11 | 项目计划：代码未动，计划先行

2019-03-21 宝玉 来自北京

《软件工程之美》

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?

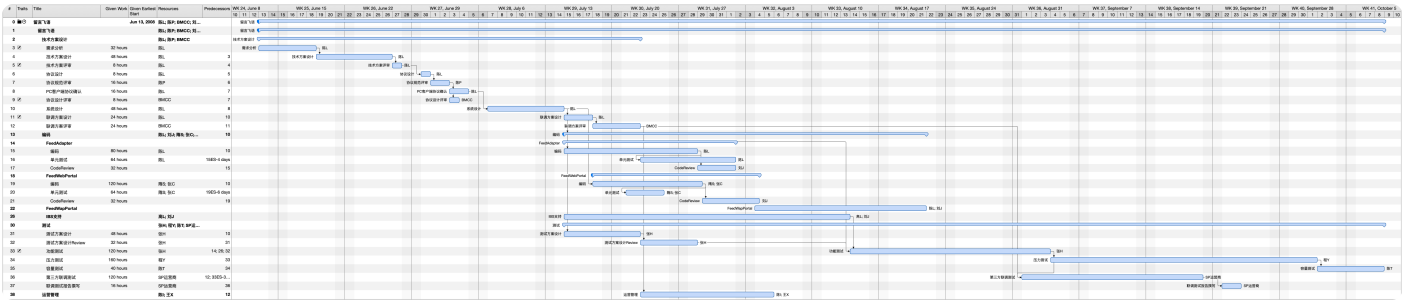
制定项目计划，通常有三个基本步骤：

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。

可以看得出，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。

	▼ **留言飞语**		**Jun 13, 2008**	**陈L; 陈P; BMCC; 刘...**
	▼ **留言飞语**			**陈L; 陈P; BMCC; 刘...**
	▼ **技术方案设计**			**陈L; 陈P; BMCC**
	▷ 需求分析	32 hours		陈L
	▷ 技术方案设计	48 hours		陈L
	▷ 技术方案评审	8 hours		陈L
	▷ 协议设计	8 hours		陈L
	▷ 协议规范评审	16 hours		陈P
	▷ PC客户端协议确认	16 hours		陈L
	▷ 协议设计评审	8 hours		BMCC
	▷ 系统设计	48 hours		陈L
	▷ 联调方案设计	24 hours		陈L
	▷ 联调方案评审	24 hours		BMCC
	▼ **编码**			**陈L; 刘J; 隋S; 张C;...**
	▼ **FeedAdapter**			
	▷ 编码	80 hours		陈L
	▷ 单元测试	64 hours		陈L
	▷ CodeReview	32 hours		
	▼ **FeedWebPortal**			
	▷ 编码	120 hours		隋S; 张C
	▷ 单元测试	64 hours		隋S; 张C
	▷ CodeReview	32 hours		
	▶ **FeedWapPortal**			
	▶ **IBS支持**			**高L; 刘J**
	▼ **测试**			**张H; 程Y; 陈T; SP运...**
	▷ 测试方案设计	48 hours		张H
	▷ 测试方案设计Review	32 hours		张H
	▷ 功能测试	120 hours		张H
	▷ 压力测试	160 hours		程Y
	▷ 容量测试	40 hours		陈T
	▷ 第三方联调测试	120 hours		SP运营商
	▷ 联调测试报告撰写	16 hours		SP运营商

