

ITモダナイゼーションの成功事例と 最新の要素技術

2016/04/26

株式会社 日立製作所 ICT事業統括本部
アプリケーションサービス事業部
サービスソリューション本部 サービス統括部

新家 博文

Contents

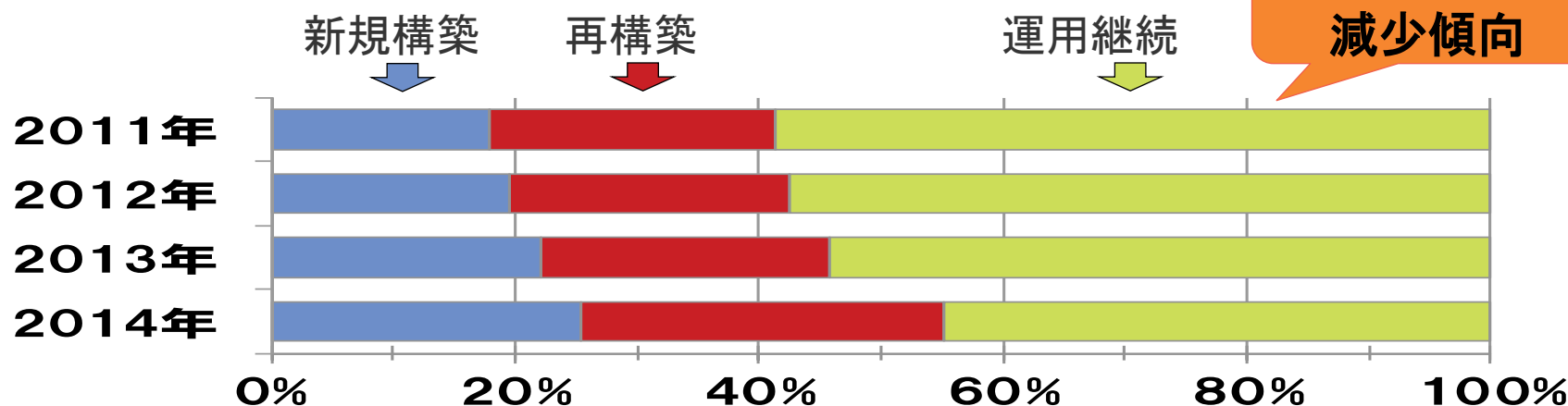
1. はじめに
2. 従来技術の成熟
～ リホスト、リライト
3. 移行をサポートする技術
～ リファクター、リドキュメント
4. 最新の提供サービスと要素技術
～ リビルドとこれからの技術
5. まとめ

1. はじめに

1-1. 企業情報システムを取り巻く状況

- 企業が勝ち残っていくためには
「戦略性の高いシステムへの投資を続けていかなければならない」

情報システムの取り組み状況 件数ベース



出典: 経済産業省 平成26年情報処理実態調査結果報告書 情報システムの取り組み状況

システム再構築への投資(取組み)が増加

- 戦略的なシステム投資として新規構築のみならず、再構築案件は拡大しており、変化に対応できる再構築が重要性を増している。

● 既存システムの課題



ビジネス環境や法制度対応など、
変化に対応できる強いIT基盤が必要

システム移行のための技術群「モダナイゼーション」で
これらの課題に対応

リファクター

機能をそのままに現行のコードを整理する

性能や保守性を向上させる為に、現行アプリケーションの設計や実装を見直すことです。マイグレーションの文脈においては、必要な資産の範囲を見極める「棚卸」が重要です。

リドキュメント

現行システムを可視化、ドキュメントを再整備

現行システムの資産分析やヒヤリングを通じて、現行システムの仕様、設計情報を可視化し、ドキュメントを整備する技術です。

リホスト

既存システム資産をそのまま利用する方式

互換性のあるミドルウェアやエミュレーションを利用して既存アプリケーションをそのまま利用する方式です。移行費用を押さえつつ短期にインフラを刷新したい要望に応えることができます。

リライト

変換ツールでコードを他言語に書き換える方式

既存のプログラムを変換ツールで他言語に書き換えて、移行を実施する方式です。開発・保守の環境含めを新しい言語に統一したい要望に応えることができます。

リビルド

既存システムの業務仕様を再実装する方式

既存システムの仕様を生かし、再実装する方式です。移行方式としては最も自由度が高く、非機能要件に対し、最適な設計／実装の選択が可能です。最もコストがかかります。

2. 従来技術の成熟 ～ リホスト・リライト

リホスト

既存システム資産をそのまま利用する方式

互換性のあるミドルウェアやエミュレーションを利用して既存アプリケーションをそのまま利用する方式です。移行費用を押さえつつ短期にインフラを刷新したい要望に応えることができます。

リライト

変換ツールでコードを他言語に書き換える方式

既存のプログラムを変換ツールで他言語に書き換えて、移行を実施する方式です。開発・保守の環境含めを新しい言語に統一したい要望に応えることができます。

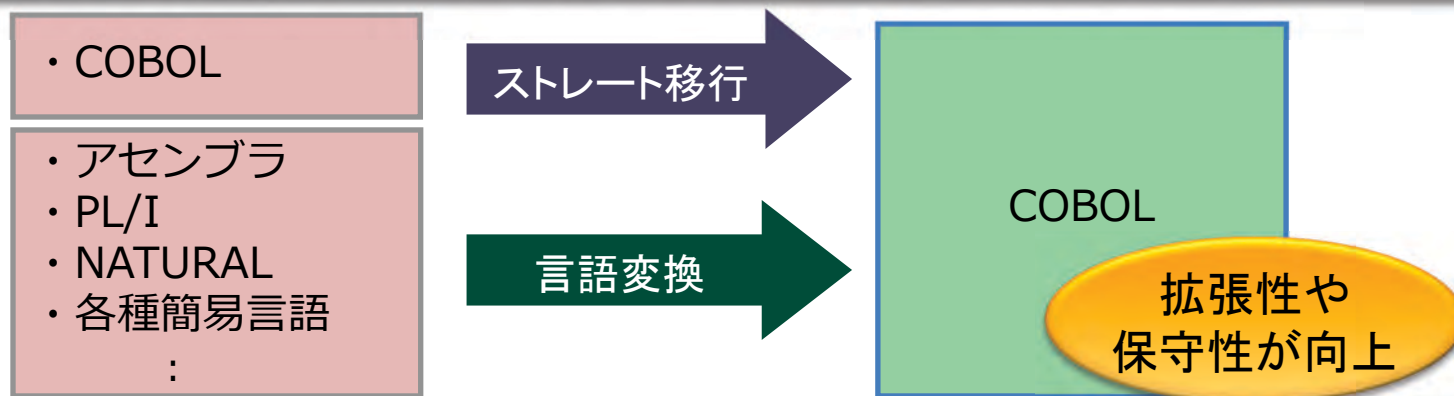
リビルド

既存システムの業務仕様を再実装する方式

既存システムの仕様を生かし、再実装する方式です。移行方式としては最も自由度が高く、非機能要件に対し、最適な設計／実装の選択が可能です。最もコストがかかります。

2-1. メインフレームからの移行 ～COBOLを用いたリHOST・リライト

- 再利用するプログラムは業務システム向きのCOBOLで移行
- COBOL以外で作成されたものは、原則としてCOBOL化



COBOLは世界のほとんどの 商用基幹アプリケーションで稼働

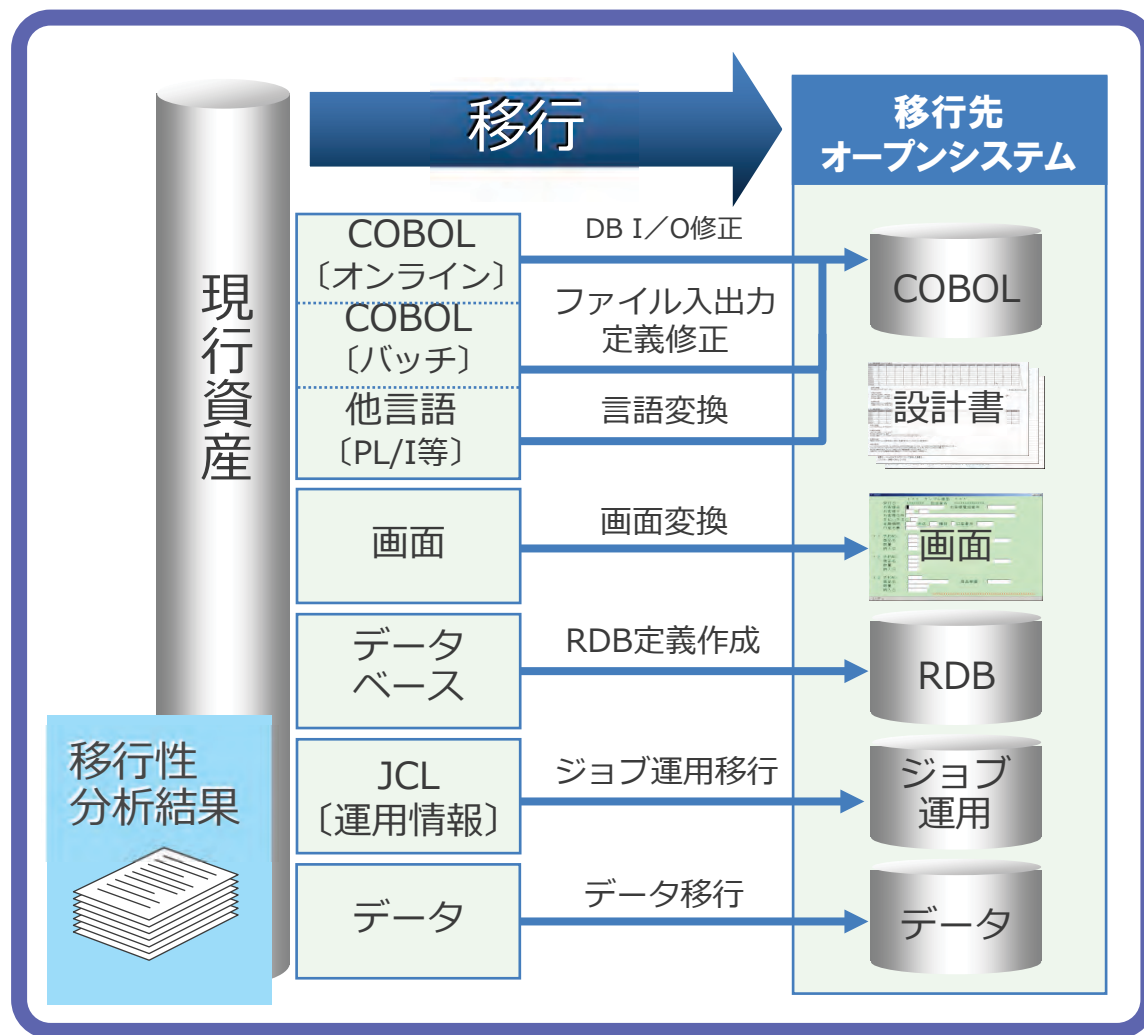
- ・ ビジネス・データの75%
- ・ 金融取引では90%を処理
- ・ 2,000億行のコードと推定

