

IT資産継承勉強会 第2回

VB資産移行について

2011年 10月 26日

 宇部情報システム

URL <http://www.uis-inf.co.jp>



当社HPも是非ご覧下さい。
クリックで、HPへ移動します。

Visual Basic .NETが提供されて10年が経とうとしていきます。OSのバージョンアップのたびに旧来のVB6システムの稼働環境は厳しい状況に追い込まれてきております。

そこで「IT資産勉強会 第2回目」として、VB資産移行に関する弊社の取り組みをご紹介します。

VB6システムのマイグレーションを行う際のご参考にしていただければ幸いです。

1. Visual Basic の動向

- (1) VB6の動作作環境
- (2) VisualBasic.NETの登場
- (3) MSアップグレードツール

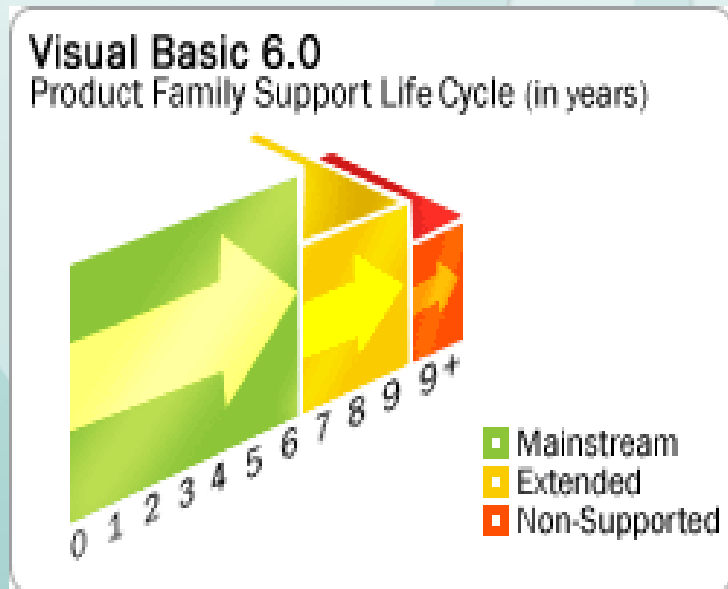
1-(1) VB6の動作環境 -1

◆VB6のライフサイクル

開発ツールとしてのメインストリームを .NETに引き継ぎ、Visual Basic 6.0 は2008年4月8日を以て製品サポートを終了した。

＜Visual Basic 6.0 のライフサイクル＞

VBの製品サポートは、販売開始から6年間で「メインストリームフェーズ」、その後3年間で「延長フェーズ」とし、2008年4月8日に9年間にわたる製品サポートを終了した。



1-(1) VB6の動作環境 -2

◆VB6のサポートは終了したもののVB6は今でも多くの企業システムで利用されているのが現状。
この状況への配慮か、マイクロソフトは新OSでもVB6の動作環境を提供している。

- VB6実行環境
- 開発ツールの動作環境

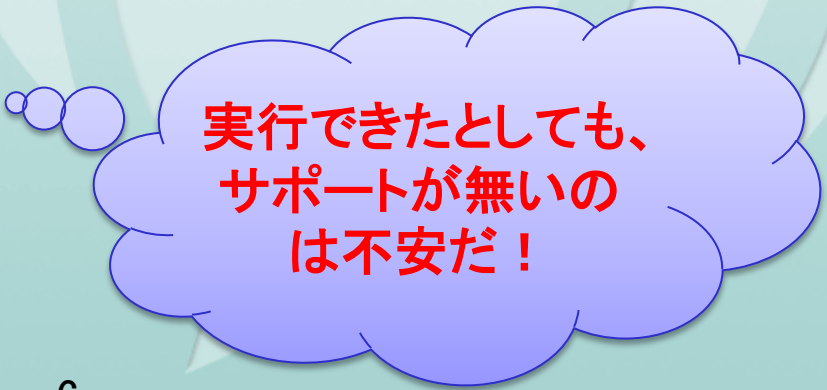


今後もVB6は
使用できる？

◆ただし、VB6のサポートが延長されたということではない。

◆VB6実行環境

- マイクロソフトは、VB6アプリケーションがWindows Vista/7, Windows Server 2008 でも“そのまま”動作するライブラリ(VB6ランタイム)を提供。
- これにより問題なく動作するVBプログラムも多数存在する。
- サードパーティ製ライブラリ(ocx,dll)の動作はサポートされていない。



実行できたとしても、
サポートが無いの
は不安だ！

◆開発ツールの動作環境 (Visual Studio 6.0)

- Windows XP, Vista, 2003 Server で動作する。
- Windows7 では動作しない。旧OS上に構築した開発環境を維持していく必要がある。
- 実行環境(ランタイム)の扱いと同様で、利用は自己責任が前提となっている。
- サードパーティ製品を開発に使うことはサポートされない。
- 64bit版のプログラム開発をサポートしていない。

VB 64bit非対応

1-(1) VB6の動作環境 -5



「VB6実行環境(ランタイム)」の参考ページ

<http://www.microsoft.com/japan/msdn/vstudio/support/tools.aspx>

Windows Vista®、Windows Server 2008®、および Windows 7 における Visual Basic 6.0 のサポートについて

概要

Visual Basic チームは、Visual Basic 6.0 アプリケーションが Windows Vista と、Windows Server 2008 および Windows 7 でも "そのまま" 動作する互換性を提供することを表明します。

Visual Basic チームは、Windows XP で実行できる Visual Basic 6.0 アプリケーションが Windows Vista と、Windows Server 2008 および Windows 7 でも実行できるようにすることを目標としています。また、Windows Vista と、Windows Server 2008 および Windows 7 で実行できる Visual Basic 6.0 の開発環境を提供するための取り組みも行っています。このドキュメントで説明するように、Visual Basic 6.0 ランタイムは、Windows Vista と、Windows Server 2008、および Windows 7 のライフサイクルをとおしてサポートされます (メインストリーム サポートが 5 年、延長サポートが 5 年の合計 10 年になります)。サポート ライフサイクルの詳細については、<http://support.microsoft.com/gp/lifepolicy> を参照してください。

Visual Basic 6.0 のサポート ライフサイクル

Visual Basic 6.0 のランタイム バイナリは Windows Vista、Windows Server 2008 または Windows 7 に同梱されており、これらのファイルのサポートも提供されますが、Visual Basic 6.0 IDE や Visual Studio 6.0 自体のサポート ポリシーに変更はありません。これらの製品の延長サポートは 2008 年 4 月 8 日で終了します。

