
大規模基幹システム開発の“リスク”をどう乗り越えたか

～Topjax（フレームワーク）/Oh-Pa 1/3(オンメモリ DB) の圧倒的な有効性 ～

(株) 富士通ビー・エス・シー

■ 執筆者Profile ■



阿部 克利

1991年 KDDI 株式会社 入社
法人データ系サービスのシステム開発担当
2010年 現在 情報システム本部
ソリューションシステム部
開発3グループ
課長補佐



山口 均

1988年 (株) 富士通ビー・エス・シー 入社
システム業務担当
2010年 現在 第一システム本部
第一システム事業部
第一システム部 所属
担当課長 (PRESLEY 業務担当)



濱本 敬樹

1986年 (株) 富士通ビー・エス・シー 入社
システム業務担当
2010年 現在 Topjax 推進室 所属
担当課長

■ 論文要旨 ■

KDDI 様新法人データ系システムは、2008 年 4 月から要件定義を開始したが、要件定義と設計工程で発生した遅れから当初のサービスインに間に合わない大幅な遅延が発生した。

この問題を打破するためには、製造工程においてお客様を説得できる技術力が必要となり、開発手法として当社の得意技術である「Topjax Solution」を提案した。

結果として「Topjax Solution」を導入したことで、最低限度の遅延で無事サービスインすることができた。また、その後のフェーズにおいてはすべて予定通りサービスインする事ができた。

この論文では、大規模基幹システム開発におけるリスクを打破した「Topjax Solution」及び移行作業で大きな力を発揮した「Oh-Pa 1/3」のツール導入による有効性と、プロジェクトを成功裏に導けたお客様とのコミュニケーション方法、プロジェクト管理の工夫点について論じている。

■ 論文目次 ■

1. はじめに	《 4》
1. 1 PRESLEY とは	
2. フェーズ1開発	《 5》
2. 1 大幅な遅延はすぐに訪れた	
2. 2 何故、大幅な遅延に陥ったのか？	
2. 3 大幅な遅延を打破するための三つの施策	
2. 4 Topjax とは	
2. 5 何故 Topjax であったか？	
2. 6 Topjax 開発の成果	
2. 7 プロジェクト可視化の対応と結果	
2. 8 フェーズ1のリリースを迎えて	
3. フェーズ2開発	《 13》
3. 1 引き続き困難な状況に直面	
3. 2 フェーズ2開発の問題点	
3. 3 フェーズ2開発の光明	
4. フェーズ3開発	《 15》
4. 1 過去の問題点を活かしたフェーズ3の成果	
5. まとめ	《 19》

■ 図表一覧 ■

図1 PRESLEYシステム俯瞰図	《 4》
図2 PRESLEY開発スケジュール（当初予定）	《 5》
図3 PRESLEY開発体制図	《 5》
図4 概要スケジュールと実績状況	《 6》
図5 PRESLEY開発体制図（見直し版）	《 7》
図6 Topjax Solution概説図	《 8》
図7 開発環境の整備作業の比較	《 9》
図8 見直し後の画面遷移図	《 11》
図9 画面遷移図を利用した進捗管理	《 11》
図10 オンライン系画面の進捗状況	《 12》
図11 PERT図	《 16》
図12 影響一覧	《 17》
図13 故障持ち回りグラフ	《 17》
図14 故障原因グラフ	《 18》
表1 フェーズ1での想定作業ボリューム	《 6》

1. はじめに

システム開発は、開発ツールやソリューション提供により日々進歩を遂げているが、トラブルが絶える事がない状況にある。特に上流工程における要求仕様の検討不足や漏れが下流工程に与える影響は、スケジュール遵守のためのその場しのぎの突貫作業における品質劣化やメンテナンス性の悪化を引き起こし、保守フェーズや次期開発作業においてコストが嵩む一方である。

当社は、KDDI 株式会社様（以下、「KDDI 様」という）の PRESLEY システム開発において、上流工程での躓きに伴う大幅な遅延の発生に対して、開発工程の効率化とお客様とのコミュニケーション向上を図ったことにより予定通りのリリースを行う事ができた。

本論文では、発生した問題の背景と、それに対する対策や工夫点を紹介する。

1. 1 PRESLEY とは

「PRESLEY」とは、KDDI 様新法人データ系システムの通称であり、KDDI 様の法人様向けネットワークサービスにおいてサービスの申込みから工事完了・サービス開始までの作業状況を管理するシステムである。KDDI 様では、会社の合併によるシステムの重複と、維持管理コストの増大という問題を抱え、その対策としてシステム統合化の必要があった。その一環として開発検討された PRESLEY はシステム統合化の先陣をきったシステムである。本システムでは図 1 及び図 2 に示すように開発を三つのフェーズに分けた。各フェーズの開発対象は以下のとおりである。

- ・フェーズ 1：新サービス
- ・フェーズ 2：現行システム C、D の巻き取り
- ・フェーズ 3：サービス・メニュー追加及び現行システム A、B の巻き取り

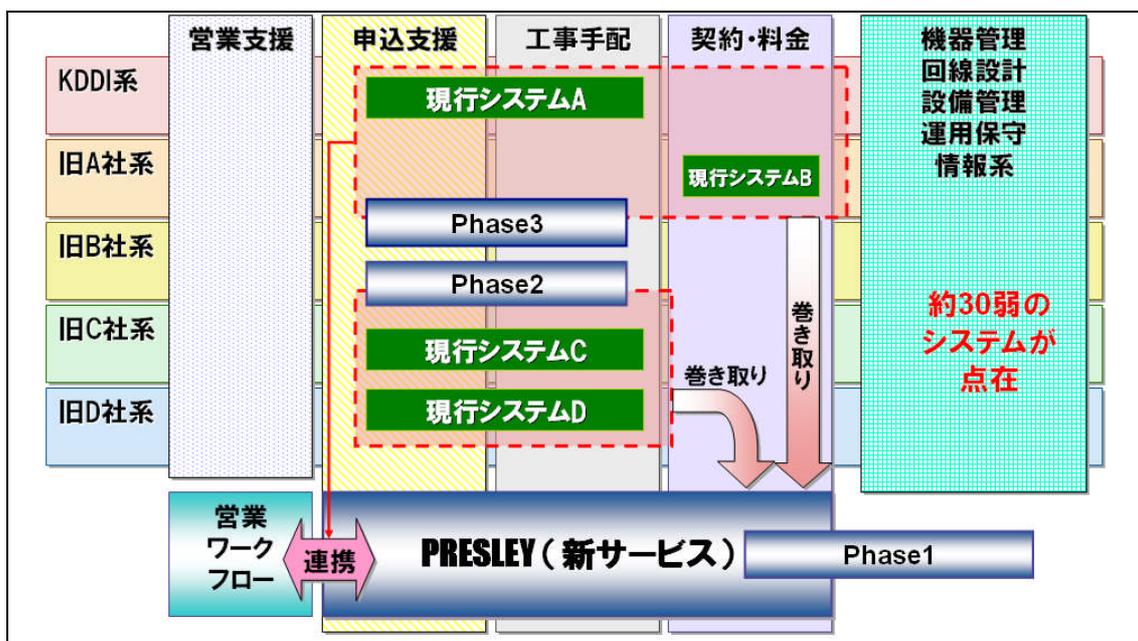


図 1 PRESLEY システム俯瞰図

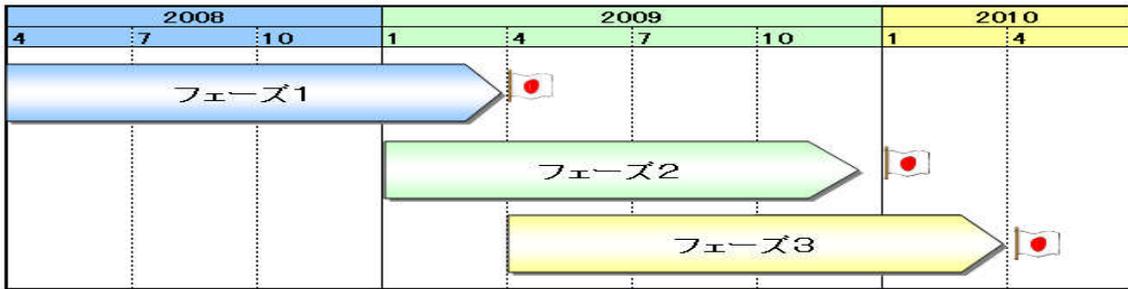


図2 PRESLEY 開発スケジュール (当初予定)

