提高开发效率这方面，业界研究得更多，主要途径有两点：加快开发速度 ++合理分层

提高开发效率这方面，业界研究得更多，主要途径有两点：加快开发速度，减少变更代价。怎样才能加快开发速度呢？如果我们的开发不是重新造轮子，而是 每一次做新产品都可以利用已有的东西，那就会好很多。怎样才能减少变更代价呢？如果我们能够理清模块之间的关系，合理分层，每次变更只需要修改其中某个部 分，甚至不需要修改代码，仅仅是改变配置就可以，那就更好了。 我们先不看软件行业，来看一下制造行业，比如汽车制造业，他们是怎么造汽车的呢？造汽车之前，先设计，把整个汽车分解为不同部件，比如轮子，引擎，车门， 座椅等等，分别生产，最后再组装，所以它的制造过程可以较快。如果一辆汽车轮胎被扎破了，需要送去维修，维修的人也没有在每个地方都修一下，而是只把轮胎 拆下来修修就好了，这个轮胎要是实在坏得厉害，就干脆换上个新的，整个过程不需要很多时间。

[](http://blog.jobbole.com/)

* **[首页](http://blog.jobbole.com/)**
* **[最新文章](http://blog.jobbole.com/all-posts/)**
* **[经典回顾](http://blog.jobbole.com/tag/featuredpost/)**
* **[开发](http://blog.jobbole.com/category/programmer/)**
* **[设计](http://design.jobbole.com/)**
* **[极客](http://blog.jobbole.com/category/geeks/)**
* **[IT技术](http://blog.jobbole.com/category/it-tech/)**
* **[业界](http://top.jobbole.com/category/news/)**
* **[职场](http://blog.jobbole.com/category/career/)**
* **[创业](http://blog.jobbole.com/category/startup/)**
* **[访谈](http://blog.jobbole.com/category/interview/)**
* **[在国外](http://blog.jobbole.com/category/overseas/)**

[伯乐在线](http://www.jobbole.com/" \o "Go to 伯乐在线.) > [首页](http://blog.jobbole.com/) > [所有文章](http://blog.jobbole.com/all-posts/) > [Web前端](http://blog.jobbole.com/category/webfront/) > Web应用的组件化开发（一）

# Web应用的组件化开发（一）

2014/01/16 · [Web前端](http://blog.jobbole.com/category/webfront/" \o "查看 Web前端 中的全部文章), [开发](http://blog.jobbole.com/category/programmer/" \o "查看 开发 中的全部文章) · [前端开发](http://blog.jobbole.com/tag/%e5%89%8d%e7%ab%af%e5%bc%80%e5%8f%91/), [徐飞](http://blog.jobbole.com/tag/%e5%be%90%e9%a3%9e/)

分享到：**30**

* **[电商大数据应用之用户画像](http://www.imooc.com/view/460?from=jobboleblog" \o "电商大数据应用之用户画像" \t "http://blog.jobbole.com/56161/_blank)**
* **[HTML5之元素与标签结构](http://www.imooc.com/view/24?from=jobboleblog" \o "HTML5之元素与标签结构" \t "http://blog.jobbole.com/56161/_blank)**
* **[Android-节日短信送祝福（功能篇）](http://www.imooc.com/view/517?from=jobboleblog" \o "Android-节日短信送祝福（功能篇）" \t "http://blog.jobbole.com/56161/_blank)**
* **[Hibernate初探之多对多映射](http://www.imooc.com/view/452?from=jobboleblog" \o "Hibernate初探之多对多映射" \t "http://blog.jobbole.com/56161/_blank)**

原文出处： [徐飞的博客（@民工精髓V）](http://blog.xufei.gitpress.org/~posts/2013-11-20-Web%E5%BA%94%E7%94%A8%E7%9A%84%E7%BB%84%E4%BB%B6%E5%8C%96%E5%BC%80%E5%8F%91%EF%BC%88%E4%B8%80%EF%BC%89.md" \t "http://blog.jobbole.com/56161/_blank)

## **基本思路**

# **1. 为什么要做组件化？**

无论前端也好，后端也好，都是整个软件体系的一部分。软件产品也是产品，它的研发过程也必然是有其目的。绝大多数软件产品是追逐利润的，在产品目标确定的情况下，成本有两个途径来优化：减少部署成本，提高开发效率。

减少部署成本的方面，业界研究得非常多，比如近几年很流行的“去IOE”，就是很典型的，从一些费用较高的高性能产品迁移到开源的易替换的产品集群，又比如使用Linux + Mono来部署.net应用，避开Windows Server的费用。

提高开发效率这方面，业界研究得更多，主要途径有两点：加快开发速度，减少变更代价。怎样才能加快开发速度呢？如果我们的开发不是重新造轮子，而是 每一次做新产品都可以利用已有的东西，那就会好很多。怎样才能减少变更代价呢？如果我们能够理清模块之间的关系，合理分层，每次变更只需要修改其中某个部 分，甚至不需要修改代码，仅仅是改变配置就可以，那就更好了。 我们先不看软件行业，来看一下制造行业，比如汽车制造业，他们是怎么造汽车的呢？造汽车之前，先设计，把整个汽车分解为不同部件，比如轮子，引擎，车门， 座椅等等，分别生产，最后再组装，所以它的制造过程可以较快。如果一辆汽车轮胎被扎破了，需要送去维修，维修的人也没有在每个地方都修一下，而是只把轮胎 拆下来修修就好了，这个轮胎要是实在坏得厉害，就干脆换上个新的，整个过程不需要很多时间。

席德梅尔出过一款很不错的游戏，叫做《文明》（Civilization），在第三代里面，有一项科技研究成功之后，会让工人工作效率加倍，这项科技的名字就叫做：可替换部件（Replacement Parts）。所以，软件行业也应当引入可替换的部件，一般称为组件。