



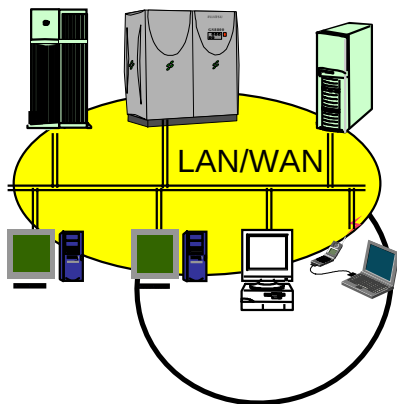
COBOL資産を活用した .NETプラットフォームでのシステム構築事例

平成15年 12月 11日

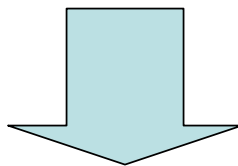
(株)富士通ビジネスシステム
システム本部
Webサービスビジネス統括部
コンポーネント開発センター
西面 孝史



弊社が求めた新商品



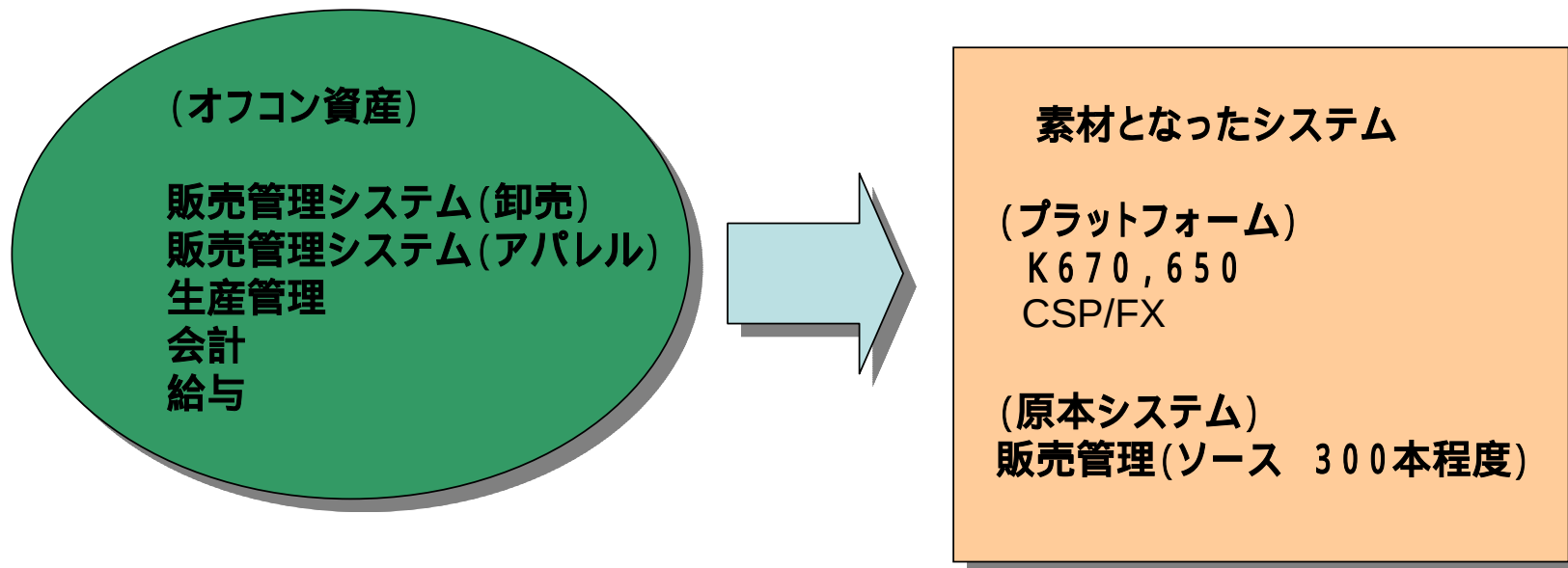
- ・COBOL言語ならソースが読める
- ・既存COBOL資産が膨大
- ・似て非なる業務資産が多数



