

11 | 项目计划：代码未动，计划先行

2019-03-21 宝玉

软件工程之美

[进入课程 >](#)



讲述：宝玉

时长 15:41 大小 14.37M



你好，我是宝玉，我今天想与你聊一聊“项目计划”的问题。

若干年前，我接手一个陷入困境的项目，当时的项目经理刚从技术高手转型项目管理，还是没有摆脱技术思维，项目没有什么计划。

他把关键模块分给了自己开发，同时还要兼顾项目管理，导致自己的工作遇到瓶颈，其他人的进度也受影响，大家加班加点也没什么进展，士气低落。

我接手后，第一件事是重新制定项目计划，在排任务时，避免了对某个人的过度依赖，设置了几个关键里程碑。我还特地把第一个里程碑设置的相对容易一点，只需要运行核心功能。

这样大家重整旗鼓，很快就完成了第一个里程碑。达到第一个里程碑的目标后，团队成员很受鼓舞，士气很快就上来了，后面按照新的计划，并没有太多加班加点，就完成了一个个的里程碑，最后顺利完成项目。

你看，**如果没有计划，你的项目可能会陷入一种无序和混乱中。**

计划，就像我们出行用的导航，你可以清楚地看到项目整体的安排，同时它还时刻提醒我们目标是什么，不要偏离方向。

执行计划的项目成员，就像使用导航的司机，可以知道什么时间做什么事情，保证任务得以执行。执行计划的过程，就像我们沿着导航前进，可以了解是不是项目过程中出现了偏差，及时的调整。

做技术的就不用关心计划吗？

很多程序员对计划有误解，也不愿意做计划，他们通常都会用一些原因来说做计划是没必要的。

一种典型观点是：“既然计划总是在变，干嘛还要做计划？还不如上手就是干来的爽快！”

这就好比我看过的一个段子：“既然飞机老是晚点，还要时间表干吗？”“没有时间表，你怎么知道飞机晚点了呢？”计划也是这样，给你提供一个基准线，让你知道后面在执行的时候，是不是出现了偏差，可以根据计划不断地修正。

还有的人说，做计划那是项目经理的事，我是程序员，项目计划与我无关。

我在专栏中常说你要有大局观，不要将自己局限在程序员的身份中。试着做计划就是一个非常好的培养大局观的方式。比如说，你在制定计划的过程中，需要去综合考虑各种因素：有哪些任务要做，可能存在什么风险，任务之间的依赖关系是什么，等等。

参与做计划的过程，可以让你对项目的各种事情了然于胸，这就相当于扩大了你的上下文，让你有更高的视角看待当前工作遇到的问题。

另外，我还见过很多人抱怨项目经理制定的项目计划有问题，却很少看到会有人愿意主动参与制定项目计划。如果你不主动参与计划的制定，最终就只能按照项目经理制定的计划执行了。出现计划不合理的地方，你也只能接受，工作就会一直很被动。

当然，有时候你可能确实是没机会参与到当前的项目计划中。不过，万事皆项目，你一样要学会做计划，因为学会做计划，会对你工作生活的方方面面起到积极的作用。

比如很多人都有一些目标：要转型做管理、要移民、要写一个业余项目，然而很多目标都无疾而终了。这是因为光有目标还不够的，必须得要付诸行动。而要行动，就需要对目标进行分解，进而变成可以执行的计划。

