

Deployment First のススメ

とっちゃん(高萩 俊行)

Microsoft MVP for Development Tools – Visual C++

Since 2005/10-

<http://blogs.wankuma.com/tocchann/>



わんくま同盟 名古屋勉強会 #5

アジェンダ

- ~~自己紹介(印刷資料からは除外)~~
- インストーラとは何か？
- 製品ライフサイクルとのかかわり
- 更新と後始末
- まとめ

インストーラとは何か？

- インストーラに与えられた役割とは何か
 - 製品を起動するために必要なデータを適切な場所に配置する
 - インストールする
 - 自分が配置した製品の更新を行う
 - アップデートする
 - 自分が配置した製品の削除を行う
 - アンインストールする
- これら3つの役割が一つにまとまってはじめてインストーラと呼べるようになる

インストーラとは何か？

- インストーラが必要なときとは？
 - 開発マシン以外の環境で動作させる必要があるとき
- 開発マシン以外とは？
 - ビルドを行わないマシンのこと

製品ライフサイクルを回す

• 製品ライフサイクルの各要素

- 動機
- 設計と実装
- 配布
- 配置
- 運用
- フィードバック
- 修正
- 更新
- 終了

製品ライフサイクルを回す

- あとで...ここに図を書いてみましょう



製品ライフサイクルを回す

- **動機**

- 製品がほしい(提供したい)という思い
- これがなければ、商品は成り立たない

- **設計と実装**

- 製品として具現化していく作業
- インストーラもアプリケーション
 - 製品内のアプリの一つである

製品ライフサイクルを回す

- インストーラの関与する設計とは？
 - 製品への要求にあったテクノロジーの選択
 - Windows Installer
 - ClickOnce
 - その他
 - 配置からみた製品構成の検討
 - 配置構成に合わせたプロジェクトの構造
 - インストーラへの反映

製品ライフサイクルを回す

- インストーラ設計での注意点
 - 一番最初にユーザーが利用するアプリ
 - どのような形であれシステムを変更する
 - インストーラに失敗は許されない
 - 変化を前提とした設計を行う
- 実行情報の設定はインストーラの仕事ではない！！！！
- 最後に動かすのもインストーラである！

製品ライフサイクルを回す

- バージョンとは...
 - インストーラのバージョン管理は
 - ファイルバージョンが対象！
 - アセンブリバージョンではない！
 - バージョンがより大きいものが更新対象
- バージョン管理は、製品管理の基礎
 - 製品を管理する上で必ず必要
 - 開発マシンの外に出た時点で管理対象
 - たとえテストビルドであっても

製品ライフサイクルを回す

- 配布

- どのようにユーザーの手元に届けるのか？
- その手段はひとつなのか？複数なのか？
- 配置を考えて配布を決めねばならない

- 配置

- どうやってユーザーに実行させるか？

- 配布と配置は運用を考慮して
決めなければならない！！！！

製品ライフサイクルを回す

- 修正
 - プログラムの修正に合わせてインストーラも修正
 - バグの修正だけとは限らない
 - 状況によっては大きなバージョンアップもある
- 更新
 - すでに製品のある場所での配置を更新と呼ぶ
 - 更新とは「配布と配置」の特殊系

更新と後始末

- 更新する際に注意する部分
 - 絶対にバージョンダウンをしない
 - 同じものは更新しない
 - 更新対象はファイルだけとは限らない

更新と後始末

- 後始末とは
 - 「配置したもの」を撤去する作業
 - 単に消すではない次のための布石である

まとめ

- インストーラとは何か？
 - インストール、アップデート、アンインストールをまとめたアプリ
- 製品ライフサイクルでの役割
 - 運用以外のすべてのシーケンスで関与

まとめ

- どのタイミングから作業するのか
 - 設計の初期段階から考察が必要
- 常に更新を考慮する
 - 常に先をみた変化を考慮する
- 終わりを見据える！
- 常に次を考えて作成する