Windows 7

Windows 7 の発表に伴い、Windows 7 における Visual Basic 6.0 サポートを明確にするため、このドキュメントを更新しました。

Windows 7 における Visual Basic 6.0 ランタイムは OS に同梱する形で出荷されます。Windows 7 に同梱された Visual Basic 6.0 ランタイムは OS の一部としてサポートを提供し、OS のサポートライフサイクルに準拠いたします。Visual Basic 6.0 に対するサポート内容は Windows Vista における Visual Basic 6.0 と同様となります。なお、将来のバージョンの Windows での、Visual Basic 6.0 のランタイムの提供およびサポートは予定していません。

曖昧な
対応

Win8は
非対応

「開発ツールの動作環境」の参考ページ

<http://www.microsoft.com/japan/msdn/vstudio/support/tools.aspx>

開発ツール対応 OS 一覧

凡例
○ : 対応しています
× : 対応していません

	(*1)	(*2)	Windows Millennium Edition (Me)	Windows XP	Windows Server 2003	Windows Vista	Windows Server 2008	Windows Server 2008 R2	Windows 7
	Windows NT 4.0	Windows 98	Windows 2000						
	○ (*3)	○ (*3)	○	○ (*8)	○ (*13)	○ (*20)	○ (*20)	×	×

◆ Visual Basic .NETの登場

VB6がまだ全盛であった2002年に後継の Visual Basic .NET を発売。

.NETではオブジェクト指向への言語対応を強化したところが大きな変更点である。

◆ オブジェクト指向を強化した影響

- VB6との互換性を犠牲
- プログラマにオブジェクト指向スキルを要求

.NETへの移行を阻害する要因といえる

1-(2) VisualBasic.NETの登場 -2



◆.NET発売からのリリース

Visual Studio .NET
2002,2003

- .NET Framework 1.0、1.1

Visual Studio 2005

- .NET Framework 2.0、3.0
- VB6からのアップグレードツール付属

Visual Studio 2008

- .NET Framework 3.5
- VB6からのアップグレードツール付属

Visual Studio 2010

- .NET Framework 4
- アップグレードツールなし

※ マイクロフトは、VB6から.NETへの移行を支援するアップグレードツールを提供

◆マイクロソフトの.NET移行支援ツール

1	アップグレードツール	Visual Studio2005、2008に付属しているVB6→.NETへの変換ツール
2	Microsoft.VisualBasic.Compatibility	- VB6との互換性を保つために提供されているライブラリ - アップグレードツールでも使用される .NET Framework 1.0~3.5 .NET Framework4.0では互換性のため残されておりワーニングの扱いになってる
3	Microsoft.VisualBasic.PowerPacks	VB6の印刷コードにより近いままで.NETで印刷を実現できるライブラリ

1-(3) MSアップグレードツール -2

◆アップグレードツールの実用性

VB6と.NETのプログラムソースに互換性はない



アップグレードツールは使えるのか？

- ・.NETから変更又は廃止された命令はアップグレードされない。
- ・誤ったロジックにアップグレードされてしまうことがある。
- ・サードパーティ製のライブラリは全面的に手修正が必要になる。
 .NET版にバージョンアップしてもソースの修正は必要である。



簡単には.NET化できそうにない

2. VB6資産移行の検討

- (1) ユーザが抱える問題
- (2) 移行の動機
- (3) VB6移行ソリューション

2-(1) ユーザーが抱える問題 -1



2002年に Visual Basic のメインストリームを .NETに引き継いだVB6は、2008年にはサポートを終了。

Windows OS もバージョンアップを重ね、次期バージョンのOSからはVB6実行環境のサポート予定はなくなっている。

このような状況の中、VB6システムユーザーにどのような問題が起きているのでしょうか。。。

2-(1) ユーザーが抱える問題 -2



◆VB6システムユーザーの問題や悩み

- ①VB6システムが残っていると、新OSの導入ができない
- ②Windows XP のPC入手が困難になったため
 - VB6システムのクライアント環境の維持ができない
 - システム保守環境（開発環境）の維持ができない
- ③システム機能には満足しているが、いつまで使えるのか不安
- ④VB6システムを新OSに対応したいが、VB6システムが大量にあるため膨大なコストと時間が掛る
- ⑤内部統制等の制約によりサポートされない製品(VB6)が利用ができない

