

CRMシステム開発における 結合テスト効率の改善

Improvement of Efficiency in Integration Tests for CRM Development

中島 孝人
Takato Nakashima

概要

どんな人でもこなせる作業量には限界がある。それを越えた場合、品質や納期に問題が発生するのは明白である。従ってプロジェクトが大きくなるほど作業分担が問題になってくる場合が多い。我々が担当しているCRMシステム開発は200人月を超える場合が多く、如何に作業分担して作業を進めるかが大きな課題となっている。とりわけ結合テスト工程は規模が大きかつ緊密に連携する作業が多いため作業分担の方法が効率や品質に大きく影響を与える場合が多い。我々はこの結合テスト工程に対して作業分担に着目した改善を行い一定の成果を挙げている。

1. はじめに

我々金融ソリューションユニットが近年対応しているCRMシステム開発案件は、収益や顧客満足度等の点で一定の好成果をコンスタントに得るようになってきた。CRMシステムの開発は、その業務的な特性＝事業戦略に密接に関わる戦略的システムであること、顧客毎に慣れ親しんだ「文化」が厳然として

存在すること等＝により、大規模/高機能であるにもかかわらず短納期が求められ、かつ広範囲のカスタマイズ要望が発生しがちである。我々はこれらの課題に対して長年の経験に立脚した手法をいくつか採用することにより、顧客の要望を削減すること無しに短納期及び高い開発品質を実現することができている。本稿ではそれらの手法のうち、主に結合テスト工程で採用した手法について紹介する。

表1 開発案件一覧

案件名	開始	終了	概算工数	機能数	Tbl数	採用技術
A銀行営業支援	1998年4月	1999年9月	270人月	約200	約150	センター集中DB オラクルアプリケーションサーバ採用
B銀行渉外支援	1998年9月	2000年3月	250人月	約180	約150	営業店分散DB MS-ASPサーバ採用 携帯端末連携機能有り
C銀行セールス支援	2000年4月	2001年6月	250人月	約160	約80	センター集中DB IBMアプリケーションサーバ採用 フレームワーク採用
D銀行営業支援	2001年7月	2002年9月	200人月	約170	約90	同上
E銀行営業支援	2002年4月	2003年5月(予定)	230人月	約160	約120	同上
F銀行営業支援	2002年6月	2003年7月(予定)	230人月	約160	約140	同上

2. CRMシステム開発の特徴と課題

2.1 CRMシステムの特徴

金融機関向け営業支援システムをベースとしている関係上多くの機能が有機的に連携している。また、これらの機能とそこで取り扱われる情報はCRMが指向する「顧客とのあらゆる接点においてシームレスなサービスを提供できるよう情報共有を行い各接点で活用を行う」ように考慮されている。このため他のシステムと比較して「顧客と様々な接点に携わる多様な人々が利用するシステム」であるがゆえに以下のような特徴を保有している。

(1) 機能が多い

それぞれの用途にあわせた機能が必要となり、数段多くの機能を有する。しかしながら、システムリテラシの観点から、顧客が特に望む一部の機能を除いて比較的単純な処理形態（単純な明細/一覧表示や明細追加/更新等）

をとる場合が多い。

(2) 権限が複雑である

セキュリティや役割の観点から同じ機能においても照会範囲/順番や更新/削除権限等複数の利用権限が存在する。

(3) 業務フローが複雑

利用者に役割分担がなされている場合は、業務フローは複数の利用者を介して形成される場合が多く、それを支援するCRMシステム自体も機能連携や情報反映の考慮が必要となる。

2.2 CRMシステム開発の課題

CRMシステムの特徴は開発においては「多機能かつ権限と機能連携が複雑なシステムを如何に効率よく開発するか」という課題になる。当初我々は設計要員を増員することでこの課題を克服できるかと考えたが、実施した結果以下の根本課題に直面した。

