

15 | 风险管理：不能盲目乐观，凡事都应该有B计划

2019-03-30 宝玉

软件工程之美

[进入课程 >](#)



讲述：宝玉

时长 15:19 大小 14.04M



你好，我是宝玉，我今天想与你分享的主题是：风险管理，凡事都应该有 B 计划。

说到风险，很多人都模模糊糊有一些了解，然而在软件项目中，有风险意识的却不多。比如说：

估算一个模块工作量，程序员总是会给出一个乐观的进度，而最终实现这个模块的时候，却发现总是有些其他的事情发生影响了进度；

一个关键的程序员，突然离职了，导致项目进度停滞。其实早前就有一些迹象，而项目经理没引起重视；

技术负责人很激进的采用了一个最近很流行的新技术，结果做的过程中，发现这个技术还不太成熟，很多坑没法填，导致项目最终失败；

服务器突然挂了，才发现硬盘坏了而数据没有备份，造成巨大的损失。

这些问题其实都和风险相关，如果没有及时发现这些潜在的风险，没有应对方案，轻则导致项目进度延迟，重则导致项目失败，造成重大损失。

在软件工程里面，针对这些可能造成风险的问题，对风险进行提前识别和管理，就可以有效地应对。

什么是风险管理？

风险是指不确定的事件，一旦发生，将会造成消极的影响。风险包含两个方面的内容：

1. 发生后，会造成什么样的损失？
2. 发生的概率有多大？

所以也有人认为：**风险 = 损失 x 发生概率**。

比如说，有一次我负责一个小项目，激进的采用了刚开始流行的 React 框架，如果使用熟悉的 Angularjs 框架，正常来说一个月就能完成，当时很乐观的觉得 React 和 Angularjs 也差不多，时间应该不会多出来太多，就按照一个月时间来做项目计划。

最后到实际开发的时候，发现 React 和 Angularjs 有很多不一样的地方，必须要现学现用，原本计划一个月完成的，最后加班加点拖到一个半月时间才完成。

在这个项目中，使用了新技术就是一个风险：

造成的损失就是导致了进度延误；

发生延误的概率，如果项目开始前估算，大概在 60% 左右，项目进行这个概率就上升到 80% 了。

像软件项目中这样的风险，如果发生后就会变成问题，问题如果没有及时解决，就会影响到项目计划。如果我们能有效的对风险加以识别，加以管理，就能减少对项目的负面影响。

风险管理就是指在项目进行过程中，识别可能的风险，对风险进行评估，并加以监控，从而减少风险对项目的负面影响。

风险管理重要吗？

对于风险管理，现在软件项目中提起的不多，应用的更少。

主要原因还在于大家都比较乐观，多少有一定侥幸心理，都觉得自己不会运气那么差，风险事件正好就发生了。还有就是因为如果要做风险管理，需要额外多做一些工作。

这就跟你买保险的道理一样，风险事件没发生，相当于你买保险的钱白花了，但是一旦发生了，你才知道买保险的钱花的值得。

对软件项目风险的管理，才是体现项目管理水平的地方。我们对比下面几种应对风险的层次来看：

被动应对：风险已经发生，造成了问题才被动应对；

有备无患：事先制定好风险发生后的补救方案，但没有任何防范措施；

防患未然：对可能的风险做出防范，并把风险防范作为项目任务的一部分。

哪一种层次更体现项目管理水平，相信你心中已经有了答案。

拿前不久发生的拼多多被“薅羊毛”事件来举例。

2019 年 1 月 20 日凌晨，拼多多平台出现系统漏洞，用户可以领取 100 元无门槛券。因此开始出现大批用户借此“薅羊毛”，利用无门槛券下单虚拟商品，例如充话费或 Q 币等等，并一度有消息传出，拼多多将因此损失 200 亿元。

这个事件追究原因的话，当然可以说是开发代码没写好，可以说是测试没测好，也可以说运维没有监控好。

但另一个角度讲，如果这个项目的负责人有一点风险意识，做了风险管理，即使出现这样的问题，损失也一定不会这么大。

如何做好风险管理？

培养风险意识

风险管理其实最大的问题不是如何做，而是项目成员缺少风险意识，有了风险意识，才能去识别出来项目中可能的风险，进而去管理风险。

对于培养风险意识，我的经验就是：**项目中的任务，不能盲目乐观，都思考一下它最坏的结果是什么，如果最坏的结果不能接受，就说明要有个 B 计划，考虑风险管理了。**

