

08 | 怎样平衡软件质量与时间成本范围的关系？

2019-03-12 宝玉 来自北京

《软件工程之美》



你好，我是宝玉，我今天与你分享的主题是：怎样平衡软件质量与时间、成本、范围的关系。

在 [《从软件工程的角度解读任正非的新年公开信》](#) 这篇文章中，我已经提到了“软件项目管理金三角”的概念。由于这个内容对于软件工程来说，非常之重要，所以我今天特别展开再讲一下。

你会发现，在实际的软件项目中不乏这样的例子：

一个项目，正常估算，要三个月才能完成，但是老板或客户要压缩到一个月完成，而你不知道如何说服他们；

项目开发一半，产品经理告诉你，有一个非常紧急的功能，要增加到这个版本中，你不知道该不该拒绝，或者如何拒绝；

听说迭代模型很好，你也尝试使用迭代模型，但是每次迭代时间到了还是完不成，只能把迭代时间延长，最后又做回传统的瀑布模型了；

你们组用瀑布模型开发，一到项目后期总免不了加班加点赶进度，为什么他们用敏捷开发的加班要少一些？

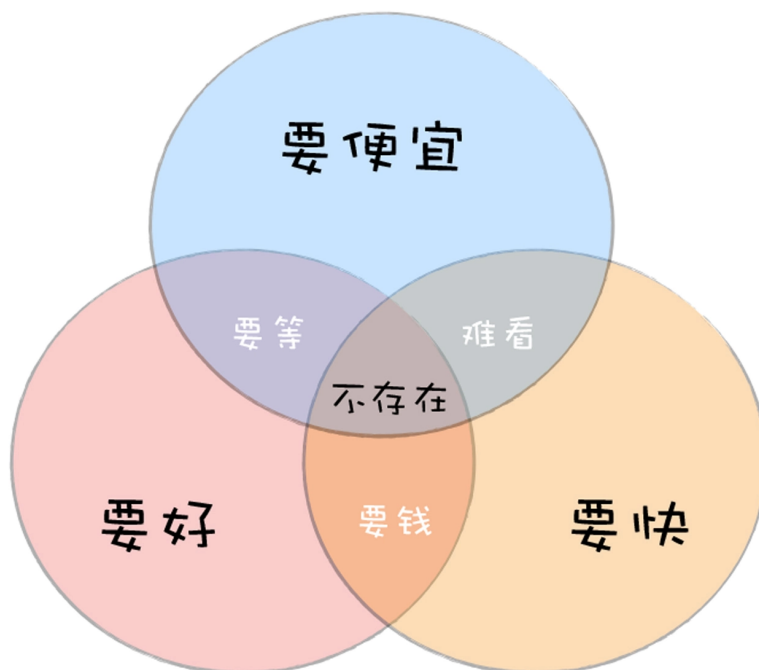
其实，这些日常项目中涉及**时间、成本和范围**的问题，都离不开“**软件项目管理金三角**”的概念。

掌握好这个知识点，学会平衡软件质量与时间成本范围的关系，可以帮助你更好的驾驭项目中的各种问题，也可以帮助你更好地理解软件工程中各个模型，尤其是瀑布模型和敏捷开发。