本システムではシステムの仕様を決定するにあたり、現行システムの各エンドユーザーとの調整を行う必要があった。本システムの開発体制図を図3に示す。

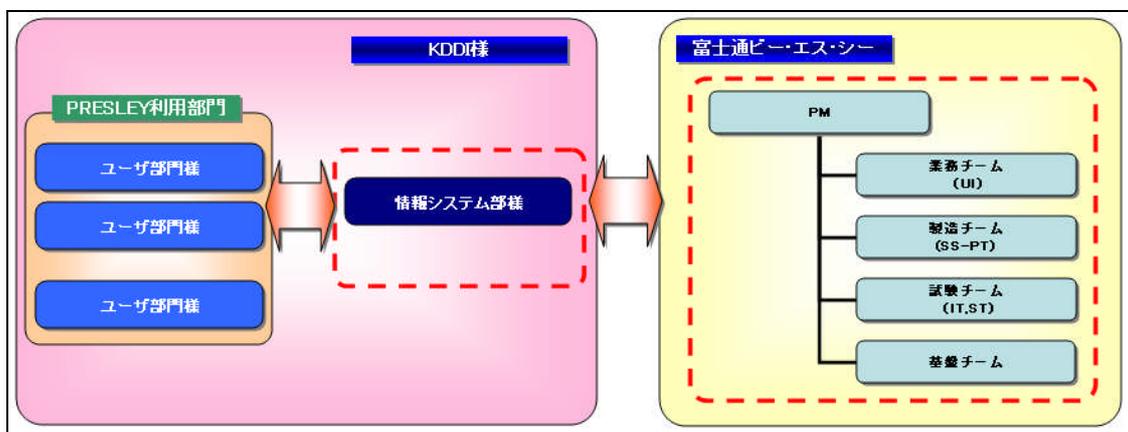


図3 PRESLEY 開発体制図

本システムの特徴は以下のとおりである。

- ・現状お客様へ提供しているサービスが 34 種¹、メニューが 160 種¹と多岐にわたり存在しており、その組み合わせにより業務フローや入出力項目が異なる。
- ・今後のサービス・メニューの追加・変更・削除を容易に行える必要がある。
- ・サービス・メニューにより入力や表示する情報が異なるので、指定したサービス・メニューにより画面表示項目を動的に変える制御や、利用者により入力できる項目や表示する項目を変動させる権限制御をマスター設定値にて行う（以下、この方式をマスタードリブンという）。
- ・業務の核となるオーダー情報の登録に関する3画面（登録・確認・詳細）が、既存システムの10画面分に相当する入力、表示項目（1画面当たり約400項目）を含んだ巨大画面となっている。

2. フェーズ1 開発

¹ PRESLEY の最終リリース（フェーズ3）までに提供する全数を指す

2. 1 大幅な遅延はすぐに訪れた

当社は、2008年6月から開発を開始した。2009年4月初めのサービス開始を第一サイクル（フェーズ1）として、対象サービスを新規の1サービスに限定したスモールスタートの形で開発作業を開始した。多くのサービス・メニューに一度に対応するのではなく、システムとしての基本的な構築を新サービスに限定した形で開始することで、既存システムへの影響を抑え、かつ今後追加していくサービス・メニューを考慮したシステム基盤構築を目指したものであり、プロジェクトのリスクを最低限に抑える方式である。

プロジェクトにおける当社の作業範囲は要件定義（RD）から総合試験工程（ST）であったが、大幅な遅延はすぐに訪れた。8月の段階で統合するシステムの要件集約が遅延したため、製造が開始できるレベルの基本設計書ができず、図4に示すように製造工程開始の予定に対して1.5ヶ月の遅延が発生した。

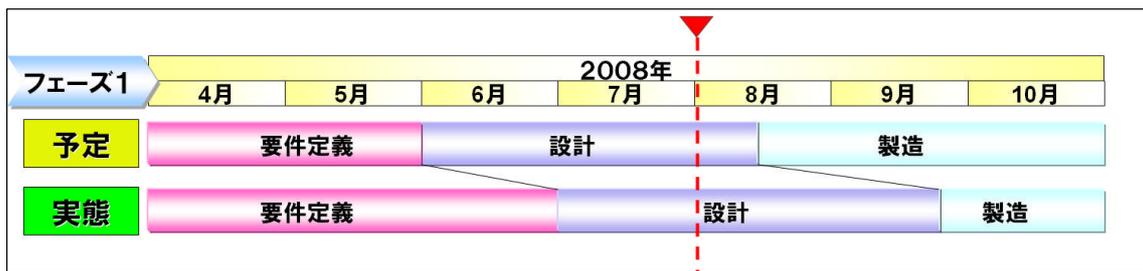


図4 概要スケジュールと実績状況

フェーズ1での想定作業ボリュームを表1に示す。

オンライン画面	オーダー管理オンライン画面	66
	料金計算オンライン画面	8
	外部委託オンライン画面	19
バッチ	オーダー管理バッチ	9
	料金計算バッチ	9
帳票	オーダー管理帳票	47

表1 フェーズ1での想定作業ボリューム

この状況では2009年1月末に予定している総合試験（ST）開始が守れない事が判明した。

2. 2 何故、大幅な遅延に陥ったのか？

大幅な遅延に陥った要因は、以下の3点であった。

- ・ システム化要件の整理に時間がかかった
- ・ 新規追加業務の要件提示が遅れた
- ・ 基本設計書のエンドユーザー中間レビューで顧客要望が多発した

前述したとおり、当システムは過去に別会社として各社で使用していたシステムを統合化する新規システムである。システム化要件の整理では、各ユーザー部門様からの、現行システム踏襲という強い要望に対応する必要がある。この要望に応えるためには、現行システムについての仕様を詳細に把握した上で、各ユーザー部門様からの要望を調整する必

要がある。仕様の把握では、旧システムのドキュメントが整備されていなかったことから、ソースコードから仕様を起こさざるを得ず、現状分析に想定以上の時間がかかり、このような遅延を発生させる要因となった。システム化要件の整理に発生したこのような遅延は新規追加業務の要件整理にも大きな影響を与えた。

結果として検討の時間が不足し、統一した仕様が固まらなかった。

基本設計書も十分にでき上がらず、レビューにおける顧客要望が多発して、検討課題が数多く残った状況で製造開始段階へ突入することとなってしまった。

進捗状況も分りづらい状況下にあった。どのような課題や問題が残されており、その影響がどの範囲まで及ぶものなのかが把握できていなかった。

フェーズ1で対象とする新サービスの運用開始は公表されており、稼動スケジュールの延期は不可能な状態の中、情報システム部様と相互で課題・問題点が共有できる状況の可視化の抜本的な対策が必要となっていた。

