Vol.2012-GN-82 No.17
Vol.2012-CDS-3 No.17
2012/1/20

OMCS 構築のプロジェクト管理における見える化

相澤正俊† 上窪真一†† 小河原孝一†††

我々は，オープンシステムで世界最先端のミッションクリティカルな大規模基幹システムの構築を通して，OMCS(Open Mission Critical System)構築技術を蓄積してきた．本論文では，OMCS の開発の特長である①オープン製品の組み合わせによる開発，②分散開発，③大規模開発，④スピード開発，⑤高品質に着目し，その課題と実際に適用した技法について，特に，見える化に着目したプロジェクト管理の観点から述べる．本論文で述べる内容は単なるコンセプトやファシリティのアイディアではなく，実際の大規模システム構築で適用され，その成功に大きく貢献した技法である．

Project Management Visualization in OMCS architectures

MASATOSHI AIZAWA† SHINICHI UWAKUBO††
KOUICHI OGAWARA†††

We, through the construction of large scale mission critical enterprise systems in the world's most advanced open systems, OMCS (Open Mission Critical System) technology has accumulated. In this paper, we feature the development of OMCS (1)product mix development, (2)distributed development, (3)large scale development, (4)rapid development, (5)high quality, and describe its problems and real challenges. In particular, we focused on project management visualization.

1. はじめに

我々は，ミッションクリティカルな大規模基幹システムの構築を通して，およそ 15 年に渡り OMCS(Open Mission Critical System)構築技術を蓄積してきた．OMCS とは，オープン製品(Best of Breed[a])やオープン技術(デファクトスタンダードな技術)を駆使して構築された大規模基幹システム，および，その基幹システムを構築するためのミドルウェアを中心とした製品群と SI(System Integration)技術のメソドロジー(方法論)の総称である[1]．OMCS の特長を以下に示す．

- 大規模基幹システム
  基幹システムが扱う情報と処理量の増大(1 桁～2 桁増)に伴い，メインフレームのスケールアップでは対応しきれない大規模基幹システムが必要とされ，メインフレーム並のミッションクリティカル性を持ちながら，スケールアウト可能な分散システムによりシステムを構築．
- オープン製品，オープン技術を利用
  メインフレームは，ハードウェア・ソフトウェアを含めて垂直統合されていたため，結果的にコスト高を招き，これに対抗する形で登場してきたオープン製品，オープン技術を利用してシステムを構築．
- スピード開発
  基幹業務の多様化，複雑化に対応しつつ，インターネット時代のスピード化に合わせ，サービスインまでの工期短縮(1/2 以下目標)を実現．

1990 年代前半の歴史的背景は，メインフレームからオープンシステムへ動き始めた時代である．メインフレーム時代は垂直統合であり，すべて同一ベンダー製の製品群でシステム構築を行うのに対し，オープンシステム時代は水平分散であり，ワールドワイドの「Best of Breed」製品を使いシステム構築を行う形態に変化した．

そのため，プロジェクト中心の進捗・品質管理等の単なる見える化だけでなく，より広い意味で，見える化指向のプロジェクト管理が重要になった．

本論文では，OMCS 構築にあたり，現実のプロジェクトで実施され，成果のあった，リスクの見える化に着目したプロジェクト管理のエッセンスについて述べる．

2. 見える化に着目した OMCS 開発の課題

このような OMCS の開発にあたっては，以下の特長をブレークスルーする対策が必要である．

(1) **オープン製品の組み合わせ開発**

- その時代時代ベストのオープン製品を組み合わせてシステムを構築．
- メインフレーム時代の自社製品の組み合わせと異なり，多様なベンダーの多様な製品を組み合わせてシステムを構築．

(2) **分散開発**

† 国際社会経済研究所 理事長（前ＮＥＣ副社長） Institute for International Socio-Economic Studies
†† 日本電気株式会社 システム技術統括本部 シニアエキスパート NEC Corporation
††† ノイテック Neutech

a グローバルにデファクトスタンダードになっているハードウェア製品やプログラムプロダクトを活用すること

Vol.2012-GN-82 No.17
Vol.2012-CDS-3 No.17
2012/1/20

- 多数のステークホルダー.
- 数百～千人超の開発要員を，国内，海外に求め分散して開発.

(3) **大規模開発**

- 数メガ～十メガ以上の規模のソフトウェア開発.

(4) **スピード開発**

- 基幹システムの再構築ゆえに，企業力アップのため開発期間短縮
  それまでのメインフレームやオープンシステムでの開発期間に対して 1/2 以下を目標とした.

(5) **高品質なシステム開発**

- OMCS は，クライアント・サーバモデルなどの一般のオープンシステムより数桁上の高信頼性・高品質・高性能・高拡張性等が要求されるシステムである．従って開発するソフトウェアも一般のソフトウェアより 1 桁上の品質が要求される．例えば，大規模サービスプラットフォームモデル・超並列モデルを採用したインターネットサービスプロバイダシステム[1]では，稼動率 99.9999%のシステムを実現.