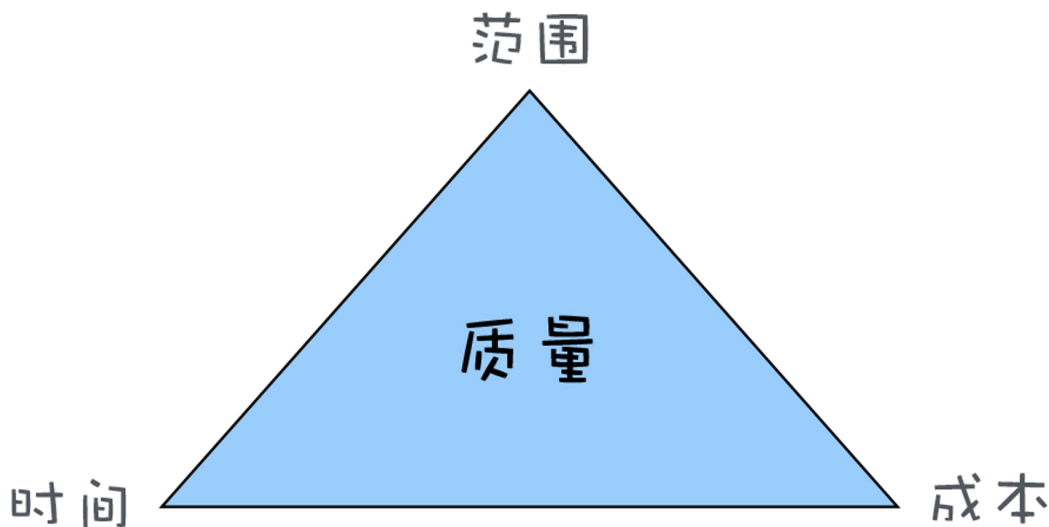
什么是软件项目管理金三角？

在现实生活中，我们都知道，做产品想“多、快、好、省”都占着，是不可能的，最多只能选两样。

想要便宜和质量好，就要花时间等；想要快还要质量好，那就得多花钱；想要又便宜又快，那就得接受难用、质量差。



而在软件项目中，也有一个类似的平衡关系，就是软件质量（产品的质量，客户的满意度）与范围（需要实现多少功能）、时间（多久可以完成）、成本（花多少钱）四个要素之间的平衡。



上面这个图就是著名的项目管理金三角（以下简称“金三角”），三条边分别是时间、成本和范围，中间是质量。

为什么四个要素，是“质量”放在三角形的中间？

因为**软件工程的目标就是要构建和维护高质量的软件**，所以项目的质量是高于一切的。也就是说，“质量”这个因素一般不会妥协，因此把“质量”放在三角形中间，然后在时间、成本、范围这三条边之间寻求平衡。

质量往往也是其他三个因素平衡后结果的体现，想要做的快、成本低、功能多，最后一定是个质量很差的产品。

如何应用“管理金三角”做决策？

我在专栏中常用“道术器”来比喻软件工程中的各个知识点，“金三角”无疑就是“道”级别的。

项目管理其实就是项目中一系列问题的平衡和妥协，而“金三角”理论则为我们的平衡提供了理论指导，了解这三个因素分别对项目其他方面产生的影响，可以帮助你做决策时进行权衡取舍。

当你接手一个项目，项目的进度、成本和范围指标很容易可以跟踪到。有了这些信息，你就可以及时发现问题，调整“金三角”的边，及时解决，以防止这些小问题发展成大问题。

我来举两个例子，看看“金三角”是如何应用的。

老板要压缩项目时间怎么办？

当项目经理，常遇到的问题之一就是时间被压缩，比如文章中开头举的例子，老板问我一个项目多久能完成，我按照经验，觉得要三个月，老板觉得三个月太久了，要砍到一个月就上线。

最开始的时候，我就是据理力争，说这不科学，肯定不行呀。老板说时间点很重要，必须要一个月上线。结果就是大家吵得不欢而散，最后还得加班加点做，质量也不好。

后来我学乖了，先用“金三角”知识分析了一下：老板希望时间是 1 个月，也就是说时间这条边被缩短了，那么结果就是会影响到另两条边：范围和成本，如果另外两条边可以调整，也不是不可以。

于是再遇到这种问题，我就换了一种方式跟老板沟通：“一个月也不是不行，就是我们的需求调整一下，第一个版本只能做一些核心功能，剩下的后面版本再加上（**调整范围**）。另外还得给我加两人，不然真做不完！（**增加成本**）”

这样的方案一提出来，就好沟通多了，最后重点就变成了砍多少功能和加多少人的事情了。

产品经理要临时加需求怎么办？

在文章开篇我提到一种情况，项目开发一半，产品经理告诉你，有一个非常紧急的功能，要增加到这个版本中，怎么办？我们拿“金三角”知识先套用一下。

增加需求，也就是范围这条边要增加，那就必然对成本和时间这两条边造成影响，要么延期，要么增加成本。

面对这种临时加需求的情况，我们也不需要直接说不能加，而是清楚的让产品经理认识到这样做的后果：进度延期，需要更多的成本。如果这个功能真的太重要，可以接受延期，也不是不可以接受，那就重新制定新的项目计划好了。

