COBOL資産移行

2011年11月07日

宇部情報システム
URL http://www.uis-inf.co.jp

はじめに

本日はお忙しい中、本セミナーの為にお越し頂き、ありがとうございます。

本セミナーにより御社で保有しているCOBOL資産についての検討材料となれば幸いです。

今回ご説明の資料の構成は以下となっております。

１．COBOL資産移行の必要性

２．資産移行のパターン

３．UIS事例紹介

どうぞ、よろしくお願いします。

用語の説明

オープンシステム：相互運用性、移植性、オープン標準を持ったコンピュータシステム。Unix、Windowsで構築されたシステムを指すことが多い。

レガシーシステム：メインフレーム（ホスト）、オフコン上で構成されたシステム。

１．COBOL資産移行の必要性

１．COBOL資産移行の必要性

■　ＣＯＢＯＬが多く使われる理由（背景）

①黎明期の１９６０年頃に誕生し、他の言語よりも先行した

②産業界の共有物で特定ハードウェアによらない共通ビジネス用言語（COmmonBusinessOrientedLanguage）

③すべてのハードウェア・メーカーがＣＯＢＯＬコンパイラを開発し当時のコンピュータでも使用に耐えられる性能を実現した

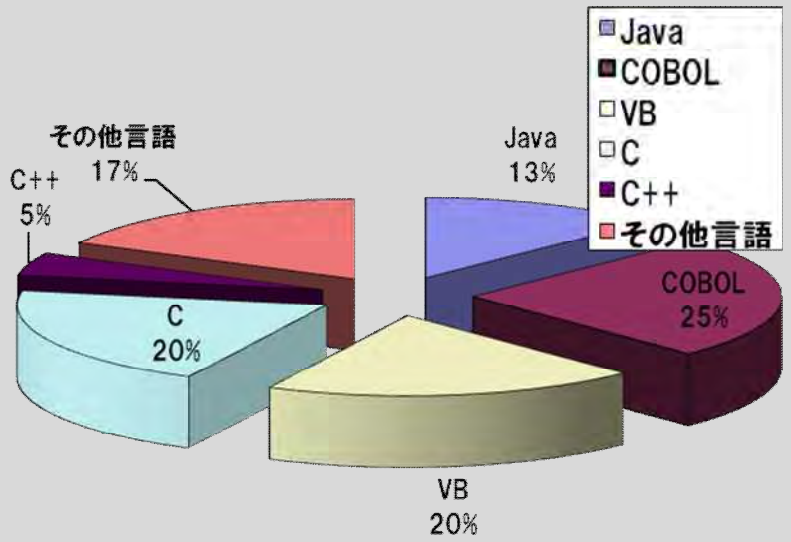
オープン化システムが主流となる中、現在も現役として多くのシステムが稼働している。

１．COBOL資産移行の必要性

■ 開発言語　（ソフトウェア開発　データ白書）

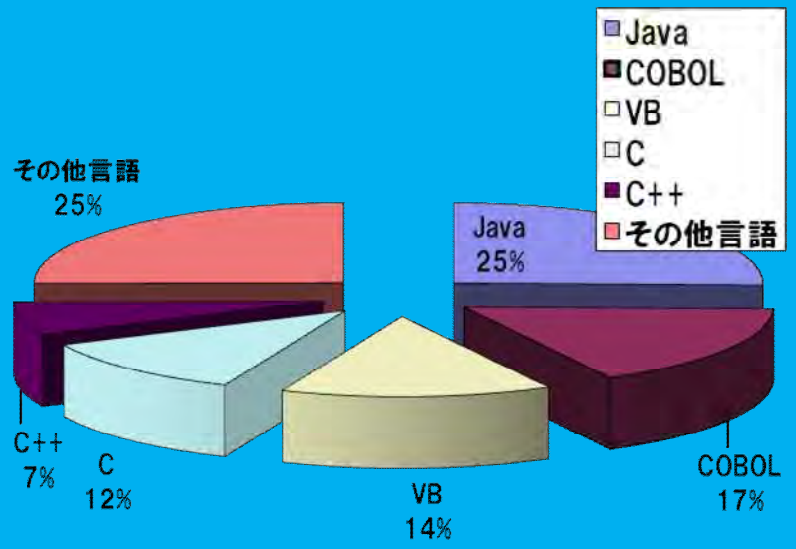
出典　独立行政法人情報処理推進機構（ＩＰＡ）、ソフトウェア・エンジニアリング・センター（ＳＥＣ）