2-(2) 移行の動機

◆VB6システムへの移行の動機

理想は、システム機能の刷新が必要になるまで使い続けたい



VB6システム移行動機の現状

ハードウェア老朽化

周辺ソフトのVB6動作保証

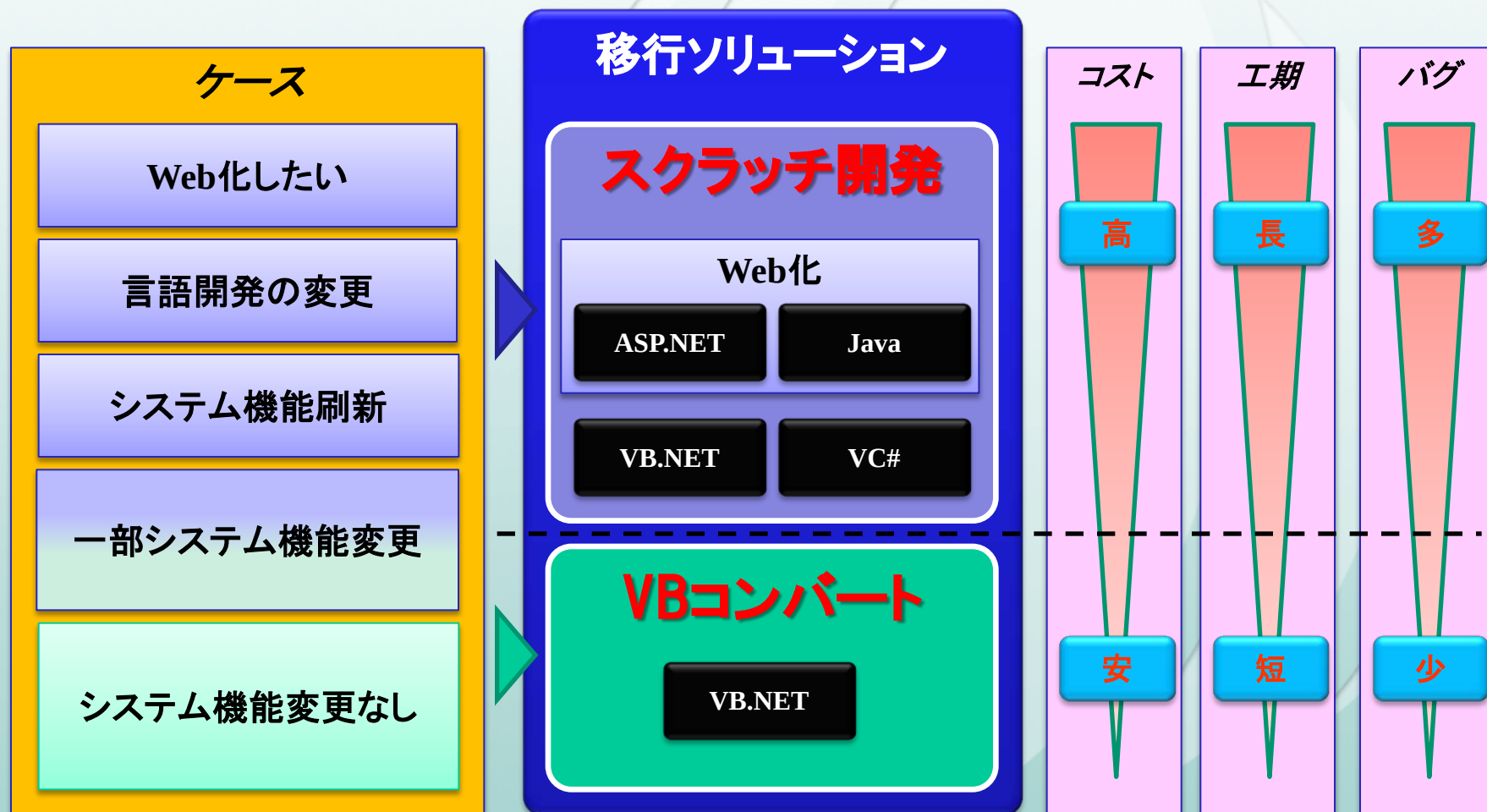
内部統制

ハードウェア環境の調達問題

クライアントOSバージョンアップ時の障害

2-(3) VB6移行ソリューション -1

◆VB6システムの移行ソリューション



2-(3) VB6移行ソリューション -2

◆こんなユーザーにお勧めしたいVBコンバート

システム機能は現行そのままが良い



.NETシステムへのコンバートがお勧め

＜VBコンバートのメリット＞

VB6プログラムソースの再利用が可能になり、

- ・短期間
- ・低コスト
- ・高品質

にシステムを再生することが可能。

VBコンバート

弊社のソリューションで更に加速します！

3. VBコンバートについて

- (1) VBコンバートとは
- (2) VBコンバートのハードル
- (3) VBコンバート手順
- (4) VBコンバートツール

◆VBコンバートでは何をするの？

- Visual Basic 6.0 (VB6) のプログラムソースコードを Visual Basic .NET (.NET) のソースコードへ置き換え、.NET環境下での稼働を可能する。
- コンバートの内容
 - ・ VB6のコントロールを.NET標準のコントロールに置換する。
 - ・ VB6のソースコードを.NETの仕様に準拠した命令(メソッド、関数)に置き換えを行う。
- コンバートの影響
 - ・ .NETで画面のコントロールが大幅に変更されたため画面の操作ロジックへの影響が大きい。
 - ・ ビジネスロジック内の命令変更は少ないので、ビジネスロジックへの影響は比較的小さい。

3-(2) VBコンバートのハードル -1



- ◆ VB6と.NETは言語仕様が大きく異なっている。
 - .NETでは非オブジェクト指向的な言語要素が廃止された。
 - Visual Studio .NETにアップグレードツールが付属しているが、このツールでコンバートしてもエラーは多く残っている。
- ◆ .NET用のコントロール(画面部品)はVB6とは互換性がない(特にサードパーティ製コントロール)