以上の各課題に着目し，特にリスクに対する見える化が重要とされる問題点を整理する.

(1)オープン製品組み合わせ開発では，障害の切り分けが見えにくくなる課題があげられる．これは，それまでのメインフレームとは異なり，システムを構成するハードウェアやソフトウェアの中身が見えなくなっているためであり，その対策としては，メインフレーム相当の障害対応の見える化が重要である.

(2)分散開発は，言語，時差，距離の違いにより，プロジェクトの状況が見えにくくなる．また，ステークホルダーが多いため，認識のずれや意思決定の遅れが生じるリスクがある. これらの対策としては, 工程毎の SI プロセスを見える化する必要がある.

(3)大規模開発は，それまでのメインフレームと比較して，システムの大規模化，機能の複雑化が著しく，そのため工程毎の状況(進捗，品質)が見えにくくなる．このような課題に対しては，リアルタイムな進捗の見える化が重要である.

(4)メインフレームやそれまでのオープンシステムでの開発はプラットフォーム構築とアプリケーション開発が独立して行われ，最終的に結合評価されていたため，システムの問題点(バグ，不整合)の発見が遅れ勝ちであり，工期の延長を招くことが多かった．そこで開発期間の短縮のために，システムの問題点を早期に見える化する対策を講じた．それが，アプリケーション開発とプラットフォーム構築を並行に(スパイラルに)行う開発手法である．[2]

(5)高品質なシステム開発は，単なるバグつぶしではなく，その真の原因の本質を把握したバグ分析が重要となり，バグの本質の見える化が重要となる.

以降，主に見える化に着目し各課題とその対策について述べる.

3. オープン製品の組み合わせ開発

障害の種類としては，オープン製品の組み合わせに起因する障害があり，製品自身の不具合，製品間のすり合わせの不具合など，予期せぬ障害発生のリスクが大きい．また，大規模システムに起因する障害としては，その規模感から拡張性，性能，負荷などが限界値に達しやすく，潜在バグが顕在化しやすい．さらに，発生した障害は，ミッションクリティカルなシステムであるため，メインフレーム並の保守体制が必要となる．1990 年代前半のクライアント・サーバシステムやミニコンレベルの UNIX では，原因分析に数日以上を要し，メインフレームで実現していたような基幹システムを構築できる状況ではなかった．ここでいうメインフレーム並とは，具体的には，1 時間以内の業務復旧と 24 時間以内の原因究明を指す.

これらの課題に対する対策として，次の取り組みを行った.

メインフレーム並の早期問題解決のため，製品提供ベンダーとの連携により問題・課題の早期解決を図る仕組みを確立した．具体的には海外ベンダーとアライアンスを強化したマルチベンダー共同体によるスーパーHA(High Availability)サポート体制を確立し, この体制を MCATSS(Mission Critical Alliance Team for System Support)と命名した．MCATSS 各社(HP，Oracle，EMC，CISCO，Microsoft 等)と連携し，問題発生と同時に MCATSS 各社が一斉に解決に向けた行動をとることで早期解決を可能とした.

また，ベンダー内に，OMCS の業務アプリケーションを動かすテストリングと呼ばれるテスト環境を構築し，プロジェクトから実際開発中の業務アプリケーションを提供して，問題解決に役立てた.

(1) **マルチベンダーサポート共同体「MCATSS」**

MCATSS は，情報共有基盤，リモート解析基盤，MCATSS 特別連携フォーメーションによって，マルチベンダーを含むトータルサポートとして，以下を実現した.

① 強固なダイレクトパス
ISV(Independent Software Vendor)/IHV(Independent Hardware Vendor)本社と直結した対応として，国内拠点だけでなく米国からもログイン(客先了承時).

② パラレルオペレーション
問題発生時に NEC と ISV/IHV 各社が一斉に行動し，エスカレーションを迅速化.

③ 境界問題への効果的な対応
特定ベンダーの問題とはいえないベンダー間のインタフェース問題などに対して，NEC がコーディネート力を発揮して対応.

④ 整合のとれたプロアクティブサポート
システムの最新情報(構成情報など)をベースとした対応.

(2) **MCATSS 連携サービスの運用**

Vol.2012-GN-82 No.17
Vol.2012-CDS-3 No.17
2012/1/20

MCATSS 各社と連携したサービスを円滑に運用するための「リモート監視/診断システム」と「リモートオペレーション解析システム」により，サービス提供組織間でのリアルタイムな情報共有を可能とした(図 1).