資産活用、新技術で新商品を？

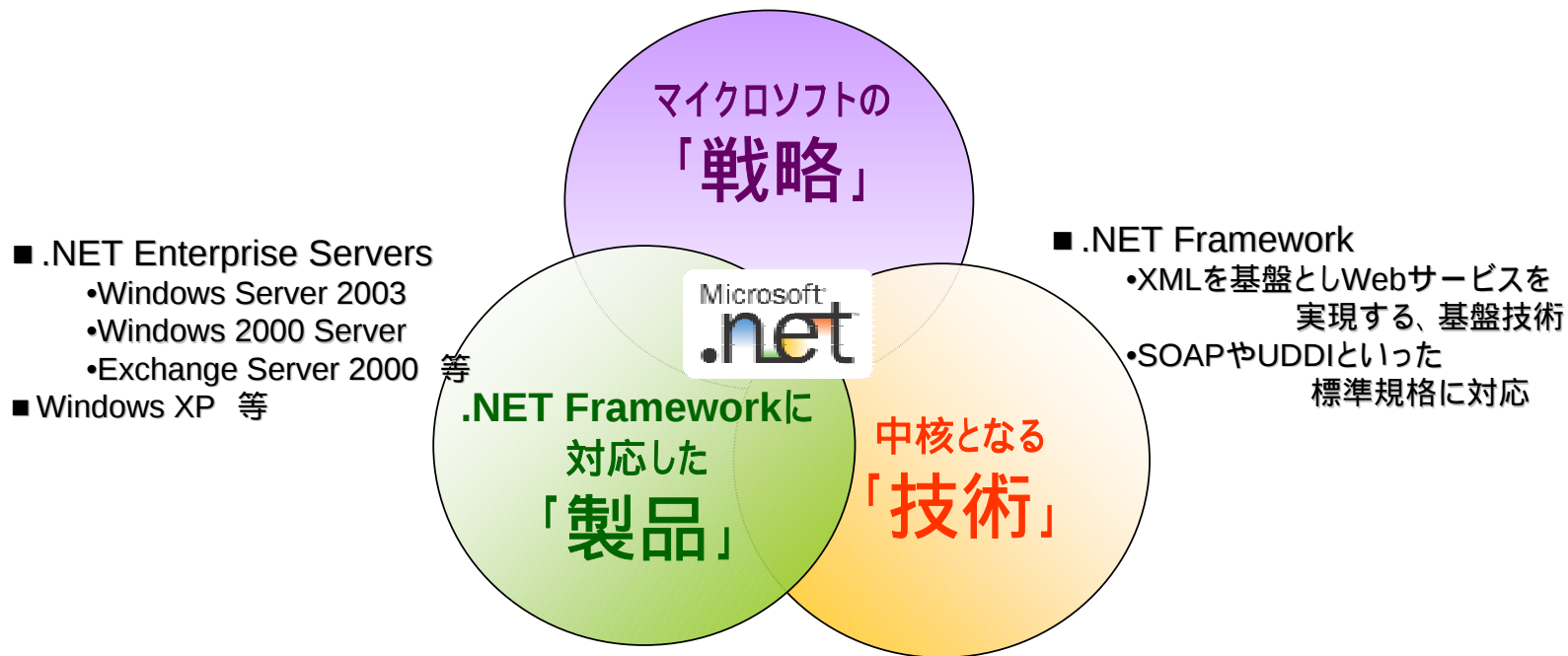


WebASComponentの前身





.NETの登場



生産性

複数言語による開発

柔軟性

クライアントが多種多様

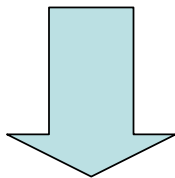
接続性

Webサービスの実装



NetCOBOL for .NETの登場

.NET Frameworkに完全対応した世界初のCOBOL