弊社のソリューション

弊社独自のVBコンバートツールを開発

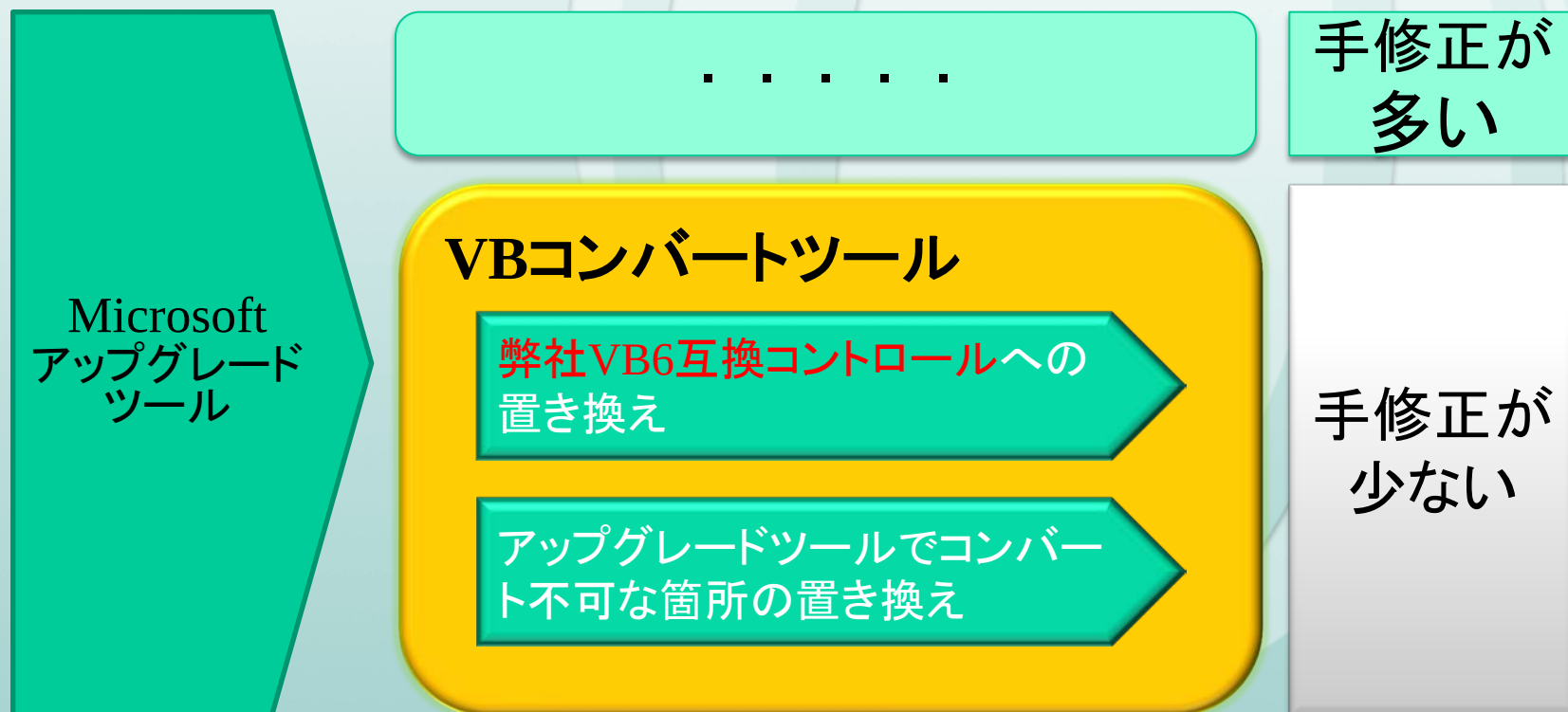
VB6用コントロールと互換性を持った**コントロール**

.NETアップグレードツールを補完する**コンバートツール**

3-(2) VBコンバートのハードル -2

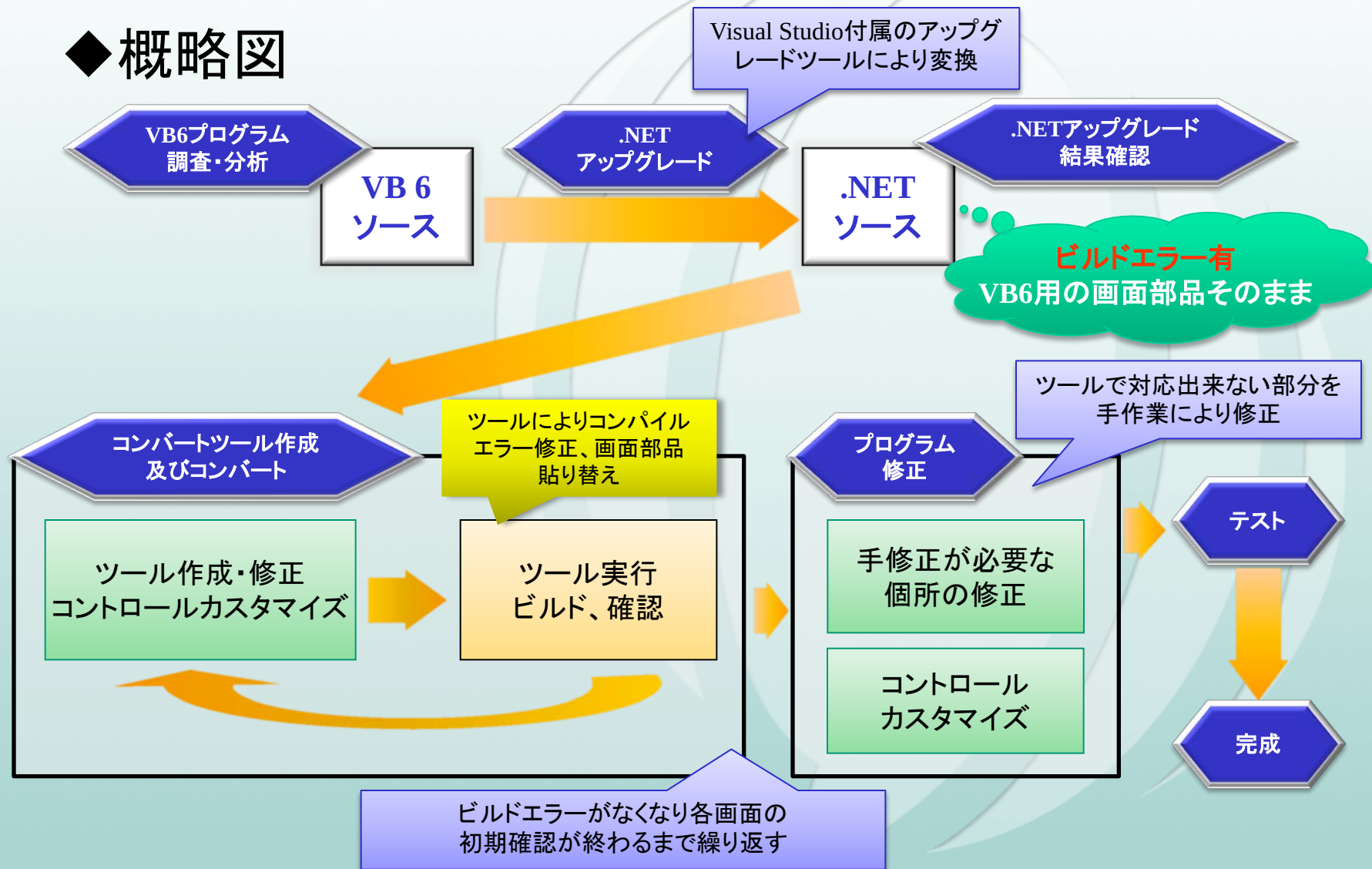
◆独自のコンバートにより手修正を軽減

- 同じ修正を規則的に修正(共通化)
- プロジェクト固有の修正もツールをカスタマイズすることにより機械的に修正



3-(3) VBコンバートの手順 -1

◆概略図



3-(3) VBコンバートの手順 -2

◆コンバート手順

VB6プログラム調査・分析

標準コンバートできるか？
データアクセスは標準的なコーディング？ ...

コンバート対象となるVB6プログラムを調査分析する

- ・機能数 ・ステップ数
- ・使用コントロール
- ・使用ユーザコントロール
- ・データアクセス／帳票ツール
- ・コーディング様式
- ・コーディングルール など

.NETアップグレード

VB6ビルド

.NET標準
アップグレード

コンバートできないビルドエラーがある

VB5などの古いバージョンの場合はVB5→VB6へのバージョンアップが必要になる

まずは、VisualStudio2005(もしくは2008)付属のアップグレードツールによりVBプロジェクト(EXE)単位にコンバートを行う。

3-(3) VBコンバートの手順 -3

.NETアップグレード 結果確認

アップグレードではアップグレード時のエラー、
ワーニング等がコメントとして挿入される

UPGRADE_WARNING: オブジェクトobjCom.AnsiLeftB の既定プロパティを解決できませんでした
UPGRADE_ISSUE: App プロパティApp.PrevInstance はアップグレードされませんでした