估算时间这事，有很多方法可以参考，主要还是得依靠以前的经验。要想估算准确，需要从两个方面入手：

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。

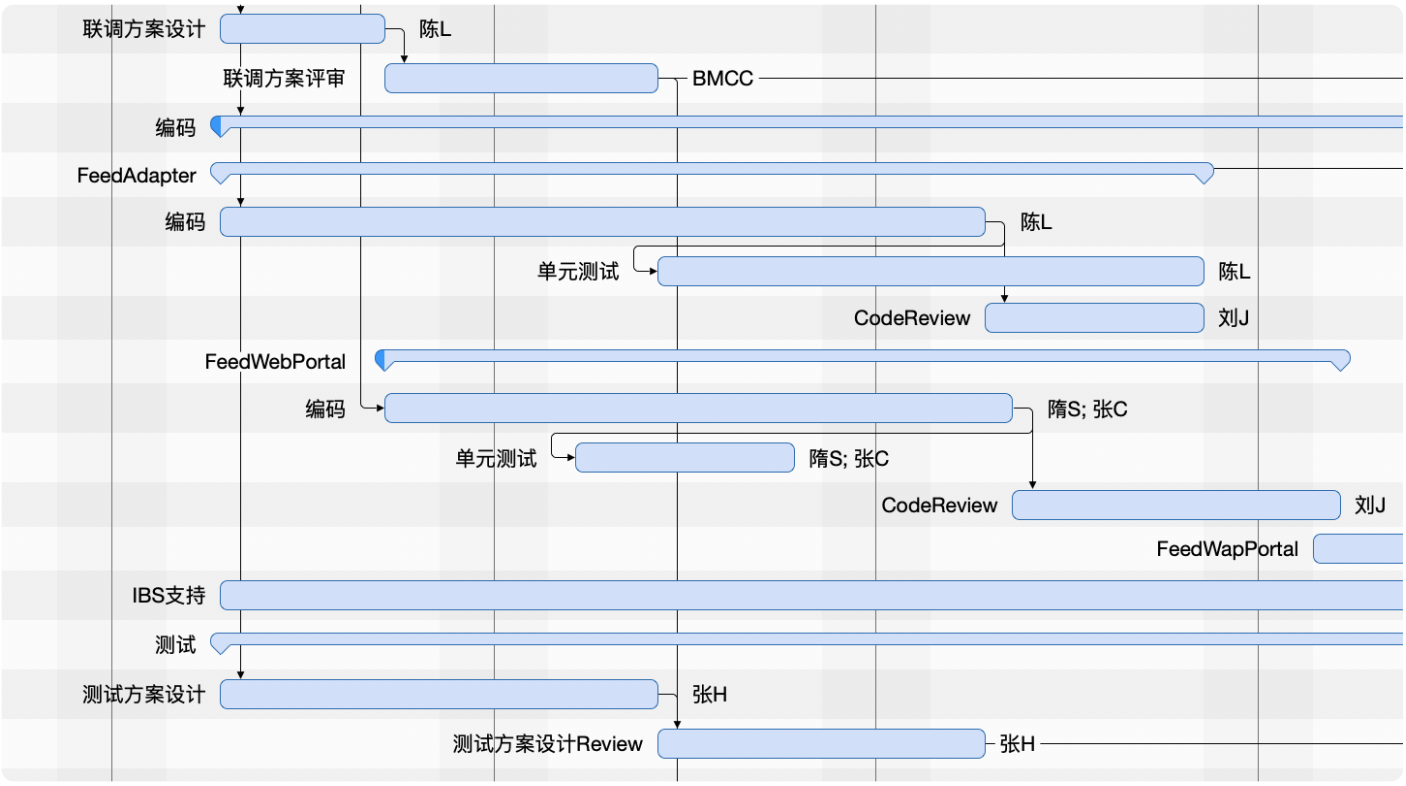
如果任务的粒度太粗，就需要进行细化，细化后就能更准确的知道结果。

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，例如可以每周一对计划做一次调整。

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。

精选留言 (43)