2.3 大幅な遅延を打破するための三つの施策

この大幅な遅延を打破するために、当社としては製造工程以降を効率良くかつ確実に計画通り進める技術的な対策を提案・説明する必要があった。

そこで、以下の3点をKDDI様へ提案した。

- ・当社で Web 開発の標準開発手法とされている「Topjax (トップジャックス) Solution (以下 Topjax という)」への開発手法の変更
- ・総合試験 (ST) について、KDDI 様への支援依頼
- ・問題点・課題を適時確認できるコミュニケーション手段として、グループウェアまたは同等の管理用ソフトウェアの導入

提案に先立ち、富士通の Topjax 統括部様のご協力を頂き、Topjax とはどのようなものであるかをお客様に説明した。結果、Topjax の有効性と導入によるスケジュール内での実現可能性を理解して頂くことができ Topjax の適用が決定した。また、総合試験についても業務フローを中心とした業務視点の試験の強化と効率化との必要性から KDDI 様に支援をして頂くことが決定し、図5に示すように開発体制の見直しが行われた。

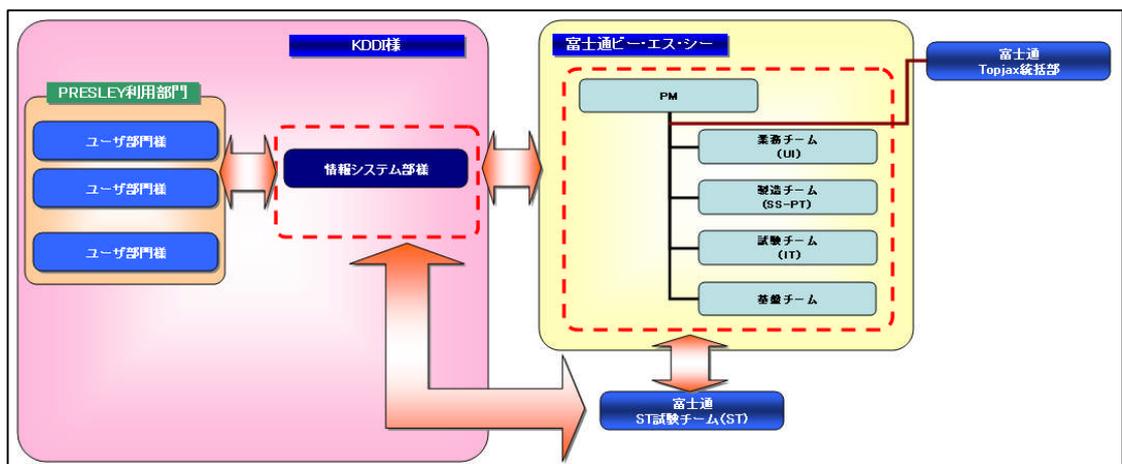


図5 PRESLEY 開発体制図 (見直し版)

2.4 Topjax とは

お客様のビジネススピードの加速に伴い、システム開発に与えられる期間は短くなってきている。また、Web システムの構築に使用されるオープンシステムは、

- ・ソフトウェア、ハードウェア選択の自由度が高い
- ・設計、開発、運用に先進技術が取り入れられやすい
- ・プライスフリーや製品間の競争により安価である

といったプラスの面がある反面、

- ・新たな方式の出現や、機能追加など、技術の変化がめまぐるしい
- ・採用する構成要素の組み合わせの判断が難しい
- ・開発・運用環境の構築に、採用した構成要素の組み合わせにおけるノウハウが必要
- ・トラブルが発生した場合の原因の切り分けが困難となる場合があり、責任の所在がいまいになりかねない

といったマイナスの面も併せ持っている。

このマイナス面に対し対策無く開発を行うと、

- ・上流工程では、構成要素の選択や実現性・性能の確保
- ・下流工程では、開発・運用環境の構築や、開発技術の習得、品質の確保

と、それぞれにおいて予定外の期間や費用を要してしまうリスクを負う可能性が高くなる。

Topjax Solution (Topjax = Total Omni platform architecture by java and xml technology) (図6にて概説) は、富士通の Topjax 統括部様が開発・運用している SI ソリューションであり、本来ビジネスソリューションの構築に向けられるべき資源が、構築の基盤となる IT 技術そのものに削がれてしまうことを防ぐとともに、加速する一方であるお客様のビジネススピードに対応できるリードタイムを確保して、お客様のビジネスへ貢献することを第一に考えて作成されている。

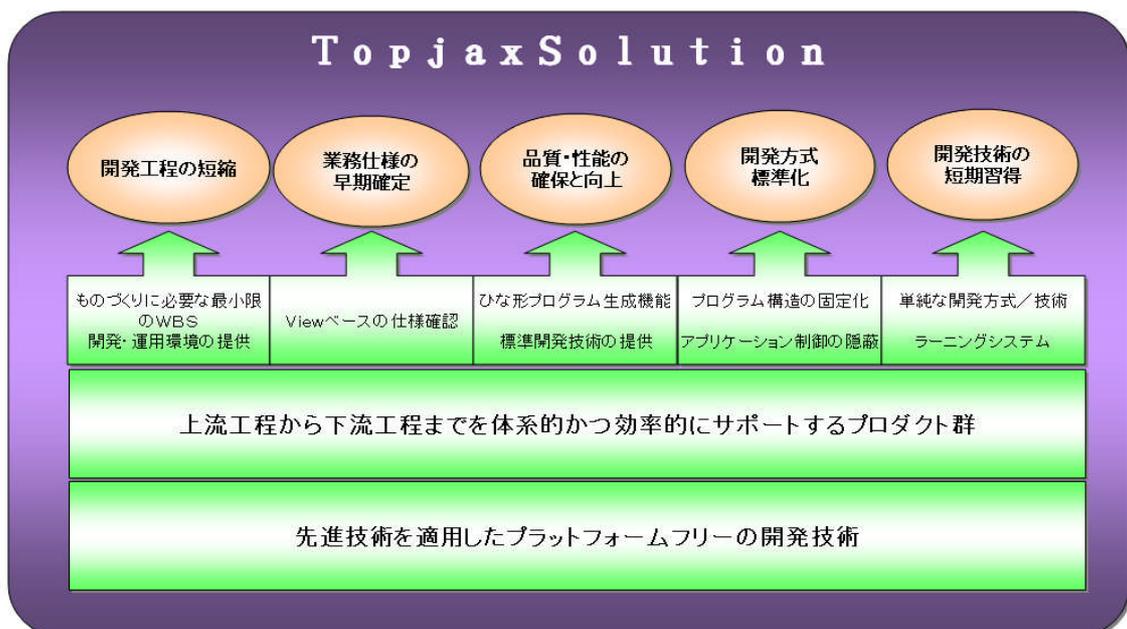


図6 Topjax Solution 概説図

特徴として、以下のようなことがあげられる。

- ・ 予め検証された技術・方式をベースとしたセミオーダーメイド開発により、短期開発での実現性・性能・品質を確保（プログラム構造固定化、アプリケーション制御機能実装不要、ものづくりに必要最低限のWBSに絞り込まれた開発工程）
- ・ 上流工程から下流工程までをトータルにサポートする、「適用審査、プロジェクト診断、教育サービス、インフラ・開発技術についてのガイドライン、各種ツール」などのソリューションを、ワンストップで提供
- ・ ツールは、Java™とXMLを使用してオープン環境でプラットフォームフリーを実現
- ・ 開発支援ツールの富士通受注プロジェクトでの利用は無料²（社内向けツール扱い）
- ・ 運用支援ツール（ランタイム）の使用権は、開発対象システムに対して無償³で提供
- ・ 開発環境、運用環境が構築済みとして提供される IDC 開発センターと、プロジェクトが自前で構築する場合に向けてのガイドラインとを提供

