

74 | 开源、云服务与外包管理

2020-01-17 许式伟

许式伟的架构课

[进入课程 >](#)



讲述：姚迪迈

时长 11:16 大小 10.34M



你好，我是七牛云许式伟。今天我们聊的话题是有关于分工的。

在这一讲之前，我们涉及到分工这个话题，基本上都局限于企业内部，且大多数情况下主要在同一个团队内部。但今天我们聊的是更大的分工：跨组织的分工与协作。

外包及其理想模型

在软件工程中，我们第一个接触的外部分工毫无疑问是外包。所谓外包，就是将我们转



但是，软件工程项目的分包实际上成功率非常低。这背后有其必然性，它主要表现在以下这些方面。

其一，任务表达的模糊，双方容易扯皮。期望需求方能够把需求边界说清楚，把产品原型画清楚，把业务流程讲清楚，这非常难。有这样专业的需求表达能力的，通常软件工程水平不低，遇到这样的需求方，绝对应该谢天谢地。这种专业型的甲方，它大部分情况下只发生在项目交付型分包，而非产品功能分包。更多的产品分包，一般是甲方不太懂技术，需要有团队替自己把事情干了，他好拿着产品去运营。

其二，交付的代码质量低下，长期维护的代价高。软件工程不是项目，它都需要长长久久地运行下去。但是接包方的选择相当重要。因为接包方的质量相当参差不齐，遇上搬砖的概率远高于设计能力优良的团队。事实上，有良好设计能力的团队，多数情况下也不甘于长期做分包。

其三，项目交接困难，知识传承效率很低。软件工程并非普通的工程，就算交付的结果理想，项目交接也非常困难。所以分包项目第一期结束后，如果运营得好，往往项目还继续会有第二期、第三期。这里的原因是你只能找同一拨人做，如果换一波人接着做，考虑到知识传承效率低下，往往需要很长的一个交接周期。

那么，分包的理想模型是什么？

上面我们已经说到，分包在通常情况下，专业的甲方需要说清楚需求，这样双方就没有分歧。但是，更好的做法其实不是分包需求，而是分包实现。

也就是说，作为专业的甲方，我自己做好需求分析，做好系统的概要设计。进一步，我们把每个模块的业务范畴与接口细化下来。我们以此作为分包边界。假设分了 N 个模块，我们可以把它们平均分给若干个接包方。

这种方式的分包，甲方相当于只留了架构师团队，实现完全交给了别人。但是它与普通的外包完全不同，因为根本不担心知识传承的问题。每个模块的接包方对甲方来说就真的只是干活的。

接包方拿到的是模块的规格说明书。他要做的是模块的详细设计的实现部分，其中最为核心的是数据结构设计。对于服务端，甲方可以规定所采用的数据库是什么，但是把表结构的设

计交出去。

进一步，如果模块的外包说明书中还规定了单元测试的案例需要包含哪些，那么这个模块发生设计偏离的可能性就很低。


外包的验收需要包含模块的实现设计文档，里面描述了数据结构 + 算法。另外，单元测试部分，每个测试场景，也填上对应的测试函数的名称。

实际会有人这样去外包么？


我不确定。但我们可以把它看作一种分工的假想实验。这个假想实验可以充分说明架构师团队的重要性。有了一个好的架构师团队，他们设计合适的系统架构，对每个模块的规格都做了相应的定义，他们验收模块的实现。

这样，项目就可以有条不紊地展开。甚至，研发进度可以自如控制。嫌项目进展太慢？找一倍的接包方，就可以让工程加速一倍。

所以，这个外包假想实验也说明了一点：我们的平常项目之所以进度无法达到预期，无他，团队缺乏优秀的架构师而已。

让我们把软件工程看作一门科学。我们以  工程师思维的严谨态度来看它。我们减少项目中的随意性，把架构设计的核心，模块规格，也就是接口，牢牢把控住。这样，项目的执行风险就完全消除了。

哦不，还有一个最大的执行风险没有消除。我怎么证明这个系统架构的分解是对的？不会出现每个模块做好了，但是最终却拼不起来？

我们前面在 “ 架构：系统的概要设计” 这一讲中实际上已经谈过这事的解决方法：系统设计的产出要有源代码，它是项目的原型。关键模块有 mock 的实现，业务系统的关键 UserStory 都串了一遍，确保系统设计的正确性。

