



下载APP



24 | 技术决策（3）：持续跟进进度，执行细节决定成败

2020-10-14 许健

技术管理案例课

[进入课程 >](#)



讲述：许健

时长 18:18 大小 16.77M



你好，我是许健。今天我们来聊一聊技术决策的执行问题。

我在 [人才招聘](#) 中讲了招聘的时候要追问细节，在 [进阶心路](#) 里提到二线经理要下沉两个汇报级别到一线了解细节，在 [危机管理](#) 中还提到过我自己被架空的一个重要原因是没有深入到日常的团队运营中去，本质上就是没有去了解关键路径的关键细节。

为什么我一直这么强调细节呢？因为我自己经历了好几次教训，追根溯源才发现都是因为忽视了细节，准确地说是对**关键路径的关键细节**不够重视。



今天我想跟你聊的就是技术经理如何把控细节，如何通过技术细节交付好业务，培养好人才。

做完决策就可以了吗?

首先, 我们做完决策就可以了吗? 答案是否定的。

我看过一本叫《执行》的书, 里面有这样一句话: “这样做领导当然舒服喽, 你只需要站在一旁, 进行一些战略性的思考, 用你的愿景目标来激励自己的员工, 而把那些无聊的具体工作交给手下的经理们。我在这里要提出的是, 这种思考问题的方法是错误的, 它很可能给你带来难以估量的危害。”

我对这句话深有体会。做决策只是起点, 如果止步于此, 等于空谈, 为什么这么说呢? 如果我们

觉得决策做完了就万事大吉, 那么常常就会有这样的后果:

第一, 是经理自己被架空。哪怕我们通过决策把工作优先级聚焦到了核心问题, 但决策落地最后还是要靠执行的。如果不能把握关键细节, 完整地跟进进度, 经理就容易把自己架空。

第二, 会导致业务价值无法按时交付。有一次总部领导质问 UX 小组半年内没有交付业务价值, 事后我跟 UX 小组的经理复盘时, 才发现我们当时对于关键依赖的解决不及时, 人员不够专注, 而且起初的设计也过于复杂。但得出这三条总结已经是事后诸葛亮了。

第三, 就是无法解决部下的困难, 再远大的目标最后还是要一步一个脚印地推进, 技术经理要了解一线的具体困难, 采取行动解决, 否则会直接影响到自己的领导力。

我这里想说的是, 无论是一线技术经理还是二三线技术经理, 尤其是二三线技术经理因为会花更多的时间考虑战略文化等事务, 所以更要时常提醒自己去了解关键路径的关键细节。

我们不要把自己架起来, 因为一个项目一个产品不是静止的, 在项目和产品的发展过程中会不断的出现问题, 这些问题你不能单方面的等着下属自下而上汇报, 也应该由上而下去了解。

只有注重了关键的细节, 你才能对你手上的关键产品和项目有比较全面的了解, 才能持续地根据实际情况做后续的技术决策, **技术决策不是一次性的买卖, 需要为了达成目标做持**

续修正, 项目没有交付前就不能放松警惕。

关键细节如何把握?

我们知道了执行细节很重要, 那具体应该怎么把握这些细节呢?

一般在执行过程中我会同时使用下面三种方式来了解执行状况, 从而保证执行达到预期效果:

方法一: 执行负责人要制定可衡量的阶段性目标

我的经验是计划会变这一点我认可, 但是我不接受因为这个理由就不去制定计划。

虽然我们可以理解很多事情很难估计时间, 强行制定时间表的话最后也许就不能交付。那么很难估计时间的话, 目标到底怎么制定呢?

