

## 派生開発における変更工数の見積精度向上策

### - 「変更依頼仕分けガイド」の提案 -

#### Estimate precision improvement plan of change cost in the Derivation Development - Suggestion of the 'Change Request Classification Guide' -

主査	: 清水 吉男 (株式会社システムクリエイツ)
副主査	: 飯泉 紀子 (株式会社日立ハイテクノロジーズ)
副主査	: 足立 久美 (株式会社デンソー)
リーダー	: 蓑島 秀明 (テクニカルジャパン株式会社)
研究員	: 瀧野 宏一 (TIS 株式会社)
	中津留 稔 (アンリツエンジニアリング株式会社)

#### 研究概要

近年、開発工数の実績が見積を超過する派生開発プロジェクトが散見され、経営リスクの一つになっている。工数を超過したプロジェクトの不具合事例を調査したところ、テスト工程で発見された不具合は、設計工程まで手戻りが発生しているケースが多く見られた。これらの原因として、顧客からの変更依頼によっては、見積時に考慮されていない「隠れた要求／仕様」が存在することが分かった。派生開発において、「要求」と「仕様」が混在する変更依頼から「隠れた要求／仕様」を見つけるには、変更依頼から「要求」と「仕様」を適切に見極めることにある。しかしながら、その見極めが担当者の見積経験や当該システムの習熟度に依存する部分が多いことより、適切に判別できていないのが現状である。

本研究において、この「隠れた要求／仕様」に気付くためのアプローチとして、変更依頼を「要求」と「仕様」に判別するための「変更依頼仕分けガイド (CRCG:Change Request Classification Guide)」を作成した。このガイドによって、「隠れた要求／仕様」に気付くことができるようになり、見積精度を高めることができる。

#### Abstract

In recent years, There are a lot of derivation development projects that a development cost exceeded an estimate cost, and that become one of business risk. We investigated some deficient cases that had cost beyond the estimate and found out there were many cases that malfunction had occurred in test process and to take measures we had had to go back to design process. The cause of that is existence of 'informal Requirements or Specification' due to lack of consideration in estimate process. It is important to classify whether a change request from clients in the derivative development is 'Requirements' or 'Specification', and in the latter case, it is also important in estimate process to consider whether 'informal Requirements or Specification' exist, but we could say that these classification have been depended on persons in charge of estimate and it is difficult. In this study, We devised 'CRCG(Change Request Classification Guide)' for a change request to distinguish 'Requirements' or 'Specification' as approach to notice 'informal Requirements or Specification'. We get possible to notice 'informal Requirements or Specification' by using this guide and can improve with estimate precision.

## 1. 研究動機

今日のソフトウェア開発は低コスト、短納期が当たり前となっているのが現状である。その要因として、小規模な派生開発が多いことが理由の一つに挙げられる。小規模な開発だけに、高精度の見積を短期間に行う必要があるのだが、必ずしも当該システムを熟知している経験者（以下、熟練者と称する）が見積を行うとは限らず、熟知していない開発担当者が見積ることも少なくない。この場合、熟練者とは違って見積りに時間が掛かるため、変更依頼に対する変更箇所および影響箇所の見極めが十分に出来ないまま、開発を進めざるを得ない状況があるのが実情である。また、我々研究メンバーの所属するプロジェクトを対象に開発工数の予実を分析したところ、結合テストまたは総合テスト工程において予実の工数乖離が大きいことが分かった。その原因が何処にあるのかを見つけるため、結合テストまたは総合テスト工程で発生した不具合事例に着目して分析した結果、これらの不具合は要求分析や設計工程において、変更箇所や影響範囲の調査・検討不足および仕様漏れが起因するものであり、その対策に多くの手戻りコストが発生していることが分かった。

分析した不具合事例の中には、変更依頼に示された内容を基にプログラムを変更したが、後の結合テストまたは総合テストにおいて、変更した箇所に起因した不具合が他モジュールおよび他システムで発見されたケースがあった。このケースにおける問題点は、顧客からの変更依頼が具体的な内容で示されていたことより、変更箇所が容易に特定できたために、担当者は他にも「変更すべき箇所がある」ことを疑わなかったことで「隠れた仕様」に気付かなかったことにある。

この「隠れた仕様」を見つけ出す手掛かりを得るために先行技術である XDDP[1]を参照した。XDDP によると顧客からのシステムに対する変更依頼には、要求と仕様が混在[2]しているという。変更依頼が「要求」で来た場合、USDM[3]を用いて要求から仕様を導き出すことは可能であるが、「仕様」で来た場合は、その変更依頼に対する背景や理由が書かれていないため、顧客へ変更内容に関する背景や理由を確認した上で、新たに要求を立てる必要がある（図 1.1）。

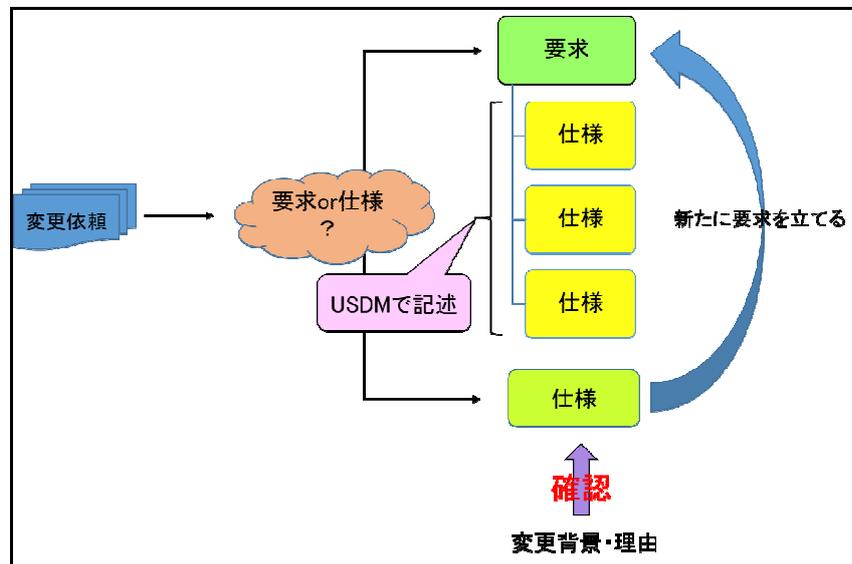


図 1.1 変更依頼からの要求・仕様分析イメージ

しかしながら、顧客からの変更依頼が「仕様」で来たものであると、担当者が見極められなければ、“新たに要求を立てる作業”が出来ず、隠れた「要求」もしくは「仕様」に気付く事は困難である。