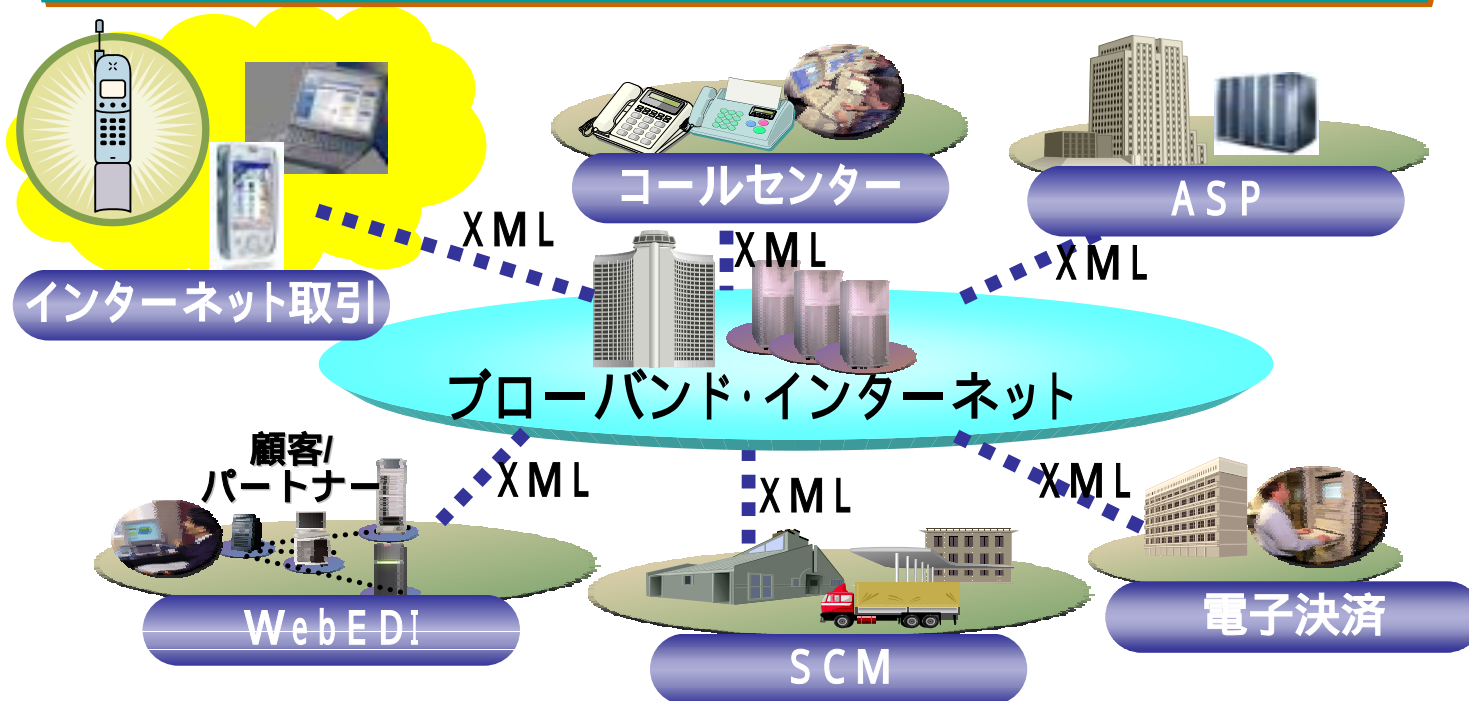


既存のCOBOLプログラムを容易に.NETフレームワーク上で利用することが出来る。



XML Webサービスで広がる可能性

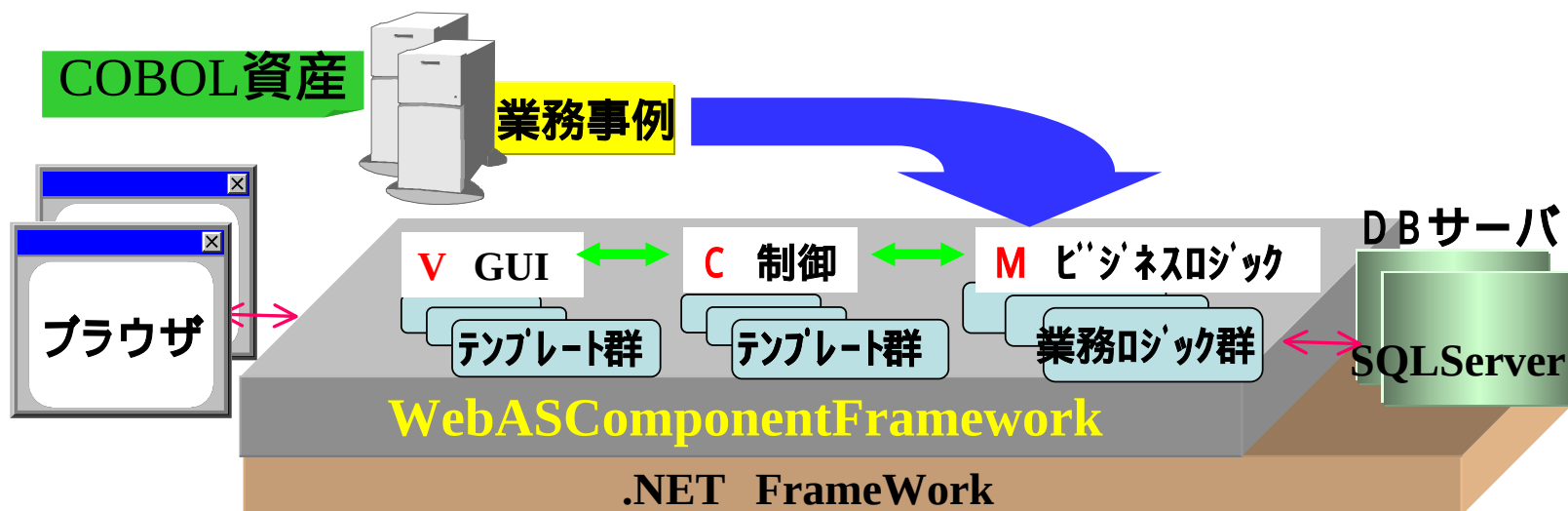
- XML-Webサービスによる企業内/外のサービス連携
- 既存資産の有効活用 / 連携
- 低コスト・短期導入・TCOの削減
- 高信頼性（24h365日，セキュリティ）
- ビジネス変化に対応出来る柔軟性・拡張性