比如说拼多多的无门槛券，最坏的情况是被无限刷，造成巨额经济损失，那这种结果是不能接受的，就需要考虑风险管理。

如何对风险进行管理？

软件项目风险管理，通常分四步来做。

第一步：风险识别，识别可能的风险

风险识别，就是看项目中有哪些可能的风险，因为只有找出来有可能存在的风险，才会有后续的步骤。

识别风险这种事，经验很重要，因为大部分风险其实都是相似的。以前看 CSDN 总裁蒋涛发过一条微博，内容引发了很多人的共鸣，每一条无不应对着软件项目中的常见风险。

10 个项目死亡的信号：

1. 第一版做太多功能；
2. 太依赖新技术平台；
3. 与公司另一个有份量的产品竞争；
4. 团队人手不足；
5. 复杂的问题，需要复杂的解法；
6. 成员开始隐藏进度落后的事实和原因；
7. 不断更改、增加的需求；
8. 2.0 症候群 - 非要更大、更强、更美；
9. 产品没有市场立足点；
10. 你根本无法解决的大问题。

一个识别风险的方法叫**检查表法**，就是可以把类似于上面这些常见风险收集整理起来，分类列成清单，按照清单去检查对照。

软件项目的风险主要分成以下几类：

项目风险：项目预算、进度、用户和需求等方面的问题；

人员风险：人员离职、人手不足等问题；

技术风险：采用的技术所可能带来的风险；

商业风险：与市场、产品策略等有关商业风险。

你也可以按照上面的分类整理出自己的风险检查表。

另外你还可以借助集体的智慧，定期针对风险问题开一些头脑风暴会议，一起发现可能的风险。另外，要有合适的渠道，让项目成员可以反馈可能发生的风险问题。

第二步：风险量化，对风险进行评估量化

在风险识别出来以后，需要从两个方面去评估：

发生的概率多大？

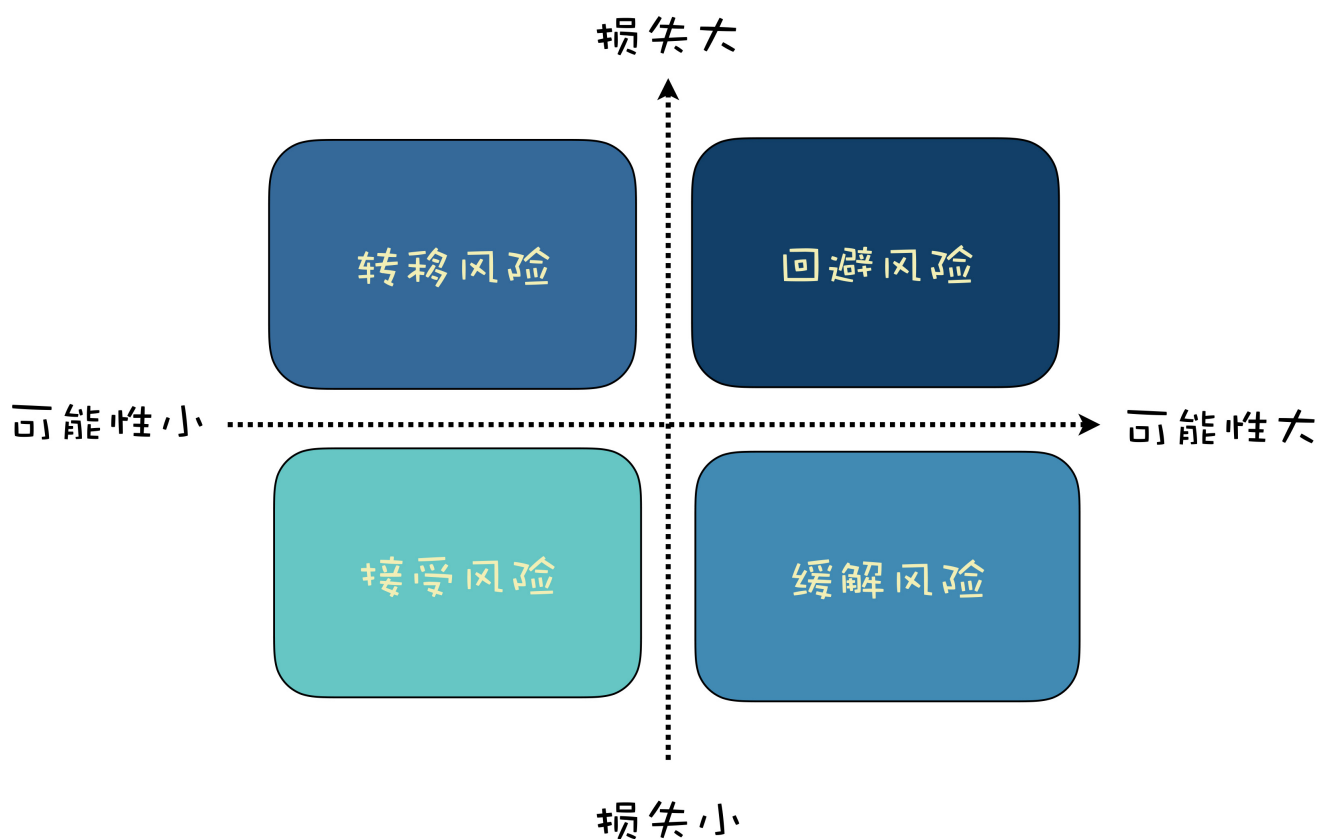
发生后，后果多严重？

对于概率大，后果严重的风险，需要高优先级重点考虑；对于概率不高但后果严重的问题也要考虑，不过优先级略低；对于概率高但后果不严重的风险事件，可以优先级很低或者不考虑；对于概率低后果不严重的，则可以不予考虑。

拿拼多多的“无门槛券”来说，这就属于一个高风险、高概率的事，需要重点考虑。

第三步：应对计划，对风险制定应对策略

在评估后，需要后续进一步考虑的，就要制定好应对的计划。针对风险，主要分成以下几个策略。



回避风险——更改导致风险的方案

回避风险很好理解，就是要对可能发生的风险，放弃或者修改导致风险的方案。这样就从根源上消除了风险，简单而彻底。

