SRE 核心理念与方法论

核心理念

核心理念一：在保障服务slo的前提下最大化迭代速度

1. 产品和运维团队间最主要的矛盾是迭代创新的速度与产品稳定程度之间的矛盾

2.sre使用"错误预算（error budget）直面这种矛盾

100%可用并非正确的才靠性目标：任何软件系统都不是，也不应该追求做到100%可靠

基于用户期待的可用性标准建立一个合理的可靠性目标，错误预算="1−可

靠性目标"

产品研发部门和sre部门可以在这个范围内将预算用于新功能上线等任何与产品创新相关的事务

3.基于错误预算，可靠性目标不再是"零事故运行"

生产事故也不再是一件坏事，而是创新流程中一个不可避免的环节

产品研发团队和sre团队通过协作共同管理这些事故

衡量可靠性目标的合理性时需要考虑的因素

1.基于用户的使用习惯，服务可靠性要达到什么程度用户才会满意?

2.如果这项服务的可靠程度不够，用户是否有其他的替代选择?

3. 服务的可靠程度是否会影响用户对这项服务的使用模式?

核心理念二：确保长期关注研发工作

1.google将sre团队的传统运维工作限制在50%以内，余下的工作时间要投入到研发中

2.无法满足该目标时，采取一些暂时性的措施将运维压力转移回开发团队处理，例如将生产环境发现的bug和 产 生的工单转给研发管理人员去分配将开发团队成员加入到轮值on-call体系中共同承担轮值压力

3.这种措施可激励研发团队设计，构建出不需要人 人工干预，可以自主运行的系统

4.所有的产品事故都应该有对应的复盘

事故发生，发现，解决的全过程

事故的根因，预防或者优化解决方案

从本质上来说，sre更倾向于通过设计，构建自动化工真兼取代人工操作

1. 基于软件工程的思想和方法论完成之前由运维团队手动完成的任务

2.sre投入到传统运维工作的时间比例不得高于50%,且倾向于通过构建能够自主运行，自动修复问题的系统，

从而消除基本的运维工作

3.sre能够消除研发与运维团队之间的冲突焦点，进而提高整个团队的产出水平

构建一个满足可靠性目标的的系统

影响系统可靠性的常见因素

■人员误操作

■雪崩效应

■未经充分、完整测试的版本发布

■基础设施故障及定期升级维护

■……

SRE的终极责任是保证服务的正常运行，为达成目标，SRE需要完成一系列工作

开发监控体系

处理紧急事件

预测需求和规划系统容量

确保事故根因被跟踪修复

管理各种类型的变更操作

设计和研发运维工具及大规模的软件系统

参与产品设计，确保其符合可靠性实践标准和最佳实践准则

SRE方法论

设计周全的监控体系

合理的指标设计和告警策略是提升系统整体可用性的必要手段

■监控系统不应该依赖人工来分析告警住处，而是由系统自动完成，仅当需要用户操作时，才发送通知

监控系统应该只有三类输出：紧急警报（需要立即处理）、工单（在给定的时间窗口内处理）和日志（调度和事后分析时使用）

应急事件处理

时间和经验一再证明，系统不但一定会出问题，而且会以预料之外的方式出问题，但所有问题总会有解决方案

■组织无论大小，其关键特质都会展现在对紧急事件的应对策略上

■任谁都很难良好地处理紧急情况，除非经过日常的、体系化的实战训练

■合理的策略是，建立和维护一套完备的训练和演习流程，并预备一批专注投入的人

■通常，人工操作是故障恢复过程最为消耗时间的因素，**建设具有“自愈”能力的系统**是首要目标

■不可避免地需要人工介入时，要有记录有清晰调度步骤和分析方法的“**运维手册**”指导快速修复

◆运维手册及其使用要确保on-call的所有成员熟练掌握

- ◆日常要有例行的故障应急演练

Google SRE的工作重心之一就是运维手册的维护

变更管理

大约70%的事故都由变更所触发

■最佳实践是要能够自动化地完成以下任务

- ◆采用**渐进式发布**机制
- ◆迅速而准确地**检测**到问题
- ◆出现问题时，要能安全、迅速地执行回滚

需求预测和容量规划

■需求预测和容量规划简单来说就是保障一个业务有足够的资源冗余度去服务预测中的未来需求

■容量对系统可用性来说至关重要，SRE必须主导容量规划的过程及资源部署的过程

■一个业务的容量规划要考虑两个方面的因素

- ◆自然增长：用户增长、流量增长等导致的资源用量上升
- ◆非自然增长：新功能发布、商业推广等

容量规划的步骤

- ◆预测模型要能准备预测自然增长
- ◆要合理、及时、准备统计各类可能的非自然增长
- ◆必须要有周期性的压力测试，以获取系统的准确容量

资源部署（provisioning）

■资源部署是指发生在容量规划发生变动的情况下要执行的资源变更

■部署新资源仅应该在必要的情况下进行

◆资源通常有较高的成本

◆部署和配置过程必须要确保能正确执行完毕

新资源的部署与配置是一个相对危险的操作

增加现有容量，通常需要启动新实例甚至是整个集群，还可能会导致大幅修改现有的　　集　群配置（配置文件、负载均衡、网络等）

◆ 同时还要执行一系列的测试，以确保新上线的资源可以正确地服务用户

效率与性能

对于商业组织来讲，必然追求高效地利用各种资源，以降低系统的总体拥有成本

■服务的利用率指标与服务的工作方式以及为其置备容量有关，于是，容量配置策略也就事关总体拥有成本

■业务在资源层面的需求通常由**用户需求**（流量）、**可用容量**和**软件的资源使用效率**驱动

度量服务风险：如何衡量系统可用性?

度量服务风险的方式

■度量服务风险承受能力的主要标准在于“计划外停机”的可接受水平

■计划外停机时间由服务预期的可用性水平体现，通常用数字“9”的个数来表示，例如99.9%或99.99%

衡量系统可用性的公式

时间维度（基于时间的可用性）：Availability = Uptime / (Uptime + Downtime)

◆对于配备了“故障隔离”机制的大型分布式系统来说，该测算方式没有实际价值，因为服务几乎总是可用

◆但部分用户的请求可能会因为局部故障而无法得到正常响应

请求维度（基于请求的可用性）：Availability = Successful requests / Total requests

故障的测量和判定

■衡量指标：即用于进行衡量的角度

■衡量目标：即在特定衡量角度上，用于区隔正常与不正常的临界值或阈值

■持续时长/统计周期：在特定衡量指标上，其值处于不正常区间的时长

◆短暂异常，持续时间较短，通常不被认作故障

◆但频发性的短暂异常，就应该被视为稳定性不佳

◆时间维度的可用性衡量机制无法对此场景进行精确描述