所以你看，如果我们能应用好“金三角”的知识，很多软件项目中问题，一下子就多了很多方案可以选择了。

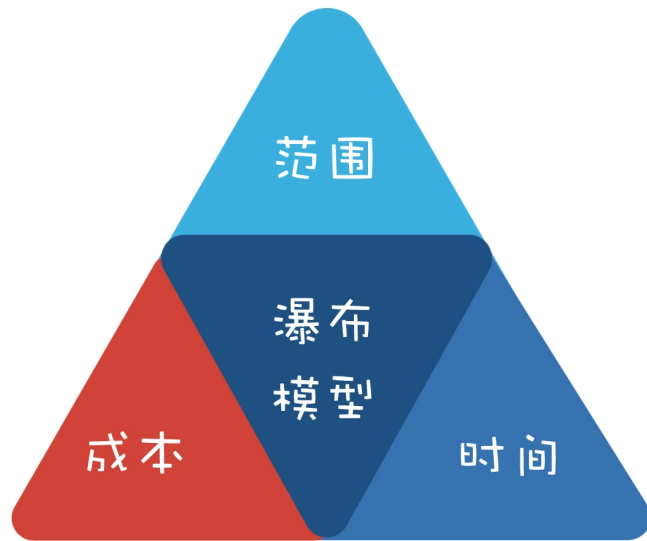
瀑布模型和敏捷开发如何平衡时间成本范围的关系？

除了可以将“金三角”的知识应用在软件项目中，还可以应用它来理解和应用软件工程中的开发模式，尤其是瀑布模型和敏捷开发这两种典型的开发模式。

瀑布模型有严格的阶段划分，有需求分析、系统设计、开发和测试等阶段，通常在开发过程中不接受需求变更，也就是说，我们可以认为**瀑布模型的范围是固定的，其他两条边时间和成本是变量。**

所以使用瀑布模型开发，如果中间发现不能如期完成进度，通常选择的方案就是延期（加班），或者往项目中加入。

固定

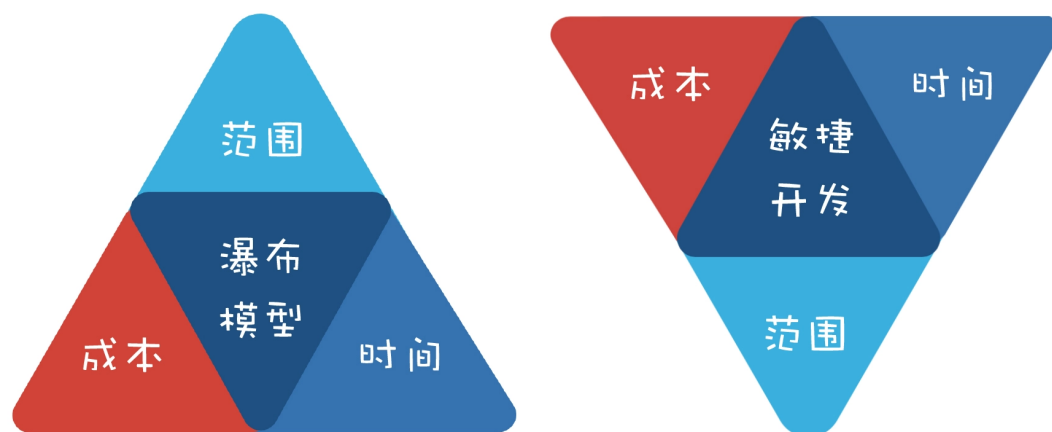


变化

我们再来看敏捷开发，敏捷开发中，是采用固定时间周期的开发模式，例如每两周一个 Sprint，团队人数也比较少。所以，**在敏捷开发中，时间和成本两条边是固定，就只有范围这条边是变量。**

这就是为什么在敏捷开发中，每个 Sprint 开始前都要开 Sprint 计划会，大家一起选择下个 Sprint 能做完的任务，甚至于在 Sprint 结束时，没能完成的任务会放到下个 Sprint 再做。

固定



变化

这时候再想想文章开头我们提到的问题：

听说迭代模型很好，你也尝试使用迭代模型，但是每次迭代时间到了还是完不成，只能把迭代时间延长，最后又做回传统的瀑布模型了。

你现在是不是就明白了：如果不能固定“时间”这条边，就会导致时间也成了变量，迭代无法正常推进。

如何平衡好软件质量与时间成本范围的关系？

那么怎么样才能平衡好软件质量与时间成本范围的关系呢？

前面我们说日常生活中“多、快、好、省”最多只能选两样，其实如何平衡好软件质量与时间成本范围的关系也是一样的道理，我们只能最多选择两样，然后在另一边或者另两条边去寻找平衡。

