伝授の巻について

Takahiro Yoshimura (@alterakey)

27.9.2012

自己紹介

- 株式会社モノリスワークス最高技術責任者
- 埼玉にある小さな企業
- Web開発やスマートフォン開発を主に
- Twitter → @alterakey

かかわり

- 伝授の巻の約8割を執筆させていただきました
- うっかり手を挙げたらいつのまにか…

これまでは

- 著者が初心者を想定して並べた「カリキュラム」
  - Activity、Intent、Service…
    - 「Activityを使用して画面を出します」
      「裏で処理をするにはServiceを使います」
      →情報サイトを著者の言葉で焼き直しただけ
  - お決まりのサンプルアプリ

で、…

- これで実用的なアプリが作れるのか？
- これで「あの」素晴らしい案を実現できるのか？
- 実際、案件はどうか？

現実は

- そんなアプリから派生したものが何になる？
  - 手元にある技術を元に企画するべきではない
    - 企画を先に立ててそれに沿って勉強すべき
      →そもそも技術は企画を実装する方法なはず
      →そのような手段から発想すること自体いびつ

現実は

- 現場には多種多様な制約がある
  - 適材適所で技術を使い分ける必要がある
    →道具はきちんと使い分けてこそ
    →端末間互換が取りにくくなっている一因
- 安易な御仕着せは害悪に他ならない

現実は

- 時が経つにつれてノウハウが陳腐化する
  - 常に最新の情報を持って対応すべき
    →最新バージョンから対応範囲を広げる
    →新しい端末で古臭いUI…残念そのもの
- 日本語の資料は二極化＋希薄
  →海外の資料が読めるかどうかが成長の鍵

現実は

- 読んでも「あの」アプリはまだ作れない…
- そんな本、本当に役に立ちますか？

では「伝授」とは？

- 「やりたいこと」への「指針」
- 考え方やアプローチであり「解答」ではない
  →唯一不変の「解答」など存在し得ないため
- 実際に考えるのはあくまでも「読者」
  →コードは全て理解を優先して抜粋

では「伝授」とは？

- 「知識」ではなく「知恵」を
  - 情報収集の知恵
  - 実装戦略を選ぶ知恵
  - 開発環境を使いこなす知恵

  →もっと速くもっと気持ち良く開発するために

例

- 情報収集（ありがち）

「Android開発における資料は公式サイトに豊富にありますので、開発の際には参考になるでしょう。」

→英語じゃんかよ！どうやって読むんだよ！

→公式サイトだけが情報ではない

例

- 情報収集（伝授）

検索キーワードの抽出方法＋ななめ読みの手法＋
サンプルコードの検索手段＋コミュニティの紹介

→本質的な「情報収集力」の強化
→ブログ類を含む幅広い情報網を活用できるようになり、コミュニティへ参加したくなる

例

- Fragment（ありがち）

「Android 3.0から追加されたFragmentを使うと、タブレット端末などの大画面を効果的に使うことができます。」

→Fragmentの役目はそれだけではない

→「Fragment＝タブレット」という誤解の一因

例

- Fragment（伝授）

「大画面対応にはFragmentも有効な手段です
よ」＋判断基準＋実装戦略、さらにUIとは無関係
な構造改善にも使えることを示す

→「Fragment＝タブレット」でないことが分か
り、積極的に利用してみたくなる

例

- バックグラウンド処理（ありがち）

「時間のかかる処理を行なうにはServiceを使います」

→ふーん、じゃこの処理は時間がかかりそうだからServiceを使うんだな…って、こんな簡単なことをするのにもか。Androidは面倒くさい…

例

- バックグラウンド処理（伝授）

「Serviceは時間のかかる処理を円滑に行なうための一つの手段です」＋他アプローチの提案＋判断基準

→Serviceを使わない方が良い場合もある

→実情に合った実装戦略を選べるようになる

例

- デバッグ（ありがち）

「Eclipseにはデバッガというものが備わっており、これを使用するとバグを簡単に特定・修正することができます」＋簡単なバグを仕込み解説

→…で？でも自分が今かかえている問題はそんな簡単じゃないんだけど？どうしてくれるの？

例

- デバッグ（伝授）

  Eclipse統合デバッガ以外も含めたデバッグ手法の紹介＋考え方＋判断基準

  →各手法の得意・不得意を知り、泥くさい方法その他の使い分けができるようになる

  →現場で重要なスキル

例

- メンテナンス（ありがち）

  「ここでXXXActivityクラスを次のように整理しておきましょう。こうすることにより後で…が簡単になります。」

  →どうして簡単になるの？ or 何が変わったの？

  →何が構造的に悪いのか伝わりにくい

例

- メンテナンス（伝授）

構造改善術（リファクタリング）の説明＋例示＋適用基準

→ありがちなまずいコードの雰囲気を感じ取り、対比することにより読みやすさを体感することができ、すぐにでも適用したくなる。

例

- 開発環境（ありがち）

「Eclipseはコード補完ができるエディタです。皆さんがカッコを入力した際にウィンドウが開きましたね。…詳しくはEclipseのヘルプを見て下さい」

→ふーん

例

- 開発環境（伝授）

  機能概覧＋使い方の指針＋カスタマイズの指針＋設定の指針

  →機能の使い方と共にどう使えば効果的なのかまで伝わり、実際に使ってみたくなる

まとめ

- 伝授の巻とは…
  - 「やりたいこと」に「指針」を提示
  - 「知識」ではなく「知恵」の集合
    情報収集、実装戦略、開発環境 etc.
    →（多分）陳腐化しない

- ご静聴ありがとうございました。