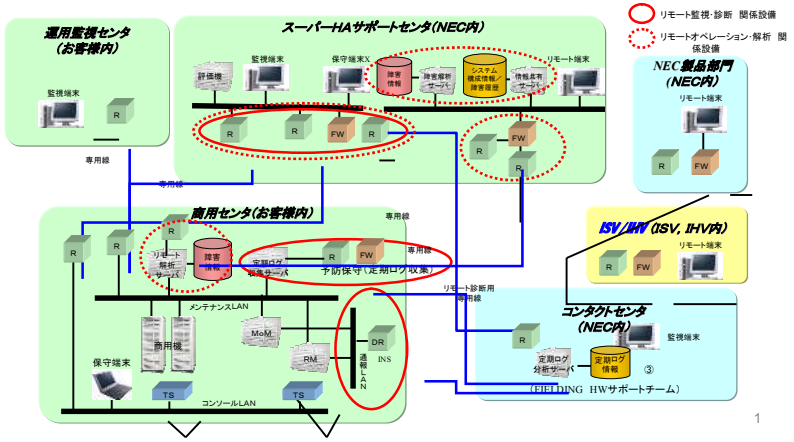

図 1 MCATSS 連携サービスの運用

(3) 対策の成果と活用実績

MCATSS を活用し，バグの解決，性能目標実現が早期化できた．最大の成果は，世界最大規模の超ミッション・クリティカルシステムであるインターネットサービスプロバイダシステムを 1 年という短期間で構築したことである．このシステム構築にあたり，ISV/IHV の米国本社との戦略的パートナーシップにより，メインフレームや自社製品と同等のレベルでトラブル処置を迅速化するとともに，100%処置を実施した．

- 米国の各ベンダーの R&D 部隊から客先の試験機へのリモートアクセスによるトラブル処置の迅速化を実施．
- 重大トラブル発生時は，客先常駐しているプロジェクトの基盤グループから米国本社にダイレクトエスカレーションを実施．
- 米国 HP には技術者 30 人，サーバ 44 台，ストレージ 2 台の大規模テストリングを構築し，性能の徹底チューニング，パッチの QA 強化等を実施．
- 全トラブル 125 件(バグ修正:94 件，性能改善:15 件，機能改善:16 件)を処置．

以上，オープン製品組み合わせ型 SI の課題対策として，MCATSS と呼ばれる新しく強固なベンダーとの関係構築とテストリングで業務アプリケーションをベンダーに持ち込み評価可能とする新しい手法を提案した．この手法により，多様なベンダーの多様な製品の組み合わせたシステムで発生する課題の早期問題抽出，短期問題解決の実績を示した．

4. 分散開発

1999 年に，日本，米国，インド，中国の 4 極体制で超ミッションクリティカル性を有するフルオブジェクト指向の BankingWeb21(地銀勘定系パッケージ)(規模 9MS)[1]の開発を実施した．当時日本では，まだオフショア開発の立ち上げ時期であり，このようなフルオブジェクト指向での大規模なパッケージ開発は，他に事例なく世界的に最先端のプロジェクトであり，次のようなグローバル分散開発体制をとっていた．

東京	業務仕様，業務間結合試験
San Jose(米国)	オブジェクト設計(Business Object，Common Object)
Chennai(インド)	業務実装(機能仕様，製造，業務内試験)
北京(中国)	業務実装(機能仕様，製造，業務内試験)

このようなグローバル分散開発の課題・リスクとしては，以下があげられる．

- 大規模開発でピーク時 1000 人超の多国籍要員での開発推進．
- 国別に工程を分担するため，工程毎の進捗・品質が見えなくなる．
- 海外の場合，言語，時差，距離等から，状況の把握が難しい．

メインフレーム時代は，ウォータフォール型開発で担当者の大部分が全工程へ関与していたが，分散開発になると，部品化技術の進歩で工程毎の分散担当が可能になってきた．さらに，グローバル開発では国別に工程を分担し，半導体のような厳格な工程管理が必須となる．そこで，1999 年当時では稀な海外分散開発において，開発ステージ毎に作業分担して，成果物の品質・納期を見える化し，責任分担を明確にするクリーンルーム方式を徹底，推進した．

開発ステージ毎に成果物の提供側は，成果物に対する品質を保証し，受入側は品質保証と成果物の内容を確認し不明点があれば，提供側へ差し戻す．一方，一旦受け入れた受入側は，受け入れた責任を持ち，自分の責任(特に費用面)で仕事を完遂しなければならない．以下にクリーンルーム方式を適用したソフトウェア開発と大規模工事について述べる．

(1) クリーンルーム方式によるソフトウェア開発

開発ステージ毎での責任範囲を明確化することにより，早期に問題を発見，解決できる(図 2).

