# 37 | 故障管理：谈谈我对故障的理解

2018-03-14 赵成

赵成的运维体系管理课 进入课程 >

**讲述：黄洲君**
时长 08:43 大小 4.00M

对于任何一个技术团队来说，最令人痛苦、最不愿面对的事情是什么？我想答案只有一个，那就是：故障。

无论是故障发生时的极度焦虑无助，还是故障处理过程中的煎熬痛苦，以及故障复盘之后的失落消沉，都是我们不愿提及的痛苦感受。在海外，故障复盘的英文单词是 Postmortem，它有另外一个意思就是验尸，想想就觉得痛苦不堪，同时还带有一丝恐怖的意味。

写故障相关的文章，也着实比较痛苦。一方面回顾各种故障场景，确实不是一件令人愉悦的体验；另一方面，故障管理这个事情，跟技术、管理、团队、人员息息相关，也是一套复杂的体系。

我们看 Google SRE 这本书（《SRE：Google 运维解密》），绝大部分章节就是在介绍故障相关的内容。其实看看这本书就能明白稳定性和故障管理这项系统工程的复杂度了，而且从本质上讲，SRE 的岗位职责在很大程度上就是应对故障。

所以，接下来的几期文章，我会谈谈我对故障管理的理解，以及一些实际经历的感受，也希望我们每一个人和团队都能够在故障管理中得到涅槃重生。

今天，先谈谈我们应该如何来看待故障这个事情。

## 系统正常，只是该系统无数异常情况下的一种特例

上面这句话，来自 Google SRE 这本书，我认为这是一个观点，但是更重要的，它更是一个事实。所以，正确理解故障，首先要接受这个现实。

故障，是一种常态，任何一个软件系统都避免不了，国内最牛的 BAT 避免不了，国外最牛的 Google、Amazon、Facebook、Twitter 等也避免不了。业务体量越大，系统越复杂，问题和故障就越多，出现故障是必然的。

可能你会有疑问，既然他们也存在各种故障，但是在我们的印象中，好像也没经常遇到这些大型网站整天出问题或不可访问的情况，这恰恰说明了这些公司的稳定性保障做得非常到位。

这里有一个非常重要的体现，就是**Design for Failure**的理念。**我们的目标和注意力不应该放在消除故障，或者不允许故障发生上，因为我们无法杜绝故障。所以，我们更应该考虑的是，怎么让系统更健壮，在一般的问题面前，仍然可以岿然不动，甚至是出现了故障，也能够让业务更快恢复起来。**

其实对这个理念的实践，我们在前面都已经介绍过了，比如限流降级、容量评估以及开关预案等技术方案的稳定性保障体系，这些技术方案本质上并不是为了杜绝故障发生，而是为了能够更好地应对故障。

同样的，我们刚提到的那些国内外超大型网站，之所以能够保持很高的稳定性和业务连续性，恰恰是说明他们在**故障隔离、快速恢复、容灾切换**这些方面做得非常优秀，一般的问题或故障，根本不会影响到业务访问。

所以，转变一下思路，重新理解系统运行的这种特点，会给我们后续在如何面对故障、管理故障的工作中带来不一样的思考方式。

## 故障永远只是表面现象，其背后技术和管理上的问题才是根因

简单表述一下，就是永远不要将注意力放在故障本身上，一定要将注意力放到故障背后的技术和管理问题上去。

这里的逻辑是这样的，技术和管理上的问题，积累到一定量通过故障的形式爆发出来，所以故障是现象，是在给我们严重提醒。

有时我们过分关注故障本身，就容易揪着跟故障相关的责任人不放，这样会给责任人造成很大的负面压力，进而导致一些负面效应的产生，这一块在后面我还会专门分享。

与之对应的改进措施，往往就容易变成如何杜绝故障。前面我们讲到，从现实情况看这是完全不可能的，所以就容易输出一些无法落地、无法量化的改进措施。

你可以思考一下，面对故障的时候，是不是经常出现上述这两种情况。

所以，想要更好地应对和管理故障，当故障发生后，我们需要考虑的问题应该是其背后存在的技术和管理问题。这里和你分享我自己在故障后的复盘中，经常会反思和提出的几个问题。

