日立グループ100周年
100th
確かな技術でつぎの100年へ

HITACHI
Inspire the Next

既存資産を蘇らせる技術 「レガシーマイグレーション」の最新動向

2010年 5月 12日

株式会社日立製作所　情報・通信システム社
ソフトウェア事業部　IT基盤ソフトウェア本部
レガシーマイグレーション推進センタ
佐藤　一浩

uVALUE

HITACHI
Inspire the Next

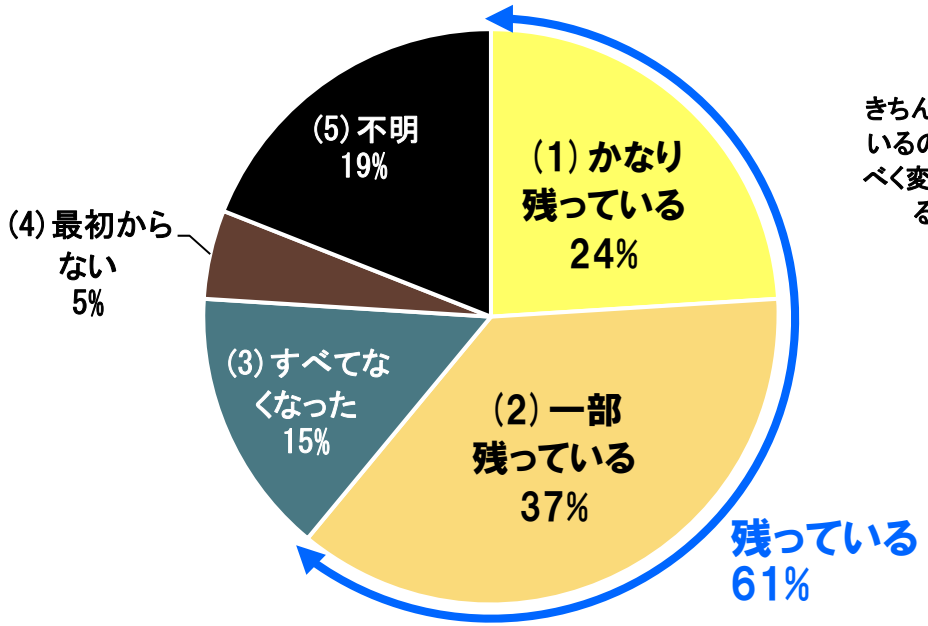
Contents

uVALUE

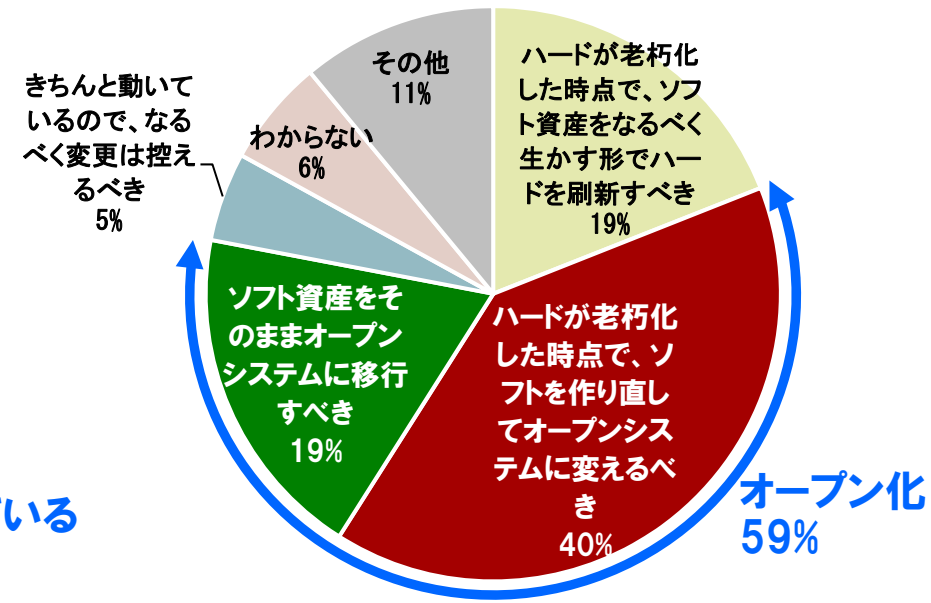
HITACHI
Inspire the Next

レガシーシステムの保有状況と今後の方向性

出典：日経コンピュータ 2010年2月3日号
読者の声



御社のレガシーシステムはどの程度残っていますか

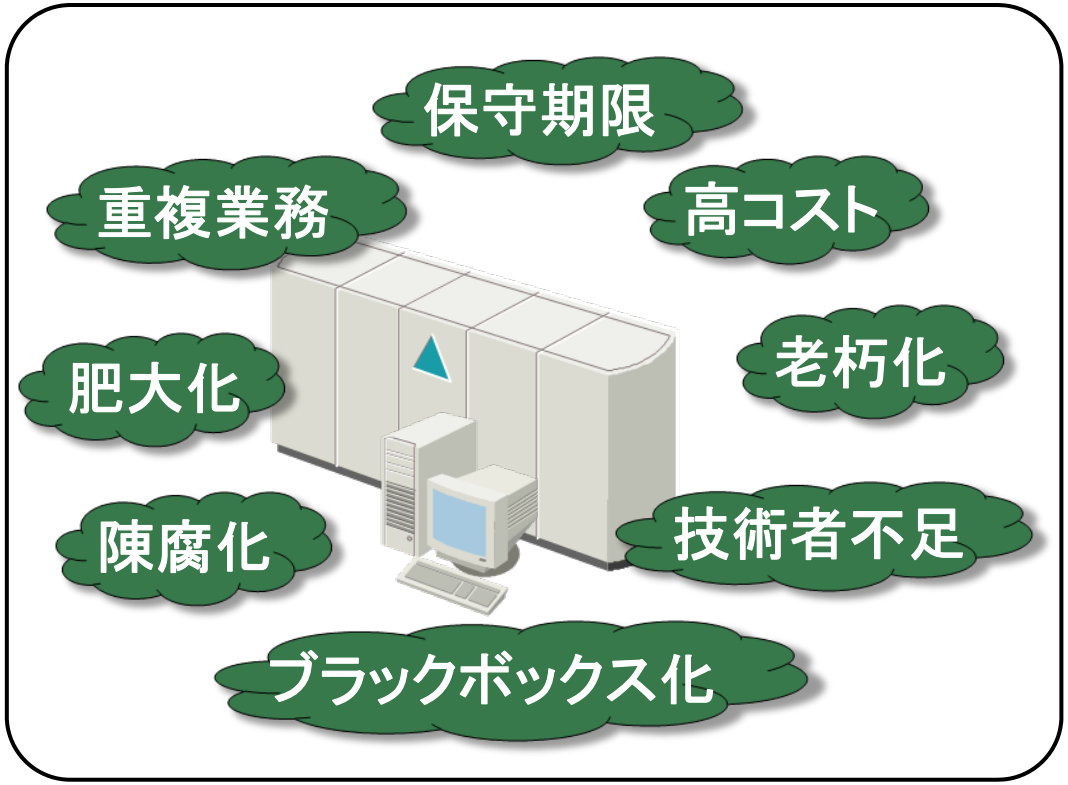


レガシーシステムはどうすべきだと思いますか
（左図で、(1)～(3) 選択者の回答）

- レガシーシステムが残っている ：61%
- オープンシステムに移行すべき：59%
- オープンシステムのレガシー化が深刻との声もある

HITACHI
Inspire the Next

メインフレーム(MF)／オープンシステムに限らず、
既存システム(レガシーシステム)の定期的な再構築が必要



既存システム見直しのきっかけ

システム再構築

- MF → MF
- MF → オープン
- オープン → オープン

1-3. 再構築システムに求められる要件

HITACHI
Inspire the Next

(1)適正なITコスト
　　再構築費用だけでなく、将来に渡るランニングコストの最適化

(2)再構築期間
　　既存システム運用に影響を与えない期間でシステムを構築

(3)信頼性
　　ハード、ミドルウェアの信頼性、アプリケーションの安定稼動

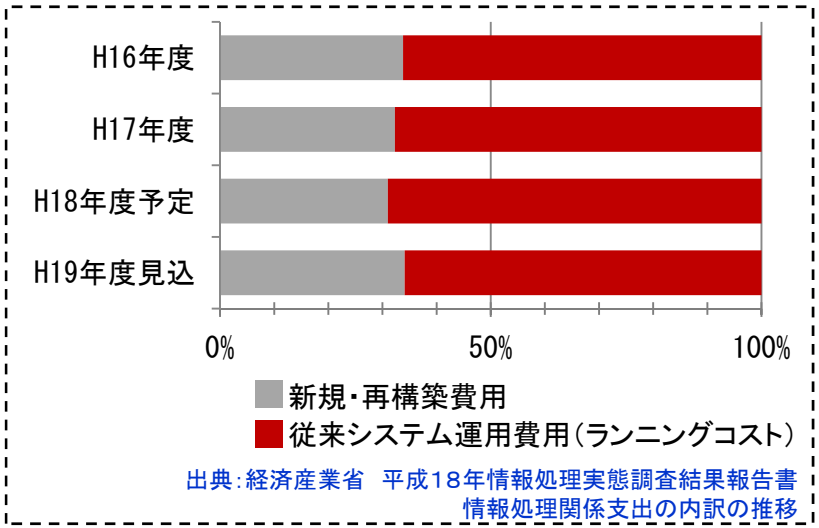
(4)柔軟性
　　戦略的機能追加に即座に対応できる柔軟なシステム

(5)資産活用
　　ノウハウが蓄積されている業務仕様の活用

1-4. 企業情報システムにおけるITコストの現状

HITACHI
Inspire the Next

IT関連のコスト構造の現状



出典：経済産業省　平成18年情報処理実態調査結果報告書
情報処理関係支出の内訳の推移

IT予算の60%以上が
「ランニングコスト」

企業が勝ち残っていくためには、

「戦略性の高い
新規システムへの投資を
続けていかなければならない」

ITにおける最大の経営課題＝ITコストの構造改革

「ランニングコストの削減」と「ITコストの適正化」

1-5.【CASE1】安易なコスト削減の落とし穴

HITACHI
Inspire the Next

メインフレームシステム〔約1,000本：1Mstep〕

一気にコスト削減を目論んで…

「ERP」による全面再構築を採用
全面アウトソーシングを実施

ところが．．．

運用メンテナンスに手間　→　1桁変えるのに200万円
ERPのバージョンアップ　→　数億円

業務を改善することが
殆ど不可能…

やはり、レガシー資産を活かすことに
〔COBOL資産再構築〕

ポイント1

新しいことをやり続けなければならない
将来に渡るランニングコストの見落とし

1-6.【CASE2】信じられない「不当な」コスト

HITACHI
Inspire the Next

2001年6月：合併吸収により、営業店が増加

これに伴い、営業店コードも増加…

現状システムでは、営業店コードの桁数が不足

社長命令の下、取り組みを開始

現行システム
メインフレームシステム〔約7,500本：5Mstep〕

システム改善作業の前の
ソース調査に…

顧客による見積り・・・調査作業のみで1億円〔2年：120人月〕

桁数増やすのに、なぜそんなにかかるのか、経営者は信じられない

ポイント2

新しいことをやるためのコストを
適正化することが本当のコスト削減

1-7. 現行システムの構造改革の必要性

HITACHI
Inspire the Next

ポイント1 新しいことをやり続けなければならない将来に渡るランニングコストの見落とし

ポイント2 新しいことをやるためのコストを適正化することが本当のコスト削減

現行システムの構造を改革 ⇒ ITコストの構造改革を実現

- 現行システムの蘇生により、ランニングコストを削減し、
- 経営スピードに負けない「変化即応型システム」へ構造を改革
  - 将来に渡るランニングコストを抑えながら、
  - ビジネススピードや経営環境の変化に即応し、
  - 戦略的機能をクイックに拡張することができる
- 半永久的に成長し続ける真の企業情報システムに生まれ変わる

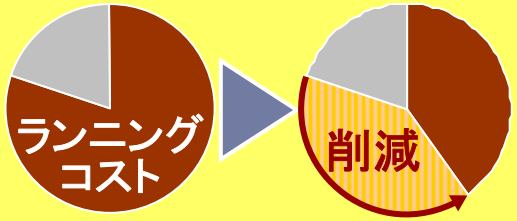
1-8. コスト最適化のステップ

HITACHI
Inspire the Next

将来に渡るコスト削減と、新しいことをやるためのコスト適正化

既存システムの再構築

既存システムの蘇生
および運用改善
（オープン化含む）

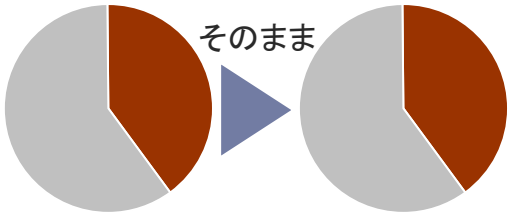


直近の
ランニングコストを
削減

【マイグレーション】

変化即応型システム

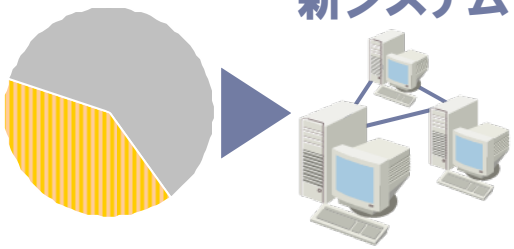
変化即応型システムへ
構造改革



将来に渡る
ランニングコストも
抑止

戦略的機能の拡張

削減したIT予算を
戦略投資

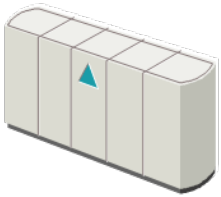


変化即応型システムへ
戦略的機能を拡張

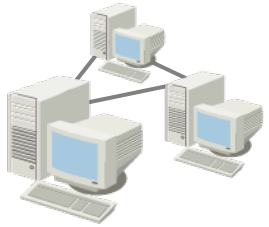
1-9. メンフレームとオープンシステムの比較

HITACHI
Inspire the Next

メインフレームをオープン化する場合は、ハード／ソフト費用以外の項目についても検討が必要



メインフレーム（MF）が優位 ← → オープンシステムが優位



信頼性
MFは絶対的な信頼性がある
オープンは台数も多く全体ではトラブル多い

安定した性能
入出力専用プロセサなど
MFは性能が安定している
オープンは性能が変動しやすい

ハードウェア費用
ソフトウェア費用
一般的なオープン化のきっかけ

トラブル対応
MFは特定ベンダのため
トラブル対応が早く、
原因究明しやすい
オープンは複数ベンダのため
原因不明のケースもあり

ベンダ選択の自由度
同一アーキテクチャでも
複数のベンダから選択できる

運用管理
MFは台数少なく運用が容易
オープンは複雑な運用となる

新技術への対応
オープンは、
最新技術への対応が容易

ライフサイクル
MFは10年程度
オープンは5年が一般的で
リプレース間隔が短い

変化への対応
オープンはビジネスへの
変化などに対応しやすい

後継機種の互換性
MFは完全互換性が一般的で
後継機種への移行が容易
オープンは互換性無い場合多い

技術者数
オープンは若い技術者が多い

uVALUE

HITACHI
Inspire the Next

Contents

uVALUE

HITACHI
Inspire the Next

システム再構築には、大きく3つの論理的再構築パターンがある

再構築パターン	手法・特徴
スクラッチ開発	・既存のソフトウェア資産を捨てて、要件定義から開発を行なう。 ・一般には、基本設計、詳細設計、コーディング、テストと工程を踏んで開発を進めるウォータフォール型の開発をすることが多い。
パッケージ適用	・パッケージの持つ機能とお客様要求仕様の適合率の確認を行なう。(FIT＆GAP) ・GAP部分はアドオン開発を行なうか、業務を変更する必要がある。基本的には業務をパッケージに合わせる。 ・クラウドサービスや、SaaS利用のケースもある。
レガシーマイグレーション（資産再利用）	・既存のプログラムを改修して新たなプラットフォームに移行する。 ・改修はミドルウェアが変更になったことへの対応等、必要最小限の範囲で実施する。 ・新たな機能はマイグレーション完了後、プログラムを改修する。