Winder

2019-03-21

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！

共 2 条评论 > 42

MiracleWong
2019-03-22

#
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工作，偶尔自己也会成为这种人），因为要介入制定计划、以及后续的调整，就觉得这不是应该是自己的工作量。
2. 就是制定计划的颗粒度粗细的问题。自己经常会遇到类似的困扰，就是一次计划，分解的太过详细，导致行动的时候因为繁琐反而拖延或直接不做。（也明白这是一种心理上的自我欺骗，认为做了计划就等于行动了，类似于买个课程就等于学习了，收藏了就等于看了）这就会导致下一次的计划时遭到“反噬”——上一次那么详细也没什么用，还不是不做或者稍微写一下呢。等到自己没有什么目标或者虚耗摸鱼时，有记起“详细计划”的好，一次次的循环。
3. 对于宝玉老师说的是否有指定计划的习惯，我经常是每个月的最后两天，指定下周的目标（类似工作计划）。将目标和自己的工作生活学习联系起来并进行分解，分散到每个月的四4周里。每周做个小结，月底再做月总结和下月目标。目前还是在尝试练习中，在逐步的形成自己做目标和总结的固定模板，省去部分重复性的工作。

一点自己的经验，第一次写评论，希望得到宝玉老师的指正。

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。

22

hua168
2019-03-21

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。

13

舒偌一
2019-03-21

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。

作者回复: 这一段总结的太好了👍

12

hua168
2019-03-21

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。

因为自学有些不懂，当过一段时间突然有想感悟又跑到从头看，虽然慢，但觉得还是在前进着...

订的长期计划就是几前年订目标10k/月，执行步骤：

1. 网管：3k-3.5k，再订收入5k,发现网络工程师可以
2. 网络工程师：培训CCNA，让人指导自学CCNP，面试达5k，因CCIE说网络没落转运维
3. win运维：6k，主要是运维公司官网用的是asp.net，再订目标10k，听说linux运维可以，转学linux，辞职学3个月，每天10小时
4. linux运维：达到目标10k，主要是运维公司官网

5.目前：17年底开始学java开发，打算1-1.5年，边运维学python，打算升运维开发

出现了中年危机，学习中带迷茫...好像自己什么都不精，中年失业还能找到工作吗？中年的出路在哪？

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：）

tcny
2019-03-22

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！

8

纯洁的憎恶

2019-03-21

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。

任务分解——WBS；
估算时间——广泛参与消除偏差+留出余量；
安排任务路径——甘特图捋关系+里程碑控制进度；
跟踪调整——信息共享充分沟通掌握计划进展+出现变化及时调整；

制定计划最好能让项目相关各方充分参与，这样计划更可行，偏差低，结果更可控、可预期。但我的经历却是需求、开发、运营、用户等角色几乎不参与制定计划，就连需求分析、功能设计、测试、验收也以工作忙为借口很少介入。项目管理人员主动拉他们，也遭到厌恶与不配合。在观念与体制不支持的环境里，如何能更好的调动各方充分参与、支持项目呢？

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

6

小伟
2019-03-23

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。

目前项目中遇到了技术性问题，导致里程碑delay一个月，但是老大也不太着急，自己感觉阻塞在这儿了。。。。

> 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。

5

LiYanbin
2020-03-09

宝玉老师，
　你好！我最近从一名专职的开发人员慢慢在带人了。最近有一个功能，由我还有另外两个人一起开发，我是项目负责人。首先，我们先对功能内容进行划分，并使划分的开发内容独立，然后把开发内容分发下去。
　然后为了保持整体开发计划的准确性，为了把握两个人的开发进度偏差，我去和另外两个人一起做好子功能的需求分析和软件设计，然后和他们一起做好开发内容和开发时间。
　我想我的这种做法是OK的。（不知道有没有哪些点还需要注意，宝玉老师可以根据自己过来人的经验来提提）
　但是我反想了，如果后面带领的是十个人，那这种方式就不太合适，如果是参与开发的人比较多，如何管理，如何控制项目整体进度偏差是比较有效的呢。我自己的想法，可以把这个十个人再分成两三个人为一个小组，然后我去管理小组组长，但是我去管理小组组长，就会导致粒度比较粗，粒度比较粗显然没办法更准去的把握这个项目的开发偏差。
　宝玉老师关于这个问题能否提供一些看法。
　此致
谢谢！