所以第一件事就是：**从时间、成本和范围这三条边中找出来固定的一条或者两条边，再去调整另一条边。**

下面，我来分析一些案例，帮助你更好地理解。

1. 淘宝网站第一个版本是怎么做到一个月上线的？

这个故事其实我是从极客时间《从 0 开始学架构》专栏看来的，李运华老师在《架构设计原则案例》一文中举了淘宝网站的例子：

2003 年 4 月 7 日马云提出成立淘宝，2003 年 5 月 10 日淘宝就上线了，中间只用了一个月时间。

好，如果你是当时的淘宝网站负责人，马云要你一个月上线淘宝网站，功能还不能少，你怎么办？

第一件事当然是先应用“金三角”分析一下：时间这条边被固定了，只能一个月；功能也不能少，范围这条边也限制住了，那就只能在成本上想办法了。要么一下子雇很多牛人，要么直接买一个现成的电子商务网站，然后修改。

显然，直接买一个网站，再雇一堆牛人的方案最好，所以淘宝网站就这样在一个买来的网站基础上，由一堆牛人快速搭建起来了。归功于淘宝网站的快速上线，刚推出后，正好赶上“非典”，网购需求增大，淘宝网一下子就火爆起来了。

从成本角度我们还有可以去做的，比如说有同学在看完 [《06 | 大厂都在用哪些敏捷方法？（上）》](#) 这篇文章后，也想在团队里面推行代码审查和 CI，但是苦于搭建这一套 git+CI 的系统没有经验，不知道该如何下手，怎么办呢？

我的建议就是刚开始就没必要自己去折腾了，买一套 GitHub 的企业版，加上支持 GitHub 的商业 CI 系统，花不了多少钱，而且可以节约大量搭建这种系统的时间。

2. 极限编程是怎么做到“极限”的？

前面在介绍敏捷开发的时候，也提到了极限编程（eXtreme Programming, XP），是目前敏捷开发主流的工程实践方法，极限编程的“极限”（Extreme），意思就是如果某个实践好，

就将其做到极限。比如：

如果做测试好，就让每个开发人员都做测试；

如果集成测试重要，就每天都做几次测试和集成；

如果简单的就是好，那么我们就尽可能的选择简单的方法实现系统功能；

.....

极限编程的“极限”理念，产生了很多优秀的实践方法，例如持续集成、自动化测试、重构等。

这些实践帮助我们可以在短时间的迭代中，产生高质量的代码。我们用“金三角”的理论来分析一下极限编程在 Sprint 中的应用。

在一个 Sprint 中，计划好了当前 Sprint 要做的工作内容后，那么极限编程怎么帮助我们提高代码质量呢？

一个 Sprint 要做的内容是确定的，相当于成本和范围这两条边都固定了，时间这条边就成为变量了。要么通过加班延长工作时间，要么通过提升效率、减少浪费帮助我们提升时间利用率。

极限编程，就是通过帮助我们提升效率和减少浪费这方面来做的。比如说：

持续集成，通过自动化的方式帮助我们部署，节约了大量需要人去手动部署的时间；

自动化测试，通过自动化测试，节约测试时间，另外，有了自动化测试，可以避免后面修改代码产生 Bug，减少了大量的浪费；

只做刚好的设计，避免设计时考虑了太多不必要的可能，造成浪费。

其实我们在项目中也有很多地方可以借鉴这种思路，比如说写代码的时候，少自己造轮子，多使用成熟的开源或者商业组件，可以提升效率；比如把需求想清楚搞清楚再去开发，可以减少很多返工的时间成本！

3. MVP 模式是怎么诞生的？

这些年流行的 MVP（minimum viable product，最小化的可行性产品）模式，是一种快速推出产品的模式：一开始只推出最核心的功能，满足用户最核心的需求，然后在用户的使用过程中收集反馈，进一步升级迭代。