就像我前面举的因为使用 React 技术导致风险的例子，可以直接放弃使用 React，用回熟悉的 Angularjs 技术，这样就可以避免技术风险的发生。

但这种方案不一定适合所有情况，例如拼多多“无门槛券”的风险，就无法采用这种方案。

转移风险——将损失转嫁出去

以前玩大富翁游戏，最开心就是有一张“嫁祸卡”，万一遇到倒霉事就可以转嫁到其他玩家身上。转移风险也是这个思路，就是为了避免承担风险损失，将损失转嫁出去的方法。

日常生活中买保险就是一个例子，发生意外，保险公司会帮助赔付。

在软件项目中，举例来说，如果你的团队对于服务器管理不是很在行，有可能会遇到服务器宕机或数据库丢失数据等风险，就可以考虑购买云服务，这样云服务商会帮你解决服务器宕

机或数据库丢失的问题，而且万一宕机或丢数据了他们也会承担一定的责任。

缓解风险——降低风险发生概率或减少可能造成的损失

缓解风险就是在风险发生前采取一定措施，降低风险发生的概率，或者减少风险可能造成的损失。

比如你要担心数据库数据丢失的风险，就需要定期备份数据库；比如你担心核心成员要离职，那就涨点工资，避免人才流失。

拼多多的无门槛券也是个典型的例子，如果对券的消费增加一定的限制，比如说每个用户有领取的上限，一个月不能超过 100 张，每张券不超过 10 元，不能用于购买 Q 币手机话费等虚拟物品，这样即使出现问题，也不会造成很大损失。

我们在《[04 | 瀑布模型之外，还有哪些开发模型？](#)》中学到了“螺旋模型”，也是这种策略：每个迭代都要做一下风险评估，再决定项目是不是继续。

接受风险——明知山有虎偏向虎山行

还有一些风险本身很难避免，或者去应对这个风险的成本超过风险发生后造成的损失，那么就没必要应对，直接选择承担风险后果就好了。

比如说前面说的采用新技术 React 导致进度延迟的案例，这个风险虽然有很大概率发生，但是进度延迟的影响可以接受，并且让团队今后在技术栈上多了新的选择，长远来看对项目更有利，那么这个风险就是可以接受的，还是可以继续做下去。

回避风险、转移风险、缓解风险、接受风险，以上就是针对风险提前准备的一些应对策略，实际项目中，可以根据实际情况来灵活运用以上策略，有效应对风险，减少可能损失。

第四步：风险监控，对风险进行监控预警

风险在没发生的时候并不会变成问题也不会造成损失，如果风险可以监控，可以预知风险即将发生，或者可以在风险发生后，第一时间知道，那么就可以马上对风险进行干预，避免变成更大的问题。