2-2. システム再構築パターンの比較

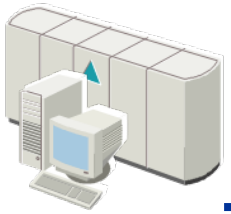
HITACHI
Inspire the Next

再構築パターン	メリット	デメリット
スクラッチ開発	・技術的に実現可能であれば、お客様の要求仕様を100%取り込むことが出来る。（最も自由度が高い） ・インフラ（ハード・ソフト）の選択もお客様のご希望を取り入れることが容易である。	・開発期間が最も長く、生産性も低い。 ・開発コストは最も高価である。 ・出来上がるプログラムの品質・完成度は最も悪い。安定稼動まで長期間を要する。 ・最も開発人員を必要とする。
パッケージ適用	・FIT&GAPの結果、適用率が高ければ短納期・安価にて構築ができる。（概ね80%以上の適用率） ・アドオン部分を除き、プログラムの完成度は最も高い。 ・新たなOSへの対応は、パッケージ開発ベンダーが実施する。	・自由度は最も低い。BPRが前提となる。 ・パッケージ自体をカスタマイズした場合、開発ベンダーのサポートが受けられない場合がある。 ・インフラに制限がある。
レガシーマイグレーション（資産再利用）	・レガシーマイグレーションの完了までは、スクラッチ開発に比べて短期間で完了できる。但し、改修（新規要件対応）は別途スケジュールを立てる必要がある。 ・既存資産を活用する為、プログラムの完成度は現状と同等である。	・新規要件は移行後改修する必要がある。 ・インフラは、移行ツールにより制限を受ける。 ・お客様のテスト工数の削減は難しい。

2-3. システム再構築パターンの選択の流れ

HITACHI
Inspire the Next

現行
システム資産



プログラム
JCL
DB
ファイル
・・・

システム再構築構想立案フェーズ

システム再構築へのアプローチには、
構築パターンに関わらずこのフェーズが
非常に重要

Step1
仕様情報再生

- 資産棚卸サービス
- 仕様回復サービス

Step2
再利用性分析

- 移行性分析サービス

判断

「新規」、「現行資産再利用」の判断

- 「現行資産再利用」を選択した方が「コスト」、「品質」、「期間」において有利
- 判断基準〔実績ベース〕
  改造部分≧20～30%→新規再構築
  改造部分＜20～30%→現行資産再利用
- 改造箇所により上記判断基準は異なる

システム再構築フェーズ

新規
再構築

現行資産
再利用

判断

スクラッチ
開発

パッケージ
適用

レガシー
マイグレーション

- マイグレーションサービス

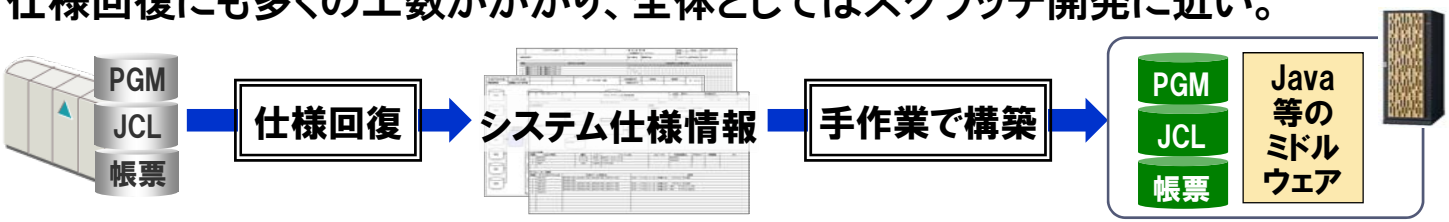
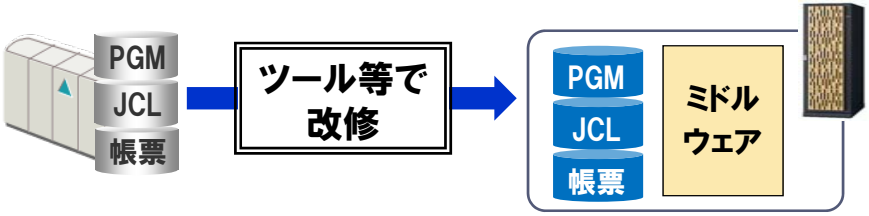
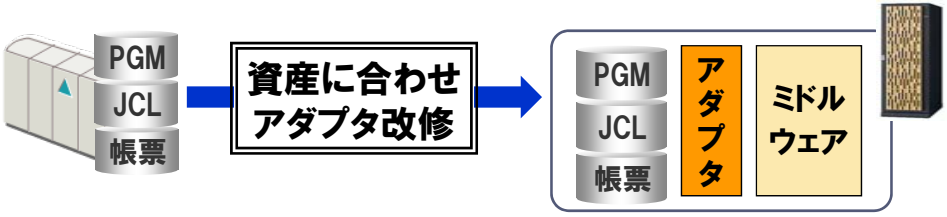
「スクラッチ」、「パッケージ」の判断

- 現状業務・システムとパッケージ仕様との適合性、現行システムの独自性により判断
- Step1～Step2＋パッケージ仕様情報をもとに判断
- 判断基準は「イニシャルコスト」と、「ランニングコスト」

uVALUE

HITACHI
Inspire the Next

現行システム資産を再利用する場合でも、大きく3つのパターンがある

利用方法	手法・特徴
仕様情報のみ利用	・既存システムの業務仕様はそのままで、Java™などで再構築。 ・仕様回復にも多くの工数がかかり、全体としてはスクラッチ開発に近い。 
資産を移行先に合わせる	・既存システム資産を、移行先のミドルウェアに合わせ改修する。 ・マイグレーションを考慮したミドルウェアも多く、移行性は向上している。 
資産をそのまま利用	・既存システム資産(COBOLプログラムやJCLなど)は基本的に変更しない。 ・マイグレーション専用ミドルウェアやアダプタ開発で、既存資産を実行。 

2-5. 仕様情報のみ利用する方式

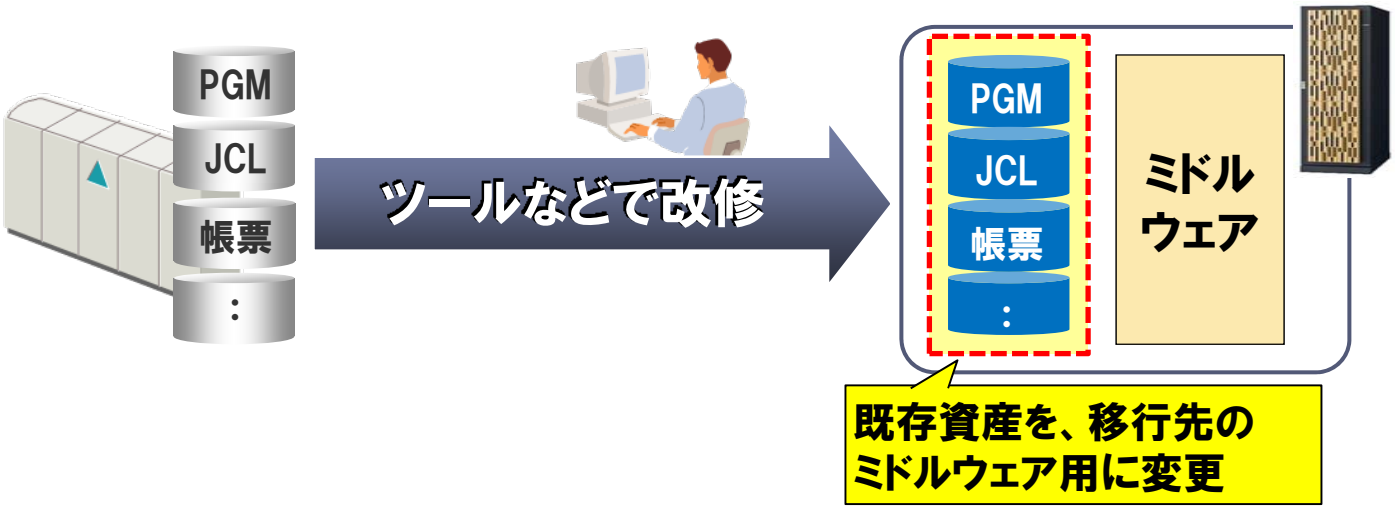
HITACHI
Inspire the Next

- 既存システムの業務仕様はそのままで、Javaなどで再構築
- 仕様回復にも多くの工数がかかり、全体としてはスクラッチ開発に近い

メリット	•業務仕様を継続したまま、Javaなど最新技術でシステム構築が可能。 •インフラ(ハード・ソフト)や開発言語の選択も、お客様のご希望を取り入れることが容易である。
デメリット	•業務仕様回復にも多くの工数がかかり、開発期間が長くなる。 •生産性が低く、開発コストが高くなる。 •出来上がるプログラムの品質・完成度が低く、安定稼動まで長期間を要する。 •多くの開発人員を必要とする。

2-6. 資産を移行先に合わせる方式

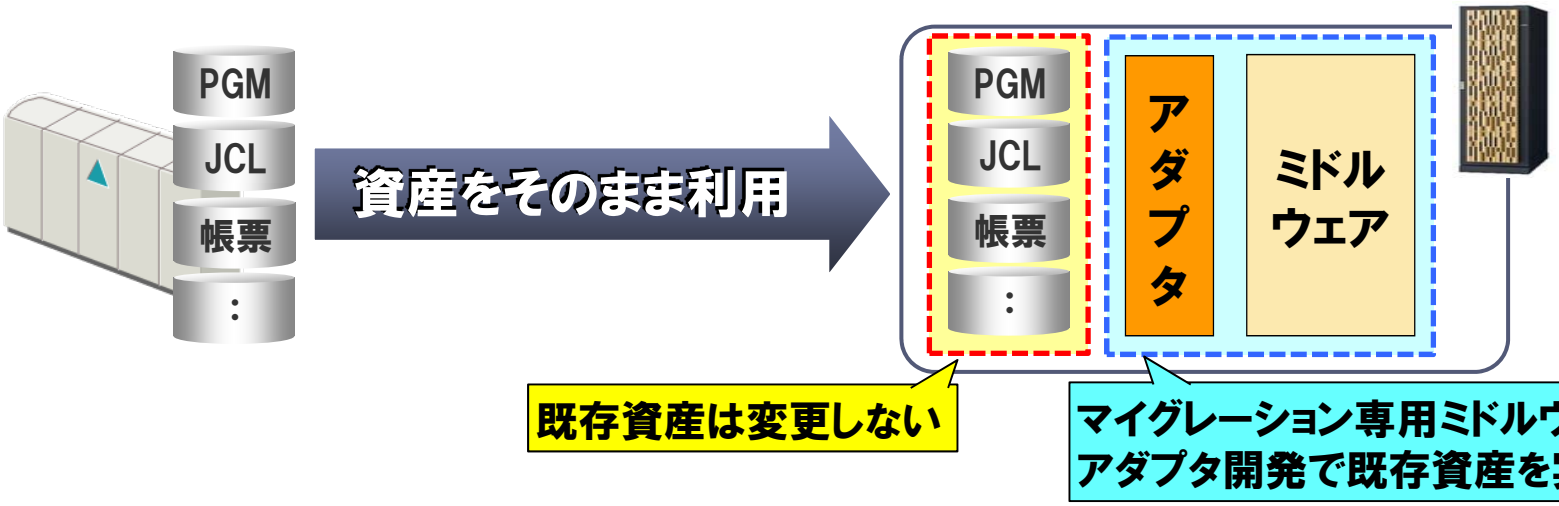
- 既存システム資産を、移行先のミドルウェアに合わせ改修する
- マイグレーションを考慮したミドルウェアも多く、移行性は向上している



メリット	•業務仕様を継続できる。現行資産の活用で、短期間で移行できる。 •実績が多いミドルウェアを活用できる。 •オープン化後、他のミドルウェアとの組合せで、戦略的機能追加が可能。 •移行ツールや移行ソリューションが整備されている。 •マイグレーションを考慮したミドルウェアも多く、移行性が向上している。
デメリット	•既存資産をミドルウェア用に改修が必要。 •ミドルウェアの対応状況により、移行費用が増減する。

2-7. 資産をそのまま利用する方式

- 既存資産(COBOLプログラムやJCLなど)は基本的に変更しない
- マイグレーション専用ミドルウェアやアダプタ開発で、既存資産を実行

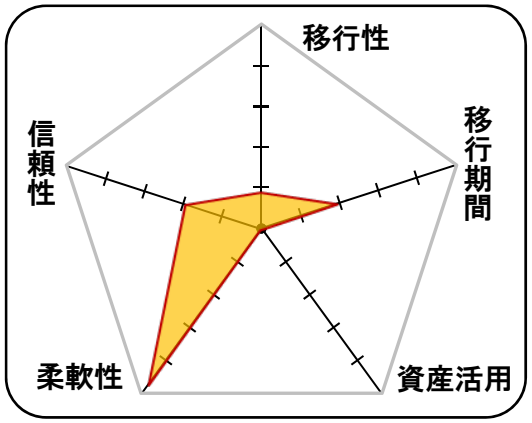


メリット	•現行資産の改変が不要で、アプリケーション品質を保持できる。 •業務仕様を継続できる。 •現行資産とミドルウェアとの差異は、マイグレーションベンダ側が対応。
デメリット	•使用できるのはメインフレームで使用している機能の範囲内が基本。 •インフラはミドルウェアにより制限を受ける。 •ミドルウェアを顧客資産に合わせる方式で、ミドルウェアの品質確保は難しい •膨大なメインフレームの全機能を使用できる訳ではないので、サポートされていない部分の改修は必要。この部分は資産を移行先に合わせる方式に近い。