- 分散したサイトでのクリーンルーム方式による開発，ステージ別開発により開発プロセス単位での品質保証と進捗の管理/運用技術を実現．
- 工程および OUTPUT 定義と受入側責任方式の明確化が可能となり早期に問題を明確化．

クリーンルーム方式の効果としては，第 1 に，開発ステージ毎に異なる会社で開発を行う形態において，その責任分担を明確にし，品質保証・フィードバックの管理が行いやすくなる．第 2 に，INPUT-OUTPUT を明確に定義し，各ステージ毎の受け入れ

Vol.2012-GN-82 No.17
Vol.2012-CDS-3 No.17
2012/1/20

チェックを相互に行うことにより，責任範囲が明確になる．

このような工程分離が，すべてきれいに実現されたわけではないが，超ミッション・クリティカルな世界初のフルオープン勘定系パッケージ(9MStep)が開発できたことは，本方式の成果である．

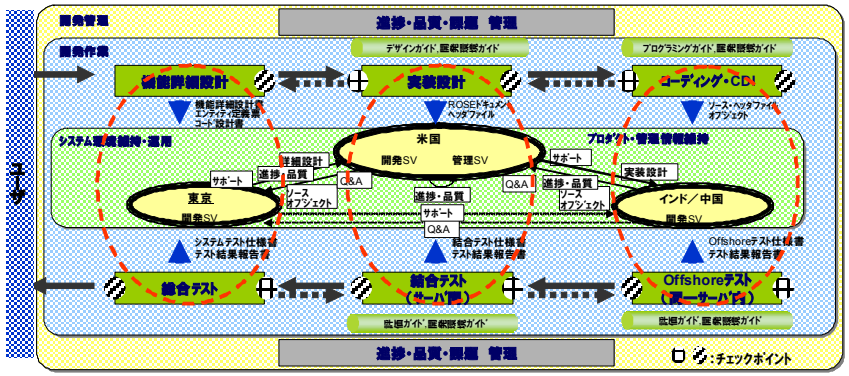

図 2 クリーンルーム方式による開発例(BankingWeb21(地銀向け勘定系パッケージ))

(2) クリーンルーム方式による大規模工事

ここでは，表 1 のような大規模工事におけるクリーンルーム方式の適用について述べる．その総ケーブル長が地球シミュレータ[b]より長い大規模 IT システムの工事では，以下のような困難を伴う．

- サーバ数百台，ネットワーク機器数百台の大規模工事のため，工事ミスは最低でも週単位の大幅な手戻りに直結する．
- 手配物量が多岐に亘り，しかも大量なため，手配ミスや納入部品の故障を見込んでおく必要がある．
- 運送業者まで含め関わり合う関係者が多いため，相互の認識の違いや作業漏れが発生する．
- 工事は短期間で行われるため，十分なリハーサルを行うことができず，失敗した場合のシステム構築への影響が大きい．

このような困難に対して，実際のハードウェアやソフトウェアの製品を設置する工事のスタイルについても，システム構築と同等以上の見える化をコンセプトとした工事技法の確立が重要となる．大規模工事を成功させるために，工事の作業フェーズに応じて以下のような見える化を行った．

b 海洋研究開発機構（JAMSTEC)に設置されている NEC 製の SX-9/E をベースマシンとしたスーパーコンピュータ

① 工事体制確立および計画策定時

- 関係部門，関係会社を集めて工事体制を確立．
- ISV/IHV を含めた工事推進体制で，構築時のトラブル発生数の削減，発生時の迅速な対応．
- 工事案件を統一的に透視できる見える化計画表で，関係者の意識を統一．
- 作業工程をブレークダウンし，作業内容の見える化と分担を関係者と調整し合意．

② 設計および施工時

- 物理的な工事が伴うため，ソフトウェア中心の SI に比べて，手戻り工数が極めて大きいため，工事完了条件を事前に定義し，各工程の担当が完了条件をクリアして引き継ぎ，問題を後工程に持ち込まないことで品質・納期を確保する．

表 1 大規模インターネットサービスプロバイダシステム[1]のシステム規模の例

項目	規模
サーバ	約 400 台
ストレージ	約 400TB
ネットワーク機器	約 500 台
ケーブル	約 10000 本
搬入時のトラック	10t 車約 75 台