作者回复: 我想你已经做的很好了，我在这里分享一点个人经验。

我在带人时会更注意发挥其主观能动性，比如说任务，我一般在开始前只是帮助指导一下方向，会让其独立去分析需求，做系统设计。当然如果是新手，会指导一下基本方法和参考案例。

在分析需求或者系统设计时，要求写简单的文档，对格式不做要求，但要求说清楚问题，要求做评审。

评审会分多次进行，一开始只讨论方向性的，逐步细化，最终才是细节。这里特别忌讳一开始就写很细节的文档，如果万一方向错了，改动成本很高，抵触心理会很强，相反如果一开始只是粗的方向性的讨论，很容易调整，也不会有太多抵触心理。

在开发的过程中，通过PR的方式，在合并master之前要对代码进行代码审查，在代码审查及时发现问题，并提出改进建议。

通过评审的方式，可以帮助其独立完成需求分析和设计，帮助其提高代码质量，通过评审的方式分享经验，让其更有参与感和收获感，降低其依赖心理，最终能独当一面。

整个过程都要注意发挥成员的主动性、独立性，多沟通，多肯定和鼓励，引导其走到正确的方向，设置合理的期望值，双方都要避免“surprise”。

人多了以后分组是很有必要的，小的设计评审可以小组长负责，但是大的设计评审还是要参与，避免大的方向上的偏差。对于代码也要抽时间进行代码审查，及时发现问题。

共 2 条评论 >

4

小学一年级

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。

4

青石
2019-03-24

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+技术负责半年多。

目前困惑：
1. 算是步入管理岗，但对管理岗位总有些力不从心、迷茫，找不到切入点。之前工程师阶段有位老大哥，对我影响很深，现在多数做事风格也与他很像，也许因为不懂，所以照猫画虎。
2. 招聘软件上很多岗位要求，管理岗和技术岗都算，对coding能力都有一定要求，导致有想学各种语言的冲动，总觉得自己欠缺的太多，不愿走出舒适区。
3. 运维出身，很多方面都有涉及，但不成体系，比方说Linux操作系统原理、编码（Python）、架构、项目管理，哪些都懂一些，但知识零散。

短期内制定的计划：
1. 因为运维经历最长，从Linux（操作系统原理）和Python入手纵向发展，一直认为基础是上层建筑的基石。
2. 保持横向发展，将原来的零散知识聚集起来，形成知识体系。
3. 根据1、2步骤制定学习计划，每天不少于2小时。
4. 现有管理机会，多学多用，不断试错，找出适合自己的岗位以及工作方式，重新定位自己。

> 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管理的知识
>
> 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。

4

一路向北
2019-03-22

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。因为没有了小时间节点，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。
计划还是要有，而且还需要分解任务做的到位，各个节点清晰，可执行，可检验。并且计划需要得到团队成员的认可。
计划是否也会有一个迭代的过程呢?

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。

4

Know-nothing的Duckl...
2019-03-22

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?

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。

4

hua168

ffda168
2019-03-21

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
2. 学会了更方便维护和排错，在DevOps，用jenkins实现CI/CD更方便
3. 大数据还是以java为主

java学了SpringBoot+dubbo、Spring Cloud，
再学,运维开发为主的P,ython或Go+简单大数据。
先接触项目，没机会的话打算自己给自己安排，
然后学技术管理

> 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。


4

bearlu
2019-03-21

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，最好自己能用或者身边人能用
> 3. 迭代开发，第一版本只做核心功能
>
> 👍支持，做做挺好的。


4

小学一年级
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。

3

alva_xu
2019-03-21

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

3

alva_xu
2019-03-21

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）。这样的话即使计划变化，并不是每个任务都变化，计划调整就快。
2，对任务进行合理排序，找出关键路径和关键节点，项目的风险就比较容易识别。计划调整就能更加及时有效。
3，通过设立指标和看板，利用项目管理工具及各种形式的会议、报告，及时收集监控项目情况，适时发现问题，及时调整计划。