そこで我々は、顧客からの変更依頼が「要求」または「仕様」であることを見極めるための術を示すことを研究の目標とした。

## 2. 現状分析

工数予実に乖離が発生する原因を分析するため、研究メンバーの所属するプロジェクトを対象に、プロジェクトの工数予実調査を実施した。その結果、収集したプロジェクト 15 事例のうち、工数予実の乖離が 5 件のプロジェクトに見られ、さらに、結合テストもしくは総合テストに顕著に表れていることが分かった（付表 A-1）。

また、上記の調査対象プロジェクトにて発生した 91 件の不具合事例を分析したところ、変更内容に対する処理の検討不足・考慮漏れが 35 件と不具合の約 40%を占めていること（付図 B-1 参照）、また、不具合の約 40%が詳細設計まで手戻りが発生していること（付図 B-2 参照）が分かった。

上記の調査結果より、設計段階で変更内容に対する処理の検討不足・考慮漏れが発生したことに起因した「工程の手戻り」が、工数予実の乖離に大きく影響していることが分かった。何故、設計段階で変更内容に対する処理の検討不足・考慮漏れが発生したのか？その手掛かりを得るために先行技術である XDDP を参照した。

XDDP では、機能変更の開発プロセスにおいて使用する 3 点セット（変更要求仕様書、トレーザビリティ・マトリクス（以下、「TM」と記述する）、変更設計書）の一部である変更要求仕様書は、顧客からのシステムに対する変更依頼を「USDM」の表記法に則り、要求と仕様を階層構造で表現したものである。しかしながら、USDM は「要求」と「何故その変更が必要なのか？」の理由や背景を基に「変更すべき箇所」を捉えるものであるが、「仕様」で来る変更依頼には、その理由や背景が示されていない事が多く、要求の本質を捉えにくい。このため、顧客へ変更内容に関する背景や理由を確認した上で、“新たに要求を立てる作業”が必要となるが、担当者が「仕様」で来る変更依頼を見極められなければ「変更すべき箇所」に気付く事は困難である。

また、見積の経験不足やシステムの習熟度が低い担当者が見積を行った際、変更依頼が具体的な内容で示されているが故に、変更箇所を容易に特定できたことで、他への影響等について何ら疑いを持たなかった結果、他の「変更すべき箇所」の存在に気付かなかった。この点については、派生開発特有の問題であると考えられる。派生開発では、既存ソフトウェアに対して新たな機能を追加する「新規機能追加」と、既存機能に対して変更を行う「既存機能変更」の異なる 2 つの開発アプローチが存在する（図 2.1）。

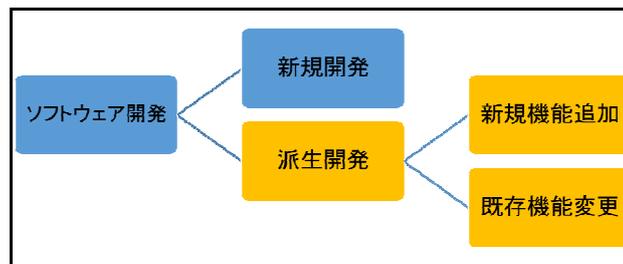


図 2.1 ソフトウェア開発形態

「新規機能追加」に関しては新規開発同様、システム全体や追加機能に関して追加する背景や理由が変更依頼に伴って示される場合が殆どであるが、「既存機能変更」に限っては、変更依頼が具体的な内容で示されるケースが少なくない。それ故に、変更依頼から容易に変更箇所が特定できたことで「他に変更箇所は無いだろう」という、担当者の「思い込み」が「隠れた仕様」を気付きにくくしている。

上記の問題を解決すべき課題として下記の点にフォーカスした。

- ・如何にして変更依頼から「要求」で来たものか「仕様」で来たものかを見極めるか。

変更依頼が「仕様」で来たものと見極められることで、“新たに要求を立てる作業”が必要であることに気付くようになり、他にも「変更すべき箇所」があるかも知れないと担当者が気付きやすくなる。その結果、工程の手戻りが低減され、見積精度も向上すると考える。

### 3. 先行研究

仕様から新たな要求を立てる方法については、先行研究として木下ら[4]が考案した「システムプロファイル」がある。これは、変更要求仕様書を利用して仕様から新たな要求を立てる際に、他システムへの変更要求を引き出すために用いるシートであるが、そもそも変更依頼の「仕様」（先行研究では「仕様レベル」と称する）について、「変更箇所がイメージできるほど具体的に示されたもの」としており、「仕様レベル」に関する具体的な定義付けがされていない（論じられていない）。特に見積の経験不足やシステムの習熟度が低い担当者が見積を行う場合、変更依頼が「仕様」であると判断できなければ、「システムプロファイル」を用いても、他への影響箇所が適切に抽出できない可能性がある。

### 4. 解決策

#### 4.1 見積を見誤る理由

派生開発の見積において工数を見誤る原因の一つに、変更依頼について、「要求」か「仕様」の見極めができず、「隠れた要求／仕様」に気付かないことにある。その理由は、見積担当者の経験不足やシステムの習熟度が低く、変更依頼が具体的な内容で示されているが故に変更内容を鵜呑みにしてしまった結果、「隠れた要求／仕様」の存在に気付くことができなかつたからである。

#### 4.2 解決策のポイント

上述の理由から、「隠れた要求／仕様」を見つけるためには、見積担当者の経験不足や当該システムの習熟度に極力依存させないことが必要である。そのため、変更依頼には「要求」として来るものと「仕様」として来るものの2種類が混在することを認識させた上で、変更依頼が「要求」または「仕様」のいずれかで来たものであるかを的確に判断させることがポイントとなる。なぜなら「仕様」と判断された変更依頼が、変更内容に関する背景や理由を考慮した上で“新たに要求を立てる作業”（以下、昇華と称する）の前提条件になるからである（図 4.1）。