出典：COBOLコンソーシアムホームページ (<http://www.cobol.gr.jp/>)

COBOL誕生50周年記念セミナー ～社会を支える“ことば”。これまでも、そしてこれからも～（2010年4月16日配布資料）
日本アイ・ビー・エム「COBOL開発ライフ・サイクルを刷新するエンタープライズ・モダナイゼーション」でのIBMの推定

2-2. メインフレームからの移行 ～COBOLを用いたリHOST・リライト

● プラットフォームの差異をミドルウェアで吸収



事例の 特徴

- 現行資産の棚卸によりスリム化を実施し、効率的にオープン化
- 確実な移行実現をめざし、2段階で移行検証を実施
 - 第1段階：小規模機能による移行設計の精度確認
 - 第2段階：重要業務・大規模機能の性能確認

全体17.5ヶ月

【課題／対応】

- システム運用費低減のための移行を低コストで実現したい
 - ⇒ COBOLによるリホスト中心の方式により低コストでの移行を実現
オープンシステム移行により運用費低減に成功
- 性能向上の実現によりエンドユーザーのシステム利用満足度をアップしたい
 - ⇒ 性能確保のための移行検証を実施
移行方式のチューニングで性能向上を達成

【資産概要】

- | | |
|----------------------------------|-------------------|
| ● COBOLソース .. 約5,730本(約2.1MStep) | ● JCL 約700本 |
| ● アセンブラ 約10本 | ● DB 約205本 |

2-4. COBOLを用いた移行事例(実績の一部)

- 大規模システムの移行実績あり
- 日立ホスト、他社ホスト含め、200システム以上の移行を実現

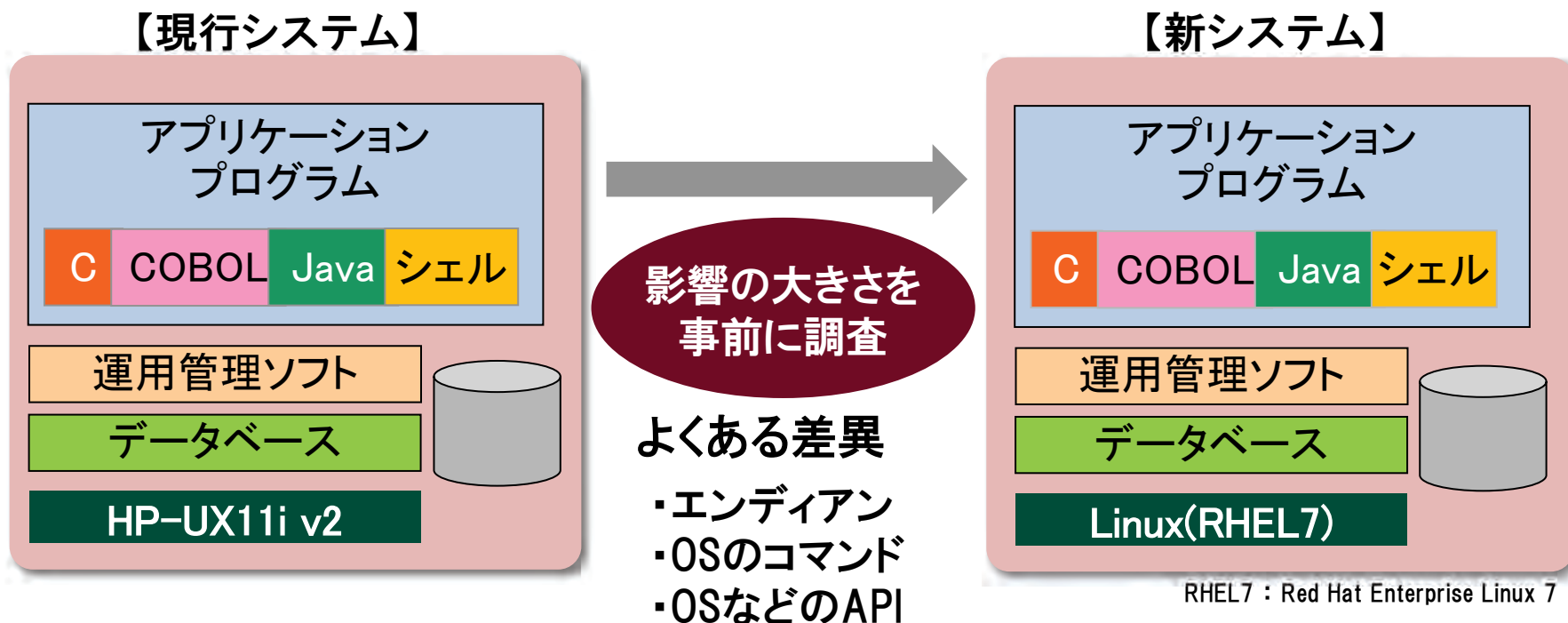
事例	業種	移行元システム	移行先システム	資産規模	期間	効果
A社	流通	日立ホスト VOS3	AIX、HiRDB	COBOL: 約5000本 JCL: 約5000本 DB: 約450本、画面: 約800本	23ヶ月	・処理性能向上 ・コスト低減 ・システム連携
B社	製造	日立ホスト VOS3、ADABAS、 DCCM3、COBOL85、 NATURAL アセンブラ	AIX、OpenTP1、 Oracle、 COBOL2002	COBOL: 約1.9MStep NATURAL: 約460KStep アセンブラ: 約300KStep JCL: 約4000本、 DB: 約450本 画面・帳票: 約600本	24ヶ月	・コスト低減 ・言語統一
C社	産業	X社ホスト MSP、AIM/DB、 AIM/ACS、PSAM、 ADJUST、COBOL85	AIX、HiRDB、 OpenTP1、XMAP3、 HOPSS3、BJEX*、 COBOL2002	COBOL: 約8500本 JCL: 約1200本 DB: 約60本 画面・帳票: 約400本	32ヶ月	・オペレータレス
D社	運輸	Y社ホスト z/OS、NATURAL、 KMAP、IMS/DC、 ADABAS	Red Hat Enterprise Linux、 COBOL2002、JSP、 JBOSS、shell	COBOL: 約71KStep アセンブラ: 約100本 NATURAL: 約40KStep JCL: 約110本 DB: 約55本、画面: 120本	9ヶ月	・低コスト移行 ・処理性能向上
E社	金融	X社ホストOS IV/MSP COBOL85 アセンブラ	Red Hat Enterprise Linux、 BJEX、COBOL2002	COBOL: 約220KStep アセンブラ: 約10本 JCL: 約700本、DB: 約3本	11ヶ月	・部分リホストによる 再構築全体コストの低減

*BJEX: uCosminexus Batch Job Execution Server

2-5. オープンサーバでの移行 ～ リホスト

- ハードウェア/OSの保守切れなどがきっかけで、リホストを検討
商用Unix ⇔ 商用Unix、商用Unix ⇔ Linux、商用Unix ⇔ Windows
など。また、同種OSのバージョンアップ版へのリホストも
- COBOL / Java のプログラム自体は、基本的に互換性あり
- 主に、ハード(CPU)、OS、ミドルウェアなどの差異がプログラムに影響

