**配置管理之发布管理**

　　发布管理是构建完成后，把要部署的版本按照一定的标准部署到目标环境的职能。发布管理一般专注于系统打包，然后把系统提测到QA环境，上线到生产环境。

　　生产环境，对于软件公司来说，就是将软件交付给客户；对于物联网企业来说就是把产品发布到线上环境。

　　发布管理是构建管理和部署管理中间这一段的职能，但是实际工作中没有必要分的那么清。通常，如果我们通过系统自动化的进行版本管理、构建管理和部署管理，中间的发布管理这一块就更体现不出来。不过实际上发布管理这一块还是有很多内容要做，包括自动化打包过程、以一个有规律、易识别的方式来对可发布版本命名，存档等。

**发布管理是研发过程中，不同团队间协调沟通的一个重要基础。**

**举个例子：**

　　有的时候研发会问**[测试](javascript:;" \t "_self)**人员，6583的测试完了么？6583是什么？是一个**[Bug](javascript:;" \t "_self)** ID？有这个可能；是一个代码的版本号？也有这个可能。比如研发人员修复了某个bug，然后提交了代码，代码的修订版本就是6583。但实际工作中，6583多数情况是那个名字为6583的包，那个将要发布的产品名字， 比如 6.1.2.6583.tar.gz。

**再举个例子：**测试人员之间也会经常这样说:"小妹儿，昨天晚上6583那个包上（线）了么？”，“那个包啊，没上。回归测试没通过，后来新打了个6584，上的是那个包”

　　发布管理的输入是研发人员的代码，输出是一个可发布的包；而发布管理输出的这个包，是测试人员的输入，是软件发布、系统上线的基础。只有保证了这一环有条不紊、井井有条，才能保证研发的步伐一步一步的往前走，而不是经常要揪心地回头。

**举个例子：**

　　路人甲：“昨天3356的包发布了么？”

　　路人乙：“没有，里面没有多语言支持的资源文件”

　　路人甲：“不对啊，我刚拿到的这个3356里面有啊”

　　路人乙：“。。。。你拿到的包和我拿到的 md5 值都不一样”

**发布管理原则**

　　－ 把构建产物进行打包、形成一个可发布的版本

　　－ 用一个不可变、易识别的 ID 标识可发布版本

　　－ 自动化打包、标识可发布版本的过程

　　－ 发布管理的过程要快速、可靠

　　－ 有快速核实可发布版本唯一性的机制

　　－ 存档可发布版本、提供回溯和审查

**把构建产物进行打包、形成一个可发布的版本**

　　构建的目标是快速的把源代码转变成二进制文件。发布的一个目标就是把这些生成物按照一定的结构打包，以便于后面部署、上线。为了好部署上线、或者做成安装盘，我们通常需要把构建出来的文件重新整理，形成一个可发布或者可上线的包。实际工作中，还诞生了很多软件公司专门作安装包的软件，例如著名的安装包制作的软件 InstallShield。