データ）：具体的に監視する計測データ
 - ▶ チェックタイミング：計測値をチェックするタイミング
 - ▶ 解釈/判定例：チェックする観点、もしくは判定する基準など
 - ▶ 是正処置例：要求品質の確保のために必要な、取りうる処置の例
- 尺度（計測データ）の可視化例

例として、要件抽出／支援フェーズの定量的管理モデルを以下に示す。ただし、ここで示す可視化例（図 2-3 図 2-4）はサンプルデータで、3章のプロジェクトとは関係ない。

計測目的（ユーザ・ベンダ別）

ユーザ目的	要件抽出・要件定義のための作業を全関係者（ステークホルダ）から、漏れなく十分に行っているか、判断する
ベンダ目的	ユーザからの要件について、機能・非機能要件が網羅されているかのチェックを十分に行っているか、また各要件の確定度を把握し、ユーザと合意を行っているか、判断する。

監視と是正処置例

表 2-3 監視と是正処置例(要件定義フェーズ)

No.	尺度（計測データ）	チェックタイミング	一般的な解釈/判定（チェックポイント）	是正処置（対策）例
1	ユーザヒアリングでの要件追加数	ユーザヒアリング完了時	追加要件が 0 件に収束しているか？	<ul style="list-style-type: none"> ● ユーザ：参加メンバーを見直して、再度ヒアリングを行う ● ベンダ：追加要件のリスクを把握し、開発計画時に構成管理、スケジュールなどについて、要件追加に配慮する
2	設計工程	要件定義完	設計工程で確定可能	● ユーザ：設計工程でのレビュー

	で確定可能な未確定要素の割合	了時	な未確定要素の割合は高くないか？	ーに参加し、対象の要件について確認する <ul style="list-style-type: none"> ● ベンダ：対象の要件の設計作業を後回しにする、もしくはベンダ側の設計の完成度が低い状態でもユーザレビューを優先して行う
3	テスト工程まで確定困難な未確定要素の割合	要件定義完了時	テスト工程まで確定困難な未確定要素の割合は高くないか？	<ul style="list-style-type: none"> ● ユーザ：選択枝や許容範囲を明確にし、テスト結果に応じて即応できるように準備する ● ベンダ：優先してテストを実施するようスケジュールを調整する

尺度（計測データ）の可視化例（サンプルデータ）

図 2-3 は、ユーザヒアリング毎の要件の追加数を示している。同様に、要件の変更数も有用と考えられる。このヒアリングは、ユーザ企業内部でシステム開発部門と運用部門、もしくはベンダとユーザ間で行われることが考えられる。システムによって要件の分類なども異なることが考えられるが、ここでは[5]の要件分類を参考にしている。

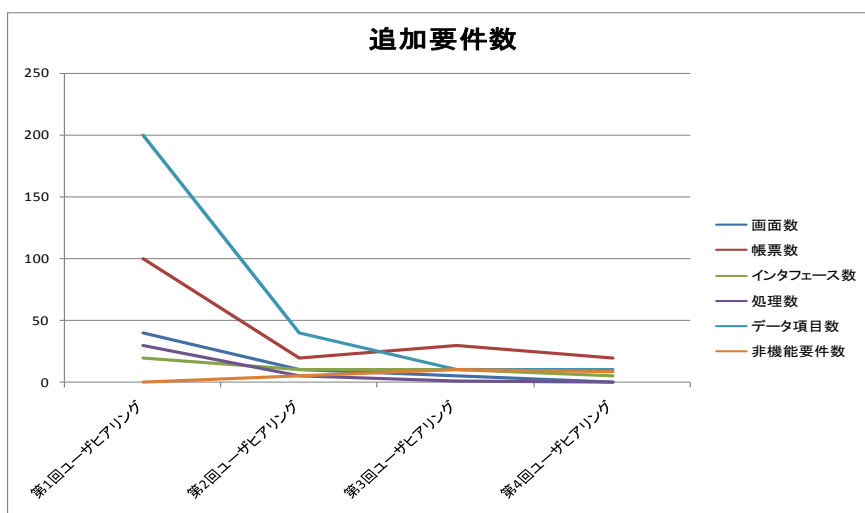


図 2-3 ユーザヒアリング毎の要件の追加数(サンプルデータ)

図 2-4 は、要件分類ごとの要件確定度を計測している。例えば画面要件に関して、100画面があるとすると、確定しているのは 20%弱で、設計工程で確定可能と推定している画面が 20%弱、テスト工程で確定もしくは変更が発生すると思われるものが約 55%，要否を含め未確定の画面が 10%弱あると解釈することができる。これは、要件定義完了時にベン

ダ・ユーザの双方で調査し、開発開始後のリスク管理や要件変更管理について事前に準備や対策について合意するために用いられる。また、後工程で実際に変更や追加が生じた際に、許容範囲の指標値として用いることもできる。例えば、確定していたはずの画面で修正や追加が発生する、画面数が 100 を超えるなどのケースでは、ベンダ・ユーザの双方でスケジュールやコストについての交渉・追加契約が必要になると考えられる。

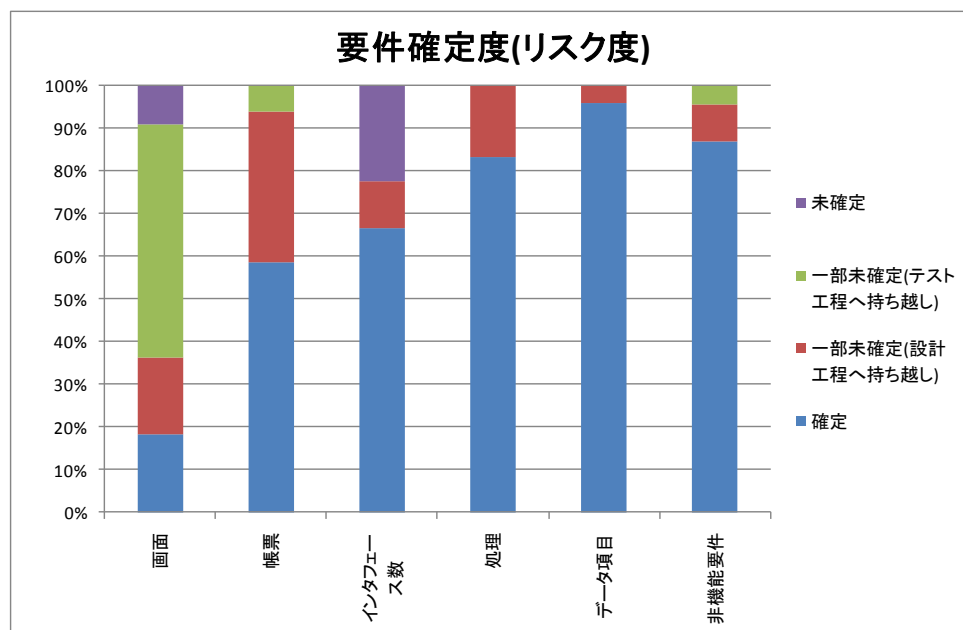


図 2-4 機能別要件確定度(サンプルデータ)

3. 定量化事例

本章では、2.2節で挙げたいくつかの定量的管理フェーズに対して、定量的管理モデルと実プロジェクトでの実測結果、インタビュー等による計測結果の評価について記述する。また、2.2節に記述しなかったフェーズとして、保守・次開発フェーズを追加する。これは、開発完了後の保守や次開発での管理、もしくは組織的なプロセス改善などが含まれる。また、あくまで完了したプロジェクトデータの事後分析であるため、「監視」と「是正処置」は、実際に行ったわけではなく、実測結果に対する開発関係者へのインタビューなどに基づき、例として挙げるものである。

3.1. 適用プロジェクトの概要

本調査で用いたデータのプロジェクト概要を以下に述べる。

- (1) システム種類：金融取引処理システムの拡張開発（新規を含む改造開発）
- (2) 開発方式：ウォーターフォールモデル（フェーズドアプローチ）
- (3) 開発言語：COBOL, C, C#, VB(VBA)
- (4) 規模：中・大規模
- (5) 開発期間：2年10か月

※提供データには開発完了後8か月後までのバグ票が含まれている。

- (6) 開発体制：システム発注者(ユーザ) → 受注者(ベンダ)