这种模式怎么诞生的呢？还是应用“金三角”理论，要快速推出产品，还想成本不用太高，那就意味着时间和成本这两条边是固定的，剩下范围这个变量。

所以最简单有效的办法就是砍掉一些重要性不那么高的功能需求，只保留最核心的需求。通过缩小范围的方式，达到快速推出高质量产品的效果。

类似的道理，我们程序员，在遇到很多功能忙不过来的时候，可以主动的去和项目经理协商，砍掉一些不那么重要的需求，把精力放在核心需求上，保证项目可以如期上线。

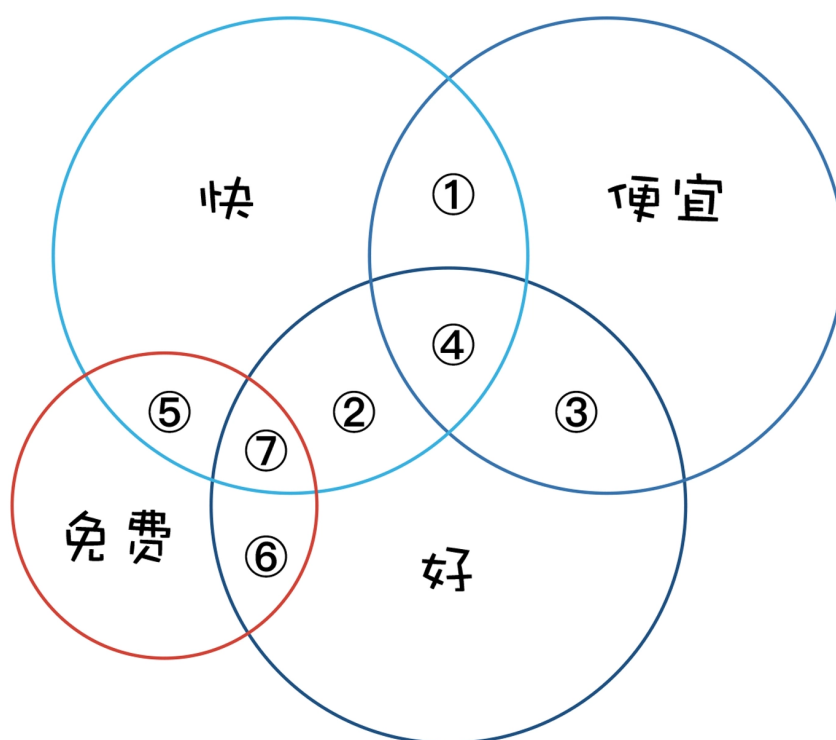
总结

其实，要平衡好软件质量与时间成本范围的关系并不难，你只需要记住，最重要的是根据“金三角”的三条边，找出来固定的一条或两条边，然后去调整剩下的边，达到平衡。

软件项目的“金三角”很多人都知道，主要是不知道如何应用到实际的项目中，希望这篇文章能为你提供一些思路，帮助你在项目中真正应用好这个非常实用的知识。

课后思考

关于今天的内容，邹欣老师在《构建之法》书中，提出了一个很好的问题。我也在这里列出来，希望你可以思考一下。



顾客对于要交付的软件和服务，都是有很多美好的需求的，用户希望软件开发的又快，又便宜（人工便宜），质量又好，最好是免费的。那么，如果只满足部分的需求，我们会得到什么样的软件呢？

例如，上图的 ① 说明，如果希望软件做得又快，又低成本（人工便宜），不考虑其他要求，那么，我们会得到大致什么样的软件呢？

例如，上图的 ⑤ 说明，如果希望软件是免费的，而且要很快交付，越快越好，那么，这样的软件有什么特点呢？

请把 ① 到 ⑦ 的需求组合会导致什么样的软件，会出现什么样的问题，都列出来。

另外，对于质量和时间成本范围的平衡，你有没有什么应用的案例？你对你当前项目的时间、范围和成本都清晰吗？有没有什么可以做的更好的地方？欢迎在留言区与我分享讨论。

感谢阅读，如果你觉得这篇文章对你有一些启发，也欢迎把它分享给你的朋友。

精选留言 (37)



javaadu

2019-03-16

1. 如果希望软件做得又快、有低成本，得到的软件应该是质量差、功能可能不全的软件
2. 如果希望软件免费，又得快速交付，那么得到的软件应该是质量没有保障，功能大概率不全的软件
3. 快速、便宜、质量好的软件是不存在的
4. 之前的工作中，好像没有一个项目可以达到质量和时间、成本之间良好的平衡，最后的结果都是为了保障质量和时间，然后增加人力成本（加班），如果加班也搞不定就对时间做出妥协（延期），范围这个指标倒是比较容易调整的一个指标。
5. 读完这篇文章的收获：（1）软件项目管理本质上就是在成本、范围、时间三个要素中做出合适的妥协的过程，质量是软件项目的目标，不能妥协；（2）管理金三角有助于在软件管理中进行良好有效的沟通，将精力花在事情上，而不是花在PK上，PK不是目标；（3）对于个人成长，也有同样的指导意义，我们的目标是取得个人的成功（名或利），影响这个目标的要素有时间、成本、范围，个人管理的过程也是在这三个要素中做妥协和平衡，例如，我现在积极参加这些知识付费的课程，就是希望用成本来提高自己的学习效率，这样可以在时间和范围上获得一些机会；我如果现在去健身，也会花钱请私教，同样是用成本来提高自己的健身效率，做出这样的选择的原因是我自己感觉相对于这些课程的价格来说，我的时间更加宝贵；但是有些时候，如果价格超出了我的承受能力，那我就不得不做出选择——自己花时间去搞定事情；有时候，成本我也不在乎，但是我的时间还是有限的，那我就只能做减法（少做一些事情）