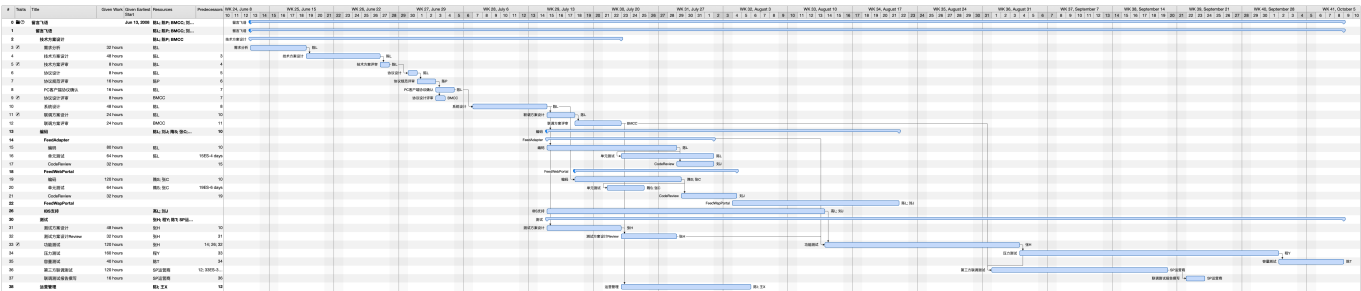
如何制定计划？

如果有一天，你接手了一个项目，通常第一件事就是得去制定一个项目计划。那么怎么制定计划呢？

制定项目计划，通常有三个基本步骤：

- 第一步：任务分解；
- 第二步：估算时间；
- 第三步：排任务路径。

以前我在飞信时，有一个项目叫“留言飞语”，就是飞信用户可以在网站或者 PC 客户端，互相留言，当时我负责这个项目的服务端，正好我还留着当年制定的计划，虽然不算一个很好的计划，但好在它是一个真实项目的计划，正好可以用它来说明一下如何制定计划。



(备注：图片较大，需要点击查看大图)

你看到的这个计划其实不是第一版，可能也不是最后一版，因为制定计划本身是一个反复迭代的过程，尤其是一开始在需求并不够明确的时候，只能比较粗粒度的分解任务和估算，在项目推进的过程中再逐步细化和完善。

第一步：任务分解

我们写程序的时候都有经验，就是要把复杂的问题要拆分成简单的问题，大的模块拆成小的模块，在工程里面这个叫“分而治之”。做计划也是一样，第一步就是要对任务进行分解。

在项目管理中，对任务分解有个专业的词汇叫 WBS，它意思是工作分解结构 (Work Breakdown Structure, WBS)。**就是把要做的事情，按照一个树形结构去组织，逐级分解，分割成小而具体的可交付结果，直到不能再拆分为止。**

下图就是“留言飞语”项目按照 WBS 拆分的结果。

# ▲	Traits	Title	C
0	 	▼ 留言飞语	
1		▼ 留言飞语	
2		▼ 技术方案设计	
3		▷ 需求分析	
4		▷ 技术方案设计	
5		▷ 技术方案评审	
6		▷ 协议设计	
7		▷ 协议规范评审	
8		▷ PC客户端协议确认	
9		▷ 协议设计评审	
10		▷ 系统设计	
11		▷ 联调方案设计	
12		▷ 联调方案评审	
13		▼ 编码	
14		▶ FeedAdapter	
18		▶ FeedWebPortal	
22		▶ FeedWapPortal	
26		▶ IBS支持	
30		▼ 测试	
31		▷ 测试方案设计	

31	▷ 测试方案设计
32	▷ 测试方案设计Review
33 	▷ 功能测试
34	▷ 压力测试
35	▷ 容量测试
36	▷ 第三方联调测试
37	▷ 联调测试报告撰写
38	▶ 运营管理

可以看的出，整个过程是按照瀑布模型来划分的，大的阶段分成技术方案设计、编码和测试，然后每一个大的阶段下面再进一步细分。

例如技术方案设计下面再有需求分析、技术方案设计和评审等；而编码阶段则是按照功能模块再进一步拆分。拆分之后，都是小而具体、可交付结果的任务，且不能再进一步拆分。

这里需要注意的是，在制定计划时，除了要拆分任务，还需要反复思考各种可能存在的问题。

比如，这个项目不仅是网站可以访问，还需要在 PC 客户端能发留言，所以还需要考虑和 PC 客户端的通信协议、什么时间可以让 PC 客户端可以测试协议等。如果上手就写，没有良好的计划，就可能会忽略这些问题，最后导致 PC 客户端都不知道怎么去调用服务端接口，也不知道什么时候可以和客户端联调。

