

42 | 反面案例：盘点那些失败的软件项目

2019-06-08 宝玉 来自北京

《软件工程之美》



你好，我是宝玉。我想你日常一定看到过很多项目失败的案例，有些失败项目的案例甚至超出我们的想象，比如说我的朋友圈就被两个项目刷过屏，一个是号称史上最烂的开发项目，开发 12 年，六百万行代码；一个是美国联邦调查局的一个软件项目，花了 1.7 亿美元，最后变成了豆腐渣工程。

也许大多数人看完这类文章后，会当作一个有趣的故事，觉得他们软件工程水平太差了，居然会把项目做成这样。当你学习完软件工程知识后，再看到这些项目失败的案例，不妨从软件工程的角度来分析一下，这些项目失败的真正原因是什么？你能从中获得什么启发？

什么样的软件项目算是失败的项目？

如果我们说一个项目是失败的项目，那么怎么算是一个失败的项目呢？

项目管理协会（PMI）认为成功的项目必须满足六个条件：

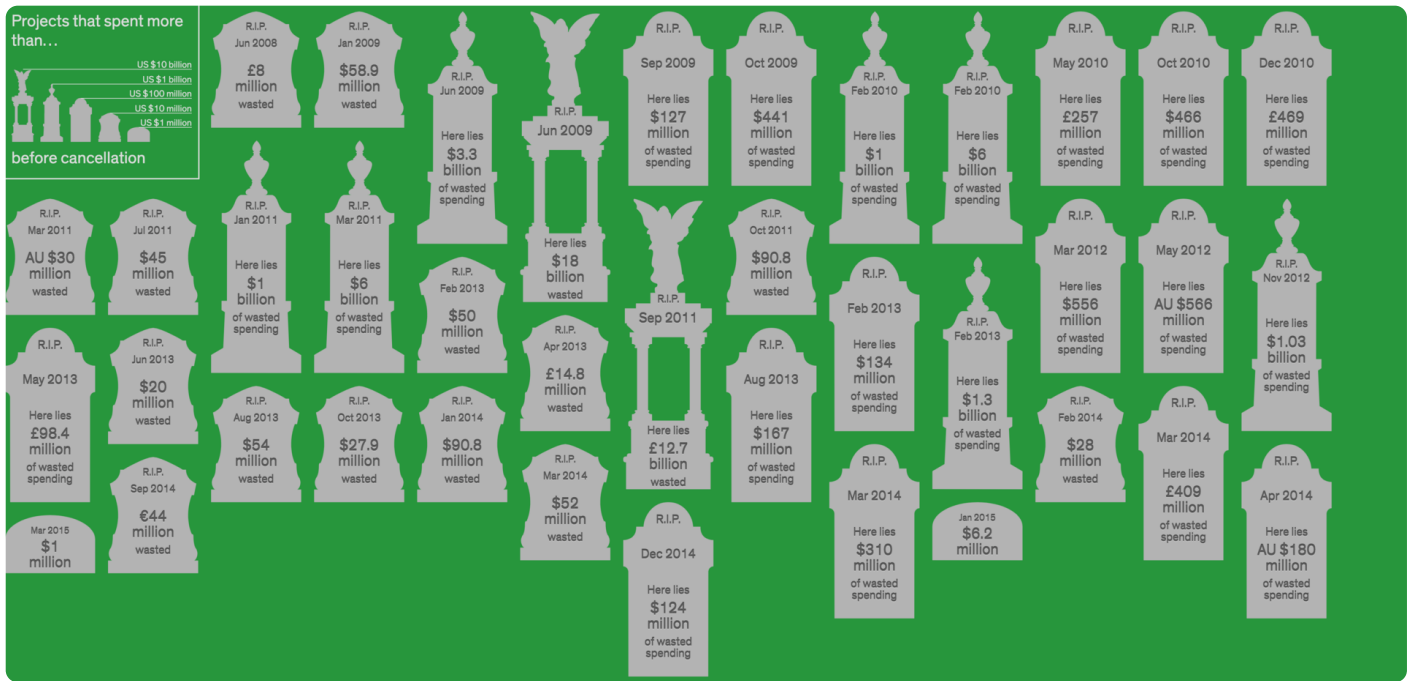
1. 按时交付。
2. 成本在预算范围内。
3. 能按照当初的设计正常运行。
4. 有人使用。
5. 满足项目最初的目标。
6. 项目出资方对项目满意。

