《技术团队的管理》交流会精华总结

一、 管理难就难在没有标准的答案。

管理可以说是比较难的，但也可以说是比较容易的。那难在哪里呢？因为管理没有标准答案，它怎样才算好，怎样才算差，没有一个很标准的答案，所以它就难。那说管理容易，又容易在哪里？比如说我说带的这两个人，好像我不用理他们，他们也能够交差，具体的事情他们也能做，那他们做好了这个功劳是算我教得好，还是算他们自己做得好？这个很难决定。所以我觉得**管理这个问题，说容易也容易，说难也难，因为它没有一个非常标准的答案。**

二、 管理、带团队的三个目标：一是项目目标的达成，二是项目是可持续发展的，三是团队是可持续发展的。

1、更快速、保质保量地完成项目的目标，发现团队中的潜力人才。

要把产品立的项目做完，这是最终目标。但是在追求目标完成的过程中，我希望能够做到几点，第一：**团队的成员能够有成长**，大家通过做完做这个事情，不管是做事情方面，还是技术上面都能有成长。第二就是：**希望能够从团队成员里看到一些比较好的苗子**，希望能够从团队成员中选出一些能够辅助我、帮助我管理团队的一些人。

除了把东西做出来，还有一个就是**通过管理能够让它质量更加有保证，能够更加快速地做出来。**如果你不做管理可能三个月才做出来，但是我做了管理的工作，可能一个月就做出来了，时间是可以控制的，不会很懒散。

大家要想清楚我们管理的目标是什么，**管理的目标，是为了要达到我们整个项目的目标，是为了我们绩效的达成。**比如说公司考核你这个团队、考核你自己，你的 KPI 在哪里，**管理就要往 KPI 上面靠。**比如说我要带领团队，要开发一个版本，现阶段的管理目标就是这个版本要按时或是提前，或者说要高质量地完成。

2、管理要达成的三个目标：达成项目目标，项目的可持续发展，团队的可持续发展。

管理的目标清晰化以后，接下来就是我们怎样才能更好地去达成这个目标。这就是如何进行管理的问题。比如说我要开发版本或者开发某个产品，要怎样才能够快速地往前赶，这个是一个目标。另外，我觉得除了这个目标以外还会有其他的目标，我从逻辑上构建了一个我认为的管理的模型。比如说我**在这个过程中发现了一些苗子，或者探索了一些新技术。**那为什么我们要去发现这些好的人才，或者说探索这些好的技术，实际上我们做这个事情也是为了后面工作的开展。比如说我后续要做其他项目，或者要做同类的事情，有了现在的积累之后就能够很快地达成这些目标。所以抽象来看，我觉得除了我们完成现阶段的目标以外，还会有两个目标，一是我们这个项目能否持续地发展，二是我这个团队是否能够持续地发展，这就是我认为团队管理要达到的三个目标：**一个是现阶段我们要达成的项目目标，二是我这个项目能否是可持续发展的，三是你的团队是否能够可持续发展。**

项目的可持续发展是什么意思？比如说我现在开发了 1.0 版本，那可能后面还要开发

2.0 版本，那 **1.0 可能是比较困难的，那要达到 2.0，可能要更快速地完成。**或者说重构的时候、做一个类似的项目的时候能否比这个速度要快，质量更好。这就涉及到我们在做 1.0 的时候是怎么考虑的，或者说是怎么去积累的。我们积累或者说探索了哪些技术，我们有没有发现一些好的苗子、人才，每个人的能力有哪些提升。通过这些，才能够去保证项目往后的发展，纵向或者横向的发展，能保证我们比较快速地达到这些目标，这就是我认为的项目的可持续发展。

第二，什么是团队的可持续发展？一个团队从零开始，它总是很困难的。所以我认为，这个团队一旦组建起来了，后面它就比前面要更容易了。比如扩充人员会更容易，战斗力也增强了。这就类似军队里面，兵都是年年变的，但是不变的是那个部队还在那里。**我们要打造一个“铁打的营盘流水的兵”**，说的也就是这个意思。所以说**人员可变动，但是它整个团队的战斗力它是不变的。**我觉得从长远来说，要达到这样的一个目标，这个是对团队的可持续发展的理解。

3、在流水的兵的情况下，怎样让整个团队的战斗力保持不变？

提问 我目前的部门涉及到的技术会比较单纯，所以希望各组员在做完自己的工作之外，可以互相去了解其他成员的工作内容。因为我们每个成员负责的东西是不一样的，我想通过这种方式让他们去了解，互相了解相关技术，这样就可以不用招很多的备用人员，有时一两个人请假或者离职，对整个团队影响也不会太大，想听听大家对这方面的看法。