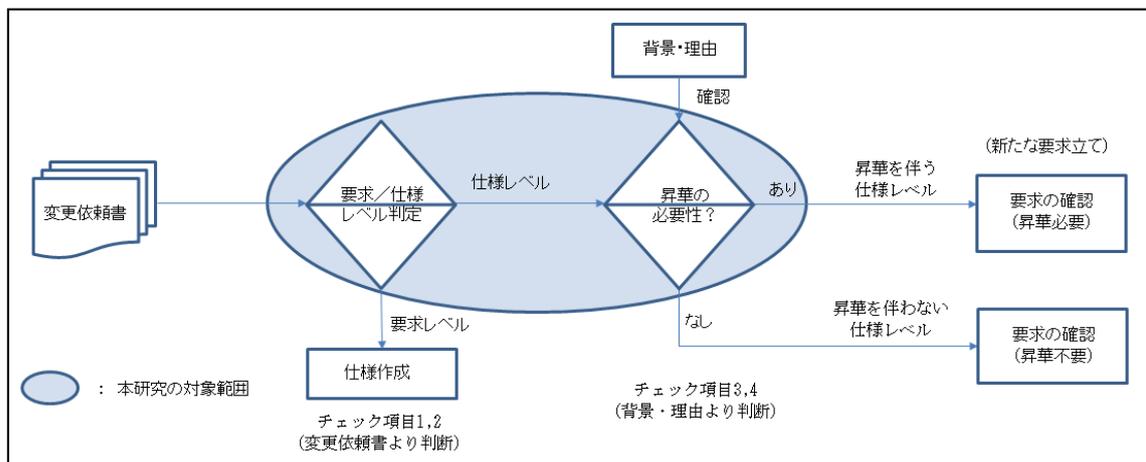


図 4.1 変更依頼から仕様作成までの流れ

#### 4.3 解決策

見積における顧客からの変更依頼の捉え方について習熟者にヒアリングし、我々研究メンバーで上述の解決策のポイントを基に、変更依頼から「要求レベル」または「仕様レベル」を判定する「変更依頼仕分けガイド (Change Request Classification Guide)」（以下、CRCGと称する）を考案した。このCRCGを使い、変更依頼の内容を分析することにより、「仕様」で来た変更依頼に気づき、「隠れた要求／仕様」の発見につながるものと考えた。

表 4.1 変更依頼仕分けガイド (CRCG)

No	チェック項目	必要性	実例
1	変更依頼から変更箇所が特定できるか。	「仕様レベル」で来た変更依頼と特定するためには、プログラムソースより「変更すべき箇所」を特定できる具体的な内容でなければなりません。変更箇所が特定できないことは、未確定項目があり、その項目から新たな「要求レベル」で来た変更依頼の項目が発生する可能性がある。	【チェック項目1を満たしていない例】 精算システムで取り扱う金額項目について内税から、外税に変更する。 (修正項目があいまいで、修正箇所の特定できず)  【チェック項目1を満たしている例】 2014年4月の消費税の変更に伴い、精算システムの精算額照会画面の「税率5%」の表示を「税率8%」に変更する。 (変更箇所が精算額照会画面と特定できており、キーワード検索を実施するまでもなく修正可能)
2	変更依頼から変更箇所を特定するために必要な「キーワード」を洗い出すことができるか。	変更箇所の特定には、プログラムソースや設計書を基にキーワードによる検索が必要である。変更依頼からそのキーワードが特定できなければ、修正箇所の漏れにつながり、見積精度を低下させる要因となる。	【チェック項目2を満たしていない例】 精算システムで取り扱う金額項目について内税から、外税に変更する。 (修正項目があいまいで、キーワードが洗い出せない)  【チェック項目2を満たしている例】 2014年4月の消費税の変更に伴い、消費税率が明確になるよう精算システムの精算金額を内税から外税に変更し消費税額の項目を保有する。 あわせて5%の消費税率を変更できるように消費税率(%)の項目も保有する。 (上記キーワードにより、プログラムソースから検索可能)
3	変更における背景・理由が明確になっているか。	変更箇所の特定が明らかでも仕様漏れが起きる場合がある。その1つに変更する背景・理由が明確に示されていないケースである。変更依頼の中に変更における背景・理由が示されていないければ、本来の要求に気付くことができず、「変更すべき箇所」が漏れてしまう。	【チェック項目3を満たしていない例】 精算システムの精算金額を内税から、外税に変更する。 (背景/理由が不明)  【チェック項目3を満たしている例】 2014年4月の消費税の変更に伴い、消費税率が明確になるよう精算システムの精算金額を内税から外税に変更し消費税額を保有する。 あわせて5%の消費税率を変更できるように消費税率(%)の項目も保有する。 (2014年4月の消費税の変更に伴い、消費税率が明確にする旨の背景・理由が明確になっている)
4	変更依頼より特定できた変更箇所が変更する背景・理由を満たしているか。	変更依頼から特定できた変更箇所が変更する背景・理由を満たしている(網羅されている)か確認する必要がある。変更箇所が特定できたとしても、それだけで顧客の実現したいことが網羅されていないことも懸念され、「隠れた仕様」につながる要因となる。	【チェック項目4を満たしていない例】 チェック項目1から項目3までに特定できた変更箇所について、プログラムを変更するだけでは変更する背景・理由を網羅できていない。  【チェック項目4を満たしている例】 チェック項目1から項目3までに特定できた変更箇所について、プログラムを変更することで変更する背景・理由を網羅できている。

4.3.1 判定結果による定義付け

図 4.2 に示す要求/仕様レベル判定フローに従い、CRCG (表 4.1) を用いた変更依頼の要求/仕様レベル判定結果を以下に示す。

【昇華を伴わない仕様レベル】

チェック項目 1 又は項目 2 に該当かつ、項目 3 および項目 4 に該当する場合。

【昇華を伴う仕様レベル】

チェック項目 1 又は項目 2 に該当かつ、項目 3 または項目 4 に非該当の場合。

【要求レベル】

チェック項目 1 および項目 2 に非該当の場合。

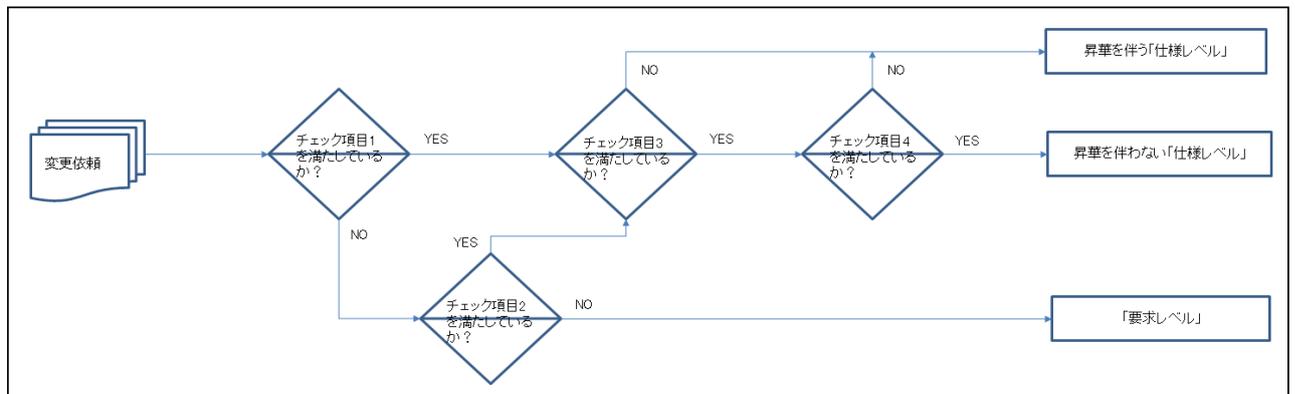


図 4.2 CRCG を用いた要求/仕様レベル判定フロー

## 5. 解決策の検証

既存のプロジェクト事例を基に、以下の観点から表 4.1 に示す CRCG の正当性を検証する。

- ① 変更依頼から「仕様レベル」または「要求レベル」を判定できるか。
- ② 「昇華を伴う仕様レベル」と判定できた変更依頼から昇華を行い、隠れた変更仕様を見つけることができるか。