アップグレード結果、VB6分析内容から、アップグレードで正しくコンバートできないコードをVBプロジェクト毎に洗い出す。

コンバートツール作成 及びコンバート

ツールの作成・修正
コントロールのカスタマイズ

共通

プロジェクト固有

ツール実行

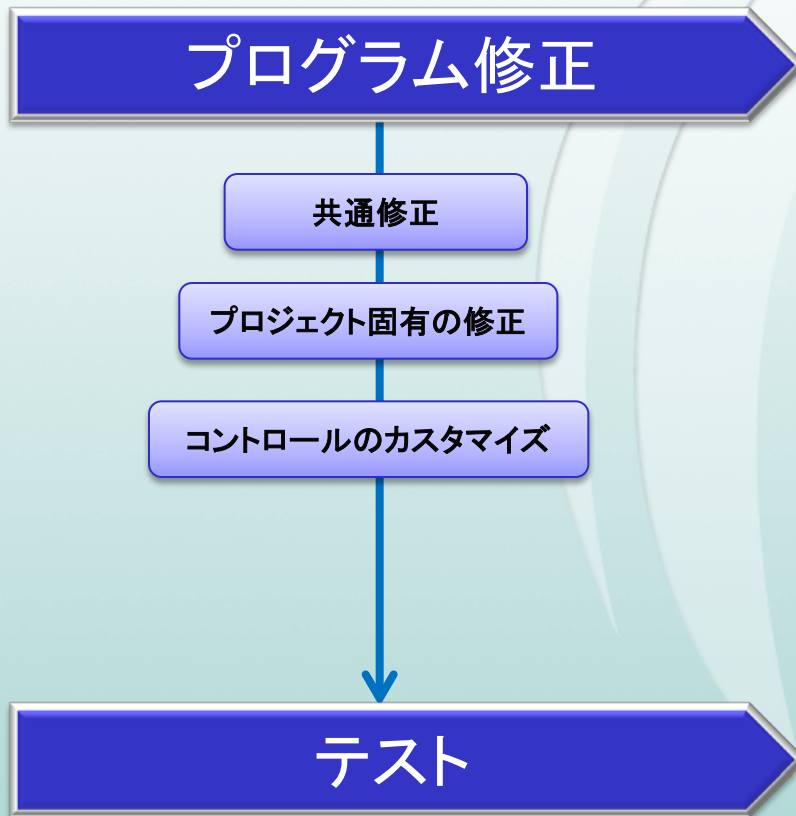
アップグレードできないコーディングを.NETの仕様に準じたコーディングに変更するためのツールをVBプロジェクト毎に作成し、コンバートする。

※正しいコーディングに置換するツールを作成するイメージ

弊社VB6互換コントロール／ユーザコントロールをカスタマイズする。

コンバートして問題が残る場合は、再度ツールを修正してコンバートする。

3-(3) VBコンバートの手順 -4



コンバートツールで対応できない部分の手修正を行う。

- ・画面レイアウトの調整
- ・Spread、Inputmanなどに対応した弊社VB6互換コントロールを修正
- ・コンバートツールにより機械的に置き換えできない個所の修正
- ・VB6時点の共通部品をDLL化 など

動作検証を行う。

ここで発見された、不具合は類似の調査も含め、別VBプロジェクトのコンバート工程・プログラム修正工程にフィードバックし早期解決を行う。

◆ 独自開発したVB6互換コントロール

- サードパーティ製コントロールを.NETコントロールへ
 - ・ サードパーティ製コントロールを使用しないため、可能な限りVB6と同じコーディングで処理できるように、.NET標準コントロールをもとに作成したコントロール群を作成。
 - ・ また、コンバートの際に必要なに応じてコントロール群のカスタマイズを行い柔軟に対応。

サードパーティ製コントロールを
推奨しない理由

- ・ 有償である
- ・ .NET版の同じライブラリを購入しても、容易に.NETコンバートができないため、多くの手修正作業が必要になる。
- ・ .NET標準コントロールで対応不可能な機能は少ない。

Spread

VSFlexGrid

InputMan

ImText、ImNumber、ImDate

QuickPak

CSCmd、CSCombo、CSOptionList...

SSPanel、SSTab、SSFrame

3-(4) VBコンバートツール -2



◆多くのサードパーティ製コントロールのコンバートが可能である。

VB6互換コントロール

Spread

- SpreadGrid.vb
- SpreadGridArray.vb
- SpreadGridDate.vb
- SpreadGridNumber.vb
- SpreadGridTextBox.vb

InputMan

- imText.vb
- imTextArray.vb
- imTextBase.vb
- imDate.vb
- imDateArray.vb
- imMask.vb
- imMaskArray.vb
- imMaskBase.vb
- imNumber.vb
- imNumberArray.vb

QuickPak

- CSCmdBtn.vb
- CSCmdBtnArray.vb
- CSComboBox.vb
- CSComboBoxArray.vb
- CSOptList.vb

VSFlexGrid

- VSFlexGrid

SHERIDAN3D

- SSPanel.vb
- SSTab.vb

FormDesigner

- xDate.vb
- xDateArray.vb
- xLabel.vb
- xKana.vb
- xNum.vb
- xNumArray.vb
- 他
- ToolStripStatusDate.vb
- ToolStripStatusTime.vb
- CryRepViewer.vb
- CrystalReport.vb

◆ 独自開発したコンバートツール

- 弊社のコンバートツールライブラリを使用することで、.NET標準アップグレードで変換できないソースコードのコンバートに対応する。
 - ・ コントロールの置き換え
 - サードパーティ製コントロールを弊社VB6互換コントロールに置き換える。
 - VB6で作成されたユーザコントロールがある場合、.NETコンバート後のユーザコントロールに置き換える。
 - ・ その他コーディングの置き換え
 - .NETの標準アップグレードツールでは正しく置き換えられない非互換コードが多くあるため、独自ツールを開発して機械的な置き換えができるように対応。
 - プロジェクト固有の非互換コードもツールを作成し対応する。

3-(4) VBコンバートツール -4



◆コンバートを可能にするツールライブラリ群

コンバートツール

Control		Lib	Source	Tool
•crystal.rb •cscmdbtn.rb •cscombobox.rb •csoptlist.rb •fpspread.rb •imagelist.rb •imcontainer.rb •imdate.rb •imdateusr.rb •immask.rb •imnumber.rb •imtext.rb •imtime.rb •listview.rb •menu.rb •msflex.rb •progress.rb •sscommand.rb •sspanel.rb	• statusbar.rb • tabstrip.rb • toolbar.rb • UsrCombo.rb • vbcontrols.rb • viewport.rb • vselastic.rb • vselasticarray.rb • vsflex.rb • vsindextab.rb • xdate.rb • xkana.rb • xlabel.rb • xnum.rb • xpass.rb	•control.rb •ctl_convert.rb •font.rb •frx.rb •prj_convert.rb •prj_refactor.rb •recordset.rb •src_convert.rb •src_warning.rb •syscolor.rb •utils.rb	•cellalignment.rb •cmdupdtype.rb •comment.rb •connector.rb •crystalprop.rb •domaininstance.rb •fields.rb •fixstring.rb •formclosed.rb •gosub.rb •im_enum.rb •me_nothing.rb •misc.rb •msflex_enum.rb •printer_comment.rb •private_structure.rb •spreadevent.rb •ss_event_args.rb •vsflex_enum.rb •vstabmethod.rb	•create_warn_script.rb •extractFrx.rb •get_control_info.rb •get_form_closed.rb •get_frx.rb •get_image_list.rb •get_recordset_info.rb •get_used_control.rb