我的方法就是必须定计划这一条没有商量余地, 但是可以做拆分, 直到拆分的单元可以很好地做衡量。

具体来说, 就是如果季度目标不准就定月度, 月度目标不准我们就每两周一次的 Sprint 目标, 再不行我们就定每一周的周计划。

方法二: 定期的项目评审和周报

为什么我要强调定期呢? 定期的作用就是给执行负责人传递一个明确的信号, 也就是他的上级技术经理正在持续关注他的进度, 愿意及时解决他的实际困难。

同时, 这么做对执行负责人也是一种压力, 每一次到了项目评审和周报的时间点, 他都需要给领导和给同仁看到进度。

关于项目评审和周报有下面三点注意事项:

第一点, 我不要听流水账, 所以负责人要对照项目之前定下的阶段性目标来汇报进度。

第二点, 如果项目进行过程中有阶段性目标达成, 或者有值得肯定和表扬的点, 我会要求负责人单独注明。

第三点, 项目中碰到的问题也需要单独标注, 这一条非常重要。我一般会建议执行负责人用绿, 黄, 红三色来对问题分类。绿色是已经解决的问题, 黄色表示虽然出现了问题, 但是执行负责人已经拿出了具体的措施在解决问题, 目前还不需要上级领导介入, 而**红色表示要求上级经理介入或者提供帮助**。

在 **辩证思维** 一节中我讲到自己的一个思维习惯, 就是重要的项目在决策过程中要挖掘痛点, 如果没有挖掘到, 我就会觉得有潜在问题, 然后继续深挖。

同样的, 在决策执行过程中我想给你分享我的另一个思维习惯, 如果我在项目评审和周报里只听到好消息, 从来听不到困难和坏消息, 我就会给自己提一个醒: 这么重要的事情, 在执行过程中这么一帆风顺正常吗? 这会直接导致我找一线骨干做 Deep Dive。

方法三: 不定时地找团队一线骨干做 Deep Dive

在日常工作中, 技术经理要下沉两个汇报级别, 去找一线骨干了解细节。其实我们在做关键项目的执行时更应该如此。

具体怎么做呢? 我是带着希望这个项目成功的心去“挑刺”的, 我通常会这样去思考: **如果有三个原因造成我这个项目失败, 是哪三个原因? 我现在可以作出什么举措降低失败几率。**

通过找团队的一线骨干了解细节, 我发现自己对项目的了解更加全面, 也能尽快发现实际操作下来经常发生什么样的问题, 下面三类问题是最常见的:

第一, 对性能压测和失效性分析重视程度不够。我们部门在吃过几次苦头以后, 在这一点上已经比较重视了。墨菲定律在我们部门负责的项目上几乎次次灵验。

第二, 对关键依赖过于乐观。现在我们部门对于关键依赖我的建议是: 没有书面承诺交付时间的, 就当这个依赖没有搞定, 应该找高级别技术主管介入。

第三, 技术实现时夹带必要功能之外的加分项功能。从我们部门目前实际的执行历史看, 在核心功能交付前, 所有锦上添花的功能都应该全部砍掉。

持续跟进的注意事项

前面我们说了关键细节如何把握, 但细节更多的还是一些关键节点, 在技术决策的执行中, 持续跟进还需要一个线性的思维, 树立起对决策、对自己角色以及对未来如何做预判的认识。

决策不等于落实

请你跟着我回顾一下自己做经理的经历, 我们总能轻易举出一系列半途而废的事情, 这些事儿都是我们做了决定但是却没有跟进, 最后不了了之, 没有落实。那么问题到底出在哪里呢?

第一种情况, 就是做了“建议”没有跟进。这里我专门用了“建议”这个词, 而没有使用“决定”这个词, 这是因为如果我们都不去跟进确认, 就不能算作决定。

我给你举一个具体的例子来说明。我们部门已经多次出现误删操作, 因为权限管理太松, 发生过不应该有权限的人误操作, 结果删除整了个测试数据库的情况。最近一次是因为数据质量问题, 系统误删了生产环境流量入口, 幸好部署了高可用 HA 防护才没有出大事。

我做复盘的时候, 想起我虽然一次一次建议要引入保底措施, 在删除逻辑中加入流量检查保护。但到目前为止, 我只是反复在强调不采取措施的风险, 这个风险就是一旦出问题影响业务, 会导致打乱我们所有的原定计划, 然后全员都要去做系统可靠性强化工作。