2. 5 何故 Topjax であったか？

一般的にシステム開発では、開発環境の整備に多くの工数を要する。開発環境の検討に始まり、インストールの後には、システム構成要素の組み合わせに応じた設定や動作確認、そしてチューニングを行う必要があるからである。

Topjax では、トータルサポートの一環として、図7に示すように、開発者が容易に作業を進められるように、統合開発環境（正式名称：「Topjax IDC」）が準備されている。統合開発環境では、デザインツール（画面遷移図・

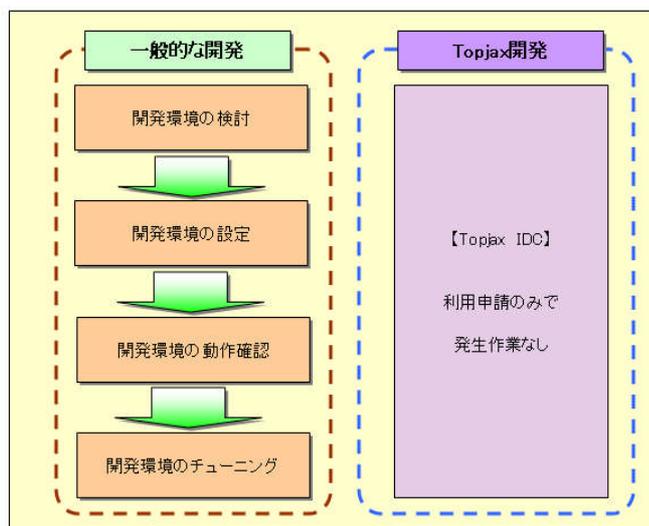


図7 開発環境の整備作業の比較

HTMLソース・Javaソースのエディタ機能を持つWebアプリケーション）と、ライブラリツール（開発資産の世代管理、規約チェック、コンパイルの機能を持つWebアプリケーション）と、SIノウハウツール（作業指示、報告、議事録管理、QA管理の機能を持つWebアプリケーション）が、Webアプリケーションサーバソフトウェア、データベースソフトウェアとともに提供される。このため、開発者にて開発サーバを構築する必要がない。また、開発者が開発端末に個別の開発用アプリケーションソフトを用意する必要もない。このような開発環境が準備されていることに加えて、統合開発環境における開発では、画面遷移図と画面HTMLによるひな型ソースプログラム自動生成機能により、画面制御に関するコーディングが不要となるので、開発担当者は業務ロジックの開発に専念することが可能とな

² 2010年8月31日現在

³ 2010年8月31日現在

る。また、コンパイル後のソース診断や開発に関する基準となる様々なドキュメントも用意されているので、ソースコードレビュー前や単体試験前に機械的に危険なコードの検出が可能となっている。

また、Topjax には Topjax Solution の情報提供、適用推進、並びにフィールドサポートのためのコミュニケーションを目的とした統合 Web サイトである、「Topjax QPIT (トップジャックスキューピット)」が用意されている。「Topjax QPIT」は「適用審査、プロジェクト診断、教育サービス、インフラ・開発技術についてのガイドライン、各種ツール」など開発技術をはじめプロジェクト運営に関する範囲まで幅広い支援を、ワンストップで提供している。Topjax を利用する際に必須となる導入の e-Learning や、Web 開発に必要な情報や作業指針が網羅されているので、技術的な不安点、不明点の解消のみならず、各担当が統一された指針のもと開発作業を行う事ができるので、品質のバラツキが最小限に抑えられる効果がある。

これらの特徴から Topjax を導入することにより短期間での開発が可能と判断し、結果としてお客様から Topjax 適用の了承を得る事ができた。

2. 6 Topjax 開発の成果

Topjax の導入が決定した 9 月末、当社の Topjax 経験者を集結させて、製造部隊を結成した。私が PRESLEY プロジェクトへ参画したのもこの時期となる。

製造部隊は全員、富士通の Topjax 統括部様に常駐し、統括部様の指導のもと、製造作業の準備に着手した。

(1) 画面遷移の見直し

「Topjax Solution」では、画面遷移図、画面設計書 (HTML)、DB 設計書の内容を入力として、ひな型ソースプログラムの自動生成を行う。画面遷移図については、基本設計段階では画面単位の画面遷移図が作成されていた。Topjax に投入するにあたっては、全体として整合性のある画面遷移図が必要となる。その段階で確定していた画面遷移図を統合すると、模造紙 3 枚に相当する複雑な遷移図となり、そのままではシステムとして矛盾が発生することが判明した。そこで、作成した画面遷移図を壁に貼り出し、業務の流れと画面遷移の整合性を確認しつつ、システム全体として矛盾無くシンプルな構造となるよう分析・調整を行った。

全体画面遷移図の作成手順は以下のとおりである。

- ・大きな機能単位にブロック分けを行い、縦方向に遷移を並べる。
- ・その機能単位に並べたブロックから横方向のほかのブロックへ遷移しないように調整 (必要に応じて自ブロックに画面を複製) を行う (スパゲッティ遷移とならないための必須条件)。
- ・複製した場合には画面単位に色分けを行い、機能によって異なる遷移が発生しないかを確認し、異なる遷移があった場合は、同一の遷移となるよう仕様調整を行う (画面単位に色分けすることで視覚的に見やすくなり、確認漏れを防ぐ効果がある)。

上記ポイントを中心に見直しを行った結果として模造紙 1 枚に収まるシンプルな画面移図が完成した。見直し後の画面遷移図のイメージを図 8 に示す。

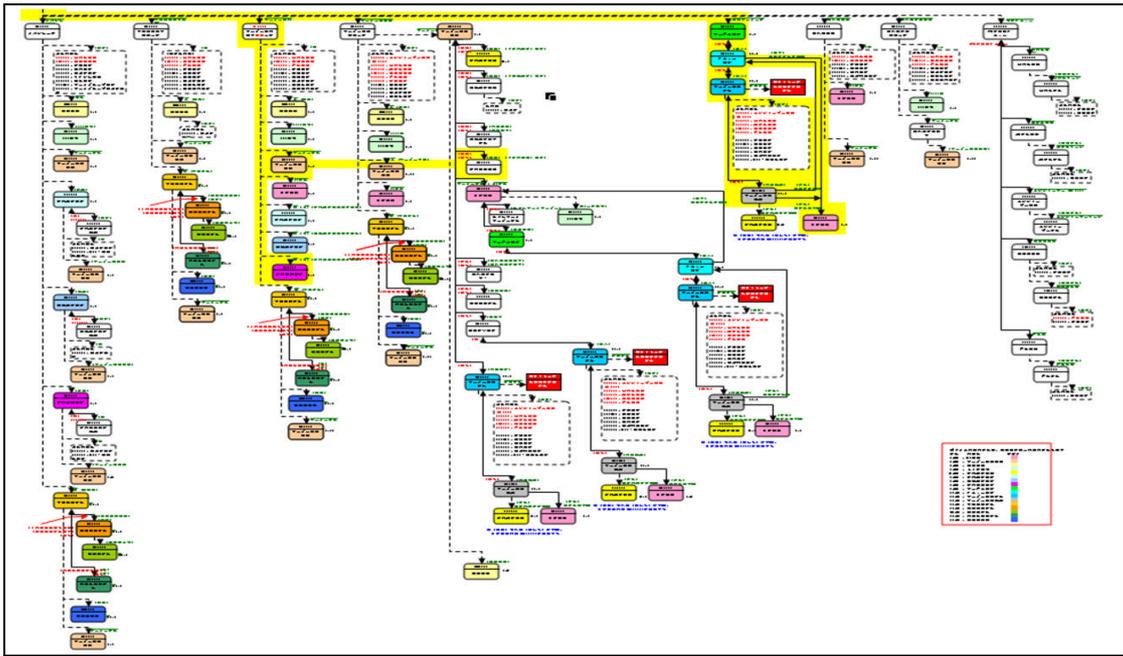


図 8 見直し後の画面遷移図

また、この画面遷移図を利用して進捗管理を実施した。図 9 にあわすように、完了、着手 (PG と PT で分類)、未着手の 3 段階 4 色にわけて日々色付けを行いこの画面遷移図を貼り出すことで、進捗状況の可視化を行った。