### 5.1 「仕様レベル」判定の検証

#### (1) シミュレーションの実施

研究メンバーから収集した不具合事例 70 件(単純ミスによる不具合 21 件については、本研究に直接関係しないため対象外とした)を基に特定できた、変更依頼 59 件を用いて CRCG の正当性を検証した。

#### (2) シミュレーションの結果

シミュレーションを実施した結果を表 5.1 に示す。検証の対象となった変更依頼 59 件のうち、17 件について「昇華を伴わない仕様レベル」、3 件について「昇華を伴う仕様レベル」と判定できた。

表 5.1 CRCG 判定結果 (一部抜粋)

#	変更依頼	判定結果
1	データ削除はWeb画面上で実施すること 削除時は、削除機能のボタンをクリックし、表示される日付設定画面で、日付を指定して実行ボタンをクリックすると指定日以前のデータもしくは指定日～指定日を選択しデータ削除を行います。  別に変更したい背景記載あり	要求レベル
2	以下のチェック処理を行うこと ・2種の線が重複する区間があること ・オーバーラップの並行区間が存在すること。	昇華を伴わない仕様レベル
3	(1)端末からサーバへのデータAのアップロードはFTPで行い、区毎の分割はサーバ側で行うこととします。 (2)データBをFTPアップロードを行い、区毎の分割はサーバ側で行うこととします。  別に変更したい背景記載あり	要求レベル
4	XXXデータはN年で自動削除	昇華を伴う仕様レベル
5	スケールを下記のとおり変更する。 25・50・100 → 25・50・100・150・200・250	昇華を伴わない仕様レベル
6	新バージョンではAPサーバを2台稼働し、ユーザからのアクセス負荷を分散させたい。	要求レベル
7	接続先装置から最終データを受信したら、接続先装置からの△△確認要求を待ち、△△確認要求を受信したら△△確認応答を送信し、その後○○配信処理を行うこと	昇華を伴う仕様レベル
8	○○の状態を表示するデバッグコマンドで○○数の表示を追加すること Aと同様の表示をBでも行うこと	昇華を伴わない仕様レベル
9	XXXデータ編集画面で編集可能な径間長の有効範囲をX～Yとする。 【理由】 径間長の有効長が範囲外の場合はエラーとし、システムで取り扱わないようにしたい。  【前回リリース以前】 径間長における有効範囲の指定なし。	昇華を伴わない仕様レベル

### 5.2 判別結果の有効性検証

#### (1) シミュレーションの実施

CRCG による判別結果の有効性を検証するため、「昇華を伴う仕様レベル」と判定できた 3 件について昇華を行い、隠れた変更仕様を見つけることができるかを検証した。

#### (2) シミュレーションの結果

シミュレーションを実施した結果、「隠れた要求/仕様」を見つけることができた。

ここでは表 5.1 項番 7 の変更依頼を用いて、表 5.2 に判定前の変更要求仕様書例を、表 5.3 に判定後の変更要求仕様書例を示す。

表 5.2 判定前の変更要求仕様書例

大分類	小分類	変更要求・変更仕様	
機能	○○○データ配信機能		
要求	DELIVERY	○○○データ配信の開始前に、接続先装置からの△△確認要求の受信・応答処理を追加する。	
	理由	オペレータが再送が必要が否かを判断するため、末端の装置へどれくらいの割合でデータが到達したかを把握したい。	
	説明		
	<input type="checkbox"/>	DELIVERY01	接続先装置からの××最終データを受信した処理の後に、△△確認要求MSGを受信する処理を追加する。 <△△確認要求MSG受信時処理>
	<input type="checkbox"/>	DELIVERY02	△△確認要求MSGを受信した場合、送信元へ正常(データの到達率)または異常(エラーコード)を返す△△確認応答MSGを送信する処理を追加する。 <△△確認要求MSG受信タイムアウト処理>
	<input type="checkbox"/>	DELIVERY03	△△確認要求MSGがxxx時間内に受信出来なかった場合(受信タイムアウト時)、装置を初期化状態に戻す処理(開数)をコールする処理を追加する。

表 5.3 「DELIVERY」は、昇華によって新たに立てた「変更要求」であり、その変更要求から USDM によって見つけた「隠れた仕様」が「DELIVERY02.01」である。

表 5.3 判定後の変更要求仕様書例

大分類	小分類	変更要求・変更仕様	
機能	○○○データ配信機能		
要求	DELIVERY	○○○データ配信処理開始前に、接続先装置に対してデータ到達率を送信する処理を追加する。但し、一定時間経過しても○○○データ配信が完了しなかった場合は、装置を初期化状態に戻す。	
	理由	オペレータが再送が必要が否かを判断するため、末端の装置へどれくらいの割合でデータが到達したかを把握したい。	
	説明	データ到達を最優先として△△確認を追加するが、従来通り古いデータの○○○配信を行わないため、一定時間経過しても、下記①、②のいずれかに該当する場合は、装置を初期化状態に戻す。 ①△△確認要求MSGが一定時間内に受信出来なかった(データ受信タイムアウト) ②○○○データ配信シーケンスが一定時間内に完了しなかった(シーケンスタイムアウト)	
要求	<input type="checkbox"/>	DELIVERY01	○○○データ配信処理を開始する前に、接続先装置からの△△確認要求MSGを受信する処理を追加する。 理由 △△確認応答を送信するため 説明
	<input type="checkbox"/>	DELIVERY01.01	接続先装置からの××最終データを受信した処理の後に、△△確認要求MSGを受信する処理を追加する。 <△△確認要求MSG受信時処理>
	<input type="checkbox"/>	DELIVERY01.02	△△確認要求MSGを受信した場合、送信元へ正常(データの到達率)または異常(エラーコード)を返す△△確認応答MSGを送信する処理を追加する。 <△△確認要求MSG受信タイムアウト処理>
要求	<input type="checkbox"/>	DELIVERY01.03	△△確認要求MSGがxxx時間内に受信出来なかった場合(受信タイムアウト時)、装置を初期化状態に戻す処理(開数)をコールする処理を追加する。
	<input type="checkbox"/>	DELIVERY02	シーケンス監視タスクの監視タイムアウト時間を変更する。 理由 △△確認要求MSGの受信・応答処理の追加に伴い、シーケンス処理時間が延伸するため 説明
	<input type="checkbox"/>	DELIVERY02.01	シーケンスタイムアウト(ST)の時間をAAAからBBBBに変更する。

