

## 【G1】オープニングセッション 「CodeGearテクノロジーマップ」

藤井 等

CodeGear マーケティングディレクター

<http://www.codegear.com/jp>



### こんなことはありませんか？

- これまでの慣例や延長で、選択している技術が、今回の開発に適当なのかどうか分からない
- いままでと違った種類の開発を行うのに、どの技術を選択すべきか決めかねる
- 新しい技術の話題を聞くが、それを採用すべきなのか気になる

## このセッションの目標

CodeGear製品のテクノロジーマップを整理すること

- 現在の製品技術の適用先を見極める
- 今後のCodeGear製品の方向性を理解する



## CodeGearの現行開発ツールのマップ



|    | Compiled             | Managed              | Dynamic           |
|----|----------------------|----------------------|-------------------|
| 長所 | デスクトップ/GUIアプリケーション   | エンタープライズWeb/SOA      | エンタープライズWeb/SOA   |
|    | クライアント/サーバー          | OS/ハードウェア可搬性         | OS/ハードウェア可搬性      |
|    | パッケージソフトウェア          | エンタープライズインテグレーション    | エンタープライズインテグレーション |
|    | パフォーマンスにシビアなアプリケーション | モバイル                 |                   |
| 短所 | ハードウェア・周辺機器の制御       | クライアント/サーバー          | ユーティリティ           |
|    | SMEアプリケーション          | ポータル                 | 企業の部門アプリケーション     |
|    | モバイル                 | アプリケーションオートメーション     | ポータル              |
|    | Webアプリケーション          | デスクトップGUIアプリケーション    | モバイル              |
| 短所 | クライアントのOS/ハードウェア可搬性  | ハードウェア・周辺機器の制御       | デスクトップC/Sアプリケーション |
|    | 大規模エンタープライズアプリケーション  | パッケージソフトウェア          | 実行パフォーマンス         |
|    | ポータル                 | パフォーマンスにシビアなアプリケーション | ハードウェア・周辺機器の制御    |
|    |                      |                      |                   |

## Web開発に適した3つのテクノロジーの違い

### Java

- ・エンタープライズフレームワーク
- ・規模の拡大に対応できる
- ・ただし複雑性を内包

### PHP

- ・画面セントリック
- ・データベースを利用したWebアプリケーションを簡単に開発

### Ruby on Rails

- ・まずはすばやく開発(アジャイル型)
- ・自動的に生成される成果物
- ・ゆるさが重要

### ツールに求められるもの



## JBuilderが目指しているもの



### 複雑性の解消

- ツールの複雑性
  - シナリオ化されたツールの活用
- フレームワークの複雑性
  - XMLを書くことがJava開発じゃない！

### JBuilderの開発方針

- オープンソースを企業システム開発に活かす
  - そのためのツールインフラ
  - オープンソースフレームワークとの協調
- フレームワーク中心の開発からアプリケーション中心へ
  - Application Factories

## Delphi for PHPが目指しているもの



### ビジュアルデザイン

- コンポーネント指向の開発
  - VCL for PHP
  - イベントドリブン開発
  - データベースアクセス
  - カスタムコンポーネント

### Delphi for PHPの開発方針

- オープンソースとの協調
  - さまざまなオープンソースフレームワーク上に製品を構築
  - コンポーネントフレームワークはオープンソースに
- DelphiライクをオープンWeb開発へ
  - Delphiの長所をPHPの世界へ、PHPらしく実装

## 3rdRailが目指しているもの



### コントロール

- Ruby on Rails開発をエンタープライズへ
  - 依存性ビューで複雑化するアプリケーションを制御
  - ツールによる標準化

### 3rdRailの開発方針

- Ruby / Ruby on Railsの長所を活かす
  - 独自のフレームワークを構築するのではなく、ツールはRuby / Ruby on Rails開発を支援する
  - 従来のコマンドライン開発を活かしながら、ツールのメリットを享受できるようにする
- 企業でのRuby / Ruby on Rails採用をにらんだ製品
  - 開発の標準化が可能なように、ヤングプログラマからエキスパートまでをカバーする機能セットを提供

## 選択の指針



## Win32ネイティブからのWebアプローチ

### 課題

- 現在の技術の延長線上で開発すべきか
- 新しい技術を取り入れるべきか

### 解決の指針

- 既存開発資産
- スキル
- 規模・トラフィック・トランザクション
- 目的・用途・ライフサイクル
- データベース・バックエンドシステム
- 納期・開発プロセス



## Delphi / C++BuilderからWebへ

### 既存開発資産を活かす場合...

- VCL for the Web
- ASP.NET
- Java + ネイティブ

### 既存開発資産は気にしない場合...

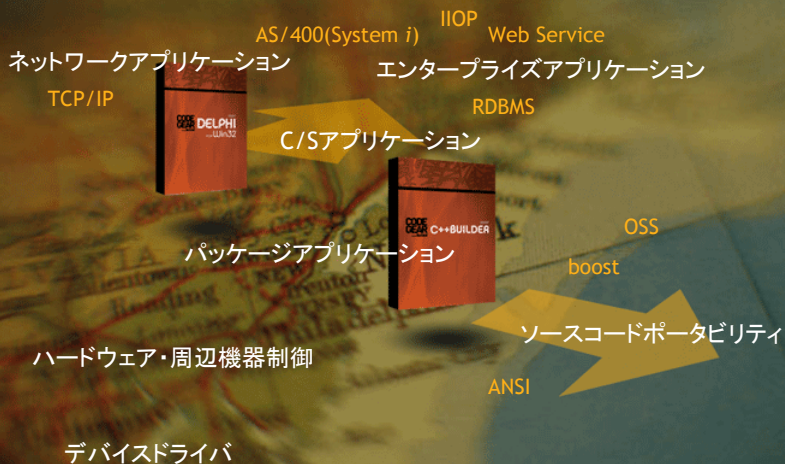
- Delphi for PHP
  - 既存の開発スタイルは継承できる
  - 開発資産は別モノ (Delphi言語コンバータではない！)
- その他
  - Java、Rubyなど (選択の指針は通常と同じ)