回答：我大学暑假的时候，在金山实习了两个月，它有两点值得我们学习借鉴。一**是它开发组会有自己开发好的库，自己积累的资源。**这个资源跟外面的不一样，他们有些库是实现了一些很常用的功能。比如我们做视频，有一些很常用的功能，我们可以把别人没有的功能创造出一些东西来，把它做成一个库，然后在开发组里面存起来，作为我们组的资源。**第二点是金山有很严格的开发规范。**严格到比如说一行代码的缩进是用 Tab 还是空格都是有规定的，如果能把组里每个人的开发习惯、开发思维都统一到很高的高度，这样的话其实不管谁来做，只要他有那个基础，做出来的产品质量都是差不多的。不过实际操作上可能没那么容易做到这种程度。所以**铁打的营盘流水的兵，主要是这个营盘，它有自己的资源，而且跟外面不一样。还有这个营盘有很铁律的规定，每个进来的人都要遵守这个规定来做，然后不管谁进来做，结果都是差不多的。**

提问：这里我有一个疑问：带 2 个人的团队跟带 10 个人、50 个人的团队，是否都是一样的做法？还是说我们一开始就要定好这个价值观，定好这个流程，就按照这么做下去。

回答：金山的例子主要说的是要从制度上建立一些规则，大家都遵守这个规则的话，比较标准的产品都是根据这一套规则做出来的。

回答：是的，其实团队与个人的区别就是因为团队它有一个营盘在这里。那现在的问题是怎么去搭建这个营盘。

刚才大家讲到的金山的做法就比较好了，他有一些库在里面，这就是它的一个积累。这就是技术沉淀的问题，所以我们要想，**在平时开发过程中怎样把这个东西沉淀下来，对已经**

是公共沉淀的资源，应该想怎样把它累积起来，这实际上就是技术沉淀的过程，也是对团队能力的构建。比如说我要快速的从 1.0 迭代到 2.0，1.0 可能要开发比较久，那 2.0 的时候可能开发就会比较快了。快的原因也是因为有 1.0 积累，这个积累可能是把代码直接就拷过去了，复制了一份。但是有些做法它就是已经固定了，做了一些库，那些库是已经非常稳定了，以后就不用再测试了。如果拿源代码可能每年都要改， A 改成 A 的模样，B 改成 B 的模样，C 改成 C 的模样，最后它就不一样，所以这实际上就是一个技术的沉淀问题，慢慢积累下来，这样实际上也构建了整个团队的能力，它不会依赖于某一个人。

另外一点，可能团队里面有好多人，怎么办？怎么备份？这就是团队的可持续发展问题，一个团队往后发展不能依赖于某一个人，如果这个人走了，难道这个团队就不运作了？这样整个团队已经绑到这个人上面去了，所以说我们就要做好备份。但如果团队里每个人都要备份，那消耗的精力是非常大的。**我认为这里要做的备份，可能就是针对一些大的项目，或者是一个团队里的核心东西和比较关键的环节，做好备份和文档，**类似金山的做法，把一些比较常见的公用的做成类库的形式，把它积累下来，这样就能够很好地保证团队的运作。因为我们做某一款产品或者某个技术，他主要核心的技术不会太多，主要你是做研发层面，在研发层面，我觉得也是形成一个库跟文档。**要从 0 到 1 上去探索，它是比较困难的，但是如果是一旦从 0 突破到 1 以后，1 到 2 它就要容易很多，它容易的原因就是因为我们在 0 到 1 的时候已经把原来比较难的形成了一些文档，形成了一些规范，只要大家往那个方向去做，结果就会不一样。**比如某个人可能是一开始研究从 0 到 1 的那个人，通过他积累上来的库和文档，那他就很容易消化 1 的事情，这样不至于走掉一个人，团队就不能运作了。

4、要根据项目的进展，灵活把握好管理三个目标的侧重点。

其次关于管理方面有哪些目标？刚才我们提出了三个问题，**一是我们要保证达成现阶段的目标，第二就是项目要可持续发展，另外就是团队可持续发展。**那我们应该怎么去做呢？我认为我们要从三个方向去构架、去管理、去跟踪。 这三个方向分别是**项目管理。技术管理，团队管理。**