## 6. 結論

### 6.1 取組みと成果

今回の研究を通し、派生開発における「既存機能変更」においては、変更依頼が「仕様レベル」で来ることが多くあることを再認識することができた。また、シミュレーションによる検証結果から、研究メンバーが携わっているシステム形態に限定ではあるが、CRCG を用いることで以下に示す効果があることがわかった。

- ① CRCG を用いることで、熟練者か否かに関わらず、変更依頼が「仕様レベル」として提示され、隠れた要求や仕様があることに気が付くことができる。
- ② CRCG により「昇華を伴う仕様レベル」と判定された項目については、昇華を行うことで新たな要求を定義することができる。
- ③ 読み手にとって誤解を与えるような曖昧な表現に気付きやすくなり、変更依頼の提示元へ確認を促す機会になる。

### 6.2 今後の課題

CRCG を用いることで、変更依頼より「要求レベル」または「仕様レベル」が判定可能であることは確認できたが、この結果、見積精度がどの程度改善されるかについては、時間的な制約から検証するに至らなかった。この点に関しては今後の課題である。

また今回の研究では、「隠れた要求／仕様」に気付かなかったことにより発生した不具合事例に着目したが、他の不具合事例について、早期発見に有効であると思われる検出手段を検討した（表 6.1）。

表 6.1 不具合事例における検出手段一覧（付表 C-2 より抜粋）

No.	不具合事例	不具合の検出手段			
		レビュー	構成管理	昇華	再確認
1	不具合事例 1	○	×	×	×
2	不具合事例 2	×	○	×	×
3	不具合事例 3	×	×	○	×
4	不具合事例 4	×	×	×	○
...	...	...	...	...	...
20	不具合事例 20	○	×	×	×

その結果、昇華の他にもレビュー（設計、コード含む）や構成管理、質問票等による仕様の再確認で不具合を事前に発見できると判断した。これらの検出手段（昇華を除く）における解決策の検討についても今後の課題である。

## 7. 参考文献

- [1] 清水 吉男, 「派生開発」を成功させるプロセス改善の技術と極意, 技術評論社, 2007
- [2] 清水吉男, SQiP 第 6 部会[派生開発]特別講演資料 5 版 派生開発プロセス[XDDP]のポイント, 2013
- [3] 清水 吉男, 【改定第 2 版】 [入門+実践] 要求を仕様化する技術・表現する技術, 2010
- [4] 木下 良介, 中澤 康郎, 大杉 仁司, 変更依頼の対応箇所を検討する前に他システムへの影響を検知する方法, 日本科学技術連盟第 27 年度ソフトウェア品質管理研究会成果報告集, 2011