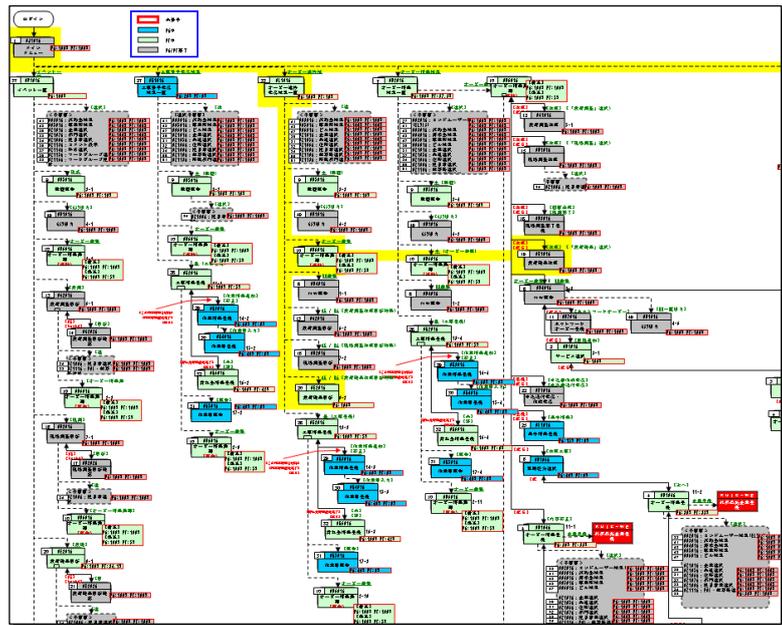


図 9 画面遷移図を利用した進捗管理

(2) 画面遷移図を入力とした雛形クラスの自動生成

Topjax は、画面遷移図と画面 HTML をツールに入力することで自動的にボタン押下時に呼び出されるクラスの Java ソースの雛形を出力するので、製造担当者は画面制御のロジックを考慮する必要がなく、業務ロジックの実装に専念すればよい。通常、画面遷移の制御も実装する必要があり遷移誤りなどが発生するが、Topjax では制御の実装が不要であるため誤りの発生が少ない。製造途中で発生した遷移先の変更に対しても、通常は作業を進めるにしたがってドキュメントと実装が乖離していくことが多いが、Topjax は画面遷移図と

いうドキュメントを入力にして雛形ソースを生成するため、ドキュメントとの乖離は発生しない。このため、保守フェーズにおけるメンテナンス性や次期開発時における作業効率が高くなる。

(3) オブジェクトプールの機能

Topjax で画面項目などの値を管理しているオブジェクトプールは、業務仕様を項目名に反映し、統一を行うことで、画面間でのデータの受け渡しの際に、同一項目名で定義された項目の自動的な受け渡しが可能である。また、画面入力項目の属性（型、フォーマット、最大値、最小値、データ長）の指定が可能であり、誤って異なる属性のデータを入れた場合、属性エラーを自動的に業務クラスへ通知する機能を持っている。通常開発側で考慮しなければいけないデータの受け渡しやエラー時の制御に関する実装を削減することができるようになっている。

(4) 強力なログ機能

Topjax は 2 種類のログが自動出力される仕組みが用意されている。Topjax 実行情報である各種のオペレーションログとシステムエラー発生時のエラーログが自動出力されるので、製造時に個別のログ出力機能を埋め込む手間が省ける。また Topjax 実行情報には実行した SQL 文を出力できるので、データアクセス系の不具合解析が瞬時に行える。これらの機能は、開発時のみならず本番リリース後の商用環境で発生した故障に対しても迅速な分析が可能となり、問題解決の早期対応に大きな役割を果たした。

図 10 が、フェーズ 1 の PG～PT 工程の進捗グラフである。予定に対して実績が途中で大きく後退している。最初は、何とか予定線に近いところで進捗していたが、11 月末に大幅な修正が発生したことで全面的に作業を見直し、最終的に 1 ヶ月遅れることになった。この間の仕様変更が 48 件、設計変更 155 件、DB 定義の大元からの変更 3 度という変更が起っていたにもかかわらず、遅れを 1 ヶ月に止めてシステムを完成することができた。

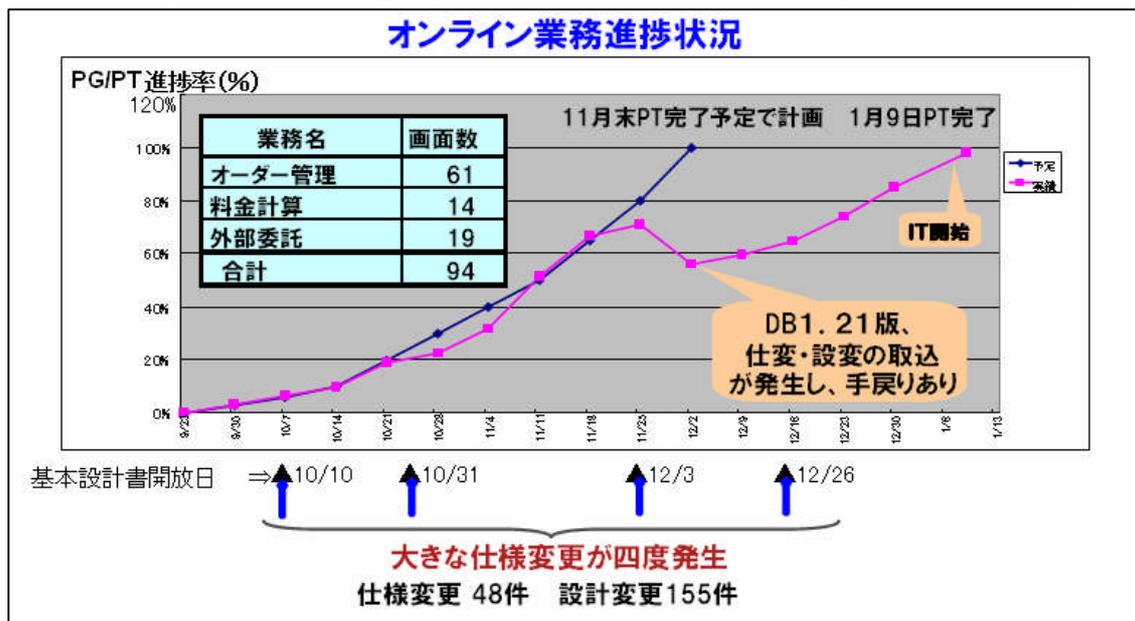


図 10 オンライン系画面の進捗状況

2. 7 プロジェクト可視化の対応と結果

問題点・課題を適時確認できるコミュニケーション手段として、サイボウズ®デヂエ® (以下、「デヂエ」と略す) を活用した。

デヂエは、Web 版のデータベースソフトウェアであり、当初は構成管理の運用のために導入されたが、柔軟なインタフェースを持っていることから、各種管理用の機能を順次取り込むようにして活用を図った。