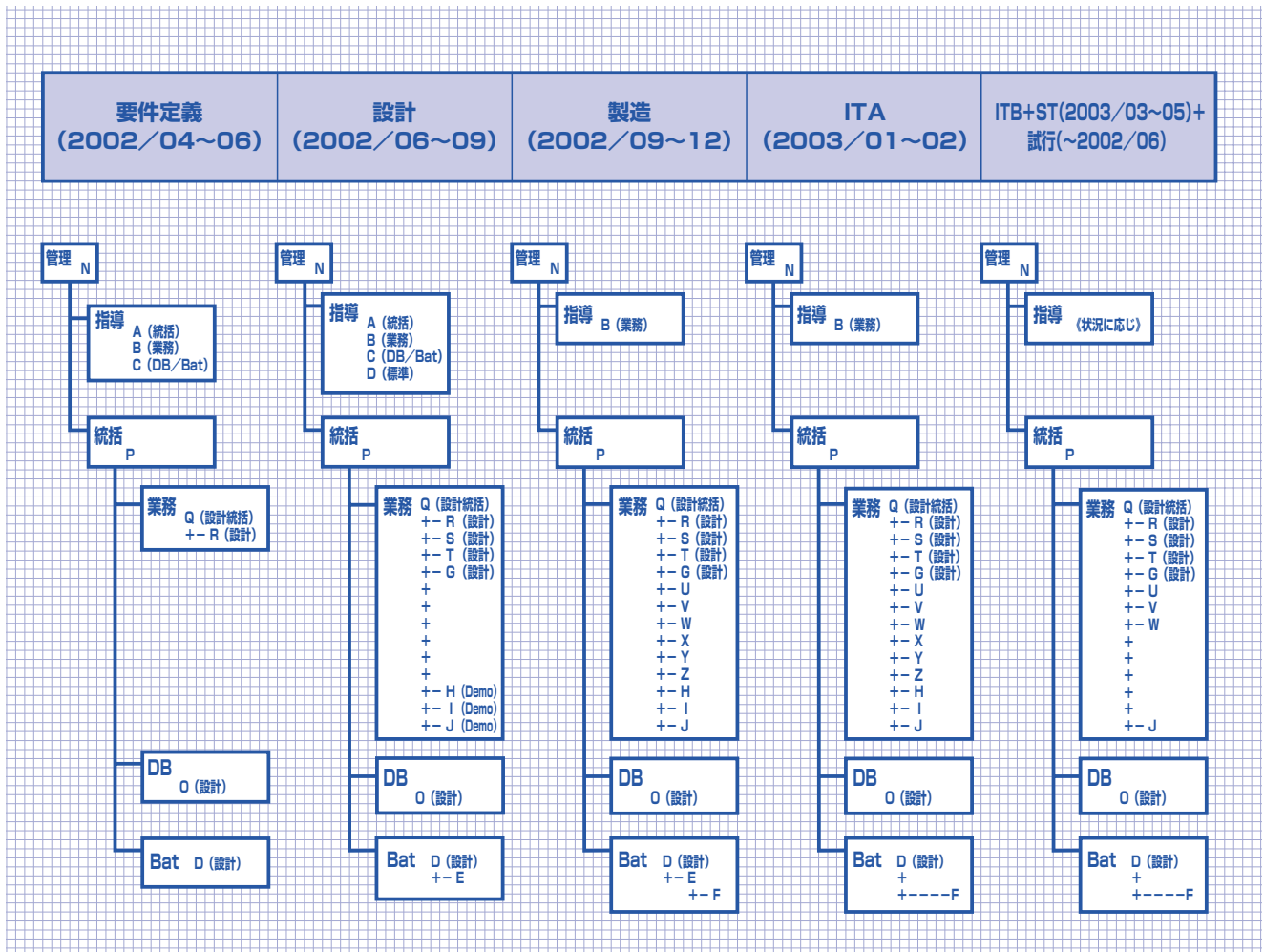


図1 工程別作業体制図 (例)

(1) 結合テストケースが作成できない

システム全体を複数のパートに分割して設計者を割り振り設計することは可能であったが、結合テスト工程において各設計者のパートを横断する結合テストケースを記述することがそれぞれの担当者ではできない。実際の現場においては設計者を取りまとめる統括設計者が存在したが、ケース作成をすべて統括設計者に担当させようとする場合その処理能力を超えることは明白でありこの観点も課題となった。