※ベンダは、統括ベンダ以下、複数の開発ベンダ（独立系、子会社、海外ベンダ等）を含む

3.2. 提供データ

以下に、システム発注企業及び統括ベンダ企業からご提供頂いた主なデータを上げる。

1. 連絡票（ベンダ→ユーザ, ユーザ→ベンダ双方向の仕様確認票）
2. 仕様変更依頼書（ユーザ→ベンダ・一覧のみ）
3. 設計書レビュー議事録（改造・機能設計書）
4. バグ票（机上デバッグ～稼働後案件検査）
5. プロジェクト計画書
6. 機能設計書（システム設計書）
7. 品質評価報告書
8. 工程会議資料

本レポートで実際に定量化に用いたデータは、1～4である。5～8に関しては、プロジェクトの背景や体制、期間、各機能の特性、規模・テスト情報など概要を把握するために用いた。いずれも電子データで提供されたが、一部はPDFなどで、定量化するにあたって、ExcelやAccessなどへの入力を行った。

3.3. データ収集と共有

図 3-1 は、対象プロジェクトでの開発工程と、授受・共有データを示している。図中に現れるデータのうち、要件定義書、プログラムコード以外はその一部、もしくは全部を提供を受けている。

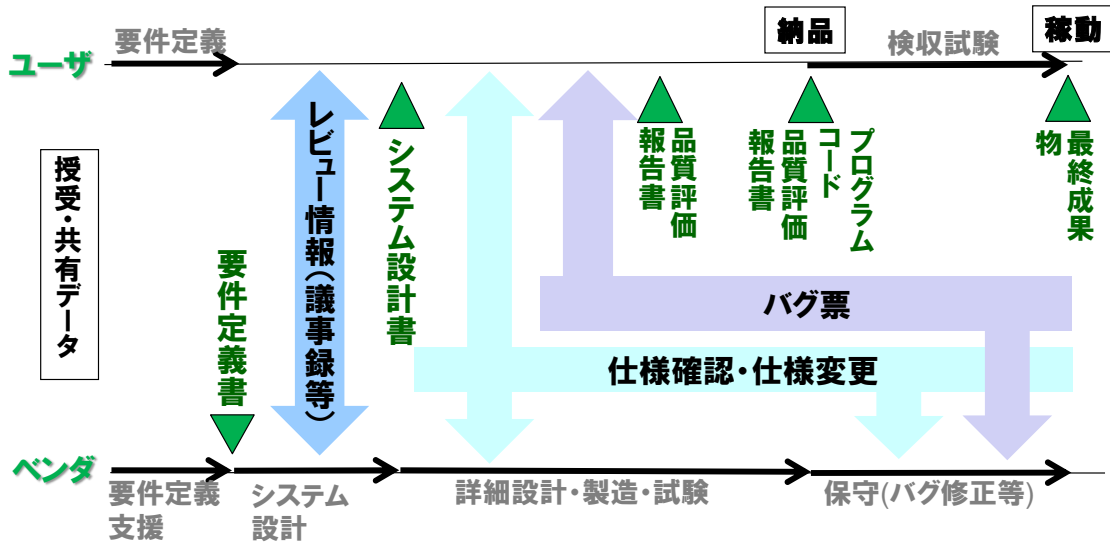


図 3-1 開発工程と授受・共有データ

3.4. 実測フェーズ

本レポートで実測したフェーズを以下に示す。それぞれのフェーズが計測対象とする作業の範囲を図 3-2 に示す。

- 計測例① 要件伝達フェーズ
- 計測例② 仕様変更／要件変更・対応フェーズ
- 計測例③ 要件実現・確認（テスト・検収）フェーズ
- 計測例④ 保守・次開発フェーズ

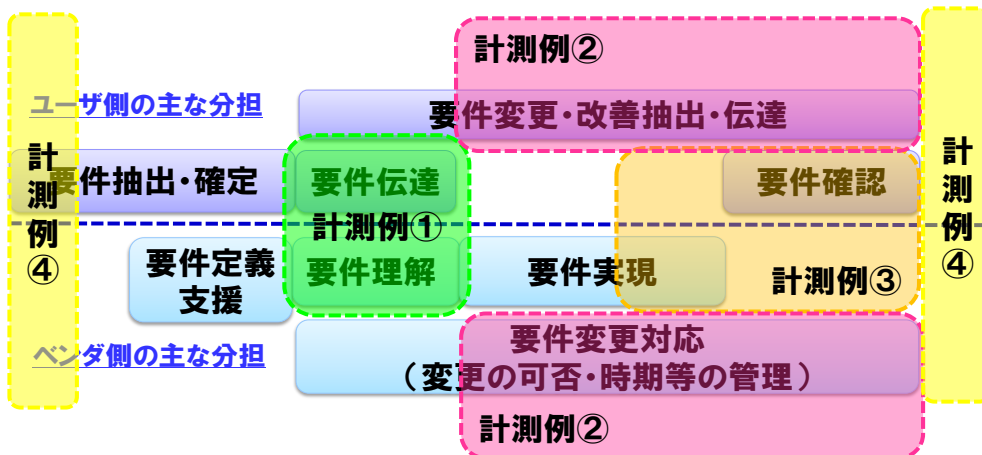


図 3-2 本レポートで実施した計測事例

3.5. 加工データ

表 3-1 は、本レポートで紹介する実際に計測したデータの一覧表である。「元（生）データ」は、提供データそのものを示し、計測事例で分析するデータは「尺度（計測データ）」であり、3.4 節の計測例①～④に対応している。「データ粒度」は、定量化・可視化の単位であり、「抽出（計算）方法」には計測データの加工方法、すなわち抽出、分類、定量化の方法を記述する。「収集期間（工程）」は、元（生）データの収集期間である。また、各行の色は、図 3-1 の授受・共有データの色に対応している。

表 3-1 要件管理フェーズと活用する計測データ
(①～④は 3.4 節のフェーズに対応。色は、図 3-1 のデータ種類に対応。)

元（生）データ	尺度（計測データ）	データ粒度	抽出（計算）方法	収集期間（工程）
ベンダ内レビュー議事録	①不良現象別指摘件数	機能別	レビュー指摘を不良現象別 ⁶ に分類した数。非不良・記述不正を除く。	システム設計
ユーザ・ベンダレビュー議事録	①不良現象別指摘件数	機能別	レビュー指摘を不良現象別に分類した数。非不良・記述不正を除く。	システム設計
仕様確認票（ベンダ→ユーザ）	②対応要否別仕様確認→回答	質問単位	対応（実際に変更等の開発作業が発生するかどうか）は、各票に対応可否の分類有。	システム設計完了後、検収テスト終了まで。
仕様変更依頼票（ユーザ→ベンダ）	②対応要否別仕様変更依頼→回答	変更依頼単位	対応（実際に変更等の開発作業が発生するかどうか）は、各票に対応可否の分類有。	システム設計完了後、検収テスト終了まで。
バグ票	③区分別要件変更発生件数（推移）	機能別	区分（判別） ⁷ が、仕様不良、仕様変更、仕様改善、ドキュメント不良（ベンダ）、ドキュメント不良（ユーザ）のそれぞれの発生数を時系列で表示。	机上デバッグ～稼働後案件対応の試験工程
	④要件変更原因別バグ件数	機能別	全バグデータから、“仕様/要件”のキーワードで抽出したバグ票について、現象・原因・対策等自由記述の内容を基に、手作業で分類。分類件数：全 285 件。	

⁶ 設計漏れ、設計誤り、設計不明瞭、設計改善など。詳細は、3.6 節参照。

⁷ その他、プログラム不良、デグレード、修正不十分、仕様通り、指摘ミス、テストミスなどがある。

図 3-3 は、表 3-1 のデータ粒度、抽出方法を示している。「生データ（実証データ）」とは、提供されたデータを単純に集計・フォーマット変換するだけで尺度（計測データ）として用いることができたデータである。一方、加工・抽出データは、人手で分類したり生データの一部を抽出したりして用いたものであり、単純に提供された生データと計測データを照合できないケースを示している。別プロジェクトで同じ分析を再現するために、どのような手順（手法）で加工・抽出したかを明確にしておく必要がある。

バグ情報を機能別に加工するにあたっては、まずバグ票の業務 ID（バグの発生したテスト対象の業務名）を機能にマッピングした。業務が未入力、あいまいなもの、複数の業務にまたがっているなどで機能に分類できなかつたりするケースは、「不明」に分類した。また、レビュー情報の機能分類とバグの機能分類が、一部異なっている。3.6 節の機能 B-1～4 は、3.8 節 3.9 節では機能 B に統合されている。バグの機能 M は別の機能のレビューに含まれて、レビュー情報を分割できなかったため、3.6 節のデータには存在しない。また、「その他」はレビュー情報とバグ票では対象が異なるため、機能名に置き換えなかった。