２００５年



対象＝８５１プロジェクト

２０１０－２０１１年



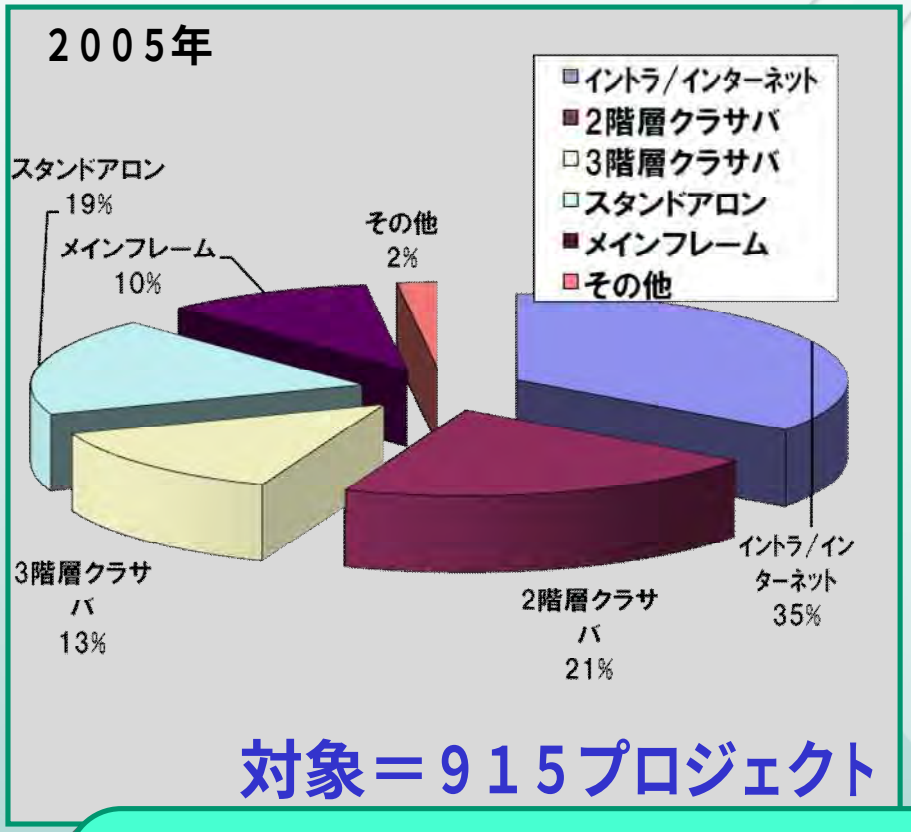
対象＝２，４１７プロジェクト

JAVA言語が主流となってきている。

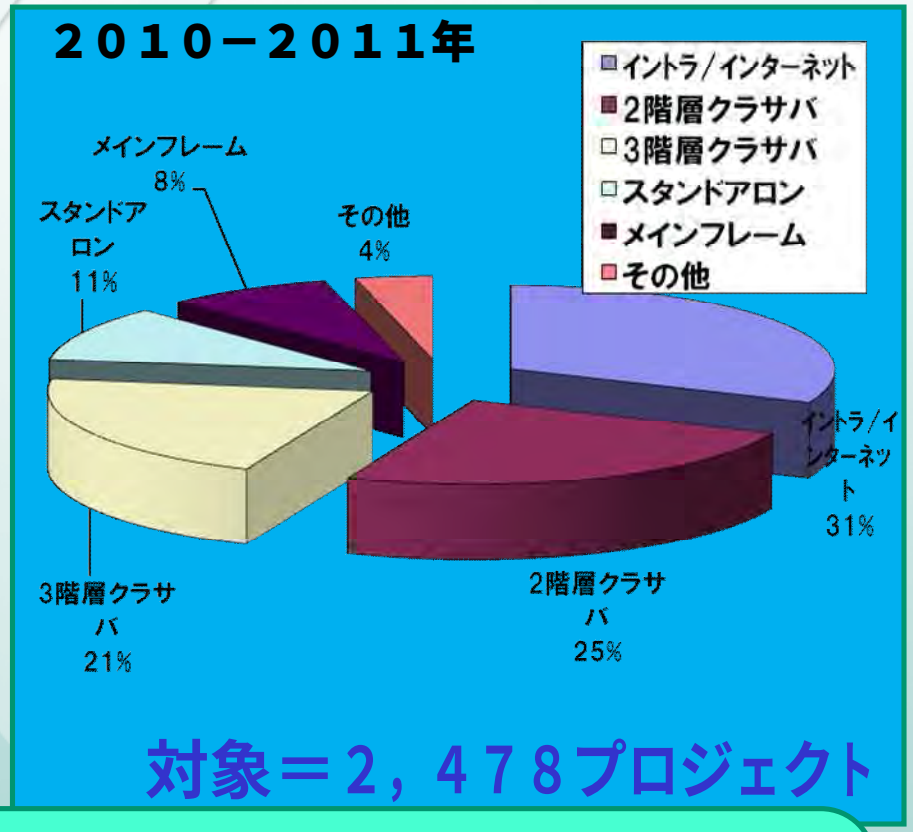
しかし、COBOL言語の需要はまだまだ高い！！

１．COBOL資産移行の必要性

■ アーキテクチャ



対象＝９１５プロジェクト

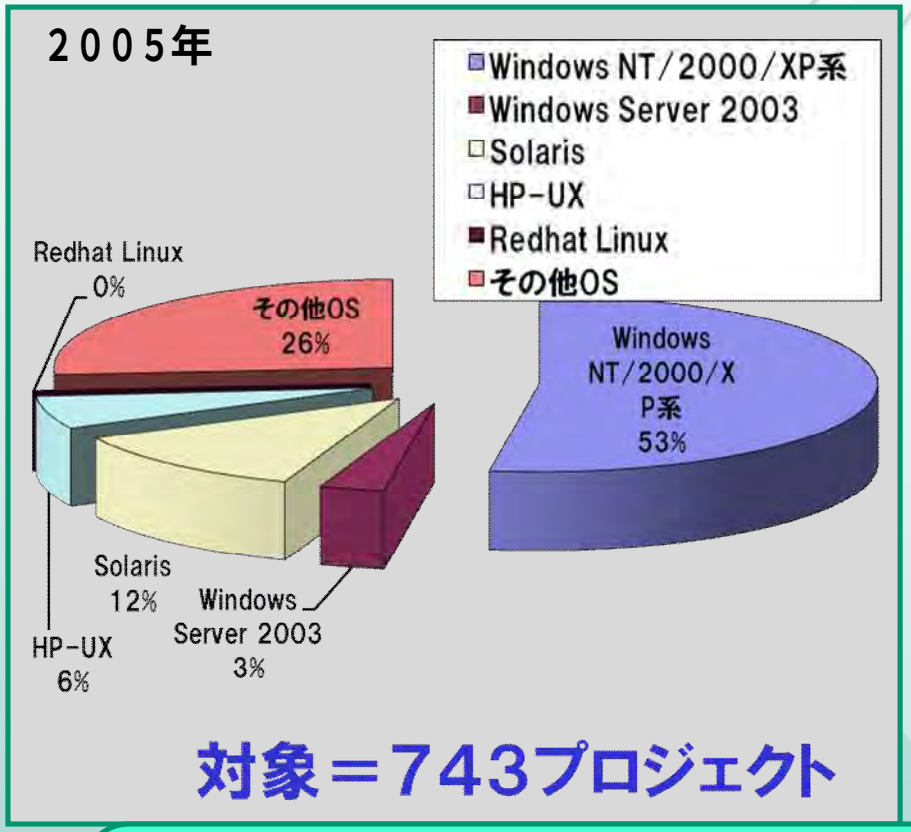


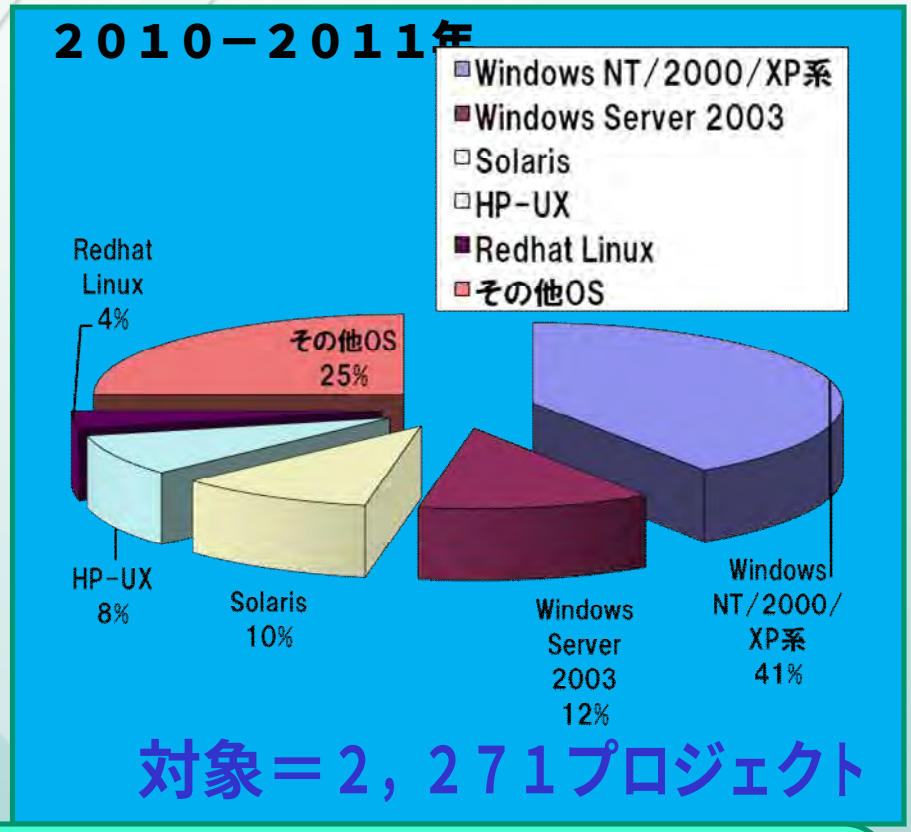
対象＝２，４７８プロジェクト

Web系が主流になっている。
オフコンはスタンドアロンに含まれる。
２００５年からの構成はほぼ横ばい

１．COBOL資産移行の必要性

■ 開発対象プラットフォーム





Windows系、UNIX系が75%を占める。

メインフレーム、オフコンはその他に含まれる。

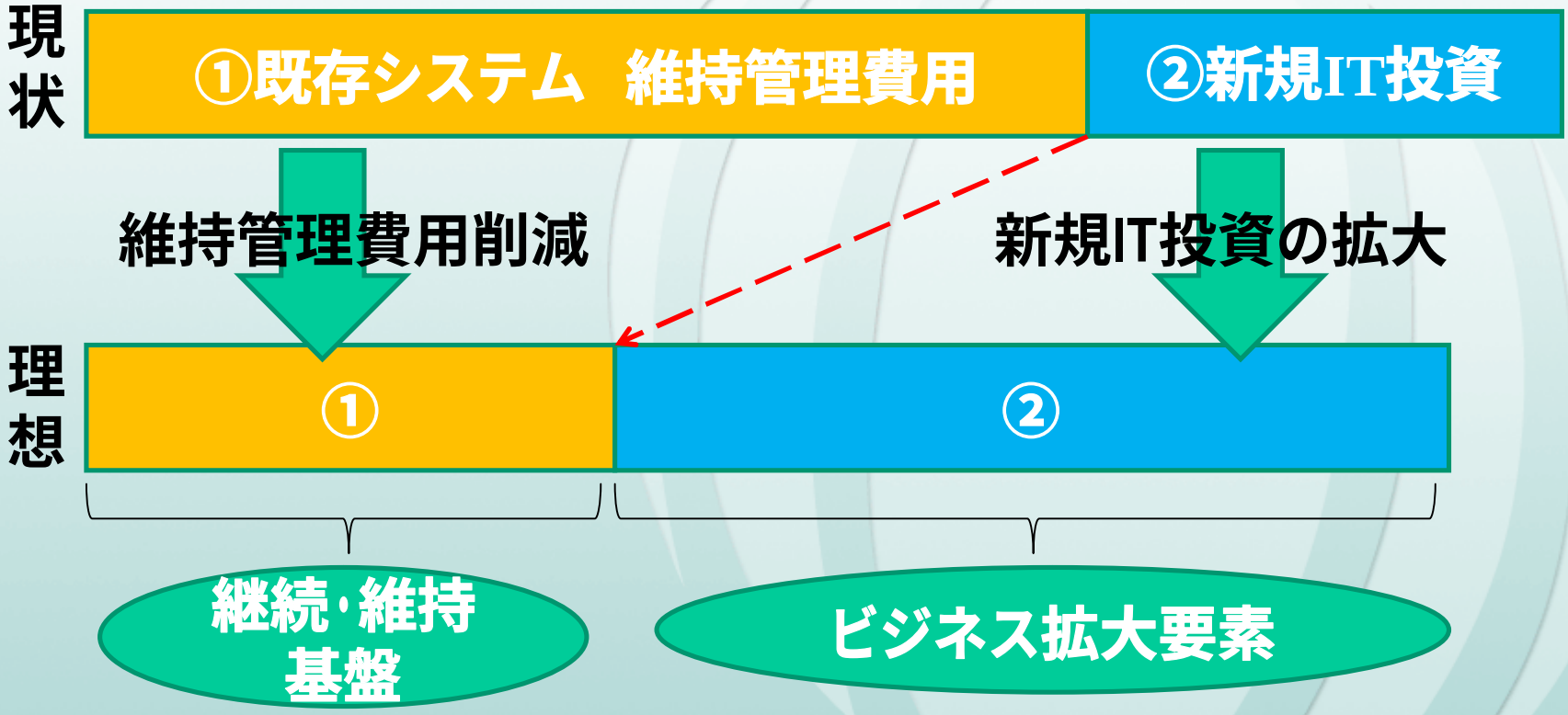
１．COBOL資産移行の必要性

- 言語、アーキテクチャ、プラットフォームから
  - オープン系言語（Java、VB、C、C＋＋）が６０％を占める
  - オープン系のアーキテクチャは７５％を占める
  - オープン系のプラットフォームが７５％を占める

２００５年までにシステムの４分の３はオープン化が進んでいます。
２００５年からも緩やかではあるが、オープン化が進んでいる状況です。
レガシーシステム利用ユーザはオープン化の効果や将来のシステム像を見据え検討していく必要があります。

１．COBOL資産移行の必要性

■ IT投資の理想型



維持管理費用を削減し、ビジネス拡大要素となる投資を増やすことが理想！！！！

１．COBOL資産移行の必要性

■ 現状

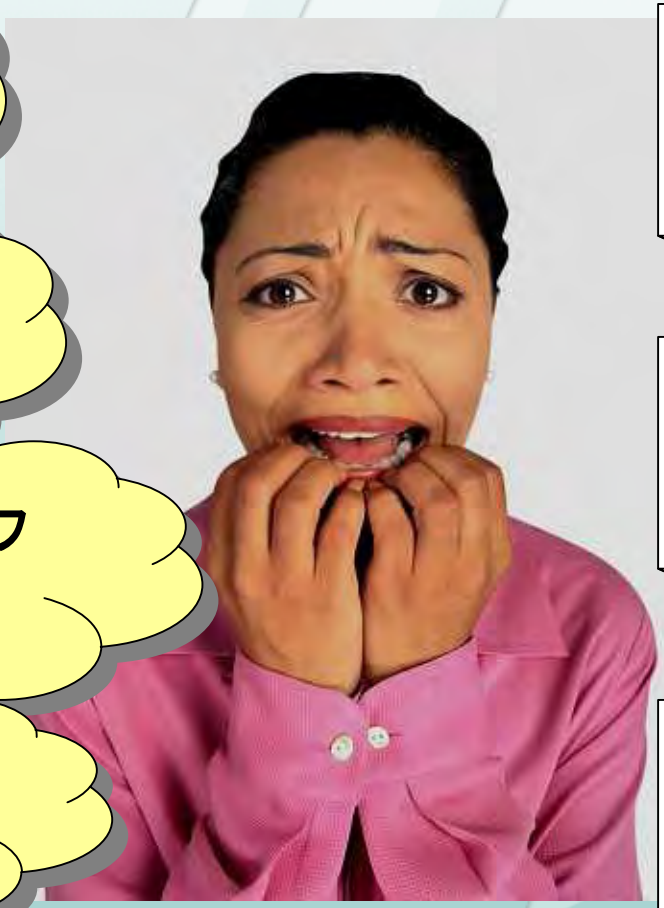
レガシーシステムユーザーの切実な現状

景気後退による
IT費用削減

レガシーシステム
の高額な維持費

レガシー製品のライフ
サイクル
(部品供給の終了)