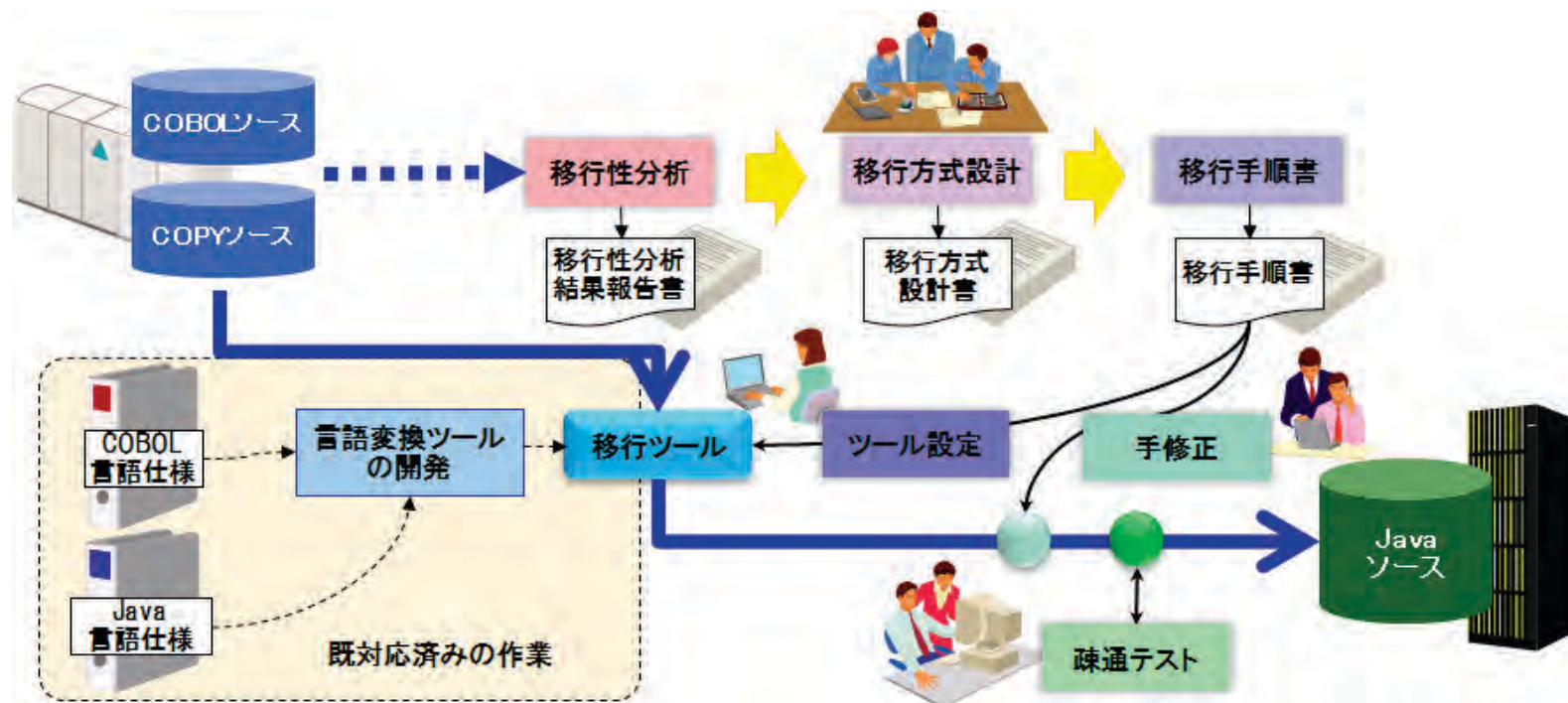
● 別OSサーバへの移行例



● COBOL言語からJava言語への移行を実現

機械変換が可能になり低コスト／短期間での移行が可能！

- 独自ライブラリを用いて、これまで難しかった特殊処理の変換を実現
- お客様の資産特性に合わせた変換仕様を設計し、高い機械変換率を実現
- これまで対応が困難であった UTF-8 に対応した変換方式を実現



事例の 特徴

- 現行システムは稼働から15年程度経過し追加要件は少ない。
新規に開発するプログラムはJavaで組んでおり、COBOLプログラムをJavaに焼き直して言語を統一

移行 4. 5ヶ月

【課題／対応】

- JavaとCOBOLで構築されたシステムを保有する為、両言語の要員を抱え、保守費用が膨らんでいた。
⇒ 移行により言語がJavaに統一され、保守費用の削減を実現
- 当該処理は多くの業務から呼ばれるため、現行性能の維持が必須
⇒ 実証実験を実施してライブラリをチューニングし、性能要件をクリア

【資産概要】

- COBOLソース ・・ 約1,215本(約315KStep)

3. 移行をサポートする技術 ～ リファクター・リドキュメント

リファクター

機能をそのままに現行のコードを修正する

性能や保守性を向上の為に、現行アプリケーションの設計や実装を見直すことです。マイグレーションの文脈においては、必要な資産の範囲を見極める「棚卸」が重要です。

リドキュメント

現行システムを可視化、ドキュメントを再整備

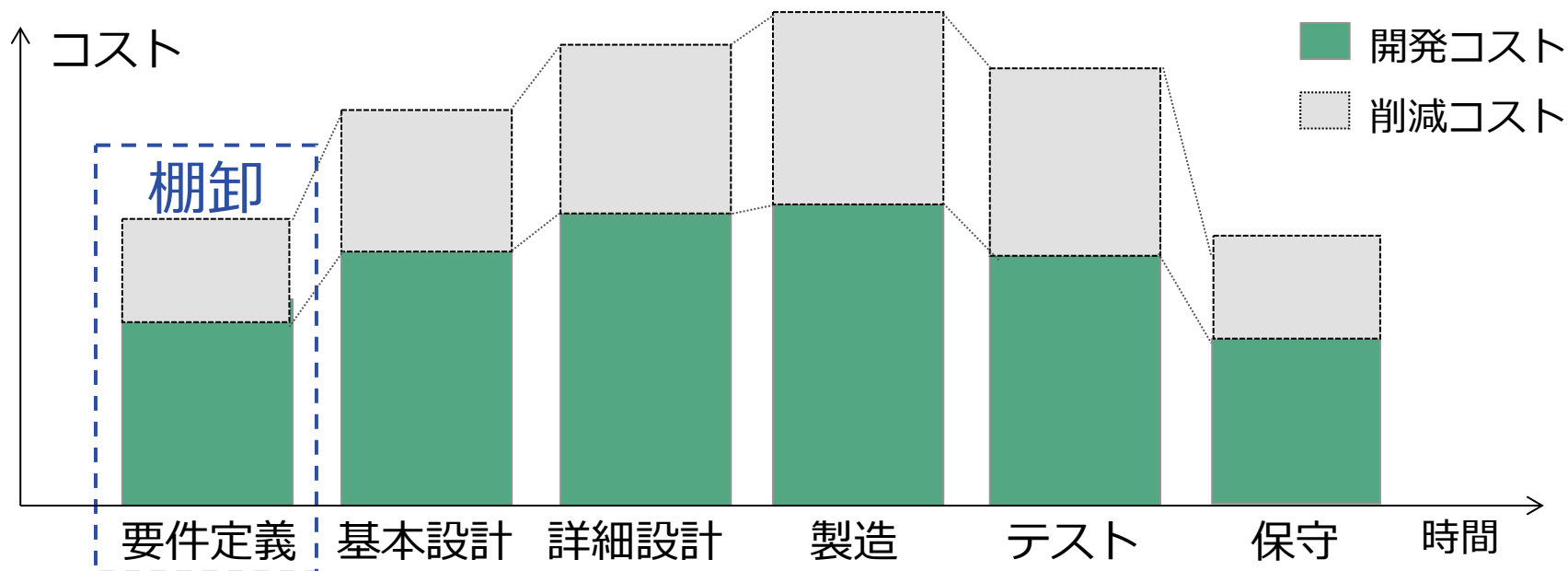
現行システムの資産分析やヒヤリングを通じて、現行システムの仕様、設計情報を可視化し、ドキュメントを整備する技術です。

リファクタリング

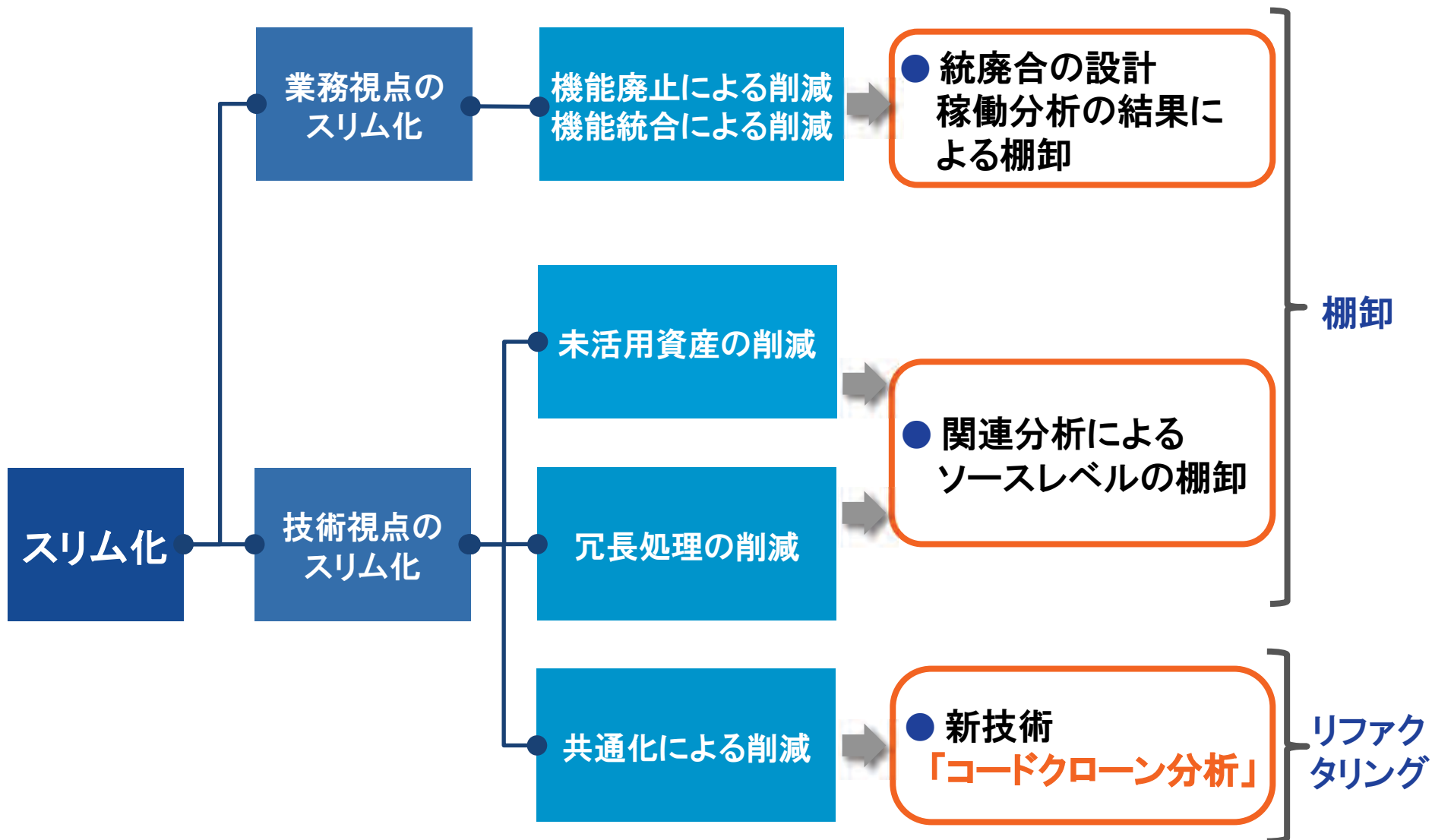
保守性、性能向上の為のプログラム設計、実装レベルの見直し。
一般には、保守フェーズの効率改善の為。
再構築は資産を整理する機会でもある。

