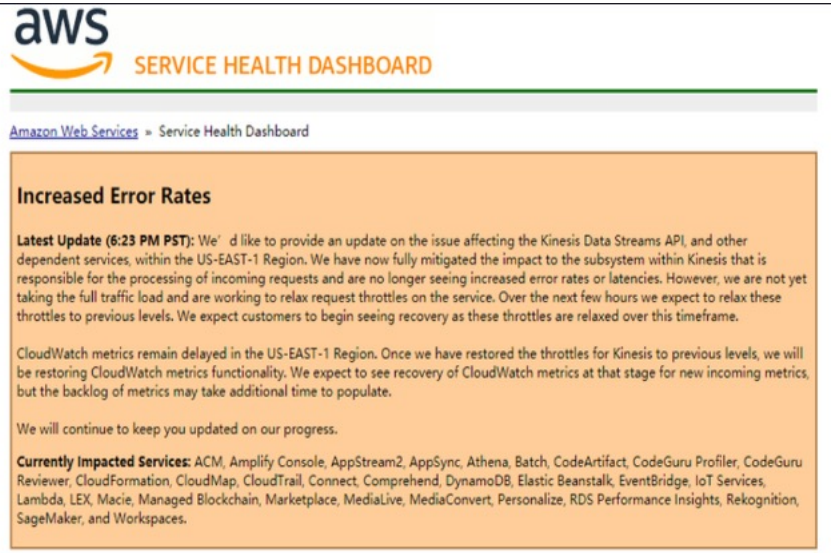
永辉超市-故障演练实战与落地

2021年11月26日

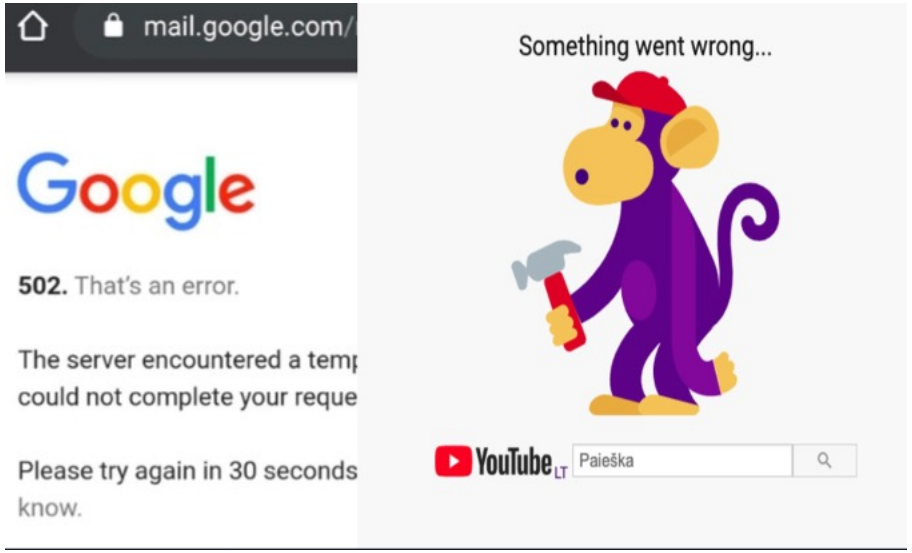
目录

故障演练落地

云厂商故障



2020.11.25 AWS北弗吉尼亚服务大面积故障，多个云产品服务受到影响，问题定位4个小时，完全修复花费15小时。



2020.12.15 谷歌出现今年第三次大规模宕机，
宕机大约45分钟，波及20亿用户，预计损失170万美元广告收入。

- 2017年10月30日，阿里云部分服务器有30多分钟的时间无法正常访问，起因是电力故障。
- 2019年3月23日， 腾讯多个产品出现大规模宕机，上海南汇网络光纤因施工被意外挖断。

要具备“用好云”能力，同时也要具备“面向云容灾”的能力

Netflix 宕机事件故事

- 2008年自建机房意外发生故障，让Netflix业务停了整整3天时间，Netflix考虑将所有数据中心搬到AWS云端；
- 2012年12月24日，AWS弹性负载均衡服务中断导致Netflix网站服务受到中断影响。
- 2015年9月20日，AWS数据中心平台上的20多种服务开始出现故障后，互联网上的一些知名网站和应用系统随之间歇性地无法使用，而Netflix网站可以很快恢复服务，问题来了他们问什么可以快速恢复故障， Netflix 这些年做了什么？