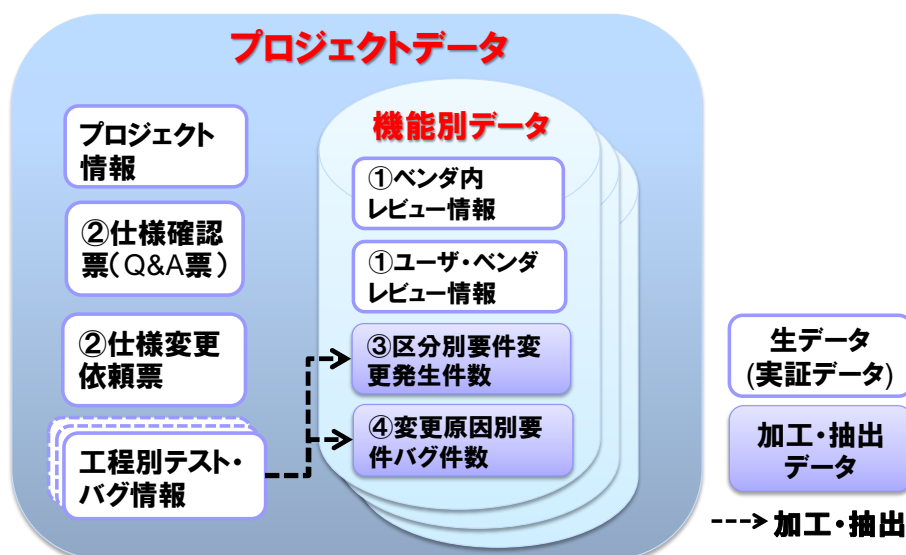


図 3-3 分析用加工データ構造

3.6. 計測例①:要件伝達フェーズ

計測目的 (ユーザ・ベンダ別)

ユーザ 目的	設計書について、十分に理解できているか、設計の間違いや不明瞭な点、改善を十分に指摘できているか、判断する
ベンダ 目的	ユーザ要件に対して、要件[定義書]が十分に理解できているか、漏れや間違いなく設計に反映できているか、判断する

監視と是正処置例

表 3-2 監視と是正処置(要件伝達フェーズ)

No.	尺度 (計測データ)	チェックタイミング	一般的な解釈/判定 (チェックポイント)	是正処置 (対策) 例
1	ベンダ内・ユーザーレビューの指摘数	ユーザーレビュー完了時	ユーザーレビューで指摘数が増加していないか?	<ul style="list-style-type: none"> ● ユーザ：要件定義書に不足や不明確な点がないか，再確認する ● ベンダ：ユーザーの理解促進のために，設計書の記述方法の改良を行う
2	設計漏れ・設計ミス数	ユーザーレビュー完了時	設計漏れや設計ミスが，ユーザーレビューで多く見つからないか?	<ul style="list-style-type: none"> ● ユーザ：要件定義書について，ベンダに対して口頭で説明する ● ベンダ：要件定義書に基づいて，設計書の再レビューを行う
3	設計改善割合	ユーザーレビュー完了時	各機能（サブシステム）に対して，設計改善要求が一定量行われているか?	<ul style="list-style-type: none"> ● ユーザ：再度，設計書について詳細に検討し，別部門のユーザーなども含めたレビューを行う ● ベンダ：設計について，後から変更・改善要望の発生の可能性が高い点について，重点的に再確認する

尺度 (計測データ) の可視化例

図 3-4 図 3-5 は，機能別のベンダ内レビュー，ユーザー・ベンダ間レビューでの指摘数と不良分類の内訳を記している。システムによって，機能別，要件分類別，担当組織別に計測することが考えられる。縦軸のスケールは統一し，件数が比較できるようになっているが，表 3-2 の項目 3 のように割合をチェックするときは，割合を示す棒グラフにする方がわかりやすい。

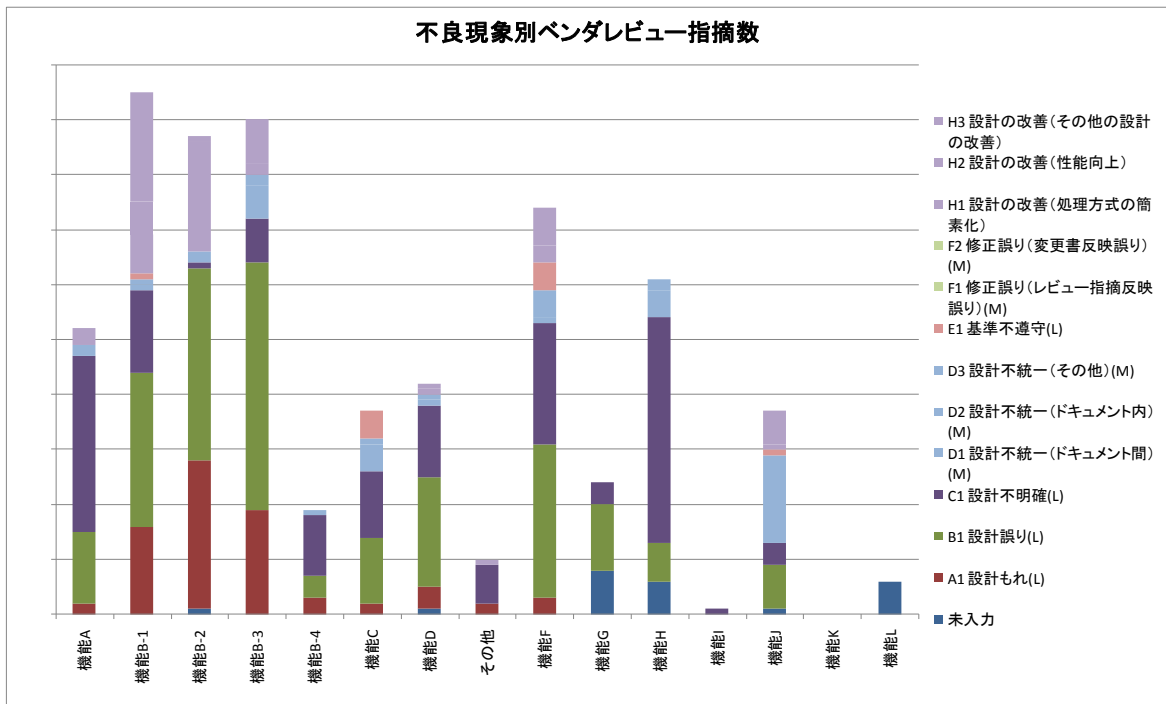


図 3-4 不良現象別ベンダレビュー指摘数

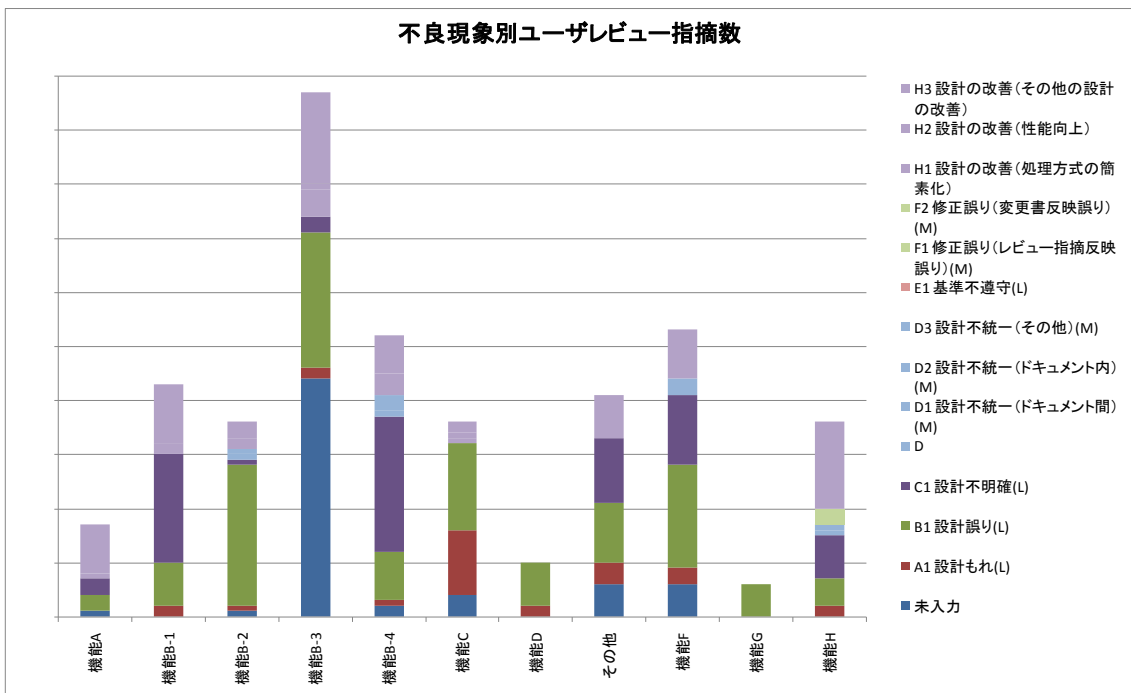


図 3-5 不良現象別ユーザレビュー指摘数

計測結果の評価

実際に計測した図 3-4 図 3-5 をインタビュー等で確認した結果、機能 B・3・B-4 は機能が複雑で、ベンダの作成した設計書がユーザ側の意図より複雑だったため、全体の指摘数