(2) 結合テストがこなしきれない

我々が対応した開発では結合テスト工程においてテストケースが1万数千ケース存在する場合が一般的であり、従来の「結合テスト工程は設計担当が中心に対処する」という方法ではとても消化しきれないボリュームである。よって我々は、結合テスト方式について抜本的な対応を行う必要があると判断し、以下で述べるようなテスト

工程と方式を編出して適用した。

3. 結合テスト作業の分類

結合テストの目的は機能全体の整合性や統一感を確認/調整する作業であるが、全てのテスト項目が関連性を持っているわけでもないので、作業方式にあわせて分担するほうが効率的である。検討の結果、前述した課題に対応するため新テスト方式で対応できる部分とできない部分を以下に記述する3種類に分類し並行作業を行うのが効率上最適であると判断した。

(1) 種類1（業務機能確認作業：新テスト方式採用部分）

オンライン機能の機能間整合性を主に確認する作業。テストシナリオやテストケース数が膨大であり前述した課題が存在するため抜本的な改善を必要とする部分。テストシナリオ上、バッチ処理と依存関係がある場合はその部分のみバッチのテストとスケジュール調整を行う。

作業工程名	担当	9月	10月	11月	12月	1月	2月	3月
設計	設計担当	■						
製造・ユニットテスト								
製造・ユニットテスト	プログラマ		■	■	■			
テスト結果レビュー	パート別設計担当		■	■	■			
結合テスト								
業務機能								
テスト計画	設計統括者			■				
ケース設定	パート別設計担当			■	■			
テストデータ作成	DB設計担当			■	■			
テスト実施【プレ】	プログラマ				■	■		
テスト実施【本番】	プログラマ					■	■	
レビュー	パート別設計担当						■	■
カテゴリ別								
ケース設定	設計統括者			■	■			
テスト実施	上級プログラマ				■	■		
レビュー	設計統括者						■	■
バッチ								
ケース設定	バッチ設計担当			■	■			
テスト実施	バッチプログラマ				■	■		
レビュー	バッチ設計担当						■	■
他システムとの結合テスト								
プレ連携テスト	バッチ担当者						■	
他システム結合【本番】	該当者							■

図2 テスト工程表例（結合テスト近辺抜粋）

(2) 種類2 (カテゴリ別確認作業)

一覧表示や更新処理等、表示やプログラム記述において同種の処理間で統一感が取れているか確認する作業。表現やプログラム記述および臨界部分での振舞い等が対象となり、テストを実施する側にもプログラム・設計両面において高度なスキル（十分な経験及びセンス）が要求される作業である。設計統括者がテストケースを作成し、前述したスキルを有するプログラマーがメンバーより選出されテストを実施する。

(3) 種類3 (バッチ側確認作業)

ジョブネットを想定した並行稼動ジョブの処理衝突の有無や障害発生時の対応の確認を行う作業。CRMシステムにおいてバッチはオンラインが稼動していない夜間に他システムとデータ連携を行うのが主な機能であり、前述した「業務機能間確認作業」に織り込んで実施するバッチテスト項目は少ない。バッチ設計者がケース設定を行いバッチプログラマーがテストとして参加する。

設計担当者がケース設定として機能毎の結合テストケースを記述する。ここで記述されるテストケースには操作権限毎にどのテストケースが確認対象かを注記する。

(3) 工程3 (結合テストデータ作成)

テスト実施者間でテストが干渉しあわないようにDB設計担当者がシナリオ進行表に基づき結合テストに必要な基礎テストデータを工程2のテストケース作成と並行して作成した。

尚、以下の工程4で整合性確認を実施し並行作業における品質上の問題点を解消している。

(4) 工程4 (結合テスト実施【プレ】)

テスト実施者の習熟度向上とテスト設計/ケース設定/テストデータ間の整合性確認を目的にテスト結果記録をとらずに結合テストを実施する。尚、この作業実施前に実施品質向上を目的にテスト実施者に対してシステム機能概要説明会を実施する。

(5) 工程5 (結合テスト実施【本番】)

テスト実施者がテスト進行表とケース設定に基づいてテストを実施する。テスト結果はテストした機能毎にテストした利用権限がわかるようなテスト場面と利用者を表す番号を付与して保管する。

(6) 工程6 (結合テスト結果レビュー)

