Introduction to code re-usability for Windows Phone

This article provides possible use cases when to consider for cross platform reusable code, WP8 & Windows 8 code sharing techniques and best practices for the reusable code

Introduction

One of the most attractive features of WP8 is its ability to share the code with the Windows 8 applications.
Especially it is more beneficial to Enterprise applications where your organization will want to make the same application available to your clients on various platforms like Desktop, mobile, tablets, etc. But this reuse is not just limited to Enterprise applications and can be leveraged by individual developers as well, who build same applications for both Windows and Windows phone platforms.

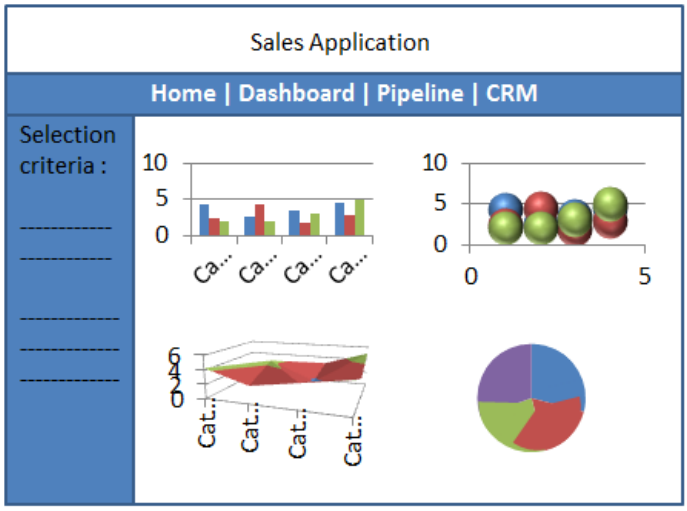
Benefits of Cross Platform Code Reuse

- Reduced Development Time : Code modification will propagate to all platforms, thus reduced development time
- Ease in code maintenance : Changes in code is now handled at one place alone and then sharing the same binaries.
- Excess code can lead to more defects : Reusable design will lead to less code and in turn less defects.
- Consistent Quality
- Improved 'Time to market'
- Reduced project cost

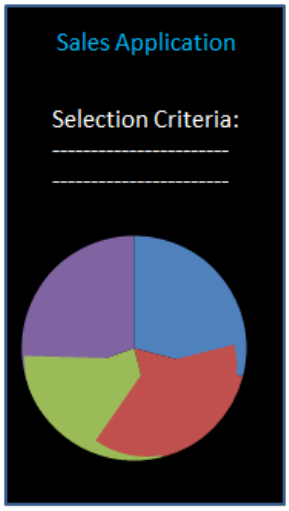
Where does cross platform code reuse makes sense

Enterprise Applications

Enterprise applications will want to make the same application available to their clients on various platforms like desktop, mobile, tablets, etc.
Example: A sales application will be made for desktop for an extensive use, but a mobile app with some crucial functionality only, will be handy for the sales guy when he is on the road. These critical functionalities can be coded in a way to take advantage of reusability.



Desktop Version



Mobile Version

Consumer Applications

Some software products have a use case in both desktop and mobile environments. Example: Skype, Facebook, etc.

Code Reuse has its 'Price'

Cross platform code reuse seems great but it comes with a price and that is the time spent in extensive analysis and design phase. So avoid going overboard when reuse doesn't make sense.
If you are sure that you application is going to be only used on WP8 devices, then obviously no sense in thinking about cross platform reusability.

(C) Copyright Nokia 2014. All rights reserved.

WP8 and Windows 8 code sharing techniques

With the availability of high memory and powerful processors Microsoft decided to align the Windows 8 and WP8.
The steps taken towards this alignment:

1. Replaced .NET Compact Framework in WP by CoreCLR which is the same .NET engine used on Windows 8
2. A large subset of Windows Runtime API is inbuilt in WP8
3. Native app support on WP8.

This three factors form the crux of the reusability feature.

Ways in which WP8 and Windows 8 can share the code

- Portable Class Libraries

Detailed information on MSDN DevCenter: Share functionality using Portable Class Libraries

- Share code with Add as Link

Detailed information on MSDN DevCenter: Share code with Add as Link

- Use the common Windows Runtime API

Detailed information on MSDN DevCenter: Share using Windows Runtime Components

- Conditional compilation

Detailed information on MSDN DevCenter: Conditional compilation with preprocessor directives

In addition, for a quick start here is a simple and good example given by Windows Phone SDK Team which covers all the above aspects and explains how each of the techniques is implemented in the example:
http://code.msdn.microsoft.com/wpapps/PixPresenter-Code-sharing-39ed631f

Cross Platform Code Reusability Best Practices

Now that we have seen the scenarios where code reuse makes sense and how can we achieve it in WP8 and Windows 8, let us have a look at some guidelines for the code that should be made reusable.

Keep the UI separate

.NET uses MVVM model i.e. view : view-model : model, where the view represents the UI layer.
WP8 and Windows have similar UI design guidelines for a consistent look and feel. But the way apps appear and interact with desktop and mobiles differ and hence it's good to avoid re-engineering too much on code reuse of this layer.

Reusing the application business logic

The best layer that can be part of reusable component is the business logic layer, which will remain the same irrespective of how it is presented on UI.

Reusing the data layer

This is a bit tricky, as in some cases this layer can be made reusable but in some cases it won't.
Example : Above mentioned Sales App, the database and services pulling data from it will be the same and can be a part of reusable components.
But consider an example of a photo editor application; here the data (image files) on which the business logic will act can be highly platform specific and hence in this case, this layer should not be a candidate for code reuse.

Reusable code must be loosely coupled and highly cohesive

Reusable code should be focused & have independent functionality. It should not be dependent on other modules. If the code is week in cohesion this will result in a code that is difficult to understand, maintain and reuse.

Summary

Cross platform code re-usability is a silent feature of WP8. If used for the appropriate use cases and on time, can prove to be much useful.

(C) Copyright Nokia 2014. All rights reserved.

(C) Copyright Nokia 2014. All rights reserved.