我的部下不是笨蛋, 如果没有落实一定有现实的困难, 需要我去了解具体的困难。很有可能落实这项工作是需要以牺牲功能开发为代价的。再联想到我一直强调“说话算数”, 也就是我们部门承诺的功能一定要按时按质交付, 我觉得问题出在自己只是做了建议, 但没有继续跟进。

我们不止一次踩了坑, 还是出问题, 想要改变一定是艰难的, 那这个艰难决定一定是组织一把手来做。

第二种情况, 就是没有落实标准。在我们部门最常见的就是出了一个生产事故, 于是在风头上大家都十分重视, 制定出一系列举措。问题是我们很少会去重现那个生产事故, 通过测试进一步确认我们落实的那些举措的有效性, 也就是在下次发生类似事故时, 我们指定的举措是否真正能够起作用。

比如说我们上个月发生的一事儿，因为一个 BUG，PaaS 系统的一个超级账号抹掉了整个测试环境监控系统，这个事故我们花了四个小时才恢复服务。

在复盘会议上，团队成员觉得以后我们可以在 20 分钟内恢复服务，但这个想法没有落实到测试环境中检验，其实很难立住脚。

所以我明确要求监控组一定要复制一套一模一样的线上系统，再做一遍删除系统后恢复的模拟测试。要知道，我们设想中可以 20 分钟完成的恢复操作，和真的做一遍，验证一下在 20 分钟内能不能恢复是不一样的。

其实本质上这类问题都反映了技术经理的重视程度，特别是和经理是否能够持续重视密切相关。作为技术主管，你选择要落实自己做过的决策，为了真的落实你就会愿意花时间。

更重要的一点是你愿意延后、甚至取消你想做的其他事情，来给你自己和你的团队腾出精力专注落实这项决策。

经理也是执行者

就算我们已经给关键项目安排了负责人，作为主管经理，我们同样也是执行者。怎么理解这句话呢？经理不是在项目启动的时候定一下方案，然后安排一下人手，接着定时听听汇报，做做 1:1 和 Deep Dive 就够了，经理得为员工办实事，这就是我说经理也是执行者的意思。

我自己在听定期的工作汇报或跟一线骨干做 Deep Dive，经常跟团队成员说的话是这样的：领导的工作不单单是给你分配任务，也包括帮你解决你的困难，领导最不希望看到的就是到了快交付了你跟领导说因为这样那样的原因，项目不能如期交付。

所以，如果你觉得有风险需要领导介入，就应该第一时间跟领导沟通。你要有这个气场可以给领导安排工作。

在我们一个冲突比较激烈的项目中，我甚至对技术负责人说过这样的话：“这个项目如果你搞不定技术实施是你失职，如果我搞不定 Alignment 是我无能。”

其实这么说，就是因为我认为经理本身也是执行者，关于这个问题，我有三条原则和你分享：

第一, 如果部下反映了问题, 我能够解决的必须马上解决, 不能拖, 更不能石沉大海。

第二, 如果部下反映的问题我不能马上解决, 那我需要告诉部下我大概在什么时候可以解决。

第三, 如果我评估下来我也无力解决, 我会告诉部下不能解决的原因。

当然, 原则还是要结合实际场景才能体现, 总结一下就是我们应该把定期的工作汇报会议变成解决问题的现场办公会议, 而不是一个给领导单项汇报工作的汇报会。作为领导, 要给部下反馈和采取后续动作, 做得好的不要吝惜表扬, 碰到问题了领导也是执行者需要切实解决困难。

早点到真实环境历练

我们都知道, 管理“未知”是最难的, 有时候我们可以预判后面执行还会有这样那样的问题, 但这些问题你是很难事前预料的, 这时候应该怎么办呢? 我的做法就是早点放到真实环境去历练, 争取尽早暴露问题。

我结合具体的业务案例来说一说, 我们公司因为要应对购物季的流量高峰, 每一年在购物季之前, 都需要流量管理团队加班, 加班的任务就是把数千对负载均衡器上的流量平均分配好。