三个方向的侧重点在哪里？

首先是项目管理。现在的项目，一般是分了很多小组，但是一般都没有专门的项目经理。团队里面没有专门的项目经理的管制度，对研发或者是对于互联网公司来说可能大部分都没有。比如一个团队里有技术 leader，这个技术 leader 并不跟进项目，另外有个项目经理来跟进，你要什么人，我就给你什么人，进度是由项目经理去跟进的。大概从 09 年开始，我所经历的项目就基本没有专门的项目经理的角色。那这样**技术就有两个职责，一是技术 leader 的问题，解决技术难题，另外团队产品的开发，产品的进度也要跟进。**每个公司做法可能不一样，有些公司可能就是产品负责制的，所有进度都是产品去跟进的。但目前对于大多数技术的 leader 来说，他既负责了技术难点的把控，另外实际上进度也是要跟进的，这个跟进就是项目的管理，所以**要有项目方面的管理意识，就是要达成目标。**

第二是技术管理。技术管理实际上是管理什么？技术的沉淀，**技术的构建，技术的预研。**比如**开发这个版本存在哪些风险点、难点，要往这个方向去研究，要安排一些人去预研。**另

外，要做一些评估，这个就是技术管理这个方向。另外在做某些版本的时候，可能要抽调一些人除了做开发外，还要专门负责去研究，把原来做过的事情分发出来，这个就是技术管理。

第三是团队的管理，它的侧重点在哪里？**既然是团队，首先要解决协作的问题**，技术人员有什么特点，可能很多技术都很强，逻辑能力非常强，但是如果两个人配合，可能不一定能配合过来。所以要解决团队协作的问题。

另外**团队管理还要考虑到人员能力的提升问题。**比如团队刚组建的时候可能有很多人过来，有些是应届生，有些工作两年、三年了，这些人能力就有高低，那要打造“铁打的营盘流水的兵”，是整个团队能力要提升。首先谈谈对人的看法，我认为人的性格是很难改变的，但是能力是可以提升的。所以对一个团队人员来说，侧重点是要提升他的能力。能力如何提升？有以下几个方面，**一个是组织一些培训/学习、讨论/分享会，部门内部小型沙龙的形式。我认为要重视这个事情，要提高整体上的能力。**再比如组织一些代码的 review，代码 review 可以把它归入到团队管理，也可以归入到前面说的技术管理里面。通过这个分享实际上也是相互去备份，大家都可以参与了解彼此的工作内容，这对于一些人员变动的处理也会是一种方便。

所以大家以后**做计划基本都是要往这个三个方面去考虑，这周或者说这个月，或者说这个季度要做什么，就要往这个三个方向去找。**比如说目前可能是开发非常紧张，现在主要是开发版本，那就要在项目管理这个模块里下多一些功夫，怎么去跟进就是这个阶段应该注重的侧重点。那在相对空闲的时候，那觉得侧重点就要放在技术管理这条线上了，在技术这条线上应该考虑什么问题，包括团队建设也是这个时候去做。所以人员或者项目都是一个起伏的往前走的过程，在什么状态应该侧重在哪条线。很多事情，可能老板都是很急的，可能明天要，或者说后天要，经常都是这样要求的，那这个要求里它应该归结为哪一个类别哪个方向呢？如果有了上述的三个方向，它就很容易归类了，所以这样思考我觉得就会清晰一些。

5、在考察团队人员的时候，需综合考虑其能力和意愿。

团队里人员能力有高低，性格很难改变，所以我**基本放弃改变成员的性格，而着重在提高其能力。**但是每个人性格不一样，我们管理应该做什么呢？管理就类似于带兵打仗一样，要怎么样去打仗？要组织人员去打仗，**首先要了解你这个团队里面每个人的性格特点，或者说能力高低到什么程度，这样你才能够把他们搭配起来。**可能一开始不清楚对方能力，那就是凭感觉来进行人员的组合，但是对人员了解多了以后就会发现，有些人他可能是不相容的，有些人搭配起来却非常顺手，或者说有些人他可能能力很强，但是他对于进度没什么感觉，那这些应该怎么去做？我们首先要定一个目标，我们对团队管理的第一目标就是要把这个事情达成，按照产品的要求做出来。

我们首先要调动这些人，那要怎样去调动呢？首先，每个人性格不一样，**第二能力有高低，要配合搭配。**人的复杂性就在于做起事情来不单单是能力有高低，还有他愿不愿意去做的问题，就是能力和意愿的问题。**管理团队要发挥每个人的主观能动性，这个才是管理的空间。**找到这种方式是能发挥巨大作用的，因为能力你通过一段时间观察基本都能够发现，所以说项目管理它的风险就有两方面，一个就是技术，有没有风险，可能一个东西根本就没办