具体的には，図 3 に示すように HW 工事部隊の完了条件，PF 構築部隊の完了条件を明確にして，クリーンルーム方式の工事を実施した．

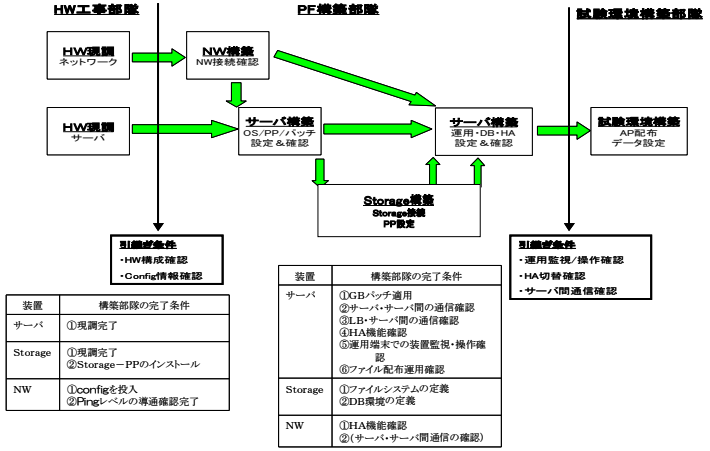

図 3 クリーンルーム方式による工事例

リスク対策の具体例としては，通信ケーブルの 2 重敷設を防止するための配線表チ

Vol.2012-GN-82 No.17
Vol.2012-CDS-3 No.17
2012/1/20

ェックツールを用意し，サーバ，ストレージ，ネットワーク機器の接続性確認(MACレイヤ[c]，ネットワークレイヤ，アプリケーションレイヤ)を半自動化して，確認漏れを防いだ．また，輸送上の事故を想定し，正系統と副系統の機器を同一トラックに載せないこととした．さらに，現調期間中の故障対応のための予備装置の手配を行った．コンテンジェンシープランとして，ソフトウェア開発の遅延リスクを想定し，商用機器の一部を試験機へ転用する準備を行った．

これらの対策により表 1 の大規模インターネットサービスプロバイダシステム工事を予定通り設置工事 1.5 ヶ月，工事確認 1 ヶ月の短期間で後戻りなく完遂した．

5. 大規模開発

数メガ～十メガ以上の大規模開発では，その規模と複雑さから，進捗や品質が見えにくくなるため，リアルタイムな進捗の見える化として，次の取り組みを行った．

(1) 情報共有

それまでは，プロジェクトの進捗状況の共有は，SIer 内部に閉じていたが，すべての情報を一元化して，ユーザ，パートナー，ベンダー3 者間で共有した．さらに，プロジェクト責任者，チーム責任者，担当のすべてのレベルで，課題・障害の見える化を図り，進捗・品質の認識を一致させることで，問題解決の協調体制を築いた．

(2) リアルタイム管理システム

インターネットを使った Web 型の管理ツールにより，リアルタイムに収集した進捗・品質・費用情報から，計画(進捗・品質・費用モデル)との差異を管理した(図 4)．

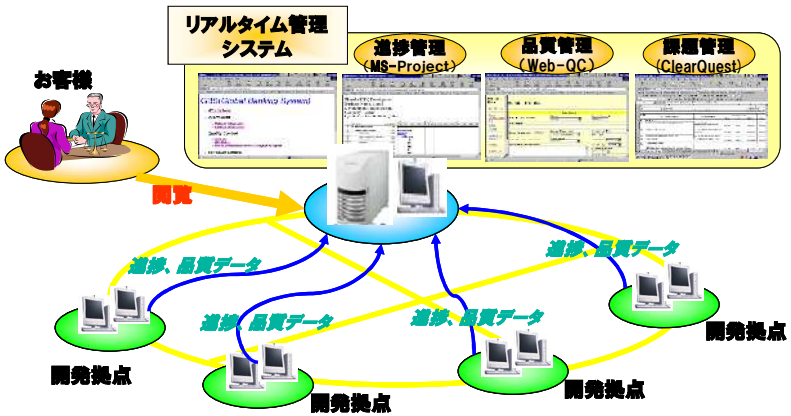

図 4 Web 型リアルタイム管理システム

(3) パート図の活用

複雑なシステムでは，パート図を活用し，クリティカルパスを中心とした進捗を管理した．従来の総合試験工程は，ガントチャートを使用することが多い．これは，各作業の開始時期，終了時期が把握しやすく，作業管理者にとって有効な進捗管理方法であるが，ある時点でのクリティカルパスとなる作業工程が明示されず，作業の重みが分からない問題がある．そこで，パート図を利用して進捗状況とリスクの認識を関係者全員で共有することで，その時点のクリティカルパスを中心とした確認，対策を実施した．

(4) 対策の成果

ここで述べた対策により，情報が一元化したため，課題・問題の優先度の決定等，プロジェクトの意思決定・方向性の統一に役立った．従来手法では，定期的に行われるの進捗・品質会議の時まで問題点が見えなかったが，リアルタイムの情報収集で，問題の早期発見と対策が早く打てた．大規模プロジェクトになればなるほど，各種トラブルに対する即断即決が重要であり，このような情報の一元化が即断即決に結びついたといえる．また，クリティカルパスを中心にし，全員で進捗を共有することにより，計画遵守の確度が高まった．

以上，大規模開発の課題対策として，情報共有，リアルタイム管理システム，パート図の活用について述べた．顧客との情報共有は，一般的には行われていない新しい手法であり，クリティカルパスの共有のためパート図の活用は，それまでにない新しい領域の適用といえる．そして，日本国内ではもとより，世界的にも例を見えない大規模プロジェクトが成功したことが最大の成果である．