WebAS Component開発へ

- 既存資産の有効活用
- システム構築を開発プロセスも含め体系化(開発標準)
- オブジェクト指向技術を意識させないIMVCモデル





開発概要と開発言語

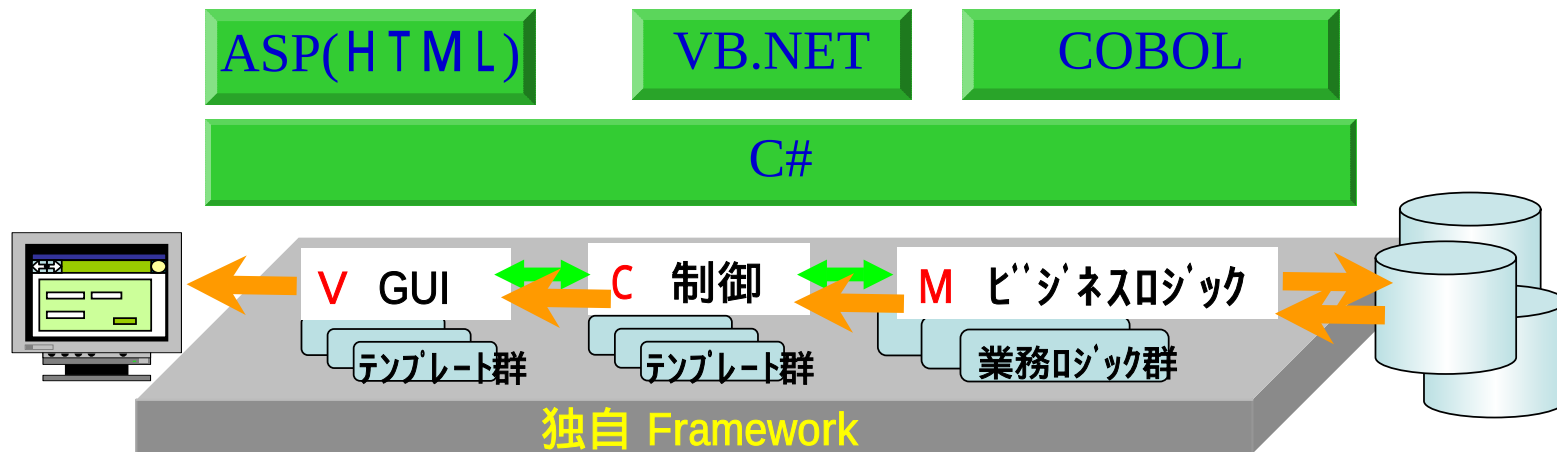
(プロジェクト概要)

既存のオフコンCOBOL資産を.NET上へ移行する

(従来オフコンでは...)