対応要員の高齢
化、縮小化



高額なシステム
更改費用

高いシステムの
更改リスク

ベンダーロック
オン

１．COBOL資産移行の必要性

■ レガシーシステムの抱える問題点

①TCOがオープンシステムに比べ、高い

②拡張性もオープンシステムに比べ低い
GUIや帳票出力においても制約が多い

③バッチ処理中心で構築されている場合が多く、リアルタイムに情報を共有できていない場合がある

④他システムとの連携に、文字コード変換やファイル連携等の制約が多い

⑤COBOL技術者が減少傾向にある

経営戦略（事業拡大、業務見直し、COST削減）に合わせてレガシーシステムに対して検討する必要も！！

１．COBOL資産移行の必要性

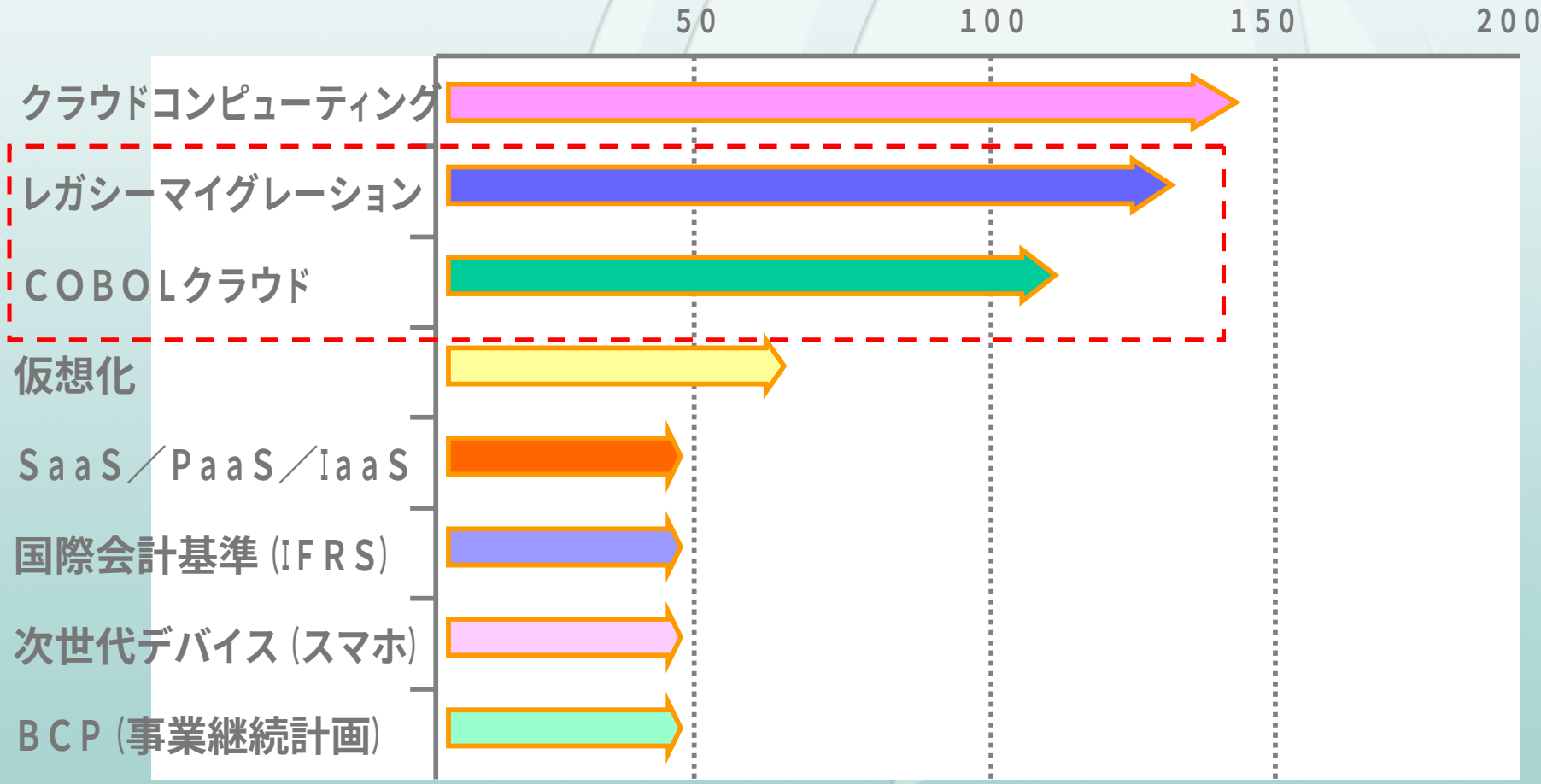
■ オープンシステムとの比較

レガシーシステム		オープンシステム
高コスト	TCO	低コスト
低い	拡張性	高い
低い	リアルタイム性	高い
減少傾向にある	対応要員	豊富
対応ベンダーが明確であり原因究明、対応が迅速	障害発生時の対応	アプリ、DB、OSなど問題原因究明に時間が掛る
ランニングコスト	COST	移行費用

１．COBOL資産移行の必要性

■ レガシーシステムユーザーの期待は？

COBOLコンソーシアム/日経BPセミナー事業センターのアンケート結果より、



１．COBOL資産移行の必要性

- レガシーシステムユーザーは

まず現状を把握し、将来に向けてのあるべき
企業情報システムの姿を検討してまいりましょう。

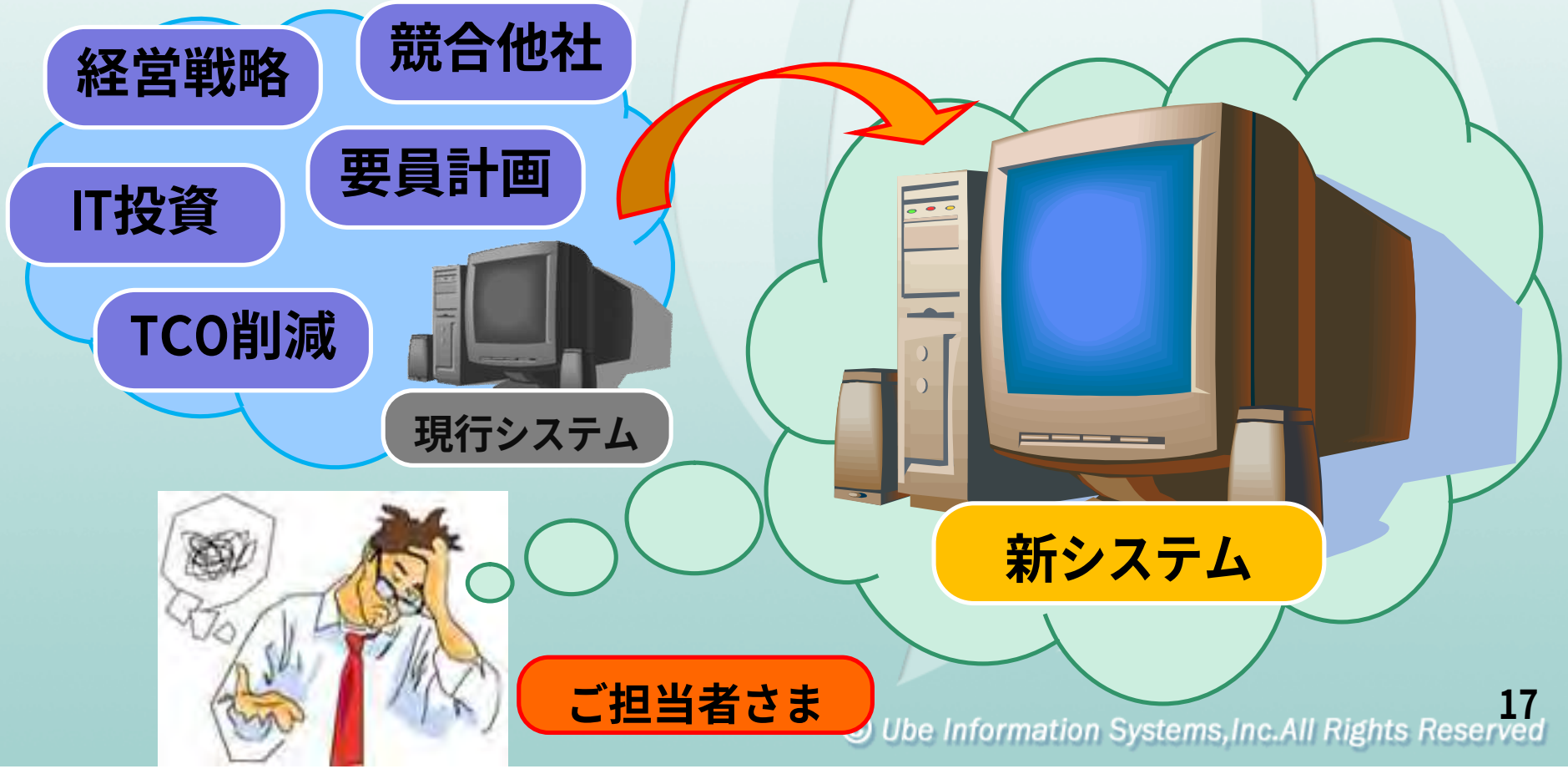
２．COBOL資産移行のパターン

２．COBOL資産移行のパターン

- ご担当者さまの悩み

  COBOLシステムはまだまだ現役である。

  だが、今後の拡張（中期、長期的な視点での最適なシステムの姿）も意識し検討する必要がある。



２．COBOL資産移行のパターン

- COBOL資産移行のパターン

	COBOL資産を維持・流用	COBOL資産を捨てる
メインフレーム・オフコンを残す	◆ラッピング ◆リファクター	
メインフレーム・オフコンを捨てる	◆リホスト ◆リライト ◆リビルド	◆BPR

今回のご説明部分

BPR:business process re-engineering

2．COBOL資産移行のパターン

■ COBOL資産移行のパターン（撤廃型）

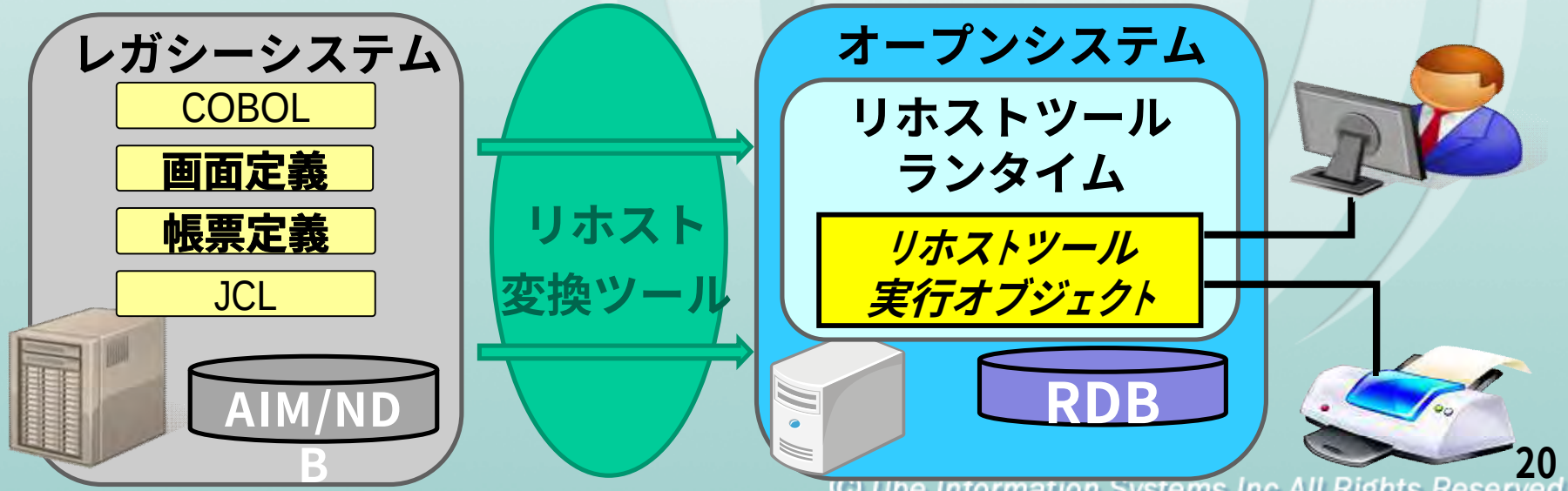
リホスト	• ソース、データをできるだけ手を加えずにオープン系システム上に移行する。
リライト	• 業務要件、業務仕様は変更せずに、オープン系の言語（Java、C#等）を用いてプログラムを組み直す。
リビルド	• 業務仕様を見直し、且つプログラムやシステム構成の変更を行う。
BPR方式	• 企業改革のために既存の組織やビジネスルールを抜本的に見直し、システムを再構築する。SAPやERPパッケージ導入もBPR方式に当たる。

