Android for .NET Developers Series
Getting Started
Android Studio

Jim Wilson

jimw@jwhh.com

@hedgehogjim

http://facebook.com/hedgehogjim

pluralsight
hardcore developer training

Outline

- The role of Android Studio
- Installing Android Studio
- Using Android Studio
- Setting the Darcula theme

A giant step in the right direction

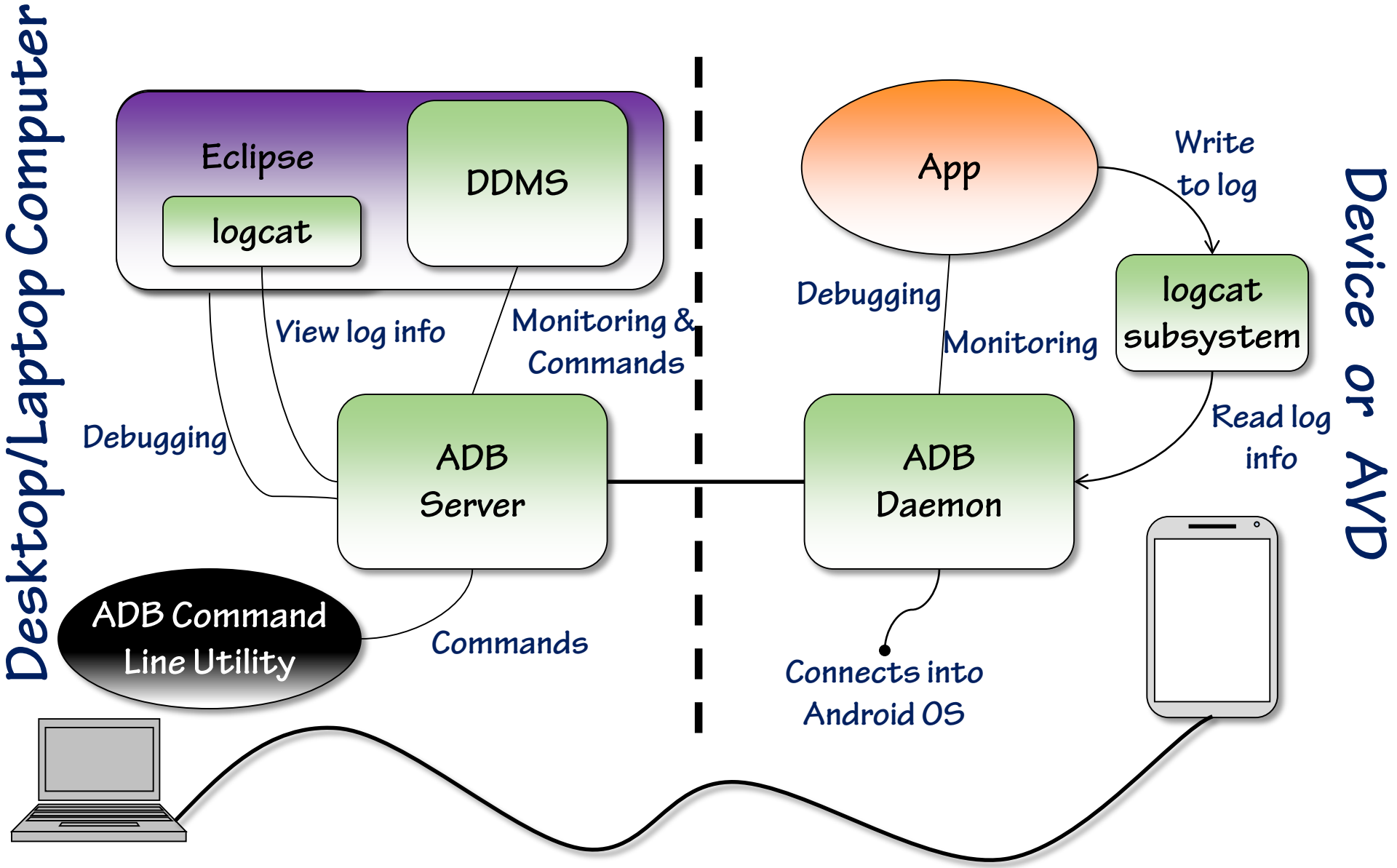
- **Android development is moving to a new IDE**
  - Since Android's inception, Eclipse has been the IDE endorsed by Google
    - But not the only IDE
  - For most of the life of Android, JetBrains has offered an alternative
    - IntelliJ IDEA
    - IntelliJ offered both a free "Community Edition" and a for-pay "Pro Edition"
  - Google I/O 2013 announced a new Android developer IDE
  - Android Studio is the new beginning
    - Based on JetBrain's IntelliJ IDEA Community Edition
    - A giant step forward from Eclipse