付録 A

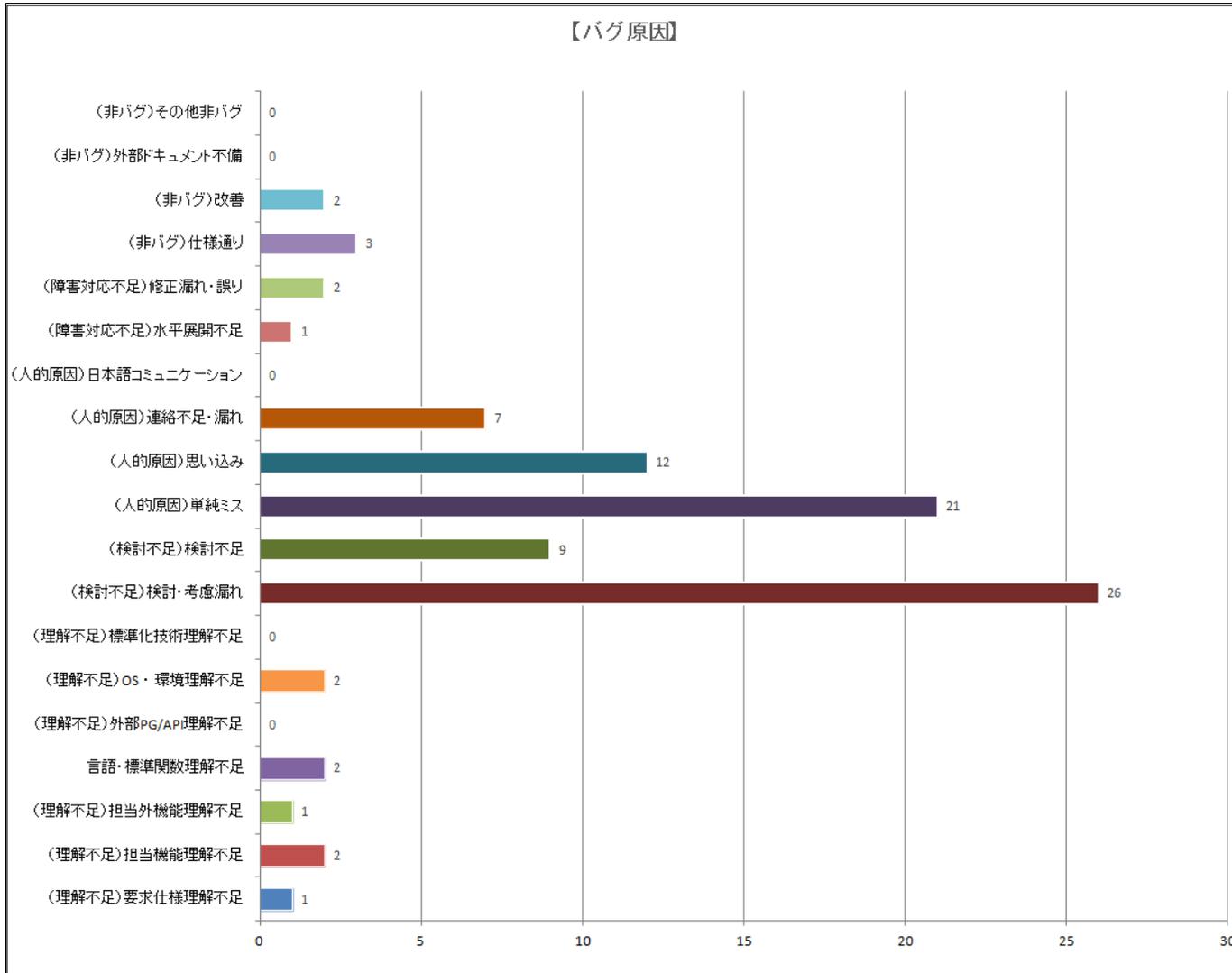
付表 A-1：プロジェクト事例

P. #	予実	作業工程									合計	予実差
		PJ管理	要求分析	基本設計	機能設計	詳細設計	コーディング	単体試験	結合試験	総合試験		
1	見積(H)	192	0	160	0	400	256	256	416	240	1,920	▲ 248
	実績(H)	192	0	128	0	384	256	256	652	300	2,168	
2	見積(H)	160	0	128	0	320	240	240	352	160	1,600	304
	実績(H)	128	0	96	0	240	184	184	336	128	1,296	
3	見積(H)	160	0	128	0	320	240	240	352	160	1,600	48
	実績(H)	160	0	128	0	288	232	232	352	160	1,552	
4	見積(H)	144	0	96	0	272	224	224	320	160	1,440	184
	実績(H)	128	0	72	0	240	216	216	272	112	1,256	
5	見積(H)	96	0	48	0	160	192	192	176	96	960	112
	実績(H)	96	0	48	0	144	160	160	160	80	848	
6	見積(H)	64	0	48	0	128	160	160	160	80	800	72
	実績(H)	64	0	40	0	112	144	144	144	80	728	
7	見積(H)	288	0	240	0	400	320	320	240	112	1,920	▲ 558
	実績(H)	304	0	242	0	404	323	323	724	158	2,478	
8	見積(H)	144	0	80	0	112	192	192	128	112	960	91
	実績(H)	144	0	77	0	104	176	176	112	80	869	
9	見積(H)	0	27	224	0	25	37	10	44	30	397	▲ 136
	実績(H)	97	27	244	0	25	60	0	50	30	533	
10	見積(H)	0	32	112	0	40	88	24	130	—	426	▲ 41
	実績(H)	38	29	129	0	14	85	10	162	—	467	
11	見積(H)	243	0	450	0	102	567	513	551	—	2,426	338
	実績(H)	139	0	328	0	79	469	559	514	—	2,088	
12	見積(H)	0	0	602	0	131	667	522	783	—	2,705	▲ 1,761
	実績(H)	0	0	968	0	210	700	763	1,825	—	4,466	
13	見積(H)	0	24	52	0	0	75	54	0	102	307	▲ 145
	実績(H)	0	28	105	0	0	88	53	0	178	452	
14	見積(H)	250	100	300	0	300	380	250	230	280	2,090	▲ 600
	実績(H)	250	100	450	0	300	600	280	250	460	2,690	
15	見積(H)	200	100	200	0	650	400	250	150	150	2,100	▲ 910
	実績(H)	250	150	320	0	430	350	300	310	900	3,010	

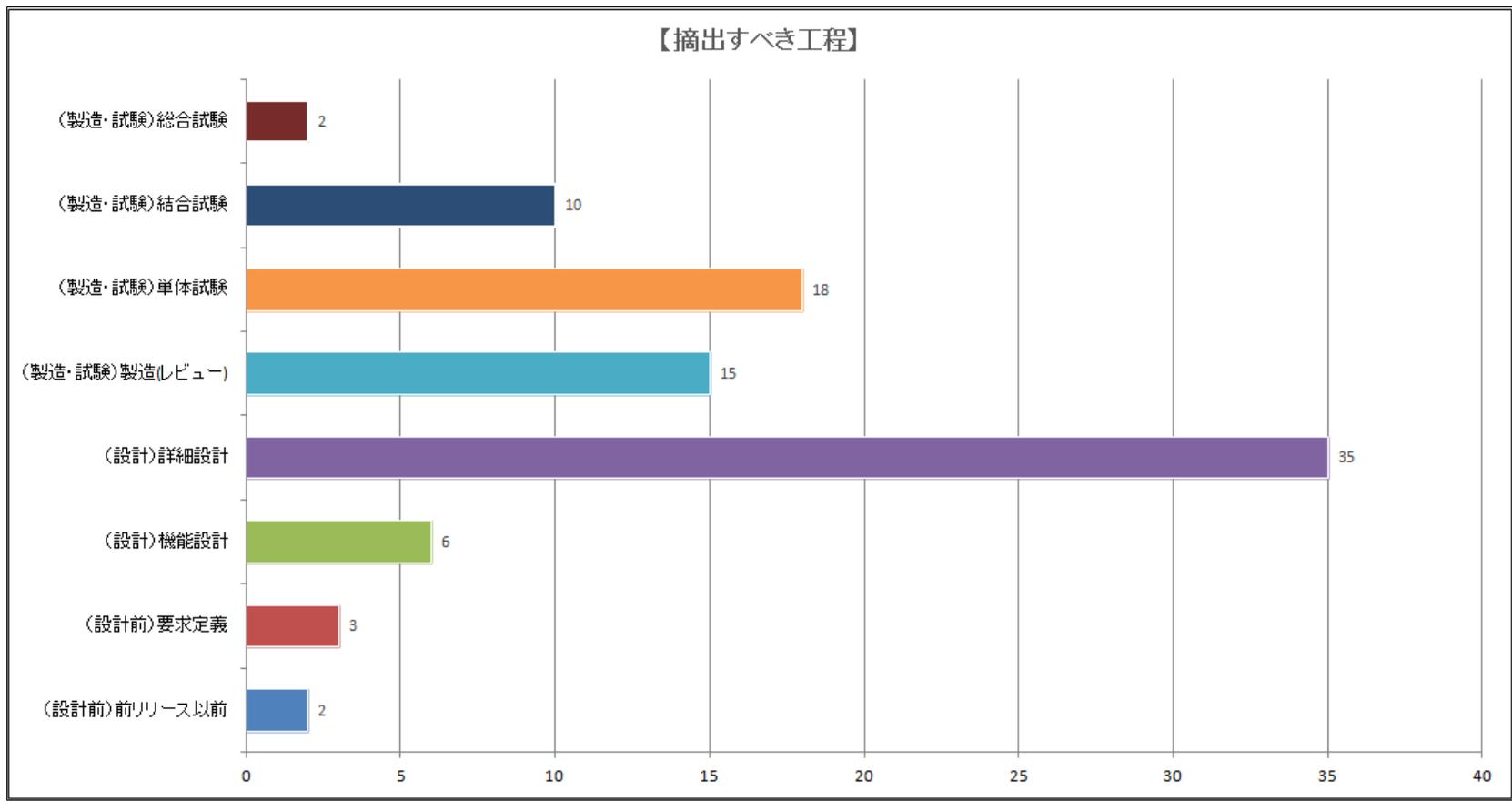
付録 B

付表 B-1 : 不具合事例 (抜粋)