作者回复：赞，你这个感悟很深刻，不只是项目中，都已经应用到日常生活中了👍

共 2 条评论 >

👍 53



koradji

2019-03-12

对金三角有了进一步的认识和理解，以前只认为它是个三角形而已，还不会用，谢谢老师

作者回复：是呀，这个真的非常实用的，软件项目中很多现象都可以从它身上找到解释，很多问题都可以通过它找到合适的方案。



👍 14



helloworld

2019-06-28

我认为文中关于三要素的说法是错的。专栏里，提到这个三角的时候又总是同时提到质量，时间成本范围又同时影响着质量，对质量的要求又影响了时间成本范围，那么质量应该是要素之一。

有些文章说三要素是时间成本质量，这也是错的，范围没有了。

PMP提到了，三要素是成本质量范围，时间属成本。

花费了成本，换了别的东西，成本沉没了，再也回不来了。时间也是，花了时间和其它成本，换了别的东西，时间沉没了，再也回不来了。

作者回复：时间可以算作成本，但是三要素不代表就必须要把成本和时间合并。

对于一个软件项目来说，时间和成本都是很重要的纬度，有无限的钱和最牛的人，不代表就可以用很短的时间做出来产品。

对于项目各种因素的约束，本身就存在各种不同的解读，我不认为一定就只有一种正确答案，你选择自己认为正确的就好。



👍 10



alva_xu

2019-03-13

老师，其实，我想表达的是，传统的大企业（不是指BAT这类大企业），比如我们企业，IT项目牵涉到三个部门，一个是业务需求部门，一个是IT部门，一个是财务预算审批部门，采取的形式一般都是采用外包方式，而且往往是固定合同，也就是合同价格是确定的，需求范围也是确定的，这样的话，金三角的两条边就定下来了，剩下的就是时间和质量的关系问题了。按照金三角的理论，我们就可以知道前面所述的场景下项目组该重点抓什么了：作为甲方项目经理，重点抓的就是质量和时间了。如何通过提高效率，使单位时间的产出比原来的多（相当于增加了时间），来提高项目的交付质量，是我们甲方IT项目经理最关心的事。所以这时候，我们的方法是建立统一软件框架、提供公共服务组件、制定代码和测试规范、培训乙方团队、搭建CICD平台和自动化测试平台、sonarqube自动代码检测平台等，使原来几周一次测试变成一周几次测试，使原来低质量的代码快速变成高质量的代码... 反正是采用各种方法，提高工作效率，用于抵消业务部门不时提出的变更导致的项目进度的风险。当然在开发模式上，也会衡量敏捷的开发模式（特别是scrum的管理模式）和传统瀑布及衍生模式哪种模式更高效。当然，理解了金三角，对于前期申请项目预算也是有帮助的，比如，可以和预算部门谈判，如果要砍预算，在时间一定的情况下，就只能减少项目范围，这是我们业务需求部门所不能接受的。这样，就可以使IT项目经理名正言顺地把预算部门和IT部门的矛盾转嫁到预算部门和业务

需求部门去。

当然，最合理的做法应该是向BAT公司看齐，IT部门转变为利润中心，自己管预算、自己有开发团队，那么金三角的三条边就都可以进行调优了。

作者回复：谢谢补充👍

非常有价值的分享！



👍 11



小先生

2019-03-15

从这篇文章，我至少学会了不要怼产品。而是要从成本，时间，范围，三原则中寻找妥协。

作者回复：👍

对，重点不是怼，而是协商



👍 9



草裡茵

2019-03-12

金三角的确很有说服力，理论级的支撑。

- 1 用户体验很差的软件。性能低，学习成本高，缺陷多。
- 2 开发成本很高。但是投入产出比会有一个边界，无限的砸钱也不可能等比提高效率与质量。
- 3 开发周期长。开发周期长，比较容易出现需求变动，错失良机等问题。
- 4 不存在。
- 5 没什么能稳定使用的功能，会牺牲用户体验（比如加广告），性能低，安全容易出纰漏。
- 6 需求面很窄，小而美的那类软件，个人情怀很到位，开发时间很长。
- 7 也不存在。