棚卸

不要なものを削除する。システム再構築では棚卸が重要。
後続工程(設計～テスト、保守)も含めたプロジェクト全体の
コストを削減できる(下図)。

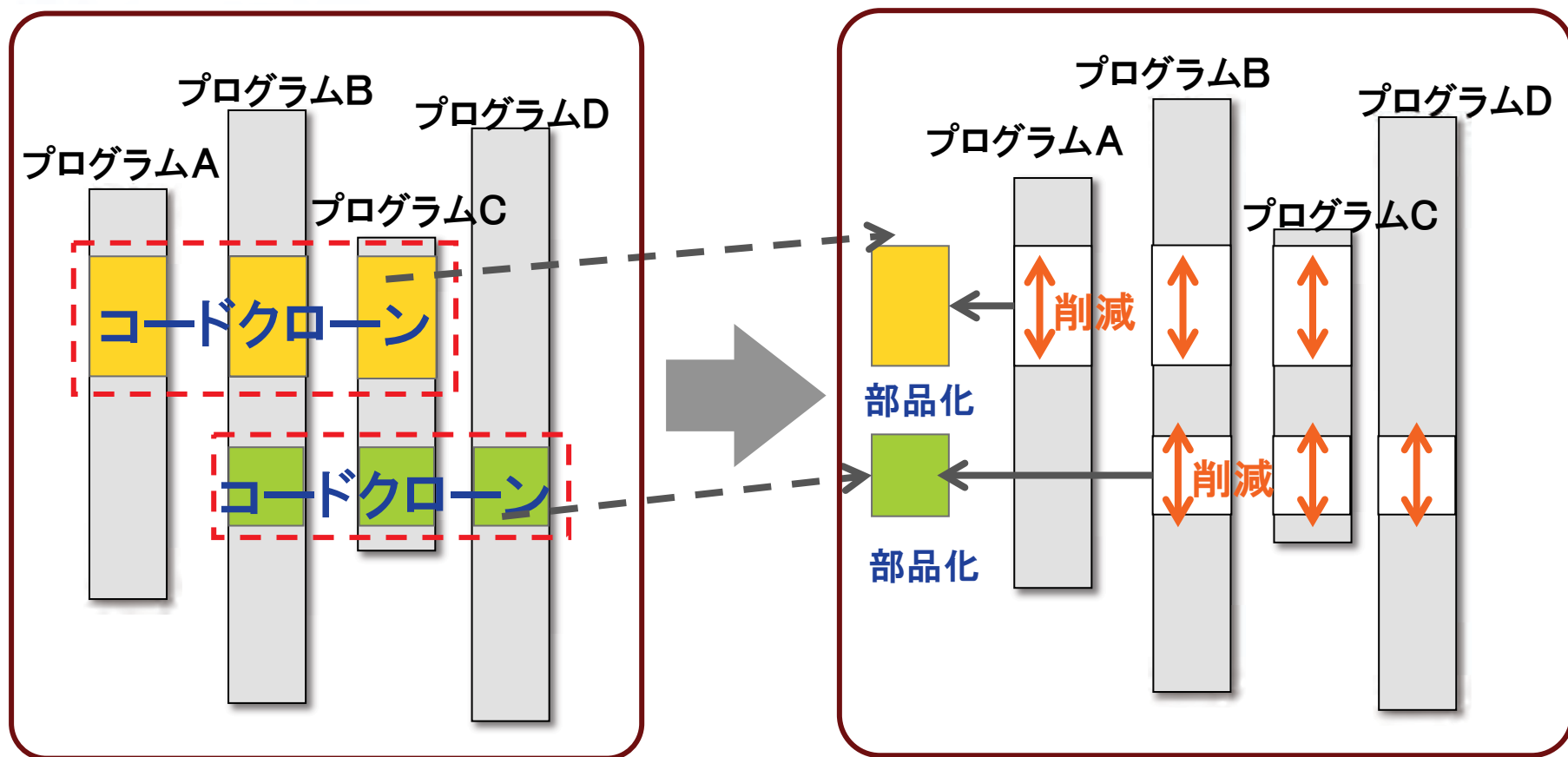


3-2. リファクター領域のサービス：スリム化



3-3. コードクローン分析

- 共通の「語」の並び(コードクローン)を、アプリケーション全体から抽出
- 抽出した共通部を、部品候補(COPYソース)として切り出し、プログラムの規模を削減



事例の 特徴

■ 棚卸による規模削減と、コードクローン分析と共通部の部品化を実施

【課題／対応】

- システム移行費用をできるだけ抑えたい
⇒ 資産棚卸により移行規模を27%削減、移行費用の抑止を実現
- コピー＆ペーストで作成されたデータ定義／処理が多くあり、保守作業の妨げになっていた
⇒ コードクローン技術を使った共通化により4%を部品として切出すことができた。これにより、保守性の向上が期待される



● COBOL(整理前) 約757KStep

● COBOL(棚卸後) 約549KStep

● COBOL(共通化後) 約527KStep

リファクター

機能をそのままに現行のコードを修正する

性能や保守性を向上の為に、現行アプリケーションの設計や実装を見直すことです。マイグレーションの文脈においては、必要な資産の範囲を見極める「棚卸」が重要です。

リドキュメント

現行システムを可視化、ドキュメントを再整備

現行システムの資産分析やヒヤリングを通じて、現行システムの仕様、設計情報を可視化し、ドキュメントを整備する技術です。

移行計画の 検討材料に

計画の検討材料としてのリドキュメント

業務フローやジョブフロー、CRUD図*等、可視化で作成したドキュメントは、移行計画やテスト計画の検討材料になります。

* どの機能がどのテーブルに対しどの操作（レコード生成(C)/参照(R)/更新(U) /削除(D)）をするかをマトリクスで表した図

問題分析 の支援策に

問題分析や影響分析の支援策としてのリドキュメント

テストでの問題発生時に既存の実装を確認する為に、既存のプログラムの可視化結果を利用することができます。問題への対応方式を検討するにあたって影響分析に用いることができます。

再構築の 前提に

再構築の前提としてのリドキュメント

特にリビルド型のマイグレーションでは、ドキュメントが保守されていない場合、必須のタスクとなります。その他の場合でも、ツールで作成した仕様情報を用いることにより作業の効率を上げることができます。

- アプリケーションライフサイクルマネジメントサービス(ALM)では、リドキュメント領域で2つのサービスメニューを準備しています。

プログラム 仕様可視化

資産解析ツールでプログラム資産レベルの情報を可視化

10年以上の実績をもつ資産解析ツールにより、現行システムの資産を解析し、可視化情報、関連情報を作成します。可視化情報は、専用ブラウザで参照できる環境を提供します。

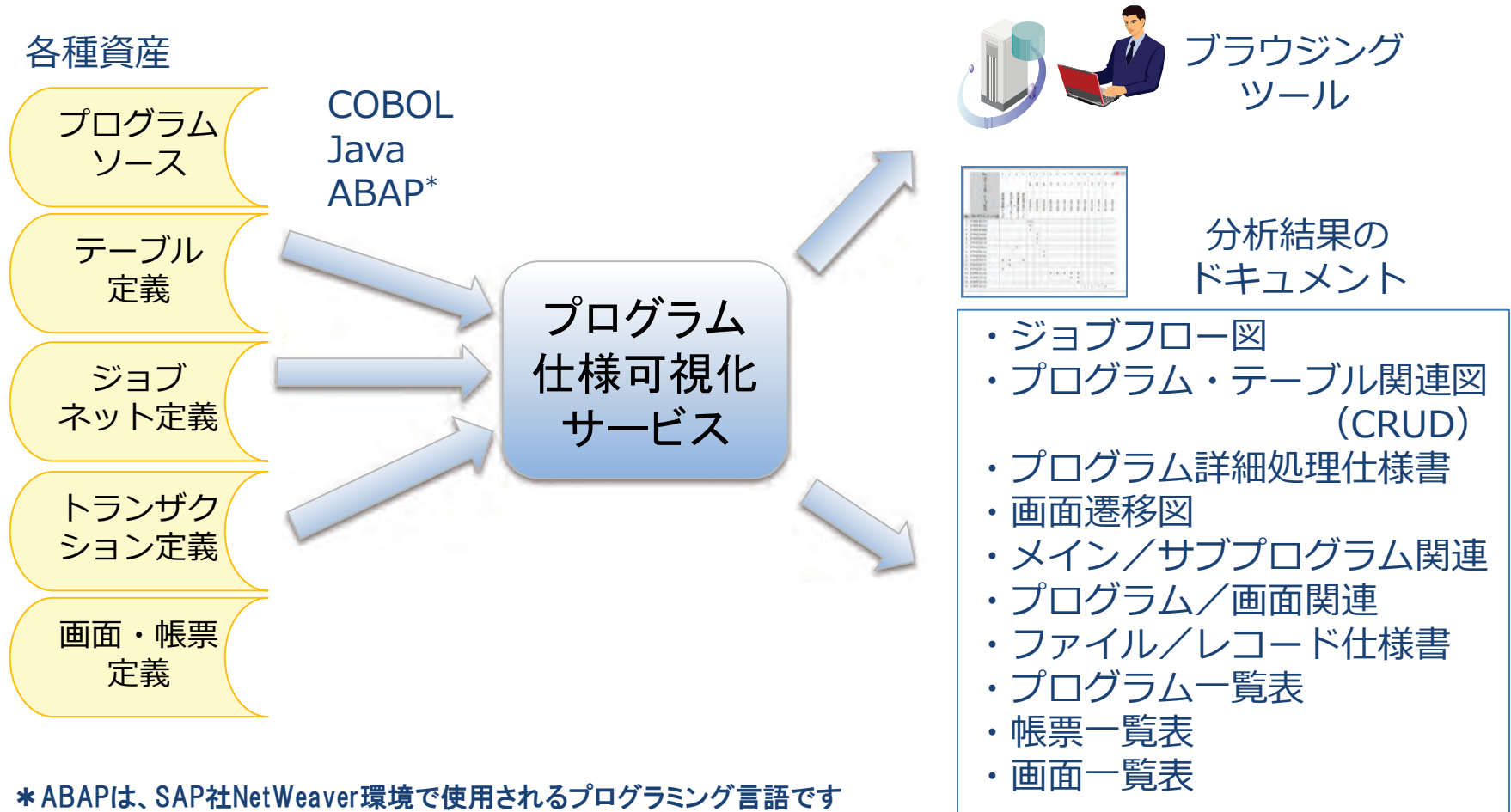
業務仕様 可視化

業務レベルの情報を可視化

プログラム仕様可視化をベースに情報収集、ドキュメント分析を実施、業務レベルの情報を関連付け、業務レベルの仕様を作成します。業務とシステムを関連付けながら調査を進めることで、抜け漏れを防止し、整合性の確保します。