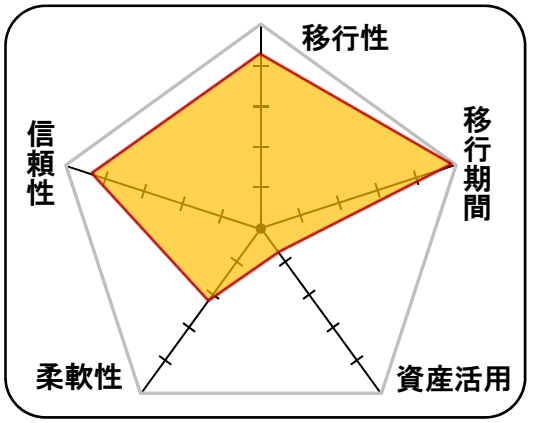
2-8. システム再構築方法による比較

HITACHI
Inspire the Next

スクラッチ開発



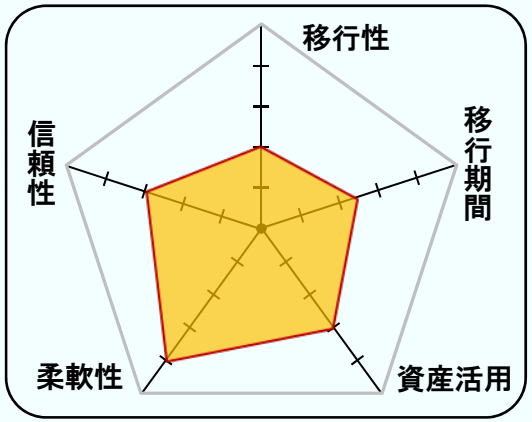
パッケージ適用



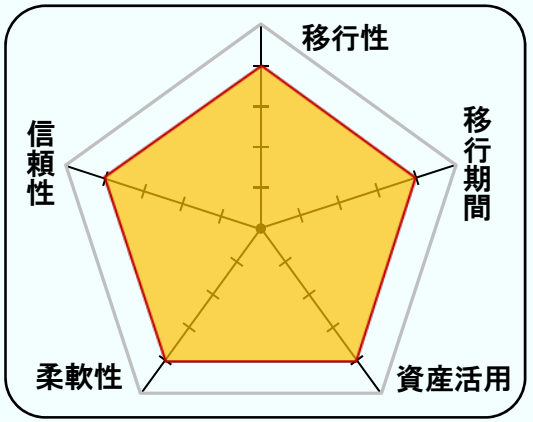
グラフは数値が大きい（外側）ほど、

- **移行性**
  開発量少ない／生産性が高い、費用が少ない
- **移行期間**
  移行期間が短い
- **資産活用**
  資産活用率が高い
- **柔軟性**
  要求仕様取り込みやシステムの拡張が容易、
  プラットフォームやミドルウェアの選択肢が多い
- **信頼性**
  構築したシステムの信頼性が高い

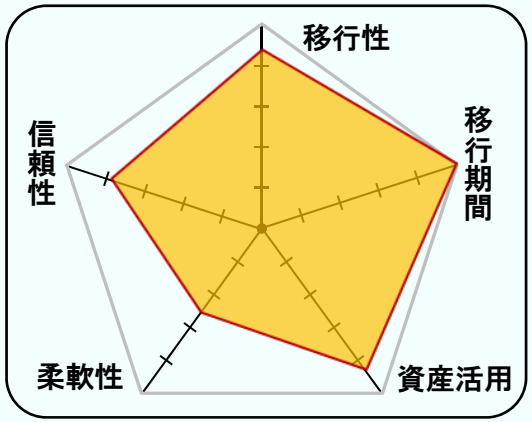
仕様情報のみ利用



資産を移行先に合わせる



資産をそのまま利用

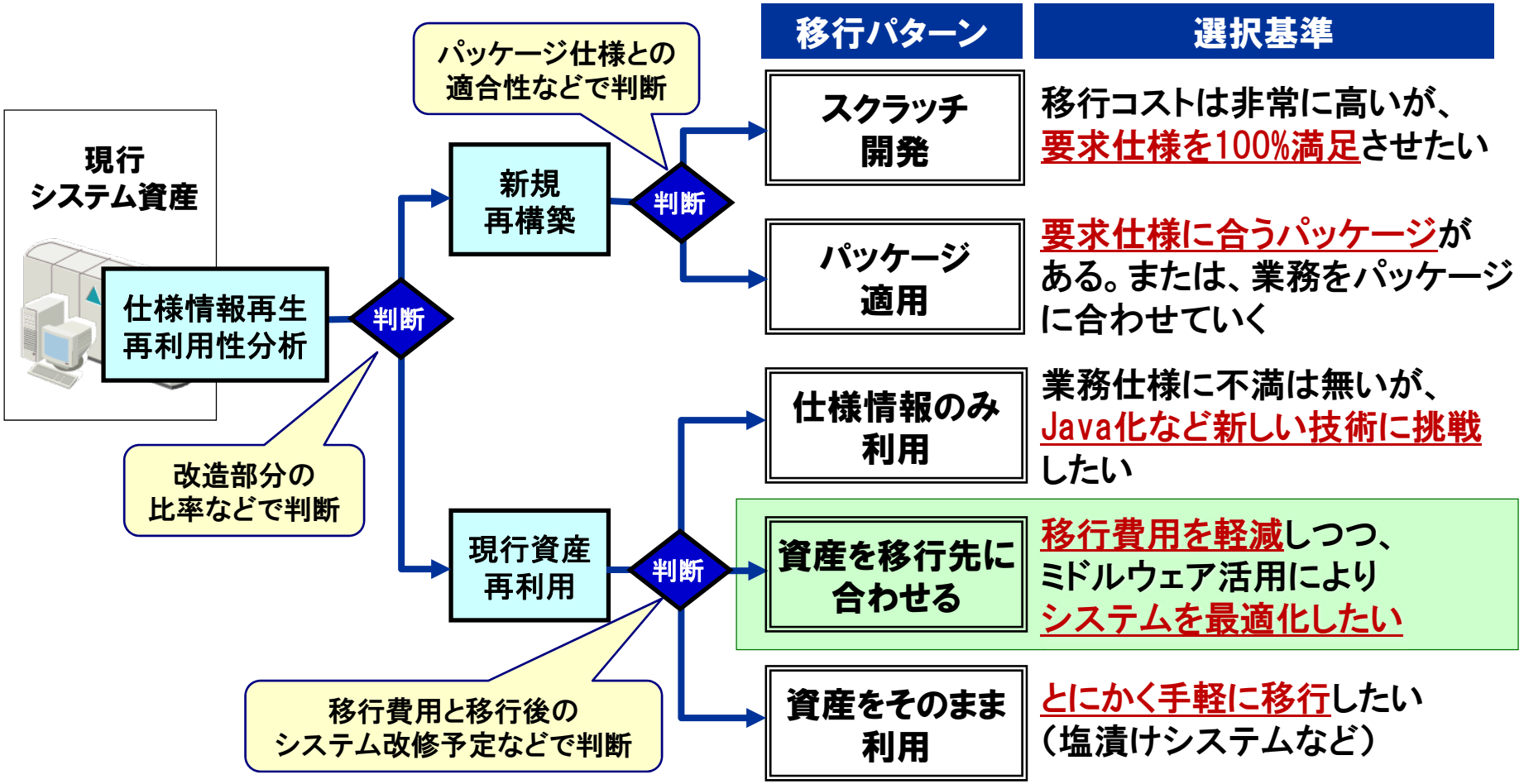


現行資産再利用型（レガシーマイグレーション）

＜注意事項＞グラフの数値は、メリット／デメリットを考慮したうえでの弊社による判断です

uVALUE

2-9. システム再構築パターンの選択基準



**3章、4章では移行費用を軽減しながら、ミドルウェア活用により
システムの最適化が可能な「資産を移行先に合わせる」方式で説明**

HITACHI
Inspire the Next

Contents

uVALUE

3-1. マイグレーション作業の流れとポイント

HITACHI
Inspire the Next

1. 資産棚卸・移行性分析

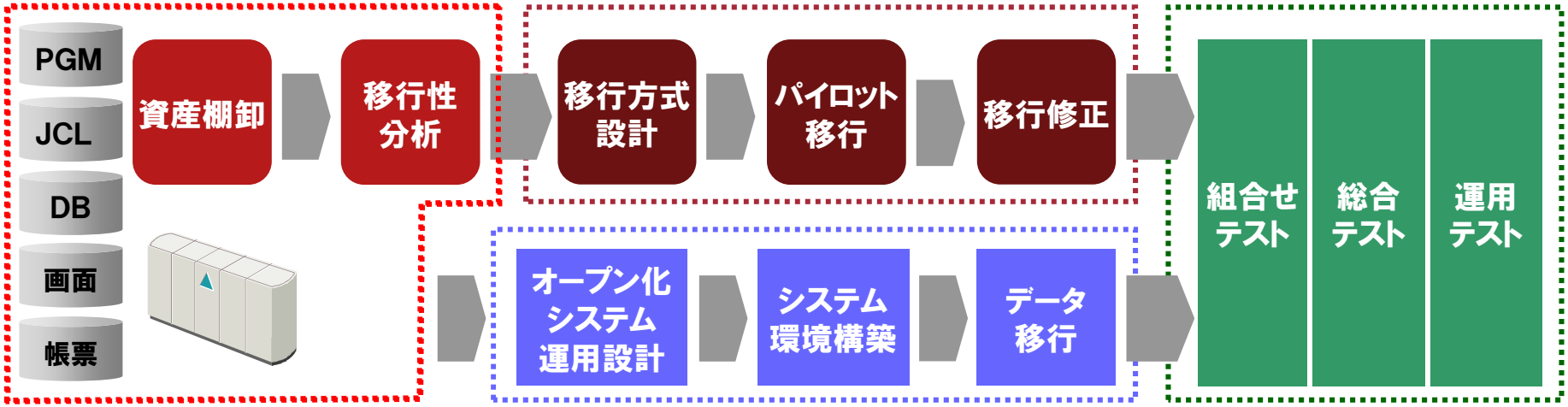
- ●正確かつ確実な資産棚卸の実施
  （後続作業ボリュームに大きく影響）
- ●移行性分析結果にもとづく現実的な移行費用の算出
- ●PGM変更箇所と修正規模の把握
- ●システム環境変更に伴う変更ポイントの把握（PGM・画面・帳票等）

2. 移行設計・本移行

- ●移行対象・移行範囲を明確にFIXすること
- ●リソース凍結とリソース管理の徹底
  （実行モジュールとソースコードの対応付け）
- ●オープン移行に関する技術課題の明確化
- ●クリティカルな技術課題の先行対策
  （性能対策：DBチューニング等）
- ●パイロット移行による移行方式内容の検証
  （本移行時のリスクを低減）

3. テスト

- ●コンペアテストが基本
- ●ストレート移行＝業務仕様変更なし
- ●チェックリスト作成及びテスト



4. オープン化システム運用設計

- ●運用方法は大きく変更
- ●オープンプラットフォームに関するスキルを持った人員が必要

5. システム環境構築

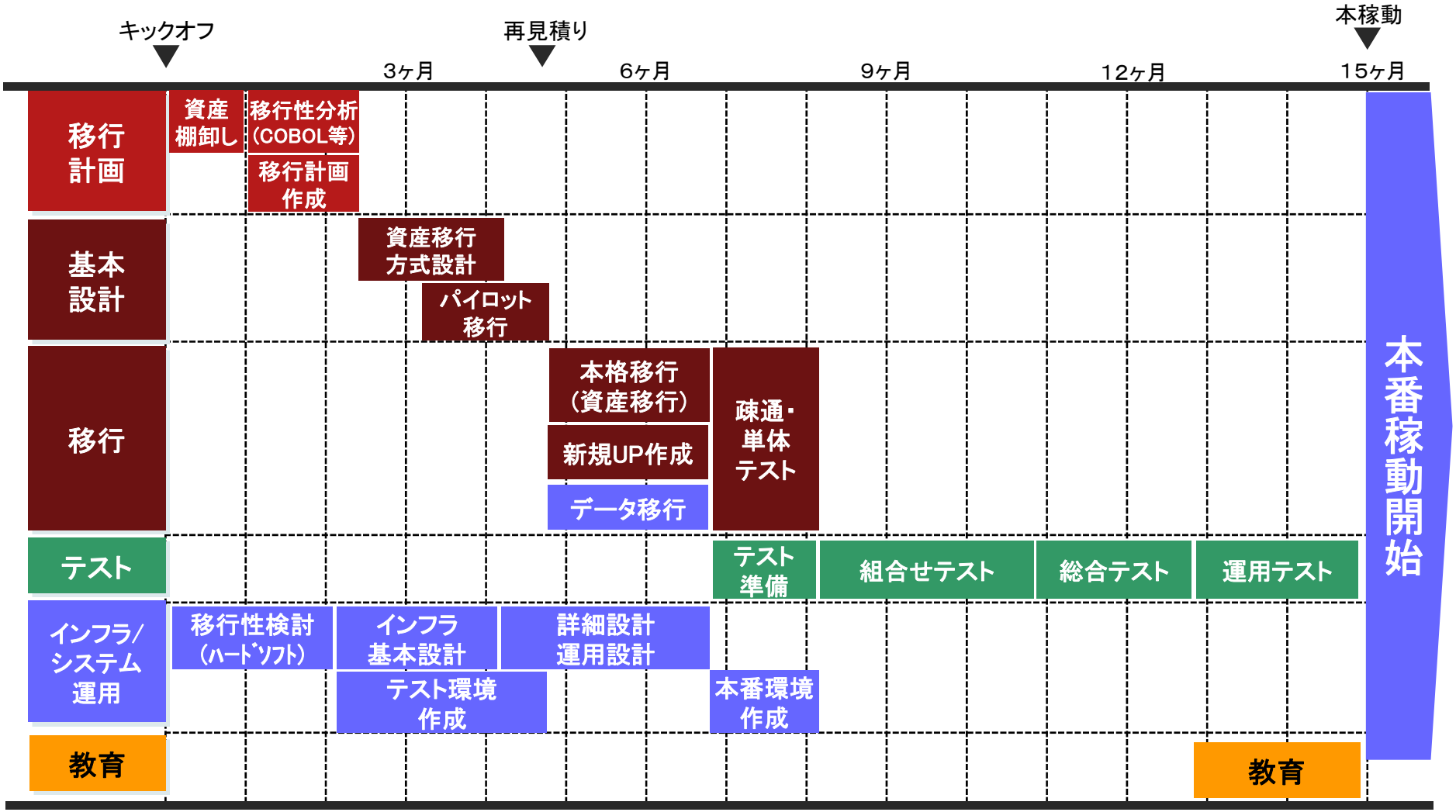
- ●開発スケジュールに対応したハード・ソフトの導入

6. データ移行＆業務移行

- ●組合せテスト以降は前提となるファイル等が必要となるためデータ移行を先行的に実施
- ●端末の切換と展開
  （ユーザへの操作教育・クライアント台数見直しなど）