がユーザレビューで増加したということがわかっている。また、機能 C については、ユーザレビューでの設計漏れがベンダレビューより多いが、参照した外部規格の仕様書が英語で、規格自体が複雑でパターンが多く、ベンダ側で理解し設計に落とすことが困難であったことが分かっている。

3.7. 計測例②:仕様変更/要件変更・対応フェーズ

計測目的 (ユーザ・ベンダ別)

ユーザ 目的	要件確認について、適切な対応を取っているか、判断する 要件変更について、優先度や重要度を加味して、適切な手順でベンダに依頼しているか、判断する
ベンダ 目的	要件確認について、質問の内容や形式が、ユーザの回答(決定)を促すように適切に行われているか、確認する 仕様変更について、変更内容や影響範囲に応じて、対応の可否を適切に判断しているか、確認する

監視と是正処置例

表 3-3 監視と是正処置(仕様確認・変更フェーズ)

No.	尺度 (計測データ)	チェックタイミング	一般的な解釈/判定 (チェックポイント)	是正処置 (対策) 例
1	変更依頼数と発生時期	設計開始後 (ベンダ側の要件実作業開始後)リアルタイム (進捗会議など定期的)	テスト工程の後半に変更依頼数が頻繁に発生していないか?	<ul style="list-style-type: none"> ● ユーザ: 依頼内容が適切か (稼働後対応や保留にできるものがないかなど)をユーザ内で確認する手順を実施する ● ベンダ: 依頼による変更可否の条件を明確にし、ユーザと合意する
2	変更依頼に対する回答までの日数	同上	変更依頼に対する回答(対応可否・影響度など)に時間がかかるケースが多くないか?	<ul style="list-style-type: none"> ● ユーザ: 変更依頼の内容を明確にする ● ベンダ: 変更依頼に関する開発(回答)担当者の負荷や回答手順を見直す
3	変更依頼の保留・見送り割合	同上	変更依頼に対して、対応が保留や見送りになる割合が高くないか?	<ul style="list-style-type: none"> ● ユーザ: 変更依頼の優先度や重要度を検討し、高いもののみを依頼するようにする

				<ul style="list-style-type: none"> ● ベンダ：依頼による変更可否の条件を明確にし，ユーザと合意する
4	要件確認数と発生時期	ベンダ側の作業（要件定義支援など）開始後，リアルタイム（進捗会議など定期的）	テスト工程の後半に仕様確認数が頻繁に発生していないか？	<ul style="list-style-type: none"> ● ユーザ：仕様確認の発生する要件について，ベンダと再レビューを行う ● ベンダ：仕様確認の発生する要件について，ユーザと再レビューを行う
5	要件確認に対する回答までの日数	同上	仕様確認に対する回答に時間がかかるケースが多くないか？	<ul style="list-style-type: none"> ● ユーザ：回答担当者の負荷や回答手順を見直す ● ベンダ：仕様確認の内容について，ユーザの決断を容易にする情報を付加し，選択式（提案式）の質問とする
6	要件変更等対応が必要な仕様確認の結果割合	同上	ドキュメント・プログラムの変更などの対応が必要になる仕様確認の割合が高くないか？	<ul style="list-style-type: none"> ● ユーザ：要件定義書や設計書を見直し，あいまいな点についてベンダに確認する ● ベンダ：仕様確認による変更可否の条件を明確にし，ユーザと合意する

尺度（計測データ）の可視化例

図 3-6 図 3-7 は，仕様確認票もしくは要件(仕様)変更依頼を可視化している．図 3-6 の要件確認の場合は，質問者(ベンダ)が赤で，回答者(ユーザ)が青いマークで示される．図 3-7 の要件(仕様)変更依頼の場合は，逆に依頼者(ユーザ)が赤で，回答者(ベンダ)が青いマークで示される．質問のマークは□△○+があるが，これは質問形式（Open-Question/Closed Question など）で分類していて，回答の容易さを示す．横軸は経過日数で，縦軸は質問もしくは要件(仕様)変更依頼番号である．つまり，横線の一本上のマークが，1つの質問，変更依頼に対するイベントを示す．塗りつぶされたマークは，質問もしくは依頼結果，何らかの対応（要件変更，プログラム変更等）が必要になったケースで，塗りつぶしていないものは回答の時点で完了したケースである．工程表などと照合し，発生した時期・工程などを把握することもでき，発生原因の解析などに有用である．質問や変更依頼の発生頻度・発生間隔，対応が必要なケースの割合，回答・解決までの日数などの数量化も可能である

が、ここでは質問・変更依頼の個別の可視化例のみ示す。

- A : Yes/No質問, (対応要) □ A : Yes/No質問, (対応不要or不明) + H : その他(対応要)
- a : Yes/No(対応要) □ a : Yes/No(対応不要or不明) + h : その他回答(対応要)
- ▲ B : 複数選択, (対応要) △ B : 複数選択, (対応不要or不明)
- ▲ b : 選択(対応要) △ b : 選択(対応不要or不明)
- C : 詳細情報の要求(対応要) ○ C : 詳細情報の要求(対応不要or不明)
- c : 詳細説明(対応要) ○ c : 詳細説明(対応不要or不明)

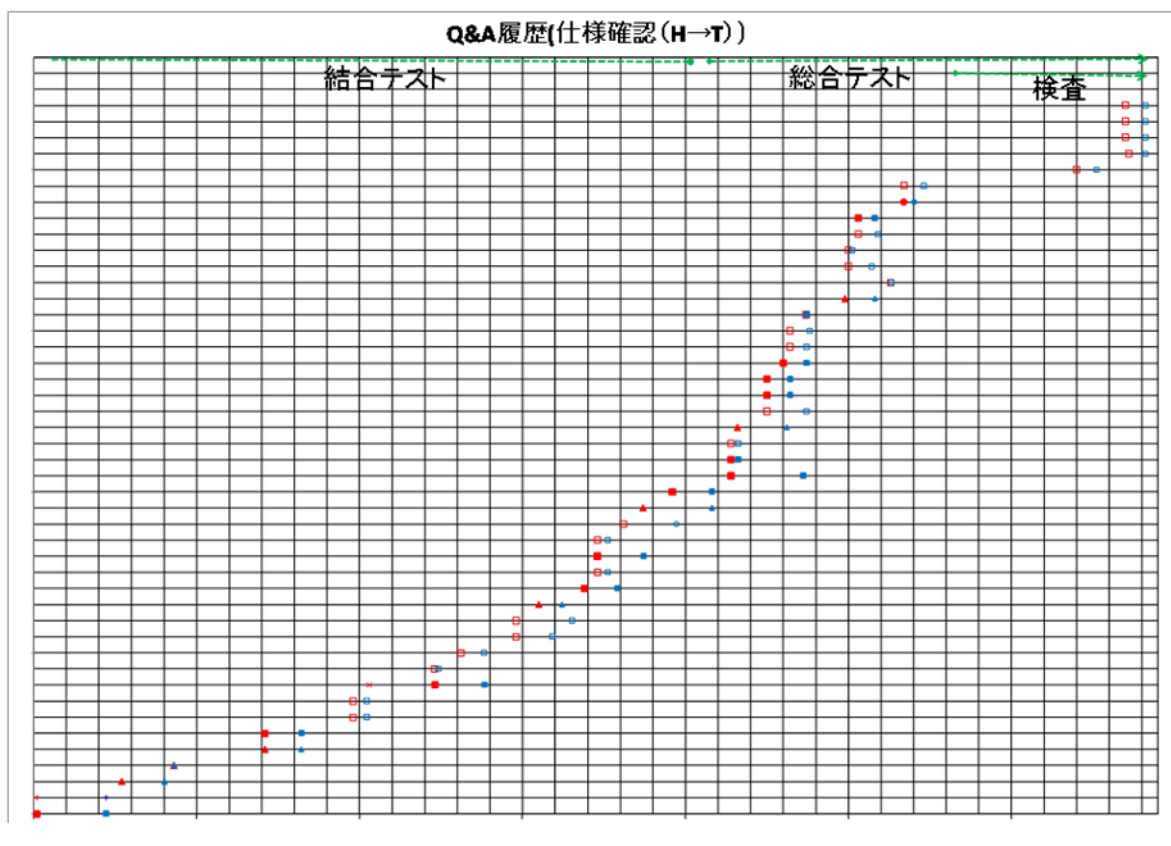


図 3-6 要件(仕様)確認(ベンダー→ユーザ)