1. 为什么会频繁出故障？是不是人员技术不过硬？人为操作太多，自动化平台不完善，操作没有闭环？代码发布后的快速回滚措施不到位？
2. 为什么一个小问题或者某个部件失效，会导致全站宕机？进一步考虑，是不是业务高速发展，技术架构上耦合太紧，任何一个小动作都可能是最后一根稻草？是不是容量评估靠拍脑袋，系统扛不住才知道容量出问题了？是不是限流降级等保障手段缺失，或者有技术方案，但是落地效果不好？
3. 为什么发生了故障没法快速知道并且快速恢复？进一步考虑，是不是监控不完善？告警太多人员麻木？定位问题效率低，迟迟找不到原因？故障隔离还不够完善？故障预案纸上谈兵？
4. 管理上，团队成员线上敬畏意识不够？还是我们宣传强调不到位？Oncall 机制是否还需要完善？故障应对时的组织协作是不是还有待提升？

总结下来，任何一个故障的原因都可以归结到具体的技术和管理问题上，在故障复盘过程中，通常会聚焦在某个故障个例上，归纳出来的是一个个非常具体的改进措施。

用一句话总结：“**理解一个系统应该如何工作并不能使人成为专家，只能靠调查系统为何不能正常工作才行。**”（From SRE，by Brian Redman）

最后，作为管理者，我会问自己一个终极问题：**下次出现类似问题，怎么才能更快地发现问题，更快地恢复业务？即使这一次的故障应对已经做得非常好了，下次是否可以有更进一步的改进？**

这个问题，会促使我个人更加全面地思考，且能够关注到更全局的关键点上。比如，是不是应该考虑有更加完善的发布系统，减少人为操作；是不是应该有整体的稳定性平台建设，包括限流降级、开关预案、强弱依赖、容量评估、全链路跟踪等子系统，以及建设完成后，应该如何一步步的落地；还有，故障预案和演练应该如何有效的组织起来，毕竟这些是从全局考虑，自上而下的一个过程。

## 最后

再表达两个观点。

**第一，出问题，管理者要先自我反省。**不能一味地揪着员工的错误不放，员工更多的是整个体系中的执行者，做得不到位，一定是体系上还存在不完善的地方或漏洞。在这一点上，管理者应该重点反思才对。

**第二，强调技术解决问题，而不是单纯地靠增加管理流程和检查环节来解决问题，技术手段暂时无法满足的，可以靠管理手段来辅助。**比如我上面提到的就基本都是技术手段，但是要建设一个完善的体系肯定要有一个过程，特别是对于创业公司。这时可以辅以一些管理措施，比如靠宣传学习，提升人员的线上安全稳定意识，必要的 Double Check，复杂操作的 Checklist 等，但是这些只能作为辅助手段，一定不能是常态，必须尽快将这些人为动作转化到技术平台中去。

这样做的原因也很明显，单纯的管理手段还是靠人，跟之前没有本质区别，只不过是更加谨小慎微了一些而已。同时，随着系统复杂度越来越高，迟早有一天会超出单纯人力的认知范围和掌控能力，各种人力的管理成本也会随之上升。

今天和你分享了我对故障这件事情的理解，期望这样一个不同角度的理解能够带给你一些启发，欢迎你留言与我讨论。

如果今天的内容对你有帮助，也欢迎你分享给身边的朋友，我们下期见！




上一篇　36 | 稳定性实践：全链路跟踪系统，技术运营能力的体现

下一篇　38 | 故障管理：故障定级和定责

## 精选留言 (1)

写留言

**yan**
2018-06-04

1

十分认同此文章所说的话，现在有些部门或团队出了问题之后就紧紧抓住那个具体责任人不放，又或者是仅仅关注流程.