3-6. プログラム仕様可視化

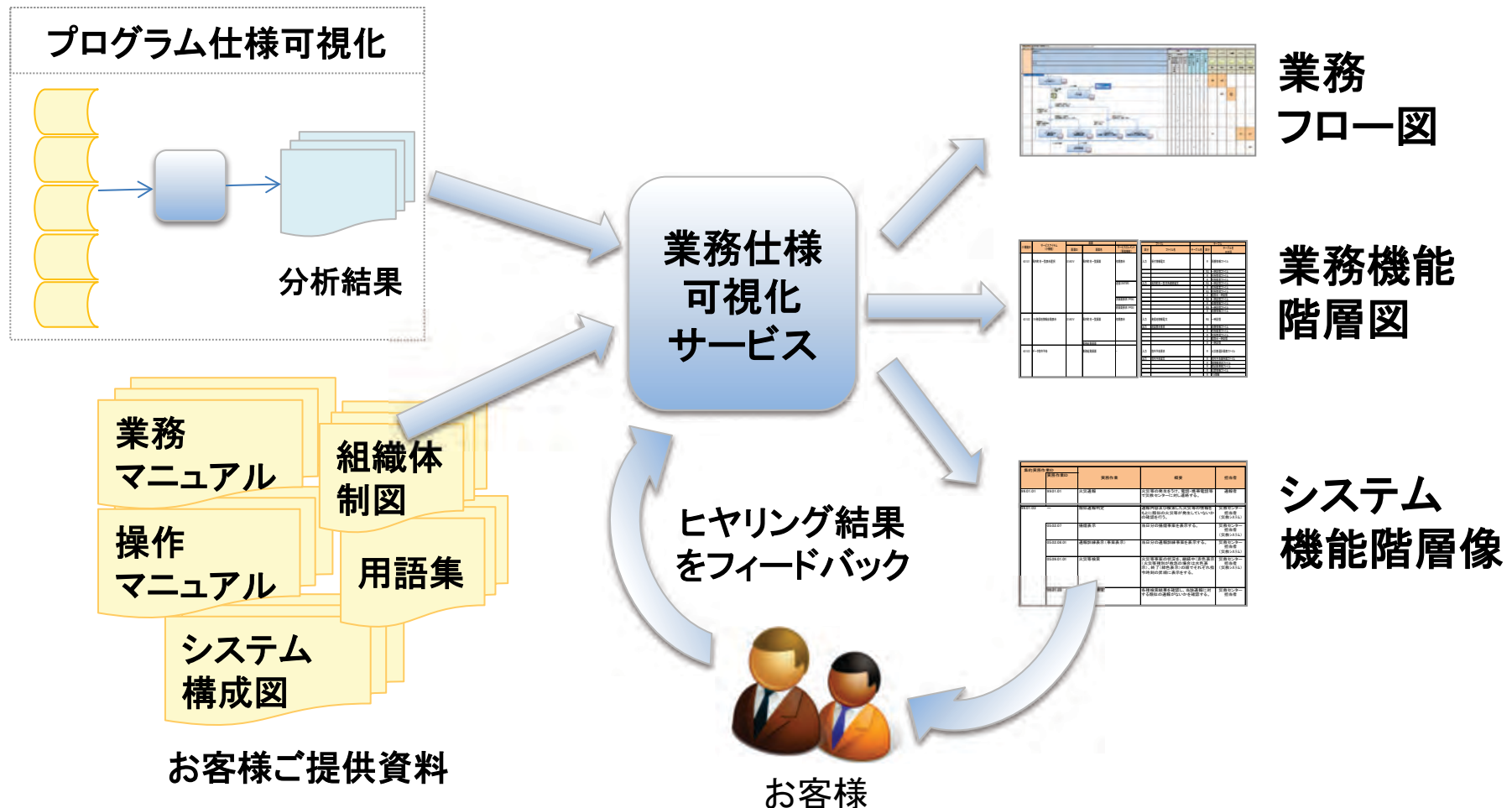
● 現行プログラム資産を解析し、プログラム構造と仕様情報を可視化



● お客様個別の課題に合わせた分析のカスタマイズも実施します。

3-7. 業務仕様可視化

- お客様資産 (プログラム、文書) より、業務レベルの仕様情報を可視化します
- 業務／システムの仕様を対応を取りながら調査することで、整合性を確保すると共に、調査の抜け漏れを防ぎます



事例の 特徴

- ツールを用いてジョブネットで処理される帳票出力処理を分析
- 帳票出力処理の統廃合の検討を効率化

要件定義 6ヶ月

【課題／対応】

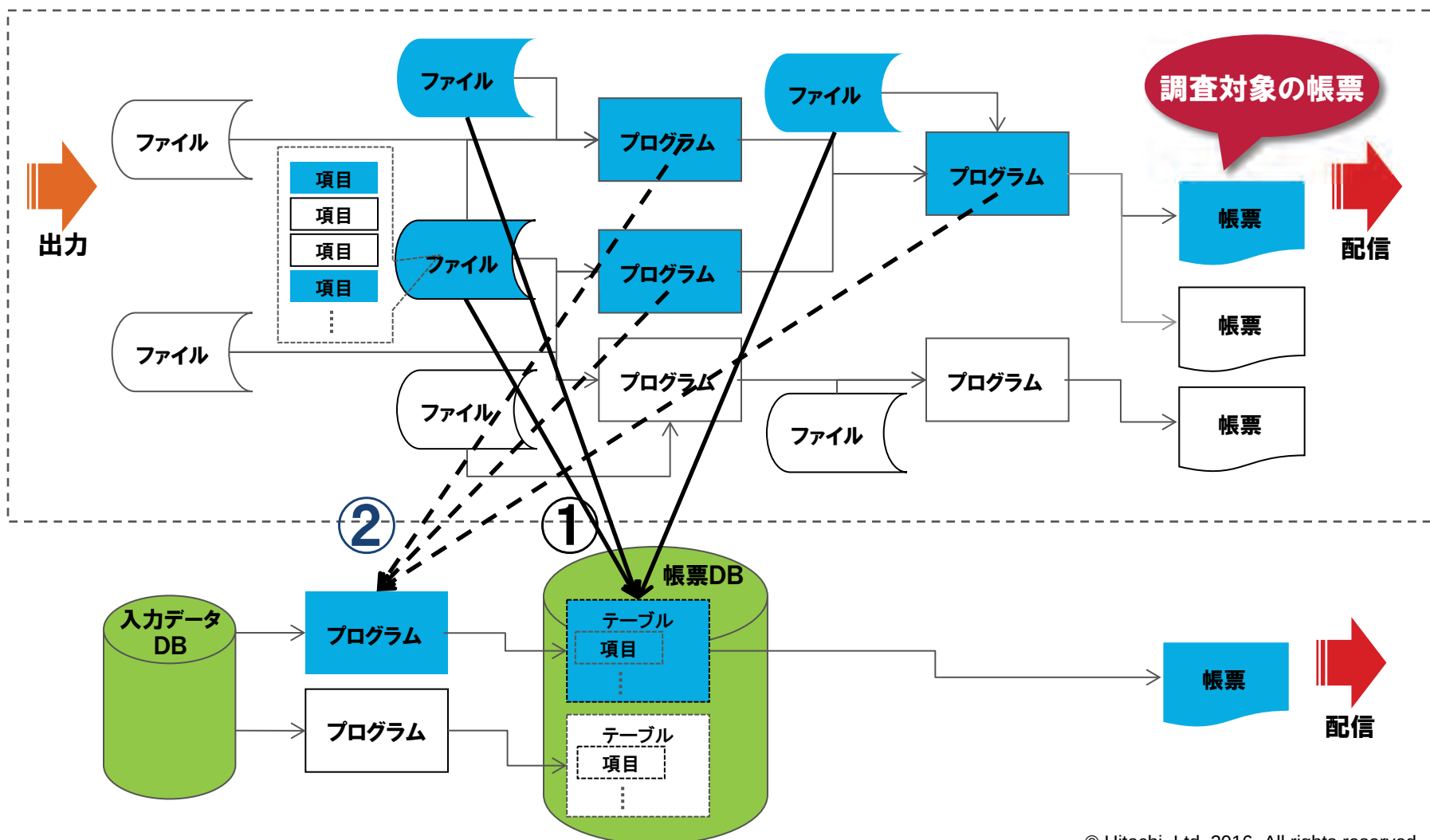
- 帳票ごとにバッチ処理が生まれ、データと編集処理が、異なる箇所に点在しており、保守性が悪い。再構築のタイミングで再設計したい。
 - ⇒ ・ツールにより帳票出力処理のデータフローを分析し、必要なファイルと項目、プログラムに点在する編集処理の箇所を特定
 - ・ツール結果による調査作業で、属人性を排除、網羅性を確保し、帳票出力処理のデータの統合、処理統合の再設計を効率よく実施

【資産概要】

- COBOLソース .. 約6,000本(約4MStep)
- JCL 約10,000本
- 帳票 約2,400種類

3-8. プログラム仕様可視化事例紹介: U社 (2/2)

- ① 対象帳票を起点に必要なデータを特定、帳票DBのテーブルへ統合
- ② 対象帳票を起点に帳票項目の編集箇所を特定、帳票編集プログラムへ統合



事例の 特徴

■ 現行の設計ドキュメントが十分でない状況で業務仕様可視化を実施、Javaでのリビルド型移行に繋げた。

仕様可視化 4ヶ月
+ プログラム再設計 8ヶ月

【課題／対応】

- システムで使われていた言語Natural、ADABASの保守期限が迫っており作り直す必要に迫られていた
⇒ 仕様回復することでリビルドでき、Javaへの移行が可能になった
- システムの保守体制が弱く、有識者に依存していた。
運用／保守ドキュメントが整備されていない。
⇒ Natural言語、JCLの解析のツール結果をたたき台としてヒヤリングを実施、現行業務を可視化。
可視化情報を基に再構築のための設計書を作成。