这个假想实验是有趣的，它可以让你想明白很多事情。甚至可以把它看作理解这个专栏的架构思维核心思想的钥匙。

我希望，它不只是一个假想实验。

开源与众包

我们把话题拉回到跨组织的分工。

除了传统的外包外，在软件工程中出现的第二类外包是众包，它以开源这样一个形态出现。

从分工角度，开源的核心思想是让全社会的程序员共同来完成一个业务系统。

开源的优势非常明显。对于一个热门的开源项目，它的迭代进度是非常惊人的，因为它撬动的资源太大了。

但不是开源了就能够获得这样的好处。

虽然成功的开源项目风风火火，但是我们也应该意识到，对于那些并没有得到关注的开源项目，它们的迭代速度完全无法保障。最终，你可能还是只能靠自己的团队来完成它的演进。

从这个意义上看，开源是一种商业选择。你得持续经营它。没有经营的开源项目不会成功。你需要宣传它，你自己也得持续迭代它，你还要为它拉客户。有客户的开源项目自然就有了生命力。

另外，开源这种形态，注定了它只能做大众市场。如果一个业务系统它的受众很少，就比较难通过开源获得足够的外部支持。


所以绝大部分成功的开源项目，都属于基础设施性质的业务系统，有极其广泛适用的场景。例如，语言、操作系统、基础库、编程框架、浏览器、应用网关、各类中间件等等。我们这个架构课重点介绍的内容，大部分都有相应的开源实现。

开源对信息科技的影响极其巨大，它极大地加速了信息科技前进的进程，是全球共同精诚协作的典范。

没有参与过开源的程序员是需要心有遗憾的。开源沉淀下来的协同方法与工作流，今天被无数公司所借鉴。

没有开源，我们无法想象这件事情：那么多形形色色的企业，今天其中绝大部分，它们的软件工程协同方法与业务流竟然如此相似。

这是开源带来的另一种无形资产。

如果大家没有忘记的话，可能能够回忆起来，在谈完软件工程的宏观视角之后，我首先聊的是“团队的共识管理”。为什么这很重要？因为它是团队协作效率的最大基础。如果连对协作的工作流都没有共识，那团队真的是一盘散沙了。

今天我们几乎不会遇到工作方式上的问题，不是别的原因，是开源给予我们的礼物。它让全球的程序员、全球的科技企业，都养成了一模一样的工程习惯。

云计算与服务外包

云服务是新的跨组织分工的形态。无论是传统的外包，还是开源的众包，它们都属于源代码外包。这类外包的共同特点是，它们不对结果负责。

对于传统外包，项目验收结束，双方一手交钱一手交货，至于用得好不好，那是甲方自己的事情。

对于开源软件来说，那更是完全免责，你爱用不用，用了有什么问题责任自负。当然有很多公司会购买开源软件的商业支持，这不难理解，除了有人能够帮助我一起完成项目上线外，最重要的是要有人能够给我分担出问题的责任。

互联网为跨组织协同带来了新的机会。我可以 24 小时为另一个组织服务，而无需跑到对方的办公室，和他们团队物理上处在一起。

这就是云计算。云计算从跨组织协同的角度来看，不过是一种新的交付方式。我们不再是源代码交付，而是服务交付。所以，你也可以把云计算看着一种外包，我们称之为服务外包。

大部分的基础设施，都可以以服务外包的方式进行交付。这中间释放的生产力是惊人的。

一方面，云计算与传统外包不同，它对结果负责，有服务 SLA 承诺。一旦出问题，问题也可以由云服务提供方自己解决，而无需业务方介入，这极大降低了双方的耦合，大家各司其职。

另一方面，它简化了业务方的业务系统，让它得以能专注自己真正的核心竞争力的构建。

站在生产效率角度看，不难理解为什么我们会坚信云服务是未来必然的方向。

外包方式的选择

任何企业都存在于社会生态之中，我们无法避开组织外部的分工协同问题。

怎么选择跨组织的协同方式？

在七牛，自成立以来我们就一直有一句话谈我们对跨组织协同的看法：

我们尽可能不要做太多事情。非核心竞争力相关的，能够外包的我们尽可能外包。

在外包选择上，我们优先选择云服务，次选开源，最后才考虑传统的外包。

这句话有它一定的道理，但也有它模糊的地方。

首先是关于“核心竞争力相关”。我们并没有太清晰地去定义什么样的东西是我们核心竞争力相关，什么不相关。

一些程序员对此理解可能会比较“技术化”，认为业务系统的核心模块就是核心竞争力。与它相关的东西就是核心竞争力相关。

但更合理的视角不是技术视角，而是业务视角。我们每一家企业都是因为服务客户而存在。所以，与服务客户的业务流越相关，越不能外包，而是要自己迭代优化，建立服务质量与效率的竞争优势。

另外，外包的选择需要非常谨慎。很多开发人员都有随意引用开源项目的习惯，这一定程度上给项目带来了不确定的风险。

我一直认为，开源项目的引入需要严格把关。严谨来说，开源项目引入大部分情况下是属于我说的“基础架构”选择的范畴，这同样是架构师团队需要承担的重要职责，一定要有正规的评估流程。

