

Android for .NET Developers Series

Getting Started

Understanding Android Versioning

Jim Wilson

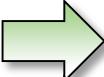
jimw@jwhh.com

@hedgehogjim

<http://facebook.com/hedgehogjim>



Outline

-  **Android version identity**
-  **Selecting Android platform and API level**
-  **Managing API Level**
-  **SDK Manager**

Android version identity

Each Android release has 2 separate identities

Platform version 1.6 2.1 4.2 4.2.2

- Collection of apps, features, and behaviors
- Identified by a decimal formatted value: $X.Y$ or $X.Y.Z$



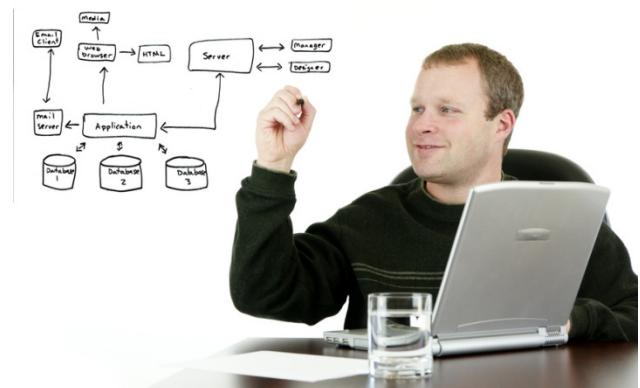
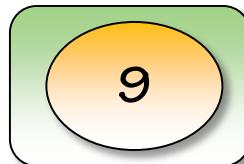
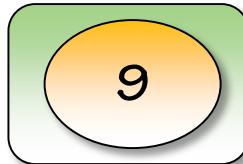
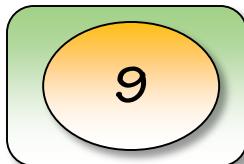
API level 1 2 15 16 17

- SDK features and capabilities exposed to application developers
- Identified by an integer

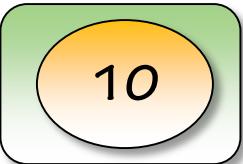
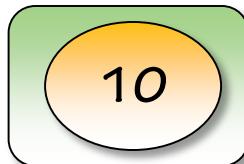
Platform version and API level relationship

- Each platform version supports a specific API level
- An API level may span multiple platform version

