

Adabas-natural移行ソリューション

Adams

ADAbas-natural Migratory Solution

Media Knowledge Industrial Co.,Ltd.

第1部 Adabas-natural移行ソリューション概要

Adams

ADAbas-natural Migratory Solution

はじめに レガシー問題の背景

- レガシーの何が問題なのか？

いつかはなくさなければならないが・・・

- 新しいシステムニーズをレガシー上に構築できない！
 - 機能の追加要望や新規に構築したいシステム要件は数多くあるが、今さら汎用機の資産をベースに投資はできない。
(新しいアーキテクチャを取込めない)
- TCO・保守コストの肥大化
 - 汎用機上のH/W、S/Wの保守コストは高コスト
 - 最低限の運用保守に必要なマンパワーも高コスト
- 業務ノウハウの伝承が行なわれない
 - 汎用機上のシステムが行なっている業務の理解は属人化しており、若手のエンジニアに継承されていない。



なぜレガシー移行の障壁が高いのか？

• システム再構築の障壁

- 成功すれば大きな成果となるが、膨大なプロジェクト予算と期間が必要になる。
(システム要件取り纏めへの危険度も高い)
- ERPで置き換える場合でも、置き換えられない部分はアドオン開発が必要であり、その部分がその企業にとってのアドバンテージとなっている。

• コンバートによる暫定的移行の障壁

- 最も低コスト・短期間で行うことが出来る手法であるが、それでも多額のコストは必要。
- そもそも、目に見えない暫定的な措置にコストをかけても、根本的な解決にはなっていない。(問題の先送り)



Adamsについて (コンバート ON フレームワーク)

- システム再構築とコンバートのメリットを同時に実現

- コンバートのメリット

- 機械的なコンバートによる生産性と品質を保証
 - 初期投資の削減、移行期間の短縮、現行機能を安全に移行



- 再構築に限りなく近いメリット

- 「コンバート ON フレームワーク」

- 言語変換の限界をブレイクスルーした最先端技術

- 世界標準技術であるJava技術を基盤に開発標準を統一することが可能
 - 閉じたテクノロジーではなく、オープンな環境に資産を移行することによる将来への発展性が広がる
 - コンバートによる言語変換とは比較にならない保守生産性を実現する

Adamsで実現すること (将来展望へのマイルストーン)

- 運用保守コストの観点から

- 汎用機とオープンシステムの保守二重投資の必要がなくなる。
 - H/W、S/W、人材それぞれのレイヤーでの冗長投資を排除
- プログラム仕様書の自動生成が可能
 - 現状の資産に仕様書が追隨していなくても、移行完了後に自動生成