4. VBコンバート留意事項



- (1) 画面の操作性
- (2) レスポンス

4-(1) 画面の操作性

◆VBコンバート後は.NET標準コントロールに準じた操作性に変更される。

- ・ サードパーティ製のコントロールを使用している場合の操作性は異なる場合がある
- ・ VB6と.NETで仕様が変わったコントロールでは動作が異なる

－ 動作や操作が異なる事例

パネル等でコントロールをグループ化している場合は、使用不可のグレー色がグループ全体に反映される

「TAB」によるフォーカス移動が、グループのタブ順序→コントロールのタブ順序の順になる(調整が必要)

一覧表示を複数行選択する際に「CTR」を押しながらの選択になる

一覧表示の一覧内「TAB」移動が異なる動作になる

4-(2) レスポンス

◆VBコンバートではMicrosoft製、弊社製のVB6互換コントロールを使用する。

- VB6の命令コードで互換処理を介し.NET本来の命令を実行する。

そのため、.NET本来の処理より遅くなることがある

運用上支障がある場合などは、調整・対応が必要

- 既知の問題

Microsoft製印刷用VB6互換ライブラリを使用するとレスポンスが悪化する場合がある

運用に耐えられないケースは、CrystalReportなどのレポートツールを使用するなど対策が必要

Microsoft及び弊社製のVB6互換コントロールを使用するとレスポンスが悪化する場合がある

運用に耐えられないケースはチューニングや.NET標準の機能を使用するなど対策が必要

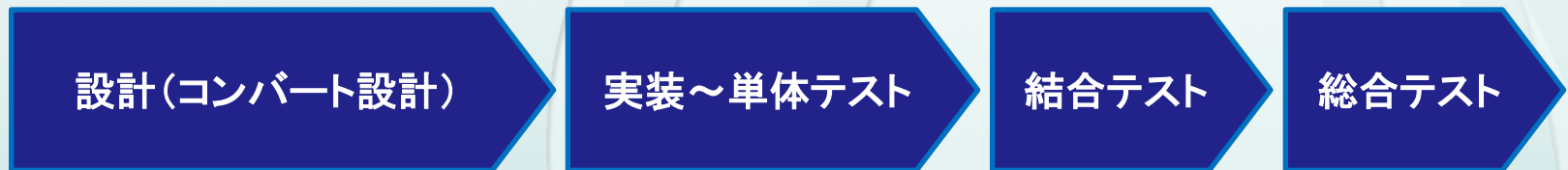
5. VBコンバート開発

- (1) 開発工程
- (2) 工程毎の作業内容

5-(1) 開発工程

◆VBコンバートの開発工程

<VBコンバートの開発工程>



<スクラッチ開発の開発工程>



※ポイント

作業工程の流れに大きな違いはない。

スクラッチ開発では「何をどう作るのか」という点に着眼して開発するが、

コンバート開発では「何処をどう変換するのか」に着眼するところが大きな違いである。

5-(2) 工程毎の作業内容

＜コンバート設計工程＞ 目的：コンバート仕様の決定

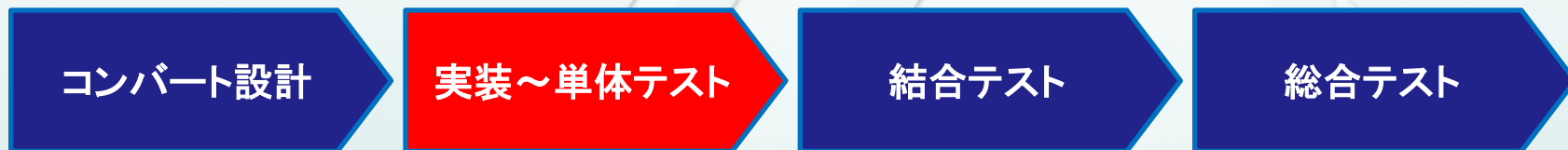


※作業内容

- ・コンバート設計では「何のコントロールをどう変換するのか」を取り決める。
- ・コンバート対象のコントロールの洗い出し
プログラムソースからコンバート対象のコントロールを洗い出す。
- ・コンバート仕様の決定
コントロールを置き換えることにより画面の操作性が変わる場合があるため
コンバート対象のコントロール毎に操作性を考慮したコンバート仕様を決定する。
操作性の違いが問題になる場合は、.NETのコントロール仕様の範囲で改善を検討。
- ・単体テスト計画
単体テスト内容、方法について決定する。
※単体テストから総合テストのテスト方法、役割については商談フェーズで取り決めておく

5-(2) 工程毎の作業内容

＜実装～単体テスト工程＞ 目的：コンバート、単体品質確認



※作業内容

- ・UIS主体の作業。お客様にはテストを円滑に進めるためのサポートをお願いします。
- ・コンバートツール開発
コンバート設計で決定したコンバート仕様に基づきコンバートツールを変更する。
ツールが無く、作業の効率化が期待できる場合は、新たにコンバートツールを開発。
- ・コンバート～コンパイル
コンバートツールと手修正により.NETのソースにコンバートする。
- ・単体テスト
商談フェーズで取り決めたテスト方法、役割の範囲で実施する。

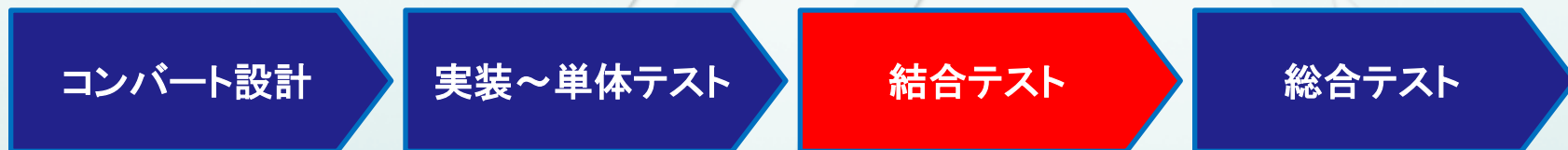
※ポイント

テストを合理的に行うため以下の貸与をお願いしています

- － 比較検証を行うためのテストデータの貸与
- － 開発時の単体テストケースの貸与

5-(2) 工程毎の作業内容

＜結合テスト工程＞ 目的：システム機能確認（開発側主体）



※作業内容

- ・UIS主体の作業。お客様にはテストを円滑に進めるためのサポートをお願いします。
- ・結合テスト
商談フェーズで取り決めたテスト方法、役割の範囲で実施する。
結合テストケース、テスト方法について前工程で決定しておく。

※ポイント

- テストを合理的に行うため以下の貸与をお願いしています
- － 比較検証を行うためのテストデータの貸与
 - － 開発時の結合テストケースの貸与

5-(2) 工程毎の作業内容

＜総合テスト工程＞ 目的：システム機能確認（お客様主体）



※作業内容

- ・お客様主体の作業です。UISは総合テストを支援させていただきます。
- ・総合テスト
商談フェーズで取り決めた役割、範囲でお客様のサポートを実施。
総合テストでは、以下のテストを実施。
 - － 単体機能確認
 - － システム機能確認