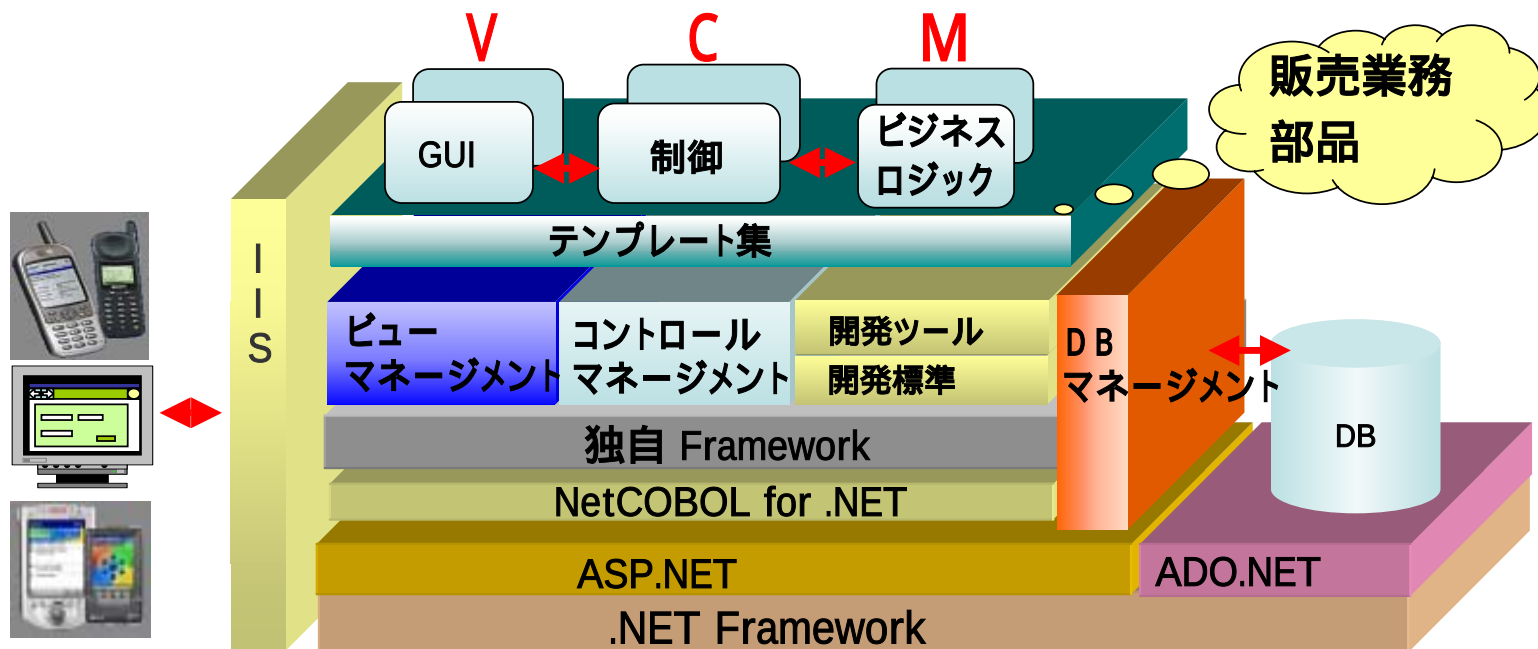
(新規システム「WebAS Component」では...MVCモデルの導入)