２．COBOL資産移行のパターン

■　リホスト方式

　現行システムのアプリケーションプログラムを、そのままの言語でオープンシステム環境下に移行して実装させる方法です。
　基本的には、リホストツールを使用してオープン環境上のCOBOL開発環境に移行します。代表的なオープン系COBOL開発・実行環境として
・NETCOBOL(富士通)
・MFCOBOL（マイクロフォーカス）
が挙げられます。

■　リホスト方式（移行イメージ）



２．COBOL資産移行のパターン

■　リホストの特性

メリット

- 現行の資産をそのまま移行するので他の手法に比べると低コストで移行可能概ねリビルドやBPRの対応の５０％程度で対応可能。
- 業務仕様に関する設計作業が不要であり、且つコンバートツールが用意されているため、比較的短期間で移行が可能。
- 業務仕様が変わらないため運用や管理形態はそのままとなる。エンドユーザの視点では、移行時に混乱が少ない。

デメリット

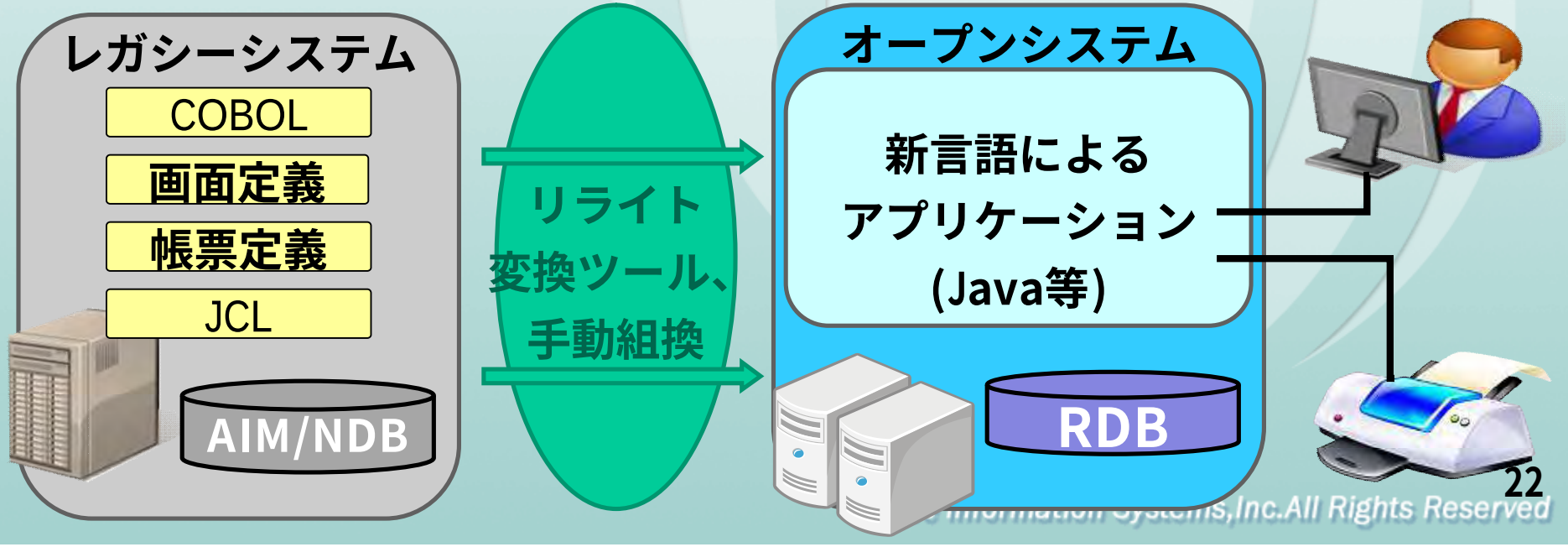
- COBOL言語での移行、リホストツール上のランタイムでの実行形式であるため、オープン系の標準言語であるJavaやC#ほど拡張性はない。
- 現行アプリケーションの保持する問題はそのまま移行される。
- DBのレイアウトもレガシーシステムのまま移行されるので、正規化等今後の見直しが必要な場合が多々ある。

２．COBOL資産移行のパターン

■　リライト方式

現行のソフトウェア仕様（ロジック、データ）はそのままで、オープンシステム環境下にオープン系言語（Java等）でシステムを構築する手法です。
COBOL→オープン系言語は単純変換となることが多く、データ構造、ロジックはCOBOL資産が継承されます。

■　リライト方式（移行イメージ）



２．COBOL資産移行のパターン

■　リライトの特性

メリット

- オープン系の言語、DBで構築し直すため、以後の拡張が容易がとなる。
- Webシステムへの展開やオープン系システムとの連携処理が容易となる。
- 業務仕様に関する設計作業が不要であるため、比較的短期間で移行が可能。

デメリット

- 現行仕様通りの移行であるので、ビジネスバリュー（移行効果）は向上しない。
- 現行の問題はそのまま移行される。
- DBのレイアウトもレガシーシステムのまま移行されるので、正規化等今後の見直しが必要な場合が多々ある。

２．COBOL資産移行のパターン

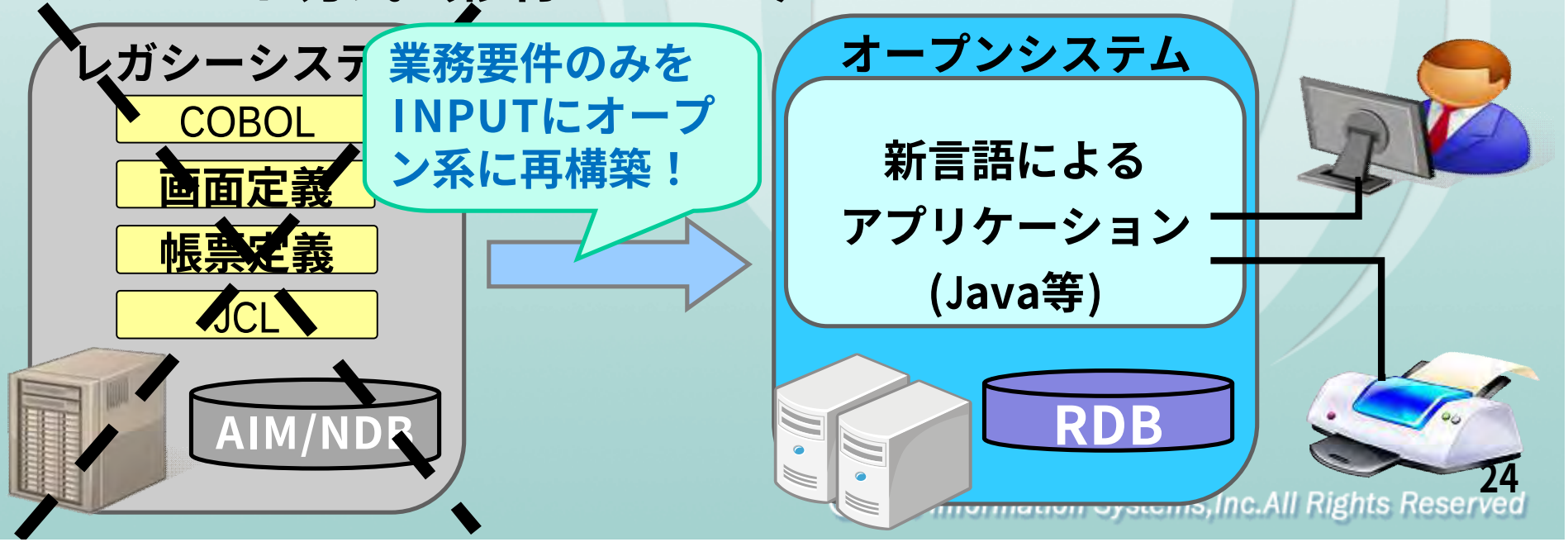
■　リビルド方式

現行の業務仕様をもとにしてオープンシステム環境下に新たなシステムを構築する手法です。

業務要件は変えませんが、業務仕様やシステム構成を設計し直し、オープン系言語にて構築します。

中規模システム（初期開発費用１０００万前後）の場合、リビルド方式での対応が多い傾向にあります。

■　リビルド方式（移行イメージ）

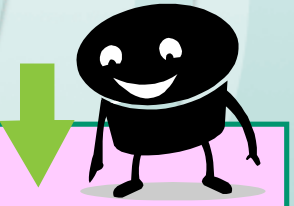


２．COBOL資産移行のパターン

■　リビルドの特性

メリット

- 対応の際に、業務仕様の見直しが行える。
- データ構造の見直しを行うことが可能。（正規化・最適化）
- 最新のアーキテクチャ技術を取り入れて開発可能な為、今後の拡張や展開が容易。



デメリット

- リホスト、リライトに比べ対応コストが掛る。
- 現行ドキュメントの精度が開発コスト、品質に影響する。
- 委託側の人的リソースもリホスト、リライトに比べ占有率が高い。
- データ構造、ロジックを組み直すため、リホスト・リライトに比べ、不具合（バグ）の発生率が高い。

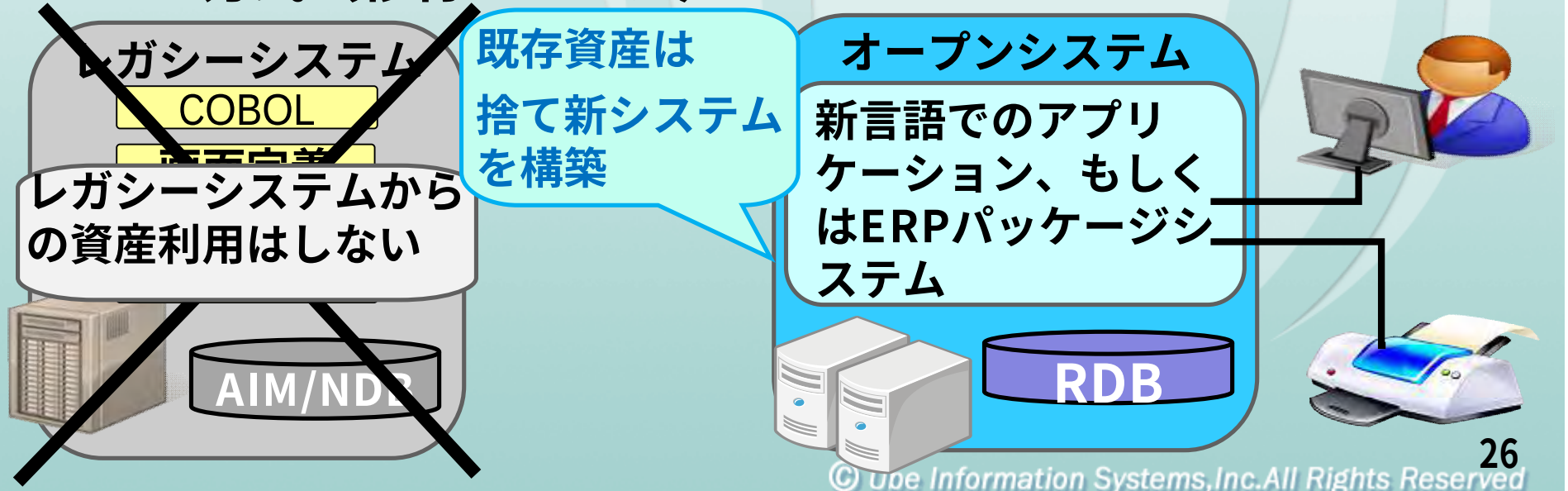
２．COBOL資産移行のパターン

BPR方式

現行システムの業務内容、業務の流れ（ビジネス・プロセス）の最適化を図るために、システムの業務要件から見直し、オープンシステムを構築する手法です。
ERP業務パッケージ導入もBPR方式に含まれます。