※ポイント

テストを効率的に行う方法として、

- ・機能単位で新旧システムの比較検証（日次や月次、年次等）
- ・並行本番

など、既存システムとの比較検証をご提案させていただきます。

5-(2) 工程毎の作業内容

<工程別の作業内容～総括>



工程	目的	主体	お客様作業	UIS作業
コンバート設計	・コンバート仕様の決定	お客様 U I S	・プログラムソース提供 ・画面コントロール操作の実態調査 ・コントロール変更方法の意思決定	・コントロールの洗い出し ・コントロール変換仕様検討 （画面表示や画面操作の変更提案）
実装～単体テスト	・コンバート作業 ・単体品質確認	U I S	・コンバート作業の問合わせ対応 プログラム仕様 単体テスト方法	・VBソースコンバート～コンパイル ・変更したコントロール等の動作テスト ・ビジネスロジックのテスト（*注意）
結合テスト	・システム品質確認	U I S	・結合テスト支援	・結合テスト（*注意）
総合テスト （受入テスト）	・システム品質確認	お客様	・単体機能確認 ・システム機能確認	・お客様テスト支援（*注意） ・バク対応

*注意： テスト内容、テスト方法、役割等について事前に取り決めます。

6. コンバートのメリット・デメリット

- (1) コンバートのメリット
- (2) コンバートのデメリット

6-(1) コンバートのメリット

これまでの導入経験から挙げられるメリット。

◆開発コストを抑制できる

- － 基本設計・詳細設計の作業量軽減
 - ・ 既存システムのドキュメントがある場合、仕様に関するドキュメントを作成する作業が不要になる。
- － ビジネスロジックの作りこみが少ない
 - ・ 通常の開発に比べ単純コンバートの場合は、仕様を理解する時間が不要になり、作業時間を大幅に削減できる。

◆新規開発よりも品質が高い

- － 既存ロジックを変更しない部分が多いことから、品質が確保される。
- － コンバートは比較的単純な作業が多く、バグを作りこむ可能性が低い。

これまでの導入経験から挙げられるメリット。

◆機能や品質面の向上は期待できない

- そのままコンバートするため機能や品質は向上されない。潜在バグもそのまま移行されてしまう。
- 機能面での投資対効果は得られない。

◆処理速度が遅くなるケースがある

- VB6のロジックをそのまま利用できるように、Microsoft社提供のライブラリおよび弊社で用意したライブラリを使用します。それらライブラリではVB6との互換性を保つための処理が組み込まれているため処理速度が低下します。場合によっては対策が必要になります。

◆ソースコード可読性の低下

- コンバートで作られるコメント行により、ソースコードの可読性が低下してしまう。
 - ・ 既にあったコメント行はそのまま残されるため、新規作成した場合と比較して見づらいプログラムソースになってしまう。
 - ・ アップグレード機能では、コンバート部分にコメント行が挿入される。(消すことは可能)

7. VBコンバート導入事例

- (1) 導入事例Ⅰ
- (2) 導入事例Ⅱ
- (3) 導入事例Ⅲ

7-(1) 導入事例 I -1



業種	製造業
システム	経理システム（メタフレーム）
開発 経緯・目的	• サーバの老朽化に伴うサーバリプレイス • 新サーバへのアプリケーション移行 • VB6資産はそのまま使用したい
VBコンバート対象	• 機能数：87（出力帳票数：30） • VB6→ .NET Framework4 コンバート
VBコンバート期間	2011年05月～08月

◆ハード・ソフトウェア概要

項目		旧システム	新システム
アプリケーション サーバ	OS	Windows Server 2000	Windows Server 2008R2 (x64) : 2サーバ
	ターミナルサービス	Citrix MetaFrame	Citrix XenApp
	アプリケーション	VB6.0	.NET Framework4
データベース サーバ	OS	Windows2000サーバ	Windows Server 2008R2 (x64)
	データベース	Oracle9g	Oracle11gR2
開発ツール		Visual Basic 6.0 Spread6.0 InputMan6.5	VisualStudio2010 SAP Crystal Reports

◆VBコンバートを選択した理由

- VB6そのままの利用を考えていたが、下記がコンバートをするきっかけとなった。
 - ・ 稼働サーバのメーカーサポート終了
 - ・ VB6開発環境サポートの終了
 - ・ 保守作業(開発環境)の維持が困難
- 機能変更が少なく、VB6の機能の焼き直しのみであった。
- 新環境への移行を短期間に済ませたかった。

※初期段階においては、新規開発やVM環境を利用した開発環境の維持など他の案も検討にあがった。

◆本開発の特徴

－ テスト作業

- ・ 弊社開発のVB6システムではないことから、仕様のヒアリングが必要であった（設計資料がメンテナンスされていないなどの問題もあった）。
- ・ 仕様説明の期間・コストを削減するために、テストをお客様が担当されることになった。
- ・ 弊社の担当作業
 - － VBコンバート、及びVB6と.NETシステムの動作比較
 - ≫ VBコンバートした箇所の検証
 - ≫ 業務から見た標準的な入力操作の範囲で新旧システムの動作を比較する方法でテストを行った。
 - － お客様テストで発生したバグ修正
- ・ お客様の担当作業
 - － テストは、テスト全体を通してシステム仕様を理解しているお客様に主体的に行っていただいた。

– VB6ソース管理の問題

- ・ コンバートを行ったプログラムのうち、5機能分が、最新版のプログラムソースではなかった。
 - 弊社が実施した新旧比較テストで発覚
- ・ 最新版がどれかわからず、仕様を確認し手修正を行った。

– レスpons

- ・ Microsoft製印刷用VB6互換用ライブラリを使用してVBコンバートを行ったが、レスポンスの悪化が発生した。
- ・ 利用頻度が高く運用上支障をきたす機能について、レスポンス改善を行った。

◆効果

- 新規開発との比較
 - ・ スクラッチ開発:VBコンバート=2:1 (スクラッチ分は弊社試算)
- 納期
 - ・ 決算前に新システムへ切り替える。
 - ・ 帳票機能レスポンス対応やVB6ソースの最新版がないという問題は起きたが、開発体制を調整し納期までに完了した。
- 品質
 - ・ お客様のテストで見つかった不具合は15件程度で、品質は良好であった。

◆課題

- VB6に比べ処理の遅い帳票機能が残っている。
 - ・ 印刷用VB6互換ライブラリを使用することによるレスポンス対応を行っていない帳票
 - ・ 運用への影響が少ないとの判断で改善対応を一旦見送り。
 - ・ 今後、段階的に改善を行っていく予定。(クリスタルレポート化)