## 既存開発資産を抱えているときの注意点

実際問題、Webアプリは別モノです

- GUIを再利用できるという幻想は捨てるべき
  - 同じ操作性が必要ならより新しいテクノロジーを検討すべきかもしれません
  - アプリケーションの通信粒度が違います
    - 細かい制御が必要ならAJAXも要検討
- ロジックの再利用が可能かは設計に大きく依存
  - 何を再利用したいのかを明確にすること
  - 再利用したいものは、分離性が高くなくてはならない

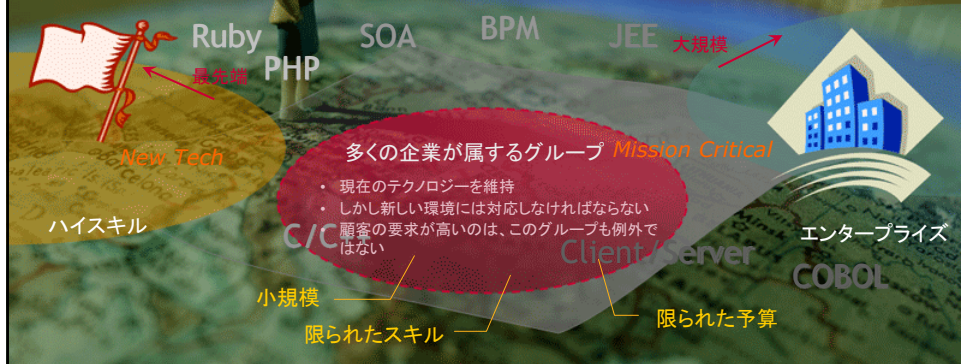
## ネイティブアプリケーションマップ



## ソフトウェア開発のランドスケープ

最先端はごく一部...

- しかし、多くの開発者は「最新」への対応にも追われている



## 多くの企業が求める開発スタイル

簡単、高性能、しかし拡張性あり

- 限られた予算、人員、スキルでの開発
- しかし高いユーザー要求に応えるきめ細かさ

新しい環境への対応（あまり労力をかけず）

- Windows Vistaへの対応
- データベースのリプレース、バージョンアップ
- Webテクノロジーの利用

本来の業務への集中

- 業務アプリケーションを作ることが業務
- 本質的ではないUIのコーディングに時間を取られるのはナンセンス



## 企業の開発の「裏」トレンド

### ネイティブアプリケーションへのゆるぎない需要

- マネージドコード (Java / .NET) が主流になると言われながらも...
- パフォーマンス、ハードウェアの制御、グラフィック処理などさまざまな理由によりネイティブアプリケーションの需要は衰えない

### リプレースプロジェクトの増加

- 過去の開発資産が、OSのサポート切れ、データベースのリプレースなどによって、更新を余儀なくされている
- 既存資産を活かして最小のコストで移行したいという要望
- 100%新規ではなく、まるごと.NETやJavaとはいかない現状

## Delphi / C++Builderの将来

### 最強のWindowsネイティブ環境の追求

- Win32 / Win64ネイティブ
- ユニコードサポートをはじめとするVCLのエンハンス

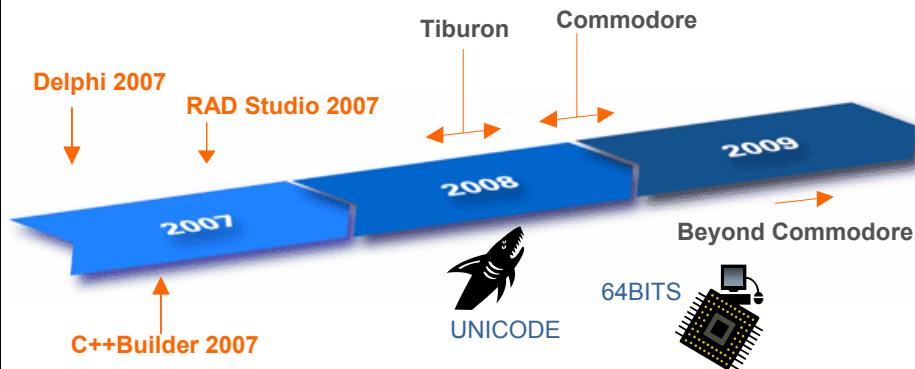
### クロスコンパイルソリューションへの展開

- Win64が最初のターゲット
- Delphiテクノロジーの応用・拡大

### C/C++の技術キャッチアップ

- 標準化に大きく関与

## Delphi / C++Builderロードマップ



この情報は、現時点でのCodeGearの製品の方向性の概略を説明したもので、この情報に依存した製品購入の決定をされないようにご注意ください。将来の開発、製品機能や特徴のリリースやそのタイミングについては、予告なく変更される場合があります。

## ここでビデオレターをご覧ください...



新井正広

CodeGear

ローカライゼーションマネージャ

## お知らせ

フィールドテストプログラムの詳細については、今後、CDN (CodeGear Developer Network) および、CodeGear ブログに注目していただきます。

CDN 日本語版

<http://dn.codegear.com/jp>

CodeGear Team Japan ブログ

<http://blogs.codegear.com/teamj>