相应的，如果上面有一个或者多个条件没有满足，那么项目就有可能是失败的，比如说：

1. 没能按时交付。
2. 成本超出预算。
3. Bug 太多，无法按照当初的设计正常运行。
4. 产品没有得到市场认可，没有人使用。
5. 产品偏移了最初的目标。
6. 项目出资方不满意。

而那些特别失败的项目，往往是多个条件甚至所有条件都不能满足，并且时间、成本、交付结果跟最初目标都相差很大，无疑都造成了巨大的损失。

IEEE（电气和电子工程师协会）有一个专门的网页，把过去十年间，那些著名的失败软件项目，做了一个墓碑来展示，墓碑里的这些项目加起来的损失大约 700 亿美元。Wikipedia 上也有一个网页（[🔗 List of failed and overbudget custom software projects](#)）列出来那些损失严重的软件项目，也是惊人的数字。



(图片来源: [Monument to Failure](#))

而这些软件项目的失败，很大程度上是可以预测和避免的。如果把问题简简单单归结为软件工程水平太差了，或者是项目实施者的水平太差了，那么我们就无法真正的从这些失败中吸取教训，在下次还会再犯同样的错误。

分析失败软件项目的原因

在航空业，如果一架飞机坠毁，会有专业的调查小组去对飞机失事原因进行详细调查，比如分析说当时的天气情况、飞机的维护记录、飞行员的性格特点，平时受到的培训是怎样的，航空公司的文化，对安全的重视程度等等，从而找到事故的根源，并且提出相应的改进方案，避免类似的灾难再次发生。

软件项目其实也是类似的，对于一个失败的软件项目案例，要去分析：外部环境、技术管理、项目管理和组织文化，这样才能帮助你找到项目失败的根源。

外部环境

在调查员去调查飞机失事原因的时候，首先会看的是不是外部环境导致的，例如恶劣的天气环境。分析软件项目失败原因，也可以首先看看外部环境。

如果你去看看历史上那些有名的失败的项目案例，其中政府主导的项目占大多数，而且通常主要因素不是成本，而是各种政治因素导致的不切实际的项目进度，或者是频繁变更的需求，从而严重的影响了成本和质量。

而对于商业软件项目，很多是由于缩减成本导致的。因为商业竞争的大环境，企业为了节约成本，总是希望用更少的人做更多的事情。

还有一些常见的场景就是在一个项目开始之前，销售为了拿下项目，通常会过度夸大项目的成果，而又会相应的压缩项目预算、时间，并且也可能低估了技术实现的难度，最终项目要开发的时候，开发人员才发现根本无法如期完成当初承诺的项目目标，最终导致项目失败。

技术管理

在调查飞机失事原因时，调查完外部环境，还要分析是不是飞机本身设计原因导致的，比如前不久的波音 737 MAX 飞机事故，就是因为软件故障导致的。类似的，分析软件项目失败原因，也一样要去分析技术管理上的问题，很多软件项目失败的原因也是技术原因导致的。

比如说在项目中使用了不成熟或不熟悉的技术，最终导致技术不可控，或者浪费大量的时间在技术的学习上。

项目的规模也会导致技术复杂度直线上升，想象一下，做一个普通的个人网站和做一个淘宝这样的网站，复杂度不可同日而语。通常越大的项目，技术越复杂，需要考虑各种软硬件的交互，服务之间的耦合。也就是说，项目规模越大，失败的概率也更大。

项目管理

调查飞机失事，飞行员是重点调查对象，因为飞行员直接决定了飞机是否能安全行驶。对于软件项目来说，项目经理在软件项目中起着至关重要的作用。很多项目失败不是因为外部环境导致的，也不是技术原因，而是因为糟糕的项目管理。

在一个软件项目中，项目经理掌握了资源的分配，还要制定项目的计划，对任务进行分配，组织分工协作，管理风险，项目成员的日常沟通等等。而这些决策通常很难量化，需要基于当时

的情况进行权衡，一旦这些决策出现大的失误，就会导致项目的失败。

组织文化

在飞机失事后，调查人员调查的最后一个领域就是所属航空公司的文化环境，看航空公司是不是足够重视安全。在软件项目中，一个开放、平等、注重沟通协作的团队或组织更容易及早发现和解决问题。