如果项目经理对技术细节不熟悉，可以邀请架构师或者技术负责人协助进行任务的分解。

第二步：估算时间

任务分解完之后，你就需要对每一个任务估算时间。就像下面这样。

任务拆分的越细致，想的越清楚，就能估算的越准确。

要让负责这个任务的人员参与估算。

举例来说，让你直接给出一个“留言飞语”这样项目的估算时间，是很难的，但对于某个具体功能模块的实现，就可以比较准确了。当把“留言飞语”这样大的项目拆分成足够小的任务时，你就可以很容易的对小的任务进行准确的估算，从而让整体的时间估算变得准确起来。

为什么要让开发人员参与估算呢？

我们来对比一下。假如说，一个任务，项目经理估计需要 3 天，但是实际执行的时候，这个任务可能要 5 天，结果导致开发人员加班。这时候开发人员心中肯定会有不满的情绪，认为是项目经理的错误估算导致了他的加班。

如果这个任务所需的时间，是由项目经理和开发人员一起估算出来的，结果最终发现错误估算了任务的难度，这时候开发人员多半会主动加班加点，努力在 3 天之内完成，也不会轻易怪罪到项目经理头上。

但这并不意味着项目经理对估算不需要控制，通常来说，项目经理需要自己有一个估算，然后再请开发人员一起评估。如果结果和自己的估算差不多，那就可以达成一致，如果估算不一致，那怎么办呢？

其实很简单，**就是要双方一起沟通，消除偏差**。特别要注意的是，开发人员预估工作量通常会很乐观，所以最后时间会偏紧，这种情况一样要去沟通消除偏差。估算的主要目的是尽可能得到准确的时间。

但是在沟通中也要注意技巧，不要采用质问的方式：“这么简单一个模块居然要 5 天？”这只会让听者产生逆反心理，无法有效的沟通。可以恰当的提一些问题来达到有效沟通的目的，比如我通常会问两个问题：

“能不能把你这个任务再细化一下？”

“能不能简单介绍一下这个模块你是打算如何实现的？”

估算出现偏差，可能是由于开发人员没想清楚，或者是项目经理自己低估了其难度。**提问可以帮助双方搞清楚真实的情况是什么样的，而且也不会招致反感**。同时项目经理还可以给予

一些建议和支持。

沟通最好的方式就是倾听和恰当的提问。

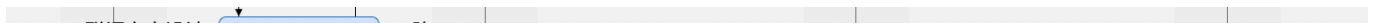
如果任务的粒度太粗，就需要进行细化，细化后就能更准确的知道结果。

对于估算的结果，通常还要考虑增加一些余量，因为实际项目执行过程中，并没办法保证是 100% 投入，有可能并行还有其他事情，或者一些突发事情、事先没有考虑到的任务都有可能影响进度。至于加多少余量，还是要根据项目的情况和经验来判断。

第三步：排路径

我们知道，项目中有些任务是可以并行做的，而有些任务之间则是有依赖关系的。比如说“留言飞语”项目中，编码和测试方案是可以同时进行的，而 Code Review，要在编码完成后进行。

所以，**排路径就是要根据任务之间的关系，资源的占用情况，排出合适的顺序。**例如下图。



排路径是一个相对比较复杂的任务，比如要注意任务的依赖关系，要注意路径的长度，尽可能让几个任务可以并行的进行，避免相互等待。如果借助像 Project 这种工具会让这个过程相对容易些，可以直观的看出来哪些任务是相互依赖的，哪些是同时进行的。没有 MS Project 这类软件，也可以用一些替代手段，例如 Excel 上画表格。

制定计划时不要担心不够准确，先有一个基本的计划，可以粒度比较粗，不那么准确，让事情先推进起来。

设置里程碑

不知道你有没有参加过那种周期很长的项目，一直看不到结果，时间一长会很疲惫。所以有经验的项目经理会在项目启动后，根据制订好的初步计划，确定几个关键的里程碑。

里程碑的时间点确定后，计划可以灵活调整，但里程碑一般不会轻易改变，因为里程碑代表着一份承诺。这对于项目成员来说，有两个重要的影响，一方面，成员会有很明显的来自 DeadLine 的进度压力，自古 DeadLine 就是第一生产力；另一方面，就是在里程碑完成后，大家会获得一种正面激励。