作者回复：👍

我也认同4，7不存在



👍 7



AICC

2019-03-12

备注：理论上快，好，便宜是不存在的，但实际能否存在呢，可以只能加进一个新的维度“少”所以围绕多，快，好，省（对应少，慢，差，贵）来确定1-7会是什么样的软件

1.快 + 便宜 = 质量很差并且多数伴随功能少

2.快 + 好 = 费用成本高多半是早期软件，功能少只实现核心功能但体验不错

3.好 + 便宜 = 开发周期长，但开发时间长费用成本不见得少，所以多半也伴随功能少

4. 快 + 便宜 + 好 = 如备注所说，这会是一个体验不错但功能少的软件，比如微信早期版本只能语音

5.免费 + 快 = 这样的软件也很多，质量基本没谱了可能还有安全风险

6.免费 + 好 = 这样的软件也很多，比如某零杀毒软件，当年就是以免费出道，但成本就很高开发周期一般也比较长

7.免费 + 好 + 快 = 单点功能的软件，比如微信里像抽奖助手的小程序，一个功能好用简单，还有像一些优秀的开源软件插件啥的

但通常免费多数是赚的流量费，广告插入到处是

作者回复：7的话可能还是有点争议，整体总结的非常好👍



👍 6



alva_xu

2019-03-13

留言中讲的买工具，提供培训，提供公共组件服务，以及搭建CICD平台和自动化测试平台，也可以理解为增加成本。但对于我们企业来说，这笔成本就不算在具体的某个开发项目里了，所以我把它归结为提高效率，实际上也可以理解为TTM（Time To Marketing）指标。

作者回复：谢谢补充💖



👍 4



少盐

2020-03-11

金三角，时间、成本、范围，在这三个因素的约束下，取得最好的结果

我想学到更多的知识，只能在时间和成本上都下功夫，如果没有足够的时间保证，成本再高也是浪费

作者回复：学知识和做项目还是有所不同，学知识，不仅仅是时间和成本，还需要通过应用知识，把知识变成技能，最终才能掌握这些知识。

就像软件工程，无论你花了多少时间成本去学习，如果没有做过真正的软件项目，没有把软件项目中的事例和知识点进行关联，没有在项目中的应用这些知识，还是无法掌握的。

共 2 条评论 >

👍 3



dancer

2019-03-18

很多的游戏公司这三个角都想要，就导致抄袭很严重！

作者回复：抄袭我觉得就相当于节约了需求分析的时间和成本

共 2 条评论 >

👍 3



Felix

2019-03-15

谢谢宝玉老师教的砍需求大法，而且还有理论支撑，以后PK再也不虚了🐵

作者回复：只要是为了“质量”，就没什么好虚的👍

另外沟通的时候，还是以协商为主，毕竟大家目标是一样的，是为了保障质量的前提一起协商做一些调整，并不是要PK个输赢：)



👍 3



风再起时

2019-03-14

老师请教一下，敏捷开发中是怎样做成本估算的？按您所说，开发过程中每个Ticket由开发团队集体估算，但在项目开始前即立项阶段，只有初步的大概的需求，具体的Ticket还没有制定出来或者不完备，如何准确确定项目的合同金额或预算？瀑布模型开发又是如何预估成本的呢？都是采用专家估算或领导拍脑袋估算成本和工期吗？软件工程书籍中介绍的估算模型如COCOMO或其他算法模型在实践中有人用吗？功能点、用例点等软件规模的估算方法实践中还有人用吗？这些方法是不是过时了？您觉得CMM或CMMI中对开发过程中的很多方面进行量化度量的做法在当今时代的软件开发中还实用吗？一口气问了这么多问题，先谢啦😊