法研发出来的，或者说我要花一年，你要我一个月研发出来，这是根本不可能的，这个就技术性的风险，难度风险。第二个就是时间风险，你要他 9 月 1 号做不出来，但是可能到 12 月 1 号是能做出来的，可 9 月 1 号就要这东西，怎么办？所以要做好技术管理或者说达成管理的目标这两个风险是最大的，一个是技术风险，一个是时间风险。那**影响这两个因素的主要就是上述的两种能力，技术能力，以及意愿，愿不愿去做，及其投入程度。**

三、 预防项目延期方法的探讨：需求的沟通，流程的完善；目标的细化、分解、各阶段的 review；发生问题后的处理。

1、延期问题导致的原因：技术和产品需求的沟通以及技术任务的分解不够到位、人员的技能不足。

那么问题来了，第一个是，平时大家遇到的一些攻克不下来的问题和规定时间内没有攻克下来的技术难题，是怎么处理的？第二个就是为什么总是延时或者说延时的原因是什么？项目为什么每次评估时间都不准？可能老板会说："你怎么就是做不出来"，你可能会说我目前只能做到这样，但是如果你一而再再而三地延期，老板估计就要跳起来了。为什么就是你评估时间不准，那别人为什么就行。

探讨：其实这个是一个综合问题，不单单是技术的问题，也有可能是产品的问题，双方配合的问题。首先从技术来讲，我们要能看得清整个技术全盘的实现，哪些应该怎么做，各个技术之间的接口怎么去协商，后端前端的协议怎么实现，再分哪些模块，这些模块应该怎么去做，其实心理是有个数的。这样的方法下来，其实做下去应该是没问题的，但实际上经常就有问题发生，那么具体就是这个开发者可能不太成熟，做得不够好，第二个就是粗心，写的代码不优雅，考虑不周，结果测试就很多 bug 了，就不断地返工，还有就是 UI 也写得不好，马马虎虎，毫无逻辑,结果设计那边就会反馈有很多问题，来来回回就花费了很多这种时间，其实这个就是一个程序员需要的磨砺、历练，也可以说是要提高对自己的要求。

2、减少延期方法的探讨：后期一般拒绝修改需求、优化测试周期。

提问：怎样才能去规避、减少这些问题？

探讨：我们目前延期还是很挺严重的。比如三个星期发一个版本，但是一般情况下都会延期的，原因可能有很多种。有些我们后面都想办法规避了，比如说有一种就是改需求的，如果是在整个产品周期后期，产品那边说要改需求，我们是不会改的。就我个人而言，如果说改一些比较简单的，一下子可以搞定的，可能会改，但是我会跟他讲这种情况以后要尽量规避。

还有一个我们现在做的是**优化测试的周期**，测试在周期中经常有一种情况会返工，比如说测试发现了一个 bug，会让你研发来改。发现一个改一个，改了之后还会再来测试，但是这时候可能会发现另外一个 bug，就是原来没有把 bug 全部发现。这样不停地逐个发现，就会造成整个产品的周期一直往后拖。然后现在我们想了一种办法，就是给测试那边一点压力，定一个时间，比如说我们是三周，最后在离版本前剩几天，定个时间，这个时间后一定要把所有的问题全部都测出来。虽然不能保证做到，但是就会给测试一点压力，让他测得更

仔细。等把这些问题全部都测好的时候，从那个时间点之后发现的新问题，如果不是很严重的，我们甚至会放到下一个版本上去再去处理，放到微循环去处理，最终的目的还是想在那个时间段内能够把那个东西做出来。因为**我们现在的迭代都是交互的，产品的研发的时间是交错的，如果直接按照时间来跑的话，都是很稳定的，但是一旦延期了就会造成整个计划都乱套了。**

现在在尝试这种方法，但是效果还不是很好，还是会有延期的情况，我也一直在找原因、找方法，有没有更好的办法能够保证项目的交付，比如说 9 月 1 号发版本，那我们就要 9.1 发版本，想知道有没有一个十全十美的办法能够把这个问题解决。

提问：这个时间是谁来定的？

回答：原来就规划好的。比如说我们原来一直在跑的，一号开始一个新版本，产品项目周期是多少，两、三个星期前就定好了，过了三个星期，那么下一个版本迭代就来了，就一个版本一个版本迭代。

3、减少延期方法的探讨：建立团队的价值观，对产品、技术、人员的评估更加准确，每天 review 项目的进程。

探讨：这里可以回到带团队的问题，在带团队中首先第一件事情就是团队的组建，组建团队首先应该要考虑的是团队的价值观。就是之前提到的**团队各成员的性格、能力各方面都不一样，但有一样东西是一定要是一样的，那就是价值观。**既然你到了我团队，不管做什么，怎么做，你跟我的想法还是要一致的。