設計担当者は担当パートの機能において全利用権限のテスト結果が出来上がった順にそのテスト結果を纏めてレビューする。

4. 採用した新テスト方式 (分業型 結合テスト)の概要

4.1 分業型 結合テスト作業の流れ

我々が新たに採用した結合テスト方式はテスト設計とケース設定およびテスト実施とレビューの作業を意図的に分担する「分業型 結合テスト方式」である。要点としては「テストシナリオとテストケースを分離した」こと、「テスト進行表を作成した」こと、「実際のテスト実施については、比較的低い理解レベルの要員でも対応可能な工程として分離独立させた」点である。この「分業型 結合テスト」の参加者は、設計統括者および設計担当とテスト実施担当兼バグ対応者の役割を持ったメンバーである。以下では「分業型 結合テスト」作業がどのような工程でどんな役割を持ったメンバーがどのような作業を行うのかを順を追って説明する。

(1) 工程1 (結合テストシナリオとテスト進行表の作成)

設計統括者がテスト設計として利用者横断の業務フローより結合テストシナリオを作成する。その後、そのシナリオの流れを縦軸とし、シナリオに関与するテスト実施者を横軸に2次元配置したテスト進行表に利用者・場面毎の機能操作有無を記入する。

(2) 工程2 (機能別 結合テストケースの作成)

4.2 実施による効果

4.1で述べた工程を採用することによって、テスト設計/ケース設定/テスト実施(レビュー内容採取)/レビュー実施の分業化が可能となった。特に、テスト設計とケース設定およびテスト実施とレビュー実施の分業化による作業平準化の効果が大きく、当初の懸案であった「結合テストケースが作りきれない」「結合テストがこなしきれない」という課題は、この工程を採用することにより解消することができた。またこの工程を採用することによって副次的な効果が数件確認されたので以下に例示する。

(1) 副次効果1 (専門化による生産性・品質低下抑制)