HITACHI
Inspire the Next

レガシーマイグレーションにおける、一般的なスケジュール



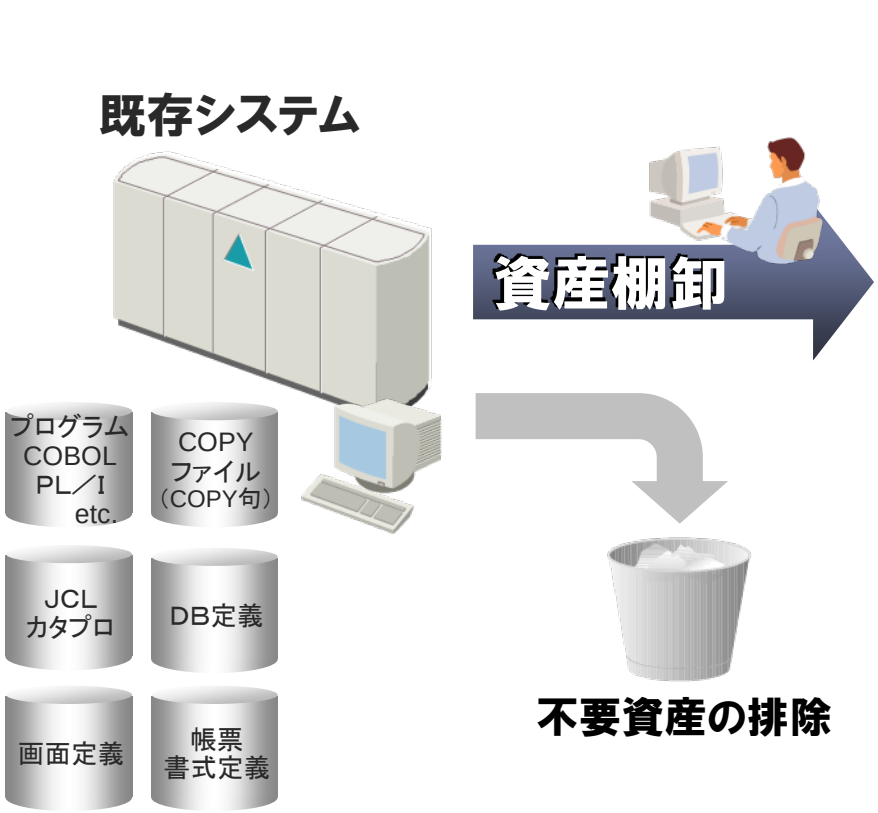
※移行期間は資産規模や移行性により変動します。

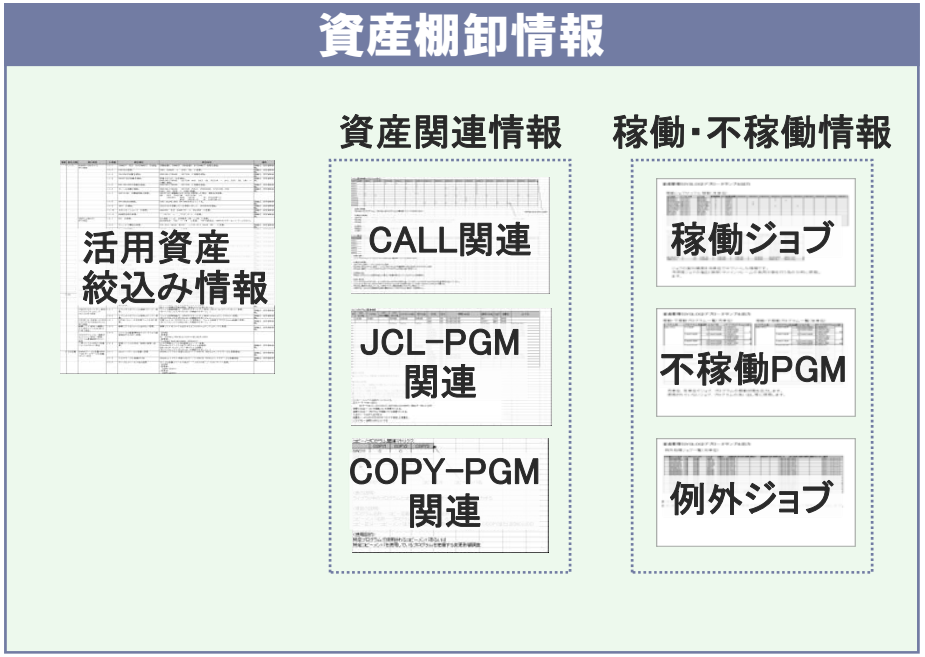
3-3. 資産棚卸

HITACHI
Inspire the Next

現行資産の棚卸を行い、移行対象資産の絞込みを行います

- 各種ツールやノウハウを駆使し、既存システム資産を調査・解析
- 不要資産の洗い出し

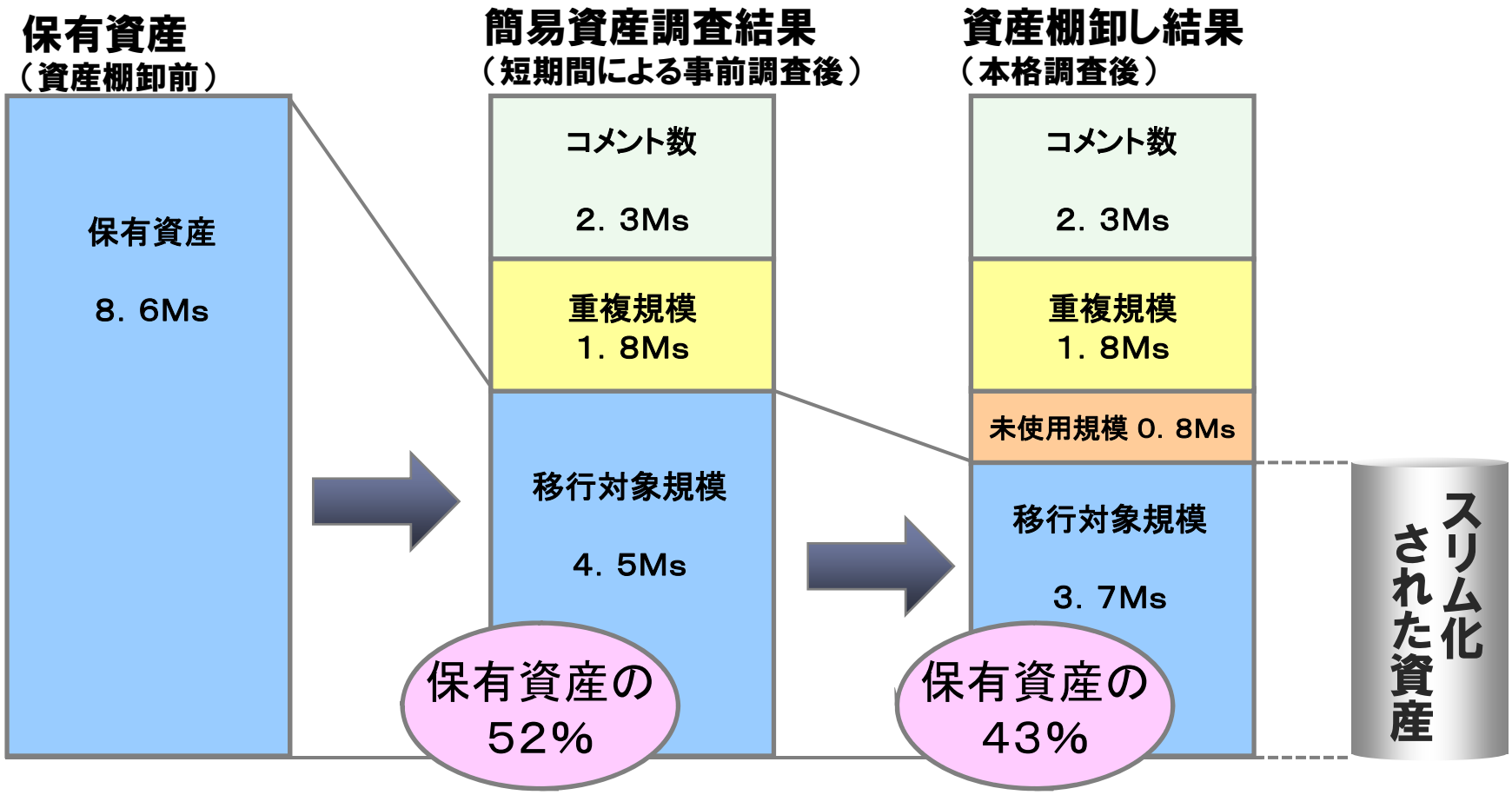




HITACHI
Inspire the Next

資産棚卸により、移行対象のCOBOL規模をスリム化した事例

■保有資産　8. 6Ms → 3. 7Ms



3-5. 移行性分析

HITACHI
Inspire the Next

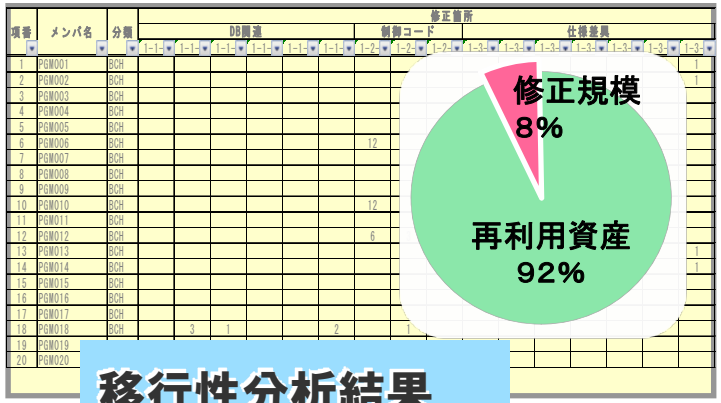
移行観点情報を元に、移行修正規模・箇所を分析調査
移行性分析結果より、早い段階でのリスク回避、品質確保が容易に可能

資産棚卸情報

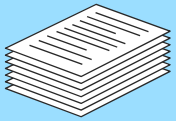
移行性分析

新システム要件との
Fit＆Gap

再利用の可否判断
改造・影響箇所の特定

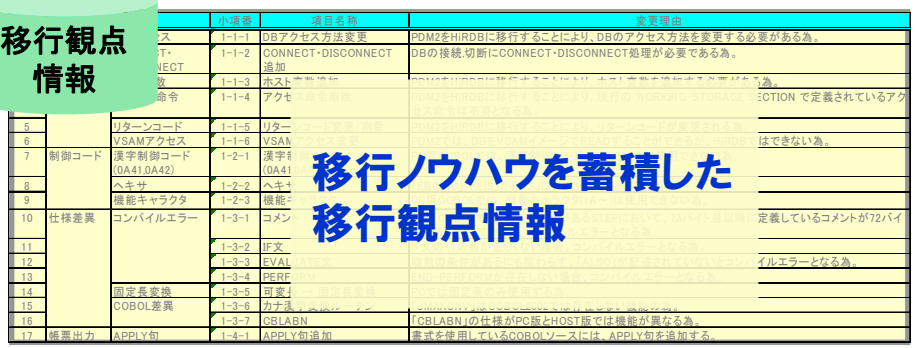


移行性分析結果



- 移行対象規模
- 移行費用
- 移行スケジュール　etc

移行観点情報



**移行ノウハウを蓄積した
移行観点情報**

HITACHI
Inspire the Next

移行性分析結果を元に、スリム化した資産をオープンシステムに移行

一般的なレガシーマイグレーション

スリム化された資産

移行

移行先
オープンシステム

COBOL〔オンライン〕	DB I／O修正	COBOL 2002
COBOL〔バッチ〕	ファイル入出力定義修正	
他言語〔PL/I等〕	言語変換	設計書
画面	画面変換	XMAP3
データベース	RDB定義作成	HiRDB
JCL〔運用情報〕	ジョブ運用移行	JP1 BJEX(*1)
データ	データ移行	データ

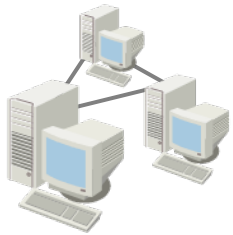
移行性分析結果

注意：移行後システムの製品名は日立の製品例

機能拡張

オープン化システムへ
新規機能を拡張

変化即応型システム



新システム

マイグレーション
完了後に、
機能を改修して
新システムを目指す

uVALUE

(*1) BJEX：uCosminexus Batch Job Execution Server

HITACHI
Inspire the Next

COBOLプログラムやJCLなどの資産だけでなく、インフラ及びシステム運用についての移行設計も必要

既存システム

■インフラ部分
- ストレージ
- プリンタ
- バックアップ装置
- ネットワーク
- etc

■システム運用
- ジョブ運用管理
- 障害監視
- バックアップ運用
- 外部とのデータ交換
- etc

現状調査

システム
利用状況

オープンシステム

■インフラ部分
- システム稼動分析
- 運用調査
- ネットワーク構成
- etc

移行性の検討
〔インフラ・運用〕

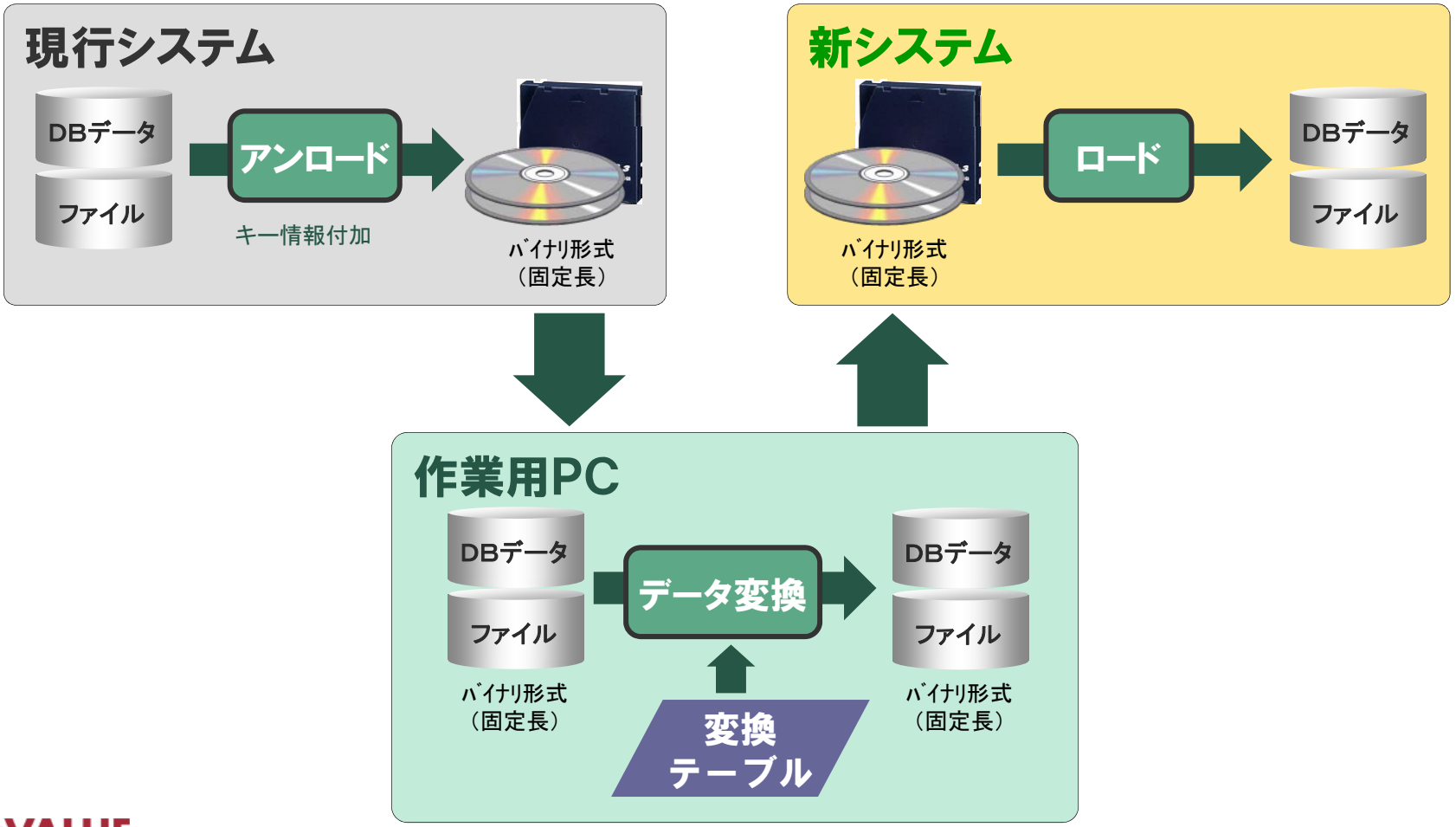
■システム運用
- ジョブ管理方法
- スケジューリング方式
- ジョブ・リカバリ運用
- 実行結果の監視方法
- etc

