链滴

从一开始的小项目逐渐变成了大项目而得臃肿，怎么办？

作者：Not-Found

原文链接：https://ld246.com/article/1578209412305

来源网站：链滴

这只是一个小项目?

公司有一个新项目，一开始被认为这只是一个小小的项目，根据需求去实现基本的功能即可。就这样目前期投入的人员并不多，两个而已。只花了几周的时间，便开发完了这个小小的项目，之后便进入项目的测试阶。测试通过后，就这样上线了V1.0版本。

在上线的后的短短几天内，这个小小的项目竟然备受欢迎！公司看到了机会，随即便召集了产品开会定制V2.0的需求。这一次会议可不得，竟然是一个大需求！罗列了一大堆的计划，并且要求在20天内成开发与测试。一开始的两个程序员，进行需求评审，结果发现两个人需要在20天内完成这个需求着困难，就算天天007也无法在20天内完成，因此也就多招了四名新的程序员。

在两名老鸟的带领下，这四名新人很快便熟悉了项目和需求。经过20个工作日后，顺利的完成了开发测试，可项目却变得有些臃肿，不过也不影响V2.0版本的上线。是的，不出所料这个项目比之前更火了。公司又再次看了机会，召开了非常重大的产品会议。

这次的需求，加了各种各样的功能，美其名曰：这是你从未没有体验过的全新版本！。经过一系列的求评审，又招进了几个程序员，并要求在10个工作日内完成这个需求。

可这次却不一样了，因为上一个版本要求给的时间太短，项目已经逐渐变得臃肿，新招的程序员费了大的功夫才把项目的部分给理顺，这次的需求开发的时间已经超过了10个工作日，不过最终还是上线。可这次上线后却是差评如潮，经常会遇到一些莫名其妙的BUG、卡顿等等。这可被老板关注到了，对大家说到：“大家这段时间辛苦了，可最近的反馈却不是很好，希望大家努力一下把这些问题给解”。有人提出来因为一开始并没有想到这个项目会有如此火爆，经过多次的版本迭代，项目已经变得其臃肿，想要解决这个问题就要将项目进行重构。

老板听后，心里却想着：“重构项目？貌似并不能带来收益。”，随后便说到：“好，那大家辛苦一了，努力在一个月内重构完！这次重构完后，给大家发一笔项目奖金。越早完成，金额越大！”。于，在老板的激励下，大家进行了重构！

项目重构

经过大家的交流讨论后，发现痛点在于项目的所有的代码都混合在一起了，新人不好直接理解这个项，而且项目中有大量的重复代码。于是决定将项目拆分——根据不同的模块、业务、功能。

首先，这个项目需要统一的去管理项目的依赖，这个项目包含了所有的模块，所有的模块都是它的子块并且继承了这个顶级项目的依赖：

```
Project >
 pom.xml -- 统一管理依赖
```

```
...
    <groupId>xin.codedream.project</groupId>
    <artifactId>Project</artifactId>
    <version>0.0.1-SNAPSHOT</version>

    <properties>
        <java.version>1.8</java.version>
        <spring-dependencies.version>2.2.2.RELEASE</spring-dependencies.version>
    </properties>

    <dependencyManagement>
        <dependencies>
            <dependency>
```

```
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-dependencies</artifactId>
            <version>${spring-dependencies.version}</version>
            <type>pom</type>
            <scope>import</scope>
        </dependency>
        ...
    </dependencies>
</dependencyManagement>
<build>
    <pluginManagement>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
                <version>${spring-dependencies.version}</version>
            </plugin>
            ...
        </plugins>
    </pluginManagement>
    ...
</build>
...
```

在确定了父级项目管理着所有依赖后，很快便建立了第一个子模块common-core:

```
Project >
 common-core > 公共的核心模块
  pom.xml -- 继承自Project
 pom.xml -- 统一管理依赖

...
<parent>
    <groupId>xin.codedream.project</groupId>
    <artifactId>Project</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <relativePath/> <!-- lookup parent from repository -->
</parent>
<groupId>xin.codedream.project</groupId>
<artifactId>common-core</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>>common-core</name>
<description>common core</description>

<dependencies>
...
</dependencies>
...
```

虽然万事开头难，但只要有了第一步，后面再增加新的模块也就快了许多。在大家的努力下，依葫芦瓢的又增加了些的模块：

```
Project >
 common-core > 公共的核心模块
  pom.xml -- 继承自Project
```

common-model > 公共Model模块，有不同项目，但需要共用的Model
 user-model > 用户Model
 image-model > 图片Model
 ...
 pom.xml
user-service > 用户服务模块
image-service > 图片服务模块
...
pom.xml -- 统一管理依赖

经过这样的拆分，所有的项目的依赖都继承自最顶层，以后无论是新增依赖还是更新依赖的版本，都常方便。不需要再将每个项目都改一遍，也不会造成依赖混乱的问题。并且再有什么新的功能添加，需要新增模块即可。还可以将不同的模块，分别交给不同的人进行开发和维护。比所有人都维护着一未拆分的大项目要容易许多。

经过大家共同的努力，仅在短短的20多天内就完成了重构。老板看到大家工作很努力，于是就发了100块的奖金，让大家平分。就这样，重构后的版本上线了，体验也比以前好了不少。但随着用户的增量，这个单体项目竟然有些要撑不住的迹象。

遭遇瓶颈，该怎么寻求突破?

这个项目的用户数量逐渐变多了，原先的单台高配置服务器，也撑不住了。经常会出现CPU使用率10%，甚至在突然爆发的流量下，还会出现宕机的情况。老板又一次发现了问题，再次召集了大家开会行讨论。这次可犯难了，大家都没有提出一些好的意见和建议，老板有些失望的散会了。可偶然的机，竟然了解到了一个新颖的名词集群...

blog:https://www.codedream.xin