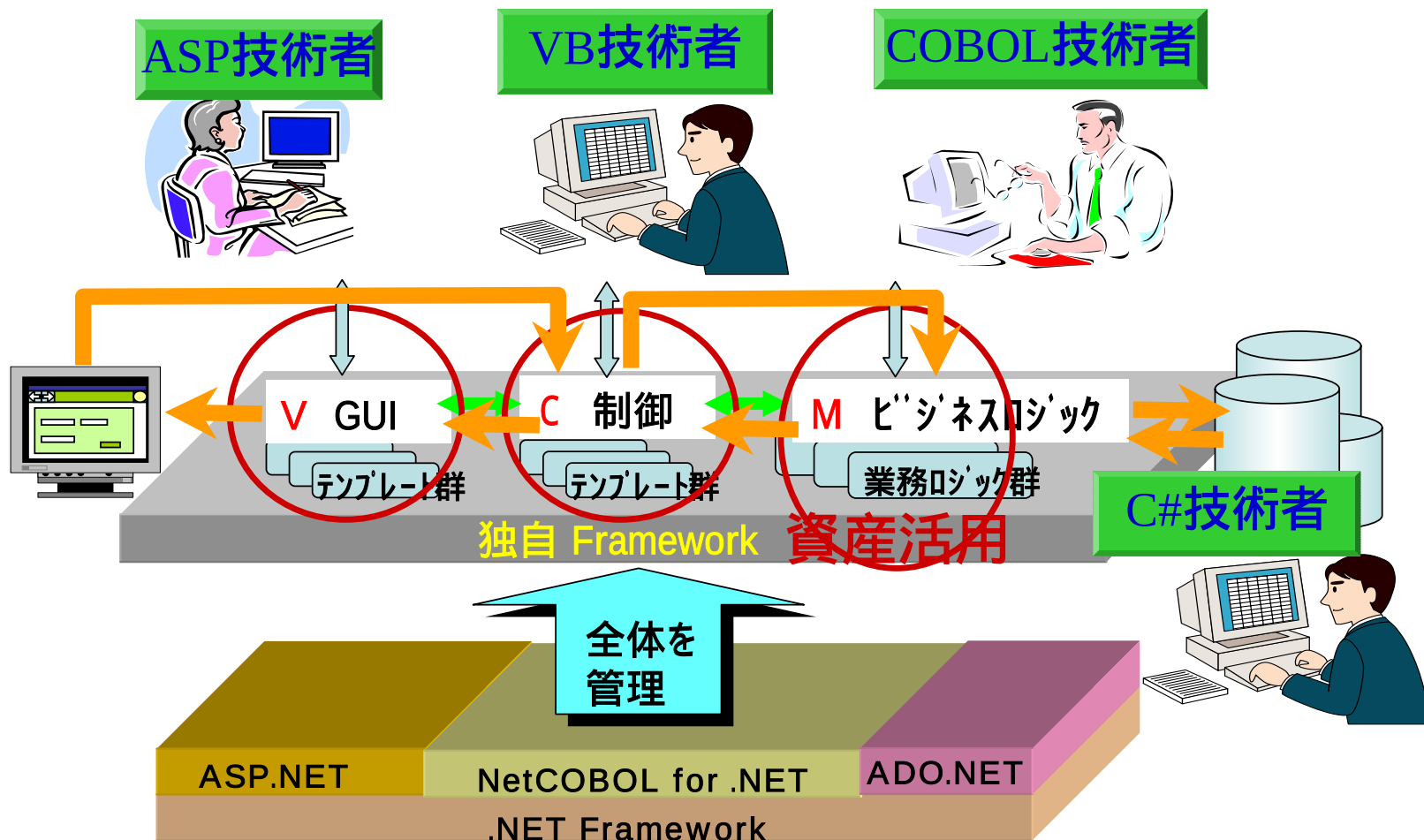




WebAS Componentの構成

- 3階層MVCモデルでのアプリケーション構築
- 開発プロセスを包含した独自フレームワーク
- .NET完全対応でのWebシステム
- 業務コンポーネントをテンプレートとして実装







弊社開発事例でのCOBOL技術者の活用

- ・業務仕様の理解には欠かせない人材が多い
- ・業務ロジックに特化してコーディングさせる
 - オブジェクト指向型COBOL (OOCOBOL)を意識させない
仕組み作りが欠かせない(テンプレートなど)