計測結果の評価

図 3-6 では、結合試験から総合試験にかけて、対応が必要な仕様確認が集中しているが、これはテストケースを作成する段階で発生した質問であったことを、開発関係者から確認している。テストケースを設計段階で作成していれば、早期に対応できた（手戻りが発生しなかった）というご意見をいただいた。また、質問形式は、ほとんどが Closed Question(Yes/No, 選択型質問)であり、ユーザへの負担が少ない質問であったことがわかる。

- ◆ G : 仕様変更要求(対応要) ◇ G : 仕様変更要求(対応不要or不明)
- ◆ g : 仕様変更回答(対応要) ◇ g : 仕様変更回答(対応不要or不明)

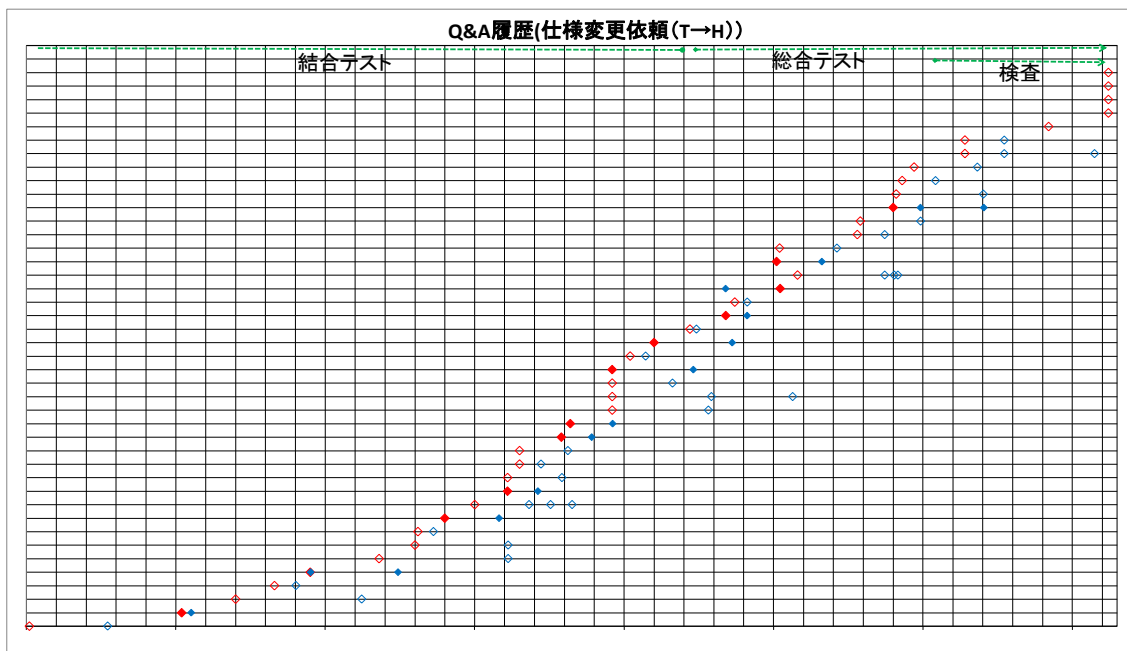


図 3-7 仕様変更依頼(ユーザーベンダ)

計測結果の評価

図 3-7 の変更依頼は、すべてを網羅していない。バグ票で管理されているものもあったため、図 3-7 以降も変更依頼は発生している。この図からは、結合テストの後半あたりで仕様変更依頼が頻発していることがわかるが、一方、対応を要したものの割合は少なく、ユーザ側で変更依頼にあたって、重要度や優先度を検討不足だったことも考えられる。最終的な回答まで非常に時間がかかっているケースも多く、ベンダ側で変更依頼に対する調査に時間がかかっていたと考えられる。この図から、要件(仕様)変更について、プログラムの変更・再テストなど実際の手戻り作業以外に、可否調査などにかかる工数も把握できる。

3.8. 計測例③:要件実現・確認(テスト・検収)フェーズ

計測目的 (ユーザ・ベンダ別)

ユーザ 目的	要件に関するバグ(票)で、ユーザ側作業に起因するものの発生数が許容範囲か、早期に発見され完了までに収束しているか、判断する
ベンダ 目的	要件に関するバグ(票)で、ベンダ側作業に起因するものの発生数が許容範囲か、早期に発見され完了までに収束しているか、判断する

監視と是正処置例

表 3-4 監視と是正処置例(要件実現・要件確認フェーズ)

No.	尺度 (計測データ)	チェックタイミング	一般的な解釈/判定 (チェックポイント)	是正処置 (対策) 例
1	仕様不良数	テスト工程開始からリアルタイム(進捗会議など定期的)	仕様不良数が多くな いか?	<ul style="list-style-type: none"> ● ユーザ・ベンダ: 要件定義書と設計書の整合性について, 双方でレビューを行う
2	仕様変更数	テスト工程開始からリアルタイム(進捗会議など定期的)	仕様変更がテスト工 程後半(検収テスト など)で増加してい ないか?	<ul style="list-style-type: none"> ● ユーザ: 仕様変更の多い機能などで, ユーザテスト(検収テスト)を集中して行う ● ベンダ: 仕様変更に対する変更可否の条件を明確にし, ユーザと合意の上, 対処する
3	ベンダドキュメント不良数	テスト工程開始からリアルタイム(進捗会議など定期的)	ベンダドキュメント 不良数が早期に収束 していないか?	<ul style="list-style-type: none"> ● ユーザ: ユーザテスト(検収テスト)を前倒しして行う ● ベンダ: 要件定義書に対応させてテスト仕様書を見直し, 再テストなどを行う

尺度 (計測データ) の可視化例

図 3-8 は, テストによる要件関連の問題の指摘効果を時系列に示している. 横軸は時間, 縦軸は発生件数(累計)を示している. 5つの分類は, 元のバグ票に存在し, バグ票記入者が入力するが, 必ずしも明確な分類基準があるわけではないようである. 提供データにおける各分類は, 開発者へのインタビューやバグ票の内容から次のように解釈している.

- ドキュメント不良 (顧客): 要件定義 (書) ミス. ユーザ責任.
- ドキュメント不良 (ベンダ): 詳細設計書, マニュアル (操作仕様書), テスト仕様書など, ベンダが作成した文書のミス, もしくはそれに起因するバグ.
- 仕様改善: 比較的軽微な要件・仕様の変更. 画面デザイン, レスポンス時間など性能に関するものが多い. ユーザからの指摘が多い.
- 仕様不良: 要件や設計書との不整合など. ユーザとのレビューで検出できなかったとすれば, ユーザ・ベンダ双方の責任と考えられるが, ベンダの詳細設計以降に混入したものであれば, ベンダの責任になる.