実際に運用された機能は以下となる。

- ・ 課題管理
- ・ 故障管理 (IT、ST、OT、商用故障)
- ・ 構成管理 (リソースのチェックイン、チェックアウト管理)
- ・ 案件管理 (仕様変更を含め新たに発生した作業案件の管理)
- ・ 作業依頼管理 (他チーム間での作業依頼と経過管理)
- ・ 掲示板
- ・ 運用スケジュール
- ・ 商用切り替え作業時のタスクと進捗状況
- ・ 用語集

これらの情報をデヂエ上で管理することで、プロジェクトメンバーのみならず、お客様においても、現在発生している問題点や課題の状況を適時確認することができ、緊密な情報の共有ができるようになった。本格的なグループウェアを導入しなくても十分なプロジェクトコミュニケーションツールとしての役割を果たした。

特に有効であったのは、一覧表示からカレンダー表示にビューを切り替えられることである。いつ、どのタスクが完了すべきかが明確となった。これにより作業進捗だけでなく、いつ何件の案件見積もりの提出やリリース案件があるかといった予定が視覚的に把握できるので、リリース案件のリリース漏れ防止管理作業の効率がアップした。また、リリース漏れ防止の観点でも効果があった。

2. 8 フェーズ1のリリースを迎えて

ほぼ無理だと言われた状況下において、Topjax 適用を中心とした作業効率化により、2009 年1月初旬に見直しスケジュールの予定通りに全画面の単体試験が完了し、結合試験が開始された。結合試験工程では、一週間の疎通試験をはじめとして、予定通りに試験は実施された。製造部隊は、試験で出た故障対応作業に並行して、追加案件の取込み作業を実施したが、構成管理の仕組みが確立されていたことと KDDI 様のご協力の結果、途中、トラブルはあったものの、試験工程を完了でき、予定通り5月の商用リリースを迎える事ができた。

3. フェーズ2開発

3. 1 引き続き困難な状況に直面

フェーズ1開発に引き続き、フェーズ2開発でも困難な状況に直面した。

フェーズ1開発は、上流工程の進め方の甘さに起因する大幅な遅延をTopjaxSolutionの適用により脱して、見直しスケジュール通りの2009年5月にサービスインすることができたが、フェーズ2開発はこれに先立つ2009年2月に既に作業を開始していた。

フェーズ1での開発対象は、1サービス1オーダーの単機能レベルであったが、フェーズ2での開発対象は、1サービス複数オーダーに対応するとともに現行システムの巻き取りも含まれていた。

当初は、フェーズ2の設計担当の主体として、フェーズ1の設計担当を予定していたが、フェーズ1の遅れを取り戻すために、商用リリースの直前まで大方の設計担当がフェーズ2へ参画できず、新メンバー主体で設計作業を進める事となった。フェーズ1の設計担当が、以前からKDDI様のシステム開発を行ってきた業務有識者主体であったことに比べて、フェーズ2の設計担当はKDDI様のシステム開発の経験のないメンバーが主体とならざるを得ない状況となったのである。フェーズ2の設計工程は業務知識や既存機能の理解がほとんどないメンバーによる作業開始となった。

結果としてフェーズ2の設計作業は、設計完了予定であった5月になっても明確な仕様確定がなされず、遅延に陥ることとなった。この状況は、フェーズ1同様TopjaxSolution適用による作業効率化の効果と、後述するフェーズ2開発での新たな対策の効果によって回復に向かい、フェーズ2開発も予定通りリリースを行う事ができた。

3. 2 フェーズ2開発の問題点

フェーズ2開発における大きな問題点は以下のとおりである。

- ・基本設計書の内容が詰められておらず、そのままでは製造に着手する事ができない（フェーズ1と同様の現象であるが、基本設計書の完成度でいうとフェーズ1より低いレベルとなっている）。
- ・マスター設定作業での入出力が明確でなく、特に中間成果物の扱いが曖昧であったためにマスター設定作業に大幅な作業遅延が発生した。このことにより、結合試験の疎通試験が進められないという状況にまで陥った。

主な原因は、フェーズ1での設計担当がフェーズ2へ合流することが遅れたことに起因する設計工程の遅れの問題であった。そのほかの原因としては、当システムの特徴でもあるマスタードリブンによる画面項目制御の本質を理解しないまま作業を進めた事がある。各データの影響範囲が押さえきれず、反映漏れが多発した。また、中間成果物の取り扱いが曖昧であったことからデータ間の不整合の多発を惹き起した。このように、フェーズ2では、マスター設定に関する作業の進め方の問題が、新たにクローズアップされる結果となった。マスタードリブンを踏襲するシステム開発においては、特にデータ特性を理解し影響範囲を確実に押さえていくことがいかに重要な課題であるかが明確になり、その後のフェーズ3の開発においては、作業手順を含めて抜本的な見直しを行う必要が認識された。

3. 3 フェーズ2開発の光明

これらのフェーズ2での問題に対して効果が発揮できた対策があった。以下三つを採り上げる。

一つめは、データ移行の対応において、当社製品のオンメモリデータベース「Oh-Pa

1/3」(オーパ・ワンサード)を適用した結果、短時間で精度の高いデータ移行を実現でき、移行時間もシステム切り替えの制約時間内で実施できたことである。通常データ移行では、移行元、移行先双方の仕様に基づいて移行仕様を決定しても、実データが仕様通りに入っていないケースもあり、

「実データ調査と仕様調整に時間がかかる」

「移行時、及び移行後に想定外データを原因としたシステムトラブルが発生する」

などの課題が発生するが、「Oh-Pa 1/3」は、大量のデータをリアルタイムにGUIで検索や絞り込み、テーブル間結合ができるため、その適用は、実データ調査の時間短縮化、移行仕様の早期決定という効果をもたらした。また、GUI操作をマクロプログラム化することができるため、開発効率の向上にも貢献した。更なる高速性から、性能チューニングを行うことなく、制約時間内で移行処理を完了できた。このことでリハーサルを繰り返し実施することが可能となり、品質を向上することができた。

二つめとしては、製造影響の早期発見と設計からの引継ぎをスムーズにし、製造効率を上げることが目的として、開発部隊が基本設計の一部を担当した。これにより、製造部隊の業務理解が進み、開発の効率化につながった。結果としてこのような状況の中でも予定通りに開発作業が完了した。

三つめは、既存システムの10画面分に相当していたオーダー情報の登録系画面に対して画面分割の提案を行い、採用されたことである。結果的にお客様にとって管理しやすいレベルの項目数の画面を提供することができたと考えられる。

4. フェーズ3開発

4.1 過去の問題点を活かしたフェーズ3の成果

作業過程では大きな問題点を抱えたが、フェーズ2は当初の予定通りに商用リリースまで進めることができた。そして、プロジェクトは最終のフェーズ3(2009年7月～2010年4月)へと進んだ。集大成となるフェーズ3では、大幅な遅延を発生させることなく、予定通りに商用リリースを迎える事ができた。

フェーズ3が、成功裏に完了したポイントは以下のとおりと考えている。

(1) 作業手順及び管理方法の見直しと効率化

作業手順の見直しとして、以下を重要課題として取り組んだ。

- a. 作業タスクごとの入出力を明確にして、影響範囲を早期に押さえる
- b. 作業の反映漏れを防ぐための横並びチェックの方式を明確にする
- c. 故障分析の結果を可視化して品質意識を向上させる

a) 作業タスクごとの入出力を明確にして、影響範囲を早期に押さえる

フェーズ2では、マスター設計において作業手順及び作業タスクの入出力が不明確で混乱した事を踏まえて、WBSと併用して作業タスク単位のPERT図(図11)を作成した。