今年因为疫情原因, eBay 网站流量高峰提前到来, 这也导致了问题的加剧。副总给了死命令要在今年三季度结束前完成流量重分配 100% 自动化。

因为这样的情况, 我在跟流量管理团队做定期审核的时候最最关心的问题, 就是什么时候可以 Dry Run (在线上模拟运行)。我担心自动化没经过试验, 存在影响业务稳定进行的风险。事实上, 我们在所有功能都开发好以后 Dry Run 的自动化成功率只有 10%, 强化一个月后达到 70%, 再强化一个月后达到 80%。

今年年底前我们需要交付的 Os Patching 也是一样的, 公司的安全合规要求越来越高, 这对我们全网 Patching 的挑战就越来越大, 十几万台机器的 OS Patching 从半年一次到一个季度一次, 甚至到明年要达到每个月一次, 我们觉得引入独立集算法可以增加升级并行度加快速度。

但真正跑起来, 应用重启后也许不能自动恢复到正常状态, 多应用同时重启压垮下游服务, NOSQL 类应用升级时数据重平衡时间过长, 这一系列问题让我们举步维艰。最后, 我们得出的结论就是早点去战场上历练吧, 我们只有提前去线上摸爬滚打, 才能把十多年积累的“垃圾”都暴露出来。

通过关键细节做人员选育

作为一名技术经理, 最后还是要回到人才培养上来。我认为重要的关键项目除了交付业务价值, 另一个重要作用就是培养人才。那具体怎么培养呢?

我最近半年最大的收获就是**指出关键细节上的差距**。技术主管只有深入一线了解关键细节, 才有可能对执行这些项目中的骨干提出有建设性的高质量意见。这些意见对于项目骨干的成长意义重大。

不知道你是否记得, 我在 [员工沟通](#) 那一节就提过跟高级别技术骨干不能光谈原则, 必须落实到具体的细节。我之所以能够重新审视我们部门的执行文化, 并作出改变, 正是因为自己的老板指出了我在细节上的差距。

当时我的老板指出我鸡血很足, 但是对部门要求并不高, 而且她拿着兄弟部门的团队业绩, 人才梯队甚至还有对派遣人员的培养计划来做佐证。如果我领导不拿着具体的细节给我看, 恐怕不会有这么大的触动效果。

这么做的好处是当你给具体的细节的时候, 听的人其实已经从具体的实例中自己得出了你想要的结论, 这时候你再说出结论, 就更容易被听的人接受。

这个抓细节的思路我也应用到了自己培养部下能力的过程中。比如我跟经理 X 确定了他今年必须有一个目标, 这个目标是培养他的左右手到没有他在也能独当一面的水平; 我还找到我们部门的技术骨干 E, 跟他探讨一下为什么我们部门的关键项目不能由一名骨干带上一批派遣人员来实施, 而兄弟部门却可以。

有了细节当作切入点, 我们对手下的培养就不是空谈的方向了, 而是拿着关键细节作为抓手, 引导员工尤其是项目骨干找到矛盾点, 在解决矛盾的过程中提升自己的水平。

总结

今天我给你分享了技术决策后对执行细节的掌控是项目成败的关键。

首先我提到了作为经理, 特别是高级别经理的一个大坑, 也就是整天关注战略层面的宏观问题, 听听报告而忽略对关键项目的关键细节的把控。

接下来, 我给你总结了掌握执行细节的三个方式, 分别是设定可衡量的阶段性目标, 定期做项目审核和周报, 以及不定时跟项目一线的骨干做 Deep Dive。我们始终要未雨绸缪, 时常问问自己如果我这个项目失败了会因为什么原因, 从而及早作出安排。

然后, 我们要认识到做技术决策不是一次性的买卖, 技术经理必须用发展的眼光做持续跟进。决策不等于落实, 技术经理可不能就听听报告, 你需要及时了解关键细节使得你能够及时指出调整。