要做好监控，第一要能对监控的内容量化，第二要设置阈值，第三就是要有后续的报警和处理机制。

很多公司都已经建立了自己的监控系统，将关键数值量化，并设置阈值，超过阈值后自动触发报警机制。

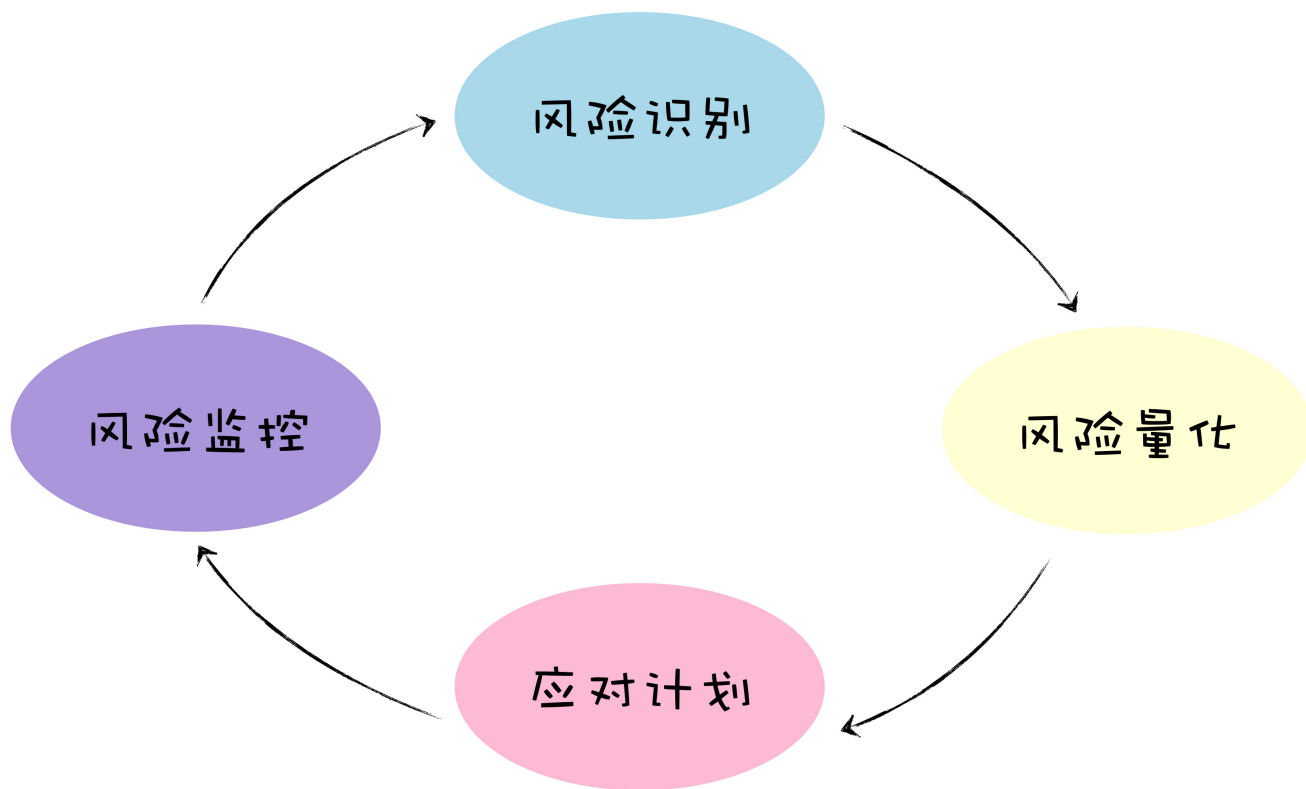
一个简单的例子就是服务器宕机了，监控系统发现机器没响应了，自动通过邮件、短信、电话等方式通知正在值班的人员。

还有稍复杂一点的方式，像网络服务，可以监控每一次请求结果的状态码，统计请求的成功率。如果单位时间内，服务出错的比例低于阈值，那说明服务是正常的；如果错误比例超过阈值，那说明是出现了问题，需要报警通知相关人员，马上处理。

再回到拼多多“无门槛优惠券”的例子，其实有很多数据可以监控到，比如说单位时间内优惠券使用数量，商品销售数量等，如果结合一些可视化视图，应该可以直观的看到当时有大量虚拟商品销售，大量的优惠款被使用。