COST的な面からERP導入のパターンが多く見られます。

BPR方式（移行イメージ）

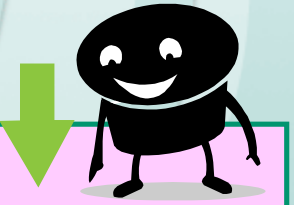


２．COBOL資産移行のパターン

■ BPR方式の特性

メリット

- 対応の際に、業務仕様の見直しが行える。
- データ構造の見直しを行うことが可能。（正規化・最適化）
- 最新のアーキテクチャ技術を取り入れて開発可能な為、今後の拡張や展開が容易。
- 業務要件から見直しているため、導入効果が向上。



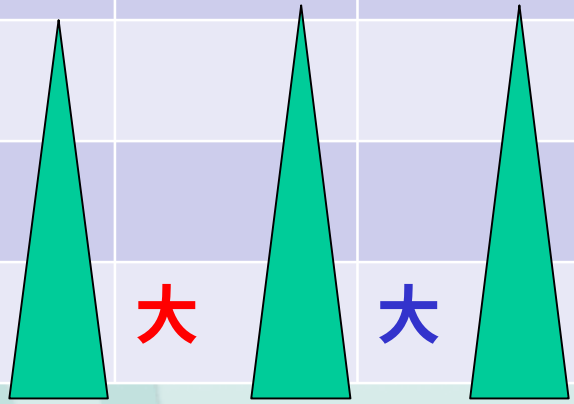
デメリット

- 開発コスト大。
- 検討から導入に関して、人的資源・経営資源を占有する。
- ERPを適用しない場合、データ構造・ロジックを組み直すため、他の方式に比べ、不具合（バグ）の発生率が高い。
- エンドユーザ視点は業務が変わるため、移行時の混乱が大きい。

２．COBOL資産移行のパターン

- 各パターンによる比較

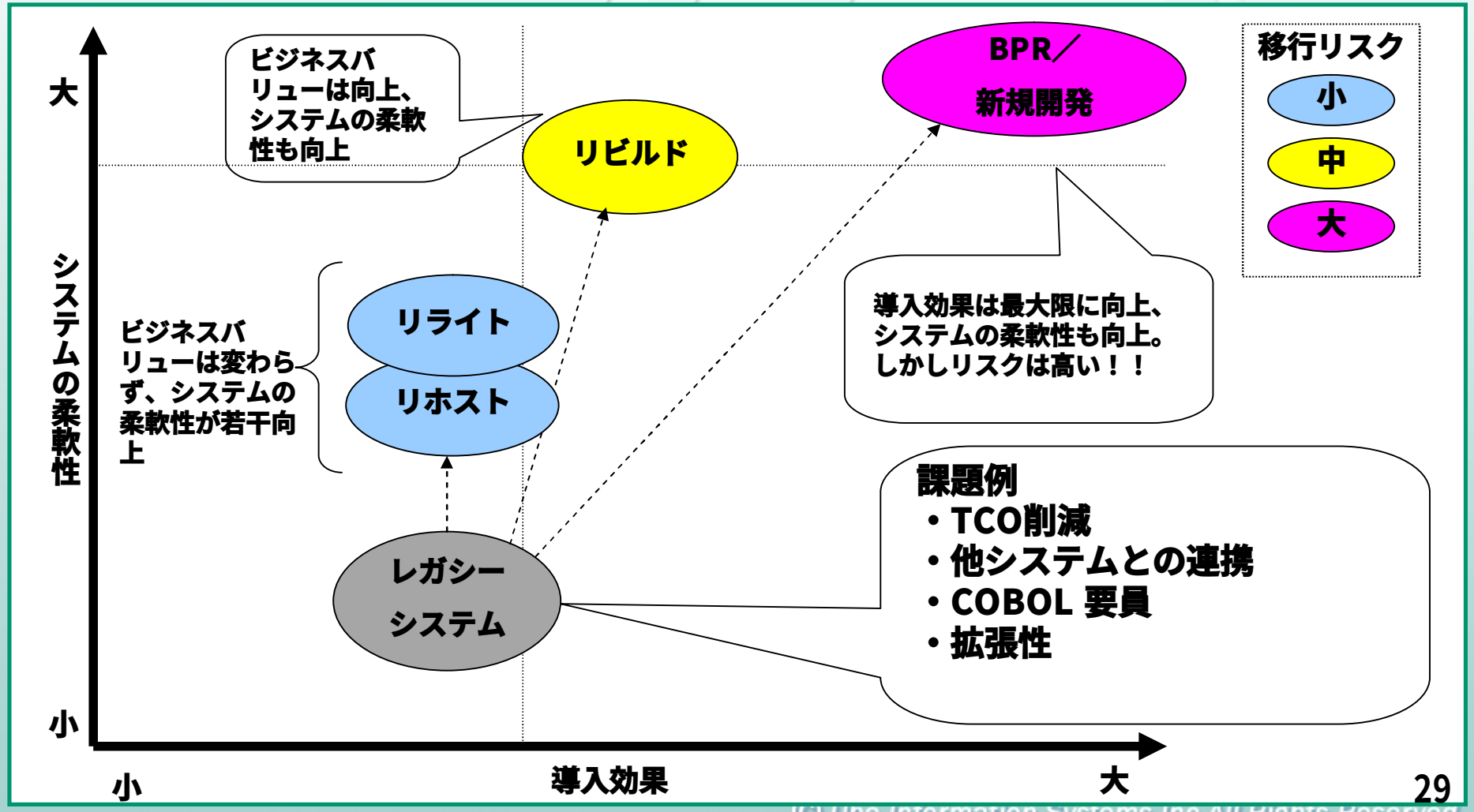
手法	言語	リスク	コスト	柔軟性	導入効果
リホスト	COBOL	小	小	小	不変
リライト	ｵｰﾌﾟﾝ言語				不変
リビルド	ｵｰﾌﾟﾝ言語				向上
BPR	ｵｰﾌﾟﾝ言語	大	大	大	大幅に向上



最終目標をオープン言語、ビジネスバリュー向上とすると、リスクやコストは大きくなる。

2．COBOL資産移行のパターン

■ 各パターンのマッピング
ビジネス・バリュー/システム柔軟性の軸で、マッピング



2．COBOL資産移行のパターン

■ 各パターンの工程別マッピング表

お客様リソースの負荷が高い工程

パターン	工程				
リホスト	業務要件	業務仕様	システム構成	ソフトウェア仕様	アプリケーション
リライト	業務要件	業務仕様	システム構成	ソフトウェア仕様	アプリケーション
リビルド	業務要件	業務仕様	システム構成	ソフトウェア仕様	アプリケーション
BPR	業務要件	業務仕様	システム構成	ソフトウェア仕様	アプリケーション
	現行アプリケーションを変換	現行システムの資産を利用		新規工程	

２．COBOL資産移行のパターン

■ 課題からのアプローチ

＜想像できる課題＞

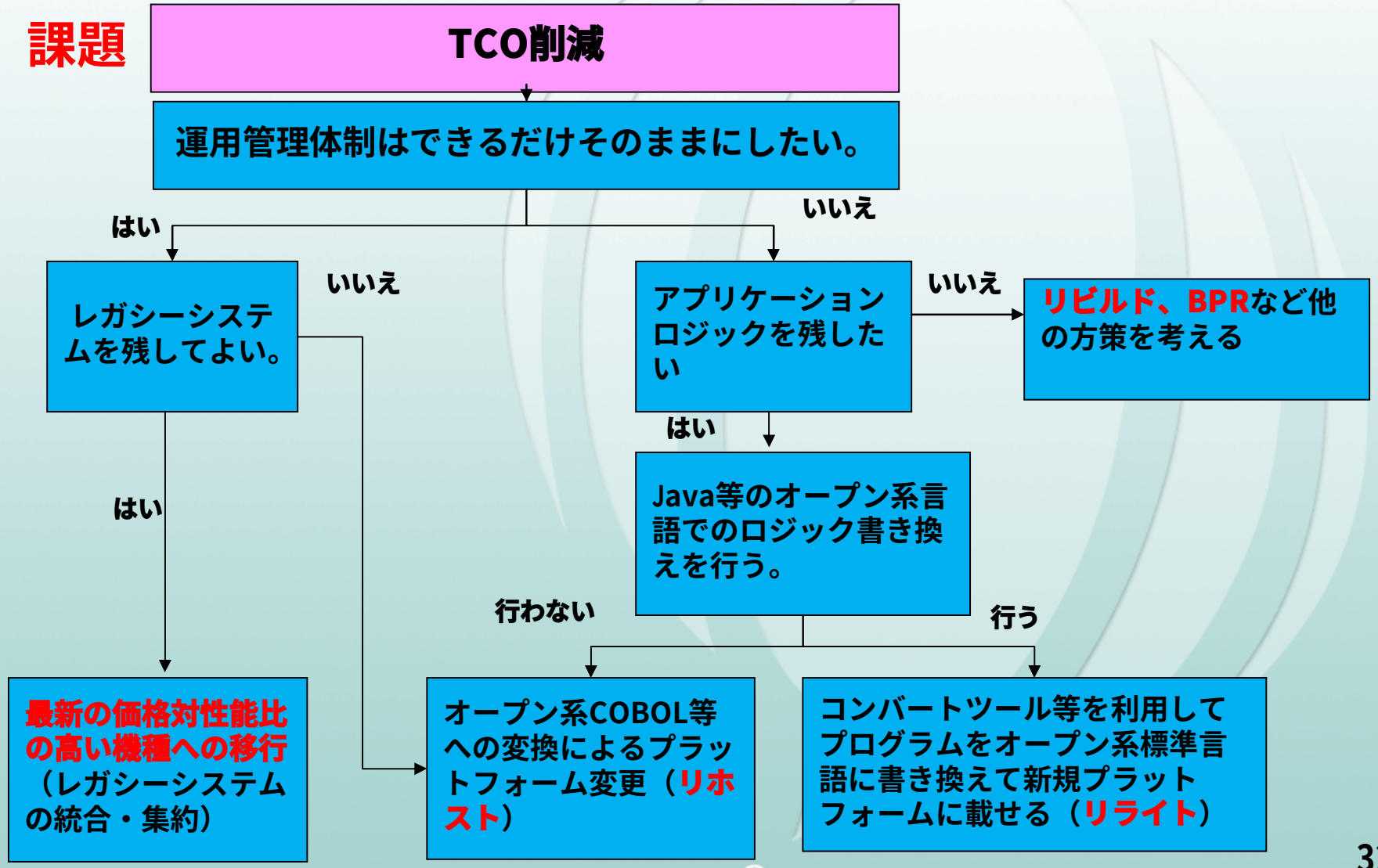
- TCO削減
- ITシステムの強化
- 要員の問題
- 情報共有（リアルタイム性）
- 外部システムとの連携