里程碑的设置，并没有特别的规则，可以是项目生命周期的特定主要时间，也可以是一些关键的时间点。拿“留言飞语”这个项目来说，有三个时间点非常关键：

第一个时间点就是确定和 PC 客户端的通信协议，这样 PC 客户端可以根据这个协议开始开发功能了；

第二个时间点就是服务端开发完成，PC 客户端可以服务端联调了；

第三个时间点就是测试验收通过，可以上线了。

最终这三个时间点被定义为里程碑。

在项目的推进过程中，根据里程碑完成的情况，你就可以很直观地知道项目的进展如何。如果发现不能如期完成里程碑，就需要进行适当的调整了，例如加班，或者砍掉一些功能需求。

当然，设置好的里程碑也不是不能调整，但是要注意调整次数不宜过多，不然就会变成“狼来了”，以后就没有人相信你的时间点了。

计划需要跟踪和调整

项目管理中，并不是计划制定好了就完事了，还需要跟踪和调整。就好比你要开车去什么地方，设置好导航还不够，还需要沿着导航前进，如果遇到障碍或者走错路了，得要及时调整。

项目的跟踪是很必要的，可以了解计划的执行情况，了解成员的工作情况，是否能按时完成，需要什么样的帮助。

跟踪进度的方式主要有两种，一种是项目经理定期收集跟踪，一种是项目成员主动汇报。项目经理挨个收集的话，会有一个沟通确认的过程，对进度会了解的更准确；项目成员主动汇报，可以减少项目经理的收集工作，但有可能不准确。

在这方面，我觉得敏捷开发的两个实践特别值得借鉴和推广。

第一个就是每日站立会议，在每天的站立会议上，每个项目成员都需要说一下自己昨天做了什么，明天计划做什么，有没有什么阻碍。通过这种方式，可以非常好的了解每个人的任务进展情况，同时对于成员遇到的困难，其他人也可以及时给予支持。

第二个就是看板，通过看板，可以非常直观的看到每个人在干什么，进展如何。

通过对项目计划的跟踪，可以很容易的看出来执行的情况，也会发现偏差，计划出现偏差是很常见的，所以需要定期进行调整，也不需要太频繁，例如可以每周对计划做一次调整。

总结

项目计划是保障软件项目成功非常重要的手段，制定计划的过程，可以让你对项目有全面的了解，跟踪计划让你知道项目进展情况，出现问题也可以及时调整。

将任务分解、估算时间、排路径，三步就可以制定出一个项目计划，制定计划不要追求完美，制定好一个初步计划后，就可以先按照计划推进起来，进行过程中还可以继续调整细化。设置里程碑可以有效的保证项目的按时交付。

最后，并不需要当项目经理才能去制定计划，生活中每件事都可以当作一个项目，都可以去制定计划来帮助你实现目标。

课后思考

你现在项目的计划制定的如何？如果你是项目经理，你会如何改进？你日常生活中，会有制定计划的习惯吗？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。

上一篇 10 | 如果你想技术转管理，先来试试管好一个项目

下一篇 12 | 流程和规范：红绿灯不是约束，而是用来提高效率

精选留言 (25)

写留言



Winder

2019-03-21

14

宝哥，能讲一下你在极客上开这个课题的计划吗？如何以面向工程的思想完成这个课题 😊

作者回复: 其实你知道吗？在这个项目中，我不是项目经理，我其实是程序员，项目经理是编辑，她定的计划，而且不同意我放出来 😊

而且她还活学活用，基于敏捷的写作模式，一个Sprint一个Sprint反复迭代，你看到的每一个发布版本可能已经迭代修改过好多版了！

#

根据自己的经验写一下：

1. 很多的时候，我们不愿意指定计划的原因，简单的说是“懒”，深层次的是不愿意“思考”，因为这需要做很多准备工作，并消耗很脑细胞，是对自己认知上的一个考验。再往深处挖则是“不愿意承担责任”（工作中由其会遇到类似的同事，我拿多少钱就做多少...