- 要件定義
- 構成案
- 運用変更点

HITACHI
Inspire the Next

データ移行時は、状況に応じて文字コード変換や外字対応が必要

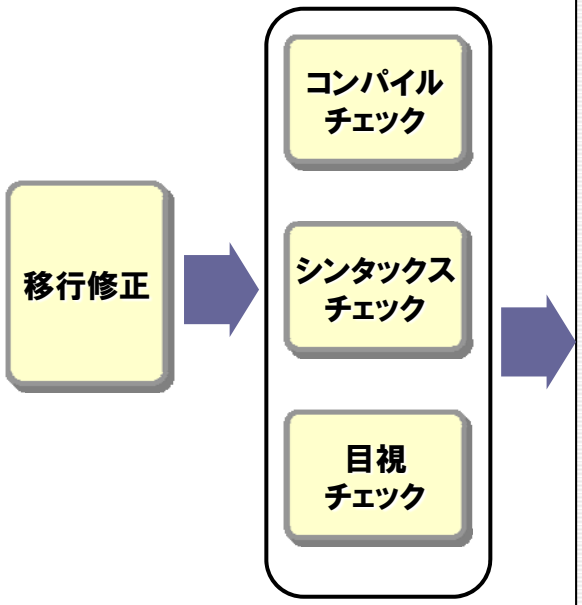
- 移行先コード変換・外字に対応した変換テーブル作成
- 対象となるマスタDB、入力ファイルの絞込みが重要



HITACHI
Inspire the Next

移行修正の一環として、疎通テスト(機能単体テスト)まで実施

- 移行性分析結果からテスト対象項目、テスト対象資産を選定
- 移行観点毎に、テスト方法を事前に分類
- 移行観点が同じ項目は、目視チェックでテストを効率化のケースあり



【疎通テスト】

移行観点情報

項番	分類		小項番	項目名称	テスト分類
1	DB関連	DBアクセス	1-1-1	DBアクセス方法変更	机上のみ
2		CONNECT・DISCONNECT	1-1-2	CONNECT・DISCONNECT追加	データ活用
3		ホスト変数	1-1-3	ホスト変数追加	データ活用
4		アクセス命令	1-1-4	アクセス命令削除	机上のみ
5		リターンコード	1-1-5	リターンコード変更/削除	0件データ

事前に分類

疎通テスト項目

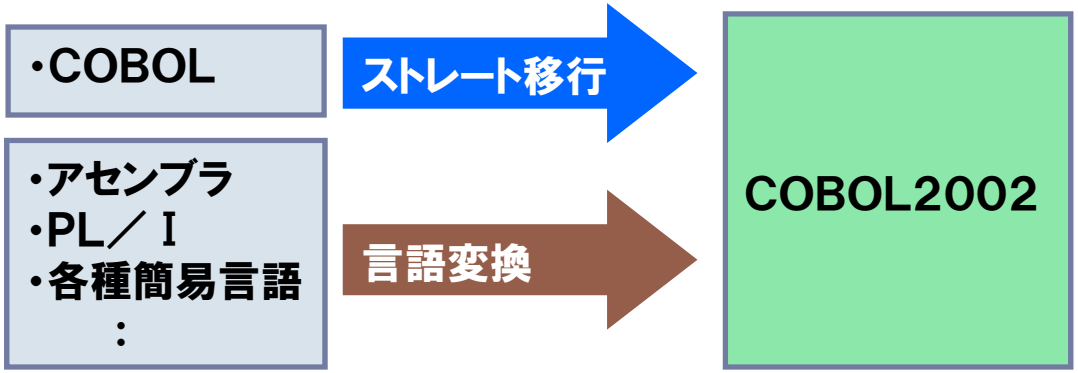
項番	メンバ名	分類	チェックリスト DB関連 1-1-1	1-1-2	1-1-3	1-1-4	1-1-5	テスト実施内容	テスト結果
1	PGM001	BCH						未	未
2	PGM002	BCH		○				データ活用	データ活用
3	PGM003	BCH			○			データ活用	未
4	PGM004	BCH					○	0件データ	0件データ
5	PGM005	BCH				○	○	0件データ	0件データ
6	PGM006	BCH				○	○	未	未
7	PGM007	BCH				○		未	未
8	PGM008	BCH	○					机上のみ	机上のみ
9	PGM009	BCH	○					未	未
10	PGM010	BCH						未	未

テスト対象プログラム

同一移行方式のためテスト省略

HITACHI
Inspire the Next

再利用するプログラムは業務システム向きのCOBOLで移行します
COBOL以外で作成されたものは、COBOL化します



現在利用されている
ビジネス・アプリケーションの70%以上
はCOBOLで書かれている(Giga Group)

全世界で
1万6千社以上の大企業
がCOBOLを使用している(GigaGroup)

出典：COBOLコンソーシアムホームページ(http://www.cobol.gr.jp/)
第11回 インターネット時代のCOBOL活用セミナー(2006年7月21日実施)
テックバイザージェイピー「レガシーマイグレーションにおけるサービス指向アーキテクチャの位置づけ 」

3-11. COBOLは業務システム向きの言語

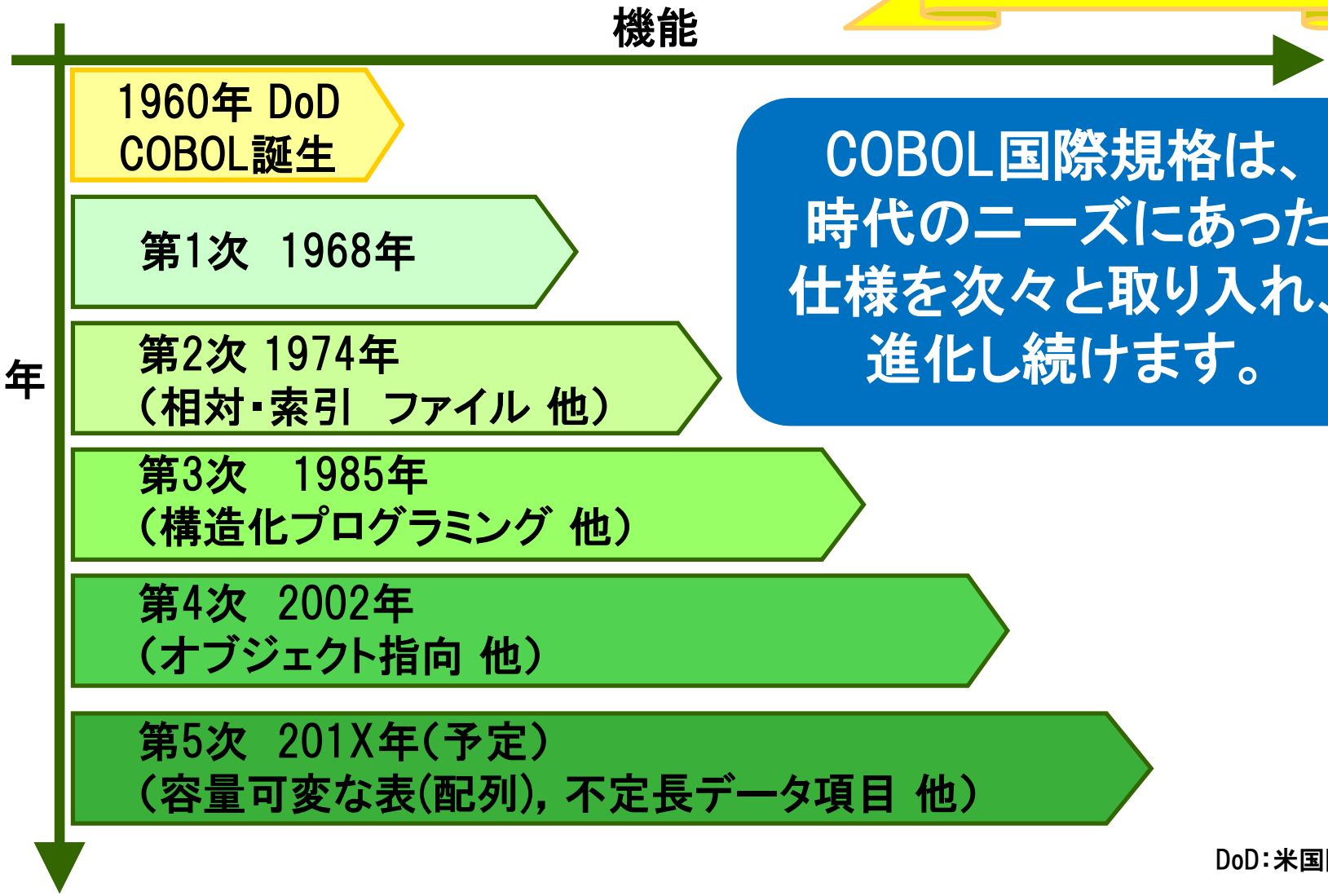
HITACHI
Inspire the Next

- メモリ設計が容易
  - 動的にメモリを取らないプログラミング
  - レコードの入出力で同じ記憶域を使い回す
- 金額計算向き
  - 10進固定小数点の変数を定義するだけで まるめ制御が可能
- 既存資産を長期にわたって活用可能
  - 国際規格が仕様を規定
  - ハード・OS間で、共通の言語仕様

HITACHI
Inspire the Next

国際規格があり将来もより安心

2010年で
COBOL生誕 50年！！



COBOL国際規格は、
時代のニーズにあった
仕様を次々と取り入れ、
進化し続けます。

DoD：米国防総省

3-13. マイグレーションにおける7つのポイント

HITACHI
Inspire the Next

Point1:『無駄な移行費用(作業)の排除』

資産棚卸

- 既存システム資産の有効リソースを正確に把握し、不要資産を排除
- 移行作業の対象となる資産を絞り込み・特定することにより、ムダな移行費用(作業)を排除

Point2:『事前にリスクを回避』

移行性分析

- 本番移行作業を実施する前に、移行対象資産全体に対するオープン化の移行性を分析
- 移行性の可否判断により、後続プロジェクトのリスクを事前に回避

Point3:『システム全体としての計画を策定』

システム再構築
全体計画策定

- 移行性分析結果をもとに、本番移行作業における計画を策定
- 他システム間連携も含めたシステム全体としての再構築計画を策定

Point4:『移行作業の効率化』

パイロット移行

- 先行的にパイロットシステムによる移行作業を実施し、移行結果を評価
- 移行結果・評価を本番移行作業へフィードバックすることにより、より効率的でスムースな本番移行を実現

3-14. マイグレーションにおける7つのポイント

HITACHI
Inspire the Next

Point5:『資産凍結後の差分管理と反映タイミング』

資産凍結

- プロジェクトスタート後は仕様凍結が基本
- 消費税率変更等、ビジネス推進上どうしても必要なシステム改修以外は実施しないこと
- リソース管理を徹底し、差分取り込みを実施する場合は、総合テストに入るまで追い付き反映は実施しないこと
  （差分取り込みによるバージョン不正・デグレード・テスト不能等の問題が発生するため）

Point6:『テスト準備とテスト実施』

テスト

- テスト環境の準備及び、テスト用チェックリスト作成、テスト用データ移行は早めに完了させておくこと

Point7:『移行時のリスクを回避』【大規模システムの場合】

段階的本移行

- 移行対象システムを全て一気に移行するのではなく、順次、段階的に移行
- これにより、本番移行時の大規模なリスク(トラブル)を回避

HITACHI
Inspire the Next

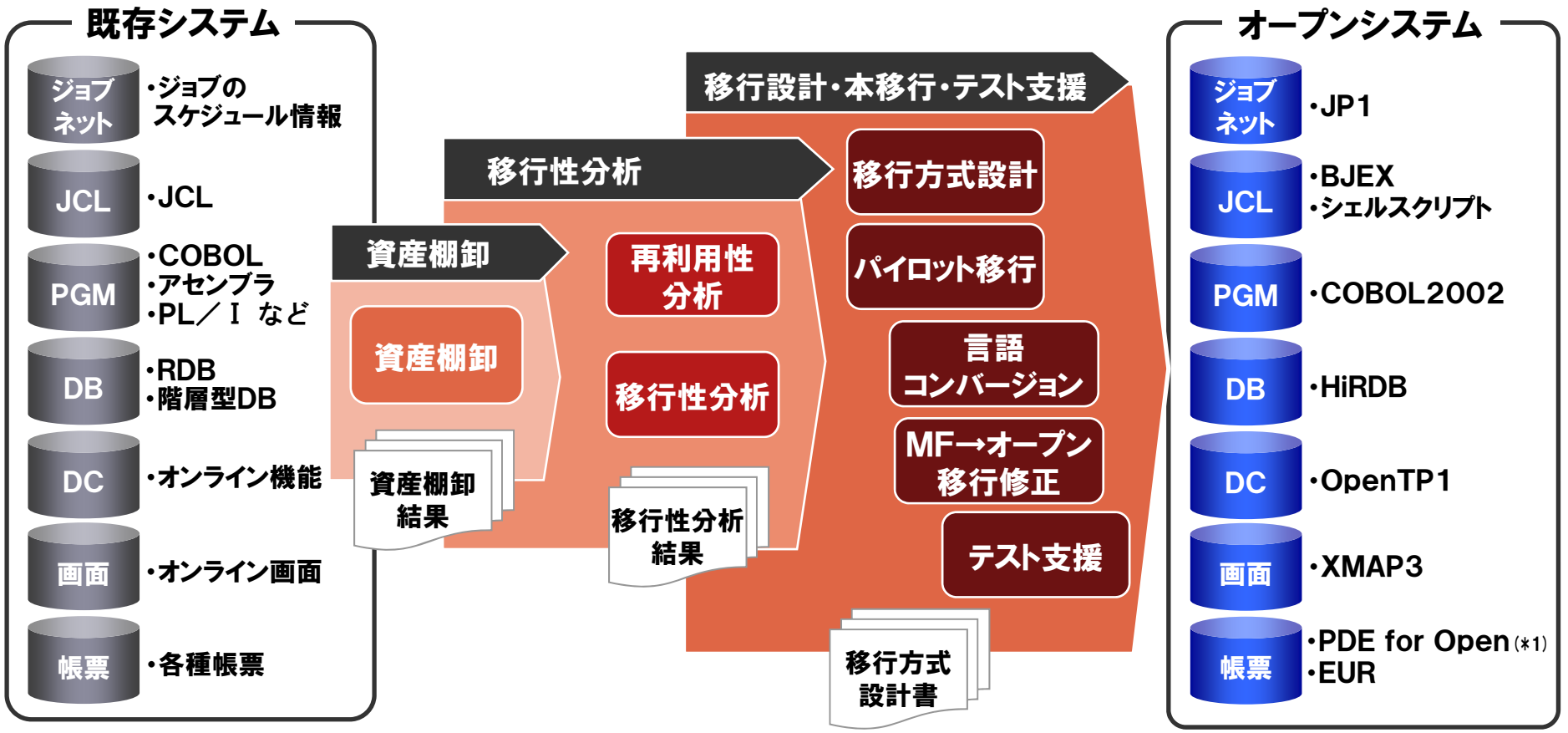
Contents

uVALUE

4-1. マイグレーションサービスの全体概要

HITACHI
Inspire the Next

- 既存システムの資産規模を短期間に把握 ⇒ 資産棚卸
- 資産特性分析による移行可否判断 ⇒ 移行性分析
- 移行ツール群と専門部隊による高品質な移行 ⇒ 移行設計・本移行・テスト支援
- マイグレーションに最適化された日立ミドルウェアの採用