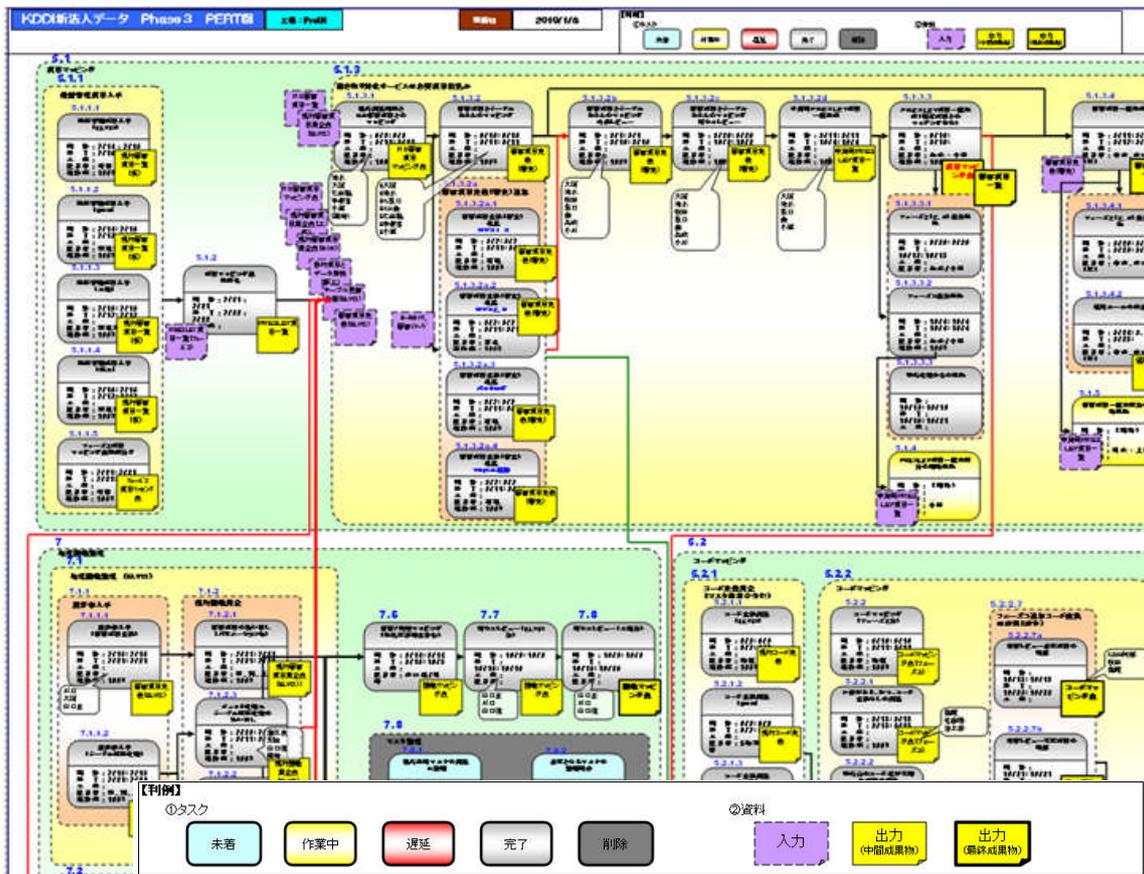


図 11 PERT 図

この PERT 図では、作業タスク単位に

- ・ 作業開始・終了日（予定／実績）
- ・ 担当者
- ・ 作業工数
- ・ 進捗率
- ・ 中間成果物を含む入力及び出力ドキュメント名

を記入し、個々のタスク以外に WBS の第一レベル、第二レベルによるブロックも表す事で、WBS との併用による作業管理を実現した。

更に、作業進捗も明確となるよう、PERT 図中のタスクに日々進捗状況を色付けした。遅れが出ているタスクは目立つよう背景色を赤にすることで、その遅れがどのタスクに影響を及ぼすかが明確となった。また日々更新版を壁に貼り出した結果、作業の見える化が進み個人のスケジュール意識の向上へと繋がった。

b) 作業の反映漏れを防ぐための横並びチェックの方式を明確にする

各作業案件の影響調査漏れを防ぐために、全サブシステムでの「影響一覧」の作成を義務付けた（図 12）。これは一部サブシステムでフェーズ2から導入していたものを全サブシステムへ適用することで、サブシステム間で発生した影響調査漏れを防ぐことを目的としたものである。

この影響一覧は、縦軸の全案件と、横軸の各サブシステムで作業対象としている画面名や機能名とで構成されるマトリクスであり、案件単位に改修が必要な箇所にマークを記述す

ることで影響範囲を確認するものである。運用ルールで「影響がない場合もない事をはっきりさせる」として、未記入はNGとした。また、マークに基準を設けて、進捗状況も同時に把握できるようにした。この結果、関係者全員が横並びの観点を意識して、対応漏れの故障を一蹴することができた。

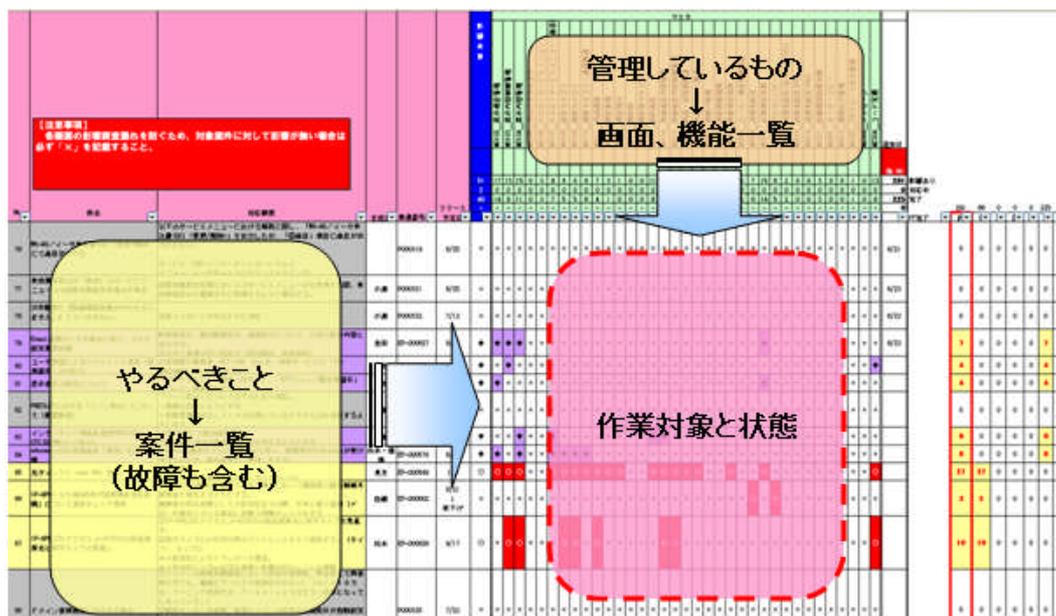


図 12 影響一覧

c) 故障分析の結果を可視化して品質意識を向上させる

商用故障をはじめに各試験工程の故障データはデジエで一元管理されており、適時 CSV ファイルへ出力できる。この CSV ファイルから故障分析を行うための Excel のグラフを作成する作業を半自動化して、毎日の朝会で棚卸しを実施した。図 13 がグラフの例である。