課題毎に対応方式を検討

その後、サマリしてパターンを決定

２．COBOL資産移行のパターン

■ 検討チャート（サンプル）



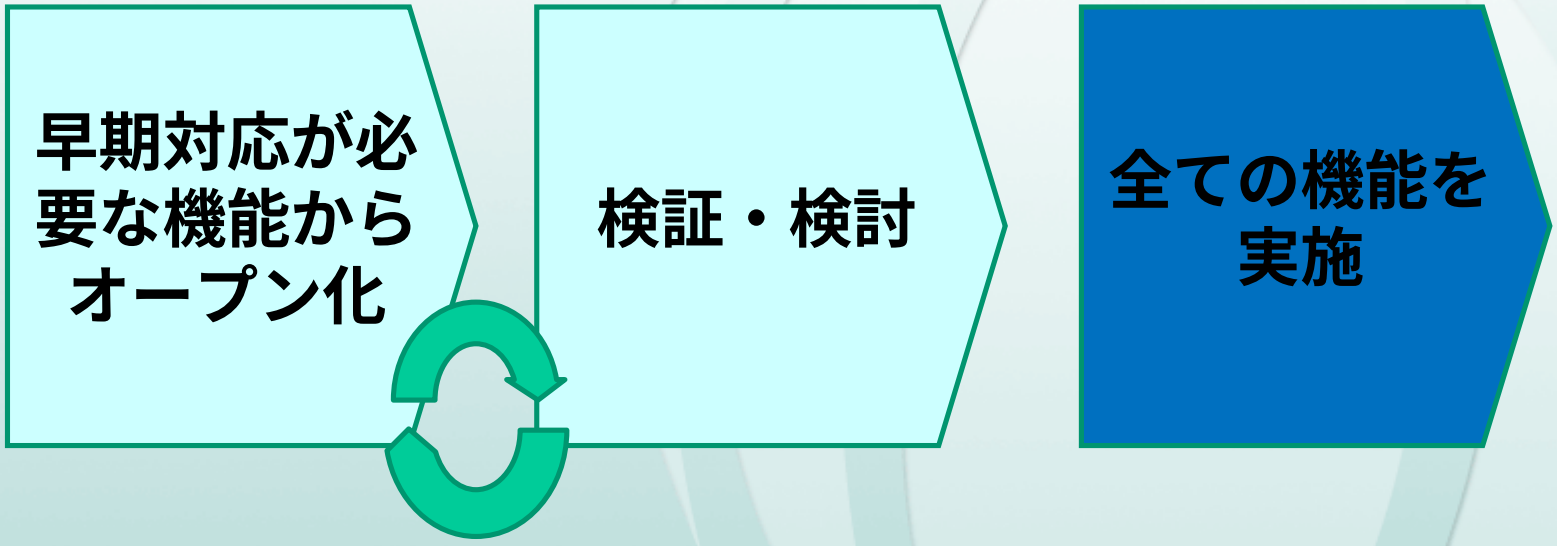
２．COBOL資産移行のパターン

■ 検討チャート（サンプル）

２．COBOL資産移行のパターン

■ 戦略面からのアプローチ

＜中・長期的に対応＞：スモールスタート



・リビルド方式での対応が多くみられる

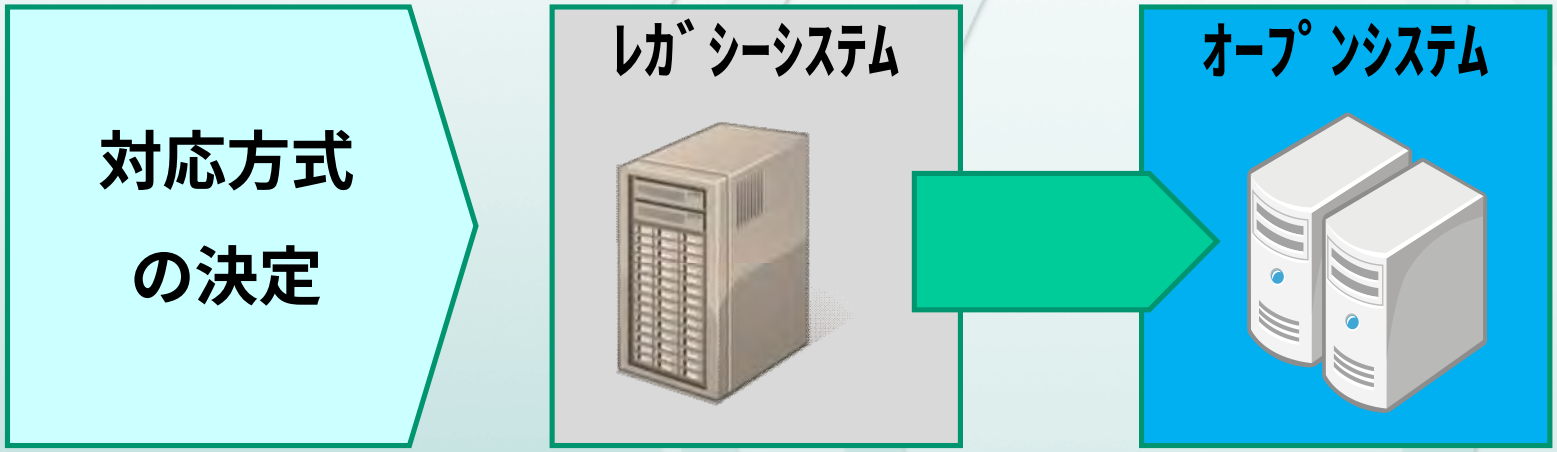
・移行リスクは減るが、メインフレーム、オフコン撤廃をゴールとすると対応に時間がかかる。また、トータルコストは大きくなる。

・ERPパッケージ適用はできない。

２．COBOL資産移行のパターン

■ 戦略面からのアプローチ

＜短期的に対応＞



- 短期間で移行できるが、リスクが大きい
- 対応方式の決定が最も重要なフェーズ。開発に着手すると後戻りが困難。
- リライト、リホストでの対応の場合、言語や効果の課題が残るため、数年後に再構築されるお客様も少なくない。

２．COBOL資産移行のパターン

- オープン化に向けて

レガシーユーザのオープン化については、各社のIT戦略、システム構成、背景（要員、競合他社）により、検討内容や対応方法も様々であります。

それでは、オープン化対応された他社の事例を見てみましょう。

３．UIS事例紹介

３．UIS事例紹介（事例①リホスト）

ﾌﾟﾛﾌｧｲﾙ

窯業系建材製品の製造、販売、並びに責任施工
その他建設関連商品の販売並びに責任施工
従業員規模　220名

導入背景

財務・販売・物流システムはメインフレーム上で構築されており、TCO削減を重要課題とし、オープン化に向け検討してきた。

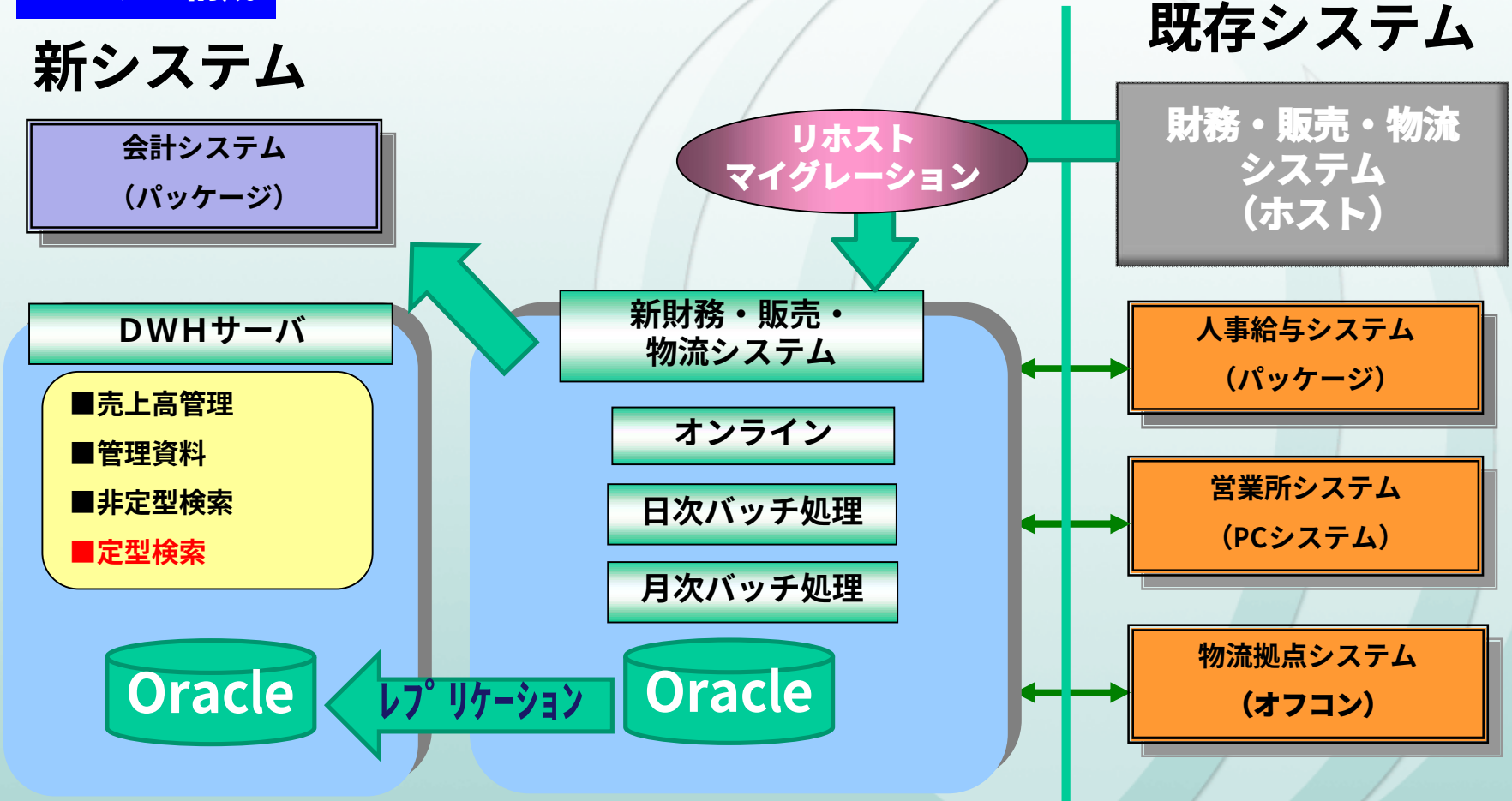
初期投資並びに運行費用については可能な限り削減する前提で検討した結果、**リホスト方式**での開発となった。