(*1) PDE for Open：PRINT DATA EXCHANGE for Open

4-2. マイグレーション事例紹介：A社(1／3)

HITACHI
Inspire the Next

事例の特徴

■資産調査、移行性分析を重点的に実施しリスク低減
■パイロット移行による移行方式内容の確認と問題点の早期抽出

【顧客ニーズ】

- 既存システムリソースはノウハウの固まりなので捨てたくない
- オープン化によりランニングコストを削減したい
- インフラ基盤、システム運用の標準化

【既存システムの課題】

- 既存メインフレームシステムのランニングコストが高い
- システム改修の際のメンテナンスに手間がかかる

【資産概要】

- COBOLソース ‥ 約900KStep
- JCL ‥‥‥‥‥‥‥‥‥‥‥ 約700本
- DB定義 ‥‥‥‥‥‥‥ 約450本
- 画面定義‥‥‥‥‥‥‥ 約350本
- 帳票定義 ‥‥‥‥‥‥‥約　50本

4-2. マイグレーション事例紹介：A社(2／3)

HITACHI
Inspire the Next

【作業スケジュール】 全体14ヶ月

資産棚卸
- 資産規模調査
- 関連調査など

移行性分析
- 修正項目抽出
- 移行性評価

2．5ヶ月

移行方式設計
- 修正方式決定
- ツール適用／手作業の切り分け
  ・COBOL　・JCL
  ・画面　・帳票
  ・DB定義　・データ移行

3．5ヶ月

パイロット移行
- パイロット移行実施
- 結果を移行方式設計書に反映

1ヶ月

運用設計
- 稼働システムの決定
- システム運用の検討

2．5ヶ月

本移行
- COBOL、JCL、画面・帳票、DB定義移行

3．5ヶ月

単体テスト
- 疎通確認

1．5ヶ月

組合せ・総合・運用テスト
- JOB実行によるテスト

4．5ヶ月

データ移行

6ヶ月

環境構築
- サーバシステムの構築

2ヶ月

システム稼働

日立作業
お客様作業

4-2. マイグレーション事例紹介：A社(3／3)

HITACHI
Inspire the Next

【サービス適用効果】

効率的なシステム移行

- 現行資産調査・移行性分析によりツール適用率及び手修正比率を事前に把握するとともに、パイロット移行による移行方式設計内容の確認および問題点の早期抽出を実施したことにより、よりリスクを低減した形でオープン環境への移行を実現
- 現行IT資産を約14ヶ月でオープン環境にストレート移行
- 早期の新システム稼働により、ランニングコスト削減が早期に実現

移行後の効果

- 顧客インフラ基盤の標準化を実現
- システム運用の標準化による、システムメンテナンスの効率化を実現

4-3. マイグレーション事例紹介：B社(1／3)

HITACHI
Inspire the Next

事例の特徴

■複数の言語で開発されたアプリケーションをCOBOL2002に統一
■2つの業務を段階的にマイグレーション

【顧客ニーズ】

- システム運用のランニングコスト低減
- メインフレームによる管理主体の統合システムから、管理集中・処理分散のオープンシステムへ移行
- 現行資産を活用して、業務ノウハウの継承を図る

【既存システムの課題】

- ランニングコストが高い
- 複数の言語で構成されるアプリケーションプログラム

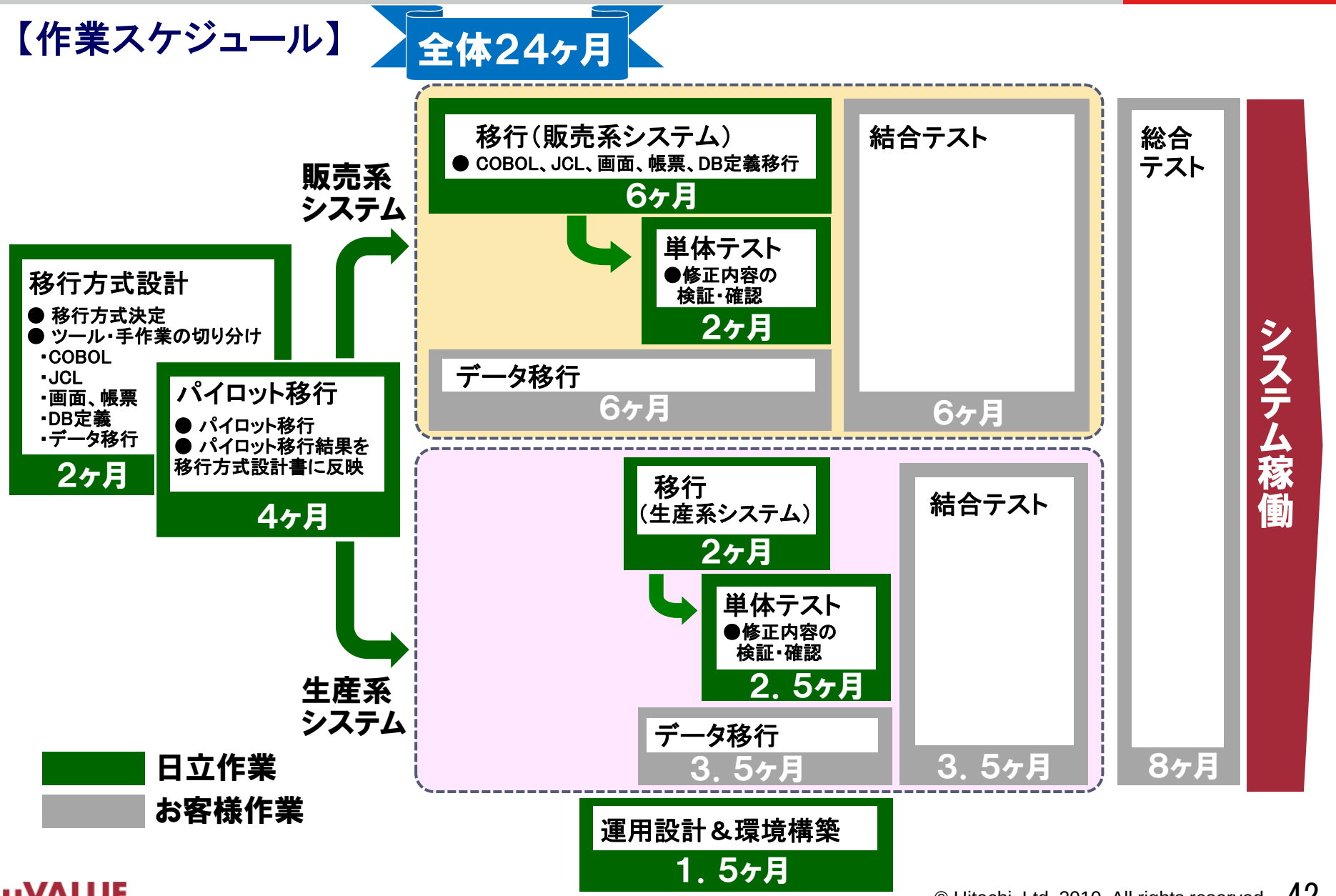
【資産概要】

- ソースプログラム(合計 約2.7MStep)
  - COBOL…… 約3400本(約1.9MStep)
  - PL/I …… 約20本(約10KStep)
  - NATURAL ‥ 約1400本(約460KStep)
  - アセンブラ‥ 約700本(約300KStep)
- JCL …… 約4000本
- DB定義 …… 約450本
- 画面・帳票 ‥ 約600本

4-3. マイグレーション事例紹介：B社(2／3)

HITACHI
Inspire the Next

【作業スケジュール】



4-3. マイグレーション事例紹介：B社(3／3)

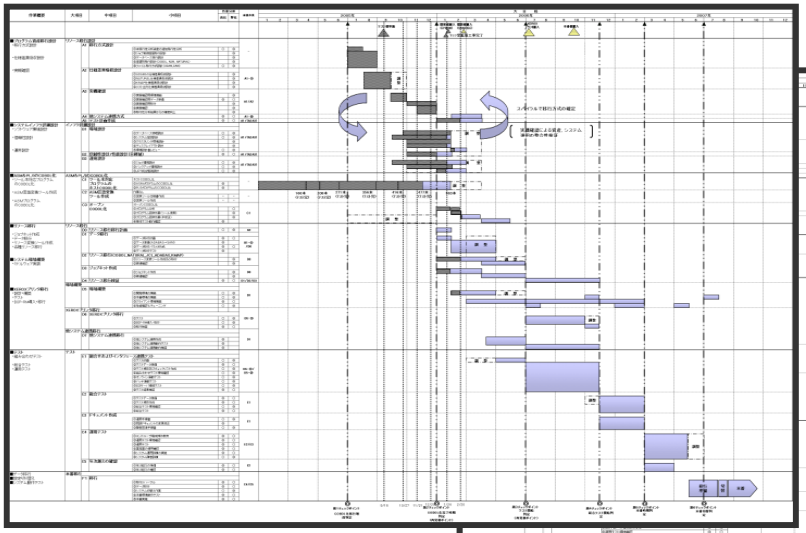
HITACHI
Inspire the Next

【サービス適用効果】

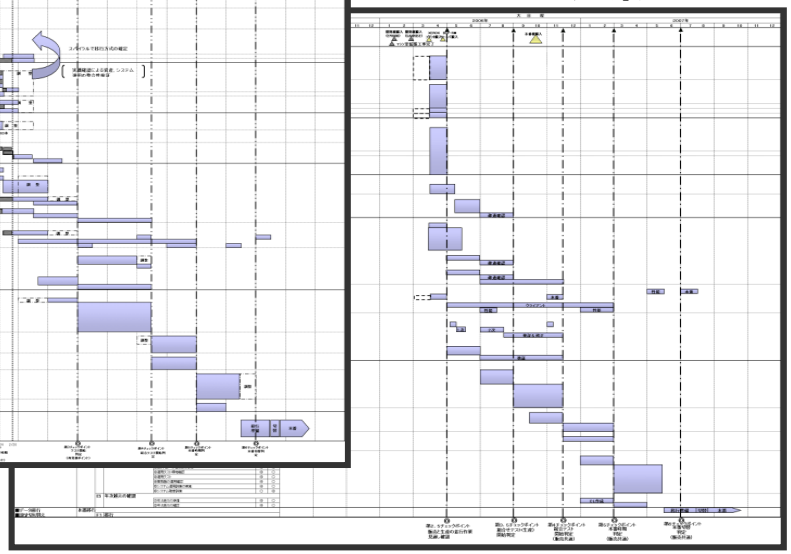
システム移行の効果

- 複数言語のアプリケーションをCOBOL2002に統一で生産性向上
  複数言語(2.7MStep) → COBOL2002(5.2MStep)に移行
- 2つの業務システムを2年で段階的にオープンシステムに移行
- 早期のシステム稼動により、ランニングコストの低減

販売系システム



生産系システム



4-4. マイグレーション事例紹介：C社(1／3)

HITACHI
Inspire the Next

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
■新規業務の追加も行い新システムを稼動

【顧客ニーズ】

- 効率的かつリスクを低減した形でオープン化への移行を実現したい
- エンドユーザに対するサービスの向上が容易に行えるようにしたい
- ランニングコストを低減させたい