7-(2) 導入事例Ⅱ -1



業種	仕入販売業
システム	販売管理システム
開発 経緯・目的	・ サーバの老朽化に伴うサーバリプレイス ・ 新サーバへのアプリケーション移行 ・ 新機能追加 ・ WEB化などの実現方法の検討
VBコンバート対象	・ 機能数:25（出力帳票数:24） ・ VB5→.NET Framework2.0 コンバート
VBコンバート期間	2008年06月～07月

◆ハード・ソフトウェア概要

項目	旧システム	新システム
データベースサーバ	OS:Windows NT4.0 DB:Oracle8	OS:Windows 2000 Server DB:Oracle10g
クライアント	OS:Windows NT4.0 APP:VB 5.0	OS:Windows XP、Windows Vista APP:.Net Framework2.0
開発ツール	Visual Basic 5.0 • InputMan 5.0J • SPREAD 3.0J • QuickPak 4.0J 帳票 • Crystal Report 6.0 • Excel 2000 • Access 2000	Visual Studio 2005 帳票 • Crystal Report XI • Excel 2003

◆VBコンバートを選択した理由

- 新システムの要件
 - ・ 新たな機能の追加
 - ・ 既存機能カスタマイズ
- VBコンバートの理由
 - ・ 複数の案を比較、検討。コストが重要視された。
 - ・ 下記はパターンの例(②が今回の開発内容)。
 - ①既存機能のWEB化
 - ②コンバート+コンバート後のカスタマイズ+新規機能追加
 - ・ 予算内でカスタマイズ・新規機能追加を行えるVBコンバートを選択した。

◆本開発の特徴

- カスタマイズ＋新規機能の追加
 - ・ VBコンバート後に、カスタマイズ／新機能追加を行った。
- 弊社開発のシステムである
 - ・ 弊社が開発したシステムであることから仕様を理解しており、開発側のテストは新旧比較しながらの仕様に準じたテストを行った。
- レスポンス等に問題はなかった
 - ・ 理由としては…
 - ハードウェアのスペック向上
 - もともと高負荷の処理が少ない。
 - VB6互換処理が行われるライブラリ、コントロールの使用箇所が少なかった。

◆効果

－ 機能の充実

- ・ 予算の範囲で、システム移行＋カスタマイズ＋新規機能追加が可能となった。
- ・ 既存機能のWEB化と本開発の作業量比較
 - － 既存機能のWEB化：本開発＝1. 2：1

－ 高品質である

◆課題

－ 特になし

7-(3) 導入事例Ⅲ -1

業種	出版業
システム	商品受注システム
開発 経緯・目的	・ サーバリプレイス ・ ホストのリプレイス ・ サーバのアプリケーション移行 ・ VB6サポート切れによるクアイアントVB6アプリケーションの移行 ・ 商品変更によるリニューアル (カスタマイズ／新機能追加)
VBコンバート対象	・ クライント(VB)⇔サーバ(COBOL) ・ 受注システム 機能数:133 ・ (弊社担当分、帳票機能はサーバ側) ・ VB6→ .NET Framework2.0 コンバート
VBコンバート期間	2008年01月、04月～08月

◆ハード・ソフトウェア概要

項目	旧システム	新システム
データベースサーバ兼 アプリケーションサーバ	OS、DB:富士通ホスト APP:COBOL	OS:IBM I5 DB:DB2 APP:COBOL
クライアント	OS:Windows 2000、XP APP:VB 6.0	OS:Windows 2000、XP APP:.NET Framework2.0
開発ツール	Visual Basic 5.0 • InputMan 5.0J • SPREAD 3.0J • QuickPak 4.0J ほか	Visual Studio 2005

◆VBコンバートを選択した理由

- 新システムの要件
 - ・ 新たな商品に対応した機能追加
 - ・ VB6システムの移行
- VBコンバートの理由
 - ・ 開発範囲が大きいことから、コスト削減・開発期間短縮を行うため。
 - クライアントのVBシステムでは商品の種類や業務により複数のサブシステムに分かれていてシステム規模が大きい。
 - ≫ 2社体制でVBコンバートを実施
 - サーバリプレイスによるサーバアプリケーションの移行も必要になる。

◆本開発の特徴

- カスタマイズ＋新規機能の追加
 - ・ VBコンバート後に、カスタマイズ／新機能追加を行った。
 - ・ 開発中にエンドユーザから仕様の変更や追加要望によりVBコンバート以外の部分で調整が多発。
- レスポンス
 - ・ レスポンス問題が少し発生したが、チューニングにより回避。
 - ・ 運用上問題がない遅さに対応していない。
 - クライアント(VB)⇔サーバ(COBOL)のシステムであるため旧来から体感としては少し遅い。
- 2社でのVBコンバート
 - ・ 2社間でVBコンバート様式を検討し決定した。
 - ・ 弊社ではコンバートツールに決定内容を反映して対応した。

◆効果

－ 期間の短縮

- ・ VBコンバートを行うことによりサーバ(COBOL)側の移行スケジュールに合わせることが可能になった。
 - － 追加の変更要望も多かったことから、新規開発を行っていた場合は、仕様が定まらずスケジュールの延長になっていたかもしれない。

－ 高品質である

◆課題

－ 開発中の変更要望追加による影響

- ・ 品質確保のためVBコンバート・動作確認後にカスタマイズを行う。開発中の追加変更が多いとスケジュール内での調整が困難になる。
 - － スクラッチ開発では規模が大きくなり、期間が長く、調整の余地がある。
 - － 実際数件の要望は本番稼働後に別途対応することになった。
 - ≫ 動作はするので、本番稼働後の調整にすることはできる。

- Microsoft製品のWeb情報
 - ・ Visual Studio サポートについて
 - <http://www.microsoft.com/japan/msdn/vstudio/support/>
 - ・ Visual Basic 6.0 ファミリ製品のライフ サイクル ガイドライン
 - <http://msdn.microsoft.com/ja-jp/vbasic/cc707266.aspx>
 - ・ マイクロソフト プロダクト サポート ライフサイクル
 - <http://support.microsoft.com/lifecycle/?LN=ja&c1=501>
 - ・ Windows Vista®、Windows Server 2008®、および Windows 7 におけるVisual Basic 6.0 のサポートについて
 - <http://msdn.microsoft.com/ja-jp/vbasic/cc707268>
 - [http://msdn.microsoft.com/ja-jp/vbrun/ms788708\(en-us\).aspx](http://msdn.microsoft.com/ja-jp/vbrun/ms788708(en-us).aspx)
 - ・ Visual Studio 開発ツール対応 OS 一覧
 - <http://www.microsoft.com/japan/msdn/vstudio/support/tools.aspx>
- グレープシティ製品のWeb情報(Spread、InputManなど)
 - ・ .NET移行情報
 - <http://www.grapecity.com/tools/support/migration/>