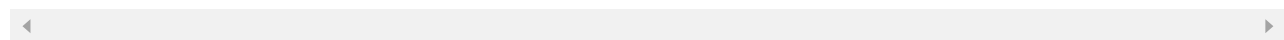
展开 ∨

作者回复: 谢谢分享和补充，讲的特别好，尤其是那句心理欺骗的，说到心坎里面了，我也常犯这样的错误！

建议计划一开始不要太细，先粗一点，定好里程碑，然后进入下一个阶段前细化，细化时拉上参与计划的人一起。这部分建议你看一下我给一路向北的留言回复。

另外个人计划我觉得你已经做的很好了 🍻

我建议你也可以多考虑一些长远的规划，例如五年十年的，然后针对这些大目标设置几个大的里程碑，这样再定细节的计划会更有方向感。



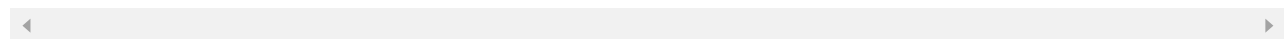
hua168

2019-03-21

👍 6

很感谢老师，很负责，每个问题都认真回答，目前是我购买这么多专栏最负责的一位没有之一，敬业呀 🍻，遇到负责的大牛老师就相当遇到贵人了！

作者回复: 不用客气，有些问题也只是一家之言，你自己的情况我其实了解不多，建议你还是自己多想想，多问问你的朋友和领导们，他们可能能给出更具体的意见。



天之大舒

2019-03-21

👍 3

没计划，怎么知道变化。
没计划，怎么知道进度要求。
没计划，怎么知道范围要求。
没计划，怎么知道质量要求。
没计划，怎么知道资源要求。

展开 ∨

作者回复: 这一段总结的太好了 🍵

hua168

2019-03-21

🍵 3

我是做运维的，17年底开始计划学java开发，没有做过项目，以往习惯每年年底订明年目标计划。

订好计划之后再分月-->周（上班及周末），周末看视频学习，周1-5有时间学练习视频讲的内容。

...

展开 ▾

作者回复: 能理解，迷茫在于对于方向的不明确，以及未来的不确定。

建议在方向上先选准了，想想5年后做什么10年后做什么，而不仅仅是一年后做什么。

如果是5年后还想继续运维，那么想做到什么高度？是不是要去大公司锻炼一下？

如果5年后是想做管理，那么是不是得准备学习一些专业知识，是不是让自己尽早有机会管人？

想清楚未来做什么，给自己设一些里程碑，一步一个脚印，应该就没那么迷茫了。

一点建议，仅供参考：)

小伟

2019-03-23

🍵 2

计划确实很重要，即使变化了，但是有个基准，也能帮助在项目复盘的时候知道问题出在哪儿，下次可以做的更好。

另外，项目排期一定要和其他人达成共识，并有专人去确认进度，有文档记录。

...

展开 ▾

作者回复: 遇到阻塞了要让老板知道，让老板提供帮助，不要一个人耗着，不然难有进展。

另外也可以尝试项目之外寻求帮助，例如搜索引擎、技术论坛、stackoverflow，还有像github上找找同类项目。

tcny

2019-03-22

👍 2

如果因为开发不紧不慢耽误了时间，如何处理呢。应该设置什么样的奖惩制度呢？

作者回复: 这是个好问题！

计划恰恰就是为了预防类似于开发不紧不慢耽误了时间的问题。

具体例子，一个模块，正常估算（开发和PM都认可）需要5天，但是如果你的计划粒度是5天，那么你到最后一天才能知道是不是会延迟，这时候补救已经晚了。

如果你能把粒度设置到半天一天，那么第二或第三天你大概就能知道进度是不是有问题，然后马上作出调整，要么加班，要么找人帮忙，要么换人，要么改计划。这样才可以做到防患未然！

至于奖惩制度，只是手段，而不是目的！

一路向北

2019-03-22

👍 2

以前觉得计划很有用，然后在计划上花了很多时间，最后实际项目也没跟上计划。很大的一个原因是对项目的分解不够，粒度太粗，导致预估的时间和实际的时间差距太大。慢慢的就不做细的计划了，没有计划之后，发现做项目更加时间不可控。因为有了小时时间节点，没有deadline，就没有方向，没了压力。直到大的时间点到了之后，发现离目标差距太大。...