従来の手続き型COBOLを極力利用可能にする具体例

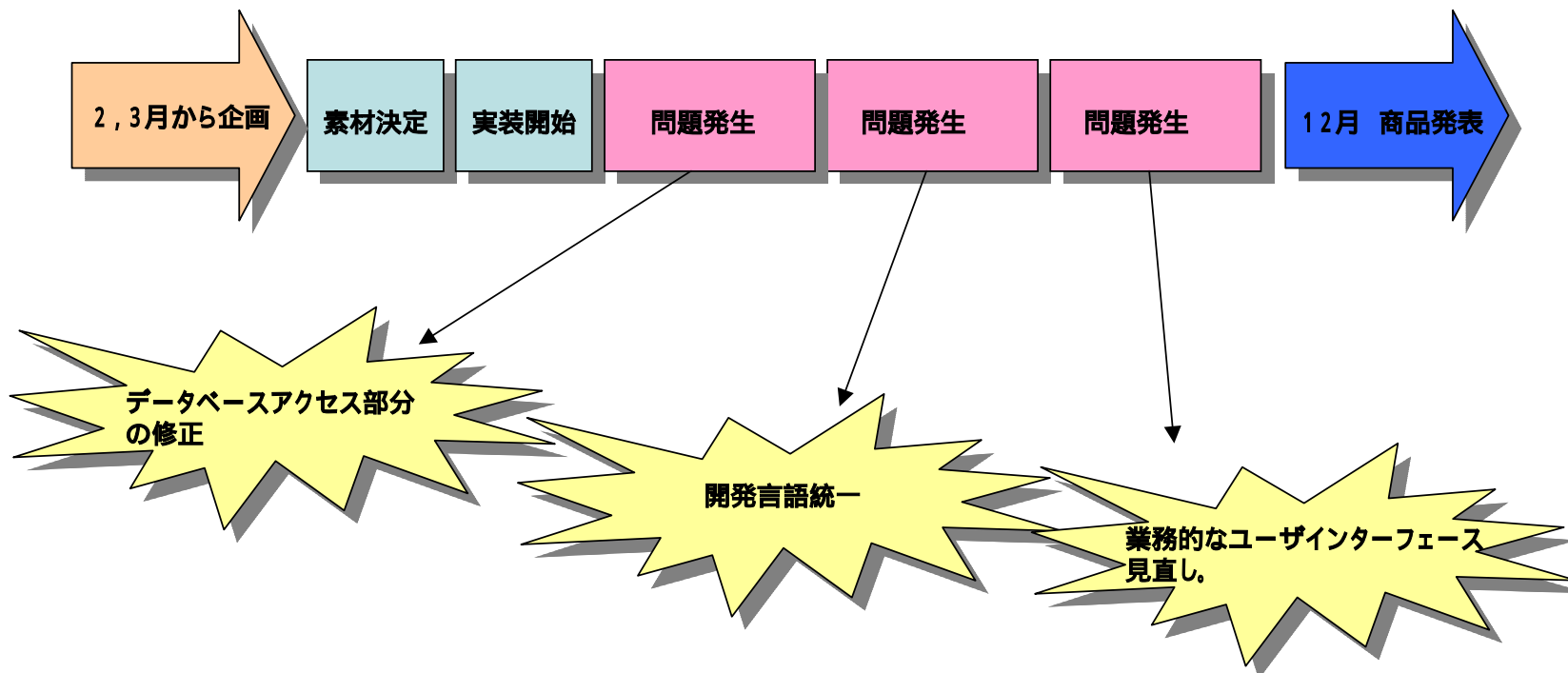
- ・プログラム間連絡機能は、INVOKE文等のオブジェクト呼び出しを極力利用しないで、PERFORM文等の従来文法を多用する
 - ・サブルーチンには、IDENTIFICATION DIVISIONで始まる既存資産をそのままを流用する
-
- ・開発中にオブジェクト指向を習得してもらう



移行の結果

移行期間 : 約10ヶ月

開発メンバー : 約8名 (開発ピーク時)





COBOLと.NETの融合（課題）

.NETで活用するためには(課題)

- ・ファイル構造をどうする？
- ・データアクセス部分をどうする？
- ・画面系のロジックをどうする？
- ・オフコン固有の機能(VLD = 仮想ファイル)をどうする？
- ・データタイプをどうする？



COBOLと.NETの融合（解決策）

.NETで活用するためには(解決策)

- ・ファイル構造をどうする？
RDB構造へ単純移行(→お勧めできない)
- ・データアクセス部分をどうする？
従来利用していたREAD / WRITEインターフェースで
データアクセスできるクラスを用意(文法化)
- ・画面系のロジックをどうする？
再利用不可のため、別途新規開発が必要
- ・オフコン固有の機能(VLD = 仮想ファイル)をどうする？
トリガーなどデータベース機能を利用
- ・データタイプをどうする？
データ型のマッピングを規約化