- システムの拡張性の観点から

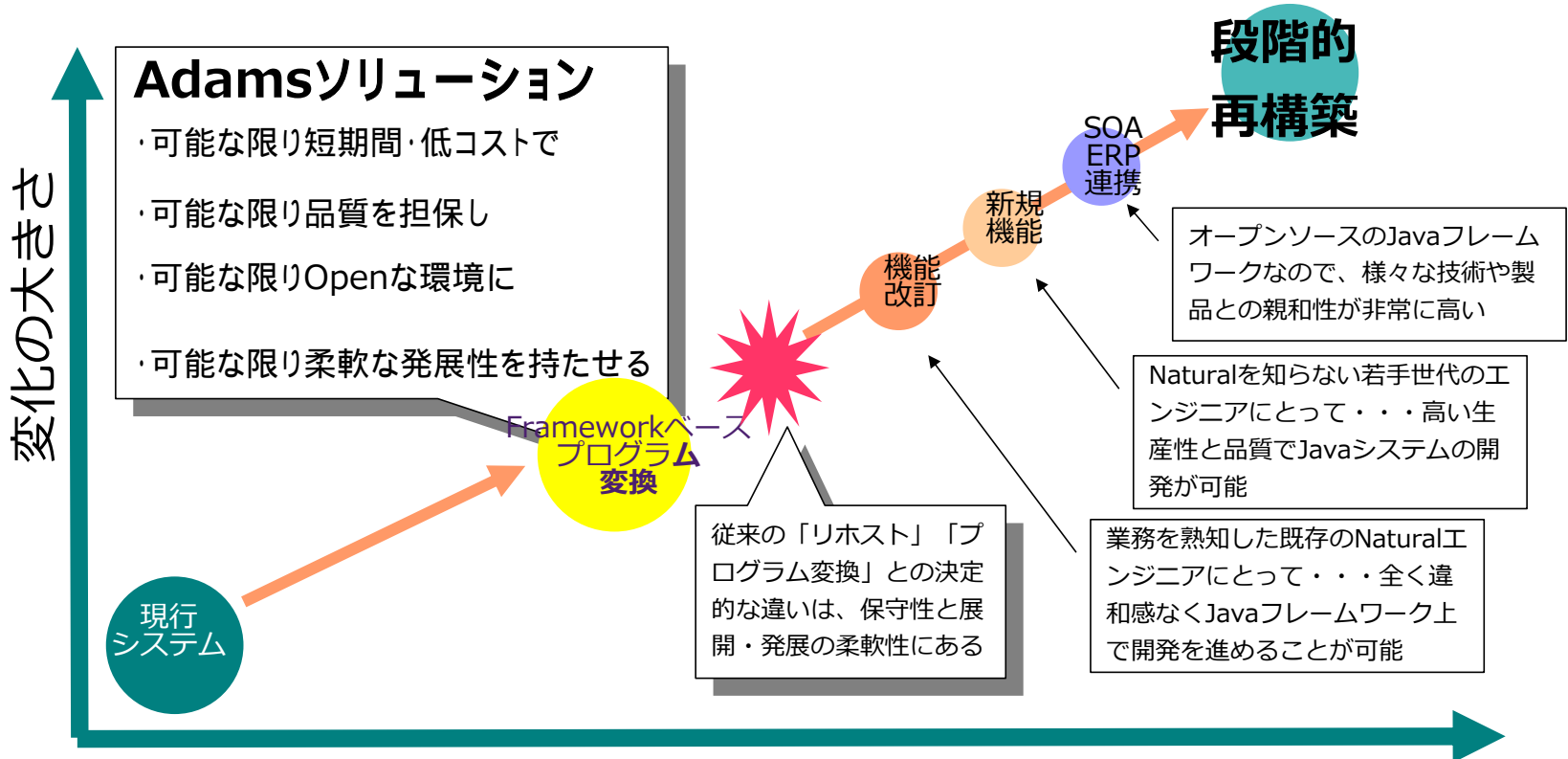
- 全てがオープンな世界標準技術へと移行される為、ERP,SOA、Webサービスなど様々なシステムとの柔軟な連携を実現するためのマイルストーンとなる。

- Humanリソースの観点から（新旧エンジニアの技術移転）

- 現行システムの業務仕様を把握しているが、Java技術がないエンジニアを最大限に活用することが可能
- Java技術を習得している若手エンジニアは違和感なく追加改定や新規開発が可能