- 仕様変更：重要な要件・仕様の変更。機能の追加や変更。ユーザからの指摘によるものが多い。

この計測データは、計測目的によって、ユーザ・ベンダテスト別やテスト工程別に計測することも考えられる。

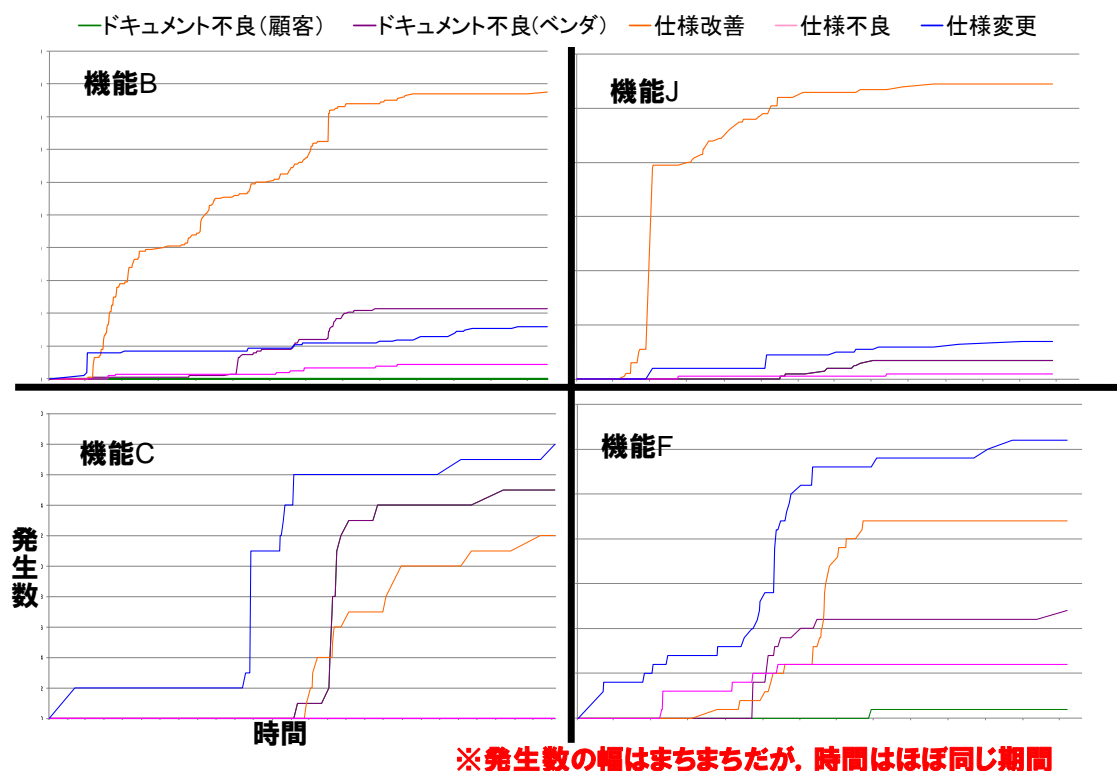


図 3-8 機能別要件・仕様バグ票発生推移

計測結果の評価

図 3-8 は、特に特徴的な 4 機能について示している。発生数の縦軸は、それぞれ範囲がバラバラで件数を比較することはできないが、横軸の時間は共通の期間を示している。「機能 B」や「機能 J」では仕様改善が多いが、「機能 J」のほぼ垂直の仕様改善数の増加は、机上で品質向上のチェックを集中的に行った結果であることが、インタビューから判明している。また、「機能 C」「機能 F」では件数は多くないが、要件変更が最も多い。要件が不安定な機能であることがわかるとともに、両機能ともほぼ同じ時期に要件変更が急増しているため、この時期の作業を見直すことによって、要件変更を前倒しして抽出するなどの改善が可能になると考えられる。

3.9. 計測例④:保守・次開発フェーズ

計測目的 (ユーザ・ベンダ別)

ユーザ目的	対象機能(サブシステム)の要件変更の主な原因から、保守・次開発時の要件管理プロセスについて監視・改善が必要か、判断する。判断内容に関してユーザ・ベンダで合意し、双方の責任分担に応じて、要件に関する監視・制御を実施する。
ベンダ目的	

監視と是正処置例

表 3-5 監視と是正処置例(保守・次開発フェーズ)

No.	尺度 (計測データ)	チェックタイミング	一般的な解釈/判定 (チェックポイント)	是正処置 (対策) 例
1	ベンダ要因要件変更の割合	開発完了時、もしくはベンダ作業完了時	ベンダ要因要件変更の割合が多くないか?	<ul style="list-style-type: none"> ● ユーザ：ベンダ作業の各段階(工程区切りなど)で、要件定義との整合性確認レビューを行う ● ベンダ：要件定義書と設計以降の作業のトレーサビリティを確保する手段を講じる
2	ユーザ要因要件変更の割合	開発完了時、もしくはベンダ作業完了時	ユーザ要因要件変更の割合が多くないか?	<ul style="list-style-type: none"> ● ユーザ：要件定義プロセスを見直す。要件確定度を開発前に明確にする。 ● ベンダ：
3	外部要因要件変更の割合	開発完了時、もしくはベンダ作業完了時	外部要因要件変更の割合が多くないか?	<ul style="list-style-type: none"> ● ユーザ：外部からの変更のリスクを把握し、リスク管理を行う ● ベンダ:変更リスクに対して、対応容易な設計を行う

尺度 (計測データ) の可視化例

図 3-9 では、要件・仕様変更の原因割合を示している。原因分類は開発担当者へのインタビューから抽出したが、責任分担者によって、大きく4つに分類できる。

- ベンダ原因：赤い部分 (1~3) は、要件・仕様に明記されているにもかかわらず、開発者のミスなどによるバグで、開発ベンダ内のコミュニケーション、あるいは確認 (チェック) が原因で発生した要件変更である。単純な仕様理解・伝達ミスその他、同一システム内のサブシステム間の I/F の不整合、メモリ容量など用いる HW や外

部ソフトウェアの仕様の調査漏れが、ベンダ側のミスとして挙げられる。

- ユーザ原因：緑の部分には、ユーザ責任が大きく、要件定義から漏れていた、品質要件などが明記されていないなどの原因による要件変更のほか、画面の色や見栄え、使い勝手（画面操作や応答時間など）比較的重要度の低い軽微な変更まで、ユーザ主体で追加・変更される要件・仕様を示している。
- ユーザ・ベンダ原因：黄色い部分は、要件漏れ・ミス・見積もりミス・性能問題でも、要件仕様書もしくは機能仕様書に書かれているにもかかわらずレビュー等で見逃した要件のミスやあるいは双方が合意の上での要件変更を示し、これらについては双方に一定の責任があると判定できる。
- 外的要因：青い部分は、外的要因による変更を示す。外部システムとの接続が追加になる、外部からの受領データの仕様が特殊であるなど、当該システムの開発の責任範囲を超える環境の変化などへの対応である。ただし、外的要因による変更リスクは、要件定義時点でユーザもしくはベンダが把握し、リスク管理することによってその影響を限定することが可能である。

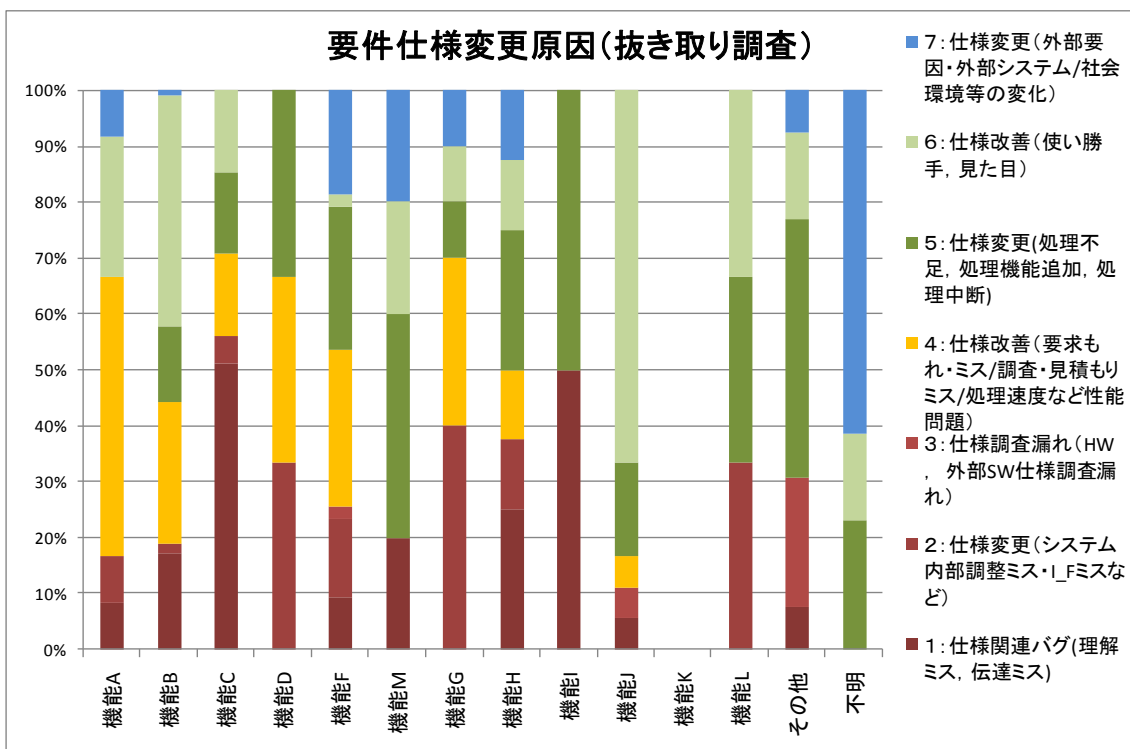


図 3-9 要件変更原因別割合

計測結果の評価

図 3-9 を見ると、「機能 C」はベンダ原因による仕様変更が多くを占めている。要件・機能が複雑で困難であったこともあるが、このようなケースでは、ユーザからベンダ間への