6. スピード開発

大規模なオープンシステムの開発は，システム構築の自由度がないメインフレーム開発と異なり，自由度があるが故，アプリケーション群を開発するプロセスと，アプリケーション群を動作させるシステムプラットフォーム構築のプロセスの両方が必要となり，以下のような種々な問題が発生した．

- アプリケーションの評価環境の構築が遅れ，アプリケーション開発が遅延．
- アプリケーション群とシステムプラットフォームの整合がとれず，誤動作・性能問題の発覚が遅れ，納期に致命的な影響．
  1990 年代前半，米国においても，サーバを数百台使用する大規模分散システムでは，実質的に開発停止や大幅遅延に追い込まれることがあった．

これらの対策として，スパイラルに同時にアプリケーションとプラットフォームを開発，構築していくことにより，システムの最小単位から業務システムとしての評価を可能とするスパイラル開発を実践した．詳細は，参考文献[2]を参照．

c Media Access Control レイヤ OSI 参照モデルではデータリンク層の下位副層に相当

Vol.2012-GN-82 No.17
Vol.2012-CDS-3 No.17
2012/1/20

(1) スパイラル開発の成果

スパイラル開発は，システムの最小単位(1/n モデル)での業務システムの評価が可能になることにより構築，作り込み時点でのバグや不整合を最小限に抑えることが可能となり業務システムとしての保証が早い段階で実証できる．また，最小機能構成で業務単体性能から検証して，評価モデルに応じた性能評価を実施可能となる．

スパイラル開発により，メインフレームでは通常 3〜4 年の期間で構築していた基幹システムを，UNIX サーバ数百台のオープンシステムであっても 1〜2 年と大幅に短縮して構築した．

7. 品質の作り込による高品質なソフトウェア開発

OMCS は，クライアント・サーバなどの一般のオープンシステムに比べ数桁上の品質が要求される．従ってプロジェクトで開発されるソフトウェアも通常のソフトウェアより 1 桁以上の高品質が要求される．例えば，地銀向けバンキングシステムに代表される高信頼性モデルで利用されたオープン勘定系パッケージ BankingWeb21[1]の開発において，競合他社が途中で撤退したことからも大規模なパッケージを高品質で開発するのは，かなり難度が高いことが伺える．

OMCS のソフトウェア開発の留意点は，やるべきことをきちんとやり抜くことに尽き，いかに品質を作り込むかにかかっている．

- 今までに経験のない高品質の開発を行う
  高い品質を保証するためには，高いレベルの開発標準が必要．経験のない中で必要なレベルの開発標準を用意しなければならないので，開発中に随時，チューニングし，開発標準にフィードバックする必要がある．
- ロスコン(失敗コスト)の極小化
  とにかく後戻り工数を減らすことが重要．開発量も大きく，短期開発のため，後戻りが発生すると，リカバリのため多大な費用(受注額の 2〜3 倍以上)が発生し，ビジネスが成り立たなくなる．NEC においても，それまでに大規模オープンシステム構築において，2 桁億におよぶロスコンを発生させていた．
- 多くの人間で開発すること
  誰か一人でもやるべきことを手抜きさせないことが大切．やるべきことをきちんとやったかチェックできることが必須

本節では，高品質なソフトウェア開発のために，いかに品質を作り込むかに向けて実践している分析手法と，現在全社規模で展開されている活動について述べる．

(1) プロセスベースの品質保証となぜ 3 分析

プロセスベースの品質保証とは，開発プロセスの標準を定めた後，開発中に摘出した問題(バグ)について，なぜその問題が作り込まれ，今まで摘出できなかったのかをプロセスベースに原因追及し，プロセス改善と横展開を各工程完了までに行うことである．このプロセスベースの原因追及の仕方をなぜ 3 分析と呼び，次の 3 段階の分析を実施した．

- なぜ 1:直接の原因．
- なぜ 2:なぜテストやレビューで見つけられないか．
- なぜ 3:なぜ管理，組織として見つけられないか．

チームリーダが担当者にヒアリングして原因を追及し，再発防止のプロセス改善と類似バグ摘出の横展開を行う．ヒアリングの観点を以下に示す．

- 設計書の書き方
- 開発プロセス
- レビュープロセス
- テスト項目レビューやテストプロセス

なぜ 3 分析は工程内で完結させるためスピードが重要となる．

(2) なぜ 3 分析の全社展開活動

OMCS のソフトウェア開発に端を発するなぜ 3 分析は，その適用領域を SI 以外にも拡大し，全社の活動として現在も展開している．活動の流れを以下に示す(図 5).