如果针对优惠款设置了阈值，例如每分钟使用超过 1000 张就触发报警，那么也可以及时发现无限刷的问题，避免造成更大损失。

以上四个风险管理的步骤是一个连续循环的过程，在整个项目期间，都要持续的对风险进行识别，对风险量化，对于风险采取应对计划，对风险进行监控。



(注：项目风险管理过程)

总结

今天带你一起学习了软件项目管理中的风险管理知识。软件项目中的风险就是指那些不确定的但是可能会造成消极影响的事件，通过对风险的管理，可以有效降低风险发生的概率，减少风险发生后的损失。

软件项目风险管理包括风险识别、风险量化、应对计划和风险监控四个过程，这四个过程是一个循环的过程，需要在项目中持续进行。

希望你在学习后，能提高风险意识，不能盲目乐观，凡事都应该有 B 计划。并且能将学到的风险管理知识应用到项目中，做到对可能的风险了然于胸，未雨绸缪，运筹帷幄。

课后思考

按照风险管理的知识，可以对你目前项目中，按照发生概率或造成后果列出 1-3 条风险，并思考如何能管理这些风险。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。

软件工程之美

重新理解软件工程

宝玉

Groupon 资深工程师
微软最有价值专家



新版升级：点击「 请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

© 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 14 | 项目管理工具：一切管理问题，都应思考能否通过工具解决

下一篇 16 | 为什么你不爱写项目文档？

精选留言 (12)

写留言



邢爱明

2019-03-30

2

目前负责在公司推行敏捷开发，第一个试点推行，也可以认为是一个项目。

个人识别出三个风险：

1. 这个试点项目的项目经理担任PO角色，虽然已多次沟通，但对敏捷的认识比较浅，觉得自己不需要做什么改变，其他人的工作不影响项目交付就行了，参与度较低。
2. 团队中开发的人员多为外包人员，对敏捷开发所需要的技能了解很少，单元测试和持...

展开

作者回复：我觉得你提到的借人参与两到三个迭代的方案是非常简单有效的办法，一个敏捷团队，如果大家没这个意识和水平，是很难推动的，但是当有几个人一起，走顺了流程，后面相对就照葫芦画瓢容易多了。

另外辅助培训也是很有必要的，有了一些基本的概念，再执行起来阻力会小很多。

对于多年的项目经理转PO，是要一个过程的，毕竟已经喜欢了以前的一套东西，一时之间很难转变观念。沟通时，建议客观分析利弊，优缺点，尤其是将一些事情和他的利益挂钩，会更有参与度一些。比如说让他意识到：这个试点项目成功对他未来的绩效、职业前景都有好处。



一路向北

2019-04-01

1

老师说的风险几乎都遇到过。但是实际每次定计划的时候几乎没有一次定过B计划，后面实施方案的时候，确实需要多思考一个B计划。

不过，想来之前的一些拖延比较长时间的项目，对项目的一些变化的估计是有的，只是在计划制定的时候，总是会对一些可能会遇到的困难说克服一下，加加班就过去了之类的话，其实是对计划不确定性的一种逃避。

展开

作者回复:

很多时候没有做Plan B，只是没意识到风险的存在！等你意识到风险的存在，以后就会条件反射的要考虑Plan B。

在实践中自然会摸索出很多适合你自己的应对风险的方案。



alva_xu

2019-04-01

1

项目的不同时间节点，项目风险及其处理手段也是不一样的。所以，在讲风险管理的时候，还要加一个时间维度。老师能不能就这个维度来谈谈风险及管控处理方法？

展开

作者回复: 其实要考虑时间维度，你只要把时间范围成本三要素的约束加上就好了。

因为时间变了，这三要素的约束也在变。

给你举个例子：一个创业公司，人少缺钱，这时候人就是个很大的风险，有人离职项目就很危险，这其实本质就是三要素的成本；等到熬过这阶段，进入发展阶段，活下来有钱了，人也多了，相对来说，成本就不是最大的约束了，人的风险就没那么大了，这时候就是求快，时间会变成约束，所以如果你的技术和架构更不上开发的效率，就会成为新的风险。



shine

2019-03-30

👍 1

1.进度/人员风险：项目组开发人员2/3是实习生。技术不熟练，需求理解的偏差可能会导致提交的代码的质量技术债务高及提交时间点延迟，影响进度

发生概率：80%

解决办法：1.技术上需要加强培训2.通过Code Review早期发现代码中的BUG

...