就像文章开头提到的美国联邦调查局的项目，当有雇员指出来项目中的问题，最后的结果竟然是被扫地出门。

当然，我们在分析盘点那些失败的软件项目时，从多个方面去分析，就是为了能找出这些项目失败的根本原因，从而避免类似的错误再次发生。

盘点那些失败的软件项目

接下来，我们来一起盘点几个著名的失败的软件项目，看看这些案例可以给我们的日常开发带来哪些启示。

在分析这些案例时，我会先分别从外部环境、技术管理、项目管理和组织文化这几个方面去分析问题和原因，最后一起总结从这些案例中收获的经验教训。

案例 1. 来自地狱的项目

案例描述：

这个案例来自法国政府，当时参与项目的一名项目成员专门为此项目开了一个博客叫 [ProjectFailures](#)，将这个项目描述为来自地狱的项目。原计划 2-3 年开发，结果干了十几年都没有完成，最终以项目负责人被以欺诈罪关进监狱而告终。详细内容可以查看中文版本：[《开发 12 年 整整 6 百万行代码：史上最烂的开发项目长这样》](#)。

案例分析：

外部环境：法国政府官员腐败，对于项目进度并没有施加压力；

技术管理：没有好的开发实践，完全 C++ 开发，600 万行代码，版本控制一团糟；

项目管理：糟糕的项目管理，团队成员 55 人，35 名经理，20 名开发人员，管理人员比开发人员还多；不断开会，只是展示 PPT；

组织文化：禁止超过 9 点打卡，禁止喝咖啡等奇葩要求。

案例 2. 美国联邦调查局虚拟案件文档系统

案例描述：

FBI（美国联邦调查局）虚拟案件文档系统的项目开始与 2001 年，项目初始目标是 3 年内将原有的 FBI 案件文档管理系统升级，但因为 911 恐怖袭击事件爆发，项目目标从升级变成了重写。最终 2005 年项目宣布废弃，而此时已经在这个项目上花费了 1.7 亿美元。有关项目的细节可以参考：《[著名豆腐渣软件项目：美国联邦调查局虚拟案件文档系统](#)》。

案例分析：

外部环境：FBI 没有真正懂技术的负责人领导和管控项目，对承包商缺少控制；

技术管理：无法解决项目的复杂性，系统在设计上不完整，不充分，不到位，以至于在现实场景中完全无法使用，上线前没有测试；

项目管理：开发方和客户之间沟通不畅；频繁需求变更，项目管理混乱，外行领导内行；

组织文化：指出问题的雇员反而被调查和开除。

案例 3. 微软 Vista 项目

案例描述：

微软的 Windows Vista 项目开始与 2001 年 7 月，预计 2003 年发布。比尔盖茨为 Vista 提出了三大目标：1. 完全使用 C# 提升开发效率；2. 使用数据库作为新的文件系统 WinFS；3. 使用全新的显示技术 Avalon（后来改名为 WPF），打破桌面软件和网站的用户界面界限，提升微软竞争力。

目标非常好，但技术难度非常大，结果三年后也未能开发完成，不得不在 2004 年对目标进行调整：不用 C#、取消 WinFS、删改 Avalon，一开始的三大目标就这样被完全否决，最终 2007 年才发布 Vista。参考文章：《[🔗 五年磨砺: 微软 Vista 开发过程全记录](#)》。

案例分析：

外部环境：在目标的设定上，主要不是为了满足用户需求，而是为了商业上的竞争需要；

技术管理：技术上难度过大，超出团队控制范围，无法完成任务；

项目管理：比尔盖茨对项目直接干预较多，项目周期太长；

组织文化：盖茨制定目标后，核心团队明知困难，却不敢也没有反对，当看到任务无法完成时，他们不再努力工作，只想着如何推卸责任。

通过对这些项目的分析，再结合我们之前学习过的软件工程知识，其实软件工程对这些问题都有方案可以应对。

在设定项目目标的时候，如果真正的将可行性分析落到实处，那么像 Vista 这样的技术不可行的项目目标，也许一开始就可以进行调整，而避免后续更大的损失。

如果在项目开始的时候，有认真的对需求进行分析，和客户有很好的沟通，对于需求的随意变更有管理和控制，那么像 FBI 这样的项目，就有机会做出来满足用户需求的软件项目。