【資産概要】

- Naturalソース .. 約1437本(約405KStep)
- JCL 約451本
- DB .. 約101定義
- 画面 約759本
- 帳票..... 約177本

4. 最新の提供サービスと要素技術 ～ リビルドとこれからの技術

リホスト

既存システム資産をそのまま利用する方式

互換性のあるミドルウェアやエミュレーションを利用して既存アプリケーションをそのまま利用する方式です。移行費用を押さえつつ短期にインフラを刷新したい要望に応えることができます。

リライト

変換ツールでコードを他言語に書き換える方式

既存のプログラムを変換ツールで他言語に書き換えて、移行を実施する方式です。開発・保守の環境含めを新しい言語に統一したい要望に応えることができます。

リビルド

既存システムの業務仕様を再実装する方式

既存システムの仕様を生かし、再実装する方式です。移行方式としては最も自由度が高く、非機能要件に対し、最適な設計／実装の選択が可能です。最もコストがかかります。

現行仕様の 把握が必要

構築の前提となる現行の仕様情報の把握が必要

現行のドキュメントが保守されていない場合、移行に先立ってリドキュメントの作業が必要となります。資産分析ツールでの分析に加え、運用ドキュメントの調査やヒヤリングなど人手の作業も必要となります。

移行コスト が高い

他の移行手段と比べるとコスト高

再実装の工程を含む為、他の移行方式、リライト・リホストと比べるとコストが高くなります。構築の前提となるドキュメントが無い場合は、さらにコストがかかります。

開発手順 の必要性

開発工程になると再構築と同等となり統制が必要

開発工程の進め方は、スクラッチ開発と似たものになる為、開発プロセスに対し統制が必要となります。

リドキュメント 技術との パッケージ

仕様把握で必要となるリドキュメント技術을合わせて提供

リドキュメントで必要となる現行資産分析の技術により、仕様のドキュメント化と可視化情報を提供。

仕様情報の 効率的利用

仕様情報を構築フェーズに連携することによるコスト低減

可視化技術により作成した仕様情報を、構築フェーズにてアプリケーション開発フレームワークの設計情報や、BRMS*ルールエンジンと連携することによりコスト低減を実現。

* BRMS(Business Rule Management System) : ルールとして定義した業務上の判断をアプリケーションから呼び出して実行する仕組み

開発フレーム ワークの提供

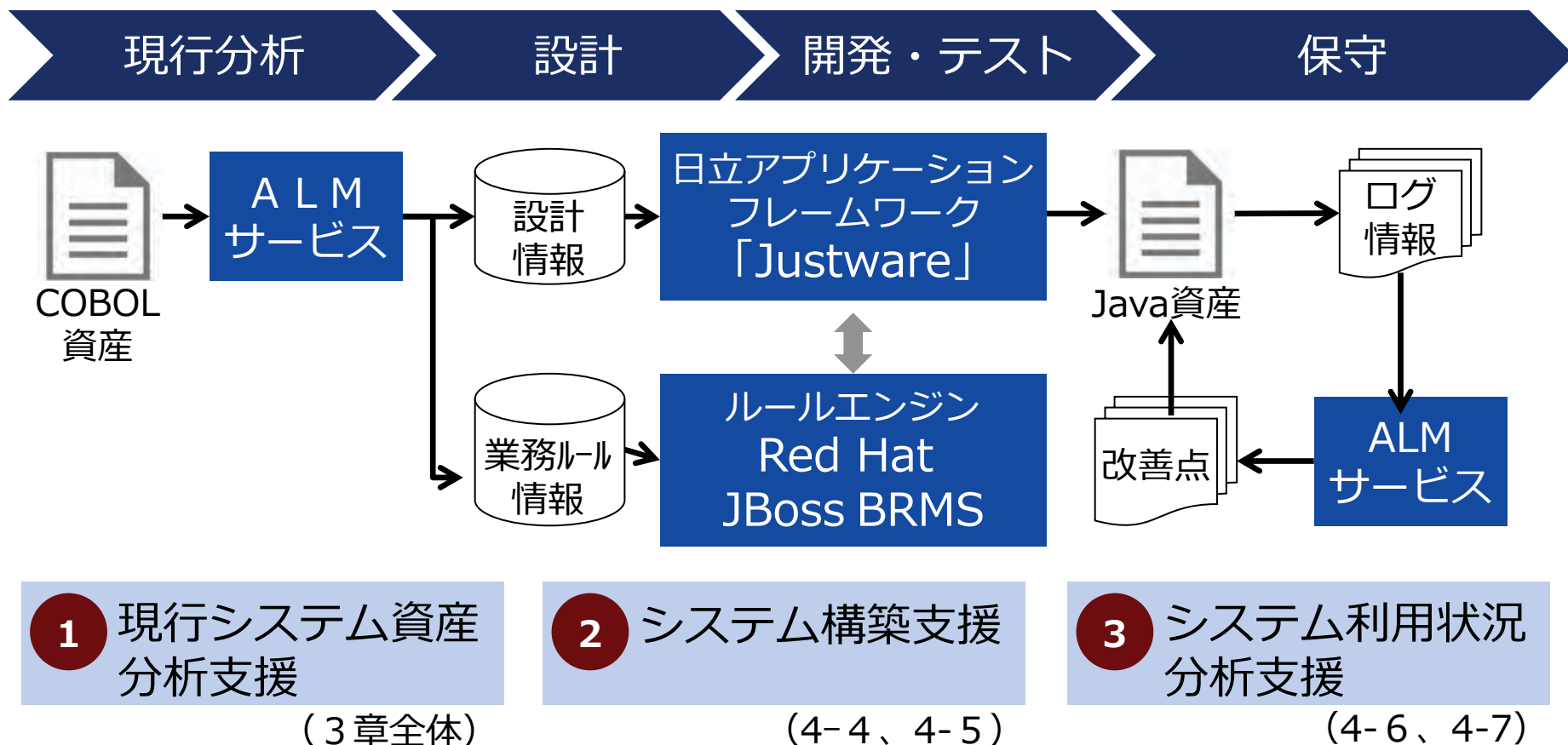
実績ある開発フレームワーク、開発標準の適用による 効率的かつ統制のとれた開発の実現

これまでの大規模開発等で培った開発ノウハウが活かされた開発標準を適用することで、統制のとれた均質的な開発が可能。これにより、手戻りを防止でき、効率的な開発を実現。

4-3. リビルドの課題を解決するサービス

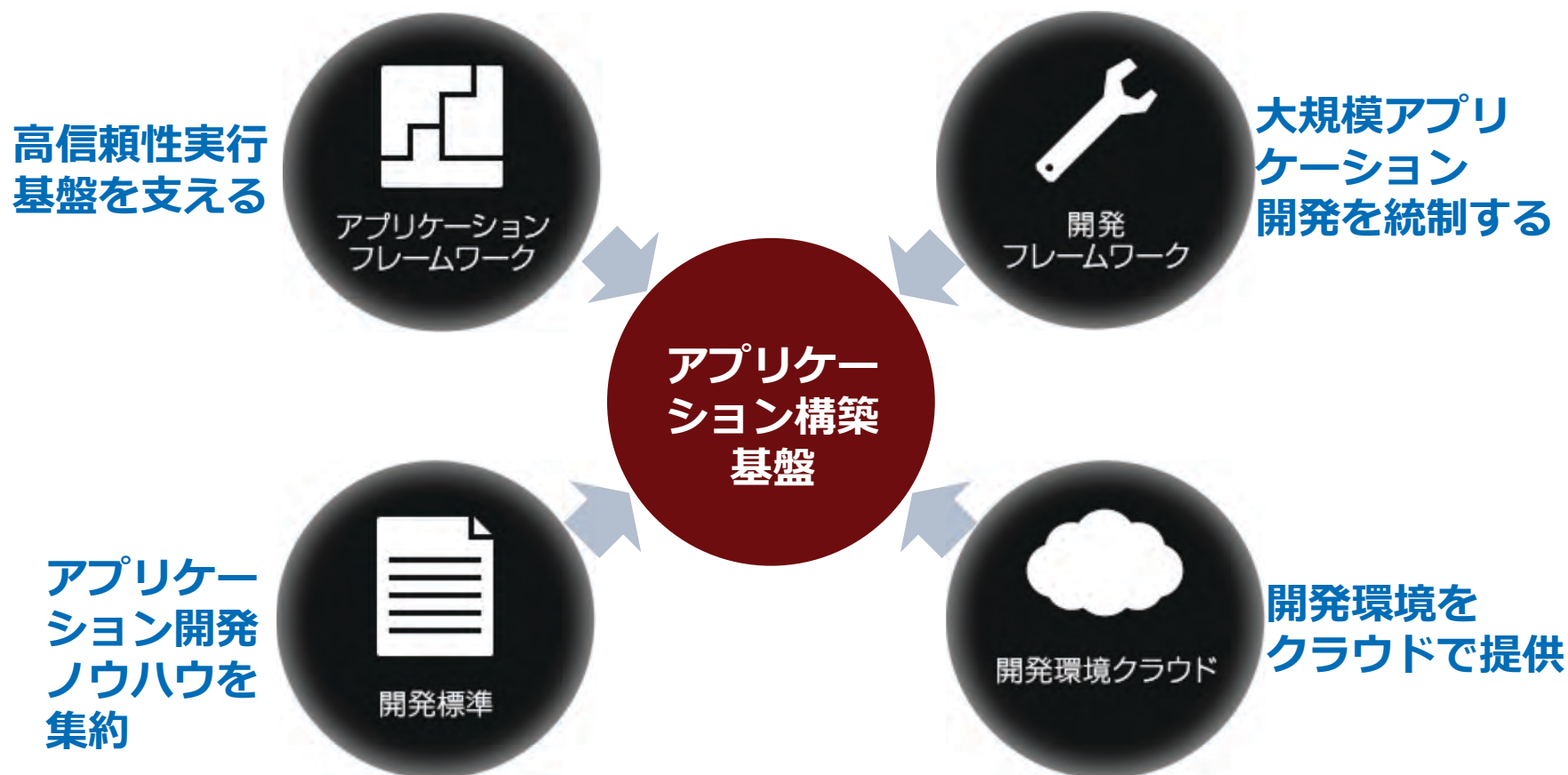
● COBOL→Javaのリビルドを支援する「COBOLマイグレーションサービス」