Adamsのマイグレーションモデル

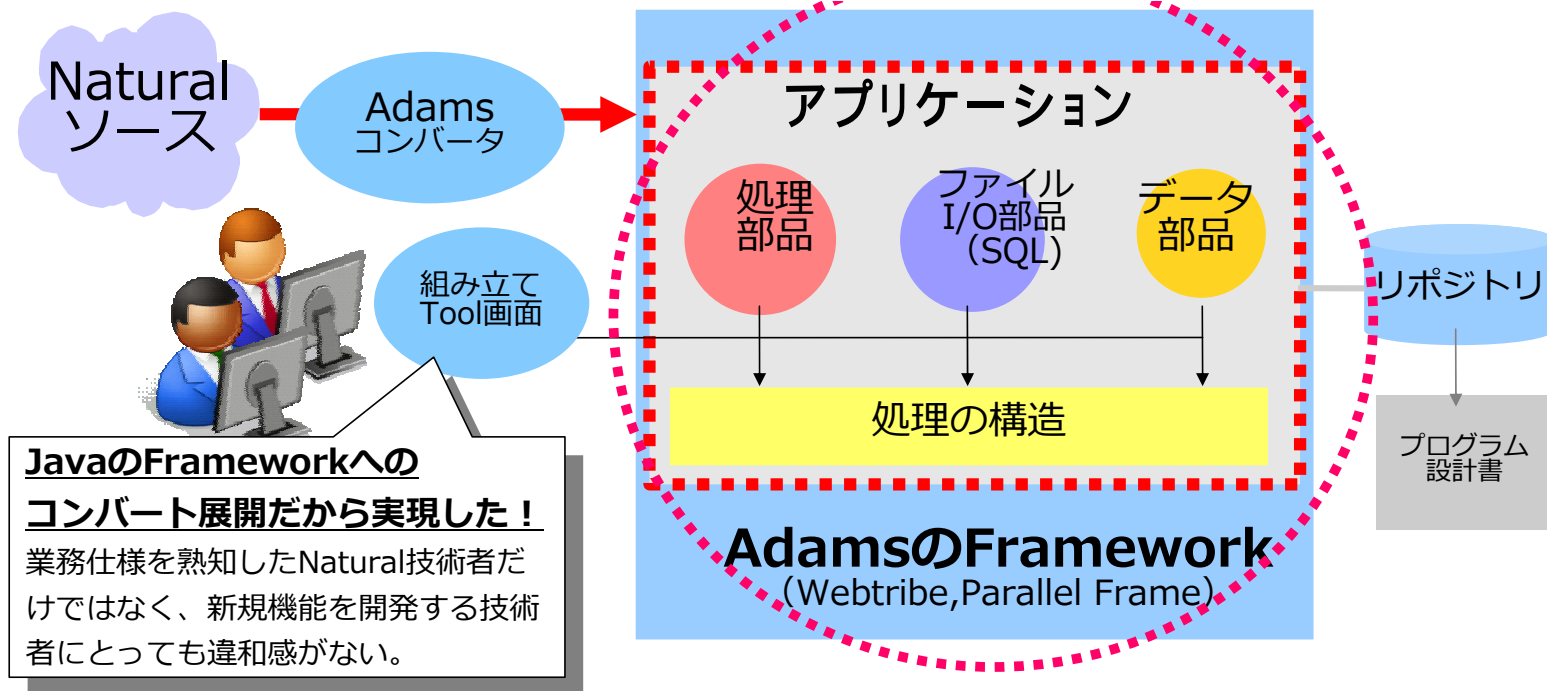
プログラム変換のメリット「品質・コスト・期間」と再構築に迫る拡張性を融合し、段階的再構築を目指す！



移行手法の概念イメージ

既存Naturalの命令などを部品として生成するだけでなく、処理の構造までをそのままフレームワーク上に展開する為、保守性・拡張性が飛躍的に向上

フレームワークを含め全てオープンソース



プランニングと役割分担

- 基本料金制
 <成果物>
 ・簡易棚卸報告書
 ・システム分析報告書
 ・基盤方式ご提案書

お見積り

- 一括請負
 但し、単体テストの範囲
 によって費用が変動。
 全件の実施ではなく、
 品質を保証できる範囲に
 留めることで費用の極小化が
 可能になります。

- 工数請負
 お客様主体の作業となり、
 弊社の要員は作業支援及び
 フレームワーク教育などを
 担当します。

工程	作業概要	役割分担	
		お客様	弊社
1.事前調査/システム棚卸	1. 当社業務提携先であるソフトウェアジェネレーション社のエンジニアリングToolを使用したシステム分析・システム棚卸 2. 現行システムの状況調査・把握、Q&A	○ 各種リソースのご提供	○
2.パイロットプロジェクト	1. 全ての資産に対してコンバートを実施し、お客様の独自要件を抽出 2. 抽出された独自要件に対してのツールカスタマイズ 3. 最終ツール検証	○ 独自要件に関するQ&A対応	○ 独自要件に関する調査・カスタマイズ対応
3.最終コンバート及び開発	1. 全てのプログラムのコンバート 2. 画面や帳票など、手作業による開発		○
4.単体テスト	バッチはジョブ単位、オンラインは画面遷移などの一連の処理を単位としてテストを実施し、現行システムとのコンペアによるテスト検証を実施	○ テストデータの準備、コンペア用のテストデータの準備。テスト確認作業	○ 単体テストの実施
5.各種システムテスト	1. 他システムとの連携検証 2. 処理パターンにあわせた運用テスト 3. 性能評価及びJCLチューニング	○	○ BT処理性能向上のためのJCLチューニング
6.データ移行/運用切替	1. データ移行作業 2. 切替計画・切替リハーサル、その他	○	○ 関連する作業全体の支援及びフォロー作業