ｼｽﾃﾑ化方針

■現行のシステム資産を原則、有効活用し機能を継承しつつ、脱ホスト化を実現する。但し、必要最小限の梃入れは実施する。
■既存の生産系システムと今回脱ホスト化するシステム間で必要な情報はシステム的に連携する。
■データを分析・加工し易いシステム環境を提供して、情報の有効活用を図ると共に管理間接部門の生産性向上を支援する。

３．UIS事例紹介（事例①リホスト）

システム構成

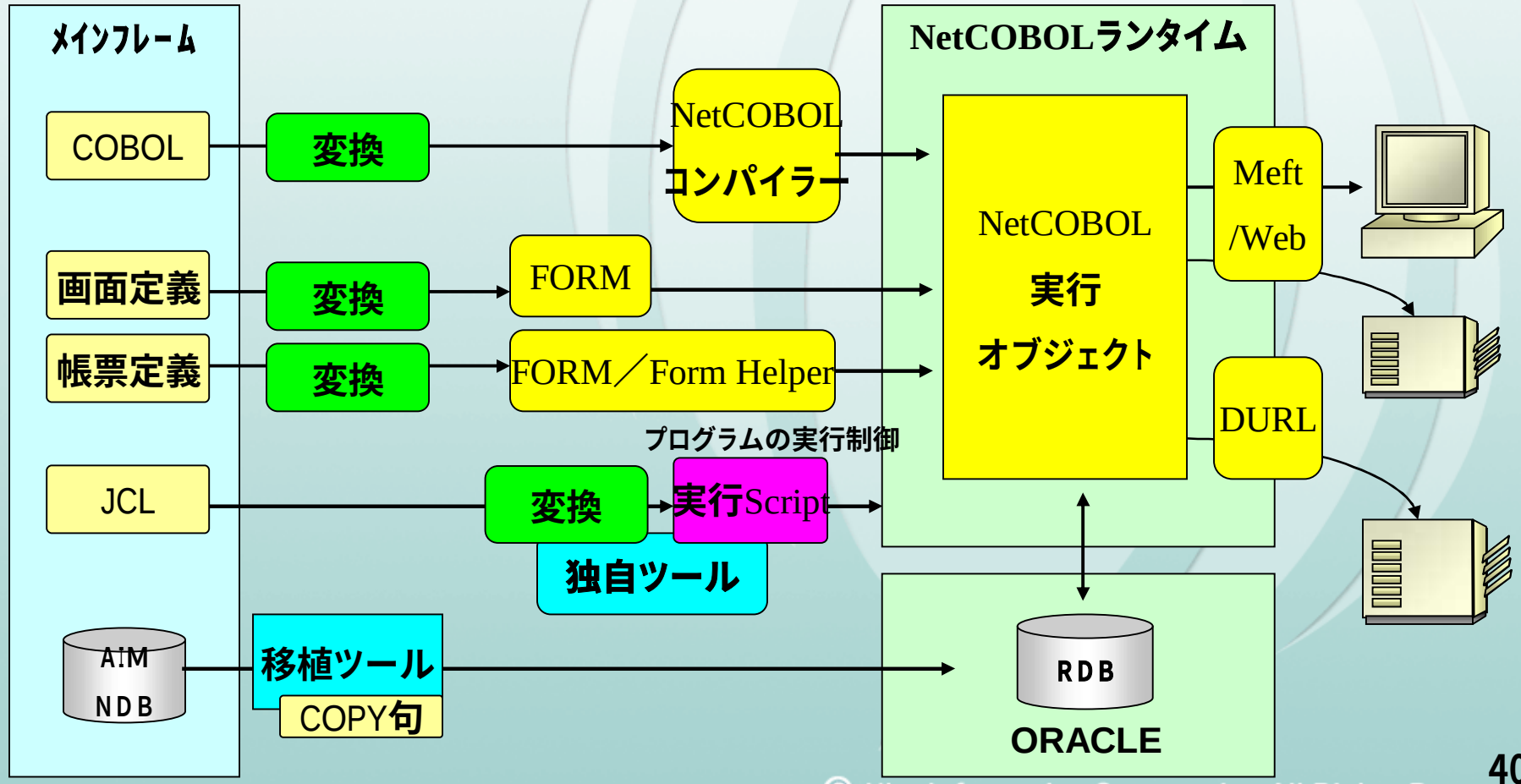


マイグレーションと同時にﾃﾞｰﾀｳｪｱﾊｳｽ（DWH）、会計パッケージを導入

３．UIS事例紹介（事例①リホスト）

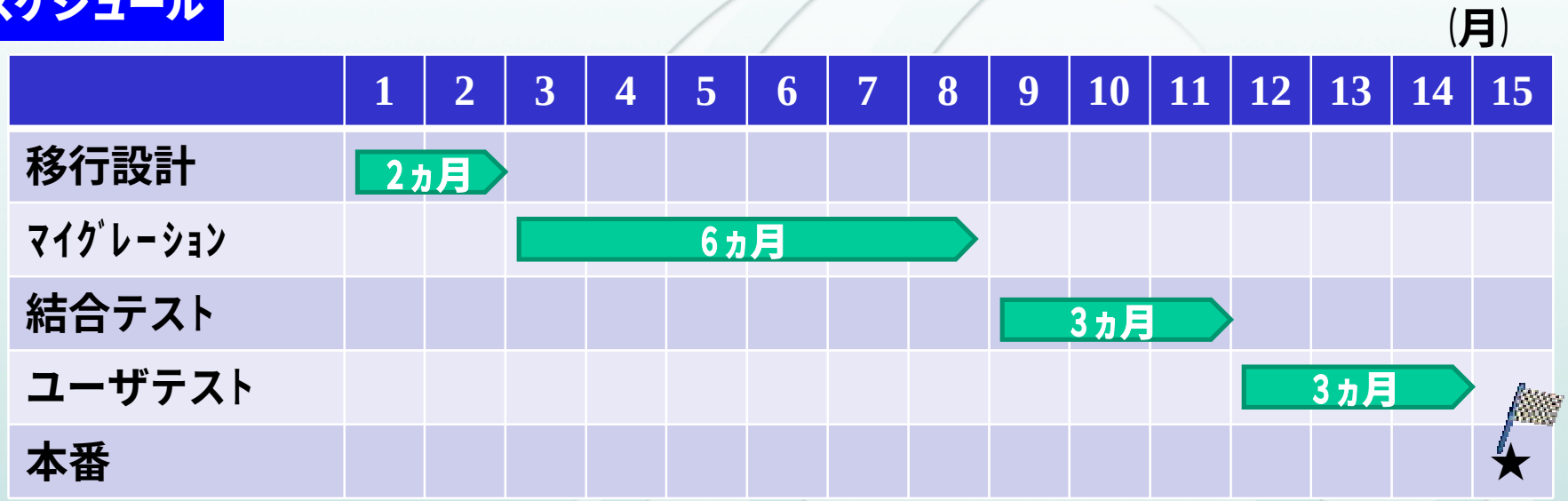
NetCOBOL による移行方式概要

メインフレームの各資源をNetCOBOLの提供するツール及び独自開発を行なったツールを基に、オープン環境（Windowsサーバ）へマイグレーションします。



３．UIS事例紹介（事例①リホスト）

スケジュール



導入効果

■ＴＣＯの削減
■バッチ処理レスポンスの改善。約２０分の１に。
■ＲＤＢ、ＤＷＨサーバ導入によりリアルタイムに情報が取得可能に。管理・分析レベルの向上
■経理システム導入により、仕訳、原価管理の手動業務軽減

３．UIS事例紹介（事例②リビルド）

ﾌﾟﾛﾌｧｲﾙ

コンテナ輸送事業、自社船舶の配船・配乗・メンテナンス、
港湾内での倉庫保管、管理等、船舶に係る物流事業。
従業員規模　３８０名

導入背景

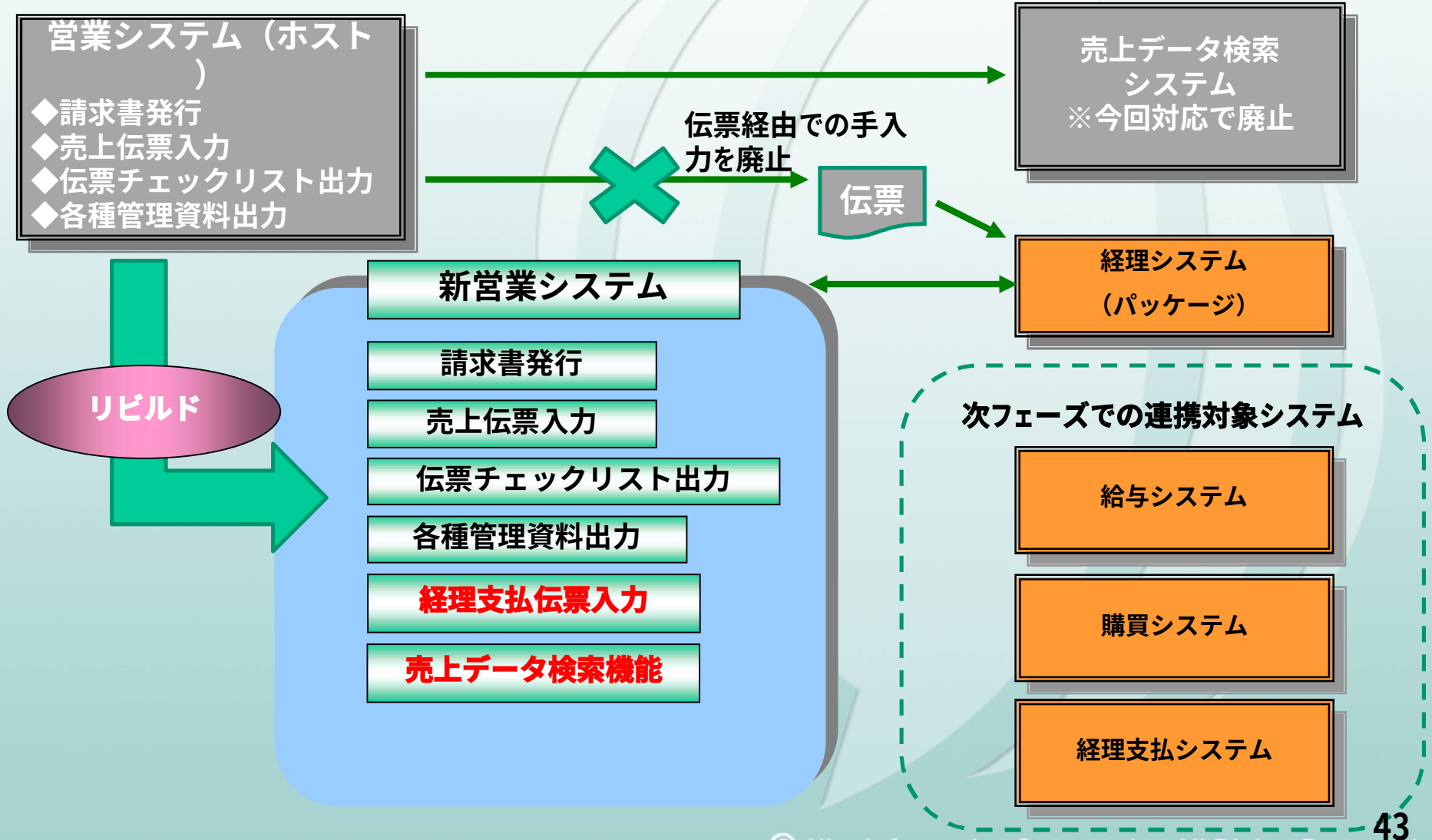
ホスト営業システムの陳腐化により、システムの有効利用がなされていない背景から、効率的な業務運営を実現するために、営業システムの再構築を検討してきた。パッケージソフト導入による営業システムの再構築を検討したが、業務に合ったパッケージソフトが選定できなかったため、既存ホスト営業システムを基本にし、各担当者からの要望としてあがった機能を含めてクライアント／サーバ型システムの構築（**リビルド方式**）を行う事となった。

