

41 | 为什么程序员的业余项目大多都死了？

2019-06-06 宝玉

软件工程之美

[进入课程 >](#)



讲述：宝玉

时长 13:56 大小 12.76M



你好，我是宝玉。对于不会写程序的人来说，想做一个软件项目，就得找程序员帮忙写程序。而对于程序员来说，想做一个软件项目，写程序不是问题，产品设计自己也能做一点。所以对于很多程序员来说，一旦有了一个想法，可能会利用工作之外的时间，做点自己的业余项目（也叫 Side Project）。

然而大部分项目，都是怀着美好的期望开始，结果做到一半就无疾而终，就算少数坚持到了上线发布，最终还是因为少人问津而不得不放弃。

所以今天将带你一起分析一下，为什么程序员的业余项目大多都死了？怎么样可以借助学习到的软件工程知识提升业余项目成功的概率？

为什么程序员的业余项目大多都死了？

作为程序员，我也很热衷于做业余项目，周围的程序员朋友们也有不少做业余项目的案例，通过对这些案例的观察和分析，我觉得程序员做的业余项目，主要死于以下这些情况。

1. 想法大，时间少

我有个朋友，前一段突然有了一个想法，想做一个类似于 Excel 的基于网页的在线电子表格程序，这是个很大的想法，毕竟微软和谷歌都是有一个团队在完成这样的项目。

但他觉得如果只是实现最核心功能还是可行的，于是激情满满地找资料，写原型代码。然而现实还是很残酷的，他上班就忙，经常加班，下班还要带娃，留给自己的时间其实不算多，一段时间看不到成果后，慢慢的激情就消逝了，这个项目也就不了了之了，而现在他已经又在尝试其他项目了。

这是一个常见的现象，很多程序员在业余做项目开始之前激情满满，经过一段时间没有进展，没有正向反馈，很容易就激情消逝，不想再继续了。尤其是一段时间后，可能又有新的项目想法了，于是就又开始了一个新的循环。

2. 过于追求技术，缺少约束

在公司的项目中，我还是比较保守的，毕竟要受限于项目的成本、时间和范围的限制，而且有 Dead Line 的强约束，所以不会太激进，能稳定上线运行是第一位的。而我一旦去做业余项目，就会陷入过于追求技术的困境。

我前年的时候，想对自己的一个网站进行升级，如果用传统的熟悉的技术方案应该不需要太长时间，但我想体验当时比较新的 React Universal 技术，后端 API 要使用当时时髦的 GraphQL，要考虑多数据库的支持，还要用上 WebSocket 保持内容及时更新。结果有些知识还需要边学边用，虽然在学习这些知识的时候收获不小，但是项目进度确是惨不忍睹，到今天还只有一点雏形。

程序员的业余项目，因为缺少成本、时间和范围的限制，没有设置 Dead Line 约束，所以经常会天马行空，只为了追求技术上的兴奋点，恨不得把新酷技术都用上。

但如果看看项目的定义：

项目是指一系列独特的、复杂的并相互关联的活动，这些活动有着一个明确的目标或目的，必须在特定的时间、预算、资源限定内，依据规范完成。

你就可以发现，项目是要有目标要有约束的。一个缺少目标和约束的项目，是难以成功的。

3. 缺少产品能力和运营能力

有一些程序员，是有做产品的梦想的，希望能打造一款好的产品，解决工作生活中的一些问题，或者就是想通过做产品去赚点钱。其实这样的不乏成功案例，比如像业余时间打造出知名的 iOS 应用 Pin 和 JSBox 的[钟颖](#)，还有像[图拉鼎](#)这样的人，他们先是程序员，后来因为有了成功的产品而做独立开发者的例子。

但这样的例子并不多，普遍存在的情况是，当程序员们真正要去打造产品的时候，却发现要做一个产品并不是那么容易的事情，缺少产品能力就无法设计出好的产品，缺少运营能力就算产品做出来也鲜有人问津。而那些真正成功的独立开发者，无不是能兼顾产品设计能力和产品运营能力，既能设计出真正解决用户需求的产品，又能通过一定的运营让用户了解产品，为之买单付钱的人。

怎样提升业余项目成功的概率？

想法大，时间少；过于追求技术，缺少约束；缺少产品能力和运营能力。这几点是程序员业余项目失败的主要原因。如果要应用软件工程知识来寻找这些问题的解决方案，你会想到什么方案呢？

1. 怎么样让项目不至于半途而废？

想法大，时间少怎么办？怎么样让项目不至于半途而废呢？

回想一下专栏文章《[08 | 怎样平衡软件质量与时间成本范围的关系？](#)》中的金三角理论，在这一篇中，我解释了如何去平衡软件质量与时间成本范围的关系，今后你会发现项目中很多问题都能应用到金三角的理论。