【既存システムの課題】

- システムが老朽化
- 業務的な観点での不要資産が多い

【資産概要】

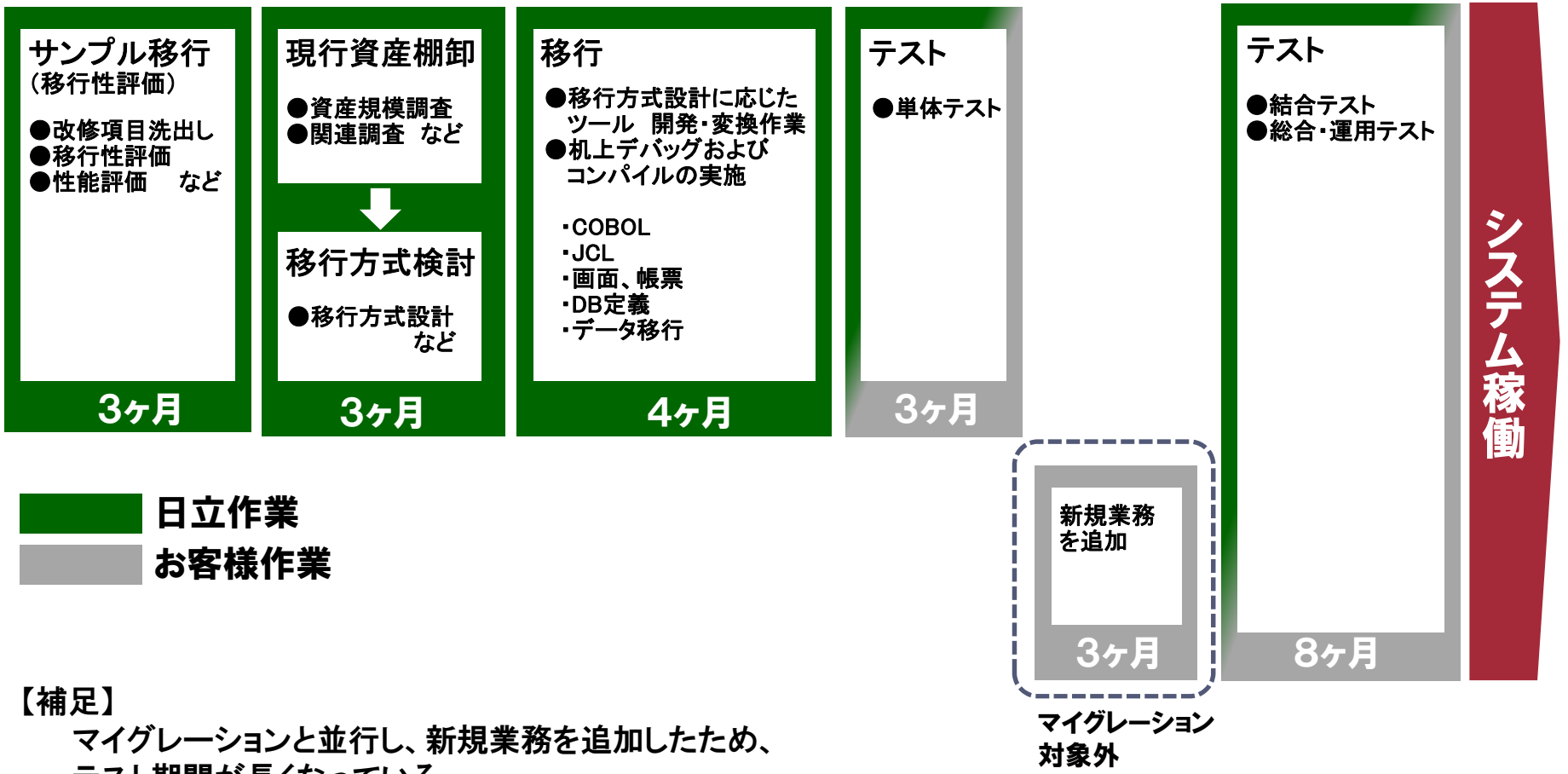
- COBOLソース ‥ 約4800本(約3.8MStep)
- アセンブラ ‥‥‥ 約30本
- JCL ‥‥‥‥‥‥ 約3400本
- DB ‥‥‥‥ 約100本
- 画面定義 ‥ 約930本
- 帳票定義 ‥ 約750本

4-4. マイグレーション事例紹介：C社(2／3)

HITACHI
Inspire the Next

【作業スケジュール】

全体24ヶ月



日立作業

お客様作業

【補足】

マイグレーションと並行し、新規業務を追加したため、テスト期間が長くなっている。

4-4. マイグレーション事例紹介：C社(3／3)

HITACHI
Inspire the Next

【サービス適用効果】

資産のスリム化

- 資産棚卸により不要資産を特定し、お客様が最終的な要・不要を判断
  資産全体のスリム化を実現し、移行費用の削減に成功

資産	棚卸前	棚卸後	削減率
COBOL	3.8MStep	2.6MStep	32%
JCL	3400本	720本	79%
画面	930本	260本	72%
帳票	750本	110本	85%

効率的なシステム移行

- サンプル移行を事前に実施し、マイグレーションのリスクを低減
- 資産棚卸からプログラム単位でのテストまでを10ヶ月で実現
  (その後、新規機能要件の追加作業、テストをお客様／SE側で対応)

4-5. 仕様回復サービス

HITACHI
Inspire the Next

【サービス内容】

- 『独自のリエンジニアリングツールとノウハウ』を駆使し、現行システムの仕様情報を、お客様の目的に応じて、回復(再生)するサービス。
- 『Web環境で仕様情報を確認できるReBIRTHツール(※1)』をご提供することで、システム改修に伴う仕様変更を継続的にサポート。

※1：仕様情報をWebブラウザ上でインタラクティブかつ効率的に参照するためのツール

【お客様のニーズ】

- システムの最新仕様を明らかにしたい〔ホワイトボックス化〕。
- システムの保守作業を効率化したい。
- システムを改修したときの影響範囲を知りたい。
- 次期システム検討に向けて現行システム仕様を把握したい。
- 後継者への引継ぎをしたい〔2007年問題対策〕。

etc.

【適用効果】

仕様回復サービスの成果は、将来に渡るシステムのライフサイクルに活かすことができます。

4-6. 仕様回復サービスの概要

HITACHI
Inspire the Next

**プログラム、JCLを解析し、
仕様情報を回復します。**

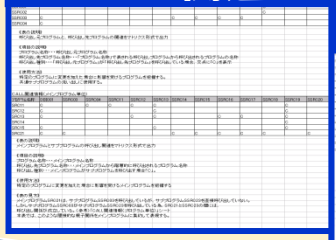
システム
リソース

・プログラム解析
・JCL解析
・ブラウザ化

追い付きサービス
（変更差分反映）

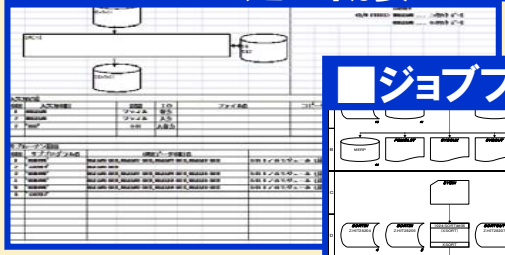
仕様情報

■CALL関連

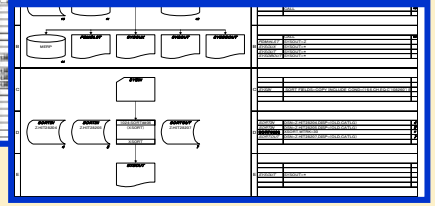


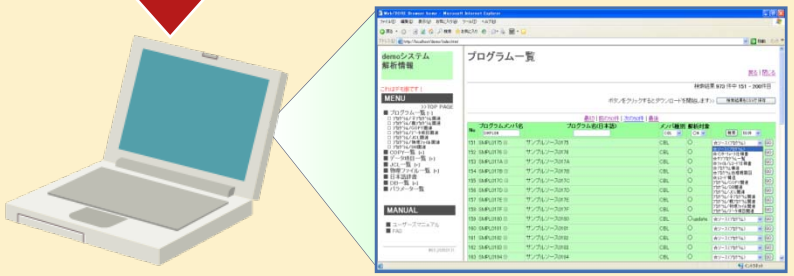
■プログラム－JCL関連

■プログラム処理概要



■ジョブフロー





**各種ドキュメントを、
ブラウザで参照することができます。**

◆Webベースのツールにより
- 仕様情報の検索効率UP !!
- 関連情報を漏れなく調査 !!

◆追い付きサービスにより、
常に最新の仕様情報管理

uVALUE

4-7. 仕様回復サービス事例紹介：D社(1／3)

HITACHI
Inspire the Next

事例の特徴

■仕様回復サービスによる、既存システムの仕様情報を回復
■回復された仕様情報を元に、Javaにて再構築

【顧客ニーズ】

- 既存システム仕様はそのままでシステム再構築したい
- プログラム言語をCOBOLからJavaに変更したい
- 既存のプログラムをなるべく早く確認する方法が欲しい。
  （従来はメインフレームを直接覗いていた）

【既存システムの課題】

- システム改修にあわせて仕様情報が反映されていない
- 参考になる仕様情報が紙でしか存在しない
- 仕様情報がすべて揃っていない

【資産概要】

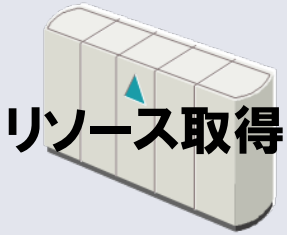
- COBOLソース ・・・・・・・・・・ 約1万本(約8MStep)
- COPYファイル(COPY句)・・・ 約2万本
- JCL ・・・・・・・・・・・・・・・・・・・・ 約2万本

4-7. 仕様回復サービス事例紹介：D社(2／3)

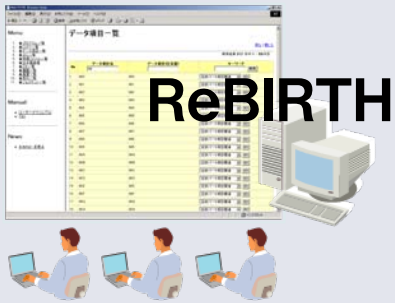
HITACHI
Inspire the Next

【仕様回復サービス＆差分回復サービスの概要】

お客さま環境



リソース取得



ブラウザツールを活用した
システム再構築

リソース借用

【差分回復サービス】
・約2年間に渡り実施
・差分回復：1回／3ヶ月

仕様回復データ提供

日立

①資産棚卸
- ■ 既存リソースの調査
- ■ 仕様回復対象資産の特定
  - ・リソース改修分の調査
  - ・仕様回復対象資産の特定

②仕様回復
- ■ プログラム・JCLなどを解析
- ■ 仕様情報を回復

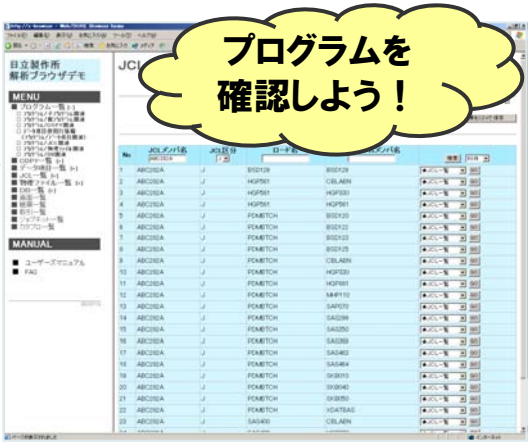
③ブラウザ化
- ■ 結果をブラウザデータに変換
- ■ お客様環境での動作確認

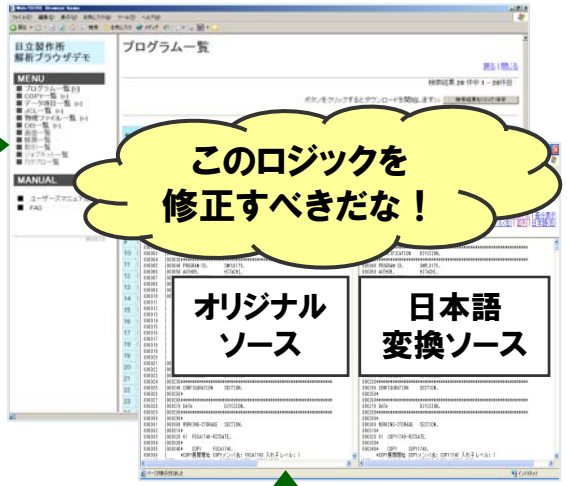
➡ **既存システムの仕様を活かした**
効率的なシステム再構築を実現

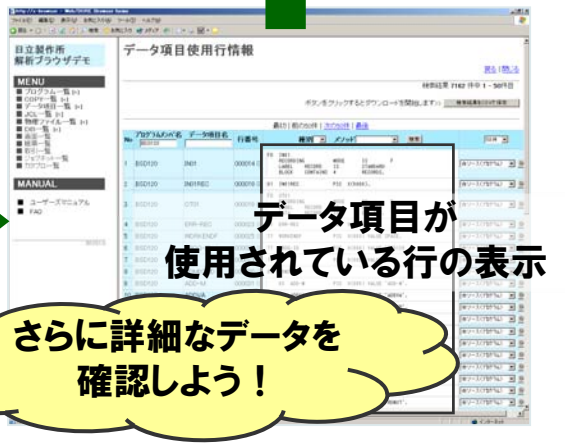
4-7. 仕様回復サービス事例紹介：D社(3／3)

HITACHI
Inspire the Next

【ReBIRTHツールの表示例】







4-8. マイグレーションを支える日立ミドルウェア

HITACHI
Inspire the Next

最先端のITを駆使した幅広い製品ラインアップで、お客様の多様なニーズに対応する日立ミドルウェアを提供します。

製品情報サイト
http://www.hitachi.co.jp/soft/

【マイグレーションで特に有効なミドルウェア】

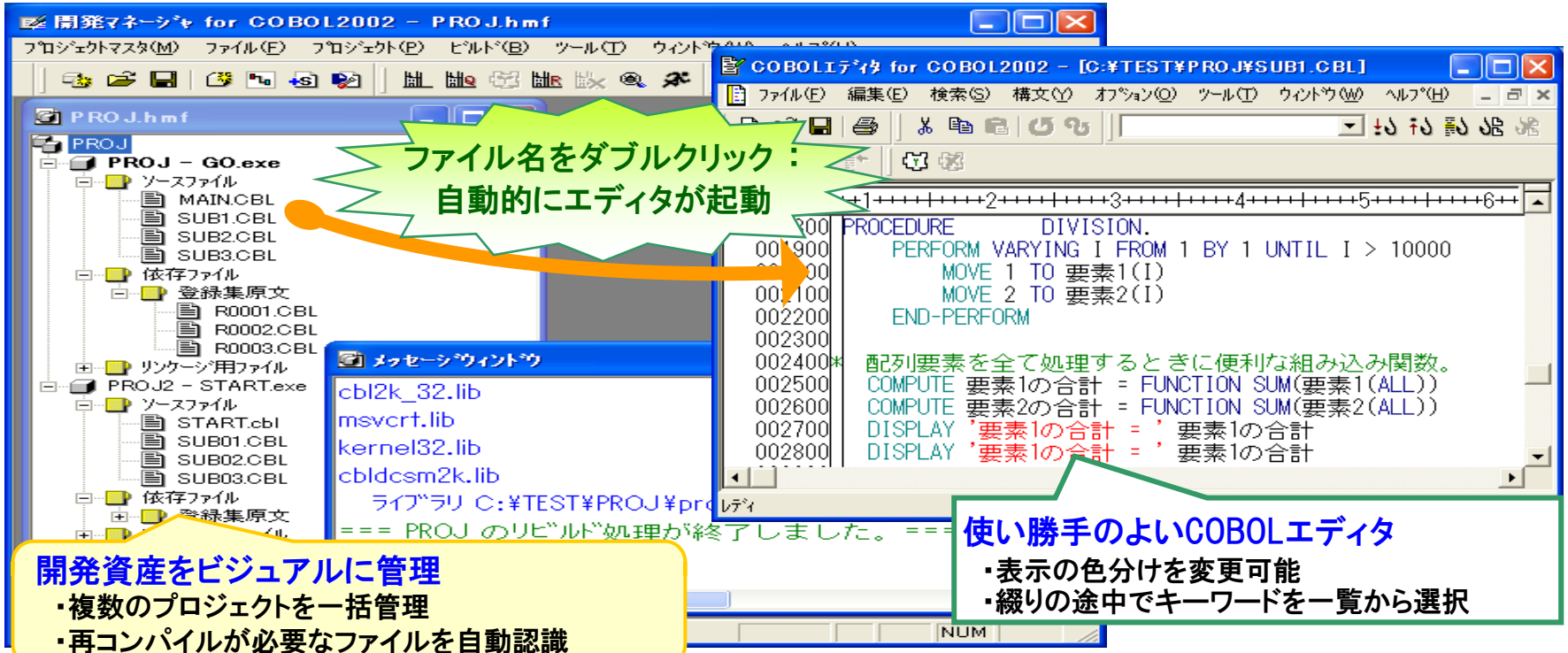
COBOL2002	最新のCOBOL国際規格(COBOL2002) に対応した開発・運用環境 http://www.hitachi.co.jp/soft/cobol/
JP1	統合システム運用管理 http://www.hitachi.co.jp/soft/jp1/
BJEX	バッチジョブ実行基盤: uCosminexus Batch Job Execution Server http://www.hitachi.co.jp/soft/cosminexus/bjex/
HiRDB	高信頼ノンストップデータベース http://www.hitachi.co.jp/soft/hirdb/
OpenTP1	分散トランザクションマネージャ: uCosminexus OpenTP1 http://www.hitachi.co.jp/soft/opentp1/
XMAP3	オンライン画面・帳票サポートシステム http://www.hitachi.co.jp/soft/xmap3/
PDE for Open	バッチ業務の帳票システム:PRINT DATA EXCHANGEオープンタイプ http://www.hitachi.co.jp/soft/pde/
SORT EE	多彩なソート・マージ: SORT Version8 - Extended Edition http://www.hitachi.co.jp/soft/sort/

