

# Maven3 实战笔记 16Maven 总结

刘岩

Email:suahuanzheng7784877@163.com

## 1. 前言

用了 15 篇 Blog 做 Maven3 的学习笔记，最后想想好像觉得总是比较冷冰冰的，将知识点列出来，停留在如何使用 Maven 的层面。那么咱们抛开书本，回过头来再看看哪些 Maven 的功能点，再来总结总结 Maven 到底为我们做了些什么，使用 Maven 的好处到底体现在哪里。

## 2. Maven 改变个人习惯

在没有 Maven 之前，我们开发一个新项目的时候，总是要引入很多第三方的组件为我们提供相应的服务功能，一般是以 jar 包的形式提供给我们。那么我们作为开发者一般都是比较关注直接依赖包，比如项目用到了 xmemcached 作为缓存客户端，而 xmemcached 组件 ( 1.2.5 本版本之前 ) 要想正常使用，还依赖于该组件的同一个作者创建的 slf4j 组件。slf4j 在咱们这个新建的项目中就属于非直接依赖，间接依赖。在没有别人或者相关文档辅导的情况下，当拿到 xmemcached 包后是过程中报出了相关错误后才知道缺少了 slf4j 相关包，之后再打开 Google 搜索 slf4j 关键字，从而下载相关 slf4j 组件。很多时候我们都是这么做的。自从有了 Maven 后，开发人员其实只需要关注直接依赖就可以了，构建项目时候 Maven 会根据组件自身的描述文件 pom.xml 自动去找寻该组件的依赖包，从而保证组件的正常使用。

而且项目还可以及时使用第三方的组件最新功能体现 ( 稳定版本，非 SNAPSHOT )，第三方组件的最新版本发布后，Maven 仓库会及时反映，下载到本地。

Maven 倡导的是约定大于配置，Maven 项目的源代码包位置、配置文件位置、测试用例

代码位置都是有约定的。出现 Maven 之前，可能有一些项目，尤其是小项目，把配置文件、源代码、脚本一股脑全放在 src 下面，这样做其实呢，也不能说不对，就是给其他人看起来代码结构有点乱。Maven 规定 source code 就放在 src/main/java 下面，配置文件就是放在 src/main/resource，这样给人看起来比较清晰，不会给接手项目的人“一坨 XX”的感觉。

打包方便，在项目前期，先将 pom 配置文件写好，之后根据配置文件的打包类型进行打包，这个功能实际上在 EJB 项目打 ear 包特别明显。当然了使用 IDE，如 Eclipse 进行打包也麻烦不到哪儿去，只不过有些手工过程比较重复罢了。

### 3. Maven 改变团队

对于团队来说，Maven 可以帮助团队内部、以及不同团队之间进行更好的沟通。团队内部可能进行着不同小模块的开发，但是小模块之间是有耦合、调用、依赖关系的，模块程序更新后，可以及时反映到内部 Maven 库中，团队内部有一个比较良好的及时构建和使用反馈。当然使用 IDE 配合 SVN 插件也可以做到，这种方式仍然是目前很多团队的项目构建方式，根据项目的规模，具体情况具体分析吧，有可能项目比较庞大，同一个团队的人还不在于同一个地域，只能对外开放一个 url，可以选择 IDE 配合 SVN、也可以选择 Maven 配合 SVN。这个时候可能 Maven 就比较有优势了，Maven 构建的项目能够及时反映和自动下载、构建。

项目可以配合 Hudson 进行持续集成，持续测试，持续生成报告，持续反映项目的健康状况，持续反映团队状况。对于项目、公司、团队的队员工作情况都是比较清晰的，用数据说话，用邮件提醒驱动开发人员。个人认为，Maven 确实能用这些数据反映项目、反映团队、反映队员的状况，也发挥了团队队员的主观能动性。从另一个层面来看，其实是 Maven 从敏捷开发中得到的其实吧。要求每个队员的水平也比较高，定期就得看看

自己修改的部分在持续构建中有没有出问题，出问题立刻打起精神，找问题、解决问题，让项目更健壮。唉~~~资本主义就是要剥削劳动人民的剩余价值，从员工自身的观点来看这种比较紧张的工作气氛，到底是好是坏，短期也许还行，长期如此，精神上、心理压力、体力、精力是否能够承受，对于这个问题笔者觉得还是值得大家讨论的。但是有一点可以肯定，对公司发展、对项目交付、对客户的满意度，起到的作用绝对是积极的.....

#### 4. Maven 改变项目

Maven 项目都有特点，源代码、配置文件、单元测试文件夹一般都是按 Maven 规定，所以一般的 Maven 项目都比较好阅读，代码放置的位置比较清晰。项目打包后也可以安装到 Maven 库中，现在很多第三方的开源项目，尤其以 Apache 项目基本上都是符合 Maven 规范的。很多研究 Apache 组织下开源项目的朋友们拿到源代码后都是按照 Maven 标准去找寻原文件的。这样有一个统一的规范，比如自己开发个小组件，别人阅读起来也能从容入手。基本上从网上下载的第三方开源 jar 组件，打开 META-INF 文件夹下都有一个 maven 文件夹，里面就是该开源项目的 Maven 描述文件 pom.xml 和 pom.properties。Maven 还能生成项目静态站点，该站点风格有点像 wiki。该站点既包含了项目的一些基本信息，例如项目介绍、项目成员介绍、项目信息的链接，还能利用 Maven 插件生成一些报告数据，如测试覆盖率、代码提交情况等等，这样反映一个项目整体状况，也是用数据说话，这个站点大多数情况下是给用户看的，客户除了关心交付物，也关心项目质量，而在用户使用软件之前，最能体现质量的就是这些写在明处的报告信息。Maven 出现之前可能是根据项目作出的不同阶段来提交给客户不同的成果物，比如软件的功能做完了，应该提交单元测试覆盖率报告，一般使用 Eclipse 插件完成，之后生成一个测试报告，也是 HTML 形势的。到了客户使用后出现的一些问题答疑和使用问题总结阶段，可能会出现一份 word 文档或者高级一点的 Trace 来记述客户使用反馈信息。其实这些都可

以使用风格统一的 Maven 生成站点来体现，到最后最终的站点实际上是一份完整的项目文档，之后项目终验阶段，这份完整的项目信息站点还是很有参考价值的。

## 5. Maven 对于客户来说

其实前面那么多铺垫，都是为了维系客户关系，下次做项目的时候，客户招标就会多倾向你们一些。Maven 功不可没，让客户觉得该公司的团队比较“专业”。开发中暴露的问题越早，改得越早，到后来客户那边反馈的问题就越少，中间的交流、沟通成本也就越低。文档越详细，客户使用中就越顺手，满意度也就越高。开发人员节省了项目周期（当然，工作量还是那么多，只不过使用 Maven 造成的结果是换了一种比较“高雅”的剥削方式罢了），客户付出的成本和时间也就越低，从商务的角度讲，在固定的一段时间内，固有的资金得到了更多的流动，收益也就相应提高。这对于客户来说是比较重视时间的。当然，客户可能不懂技术，但是节省的成本是实实在在的，懂技术客户可能知道 Maven，如果项目引入 Maven，客户会觉得团队与国际接轨，比较专业，比较靠谱，接触，沟通，包括之后的招标都是比较有利的。

总的来说 Maven 带来的益处是比较明显的，而且 Maven 也日趋成熟，使用 Maven 项目的团队也越来越多。