COBOL資産のリビルドを現行分析から構築までサポートし、Java資産へスムーズな移行を実現します。また、ログ情報をもとに、移行後のシステムの利用状況进行分析し、業務上の課題を抽出することで、継続的な業務改善を支援します。



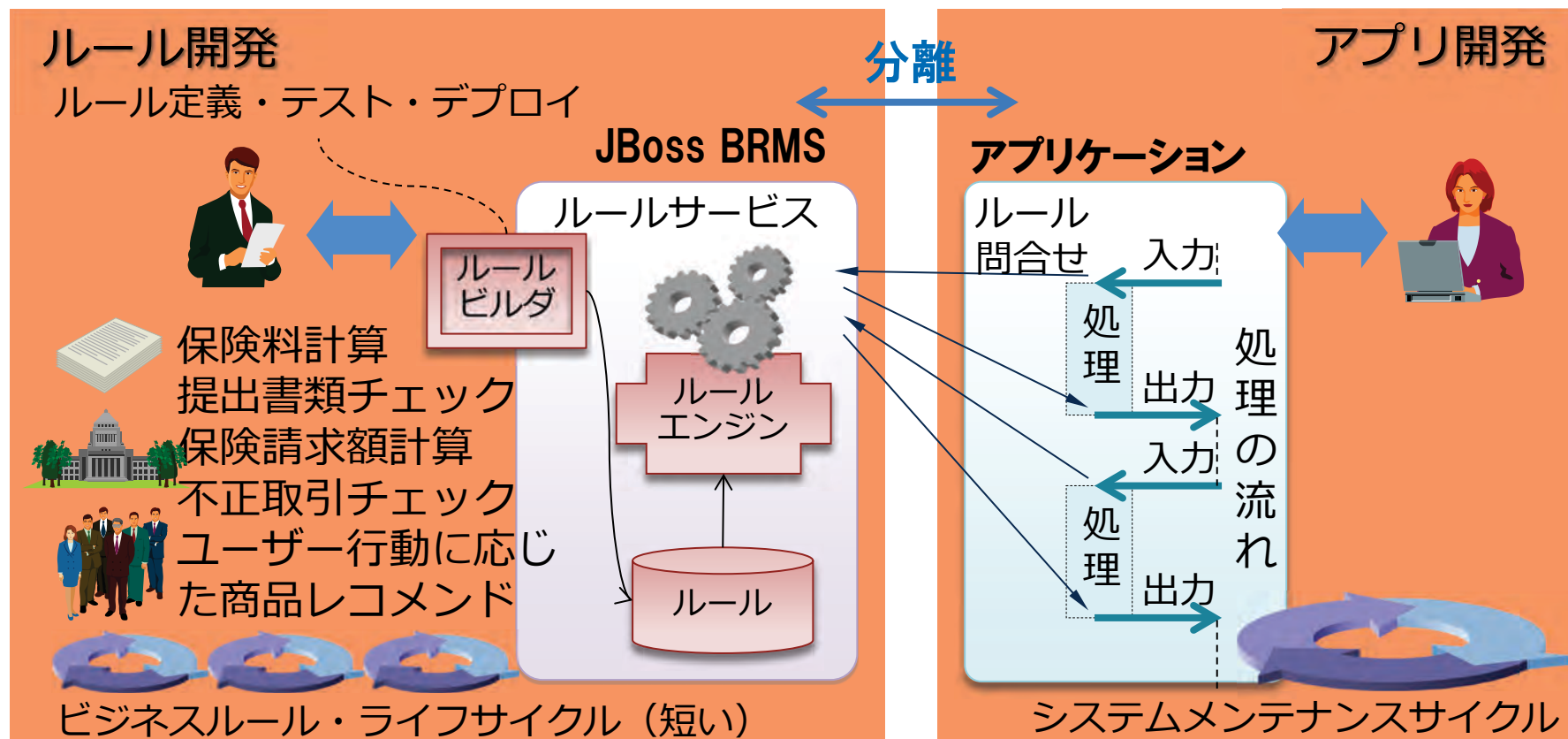
4-4. 高信頼のアプリ開発基盤によるリビルドの実現

● アプリケーションの開発と実行を支援する統合アプリケーション構築基盤 「Justware」



4-5. ルールエンジンとの連携による保守効率化の実現

● ビジネスルールを開発・管理するための統合基盤「Red Hat JBoss BRMS」



ビジネスロジックの可視化 変更の多いビジネスルールを分離して一元管理

ビジネスニーズの変化に柔軟に対応 ノンコーディングでルールを素早く反映

開発の簡素化 = 効率化／生産性向上 保守プロセスの簡素化により保守を効率化

● 業務やシステムの課題を抽出する「システム利用状況分析支援サービス」

移行後のシステム利用状況の分析による業務プロセスやシステム運用上の課題を抽出し、業務改善を継続的に支援します。

業務改善の一部作業に弊社システムを利用することで業務改善のスピードアップを図ります。

システム利用状況分析支援サービスの特徴を以下に示します。

業務上の 課題抽出

ログ情報を弊社ツールで解析し、システムの利用状況を分析することで、お客様が把握しきれない業務上の課題抽出を行います。

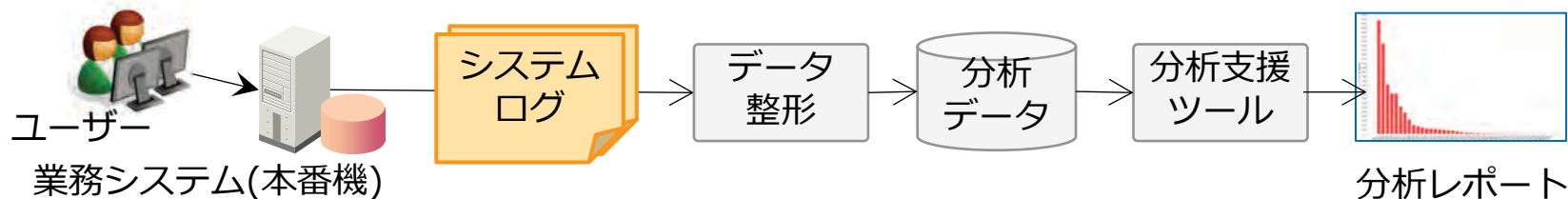
継続的な 支援

システムの利用状況の分析を、業務改善を実施する度に定期的に実施することで、継続的な改善を支援します。

4-7. 利用状況分析サービスの分析の特徴

アプリケーションのログ情報等をもとに、システムの利用状況を分析して、お客様の業務とシステムの潜在的な課題を探ります。

- 「誰が(人)」「いつ(時間)」「何の(業務)」を行っているかに関する情報を抽出
- 特異パターンや想定される使われ方とのズレを把握し、課題候補を抽出



分析で分かることと課題の例

- 機能別利用傾向 使われない機能がある ⇒ 不要機能削除
- ユーザー別利用傾向 ユーザー毎の負荷の偏り ⇒ 業務プロセスの見直し
- 拠点・月別利用傾向 拠点ごとに利用される機能が異なる ⇒ 業務の標準化
- リードタイム傾向 ある作業で滞留が発生しがち ⇒ 業務プロセスの見直し
- ボトルネック傾向 ある作業がボトルネック ⇒ 業務プロセスの見直し
- 例外分岐傾向 好ましくない処理への分岐が多い ⇒ プロセスや機能の見直し

● 仕様書がない/実態と一致しないシステムから仕様を抽出する技術

情報システム スキャンング 技術

現行システムの仕様調査を高品質化・高効率化

アプリケーションのソース分析だけでなく、ログ、データの分析を組み合わせ、現行システムの実態に即した仕様の抽出をめざします。要件定義漏れによるシステム事故の防止や納期遅延リスクを低減します。