分業化により参加者の専門化を推進することができた。これによって参加者は担当の作業に専念することが可能となり、品質と生産性を安定させることができるように

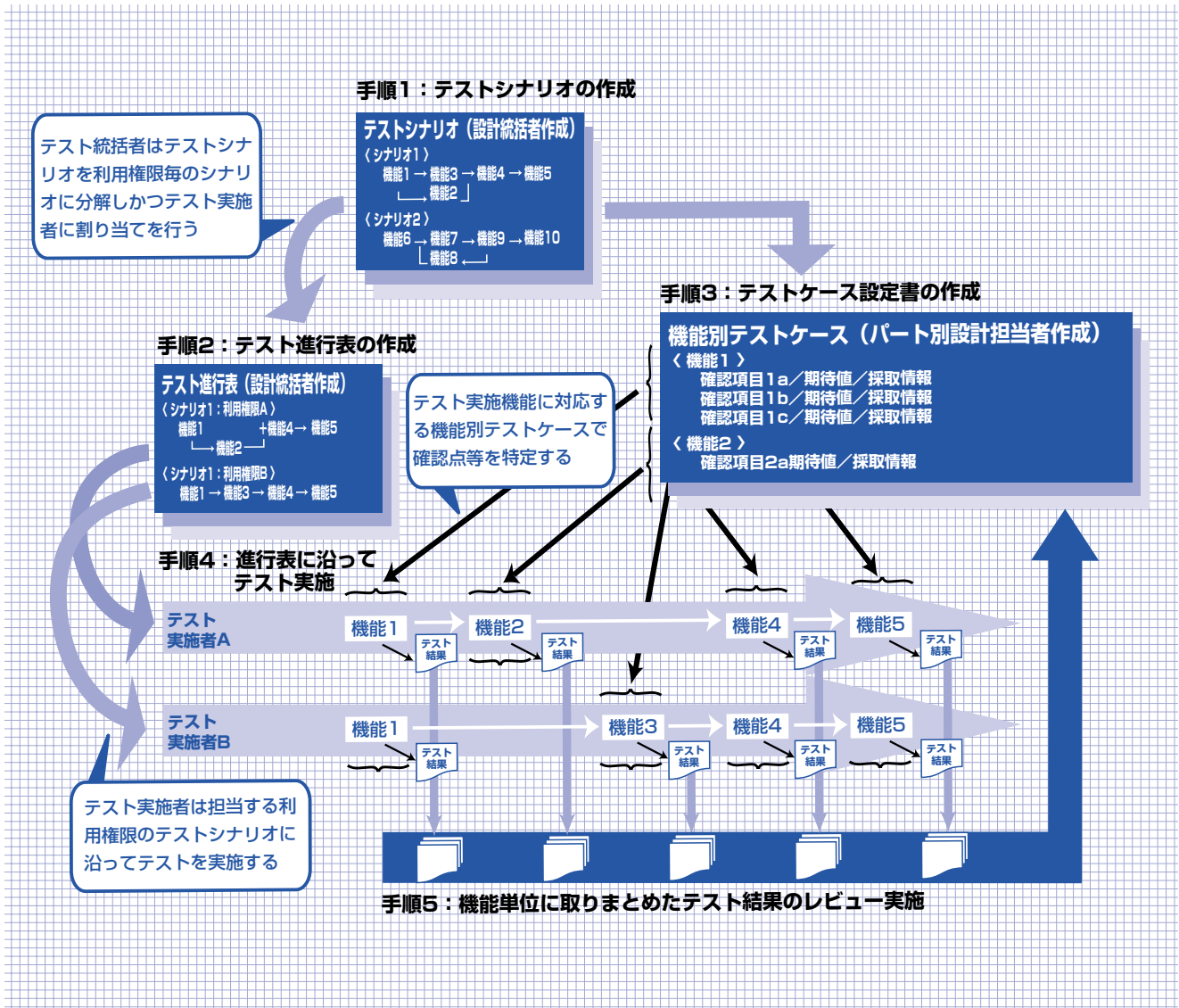


図3 分業型テストの進め方

なった。

- (2) 副次効果2（結合テスト期間が短縮化され運営リスクが軽減した）

分業化によりボトルネックが改善され結果的にテスト期間の短縮が可能となった。これによって結合テスト工程の前後の工程となりうる製造・UTや他システム連携テスト等との工程重複による運営リスクを極力回避することができるようになった。

- (3) 副次効果3（効率のより体制作りが可能となった）

プログラマでも実施可能なテスト実施工程を分離でき、この工程に製造工程で採用したプログラマをそのまま転用することができた。これによって要員リスクや運営コストをおさえつつ修正体力をもった結合テスト体制が構

築できた。

5. おわりに

5.1 現状の留意点

新テスト方式は、従来に比べて比較的大人数でも効率的な分業化がなされ当初の課題を克服したテスト方式であるが、実施の結果以下の留意点も発生してきている。これらの留意点については都度対策を実施しており 効果の確認が出来次第 別途報告することにした。

- (1) 留意点1（複雑なテスト手順を如何にテスト実施者に理解させるか）

テスト実施者は基本的にテスト進行表とテストケースに基づいたテストを実施することになっているが、一部、

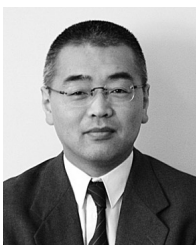
機能の複雑な部分については簡単な表現では記述が困難な場合がある。プレテスト実施等でテスト実施者にイメージアップを推進しているが、シナリオやケースの記述においても改良を加えていく必要がある。

(2) 留意点2 (テスト実施者間のテスト品質を如何に均一化させるか)

我々が採用した新テスト方式は多くのテスト実施者を参加させる方式を採用している。この方式は本質的にテスト実施者間の作業品質が課題となる。我々もこの課題に対して教育実施や日々の作業状況ヒアリングとアドバイスの励行によって品質維持に努めているが、ケース記述や実施方式の標準化も推進させていく必要がある。

5.2 最後に

以上、我々がCRMシステム開発において採用し効果を挙げたテスト方式について成果報告を行った。同様の規模や機能構成で高い開発パフォーマンスが期待されるプロジェクトにおいては、今回報告した結合テスト工程の作業パフォーマンスはいずれも大きな課題であると思われる。我々が採用したテスト方式や改善アプローチが何らかの参考になれば幸いである。



中島 孝人
Takato Nakashima

eサービス事業本部・CRMソリューション部
CRMシステム開発のプロジェクト・マネージメントに従事