在项目开发之前，如果做了架构设计，做了技术选型，那么像法国政府项目、FBI 项目，也许可以有更简单可行的技术方案，要知道架构设计就是控制技术复杂的最好手段。

在项目开发的时候，如果做好版本控制，持续集成，自动化测试，那么像法国政府项目、FBI 项目，质量上就更有保障，不至于一测试全是问题。

在设置项目周期的时候，如果能缩短版本发布周期，尽快发布第一个版本，那么很多延期本可以避免或者不至于那么严重。想想看法国政府项目花了 12 年，如果他们在第一年内能先发布一个简单的版本，后续再逐步迭代，也许结果会完全不一样。

缩短项目周期也是微软在 Vista 项目上收获的一大教训，在 Vista 之后，微软的项目周期都大幅缩短，而且发布频率也大幅提高，每天都有内部测试版本发布。缩短周期后，可以尽早发布，尽早验证项目的可行性，也让测试可以尽早介入。

在团队的文化上，如果日常营造平等的沟通协作的氛围，让项目成员敢于提出不同的意见，那么像 FBI、Vista 这样的错误也许可以早点被修正。

类似于这样的项目还有很多，比如有一本书叫《[💡梦断代码](#)》，讲述了一堆优秀程序员，一起开发一个大型的开源项目，最终如何走向失败的过程，有兴趣可以看看。邹欣老师对这本书也有非常独到的点评：《[💡梦醒时分 - 梦断代码读后感](#)》

以后你遇到类似的案例，也可以尝试去对它们进行盘点分析，找出它们失败的根本原因，能从中吸取教训，避免类似错误发生。

总结

今天我带你一起学习了如何从软件工程的角度分析失败的软件项目。

通过借鉴航空业对飞机坠毁原因的调查，也可以从四个方面去分析软件项目失败的原因，那就是外部环境、技术管理、项目管理和组织文化。

如果细化一下，还可以总结出一些具体的常见的失败原因：

- 不切实际或者不明确的项目目标；

- 对项目所需要的资源估算不准确；

- 需求不明确或者频繁变更；

- 没有对风险进行有效管理；

- 和客户之间沟通不畅；

- 无法解决项目的复杂性；

- 没有好的开发实践；

糟糕的项目管理；
上层的政治斗争；
商业压力。

其实软件项目失败并不可怕，最重要的还是在失败后，总结原因，吸取教训。就像微软在 Vista 项目失败后，总结经验，改进了开发流程，加快了发布周期，在 Windows 7 项目上重新取得了巨大的成功。还有像暴雪，在泰坦项目失败后，基于泰坦项目开发出了大受欢迎的守望先锋游戏。

课后思考

你有经历过或者听说过印象深刻的失败的软件项目吗？你觉得原因是什么？有哪些经验教训？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。

© 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。

精选留言 (11)



果然如此

2019-06-10

回复纯洁的憎恶：软件工程是过程控制的方法论，而产品设计才是保证伟大的产品，两者应该结合。

作者回复：👍很有道理



👍 12



纯洁的憎恶

2019-06-09

深深地感受到，软件工程不是为了创造最伟大的软件项目而存在，却是为了保障每一个项目的成本、质量、工期、目标等等可控而存在的。

作者回复: 谢谢分享, 帮转发一下@果然如此 的回复, 我也觉得有一定道理, 供参考:)

@果然如此 2019-06-09 19:54

回复纯洁的憎恶: 软件工程是过程控制的方法论, 而产品设计才是保证伟大的产品, 两者应该结合。



👍 5



大王叫我来巡山

2019-08-07

政府单位干黄的项目, 基本都是人祸, 没有之一, 决策者敢想, 企业敢吹, 出了问题敢捂, 一层一层敢骗。



👍 3



freda

2020-07-03

对于多变, 决策难产的领导怎么应对, 目前应对就是暂缓2-3天执行。

作者回复: 遇到这种领导很难应对, 决策难产还好, 你可以帮助做一些方案, 让他选择一个就好! 最怕的就是多变, 今天刚做好的决策, 你还没开始设计开发, 他那边明天又变掉了, 这种什么软件工程理论都没用!