知識・技術移転などを重点的に行うべきと考えられる。「機能 F」「機能 M」については、外部のシステムとの接続仕様書の不備・変更による修正，追加の取引種類への対応などで，不可避な変更であったと考えられる。これらについては，今後の保守，運用においても変更のリスクが生じることを踏まえた保守・運用計画を必要とする。また，「機能 J」では，ユーザの操作画面が主な機能であり，見栄えや操作性など見てみないと分からないような仕様変更が多発することは事前に予測できる。このようなケースでは，本プロジェクトでもそうであったように，顧客レビューを行わず，テストを早期に始め，ユーザからの要求に対応する体制・時間・変更容易性を確保しておくことが重要となってくる。

3.10. 定量化事例に対する考察

データ収集について

計測例④（3.9 節）以外は，分析に必要なデータの属性はすべて入力済みで，データ収集は問題ない。④の分類項目（要件変更原因）が必要であるが，入力者がベンダの場合が多いので，入力基準を明確にしないと偏る可能性がある。むしろ，バグ票の情報から，第 3 者が判定した方が正確かもしれない。約 500 件のデータから，285 件の要件変更データを取り出し，さらに原因分類を行うのに，2 人日程度必要であった。システムについての知識がなくても，ソフトウェア開発の一般的な知識があれば可能である。

データ加工・可視化について

すべて電子データで提供されていたので，分析のための加工・可視化は，市販の表計算ソフト（MS-Access，MS-Excel など）で事務作業的にすることは可能。機能別の分類など 3.5 節で述べたとおり加工が必要であり，工程ごとに必ずしも分類が一致していないなどの問題がある。特に，分析の粒度についてはプロジェクト計画時（定量的プロジェクト管理計画時）に協議し，データの収集粒度・単位などについて統一するよう，ユーザ・ベンダで合意しておく必要がある。

相互確認・是正処置について

相互確認と是正処置（フィードバック・アクション）は実施していないが，指標値がなくてもデータから異常な状況や原因を開発担当者が把握することはできた。従って，進捗会議などで，ユーザ・ベンダ双方の開発担当者がチェックする材料としては，実績データの情報量・加工形態は十分であると思われる。是正処置は，開発完了後だったので実際に行われなかったが，データ解釈から可能と思われる対策・改善は，開発関係者からいくつか引き出すことができた。

4. 今後の課題

4.1. 現場での適用支援

定量的管理フェーズは、プロジェクトの規模やシステムの種類、開発プロセスモデルによって、異なる可能性がある。例えば、金融・社会インフラなどのシステムと、家電組み込みシステムでは、要件抽出方法や要件定義工程の重要度が異なる。そのため、すべてのプロジェクトにおいて、同じ管理フェーズや管理モデルを用いることはできない。

プロジェクトの特性やシステムの特性に合わせて、定量的管理フェーズの抽出・管理モデルの構築方法や、既存のモデルのカスタマイズ方法などを具体的に示し、現場のプロジェクト管理者の適用を支援する情報を提供する必要がある。

4.2. 定量的管理モデルの記述方法について

本レポートでは、定量的管理モデルについて、2.3節に記述内容をまとめているが、この内容で現場への適用・普及に十分な情報であるかの検証が必要である。3章の実プロジェクトでの事例や現場管理者の意見なども取り入れて、モデル化に必要な情報の調査・分析し、記述方法を統一するなど、普及に向けた準備が必要である。

4.3. 定量化事例について

図 4-1 に 3 章で実測結果を示すことができなかったフェーズを示す。

- 要件抽出／支援フェーズ：開発者からはユーザ・ベンダ間でチェックリスト等を用いた作業が行われていたとうかがっているが、データの提供が得られず計測できなかった。しかし、重要なフェーズであることから、計画的に実データによる事例を作成したいと考えている。
- 要件実現フェーズ：実質ベンダ内ではこの部分は厳密に管理されており、ユーザも重要性を認識していることから、ベンダで管理されている部分からユーザに関連するデータを抽出し、共有していく方向で、検討が必要である。組織間の機密保持とデータ共有の仕組みについて、分散・マルチベンダ開発でのプロジェクト管理方法[12]などが参考になると考えている。

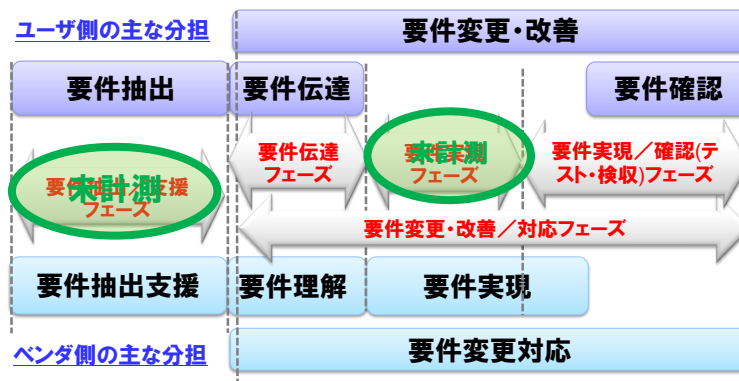


図 4-1 未計測管理フェーズ

4.4. 適用事例の蓄積と公開

本レポートで参照したプロジェクトは 1 つだけであるが、より多くの同じくユーザ・ベンダ（発注者・受注者）の関わるプロジェクトのデータを、プロジェクトの進行中だけでなく開発完了後の事後分析も含めて、提案する要件管理プロセスを適用し、その結果を蓄積・公開していくことによって、以下の効果が期待できる。

指標値の抽出

定量的な管理には、いくつかの計測値に関して指標値が必要となる。本レポートの定量化モデルでは、「監視と是正処置例」の解釈/判定に「～多いか?」「～割合が高くないか?」といった記述になっているが、実運用を行うにあたっては具体的指標値、例えば「10 件以上か?」「30～50%以内か?」などといった数値目標を上げるべきである。指標値を示すことによって、第三者、たとえばユーザ企業の経営者やシステム利用者が、客観的にそのプロジェクトの状況を把握することができるようになり、組織的なプロセス改善やユーザ・ベンダ間のプロジェクト管理に関する合意形成を容易にする。

適用の促進

プロジェクトの成否にかかわらず、ユーザ・ベンダ双方が公開されたデータをもとに、各自が抱えている問題や目標に役立つと判断できれば、適用を促進することができる。定量的管理に有効な方策として、実績データの開示や成功事例を挙げるユーザは多い(付録図 3 参照)。そのため、適用コストやカスタマイズの方法などを含め、できるだけ多くの適用事例を関係分野の組織、個人に公開する体制を作る必要がある。多くの場合、開発プロジェクトの事例は機密事項が多いため公開の制約は多いので、その点を考慮しつつ公開するシステムの構築が必要である。

ツールの整備

本レポートでの事例では、MS-Office などのソフトウェアを用いて、手作業でデータの集計、データベース化、分析、可視化を行った。しかし、ユーザもしくはベンダ企業で新たな収集・分析手法を取り込むには、手作業ではコストが高い。また、他のプロジェクトのデータを蓄積し、統計的に分析するなどの機能も必要になってくる。そのため、適用事例の蓄積から、汎用化できる部分、個別にカスタマイズが必要な部分などを切り分けた上で、ツール要件を抽出し、自動化を進める。

謝辞

本研究の一部は、文部科学省「次世代 IT 基盤構築のための研究開発」の委託に基づいて行われた。本研究にあたり多くのご協力を頂いたソフトウェアタグ規格技術委員会、ソフトウェア構築可視化に伴う法的諸問題に関する検討委員会の関係諸氏に感謝します。また、本レポートのデータ収集・分析活動にご協力いただきました株式会社東京証券取引所様、株式会社日立製作所様の関係各諸氏に心から感謝いたします。