我特别强调了技术经理应该把工作汇报的会议开成现场办公的会议。我们不可能一次性掌握所有情况, 很多事还处于未知状态, 我们就很难推演怎么降低风险。所以, 我建议是尽早到逼近真实战场的环境中去历练, 毕竟实践出真知。

最后我们还是回到执行细节对人才培育问题上的意义。我自己对部门的要求还不够高, 这一点我觉得是我们部门这两年来人才培养上的两大缺陷之一 (另一点是给关键骨干全权负责关键项目的机会)。意识到这一点, 我在今后还会继续提高, 通过细节入手, 让团队成员变得更强。

我的建议是, 对我们的骨干要高标准、严要求, 而指出他们的问题时, 就要拿出具体的细节上的差距。**高标准严要求, 甚至是用严厉的语气明确说出来哪个具体的点做得不够好, 这对唤醒人的自我觉悟, 提升人的能力帮助很大。**

思考题

请结合你的实际经历, 分享一个你自己注重执行细节的正面案例或者反面案例。

我们大部门有一个小组既要维护现有系统又承诺了新系统的多个子系统的研发, 但是该小组经常需要救火, 承诺的时间点也出现不能准时交付的情况。总部领导跟我谈起这个状况时, 说他会给该小组经理更多的压力, 你怎么看待总部领导给更多压力的解决方案呢?

欢迎在留言区晒出你的经历和疑问。如果有所收获, 也欢迎你把这篇文章分享给你的朋友, 说不定就能给他一些启发。

提建议

极客时间 3 周年

做任务
得千元礼包



【点击】图片, 立即参加 >>>

© 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 23 | 技术决策 (2) : 拥有辩证思维, 才能在纠结中负重前行

下一篇 结束语 | 时时反思, 优化管理能力

精选留言 (3)

写留言



Weehua

2020-10-24

问题1, 我最开始带团队的时候, 团队只有两个人, 当时在做第一个项目的时候, 我并没有去review细节, 而且甩手掌柜, 项目很重要, 是我们部门的第一个对外合作项目, 两周开发时间。后来到了第二周的时候, 一个小伙伴私底下告诉我另一个小伙伴的代码质量很差, 感觉不能被信任, 处于对结果的保证, 我们沟通下来是他把另一个人的工作也都做了, 加班做了! 最后成功交付, 但几乎90%的工作都是他自己搞定的! 后来想想, 我最...
展开 ∨

作者回复: 问题一, 我觉得你做的很好了。到了二线经理以后, 我现在能做到的是review 项目, 但是我后来发现这样不够, 我还得去看看customer support. 这些我还是可以覆盖的, 代码层次, 我会让一线的人给我看他们的测试用例。但是我已经很少坐下来很仔细的跟大家一起做code review 了, 这一点是个遗憾, 除非我能进一步优化时间管理, 比如一些效率不高的会议就不要去了。我装了一个rescue time 的app.

问题二: 对的, 我觉得大家要知道自己的部下不是傻子, 一定有具体的困难, 二三线领导应该切实解决下属的困难。



1

**此方彼方Francis**

2020-10-20

“如果我在项目评审和周报里只听到好消息, 从来听不到困难和坏消息, 我就会给自己提一个醒: 这么重要的事情, 在执行过程中这么一帆风顺正常吗?”

类似的, 在开发一个复杂功能的迭代中, 如果bug数量比较少, 我也会在心里反问自己一句: 代码质量真的这么高?

展开 ∨

作者回复: 能量守恒, 要么就是你真的花了大力气在事前的Test Coverage, CI/CD 上面。肯定不会无缘无故就形势一片大好。



1

**我能走多远**

2020-10-24

说一下个人执行细节的案例吧!

上家公司没有完善的项目管理流程, 导致项目交付质量差, 体现在开发因方案频繁变更导致交付延期; 代码review过程中提出新的解决方案要求开发重新修改; 交付项目没有自测报告及影响点; 项目上线后没有和测试对齐等等。

我做的几个项目都是严格按照项目交付流程, 设计分析会拉上技术经理及相关模块责任...

展开

作者回复: 怎么理解 发布后及时和测试对齐 这句话？