NetCOBOL for .NETの開発画面

The screenshot displays the NetCOBOL for .NET development environment. The main window shows the source code for the SUITAN.cob file. The code is written in COBOL and includes sections for initialization, data movement, and performance. The left pane shows the project structure, and the right pane shows the file explorer. The bottom status bar indicates the current line and column.

```
Microsoft Development Environment [FJB] - ServiceMaster.cob
L1SUBSTAN - Fujitsu NetCOBOL for .NET [デザイン] - SUITAN.cob
ファイル 編集 表示 プロジェクト 拡張 参照 グループ ツール ウンドゥ 上代
SUITAN.cob ServiceMaster.cob
SUITAN.cob
SUBROUTINE SUITAN
  ADDEND-DE.
  EXIT.
  .....
  JYODAI-SEC SECTION.
  .....
  JYODAI-DE.
  * ( 上代準備宣言 )
  * ( 金種別商品マスタより )
  INITIALIZE
    MOVE SRS-TMRITI TO SY-TANSA-AREA.
    MOVE SRS-TMRREN TO SY-TMRREN.
  PERFORM VARYING IDEX FROM 1 BY 1 UNTIL IDEX > 2
    MOVE SRS-JYODMD(IDEX) TO SW-KAIYMD(IDEX)
    MOVE SRS-JYOTMI(IDEX) TO SW-TN (IDEX, 1)
    MOVE SRS-JYOTNI(IDEX) TO SW-TN (IDEX, 2)
    MOVE SRS-JYOTNO(IDEX) TO SW-TN (IDEX, 3)
  END-PERFORM.
  PERFORM TANSA-SEC.
  IF SY-TANSA NOT = ZERO
    MOVE SY-TANSA TO SBTAN-JYOTM
    GO TO JYODAI-DE
  END-IF.
  * ( 商品マスタより )
  INITIALIZE
    MOVE SYO-TMRITI TO SY-TANSA-AREA.
    MOVE SYO-TMRREN TO SY-TMRREN.
  PERFORM VARYING IDEX FROM 1 BY 1 UNTIL IDEX > 2
    MOVE SYO-JYODMD(IDEX) TO SW-KAIYMD(IDEX)
    MOVE SYO-JYOTMI(IDEX) TO SW-TN (IDEX, 1)
    MOVE SYO-JYOTNI(IDEX) TO SW-TN (IDEX, 2)
    MOVE SYO-JYOTNO(IDEX) TO SW-TN (IDEX, 3)
  END-PERFORM.
  .....
END-SUBROUTINE SUITAN
```



問題点

- ・COBOLでの画面制御はコーディング量が増える
- ・クラスライブラリの扱いが他の.NET言語と違う
- ・インテリセンス機能がピリオドで発動しない
- ・COBOLのデータ型と.NETのデータ型が異なる

COBOLを活かす指針

- ・プレゼンテーション層、ロジック層で言語を使い分ける
- ・数値計算など事務処理言語が得意とする部分を活用する
(ex:VBで作成した計算ロジックで数値誤差が発生)



移行開発では、従来のCOBOL言語の利点を活かせること、新しい機能(.NET Framework)を効率的に活かせること、この2点の「バランス」が大事だと思います。

また、COBOLアプリケーションがしっかり構造化されていれば、資産の再活用が実現しやすいと思います。

なぜなら、「手続き型言語(COBOL)は構造化」、「オブジェクト指向言語(C#、VB.NET)はクラス化」、詰まる所、この2つの言語は「再利用、可読性、拡張性」という同じことを求めているからです。

そしてCOBOLでの構造化がなされていれば過去の資産の活用を実現しやすくなるはずです。



導入事例 1 基幹業務の全面Web再構築

企業名 ・ A株式会社
年商 ・ 約10億円
業種 ・ 包装資材卸売業

評価ポイント

- ・ 顧客要件を満たしたシステム機能(カスタマイズ込)
- ・ 最新技術(.NET)を使用した素材への将来性(安心感)
- ・ 既存PCと新規導入PCの混在によるPC資産の有効活用
- ・ 情報分析ツール(DataNature 4, DBPortSE)の導入

C / S パッケージシステムからのレベルアップ。既存PCも再利用する等TCOの削減を主眼に行いつつ、現行システムと同等のカスタマイズも行い、機能面も維持したいという要望を実現。
情報分析ツールにより基幹データを有効利用し、販売状況の分析に活用。
インターネットのイメージで運用でき、パート従業員も無理なく運用が可能。





導入事例2 発注業務の改革

企業名 ・ C 株式会社
年商 ・ 約200億円
業種 ・ 機械器具卸売業

評価ポイント

- ・ 既存のホスト基幹システムとの連携
- ・ 業務プロセスの短縮を実現
- ・ 関連会社を含めた集中購買によるコスト削減
- ・ 約2ヶ月での構築が可能

今まで各事業所バラバラに手作業・紙で行っていた取引先への見積依頼・発注業務をシステム化し、調達本部と各事業所間の業務のやりとりを社内ネットワークを用いて行いたいとの顧客要望から商談化。今回、購買調達システム(WebAS Component)導入により分散購買から集中購買へと業務スタイルの変革に成功。