参考文献

- [1] 「ソフトウェアタグ規格 第 1.0 版」, 平成 20 年 10 月 14 日 StagE プロジェクト (奈良先端科学技術大学院大学・大阪大学) 発行
- [2] Ian Sommerville and Pete Sawyer, Requirements Engineering: A Good Practice Guide , John Wiley & Sons Inc Print on (1997/4/16)
- [3] 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編, 共通フレーム 2007 共通フレーム 2007—経営者、業務部門が参画するシステム開発および取引のために (SEC BOOKS) , オーム社 , 2007/10.
- [4] 情報サービス産業協会(JISA), “情報サービス産業における受注ソフトウェア開発の実態アンケート調査結果報告,” JISA 会報, No.79, pp.53-85, October 2005.
- [5] 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編, 経営者が参画する要求品質の確保～超上流から攻める IT 化の勘どころ～ (SEC BOOKS) , オーム社 , 2006/5.
- [6] 久保浩三, “ソフトウェア構築可視化に伴う法的諸問題に関する研究,” 平成 20 年度第 1 回エンピリカルソフトウェア工学研究会, October 2008.
- [7] 経済産業省, 情報システムの信頼性向上のための取引慣行・契約に関する研究会～情報システム・モデル取引・契約書～ (受託開発 (一部企画を含む)、保守運用) < 第一版 >, 2007 年 4 月 13 日.
- [8] 開発のための CMMI®(CMMI-DEV) 1.2 版 公式日本語翻訳版, CMMI 成果物チーム, 技術報告書 CMU/SEI-2006-TR-008.
<http://www.sei.cmu.edu/cmmi/translations/japanese/models/dev-v12-abstract-j.html>
- [9] Project Management Institute, プロジェクトマネジメント知識体系ガイド(第3版) PMBOK ガイド, 2004.
- [10] 日経コンピュータ, “動かないコンピュータ別冊特別編集版,” 2009 年.
- [11] Ian Sommerville, Software Engineering Fifth Edition, Addison-Wesley, 1996.
- [12] 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.J92-D, No.11, pp.1974-1986, Nov. 2009.
- [13] 清田 辰巳, “ソフトウェアタグへの期待,” 平成 20 年度第 1 回エンピリカルソフトウェア工学研究会, October 2008.

付録

StagE Web アンケート(ユーザ編)の結果から(抜粋)

アンケート概要

調査件名：システム開発における見積と実態の差異に関する調査（ユーザ編／ベンダ編）

調査方法：Web による自記式アンケート

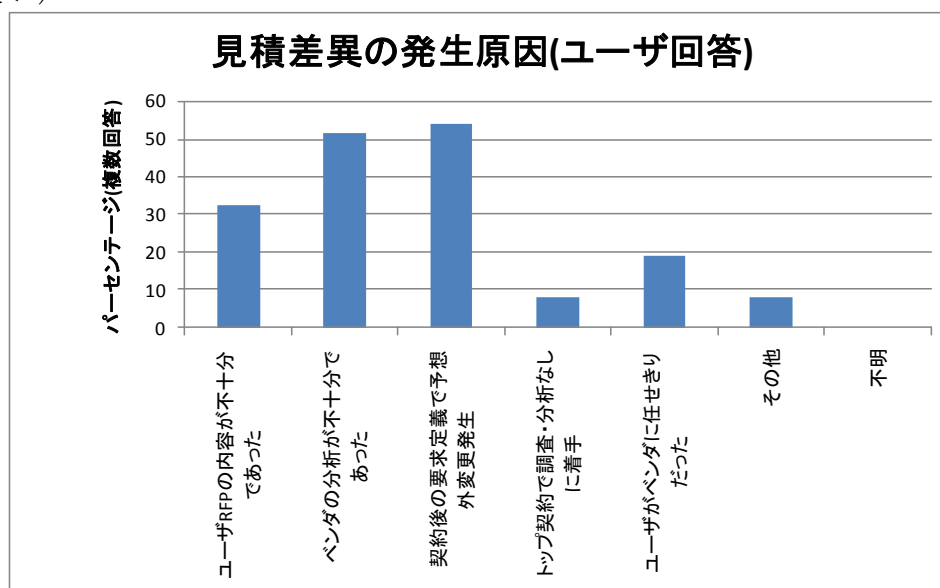
調査告知媒体：@IT パネルメンバー(調査協力意向者 46341 名)へのオプトインメール

調査対象：a) システム開発を依頼する立場（ユーザ）， b) システム開発を受託する立場（ベンダ）

調査期間：2009年3月12日～3月20日

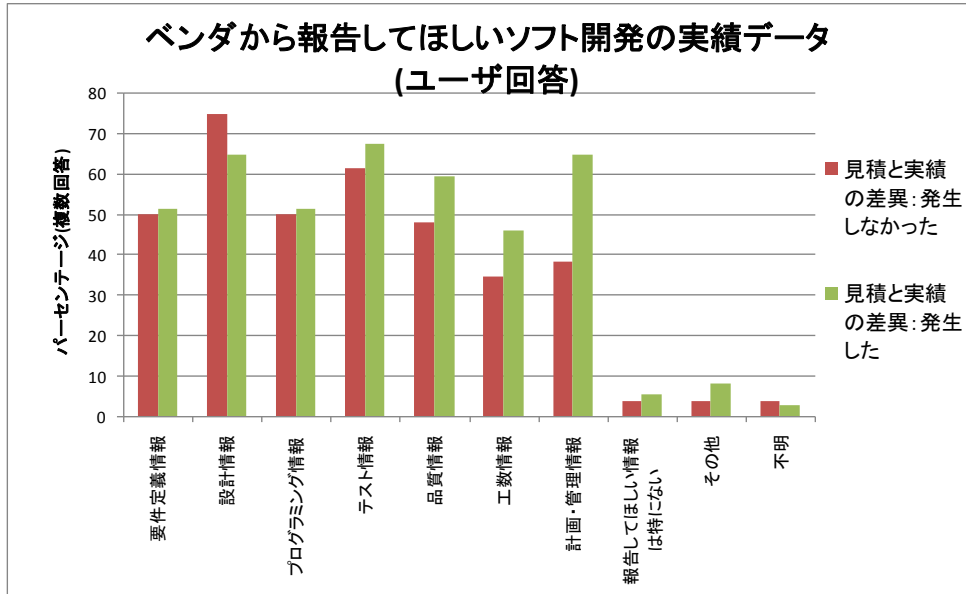
有効回答数：a) ユーザ編：89件， b) ベンダ編：213件

【Q8】 Q4で見積金額との差異が「2. 発生した」と回答されたユーザにお伺いします。その原因は何ですか。該当のものをお選びください。（あてはまるものを全てお答えください）



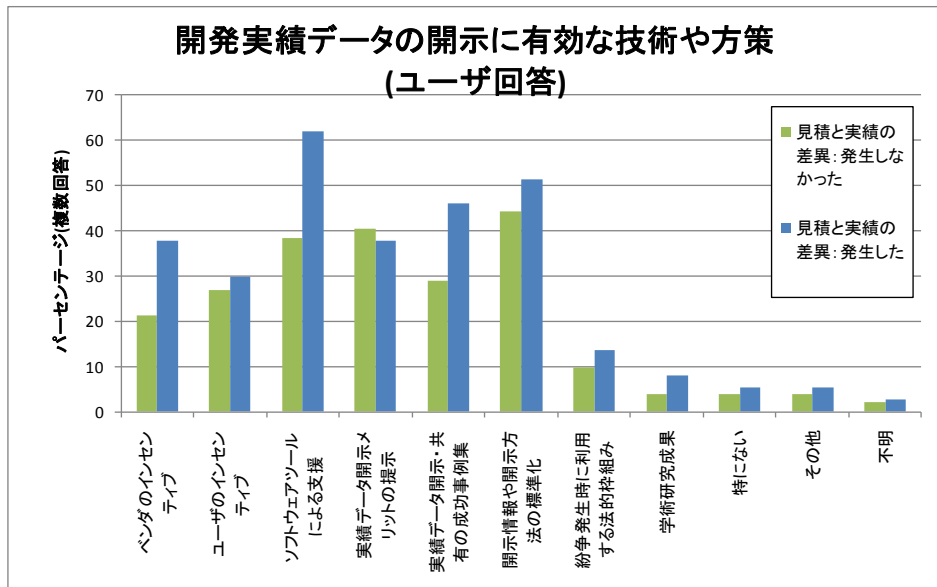
付録 図 1 見積もり差異の発生原因(ユーザ回答)

【Q14】納品されたソフトウェアの運用・保守のために、ベンダから報告してほしい（ベンダに報告を求めたい）ソフトウェア開発の実績データ（管理情報）をお答えください。（あてはまるものを全てお答えください）



付録 図 2 ベンダから報告してほしいソフト開発の実績データ

【Q15】より多くのソフトウェア開発の実績データ（管理情報）をベンダに開示してもらうために有効と考えられる技術や方策をお答えください。（あてはまるものを全てお答えください）



付録 図 3 開発実績データの開示に有効な技術や方策