在管理或者是组建一个团队的时候首先要做的一件事情，就是建立这个团队的价值观。那这个团队的价值观是怎么来的？之前有提到的团队的目标，我们建立的这个价值观就是要冲着这个目标来做的，所以**第一件事情就是要把团队的价值观传达给整个团队。**再回归到进度和时间技术难度的问题，这个问题我们也遇到过很多次，很多时候都解决不了，但是有一些感想，目前正在尝试着去做一些努力。其实**不能够如期上线，最根本的一个问题就是评估不准，包含产品的评估不准，技术的评估不准，人员的评估不准**等。定了一个三个月要开发出来的周期，为什么这三个月开发不出来，就因为最开始认为，这件事情的需求就这么多，不会再增加了，所以产品评估不准。然后认为这个技术能够实现，却发现这个技术实现不了，所以技术不准。然后把这件事情交给这个人做，他做不出来能力达不到，所以这是对人的评估不准。归根到底就是一点：前期的预期不到位，这个问题要怎么去解决呢？前期是没办法解决前期评估的这个问题的，就算做再多的努力，这个事情在进展的过程中，总在不断的变化的，是预想不到这个事情变化的因素的，所以前期的评估一定是会有不准的，不管你怎么做。

那怎么解决这个问题呢？我们可以在这个过程中不断地去调整，这个可能稍微能够缓解这个不准的一个情况。就比如现在做 4.0 的版本，在前期做方案的时候，我们订了 **deadline**，9 月底要做出来，但是这个时候产品案没出，产品细节没出，技术方案没出，人员没定，但是时间已经定了。没得办法后来产品案出来了，我就不断地去想，不断地去问这个产品到底

要做成什么样子，需求是什么样子，然后不断去调整，每天都要去确定它的每个需求，这是从产品层面的。然后从技术层面也是每天不断地去调整，因为我每天都要思考它的这个需求怎样去实现，这个技术方案怎样才能够落地，要调整技术方案，**每天我都会拉我们组的人去讨论，这个点能不能实现，要怎么做，能不能在预期时间内按照这个技术方案去实现。**然后就是至于人员的话，我现在也是**要求团队里面每个组长至少要每天去了解每个团队成员里面每天要做的事情**，我只能这么做了，我**要精确到团队里每个成员他今天我给他定的目标定的事情他有没有做到。**其实**往往不准就是很多这种每天应该做的事情没有做到，然后到后面产生了一个很大的误差**，这个到后面你再来补这个误差，你发现时间已经不允许，你已经完全拉不回来了，所以我觉得我只能够在前期每一天去发现这个误差，对这个误差进行调整。比如今天你做不完了，是什么原因做不完，这个技术难度发现真的做不下来了，那我要在今天及时地想出一个方案来解决这个技术难题；然后这个产品的需求发现越做越多，要及时地跟产品沟通清楚，你是不是没想清楚这一点，那后面是不是我要继续把你提出来的东西做完，还是说为了保证那个时间，你的需求要适当地去调整。然后对于人员方面，就是我今天发现这个人，这个事情那么简单他却做不完，那就是这个人员他本身的能力是有问题的，那我是否要从其他地方协调人员来解决他这个问题。所以我只能够**每天不断地去 review 这个项目的进程，来确保项目的进度**，如果真的最后都是要延期的话，那我已经没有办法了，这是我在这个过程中的一些感想吧。

探讨：我们现在也是有遇到这个问题，我觉得有一些细节上是可以考虑的，把一个任务分配到一个人的时候，他第一次可以慢可以拖，但是到第二次第三次的时候就不允许了，我们可以从细节上面去做一些调整，从一些制度上做一些硬性的规定，让他变得越来越好，越来越接近我们的这个价值观，这是从这个开发语言上来考虑。然后从我们的管理上来说，我们尽量在前期做好一些细节上的评估，但是我们也很可能是没有办法做到这么细，因为可能是当时方案没有定，但是时间已经定下来，这种情况也是有的，也是难以避免的。这种情况下，我们尽量是要评估得细一点的，这样就会使那个评估的时间会准确一点，这是第二点。第三点就是其实我觉得开发人员的潜力其实都是有的，就是看他投入了多少，这个我们可以用一种比较完善的 KPI 的制度去激励。

探讨：前期就是精确规划，第一就是每天要完成一个小目标，把大目标细化成小目标，然后在执行的过程中，每天去跟踪进度，然后谁没有完成进度，第二天就要去弥补，然后始终要自己去掌控进度，这样的话是基本上能够按时完成。要拖也不会拖太久。