- 社内トップレベルまで報告のあがる重要品質問題について，すべてなぜ 3 分析を実施する．
- 真の原因となった課題について，組織的な改善課題として取り組む．
- 企画本部・事業本部が運営/管理し，APPEAL 推進会議(社内 SI 革新推進活動のひとつ)で横展開する．

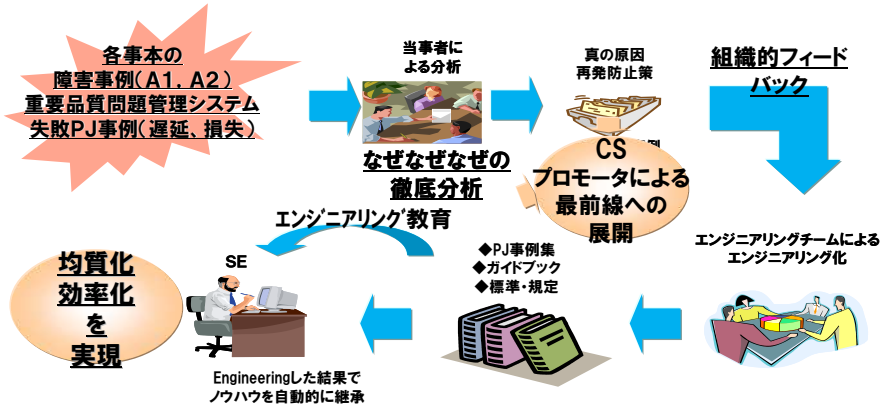

図 5 なぜ 3 分析の全社展開活動

Vol.2012-GN-82 No.17
Vol.2012-CDS-3 No.17
2012/1/20

(3) 対策の実績

このような活動により，大規模サービスプラットフォームモデル・超並列モデルを適用したインターネットサービスプロバイダシステム[1]で，稼動率 99.9999%のシステムを実現した．

社内 SI 革新推進活動の一環として行われた「なぜ 3 分析検討会」は，8 年間で，33 回，のべ出席者数は約 3000 人におよんでいる．

8. 大規模システムのプロジェクト管理の特徴と見える化のポイント

OMCS が対象とする大規模な基幹システム開発で実施した「見える化」に着目したプロジェクト管理の特徴をまとめると表 2 の様になる．

表 2：見える化のポイントのまとめ

開発の特徴	課題	見える化の因子	対策	ねらい
オープン製品利用	障害の切り分け	MF相当の障害対応の見える化	MCATSS	製品の整合性確保とバグ解決の早期化
			テストリング	性能目標実現が早期化
グローバル開発	言語、時差、距離の違い	工程ごとのSIプロセス見える化	クリーンルーム方式	各ステージのINPUT-OUTPUTを明確に定義し、責任範囲を明確化
大規模開発	納期厳守	進捗の見える化	リアルタイム管理	パートナ、ベンダー、ユーザが同一情報の下で、課題の状況をリアルタイムに明確化
			パート図の活用	クリティカルパスを中心に進捗管理
	品質厳守	品質問題の原因の本質の見える化	なぜ3	バグ内容を見える化して共有し同一傾向の潜在バグ摘出と再発防止
スピード開発	分業・並行開発	システム上の問題点を早期発見	スパイラル開発	業務システムとしての機能/性能を早い段階で検証

これらの施策の導入前 2 年間と導入後 2 年間で失敗コストは約 20%削減されている．

9. BPM における見える化の推進

OMCS が対象とするシステムは，企業の基幹システムの再構築である．新しい基幹システムが企業活動に真に役立つためには，ビジネスプロセスの見直しが必須である．

本章では，NEC を対象としてビジネスプロセスリエンジニアリング(BPR)へのメソドロジーを実践した成果をまとめた．基本コンセプトは，BPR へ参加する組織，個人へ BPR そのものの内容をいかに見える化し賛同を得るかである．SAP ベースでは世界最大規模となる NEC の基幹システム構築プロジェクト(G1:Global One)[1]において，プラットフォーム構築，アプリケーション開発からさらに，財務を含む BPR を行っているが，見える化の対象として BPR までスコープの拡大を行った(図 6)．

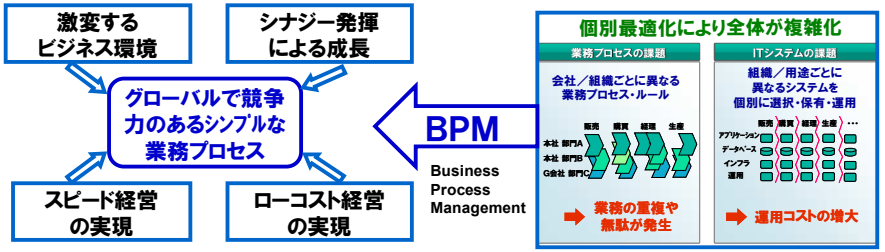

図 6 業務プロセス/IT 改革の狙い

標準化をテコにした改革活動を円滑かつ着実に推進するために，以下のような見える化が有効である(図 7)．

(1) 目標と成果の見える化
- ●改革目標/改革方針の明示と経営トップ層からの定期的なメッセージ発信
- ●主要改革指標の設定(業務効率化，TCO(Total Cost of Ownership)削減，リスクコントロール数削減，決算日程短縮等)