关键的是, 你要有东西来“约束”领导的“多变”, 这种约束可以是领导的领导、可以是流程、可以是漂亮的PM妹子, 但你不解决如何约束多变的问题, 后面的软件工程理论是无法开展的。

共 2 条评论 >

👍 1



Harold

2020-03-30

互联网项目大多数是失败的。六个原因中, 我个人觉得最多的是: 产品没有得到市场认可, 没有人使用。

作者回复: 项目开发和产品推广其实算是两个领域, 软件工程更多还是关注软件项目开发, 至于产品如何得到市场认可, 这可能超出了软件工程范畴:)



👍 1





ifelse
2022-07-09

其实软件项目失败并不可怕，最重要的还是在失败后，总结原因，吸取教训。--记下来



aoe
2022-02-16

暗黑3为了避免游戏币崩溃，上线不久后就推出了拍卖系统，起初主要以游戏币交易装备。结果进一步加剧了外挂刷钱，金币瞬间通货膨胀，拍卖系统匆匆关闭。不要低估玩家对装备的渴望。



williamcai
2020-08-26

开始与客户确定好了需求，快要发布了，客户突然要修改，前前后后不下10次，导致了2个月的工期硬生生的搞成了5个月，还要后期维护的成本也大大增加，总体



mithril
2020-03-27

老师的案例让我想起了美国现在的联合攻击战斗机项目，简直都是一个模子里刻出来的问题

作者回复: 特地去Google了一下你说的这个项目，不过没有找到好的分析文章，如果有机会，也欢迎分享：)



test
2020-01-01

老师，作为一个普通非管理的技术人员，在项目管理不科学的情况下，如何提高自己的软件工程水平和对项目产生积极的贡献呢？

作者回复: 作为普通程序员，看起来对项目影响有限，但是实际上一样还是可以利用所学软件工程知识做一些事情。

首先是做好自己的事情

作为程序员，每天主要工作都是写代码，但是怎么写其实很有讲究的。比如说：

- 你在一个任务开始之前，是不是会做设计？哪怕简单的设计，设计做好了，会不会找同事评点你的设计？
- 写代码的时候，代码质量是不是够简洁明了？对需求或架构有疑问是不是能及时去沟通确认
- 写完代码有没有自己测试？有没有写自动化测试，保证自动化测试代码的覆盖率？
- 上线后，有没有观测上线的后的错误日志和数据，收集用户的反馈？
- 对于技术债务，有没有去尝试优化解决

如果能把自己的事情做好，那么你已经对项目产生了很大的积极贡献，同时你也逐步建立了自己在团队中的影响力，让其他成员看到，软件工程可以帮助你完成任务，把代码写好。

然后是用好工具

工具在软件工程中作用很大，同时引入成本也相对低，不需要领导审批，也不需要申请经费。比如说：

- 项目管理工具，如果你们的日常任务都是口头说明的，那么可以借助任务管理跟踪工具，像Jira、禅道等，把日常任务跟踪管理起来，哪怕是你自己的、或者自己小组的任务。这样日常要做哪些事情，时间点，进度，可以一目了然。当你手头已经有很多任务了，产品经理还想给你塞需求，你就把项目管理工具上的任务给他看，让他清楚你当前已经没法做更多的事情了，除非等手头的事情忙完，或者推迟手头的事情。
- 源代码管理工具，git这种源码管理工具现在已经是标配了，如果没有用当然应该赶紧用起来，用了也可以借鉴一些成熟的开发流程，例如Github Flow（[O网页链接](#)）
- CI/CD，持续集成持续部署似乎还不够普及，如果没有的话，要考虑做起来，花一点时间，把CI/CD环境搭建起来，用起来。用CI/CD结合开发流程，降低测试和部署的成本，降低代码集成可能产生的问题风险，把自动化测试覆盖搞上去，让QA可以及时测试到最新代码。

用好工具，可以提升开发效率提高项目质量，如果只是自己负责的项目用，相对阻力要小，可以从小的见效快的事情做起，比如自动化部署、自动化测试这些，然后再逐步扩大范围。

最后就是影响更多的人

要想让软件工程在项目中发挥大的作用，仅仅自己用是不够的，需要整个项目都用起来，都用好，才能最大化的发挥作用。所以终极目标还是要通过前面做的这些事情，改变大家的观念，引起大家的重视，让更多的人用起来。

从小事情做起，先在小项目作出试点，让团队成员知道怎么用，用了有什么积极的效果。这就像当年改革开放时的小岗村，几个人先做起来，作出成绩，然后就可以更大范围试点，更大范围推广。





纯洁的憎恶

2019-06-10

谢谢！细想还真是这么回事！