4、减少延期方法的总结：完善沟通流程，精细化目标分解，合适的问题补救措施。

总结：我觉得大家都讲得比较好了，也基本是这些方法，我总结下大家刚才讲到的几个方面，**一个是流程方面的沟通**，比如说技术跟产品这边，怎样去通过一些流程来规避这个问题。**第二方面就是一个目标的分解问题**，刚才都讲到了我可能有一个月的任务，这个一个月里我怎样把目标去分解呢，分解成一个小目标，或者是分解成一个里程碑的东西，**然后每个阶段不断地去 review**，去调整。**第三个方面就是发生问题以后怎样去补救。**因为中途可能都会遇到问题，我们应该怎样去补救它。

刚才我们也讲到我们要预测到这些问题，怎么样才能够预测这个问题呢？我觉得我也没办法讲出一个非常标准的答案，我讲一下我们这边目前怎么做的，针对这个问题，因为首先就是产品，因为产品你说**不改需求好像是不可能，但是我觉得要能够控制在 15%或者 20%以内，不能够说我把 50%的需求都改掉了。**我觉得通过这种方式去逐步规避这个问题，那我觉得后面其他的就可以定了。我觉得**我定出来的时间应该基本都没有偏差，基本可能就偏差一两天，比如说一个月的项目可能最多就差一天或两天，应该就比较极少了，就中途过程中有偏差，但总体时间应该来说是几乎都没有偏差的。**那我觉得我的做法刚才也讲了，我们技术跟产品之间可能一开始没做得那么好，但是后面要相互探讨一些问题，那你说每个需求都会改掉 50%，那你觉得这个需求合格了吗？那我觉得这个问题要从产品方面去找原因，要去沟通，对大多数人来说我们产品的需求要控制在 20%以内左右，这块是这样，这样我们才能控制这个时间。**第二个就是技术跟产品之间的流程、沟通问题了**，我觉得除了开发某个版本以外，我们刚才也讲到了我们互联网现在都很快，可能我开发 1.0 就要考虑 2.0 的事情了，那这个事情我觉得就是**要大致知道，比如说我开发 1.0 以后 2.0 它大的方向在哪里。**刚才大家也讲到思想统一的问题，我觉得**要达到思想的统一，不是说所有成员，只要大部分的成员都能做到就基本可以了**，因为有些人你给他天天讲都没用，有些人你点到他就懂了。所以我觉得这个问题就要做到，我们这个团队大多数的成员至少是知道，比如说我们今天做什么，比如说开发的产品怎么用，我希望就是不光光是从一个实现的角度，要从产品的角度去理解我们在做件什么事情，怎么样能做好它。然后就是我们在做 1.0 的时候，是我会跟产品沟通，比如说是上版本，大概是哪个方向就没有详细探讨可能他也没想出来，但是大概知道是哪个方向了，至少在那这个时候我觉得就能够评估到我未来这个版本至少可能存在的比较大的技术难度的风险在哪里。我觉得有些是预知的，比如说我 1.0 跳到 3.0 可能不知道，但是从 1.0 到 2. 0，那 2. 0 上可能会有一些比较大的技术风险，这样我觉得是很难百分百知道，但基本上我觉得应该至少有一半就知道了。那如果遇到比较大的风险，我就会安排一些人员去一线做一些预研，去规避风险。

第三，我们潮动这边我现在**做了非常详细的开发计划，我的要求就是任务不能是超过一天的，可能少数也会是两天，就是我分解任务基本都是在一天以内。**比如说是四个小时，或者两个小时或者半个小时，就很少有说超过两天，那我会发现一些能力可能比较强的人，为什么也会延时呢？实际上就是因为他没有看着这个目标来做，这个目标就包括我要实现这个东西。另外有个目标，就是我要在规定的时间内实现这个东西，可能他做得很好，但是他需要更多的时间，所以这个问题我觉得就涉及到一个目标分解的问题，那目标分解我觉得是一个 leader 的责任，我们底下也会分为几个小组，我觉得**小组长要跟他底下的人一起去去分解、去定这个东西，把它拎出来，**因为有些人可能能力很强，或者说是做事也都不错，但是他没看着目标走，所以走着走着就偏了，所以我就要求刚才讲到大家要不断去 review，我觉得就是除了这个 review 以外，**前期我就是要求他要把这个任务分解得比较详细一点，也要看着这个目标去走**，那中途如果遇到真的困难，实际上我们也有遇到，比如说可能的一些风险点，因为具体的技术问题，实际上有些东西我也不清楚，那我就会找那些骨干或者说 **leader**，比如说我要开发这个版本有哪些难点，哪些是有风险的，可能有些问题那个 leader 也没预估到，我觉得作为一个总负责的话，我要预测到这个风险基本在哪里，就一旦有问题