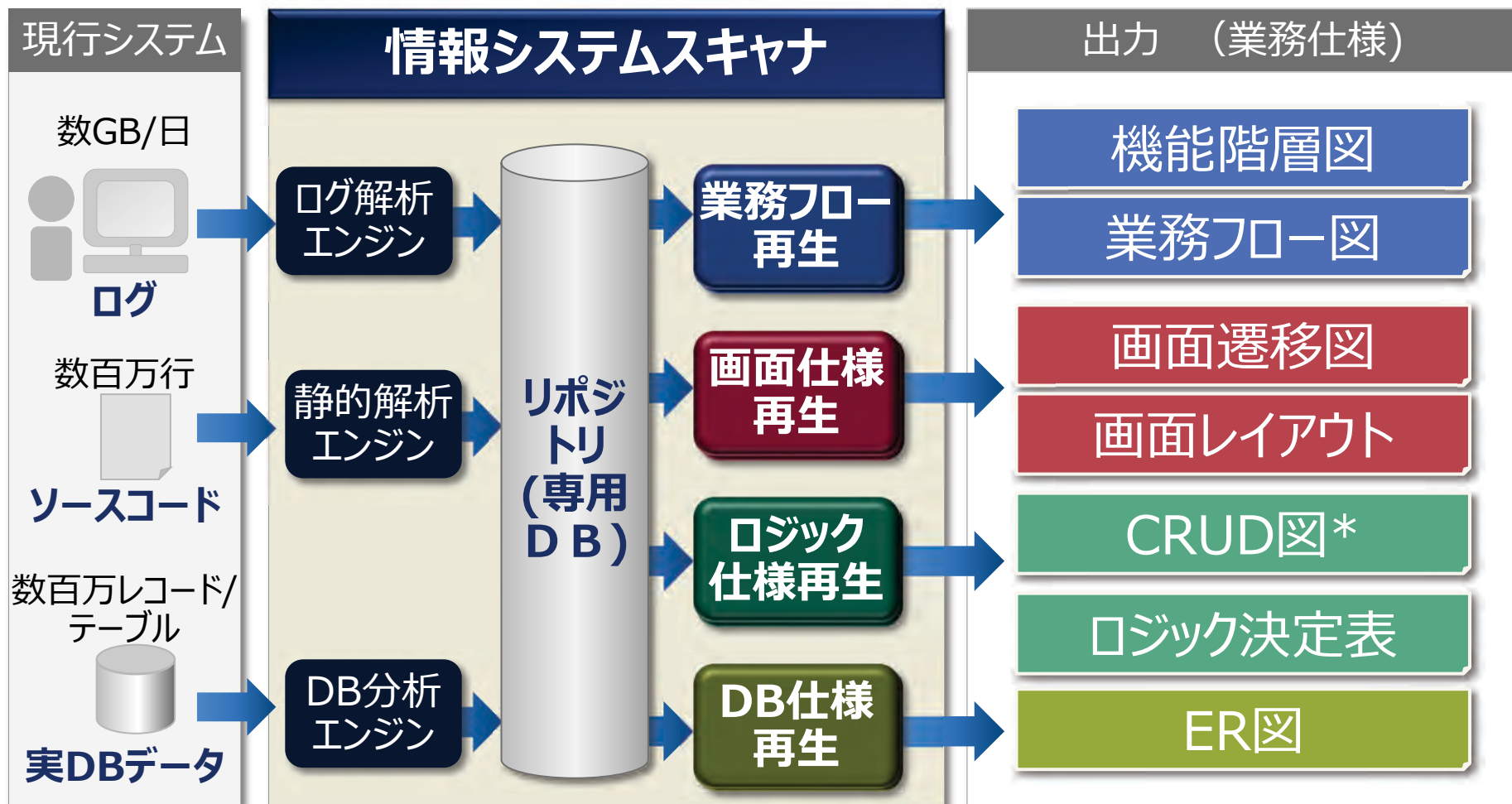
DBデータ 実態可視化 技術

現行DBの仕様調査や異常データ発見を効率化

既存システムのデータを分析し、データ分布や仕様外データのパターンを抽出、データ移行の問題点の早期発見やテストデータの自動生成によるテストの効率向上に寄与します。

4-8-1. 情報システムスキャン技術

- ログ・プログラム・既存DBの分析結果をより合わせて整合のとれた現行システムの仕様を抽出します。抽出した仕様を用いることで、要件定義漏れによるシステム事故の防止や納期遅延リスクを低減できます。



4-8-1. 情報システムスキャンニング技術

- 各再生仕様間の関連に基づき、ログなどの「使われ方」とソースなどの「作り」を繋ぎ、仕様再生に必要なビューを生成します。

※ 各再生技術は独立しており、再生したいビューやお客様からご提供いただけるデータに基づき、適宜組み合わせで仕様再生します

業務フロー再生

業務の流れを俯瞰して
見ることができます。

現行業務で用いられる
機能の利用実態が
明らかになります。

機能階層図

業務フロー図

画面仕様再生

画面ごとの流れと、
つながりが分かります。

また画面ごとの入出力
項目やその値なども
分かります。

画面遷移図

画面レイアウト

ロジック仕様再生

どの機能がどのデータ
に操作を行うかにつ
いての仕様を整理し
一覧で表示します。

CRUD図

ロジック決定表

DB仕様再生

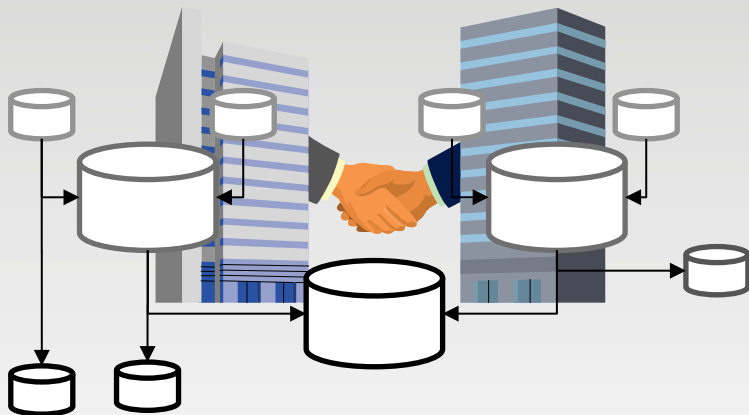
データのつながりを
ER図で図式化します。

ER図

4-8-2. DBデータ実態可視化技術

- 近年の情報システム開発プロジェクトでは、必ずと言って良いほど何らかの既存システムや関連システムが存在
- 長年運用されたシステムでは…

度重なる統廃合による仕様の複雑化



手順外運用による異常データの発生



仕様書のメンテナンス漏れによる実態との乖離

仕様書どおりの
ハズです



実データが仕様書と全然違うぞ...
とても間に合わない!



4-8-2. DBデータ実態可視化技術

主要課題

- 移行時にデータクレンジングが必要か判断したい。
- 現行DBの仕様や異常データを漏れなく調査し、移行をスムーズに行いたい。

解決策

- 既存DBの業務データを独自のツールで分析。
- 分析結果から現行DBの仕様や異常データの有無が浮き彫りになります。

分析、可視化



結果調査

そのデータは
対策が必要ですね...



対策

この方針で
対策しましょう



仕様書に現行仕様を明記
異常データ対策を追加 など

分析レポートの一例(機能1:バリエーション可視化)

カラム名	パターン	例	割合
顧客ID	ITM[0-9]{5}	ITM01234	70%
	SYH[0-9]{6}	SYH012345	29%
	<<特異値>>	ZZZZZZZZ	1%



日立独自の
実態分析ツール

■ 情報システムスキャンニング技術の事例

総務系webシステムにて本技術を試行
Java 3Mstep,ログ3か月分から業務仕様を再生し、
システムとドキュメントを一致化

■ DBデータ実態可視化技術の事例

社内某システムのデータ移行時の障害を分析
本技術により70%(143/205件)の障害を防止可能な見込み

5. まとめ

保有する資産を活かしたい場合、モダナイゼーションはシステム再構築の有効な選択肢です。

しかし、その上で大事なのは、システム再構築後に

『事業発展に貢献できるシステム』

へと成長させていくことです。

そのためには、資産／業務の特性に合わせた適切な方法を選択してモダナイゼーションを行うことが成功の鍵です。

リHOST系のサービス

■ マイグレーションサービス

日立COBOL2002をはじめ、互換性を提供する各種ミドルウェアの組み合わせにより移行を実現するサービスです。

リライト系のサービス

■ 言語リニューアルサービス

言語変換の技術を用い、低コストに移行を実現したサービスです。(PL/I→COBOL、COBOL→Java等)

リドキュメント、リファクター系のサービス

■ アプリケーションライフサイクルマネジメントサービス

アプリケーションの分析を通じて、システムの高度化をサポートするサービスです。

リビルド系のサービス

■ COBOLマイグレーションサービス

現行分析から構築までサポートすることにより、COBOLからJavaへの再構築型の移行を実現します。BRMSの導入で保守性を向上すると共に、利用状況を分析し業務上の課題を抽出することで、移行後の継続的な業務改善を支援します。

モダナイゼーション技術についての問合せ

- 株式会社 日立製作所 サービスプラットフォーム事業本部
システム基盤ソリューション部 ミドルウェアソリューション窓口
E-mail: midsol-p@itg.hitachi.co.jp

関連するURL

- ミドルウェア ソリューション
 - ・サーバー移行支援 ・ミドルウェア移行支援
 - ・COBOL活用 ・レガシーマイグレーション支援

<http://www.hitachi.co.jp/Prod/comp/soft1/solution/index.html>
- COBOL2002

<http://www.hitachi.co.jp/Prod/comp/soft1/cobol/index.html>
- COBOLマイグレーションサービス

<http://www.hitachi.co.jp/cobol-migration/>

付録. ITモダナイゼーションの最新情報やイベントの先行案内、 特別資料ダウンロードなど特典が盛りだくさん！

●日立ID会員サービスとは？

日立製作所が提供するインターネットを利用した無料の会員サービスです。

ひとつのIDでメールマガジンや
セミナー／イベントに簡単登録！

日立ID会員サービス



●日立ID会員サービスにご登録すると？

(1)日立IDにご登録いただいたお客様は、以下のような日立ID会員サービスを、会員専用ページから簡単にご利用することができます。

- ✓ 各種メールマガジンの配信登録、停止申し込み
(メールマガジンは適宜追加してまいります)
- ✓ セミナー／イベントへの参加登録、登録内容の確認・変更
- ✓ 日立グループの情報・通信事業に関する各種ご案内の配信登録

(2)他にも、ご登録者さまが興味をお持ちの分野・製品に関する情報を適宜提供してまいります。

●是非、日立ID会員サービスへご登録ください！

日立ID

検索



クリック！

<https://hjid.ext.hitachi.co.jp/public/application/add/607>

Hitachi IT Platform Magazine



- HITACHI, Justware, HiRDB, DCCM, OpenTP1, XMAPは、株式会社 日立製作所の商標または登録商標です。
- AIXは、世界の多くの国で登録されたInternational Business Machines Corporationの商標です。
- IBM, z/OS、MVSおよびIMSは、世界の多くの国で登録されたInternational Business Machines Corporationの商標です。
- Linuxは、Linus Torvalds氏の日本およびその他の国における登録商標または商標です。
- Red HatおよびJBossは、米国およびその他の国でRed Hat, Inc. の登録商標もしくは商標です。
- MicrosoftおよびWindowsは、米国Microsoft Corporationの米国およびその他の国における登録商標または商標です。
- OracleとJavaは、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。
- UNIXは、The Open Groupの米国ならびに他の国における登録商標です。
- HP-UXは、Hewlett-Packard Development Company, L.P.のオペレーティングシステムの名称です。
- NATURALは、独国Software AGの商品名称です。
- Q言語は、日立システムズの商品名称です。
- SAP, ABAP, およびSAP NetWeaverは、ドイツおよびその他の国々における SAP AG の商標または登録商標です。
- その他記載の会社名、製品名などは、それぞれの会社の商標もしくは登録商標です。

END



I Tモダナイゼーションの成功事例と 最新の要素技術

2016/04/26

株式会社 日立製作所 ICT事業統括本部
アプリケーションサービス事業部
サービスソリューション本部 サービス統括部

新家 博文