ｼｽﾃﾑ化方針

■既存のホスト営業システムの機能を基本とし、要望としてあがった「支払管理」「予算実績管理」「自動仕訳経理連携」の各機能を追加する。
■必要な情報がよりオープンな形で、より速く取り扱え、フレキシブルなデータ活用を可能にする。
■投資コスト、運用コストを可能な限り抑える。

３．UIS事例紹介（事例②リビルド）

システム構成



３．UIS事例紹介（事例②リビルド）

スケジュール

（月）

	1	2	3	4	5	6	7
業務仕様設計	2ヵ月						
実装			2ヵ月				
結合テスト					1ヵ月		
ユーザテスト						1ヵ月	
本番							★

導入効果

■手作業で集計や配信処理を行っていた、「支払管理」「予算実績管理」「仕訳経理連携」の自動化。
■業務仕様の見直しにより、出力帳票を削減。（約半数に）

３．UIS事例紹介（事例③BPR）

ﾌﾟﾛﾌｧｲﾙ

化学樹脂製品の製造・販売
従業員規模　160名

導入背景

オフコンの老朽化、ＴＣＯ削減の問題があり、例年検討を行っていたが、費用の問題やＥＲＰ選定に時間を要していた。

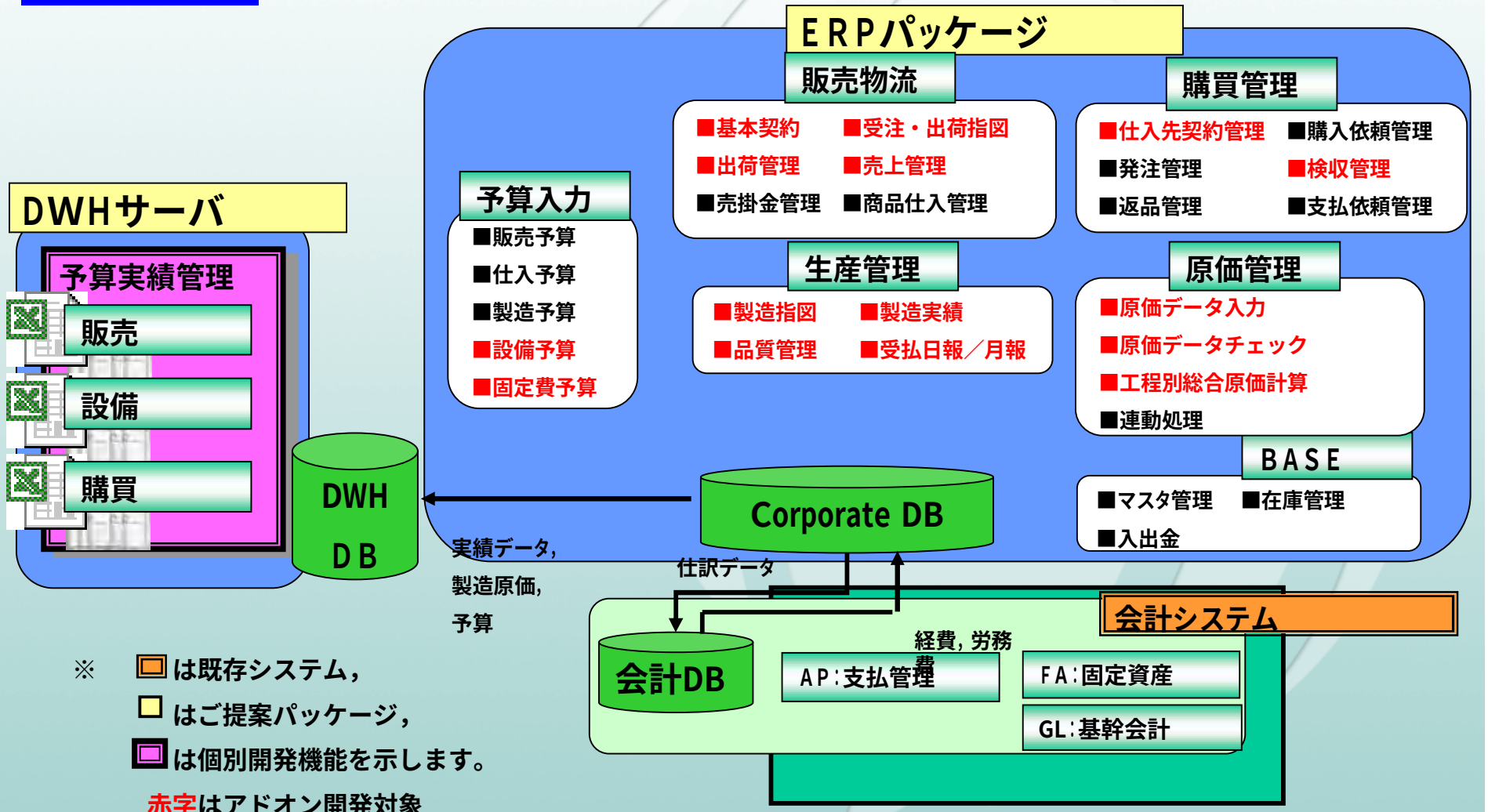
グループ企業同士での基幹システム共同運用計画（各社での脱ホスト化）へ参画し**ＥＲＰ業務パッケージを導入**することでシステム構築・維持管理コストの低減を図るとともに、オープンシステム化の実現に至った。

ｼｽﾃﾑ化方針

■ＥＲＰパッケージの機能を最大限に利用し現行システムの不足部分を補充する。
■基本的にパッケージ機能を流用するが、戦略的に差別化が必要な機能については、アドオン開発にて追加対応をおこなう。
■データを分析・加工し易いシステム環境を提供して、情報の有効活用を図ると共に管理間接部門の生産性向上を支援する。

３．UIS事例紹介（事例③BPR）

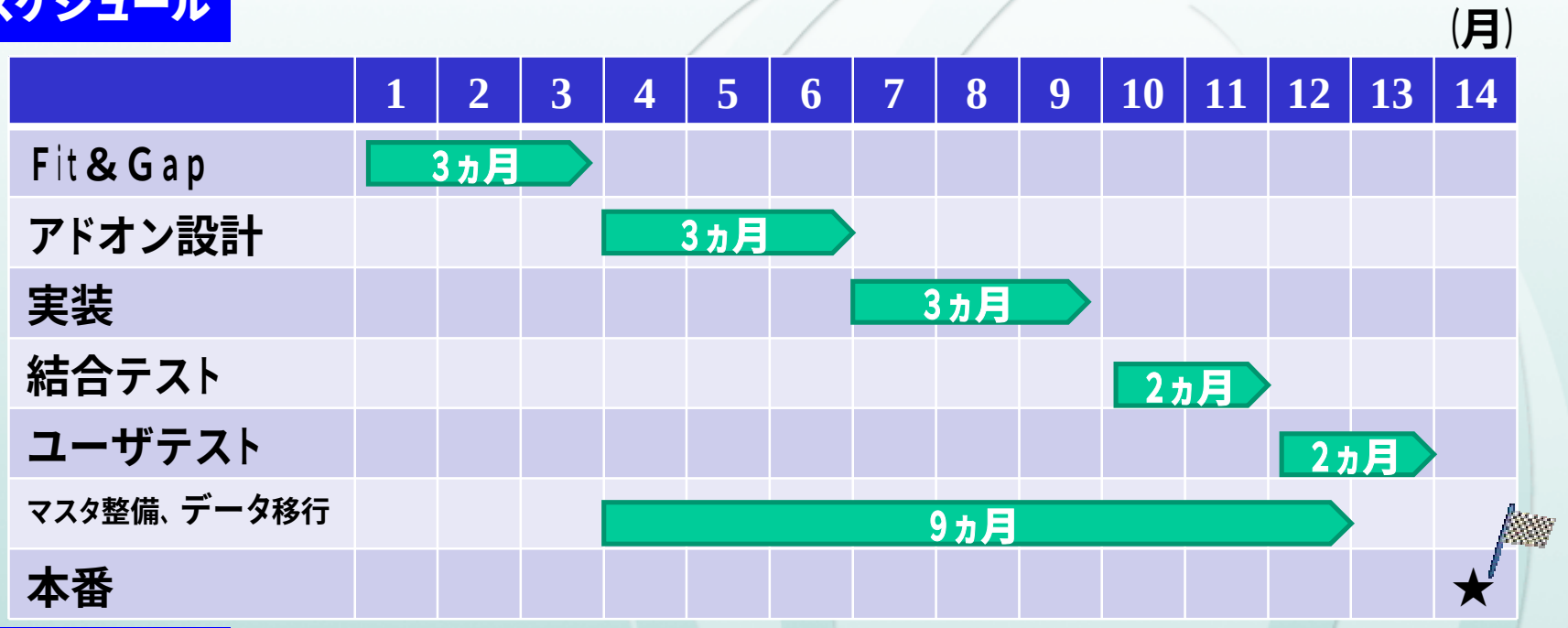
システム構成



※ ▭ は既存システム，
□ はご提案パッケージ，
▣ は個別開発機能を示します。
赤字はアドオン開発対象

３．UIS事例紹介（事例③BPR）

スケジュール



導入効果

■ 2重入力業務の廃止
■不安定な稼働状況から、安定した稼働状況へ
■ＲＤＢ、ＤＷＨサーバ導入によりリアルタイムに情報が取得可能に。また、原価管理機能の導入により、製品ごとの管理が可能となった。
■他システムとの連携が容易となったため、品質管理等の追加開発に拡張性が広がった

今後にむけて

■ ＵＩＳのできること

- 移行方式
- 事例紹介
- 検討サポート
- 概算調査
- 実現
- 維持管理

今回のほか、ホスト・オフコンを残す手法、クラウドサービス利用手法等があります。

SAP導入等、多々事例があります。

御社業務や特定業務に詳しい要員が対応します。

新システム構成で
必要な費用を概算します。

開発、パッケージ導入、環境構築等、各種専門担当が最適なシステム構築を実現します。

安定したシステム運用を行います。

今後にむけて

- UISからのお願い

御社のIT戦略にあったシステム開発に是非ご協力させてください。御社の永続的な発展に向けよきビジネスパートナーとなれることを深く願ってます。