4-9. COBOL2002の特長（1/2）

HITACHI
Inspire the Next

- COBOL最新国際規格（COBOL2002規格）に対応
  → オブジェクト指向をはじめとする新技術へ積極的に対応しています
- さまざまなプラットフォームに対応（AIX、Linux®、Windows®、HP-UXなど）
- コンパイラやCOBOLエディタ，テストデバッガ，カバレージ等の開発ツールを統合する開発マネージャを提供（Windows版）

＜開発マネージャ＞

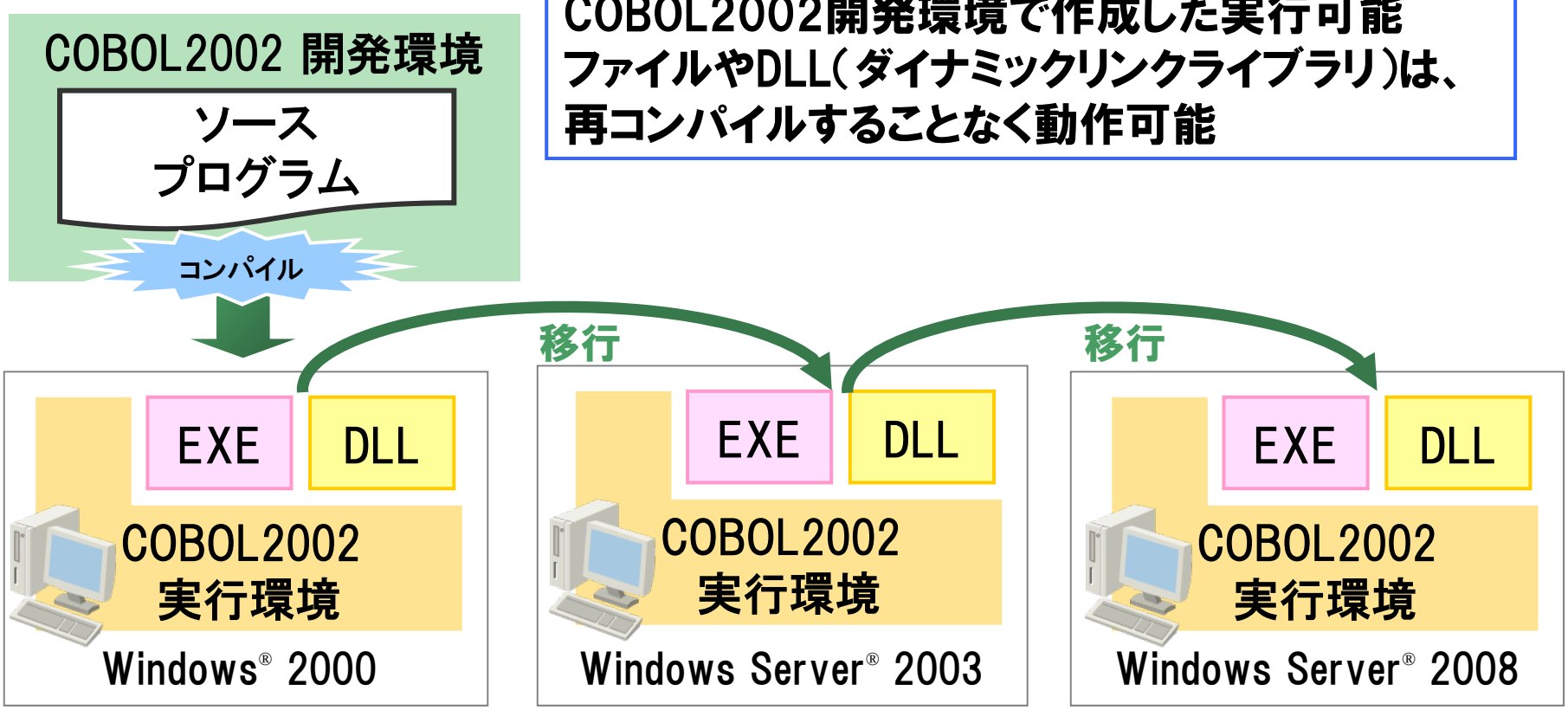


4-9. COBOL2002の特長（2/2）

HITACHI
Inspire the Next

- メインフレームCOBOL85資産を、オープン環境で活かす互換性に配慮
  → 日立独自仕様（画面、日本語、アドレス操作等）や互換用オプションを提供
- オープン環境のCOBOL85ソースとの互換性オプションを提供
- OSをバージョンアップしても、実行可能ファイルやDLLはそのまま動作

COBOL2002開発環境で作成した実行可能ファイルやDLL（ダイナミックリンクライブラリ）は、再コンパイルすることなく動作可能

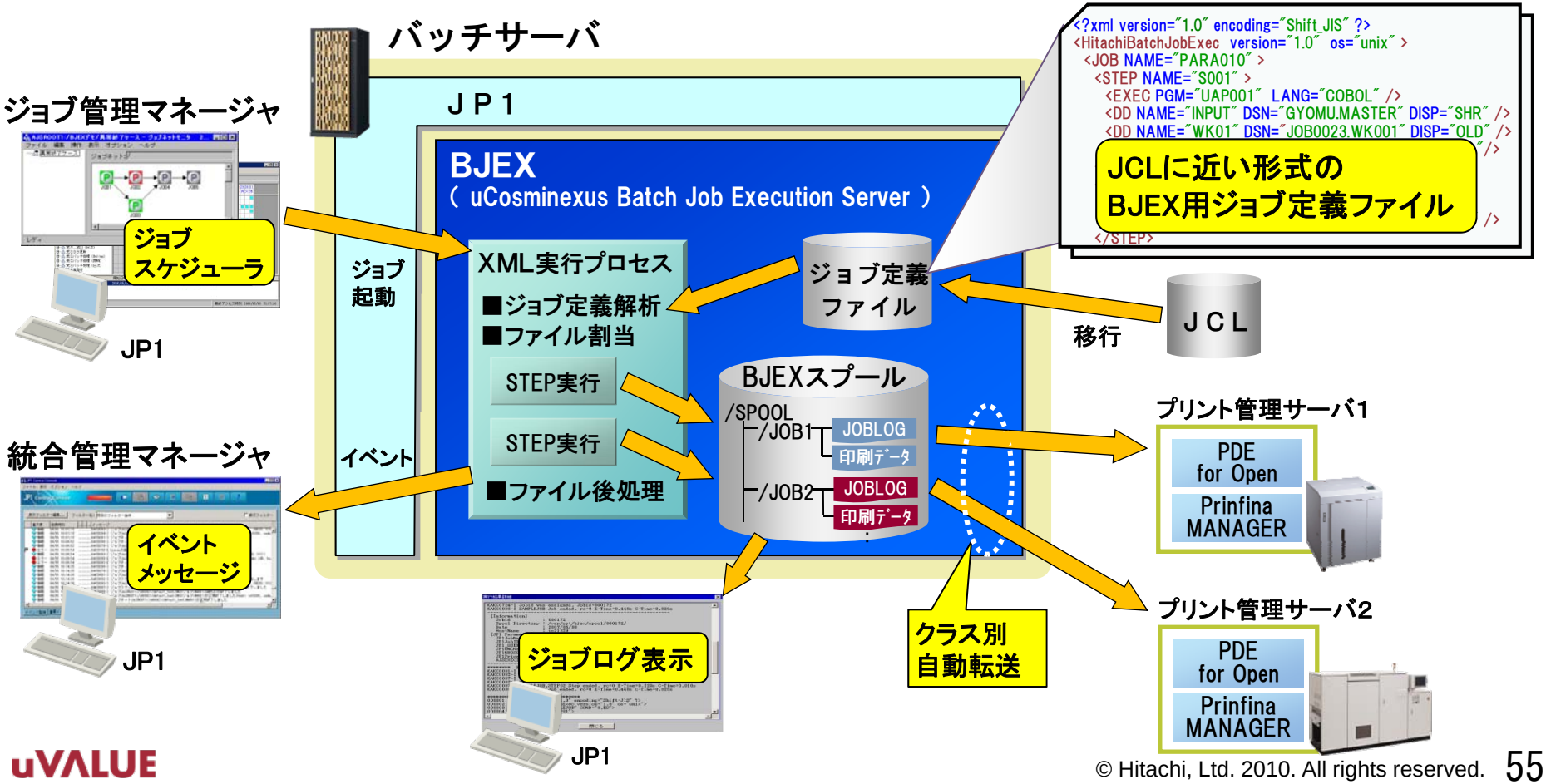


4-10. バッチジョブ実行基盤 BJEXの特長

HITACHI
Inspire the Next

メインフレームのノウハウを活用し、オープン環境でのバッチ業務を支援

- メインフレームのJCLに近い簡単なジョブ定義。XML形式化により高度な編集が可能
- ファイルの一括排他や世代ファイルなど豊富なファイル機能
- BJEXスプール機能による実行結果の一元管理、JP1と連携したジョブログ機能
- PDE for Openとの連携による、帳票運用性の継承(出力クラス運用など)



uVALUE

4-11. 画面・帳票サポートシステム XMAP3の特長

HITACHI
Inspire the Next

メインフレームのオンラインXMAP機能に相当する機能を提供します

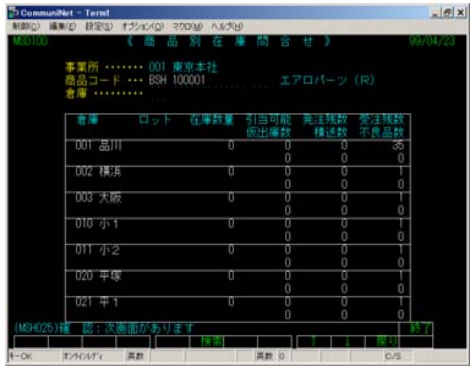
- ストレート移行では、メインフレームイメージのCUI画面で単純移行
- COBOLアプリケーションを流用し、GUI画面にすることも可能

段階的移行が可能

＜ XMAP画面の移行イメージ ＞

メインフレーム ←→ オープン環境

【XMAP2画面】

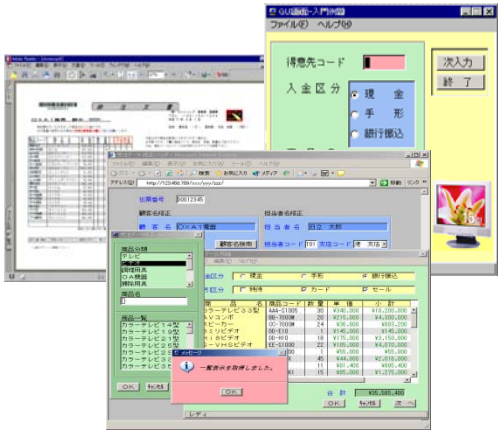


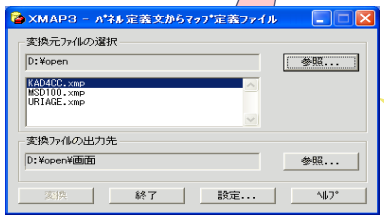
STEP1

【XMAP3のCUI画面】

STEP2

【XMAP3のGUI画面】





XMAP3インポート機能

移行ツールで単純変換

XMAP3のドロー機能などでGUIにカスタマイズ

4-12. 高信頼ノンストップデータベース HiRDBの特長

HITACHI
Inspire the Next

国際標準SQL	国際標準・業界標準RDBMSインターフェースに対応 ●TPモニタ、アプリケーションサーバとスムーズに連携できます。
高信頼性設計	HiRDBはMF高信頼性設計に基づいて開発 ●メインフレームで培われた高信頼設計基準をクリアしています。
開発者直結	HiRDBは国内にソースを保有 ●開発者に直結した迅速かつ的確なサポートを提供
上位互換性	HiRDBはAPおよびDBの上位互換性を保証 ●バージョンアップ時に、アプリケーションプログラム(UAP)の書き換え不要 ●バージョンアップしてもデータベースはそのまま使用可能

計画中

RDB(関係型DB)に加え、階層型DB(構造型DB)にも対応。移行性がさらに向上。

HITACHI
Inspire the Next

Contents

uVALUE

5. まとめ

HITACHI
Inspire the Next

■長年培ってきた資産を安易に捨て去るのではなく、第一に蘇生を考えること

- スクラッチ開発は、安全面でのリスク大＆コスト高で、効果がなかなか出ない
- 用意周到な構想書や計画を練り上げてからのシステム構築ではなく、企業独自のノウハウが蓄積されている既存資産を活かしたシステム構築

■直近だけではなく、将来を見据えた「ITコストの構造改革」を考えること

- まずは、既存資産のスリム化＆オープン化により、ランニングコストを削減
- 次に、変化に強いシステム構造を構築し、将来に渡りランニングコストを抑止
- そして、競争に勝ち残るための新規システム機能拡張へ戦略投資

■マイグレーションの本質は「ITコストの適正化」と「企業競争力の維持・継承」

- 目指すべきは、「企業の強みを活かしたハイスピード＆ローコストなシステムづくり」
- 新しいことをやるためのコストを適正化することが本当のコスト削減
- レガシー資産は企業独自の強みを具現化した第四の経営資源。これを活かしたシステムづくりが企業競争力の維持・継承

参考文献・商標関連

HITACHI
Inspire the Next

【参考文献】

- 経済産業省　平成18年情報処理実態調査結果報告書
  情報処理関係支出の内訳の推移

- COBOLコンソーシアムホームページ(http://www.cobol.gr.jp/)
  第11回 インターネット時代のCOBOL活用セミナー(2006年7月21日実施)
  テックバイザージェイピー「レガシーマイグレーションにおける
  サービス指向アーキテクチャの位置づけ 」

- 日経コンピュータ 2010年2月3日号　読者の声

【商標、他社商品名等の引用に関する表示】


HITACHI
Inspire the Next