展开 ∨

作者回复: 🐼 这些风险都很真实。

给你一些应对方法上的建议供参考。

1. 实习生比例高问题我以前遇到过，靠培养是很慢的，再聪明也需要两三年时间才能独当一面。建议要适当社招补充有经验的程序员，不然管理者和现有的资深程序员会很辛苦，要大量时间救火，短期看大量实习生省钱，长时间看其实成本更高的。

2. 需求评审是很必要的，需要落实，而且要安排多次，刚开始有初步需求文档的时候就要做，这样修改成本低！另外建议产品设计要多用原型设计，原型设计是很好的避免需求模糊的方法，也可以帮助产品设计事先把很多问题想清楚。

3. 还是在第一条里面提到的，不仅要培养人，更要招人开人。



kirogiyi

2019-03-30

👍 1

非常感谢宝玉老师每一次地回复和解答，使我能有更多感悟和进步。

风险管理是否到位，取决于管理者对整个项目完整度的认识程度，认识越深刻，考虑越全面，就越容易去发现并预防潜在风险的发生。

...

展开 ∨

作者回复: 关于需求优先级的的问题，这个我建议你试试敏捷开发，或者采用迭代开发，因为每个Sprint时间固定，所以逼着你每次Sprint计划时选择优先级最高的需求，这样其他无关紧要的需求可以无限后延。

技术选型是个很重要的事，后面还会再讲讲，希望下次选型时能跟慎重的分析论证。

做项目，想不犯错误也是不现实的，最重要是每一次比上一次做的更好，上一次犯的错误争取不要再犯，这样自然能一次比一次做的更好。



Charles

2019-03-30

👍 1

1. pc用户管理中心后台采用ant design中后台框架，当时可能考虑体验相对较好自己设计不一定能做好，而且前后端分离可以分担后端人少的压力可以类似移动端的开发方式，复用大部分API，但是这里感觉有一个风险就是客户那边浏览器环境没有做足够的调研，可能需要用户升级浏览器，本身也是一个很大的体验影响

...

展开 ∨

作者回复: 赞，有风险意识是很重要的一步👍
后面就可以针对性去识别和预防



谭鹏

2019-03-30

👍 1

在项目中使用了React native，那是React native还不太成熟,那但是项目不紧急也不算太大,出了问题 改成原生开发 也不会浪费太长时间.继续使用React native 能丰富团队的技术栈，所以接受风险

展开 ∨

作者回复: 说到这个问题，回头技术选型那篇还要再讲讲。



Tiger

2019-04-14

👍

老师，我们做的项目外包，项目组的人数是固定的，每次都是项目组要离职一个才会再招一个人进来补充，这种情况无法培养技术后备，这种人员风险怎么把控？

展开 ∨

作者回复: 这种确实有点困难, 有两种策略你可以考虑:

1. 减少对人的依赖, 让人来了跟流水线工人一样可以马上上手。

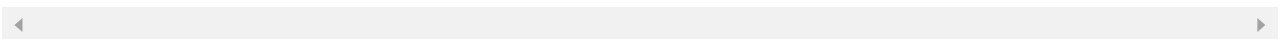
如果你的项目类型比较类似, 其实可以考虑将相同部分通过架构简化, 通过配置或者定制化适用于不同项目。

补充自微博回复:

@bobyuxinyang: 尽量开发流程标准化, 然后文档, 文档, 文档。。。

2. 培养现有的人, 提升现有人能力, 归属感, 减少离职率。

都不容易做到, 但都可以试试, 或者你也可以想到更好的办法。



javaadu

2019-04-07



风险管理包括两个方面: 降低风险发生德概率; 降低风险发生后的损失

展开 ∨



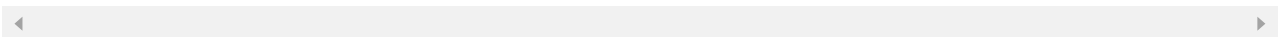
dancer

2019-03-31



在做需求分析、软件设计时, 就要识别出可能潜在的风险点, 并且对风险也做相应的设计, 以及监控报警。要把风险也当作项目的一个组成部分。

作者回复: 是的, 包括做技术选型的时候, 也一样要考虑风险。



小先生

2019-03-31

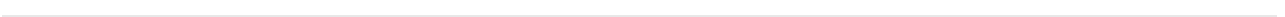


应对风险的四种办法。

回避风险, 转移风险, 接受风险, 缓解风险。

根据不同情况进行应对。

展开 ∨





小先生

2019-03-31



今日得到。

平常项目进展中，具备风险意识。

通过风险识别，风险量化，风险应对，风险监控来进行风险管理。

展开 ∨