No	発見工程	判定	内容	原因 (赤字: 原因となった本質)	対策	バグ分類	バグ原因	発生すべき工程
1	総合試験	P不良	データ削除画面にて、一部のデータを削除しようとするが、該当データがない場合に例外エラーが発生する。	データの参照先(関連先)にのみデータが存在しないデータを削除しようとするが発生する。 0からデータ名を取得する関数において、戻り値に空文字が返ってくるが、空文字の判別(処理)が実装されておらず例外が出力した。 一部のデータとは従来振出し新たに追加された種類のデータであるが、イレギュラーな処理に関する変更箇所抽出が漏れていた。	左記関数内の処理に空文字の判別処理を追加した。	バグ/処理論理誤り	(検討不足)検討・考慮漏れ	(製造・試験)単件試験
2	総合試験	P不良	負数を持つデータが正常に登録できない。	実数にunsignedを使用していた。 負数は無いと認識していた。(旧システムがそうだから) 新システムでも登録データに正負の両方を持つようになった。 →不具合発生後、顧客から情報開示あり。	負数が扱えるよう、実数を見なおした。	バグ/処理論理誤り	(人的原因)連絡不足・漏れ	(製造・試験)結合試験
3	総合試験	P不良	昇順に並んだデータが正常に登録できない	昇順を考慮していなかった。 降順しか無いと認識していた。(旧システムがそうだから) 新システムでも昇降順両方でデータを持つようになった。 →不具合発生後、顧客から情報開示あり。	上りの場合の処理があるか具体的に見直し	バグ/処理論理誤り	(人的原因)連絡不足・漏れ	(製造・試験)結合試験
4	総合試験	D不良	コメントの編集はメンテナンス担当者のみが行えることとあるが、Administrator権限でログインしている場合はメンテナンス担当者であってもコメント編集が可能になっている	ログインユーザがAdministrator権限、もしくはメンテナンス担当の場合にコメント編集が可能としていた。 改修前のままで変更が無いと認識していた。(改修漏れ)	ログインユーザがメンテナンス担当の場合のみ、コメント編集が可能よう修正した。	バグ/処理論理誤り	(人的原因)思い込み	(設計)詳細設計
5	総合試験	D不良	データ一覧で選択したデータの測定日が、入力欄に表示されない	実装漏れ。 改修前のままで変更が無いと認識していた。(改修漏れ)	処理を追加した。	バグ/その他、デグレード	(人的原因)思い込み	(設計)詳細設計
6	総合試験	P不良	指定の期間に該当する帳票が出力されない	日付範囲の条件式がうまく動かしなかった。 おそくRDBMSのバージョンアップによりパラメータを厳密に型指定しなければいけなくなったことよと思われる。 ※旧バージョンでは動かしなかった。	日付条件のフォーマットをyyyyMMdからyyyy-MM-ddとした。	バグ/その他、デグレード	(理解不足)OS/環境理解不足	(製造・試験)単件試験
7	総合試験	P不良	画面のタイトルが新バージョン用の名称でない旧バージョンのまま。	修正漏れ。 変更内容が容易でかつ作業優先度を低く設定していたため、忘れられたままになっていた。	画面のタイトルが新バージョンの名称に修正した。	バグ/処理論理誤り	(人的原因)単純ミス	(製造・試験)製造(レビュー)
8	総合試験	D不良	月割/計算で算出する値が日割/計算で算出された値になっている。	修正漏れ。 実数依存口誤。Q8A表に記載されていたが、詳細設計書への記載が漏れていた。	日割/計算を月割に修正した。	バグ/処理論理誤り	(人的原因)連絡不足・漏れ	(設計)詳細設計
9	総合試験	D不良	2台のAPサーバが稼働しているが、片方のAPサーバからファイル削除したが、他方のAPサーバのファイルは削除されていない。APサーバは双方で連携されており、ファイルも双方に配置されている仕様である。	処理実装漏れ。 詳細設計書の記載漏れ。	ファイル削除する際、対象ファイルを1系、2系APサーバより削除する処理を追加した。	バグ/処理論理不足	(検討不足)検討・考慮漏れ	(設計)詳細設計
10	総合試験	D不良	データ登録中のダイアログに、フォーマット変換中の旨のメッセージが表示されない。	表示処理の実装漏れ。 詳細設計書の記載漏れ。 変更依頼は口頭で機能仕様書に記載されていない。Q8A表への記載が漏れていた。	表示するよう処理を追加した。	バグ/処理論理不足	(検討不足)検討・考慮漏れ	(設計)詳細設計
11	総合試験	P不良	2種の線が重複する区間長が0mでもエラーが表示されない	区間長のチェック処理に関する実装漏れ。 機能仕様書に「2種の線が重複する区間があること」しか書いておらず、この内容から「重複する区間」=「区間長が0mでない」と読み取れなかった。 「重複する区間」=「他方の始点 ≤ X ≤ 終点」と読み取った。 X=一方の始点または終点 この場合、X=始点またはX=終点の場合、上記の条件に合致するが区間長は0mとなるケースが発生する。	区間長が0mのチェック処理を追加した。	バグ/処理論理不足	(検討不足)検討不足	(設計)詳細設計
12	総合試験	P不良	未登録メッセージが0日に登録された後、X日後にDBの未登録メッセージが削除されない。 X日:メッセージ保持期間	削除する条件がX+1日(後)になっていた。 [種別]保持年月日 < 現在日 (X+1日) [正]保持年月日 ≤ 現在日 (X+1日) メッセージ登録日は保持期間に含まれないと認識していた。(仕様解釈誤り)	保持年月日時でのチェックで削除するよう修正した。 [口]保持年月日 ≤ 現在日	バグ/処理論理誤り	(人的原因)思い込み	(設計前)リリース以前
13	総合試験	P不良	一部のファイルがサーバへ登録されない	対象のファイルを登録対象としていなかった。 システムで扱わないファイルだが、他のシステムで使用されるものだという認識があった(知らなかった)。	登録する処理を実装した。	バグ/処理論理誤り	(理解不足)担当外機能理解不足	(製造・試験)結合試験
14	総合試験	D不良	一部の機能で旧式のデータフォーマットを扱っている部分がある。	処理実装漏れ。 他機能への影響箇所抽出漏れ。	新機能で対応したデータフォーマットに処理を変更した。	バグ/詳細設計不備	(検討不足)検討不足	(設計)詳細設計
15	総合試験	D不良	データベースのデータを自動削除する機能において、一部のデータが削除されない。	処理実装漏れ。 一部のデータとは従来振出し新たに追加された種類のデータであるが、既存処理への変更内容が記載されていなかった。(記載漏れ)	処理を追加した。	バグ/詳細設計不備	(検討不足)検討・考慮漏れ	(設計)詳細設計