作者回复: 👍很有价值的补充，谢谢

LiYanbin
2020-03-20

宝玉老师，你好，项目组成员进度出现偏差了，要怎么调整。有哪些原则在里面吗？谢谢

作者回复: 遇到类似的问题，最基本的原则就三条：
1. 恢复生产
2. 总结原因
3. 防患未然

先说恢复生产，出现问题不要着急找原因，找终极解决方案，而是先恢复先止损，哪怕是临时的方案，等恢复生产后再找更好解决方案。
项目组成员进度出现偏差了，首先要做的就是重新调整进度，该砍需求的砍需求，该加人就加人，该加班就加班，不能因此影响了整体项目进度。

然后再说总结原因，如果问题没有找到产生的原因，那么这样的问题就会一而再再而三的发生，这次你调整了进度，下次还会出问题，所以需要分析原因。
项目组成员进度出现偏差了，是什么原因造成的？需求没有分析清楚？老是变更？技术债务？没有做设计就开发？Bug太多？

最后就是防患未然了，分析出来原因后，最重要就是找到相应的解决方案，避免类似问题再次发生。如果是需求没有分析清楚或者老是需求变更导致进度出现偏差，那么就要加强对需求的管理；如果是设计没想清楚导致开发走弯路，那么就加强设计和设计评审；如果是技术债务太多，那么就偿还技术债务；如果是任务拆分的不够细，那么把任务拆分细一点，把Sprint周期缩短一点。

遇到这类问题很正常，很多问题都可以参考上面的三个原则和步骤来处理。

共 2 条评论 >

韩俊臣
2019-10-15

我所在的项目一般是以年为单位的，年初搜集完需求并制定计划。有些子任务延期个几天倒还好，就怕那种后面的任务要提前开发，或者突然有紧急任务加进来的情况，请教下老师：处理这种事件怎么做比较好？

作者回复:
计划总是赶不上变化，最好的应对方式就是让变化变成计划的一部分。在响应变化方面，这方面敏捷

开发是做的比较好的，而且也是可以直接借鉴应用的。

首先，迭代起来！

如果你的项目是以年计，那么能不能把你的项目按照月拆分成12个迭代？甚至于更多？

当你有了迭代，你就可以把你所有的任务分布到不同的迭代中去。

后面的任务要提前开发怎么办？比如说原本要放在第10个迭代的任务，现在要提前到第5个迭代，那么把第10个迭代的任务提前，同时把原本第5个迭代的任务往后移。

突然有紧急任务加进来怎么办？完成当前迭代，把紧急任务放到下一个迭代，原本规划的下一个迭代的任务，往后移，或者砍掉一些任务。

迭代有两个主要原则要注意：
1. 迭代的周期是固定的
如果你是1个月一个迭代，那么你每个月到时间点了就应该发布一个版本，如果有困难就通过砍功能、加班加人等手段确保完成。
2. 当前迭代过程中不新增需求
如果你的需求一直在变，总是临时加进来新的任务，那么你的迭代是很难保障的，所以把握好原则：有新需求，紧急变更，放到下一个迭代，先按照计划把当前迭代完成。只要你的迭代周期设计的适当，并不会影响整体进度。

然后，把你的任务用工具管理起来

善用工具，现在有很多任务跟踪工具，比如Jira、GitHub的Issue，当你的年度计划制定后，拆分成一个的任务项，放到任务跟踪工具上去，尤其是结合看板视图，每个迭代要做的任务、进行中的任务、完成的任务一目了然，无论是项目成员，还是项目外的人员，都可以直观的看到任务的执行情况。

这样当其他人要新增紧急任务，或者把后面的任务提前，通过任务跟踪工具，可以直观的看到对当前计划的影响，也可以方便的对计划进行调整。

以上两点就是我的建议：把你的计划迭代起来；把你的计划用工具管理起来。