# 18/02/28 安卓会议

编写日期：2018.2.27

编写人：温苏彬

[18/02/28 安卓会议 1](file:////Users/vane/Desktop/18:02:28会议记录.docx#_Toc507492975)

[会议大纲 3](#_Toc507492976)

[会议总结 4](#_Toc507492977)

### 会议大纲

为方便以下会议内容的展开，会议前各小组成员请先了解GitLab-manage-document-android 中的文档。

1. 讨论蛋壳《Android开发手册》的可行性，共同确定出一套编码规范并于会议结束后正式启用。
2. 讨论蛋壳《Android适配方案》的可行性，并于会议结束后正式启用。
3. 浅谈《模块化方案》

### 会议总结

* 关于Android开发编程规范规约，大会讨论没有产生太大疑义，但有所补充，在往后的新项目中将开始实行该规约。

补充内容：

描述：在量级比较大的项目中，例如易道在项目结构上采用的是模块化方案，在编译打完整包时，出现资源命名冲突。为了解决这一问题，易道小组重新将资源文件规范命名，并添加命名约束。例如在定制间模块中，图片资源统一添加“custom\_”前缀，在定制间module的build.gradle 中的android 代码块中添加” resourcePrefix 'custom\_' ”，当该module中的资源不符合约束规范时提示警告，但可以编译通过。

android{

resourcePrefix 'custom\_'

}

讨论结果：这一方法只是起到提醒的作用，实质上还是需要我们手动更改资源名称。在前阵子与UED共同讨论制定的切图命名规范中，也只是制定了通用的简单的命名规范，对于像易道这样庞大的项目，切图之多令人乍舌，切图命名更是令人头疼。在往后遇到复杂、庞大的项目中，切图命名需要UED与技术组共同商榷，制定出一套易理解、易搜索的切图规范。

* 关于Android适配，会议通过《Android适配方案》作为基本适配方案，具体实现视项目特性而定。
* 浅谈模块化开发

如何提高臃肿模块打成完整包的编译速度？

在项目进行到后期阶段时，各模块基本定型，为了提高编译速度，我们将基本完成的(不变动)、需求变更不大的模块打成aar包，并让壳App进行依赖，这样当需要打完整包时，不变动的module则不需要再次进行编译，加大编译速度。但同时需要注意的是，当基础包(basic-lib)发生修改时，检查当成aar包的module是否造成影响，是则需要重新更新aar包。

//依赖不变更aar包

compile(name: 'user-release', ext: 'aar')

compile(name: 'social-release', ext: 'aar')

compile(name: 'store-release', ext: 'aar')

compile(name: 'operation-step-release',ext: 'aar')

compile(name: 'push-release', ext: 'aar')

//直接依赖不稳定module

compile project(':customroom')

compile project(':demand')

当依赖aar发生异常时的解决方案之一：

repositories{

flatDir{

dirs 'libs'

}

}