作者回复：首先，软件项目的成本，主要是人力成本、采购的软硬件成本以及其他办公运营等成本。

其中人力成本的估算其实就是：人员工资x时间

无论是瀑布模型还是敏捷开发，要估算人力成本，就是要算出来要多少人月

在原始需求出来后，要对需求进行分解，越细致越好，然后对于分解后的细项再凭经验估算。

基于细分的需求：

瀑布模型的话，要把阶段分出来，各个阶段预计的时间和人员投入估算出来。然后就能大致算出成本。

敏捷开发的话，要估算需要多少个Sprint，多少人参与这个项目。也能大致算出成本。

你说的“软件工程书籍中介绍的估算模型如COCOMO或其他算法模型”我不知道你指的哪本书，我也确实不太清楚这部分知识，也不知道是不是过时了。

> CMM或CMMI中对开发过程中的很多方面进行量化度量的做法在当今时代的软件开发中还实用吗？

还是看项目场景吧，如果是应用CMMI模型的软件项目，肯定是适用的。我觉得现在软件开发整体上还是在往“敏捷”上发展的。



👍 3



Bradley_Cai

2019-03-13

最怕的就是时间和范围固定，要求加成本的情况。因为在国内这个加成本往往就是加班，而且还没有加班费。

作者回复：加班就是加时间👤



👍 2



纯洁的憎恶

2019-03-12

“多、快、好、省”，软件工程的四难选择问题。由于质量是软件工程压倒一切的要素，因此“好”必须留在“盘子”里。剩下的要素都是可以根据具体情况权衡取舍的。四难选择变成了三难选择。于是，工程师在实践中面对不确定时，也能够有底气做到“不抵触，讲条件”了。

延长时间的另一面是提高效率。借助工具、优化流程、节约资源等方法，可以在一定程度上“冲销掉”延长的时间。

非常欣赏MVP模型，既可以快速见效，又降低了大量返工的可能。在瀑布模型中，通过会有

过度设计的现象。一开始想了很多，结果发现恨不能80%都是瞎想。先拿出核心功能，再根据用户使用的情况，有指向性的完善，步步迭代演进，十分靠谱。唯一令人担忧的是，在外包模式中，如果没有明确的需求，就难以估算出较为准确、合理的预算，进而无法立项、采购。如果先做一版需求申请下来预算再说，再用MVP模型步步试探。那么最后做出来的东西可能与需求文档严重不一致，存在较大的审计、内控风险。也许企业大了、规矩多了，做起事来确实别扭。

作者回复: 总结的非常棒👍

迭代模型和MVP是非常好的组合，因为迭代的时候，会优先选取最重要的功能，慢慢的那些不重要的功能甚至永远不会被加入迭代中，就因为不需要浪费时间在上面了



👍 2



BibuYing

2019-03-12

终于找到一套可以说服老板的说辞了

作者回复: 不要想着说服老板，而是给他更多选择😂



👍 2



不靠谱的琴谱

2019-08-25

遇到那种我全都要，不讲道理的老板咋整。一句话做不了滚蛋，要强制加班解决。

作者回复: 也许换个懂软件工程或者愿意学软件工程的老板也不是坏事：)



👍 1



沱长

2019-06-30

如果时间、成本、范围等比例缩小，三边围成的面积也缩小了，是否意味着软件的质量也是缩小了？但现实中一般认为质量并没有缩小。

作者回复: 金三角反应的更多是各个因素之间的关系和约束，并不是严格的比例限定，否则2个孕妇能5个月生孩子了



果然如此

2019-03-21

这篇把零散的知识点形成体系，强化了时间、成本、范围、质量的关系，更能联系实践！能够提高与产品经理打太极的水准！

作者回复: 大家目标都是为了项目质量，至于如何平衡还是要多沟通多协商，互相妥协：)



一路向北

2019-03-13

通过金三角这个概念一分析，明白了瀑布型和敏捷的你中有我，我中有你。按照时间切片，切完片之后再再来一个微型瀑布。

作者回复: 敏捷的Sprint严格来说不是一个微型瀑布，迭代模型才是。

请参考“05|敏捷开发到底想要解决什么样的问题”，里面有对这个进行对比和解释。

共 2 条评论 >



张驰

2019-03-13

金三角模式很实用，不过对于每个sprint是否可以按照需求来划分呢，因为如果是一个成熟的产品可能规定的时间内都不会有迭代的需求，而是按照产品经理某一天提出来的story进行迭代的。

作者回复: 敏捷开发的Sprint，如果按照需求来划分，而不是按照固定时间，那么最终发布的时间就很难确定，这个Sprint就会变回小瀑布的，就很难敏捷起来。

增量模型是按功能划分的。