Android 2.3 Android 2.3.1 Android 2.3.2

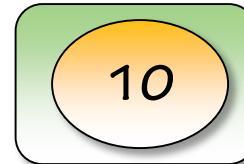


Android 2.3.3 Android 2.3.4

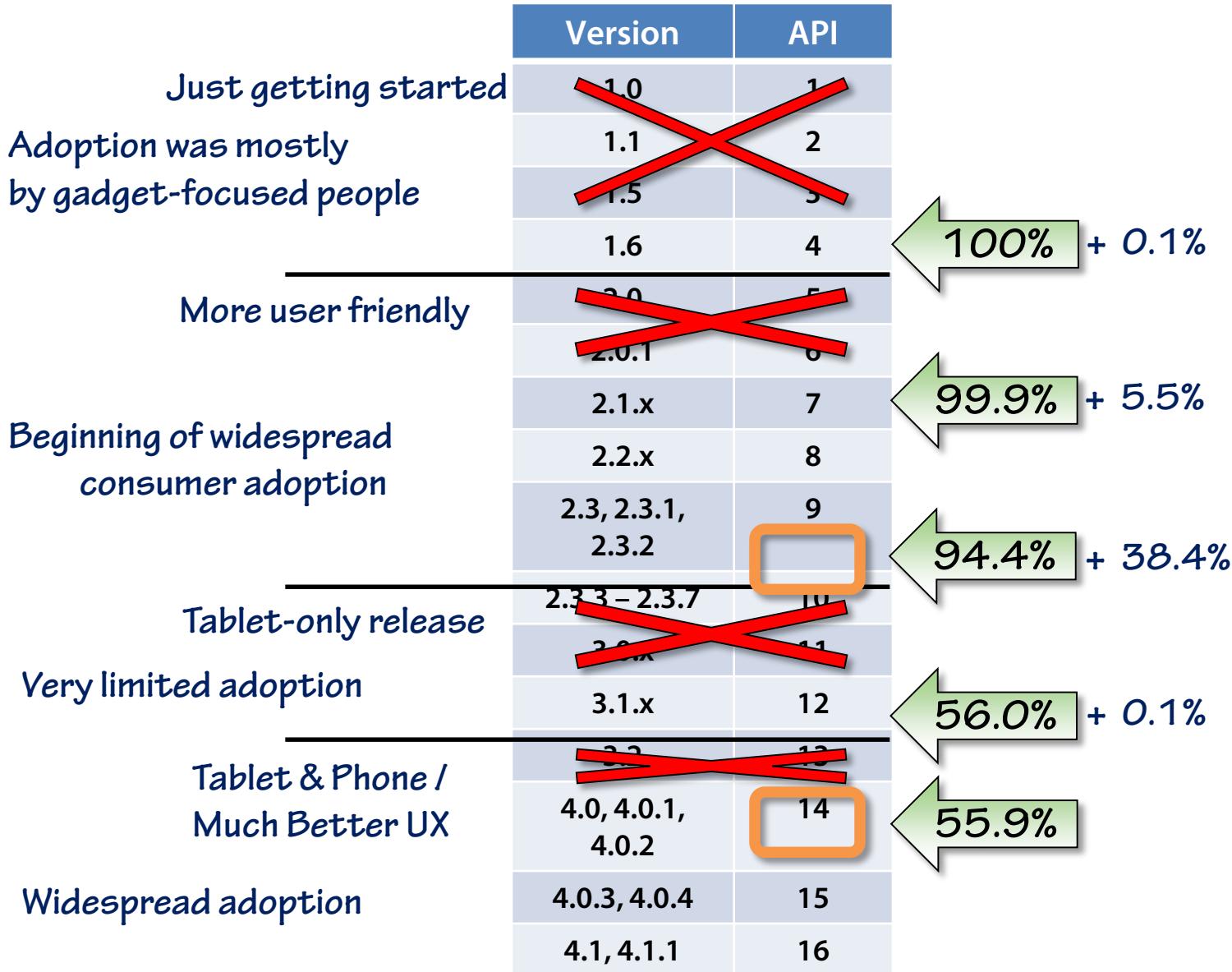


...
...

Android 2.3.7



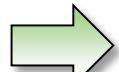
Selecting Android platform and API level



Devices in use by version

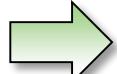
May 2013

Managing API Level

 **Android provides 2 ways to manage the API Level**

 **AndroidManifest.xml `uses-sdk` element**

- Identifies the minimum supported API Level
- Identifies the highest API Level you have tested your app with
- Can optionally identify the highest API Level you wish to support

 **Compiler**

- In most cases, compile app with the lowest API Level you want to support
- Use caution when compiling with API Level higher than lowest supported target
 - You must assure that you make no “unprotected” calls to higher level classes/methods

Android SDK Manager

→ **Android SDK Manager manages the SDK installation for developers**

→ Installs, updates, and removes features of your development environment

- Tools
- SDKs for each API Level
- Extras

→ You are responsible to check for updates

- Does not automatically keep your installation current



Android Studio



Managing API Level

- **Android provides 2 ways to manage the API Level**
 - **AndroidManifest.xml uses-sdk element**
 - Use attribute minSdkVersion
 - Identifies the lowest API Level you want to support
 - Google Play and device will prevent installing below this API Level
 - Use attribute targetSdkVersion
 - Identifies the highest API Level you have tested your app with
 - **Compiler**
 - In most cases, compile app with the lowest API Level you want to support
 - Compiler assures that you don't access any features belonging to a higher API Level
 - Use caution when compiling with API Level higher than minimum target
 - You must assure that you make no "unprotected" calls to higher level classes/methods

Summary

→ **Each Android release has 2 identities**

- Platform version identified by a decimal name
- API Level identified by an integer

→ **Use the `AndroidManifest.xml` to identify API Level support**

- Specify the minimum API Level you support
- Specify the highest API Level you tested

→ **Use the SDK Manager to manage your development environment**

- Run periodically to assure that you have the latest tools
- Add new API Level support