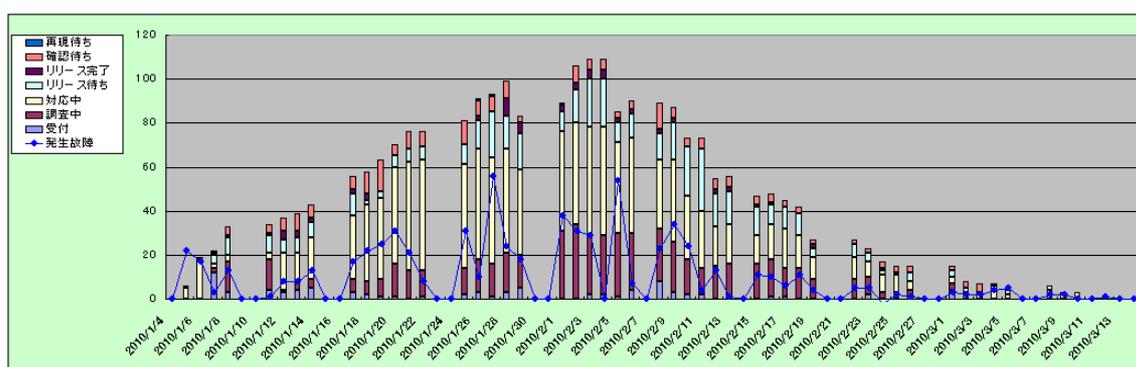


図 13 故障持ち回りグラフ

同時に残故障の状態や故障の原因（根本原因、故障混入工程、発見すべき工程）も視覚化した（図 14）。結果として、故障の傾向が明確となったことで、具体的な品質強化策が考えやすくなった。また、開発担当の品質意識向上に役立った。

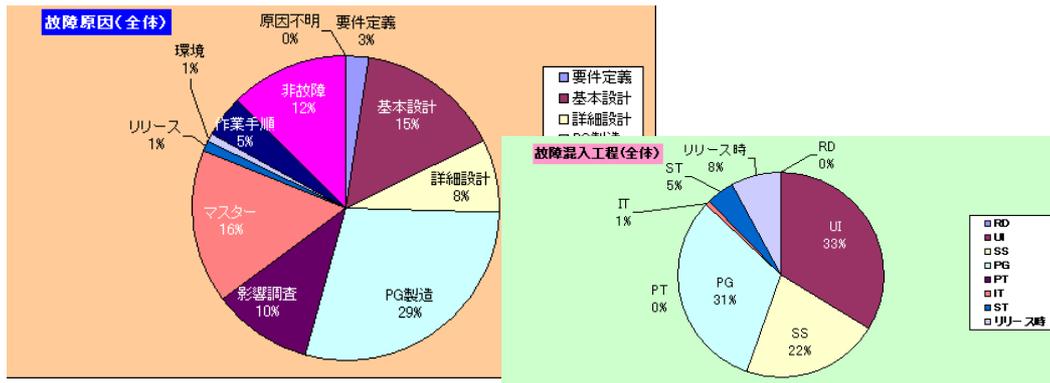


図 14 故障原因グラフ

(2) お客様とのコミュニケーション強化により、共通意識で作業を遂行

本システムの特徴であるマスタードリブンの提唱者はお客様担当である。そのお客様のシステム化のポリシーを全員で意思共有することで、システムとしてあるべき姿を再認識し、問題点や改善点を明確にするため、2日間の合宿を実施した。

この合宿は、先に述べたマスタードリブンの提唱者であるご担当様による講義と質疑応答を中心とした技術論の周知と、フェーズ3プロジェクトの進め方や過去の反省を討論したキックオフとを兼ねたものであった。その後、週に一度その担当者様を中心とした課題検討会を開催することで意思統一ができ、またフェーズ2で問題となったマスター関連の作業タスクも明確となった。

結果として、フェーズ3では単体試験段階でマスターが準備でき、マスターの品質確認を早期に行う事ができた。また試験向けのマスターを準備する手間が省けたことで試験実施側の作業効率が上がった。

(3) Topjax 開発による安定及び計画的な作業の確立

フェーズ3では、製造担当は基本設計書の作成から対応を行った。フェーズ2で一部設計書の作成から参画したことで、お客様と直接会話できる機会が増え、画面分割などの提案ができた事に加えて、設計段階で製造に関係する考慮点を意識することで製造時に発覚する設計問題の早期発見ができた事が成果として見られたが、その成果をフェーズ3でも継続させる事を目的とした。

基本設計書作成段階で、画面遷移のみを確認できるように紙芝居形式の画面を作成し、ユーザーデモを実施した。設計書のみでは明確になりづらい点のはっきりし、精度の高い画面設計が実現できた。また、この段階で画面の統廃合が実施できた事で、画面構成もシンプルとなり一定の成果があがった。

製造面においては、フェーズ3ではすべての新規画面を新人及びTopjax 初心者が製造担当したことで製造担当の底上げができた。また、既存メンバーは従来の担当と異なる画面の変更対応を行った事で、1画面を複数の担当でメンテナンスすることが可能なレベルとなった。属人化が解消されたことで、その後の保守フェーズで製造担当の人数を絞り込んでも引継ぎ時間や不具合発生時の対応時間が短縮された。

(4) 「Oh-Pa 1/3」活用による作業時間の短縮と効率化

フェーズ2で導入を行い、大きな成果をあげた「Oh-Pa 1/3」は、フェーズ3のデータ移行でも効果を発揮した。

フェーズ3では、巻取り対象サービスが大幅に拡大され、データの複雑度が格段にあがったが、フェーズ2で確立した作業手順によって、更に短期間での対応が実現できた。また、さらなる「Oh-Pa 1/3」の活用として、移行データとマスターデータの整合性チェックを実施することにより、移行データの精度を向上するだけでなく、マスターの精度向上にも貢献した。更に、本番で発生した故障に対して、大量のログファイルを「Oh-Pa 1/3」で分析することにより、迅速な原因追求と影響範囲調査を行うことができた。

5. まとめ

フェーズ1及びフェーズ2で大幅な遅延が発生したが、プロジェクトは集大成となるフェーズ3において安定した状態で作業を進める事ができた。プロジェクトは、失敗の中から如何に効率良く作業を行うかの検討を繰り返しながら、メンバー全員が共通の問題意識を持って作業の効率化を進めることができた。これは、当社のプロジェクトメンバーのみならずお客様とともに、関係者全員が一丸となったと結果といえる。

但し、想いだけではプロジェクトは進化しない。考えた事を如何に実現していくかが大きなポイントであり、その意味では今回の取り組みはプロジェクト自体が能動的に進められたことにより可能となったといえる。

特に、Topjax と Oh-Pa 1/3 の適用効果は、プロジェクトの安定性に対して大きな成果があった。このような有用な開発環境やツールを適用することで、少なくとも開発工程以降については計画がたてやすくなることは間違いない。

今後もこのような有用なツールを適用していきながら計画的なシステム開発を推進するような提案活動を進めていく。

また、今回のプロジェクト活動全体を通して効率化に役立ったノウハウは、次期開発や他案件の開発において、プロジェクトの立ち上げ時から意識的に適用することで、リスク軽減及び安定作業へと導く事ができると確信できたので、再利用に向けて開発セットとして整備を行っていく。

最後に、PRESLEY プロジェクトは、スケジュール通りにリリースができた事で、KDDI 様より Topjax 適用によるスケジュール遵守についての感謝状を頂く事ができた。

Java は、Oracle Corporation 及びその子会社、関連会社の米国及びそのほかの国における登録商標です。

サイボウズ 及び デヂエ はサイボウズ株式会社の登録商標です。