结语

今天我们聊的话题是跨组织的分工与协同。在形态上，我们可以分为：传统外包、开源与云服务。当然还有就是我们今天没有讨论的使用外部商业软件。

从形态来说，商业软件很接近传统外包，但是从它的边界来说，因为商业软件往往有明确的业务边界，所以在品质上会远高于外包。当然定制过于严重的商业软件例外，它在某种程度上来说退化为了传统外包。

在外包方式的选择上，我们的建议是：

我们尽可能不要做太多事情。非核心竞争力相关的，能够外包的我们尽可能外包。

在外包选择上，我们优先选择云服务，次选开源，最后才考虑传统的外包。

如果你对今天的内容有什么思考与解读，欢迎给我留言，我们一起讨论。下一讲我们谈谈“软件版本迭代的规划”。

如果你觉得有所收获，也欢迎把文章分享给你的朋友。感谢你的收听，我们下期再见。

更多课程推荐

数据结构与算法之美

为工程师量身打造的数据结构与算法私教课

王争

前 Google 工程师



新版升级：点击「 请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 73 | 软件质量管理：单元测试、持续构建与发布

下一篇 75 | 软件版本迭代的规划

精选留言 (9)

写留言



leslie

2020-01-17

每次看许老师的课程都能有所收获：收获不一样的视角和出发点去看待问题与解决问题；老师的课程不止是在教授架构，同时还点出了一个小企业如何去把握技术。

记得郑晔老师的课程总会在结束时强调一节课结束时记住课程结束时强调能记住一件事就好，许老师的课每次学完第一遍我基本上同样只能记住几句老师课程中的经典：

1.开源的核心思想是让全社会的程序员共同来完成一个业务系统；...

展开

1

3



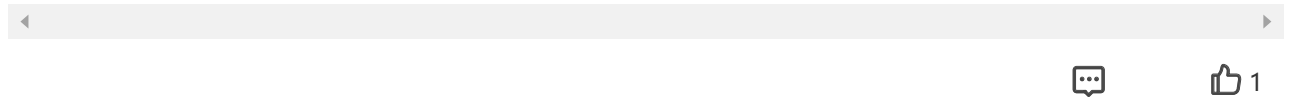
Geek_88604f

2020-01-21

目前有哪些大厂是采用服务外包的？

展开 ▾

作者回复: Netflix, 最著名的案例了



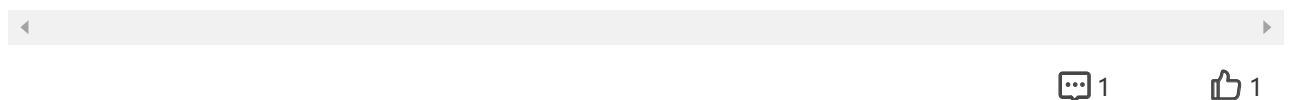
yga

2020-01-19

许老师的视角就是高！让人耳目一新。

有个问题请教一下:文中说引入开源项目一定要有正规的评估流程，不知如何评估，需要考虑哪些点

作者回复: 1、项目的活跃程度；
2、项目的业务范畴、接口规格、实现是否符合预期；
3、项目所处的状态，是快速迭代阶段还是进入稳定期、缺陷的数量与严重性如何；
4、项目活跃参与者的背景。
等等。



K战神

2020-01-17

核心业务最好是自己迭代演进。
我们把其他非核心服务外包给其他团队。
比如云供应商，这里既划清楚了边界也明确了责任。企业都在试图找到这样的点。

迄今为止，我觉得郑烨老师的课被低估了。...

展开 ▾

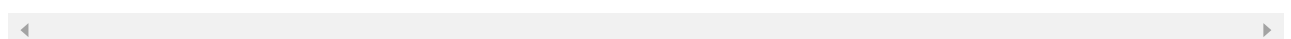


Lane

2020-01-28

外包还可以这么来？这不就是designed by apple，然后富士康、立讯精密来代工的模式吗

作者回复: 是这样





Aaron Cheung

2020-01-17

早上看完 晚上补签到 在老师的说明下 充分认识了“外包”的含义



Geek007

2020-01-17

核心业务我理解也是一个企业的核心竞争力。也就是说企业的客户为何选择你不选择其他人？如果有其他企业带着一大笔钱进入相同的市场，我们会被替代么？核心竞争力是需要牢牢把握在企业自己的手里，否则某一天客户就会被外包公司或者其他公司抢走。比如说苹果公司，它是全球协作的典范，整个产业链有无数分工写作，但是苹果的核心竞争力依然牢牢的把握在苹果自己手里。

展开 ▾

作者回复: 核心系统当然是核心竞争力，这一点毫无疑问。我们讨论更多的是哪些不是，哪些可以外包。



知行合一

2020-01-17

颠覆了我对外包的认识，老师看问题的高度就是比我们高，谢谢许老师。



Jeff.Smile

2020-01-17

许老师很是博学，每节课都感觉像是指示明灯！

展开 ▾