(2) 体制と課題の見える化
- ●関連要素を一体推進する標準化活動(制度ルール，業務プロセス/データ/システムの標準化)
- ●標準プロセスの設計と定着化に関わる責任と権限を持つプロセスオーナの任命
- ●関連スタッフ・事業ラインが参画するプロセスオーナ会議で改革方針と課題を共有/審議
- ●IT システムの標準化やコード・データの標準化についてガバナンス体制を構築

(3) 計画と進捗の見える化
- ●BPM 推進手順，設計物の検証手順，標準への移行手順，推進計画と進捗等を見える化

Vol.2012-GN-82 No.17
Vol.2012-CDS-3 No.17
2012/1/20

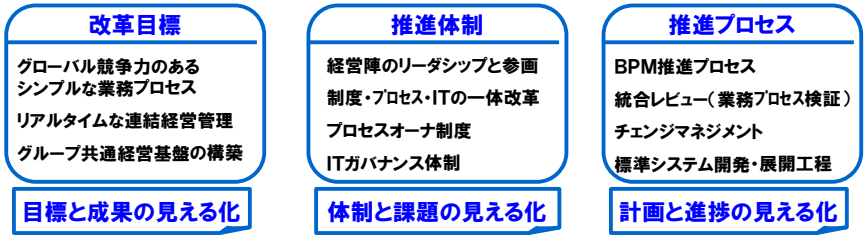

図 7 業務プロセス改革活動と標準化活動の見える化

以上，BPM の見える化について述べた．具体的な実績としては，この業務プロセス・IT 改革は，当初の予定を半年前倒しで，実質 2 年で本番実施できたことがあげられる．また，業務プロセスシンプル化の達成指標のひとつである内部統制リスクコントロールポイント数の実際の変化を見ると，標準プロセスで内部統制の補強を行ったにもかかわらず，NEC 本体単独で改革前に比べて 37%のコントロール数の削減ができ，文書化工数や監査工数の削減につながっている．

しかも，BPR のパートナーIDS Scheer(現 SAG)社の評価では欧米の同業種の大企業に比べ 1/2 の期間で構築を完了し，2010 年ドイツで開催された Process World2010[d]にて Business Process Excellence Award を受賞している．

10. まとめ

およそ 15 年に渡る OMCS のシステム構築を見える化をコンセプトとしたプロジェクト管理の観点から捉え，OMCS 開発の特徴である①オープン製品の組み合わせによる開発，②分散開発，③大規模開発，④スピード開発，⑤高品質という特徴ゆえに見えなくなっている課題を抽出し，その対策として構築実績に基づく見える化について述べてきた．それまでもプロジェクト管理および「見える化」の観点で種々の取り組みが実行されているが，本研究の様に大規模なオープン・システム構築という環境下で(単なるコンセプトではなく)見える化の対象を明確にし，実践されているものは発表されていない．言い方を変えるとそれまで大規模システムはメインフレームによるプロプライエタリな(閉じた)世界で実現されており，オープンな世界で実現されたのは世界的に見ても先進的な取り組みと考えられる．

OMCS として構築してきたシステムは，メインフレーム以上のミッションクリティカル性に富む大規模基幹システムであり，日本国内はもとより，世界的に見ても先進的な事例であり，現在も安定稼動の実績を有している．事例は，参考文献[1]に詳しい．このことは，OMCS 構築技術の有効性を示しており，見える化に着目したプロジェクト管理なくして，実現できなかったと確信している．

また，これらの成果は，NEC 社内は勿論，協力会社を含む NEC グループ数万人の SE の SI 標準プロセス(APPEAL)として活用している．

今後は，PMBOK，CMMI などの世界標準を視野に入れながら，我々の OMCS 構築技術の取り組みについてプロジェクトマネジメント学会などでの活動を通して広めてゆく．

謝辞 永きに渡り，このような数々のシステム構築の場を与えてくれた顧客およびパートナー各社の皆様，ならびに本論文をまとめるにあたり討論いただいた社内外の皆様に感謝申し上げる．特に，本論文で述べた内容は，延べ万のオーダーに登る大勢の人たちの参加と努力なくして実現できなかったものであり，一人一人の名前をあげることはできないが，関連した皆様には心から感謝申し上げる．

参考文献

1) 相澤正俊 他:OMCS 構築技術の研究
2) 相澤正俊 他:OMCS のシステムモデルによるプラットフォーム構築

商標について

d 世界最大規模の BMP 関連国際大会．