比如说，想法大，其实就是范围大，按照金三角的理论，你要去固定一条边或者两条边，然后去调整剩下的边。

对于业余项目来讲，其实时间是很难去控制的，也就是你不可能像每天上班那样投入大量的时间在上面，所以这条边要被固定起来。

然后看成本这条边，虽然理论上来说你可以花钱请人帮你，也可以花钱买成熟的商业组件，但作为一个业余项目，一般来说前期不会投入大成本的。你也可以假定它是固定的。

那么最适合调整的边就是范围这条边，毕竟作为一个业余项目，你可以先实现最核心的功能。可以采用 MVP (minimum viable product , 最小化的可行性产品) 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。

前不久一个朋友做了一款播客的应用，他就是采用的 MVP 的开发模式，先快速发布了一个只有核心功能的版本，甚至还很多 Bug。发布后邀请了几个朋友试用，收集了反馈，并且也把发现的 Bug 修复了，再逐步增加新功能。这样几个迭代后，他的 App 已经登上了新闻分类的排行榜。如果一开始他就想的是要做一个很大的项目，也许到现在还在开发中呢。

即使程序员做的是业余项目，还有必要补充的一点就是：**在决定做什么项目之前，一样要充分考虑项目的可行性研究**。关于这一点，你可以参考专栏文章《[09 | 可行性研究：一个从一开始就注定失败的跨平台项目](#)》去从经济可行性、技术可行性还有社会可行性方面，去分析一下项目是不是真的可行，再动手不迟。

2. 怎么避免陷入过于追求技术，项目难以交付的困境？

过于追求技术，缺少约束是程序员业余项目失败的另一个主要原因。

程序员追求技术是天性，这一点其实也不是坏事，重点是要有所约束，毫无约束的结果就是迷失在技术中，而忘记了项目的整体。

这其实也是我在专栏一开始就写的《[02 | 工程思维：把每件事都当作一个项目来推进](#)》中提到的，要把业余项目也当作一个正式的项目，做你的业余项目时，也要站在项目的整体去思考项目的进展，而不是沉迷于局部的技术实现。

所以你有业余项目的话，也要像专栏文章《[11 | 项目计划：代码未动，计划先行](#)》中提到的那样，去做项目计划，去设置里程碑。还要敢于把计划和里程碑分享给你的家人和朋友们，公开的做出里程碑的承诺，让他们帮助监督你的计划执行。

当你有了一个可行的计划，有了真正的 Dead Line，你的项目交付就有了基本的保障。

我前些年运营过网站，一个针对我的母校西北工业大学校友们的论坛网站叫开放实验室，我需要负责这个网站的日常运营和程序开发，所以每次升级之前，我都会在论坛发帖子公布我的升级计划，设定一个上线时间，这样网站的用户会监督我的项目进度。有了进度的压力，就会逼着我必须按时完成，而不是老想着用什么新酷的技术。

在你的业余项目难以交付的时候，记住一句话：**Dead Line 就是第一生产力。**

3. 怎么弥补你的短板？

产品能力和运营能力是大部分程序员的短板。

关于产品设计，我们专栏需求分析篇的几篇文章可以参考。《[17 | 需求分析到底要分析什么？怎么分析？](#)》可以帮助你分析用户真正的需求是什么，从而让你可以做出用户想要的，而不是自己凭空想象出来的用户需求。

产品能力的锻炼不是一朝一夕就能炼成的，但日常多模仿，多实践，还是能做出不错的产品设计，我在专栏文章《[19 | 作为程序员，你应该有产品意识](#)》中如何培养产品能力上，给出了一些具体的建议。

比如说你可以从解决自己的需求，解决家人朋友的需求开始，设定一个小的产品目标，然后借鉴类似的产品，模仿它们的产品设计、交互设计，就能做出一个基本可用的产品。

像 UI 设计，其实现在无论是网站的 UI 设计还是 App 的 UI 设计，都趋向于标准化，对于一个业余项目，使用一些标准模板，或者花点钱购买一套漂亮的界面模板，都是不错的选择。

对于产品的运营，这一点很遗憾，软件工程重点是讲如何做项目的，并没有太多运营相关的知识。我个人的一点经验就是，如果你要运营一款产品，你需要想清楚以下几个问题：

想清楚你的产品能给用户带来什么样的价值？帮助用户解决什么问题？

商业模式是什么？也就是用户是不是会为你的产品付钱？或者你的产品通过什么方式赚钱？

如何让用户知道你的产品？如何让用户知道你产品所能带来的价值？

只有想清楚了你的产品的核心价值是什么，才好去针对性的运营你的产品。具体产品的运营上，可以找你的朋友作为第一批用户，然后去像[Product Hunt](#)这样的网站发帖子自荐，还可以通过微博、Twitter 这样的社交媒体宣传。

除了自己去学习产品知识和运营知识之外，其实还有一种方式，就是组建一个小团队，找到志同道合的人一起，你写程序，有人做产品设计，有人负责运营推广，大家取长补短，一起把产品做好！