Google I/O 13

JetBRAINS

IntelliJIDEA

Android tools architecture overview

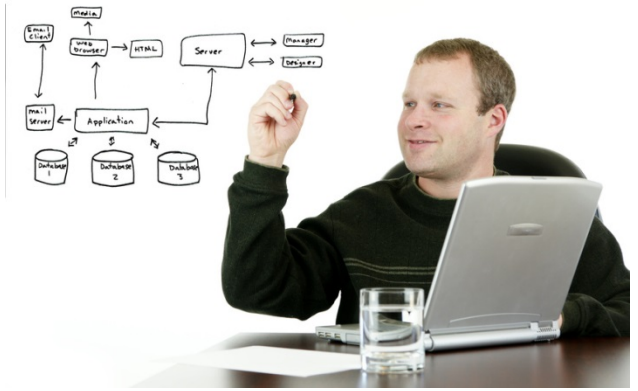


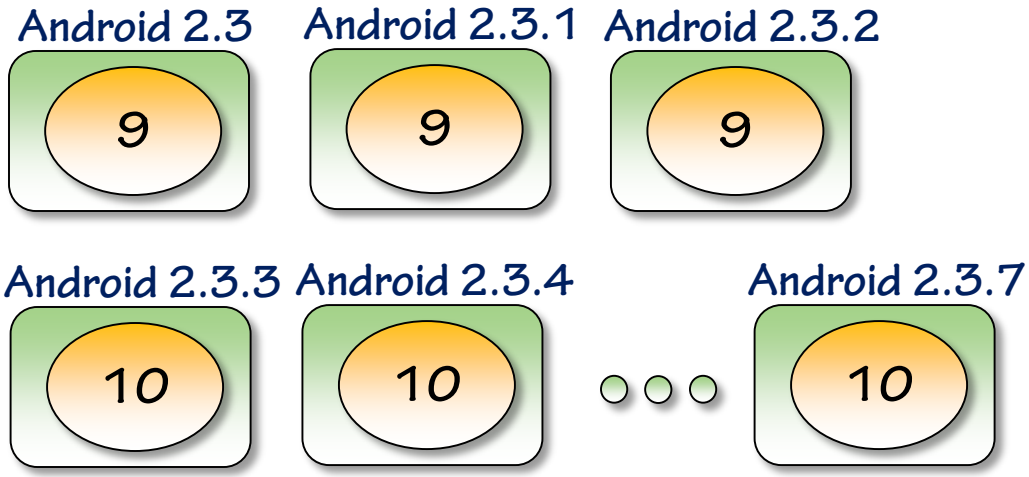
Summary

- **Android Studio is now the Google endorsed development environment**
  - Based on JetBrains IntelliJ IDEA
- **Replaces Eclipse**
  - Eclipse is still around
  - Other Android tools are used as before
- **JDK must be installed separately**
  - Remember to set the JAVA_HOME environment variable

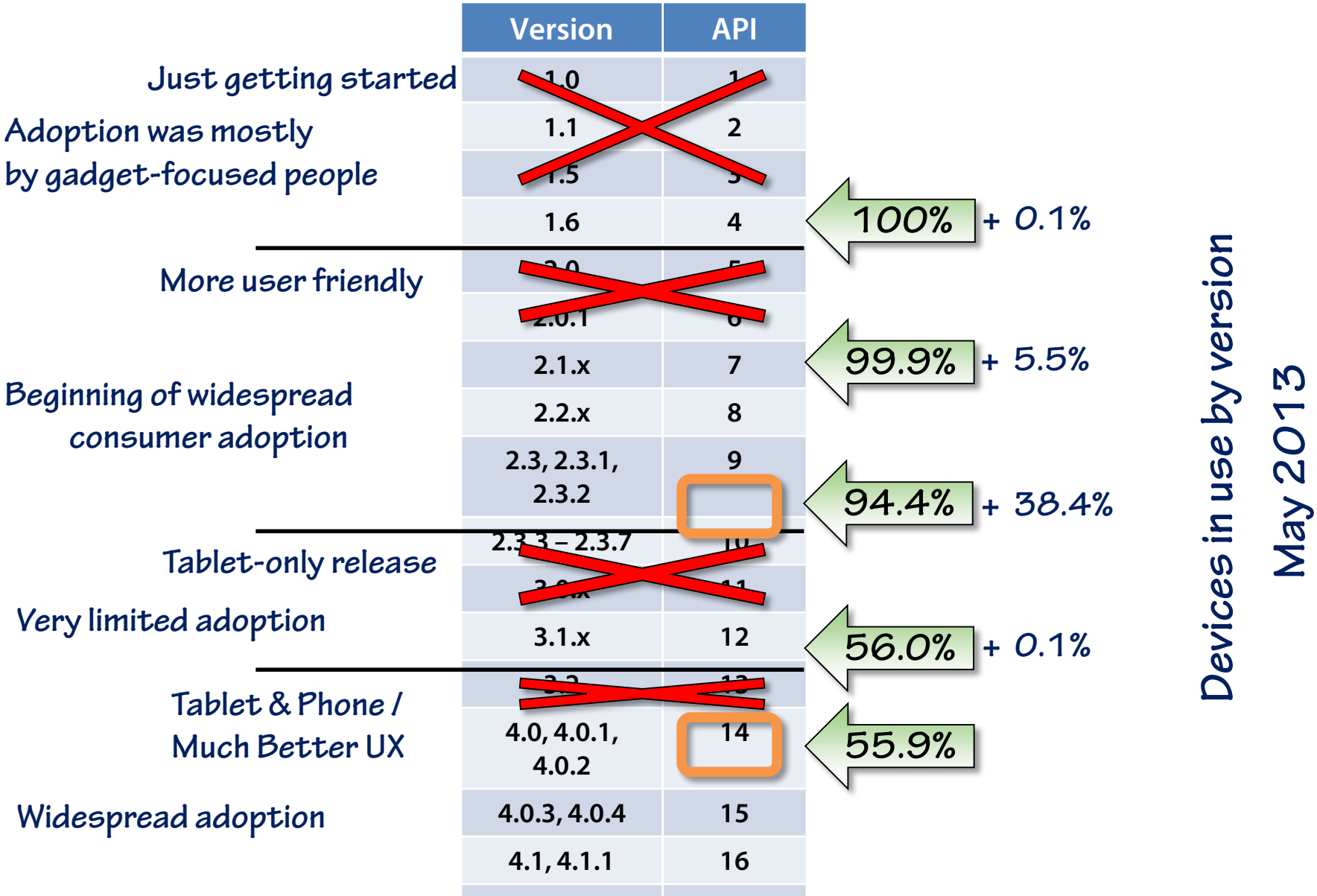
Android version identity

- **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





Selecting Android platform and API level



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