应该怎么去调整，怎么去组织力量破解。在我们开发过程中，如果是开发大版本，可能中途都会有一些小的调整，就是因为里面有风险，这个我们也遇到过好多次，但我们也基本能控制好总的时间，总的风险没发生。我们也是天天去 review，我觉得就是要这样去走。

刚才讲的都是一个流程化的问题，另外一个就是中途可能出问题，因为有些人能力非常强，但是他可能不愿意去做，或者说他开小差了，或者说有些人可能能力很强，但是你说他有什么问题，貌似也没什么问题，他也很好人，在一个团队里谁叫他，他也帮忙。问题就是有些人他可能帮其他人实现了目标，但是他自己的没有完成，有时人一旦多了以后这个问题也是比较严重的。那这个问题怎么解决呢？有一个办法就是我们刚才聊的目标的分解，除了这个分解以外，实际上另外一个就是刚才聊到的性格的问题：这个人，他可能能力非常强，但是他就是没往目标走，他就是有这个偏差。所以我觉得**对于我们管理来说，我们除了要能发挥一些人的长处以外，还要有一些方法能让他规避他的短处，但是可能没办法提高短处。**对于这个问题我举个例子，比如说有些人他不断会有偏差，我就会经常问他，因为有些人你问他一次，没用的，你要定时发邮件或者 QQ 或者微信，要经常去找他。**针对某些个别的人，可能你在搭配组合的时候要找一个人，让他去帮助规避这个问题**，这样才能够达到这个目标。比如说有些人，你问他，你那个东西没给我，他说：哦，我忘记了。所以除了惩罚以外，真的非常严重的你也是需要惩罚的，但是惩罚我觉得解决不了问题，因为他性格或者说习惯就是这样，改变习惯我觉得真的是非常非常困难，我主要是会想一些方法去规避它。

四、 提高团队成员主观能动性：从人性的角度出发，发掘其内心需求，进而逐步满足他，让其心安、安心。

另外，我想跟大家探讨一个问题，就是比如说有些人他可能不愿意做了，刚才我们提到的问题就是怎么样去发挥、提升人的能动性。可能有些人他是能做到 100 分的，但实际上他就做了 70 分 80 分，那怎样去让他发挥到 100 或者 120 分。我们都知道人的潜力是非常强的，特别是对于技术人员来说。

1、技术人员的特点：逻辑思维能力强；有想法但不一定表达出来；喜欢写代码，不喜欢被人打扰。

我讲一下我对技术人员特点的看法，这个问题我觉得对于我们组建团队来说还是很重要的。一般来说，**我认为程序员都是非常聪明的，逻辑能力方面都是很强的，但是这里头也有一个问题：他有他的想法，但是他不一定告诉你**，他不像比如说做销售的，或者说做产品的，他会把他的想法分享出来，就是他有很好的想法，或者说他也有自己的想法，但是他可能不一定讲出来，而且就是大多时间可能都不讲出来。另外一个就是其实**对于程序员来说的，他就是喜欢写代码，不喜欢别人来打扰他。**对于我自己来说也是这样，事实上我现在都写代码，我也喜欢写代码，我最喜欢的就是大家都不干扰我，我就静静地写代码，希望能能够按照我自己的想法写下来一个东西，高效地去实现一些功能。我觉得跟机器打交道比跟人打交道要爽，我就跟这个电脑打交道，我觉得跟他对话要比跟人对话要容易，这应该也是技术人员的一个特征。