展开 ▾

作者回复: 谢谢分享

计划一定是要有的，不然就完全不可控了

计划一定是个迭代的过程，计划也是个粗到细的过程。

一开始不建议特别细的计划，整体粗一点，定好大的时间节点，也就是里程碑，然后对于下一阶段的计划细化。

细化过程中要拉上具体参与的人一起制定，这样结果才科学也不会导致抵触。

里程碑定了后不要轻易变，不然就失去了DeadLine的意义，即使变也不能过于随意和频繁。



Geek_85f78...

2019-03-22

👍 2

有个疑问，如果是采用敏捷方法的项目，项目计划是否应该就是迭代计划？在这种情况下WBS的结构其实就是一轮接着一轮的规划-分析-编码-测试-集成发布-与敏捷配套的一系列总结？每一轮迭代的成果就是项目的里程碑？

展开 ∨

作者回复: 敏捷的项目计划确实有些不一样，WBS分解后会变成backlog，backlog的项会被打分（参考扑克牌打分），根据分数大致可以算出来需要多少Sprint，因为敏捷开发磨合好后，每个Sprint能做的任务分数大致相当。算出来多少Sprint就能大概知道需要多少时间。

通常里程碑不会那么密集的，一般会几个Sprint一个里程碑。

纯洁的憎恶

2019-03-21

👍 2

因为水平实在有限，且应变能力太差，我做事前一定要做周密的计划。这反而使得我在大多数重要事情上表现不算太糟。我理解计划的意义是把事情的完整过程详细推演几遍，做到进展阶段、关键难点、资源分配、潜在风险、先期准备了然于胸。即使计划赶不上变化，也能应对自如，不会惊慌失措。凡事欲则立不预则废。

...

展开 ∨

作者回复: 你这种性质的单位我确实没经历过，缺少经验。

不过我可以帮你从另一个角度分析下，就是如果我不愿意参与计划可能有这些方面原因：

1. 跟我利益不相关，做了没好处，不做没损失
2. 你已经做的够好够细了，没什么好发挥的
3. 就算参与了提了想法和意见也没用，最后还是项目经理说的算，那我还掺和个啥劲

所以你可以看看能不能让这事变成一个跟大家利益相关的事，跟绩效考评啥的扯上关系，必要的话拉上领导狐假虎威一番，然后拉他们参与时不用太细，让他们有机会参与制定，制定时能平衡好他们利益关系。

尤其是里程碑的确定，我觉得应该是和大部分人利益相关的，至少这个点得让他们参与进去。

一点浅见，供参考



alva_xu

2019-03-21

👍 2

补充一下，对于项目计划，还需要考虑计划发布的有效性，也就是，计划要及时发布到相关干系人。

作者回复: 谢谢补充👍

让所有干系人参与计划，知道计划情况是很重要的。

另外一般不是直接参与项目但利益相关的，比如客户、管理层，他们更关注里程碑



alva_xu

2019-03-21

👍 2

谈一下我对项目计划的体会

计划就是为了把项目的各种资源（人力资源，软件资源，硬件资源等）有序组织起来，以便及时识别变化、应对变化。所以做计划的时候，一要考虑如何使计划更加准，二要考虑一旦有变化、计划如何能更加容易调准。方法可能就是

1，尽量把任务拆解，和任务执行者一起确定故事点（scrum里的说法，这里借用一下） ...
展开

作者回复: 👍很有价值的补充，谢谢

hua168

2019-03-21

👍 2

大专学历，英语不怎么好，中年了去大厂难度大，所以打算在中小公司。

打算做技术管理，咨询了老大和前老大，他们都说要求懂开发做过项目管理，所以才选学java，主要考虑:

1. 成熟，教程多，传统...