总结

今天带你一起分析了程序员的业余项目失败的原因。想法大，时间少；过于追求技术，缺少约束；缺少产品能力和运营能力。这几点是程序员业余项目失败的主要原因。

针对想法大、时间少的问题，可以借助软件项目金三角的理论，去缩小范围，在做项目时，可以采用 MVP 的开发模式，先实现核心需求，再逐步增加功能。

针对过于追求技术、缺少约束的问题，应该要对你的项目制定计划，设定里程碑，把时间点告诉你的家人和朋友，让他们监督你执行，通过 Dead Line 来保障项目的进度。

针对缺少产品能力和运营能力的问题，需要有针对性地去学习相关知识，也可以去组建小团队，弥补这些方面能力的不足。

最后，即使程序员们的业余项目很可能会是以失败告终，我做过很多失败的业余项目，但我还是强烈的建议你多尝试做一做业余项目。因为做业余项目，即使项目失败了，一样可以让你收获很多：

通过业余项目，你可以学习和使用工作中不会使用的技术。你工作中做后端开发，你业余项目完全可以体验 iOS App 开发。

通过业余项目，你有机会去按照自己的想法去实现。很多时候在工作中，因为你无法去做决策，无法改变架构的设计或产品的设计，而在自己的业余项目中，你可以完全按照自己的想法去尝试，去证明自己。

通过业余项目，可以锻炼你的大局观和工程思维。当你真的去自己负责一个项目时，就会更多地去站在项目的整体去思考一个项目，而不是局限于专业领域。

通过业余项目，帮助你更好地在项目中沟通。在做过业余项目后，在工作中，和产品经理、测试沟通，你会更懂他们，因为他们的工作你也体验过了，你会体会到他们的工作其

实不像你最初想的那么容易。

课后思考

你有做过业余项目吗？你有经历过哪些成功的和失败的业余项目经历？你从中收获的经验教训是什么？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。



软件工程之美

重新理解软件工程

宝玉
Groupon 资深工程师
微软最有价值专家



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 40 | 最佳实践：小团队如何应用软件工程？

下一篇 42 | 反面案例：盘点那些失败的软件项目

精选留言 (6)

 写留言



yellowclo...
2019-06-06

 2

宝玉老师您好，对于一个程序员来说，新技术和新框架的诱惑是非常大的，一般也想在业

余项目中采用以下新技术去试试水，然后您也提到了就算做业余项目也最好建议采用成熟、比较熟悉的技术。那如何权衡业余项目具体采用的技术以及框架呢，采用很熟悉的技术有感觉没有得到锻炼，采用新技术风险又比较大。

展开 ▾

作者回复: 啊哈，有一点我没讲清楚，就是你的业余项目的目标是什么？

如果你的目标就是锻炼技术，而不是为了做一个有用的产品，那么就用你喜欢的技术就好了：)

◀ ▶



kirogiyi

2019-06-10

👍 1

人有一种惰性，想起来容易，做起来难，坚持起来更难。回想这么多年与自己一起共事过的程序员同事，在和他们的交谈中，发现他们很多想法都是切实可行的，新颖而奇特，如果按部就班实施起来的话，也是一些相当不错的项目，但最后没有几个人能真正的坚持做好做完，理由是没资源、没时间、耗不起，半途而废成了家常便饭。

展开 ▾

作者回复: 是的，知易行难！我就算懂那么多道理，自己也一样有很多失败的项目 🙄♂

◀ ▶



智超

2019-06-10

👍 1

说到开放实验室，是时候亮出校友的身份了

作者回复: 🍷 谢谢校友支持，我打算过一段升级一下开放实验室 🙄♂

◀ ▶



纯洁的憎恶

2019-06-09

👍 1

抛弃妄念，脚踏实地。切忌追求过于宏大的目标、过于新奇的技术，而最终难以落地。做事要有边界和约束，向死而生才有效率。专业短板可以尝试自行补齐，也可以求助他人取长补短。

作者回复: 🍷 感谢分享补充！

◀ ▶



hua168

2019-06-07

👍 1

我有两三个同事，用业余时间兼职做程序开发。
还要自己去找美工，有时候赶工去找前端。
一个项目才赚那么几千块钱。
我喜欢快钱，直接帮别人解决服务器故障，解决完就给钱。
优点是来钱快，缺点就是收入不稳定。 ...

展开 ▾

作者回复: 业余项目不一定要接私活，可以做一些也许短期不赚钱，但是长期对你有价值的事情。
比如说通过做一个开源项目提升能力和影响力，虽然不赚钱，但长远看收获很大。



果然如此

2019-06-06

👍 1

我的微信小游戏业务项目：方块游戏世界，包含俄罗斯方块的经典玩法和拖放玩法，欢迎试玩！呵呵

展开 ▾

作者回复: 🙏 谢谢分享