2、管理的两个要点：人性+流程；人性三需求：成就感（被认可）的需求、财富的需求、权利的需求。

对于团队的管理我觉得还是要回归到人性的问题，**管理我觉得就是一个人性，加一个适当的流程的问题**。那人性是什么呢？我觉得要发挥每个人的主观能动性，就要回归到人的人性上来。针对这个问题大家可以想一下，大家来这里，比如说来爱拍来潮动或是来奥和，你们来这里最想要的是什么，我觉得你们最想要的东西，跟团队其他成员想要的是差不多的。有些人可能想要钱，有的人可能说我想管人，或者说觉得做出来的东西很爽，所以可能每个人表现的形式不同，或者说每个人要的没那么强烈。比如说老板给我这个月加 1 万块钱，我就很爽了，有的人可能加 2000 块钱他就很爽了。但是这个问题从本质上来说它是没有区别的，他都是一个对钱的要求。对于有些人他会问我为什么要发挥 100 分甚至 120 分，我觉得我就做 60 分、70 分，我就能混下去了，这个就是他的需求。所以针对这个问题我觉得还是要回归到一个人的欲望和需求这里来。我觉得发挥人的能动性，我总结的实际上就四个字，就是人的一个心安或者说安心，我觉得做到这样管理就是最好的了，就是要做到团队里的人，要做到每个人都是心安在这里做的，然后尽力去做，去发挥他的潜能。那怎样才能做到这些呢？那就分解出来，也就是他要达到他的目标，比如说我来爱拍、潮动或者是奥和，我来这里干什么？我为了什么来这里，我为什么天天写代码，我觉得就实际上就是回归到人性的问题，那人性有哪些呢？比如说你们**需求是什么，从大的方向上我总结了一个 123 的问题。第一，可能觉得我做出来很爽，那爽是什么？你认可，我就爽，所以我觉得首先他有一个成就感的需求在里面**。做出来会不会有人认可？有人用吗？第二，就是刚才我觉得老板要给我加 1 万块钱工资，如果不加，我就不爽。或者说做出来以后老板给我去旅游，会放我两天假，这个我觉得就很爽，那就是不管是钱还是放假，我觉得可以**归结到一个对于人对财富的追求**。第三，我觉得就是**对于权利的追求**，比如说我一开始是写代码的，但是我希望带着一班兄弟去打仗，这样我会觉得很爽。所以我觉得如果抽象出来的话，他就是一个权利。实际上现在做 leader 没有多大权利，实际上跟其他兄弟差不多一样。但是有些人就喜欢，比如说我觉得从程序员提升到技术经理或者给个总监，我就很爽，像这些就是一个权利的追求在里面。但是我也会跟他讲，大家也知道，实际上现在**职位或者职称，这些其实是最不值钱的**，但是这些也是我们团队建设不可或缺的。这三方面的需求，就是人的欲望，我要让他本来有一百分的，他要做到一百分，甚至要做到 120 分，你就要从这个人的欲望里面去激发他。他才会给你做到。

3、团队的激励：让成员安心和心安；人性是没有代沟的。

这可能每个阶段大家的表现形式不一样。我是 70 后，有些人可能会认为现在跟 90 后很难沟通，他们说 90 后 00 后都变态了，和我以前需求不一样，在座的也有 90 后，我不是说你们了，我就说一个普遍现象，或者说 70 后，我没办法跟 80 后沟通，没办法跟 90 后沟通。我觉得其实还是他没有认识到人的需求在哪里，因为可能每个年代它表现出来的形式不一样。比如说我要一个玩具，或者说我看那个动漫，还是说我看二次元，都没有离开人性的内在需求。所以我觉得要发挥每个人的能动性，要去做团队建设，我觉得还是从人性的角度去考虑这个问题，才能够激发他。比如说有的人认为，你不给钱我都不爽的，我为什么达到

120 分给你，那不可能。那对于这个钱也是可以引导的或者通过其他形式来体现。所以**我觉得对一个团队来说，需要激励，就要从人性着手。我经常跟我团队的人说我希望他来这里都是安心的，就让他安心，然后他会把心放在这个事情上。**这也是我这么些年管理的心得，那其他的方面可能就是一些手段。

刚才讲到比如说我要把目标分解，还有一个要跟进，要 review，要技术沉淀，我觉得就是回归到一个手段问题，因为一开始我就讲了管理为什么难，就是因为它没有标准，那我举一个非常简单的例子，就是你们觉得我要读很多书才会管理，管理实际上可能是天生的。你看那些黑社会管理得多好，你要我干什么就干什么，实际上那个也是管理，军队也是管理，可能黑社会他管理方式实际上也就用了四个字：威逼利诱，实际上它也是没有脱离到对人性需求的把握。

其实不管哪个年代的人或者说哪个程序员，我觉得他都是受这个人性的牵制的，所以我觉得除了要利用好手段以外，还要思考一下就是人性的需求在哪里。因为**我认为人性是没有代沟的，它只是有不同的表现形式，**但是它归根结底它深层次比较深强烈的需求基本就是那几个方面，你发挥好了，人的欲望就上来了，他有了欲望，他加班也就愿意，就是这样。所以我们要发挥他的能动性，光忽悠是没有用，你要说一些他来这里，他能够达到他的目标，他能够得到他想要的东西，那他就心安了，然后他就会把心安在这里，把事情就好，这个是我对于团队管理这么多年来的一个理解。