付図 B-1 : バグ原因



付図 B-2 : 抽出すべき工程

付録C

付表 C-1：変更依頼仕分けガイド（CRCG）

No	チェック項目	必要性	実例
1	変更依頼から、変更箇所が特定できるか。	「仕様レベル」で来た変更依頼と特定するためには、プログラムソースより「変更すべき箇所」を特定できる具体的な内容でなければならぬ。 変更箇所が特定できないということは、未確定項目があり、その項目から新たな「要求レベル」で来た変更依頼の項目が発生する可能性がある。	【チェック項目1を満たしていない例】 精算システムで取り扱う金額項目について内税から、外税に変更する。 (修正項目があいまいで、修正箇所の特定できず)
			【チェック項目1を満たしている例】 2014年4月の消費税の変更に伴い、精算システムの精算額照会画面の「税率5%」の表示を「税率8%」に変更する。 (変更箇所が精算額照会画面と特定できており、キーワード検索を実施するまでもなく修正可能)
2	変更依頼から、変更箇所を特定するために必要な「キーワード」を洗い出すことができるか。	変更箇所の特定には、プログラムソースや設計書を基に、キーワードによる検索が必要である。 変更依頼からそのキーワードが特定できなければ、修正箇所の漏れにつながり、見積精度を低下させる要因となる。	【チェック項目2を満たしていない例】 精算システムで取り扱う金額項目について内税から、外税に変更する。 (修正項目があいまいで、キーワードが洗い出せない)
			【チェック項目2を満たしている例】 2014年4月の消費税の変更に伴い、消費税率が明確になるよう精算システムの精算金額を内税から外税に変更し消費税額の項目を保有する。 あわせて5%の消費税率を変更できるように消費税率(%)の項目も保有する。 (上記キーワードにより、プログラムソースから検索可能)
3	変更における背景・理由が明確になっているか。	変更箇所の特定が明らかでも仕様漏れが起きる場合がある。 その1つに変更する背景・理由が明確に示されていないケースである。 変更依頼の中に変更における背景・理由が示されていない場合は、本来の要求に気付くことができず、「変更すべき箇所」が漏れてしまう。	【チェック項目3を満たしていない例】 精算システムの精算金額を内税から、外税に変更する。 (背景/理由が不明)
			【チェック項目3を満たしている例】 2014年4月の消費税の変更に伴い、消費税率が明確になるよう精算システムの精算金額を内税から外税に変更し消費税額を保有する。 あわせて5%の消費税率を変更できるように消費税率(%)の項目も保有する。 (2014年4月の消費税の変更に伴い、消費税率が明確にする旨の背景・理由が明確になっている)
4	変更依頼より特定できた変更箇所が、変更する背景・理由を満たしている(網羅されている)か確認する必要がある。 変更箇所が特定できたとしても、それだけで顧客の実現したいことが網羅されていないことも懸念され、「隠れた仕様」につながる要因となる。		【チェック項目4を満たしていない例】 チェック項目1から項目3までに特定できた変更箇所について、プログラムを変更するだけでは変更する背景・理由を網羅できていない。
			【チェック項目4を満たしている例】 チェック項目1から項目3までに特定できた変更箇所について、プログラムを変更することで変更する背景・理由を網羅できている。

付表 C-2：不具合事例における検出手段一覧

No	不具合事例	不具合の検出手段			
		レビュー	構成管理	昇華	再確認
1	2種の線が重複する区間長が0mでもエラーが表示されない。	○	×	×	×
2	データベースのデータを自動削除する機能において、一部のデータが削除されない。	×	○	×	×
3	ポストBの位置を変更した後、登録ボタンを押下しを実行すると、ポストAの径間長が変更された値(許容範囲外の値)でデータベースに登録されている。	×	×	○	×
4	基幹・店舗XXXX表 表示不正がある。	×	×	×	○
5	出力した帳票のコメントがコメント欄に入りきらず、コメント5行目だけ欄外に出力されている。	○	×	×	×
6	帳票ファイルが指定の保存期間(年)を過ぎても自動削除されない。	×	○	×	×
7	実データが存在しない場合に4byte(年月日)のみ送信され、受信側では受信エラーとするはずが、エラーとならない。	○	×	×	×
8	XXXXファイルの合計金額が△△金額の合計額になっている。 正しくはXX金額の合計	×	×	○	×
9	支払日を入力せずに対象債権の表示もしくはCSVボタンを押下すると、必須チェックエラーとなる。支払日を入力する前に対象債権の確認をすることで、該当ボタン押下時に必須チェックは不要。	○	×	×	×
10	支払申請入力について、新規(INSERT)と更新(UPDATE)でチェックが異なる。 新規では信託終了依頼書が出力ずみの場合エラーとしているが、更新時にはエラーとならない。	○	×	×	×
11	契約照会 契約一覧(ダウンロード)について、契約一覧(CSV)の日付項目(契約日等)の出力形式が設計書通りでない。 正:YYYYMMDD 誤:YYYY/MM/DD	○	×	×	×
12	基幹データ表示(科目)について、▲▲▲▲サブで取込内容を表示した際、対◎◎◎社向けの明細が正常に取り込まれない。	○	×	×	×
13	基幹データ表示(科目)の科目出力シートについて、約定年月日の項目に年月のみが出力される。	○	×	×	×
14	電源再投入でバックアップ対象の設定データが消えてしまった。	○	×	×	×
15	AからBにデータ配信後、Bにてコマンドを実行すると、xxx項目に「-」と表示される。本来は「」(空白表示)となるはず。	×	×	×	○
16	内部データ表示コマンド実行時に、無効を示すデータ(-----)がずれて表示される。	○	×	×	×
17	データ設定後に0初期化処理がはってしまう。	○	×	×	×
18	内部ログが同一ログで埋め尽くされてしまう。	○	×	×	×
19	データ配信の正常シーケンスにてシステムダウンする。	×	×	○	×
20	AからBに指示ファイルにて要求を行った際、Bにて入力チェックエラーとなり通信できない。	○	×	×	×