展开

作者回复: 👍哪怕业余时间自己做点项目也算是项目经验的。

项目管理懂点技术会更好，不需要太深的，主要还是需要项目经验和管理经验。

bearlu

2019-03-21

👍 2

其实我一直想自己私下做一个项目，但是不知道如何开始，今日听了老师的课受益匪浅，但是是不是第一步确定做个什么软件？

作者回复: 对的，第一步先想好做什么。

给你的建议是：

1. 做个小的
2. 做个实用的，最好自己能用地或者身边人能用地
3. 迭代开发，第一版本只做核心功能

👍支持，做做挺好的。

青石

2019-03-24

👍 1

拿自己的职业规划来说，刚毕业时制定的拍脑门计划：IT行业拼到35岁，达不到预期转行（这辈子就这样，轻轻松松过小日子算了）。目前整体目标没变，年龄32岁，延迟到38-40岁之间。

工作经历：运维工程师5年，技术负责人角色4年带了3个项目（从头至尾），部门经理+...
展开 ▼

作者回复: 👍总结的非常深入

1. 管理，还是有很多理论知识在里面的，建议可以系统学习一下软件工程和软件项目的知识
2. 做技术管理，有一点coding能力还是好一点，一个便于沟通，另一个便于对方向的把握。也不需要特别深入，看几本相关语言的基础书；平时多Review同事代码；多看一些业界好的实践，看是否可以借鉴到工作中。
3. 如果你现在做技术管理，如果不是偏运维的管理，倒建议你不如花点时间补一点编程和管理方面的理论知识。跟木桶原理一样，先弥补短板。
4. 很多时候不需要太多试错，只要有理论知识指导实践，就不会偏差太多。



历知辛

2019-03-23

👍 1

宝玉你好,之前你提到"技术预研类型的项目"的特点,先对技术预研进行概述技术预研是指在产品应用前景尚不明确或技术难度较大的情况下,如果某项技术可以增强公司产品竞争力或对开发人员进行赋能,对这些前瞻性技术、关键技术或技术难点进行立项研究,着重探索 and 解决技术实现的可行性。其特点包括:

1) .技术预研的目的是验证产品技术方案或产品技术,并做技术储备;...

展开 ▾

作者回复: 我对预研类型项目没什么经验,只能说试着帮忙分析一下,给些建议:

1. 需要明确项目目标,这样才能有的放矢
2. 参考金三角,如果你想要计划性,需要把时间这条边固定下来,然后再选一条边,比如说把时间和成本固定下来,然后看能出来什么成果;或者说把时间和范围确定下来,不计成本;
3. 可以参考软件生命周期划分阶段,比如说:
 - 准备阶段: 明确目标确定计划
 - 调研阶段: 类似于需求分析阶段
 - 研发阶段: 类似于编码阶段
 - 验收阶段: 类似于测试阶段
4. 有了目标,才好定制订计划;划分了阶段,才好确定里程碑。
5. 同样,计划也是个从粗到细的逐步迭代完善的过程。



小先生

2019-03-23

👍 1

任务分解,估算时间,排路径,设置里程碑,跟踪调整。
做计划需要做好这些东西。

作者回复: 嗯,做好这些是最基本的,还需要实践中多积累经验:)



哥本

2019-04-12

👍

宝玉老师：请教一个问题，在做Code Review时要求是什么？这方面在一直没有实施起来，因为花时间让员工在做Code Review ,但有些人根据就没有认真去看代码，这个评判标准是什么？用什么纬度来衡量结果？还请宝玉老师指点迷津。

作者回复: Code Review后面章节会有谈到。

简单来说，你首先要把Code Review变成开发流程的一部分（参考前面大厂如何敏捷开发那一篇），比如说新功能，必须要Code Review之后才能合并主分支。

Review的时候，主要检查代码结构是不是符合架构，细节上有没有什么问题。其实不需要特别多的条条框框，网上也有很多实践可以参考，重点是先做起来，做的过程中逐步总结经验。

哥本

2019-04-10



宝玉老师，在做里程碑的时候需要花时间整合做集成测试吗？就比如向您说的，服务端开发完成后需要与pc客户端联调，那这就涉及到发布，环境搭建，部署。。。做WBS时要把这些时间也算进去吗？

作者回复: 做里程碑要不要整合做集成测试，取决于里程碑的目标，比如说如果目标是具备测试条件可以联调，只要能调就可以；也可以定义目标是要测试验收通过，这就需要做集成测试的。

这种需要人需要时间去做的事情，都应该放到计划里面。文章中的计划表没有放，是考虑不周。

判官

2019-04-02



宝哥，我想问下你wbs你是用什么工具制作的？

展开 ∨

作者回复: 原始文档是MS Project，后来我在Mac上用Merlin Project导出的图片