故障演练领域重要发展时间线

2010

Netflix Eng Tools团队开发出了Chaos Monkey。当时Netflix从物理基础设施迁移到AWS上，为了保证AWS实例的故障不会给Netflix的用户体验造成影响，他们开发了这个工具，用来测试系统稳定性。

2012

Netflix在 GitHub上开源了Chaos Monkey，并声称他们“已经找到了应对主要非预期故障的解决方案。通过经常性地制造故障，我们的服务因此变得更有弹性。”

2014

Chaos Monkey的升级版本FIT诞生，实现微服务级别的故障注入。

2016

混沌工程(Chaos Engineering)的概念通过《Principles Of Chaos》这本书被提出。面向失败设计也逐渐成为构建一个高可用架构产品的基本要求，混沌工程也在更多公司进行实践。

2018

混沌工程（Chaos Engineering）成为**CNCF**的一个新的技术领域。

2019

阿里巴巴把集团内部故障演练工具进行整合开源出ChaosBlade项目，目前社区生态比较活跃，覆盖故障场景比较全面。

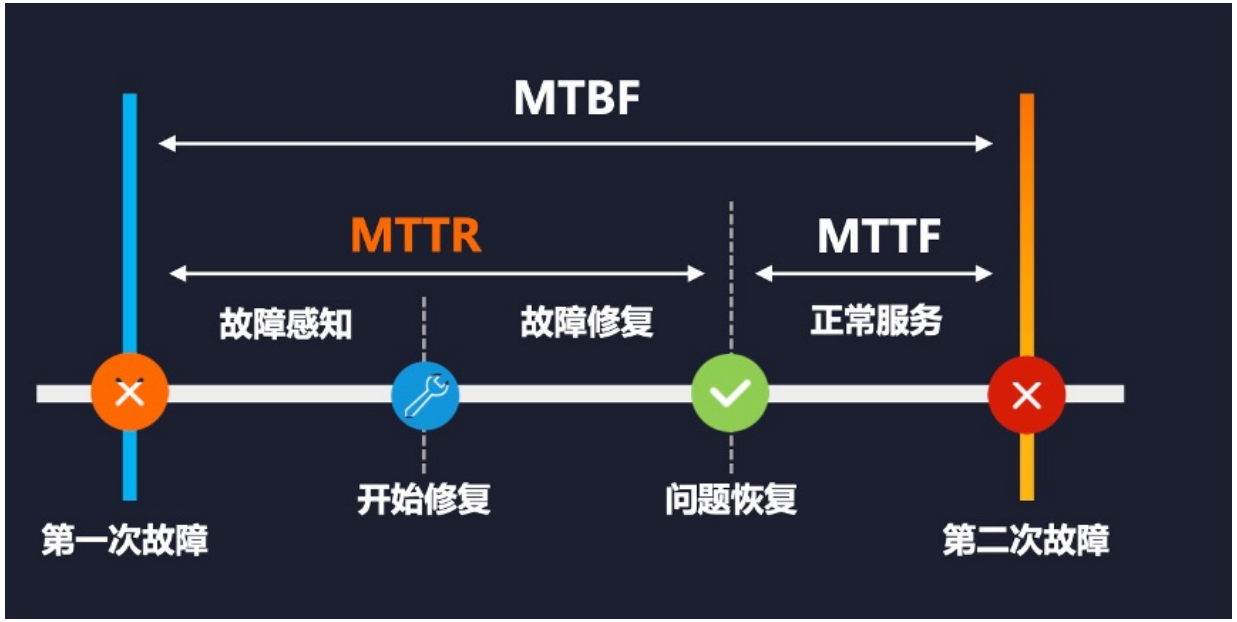
2020

PingCap引入混沌工程项目用于测试TIDB在各种极端场景下数据产产品的可用性和健壮性，
并推出Chaos Mesh开源项目和进行社区运维，这个项目已经成功进入 CNCF 沙箱托管项目

Chaos Engineering

Chaos Mesh | ChaosToolkit | ChaosBlade | chaoskube | Gremlin | Litmus | POWERFUL SEAL | steadybit

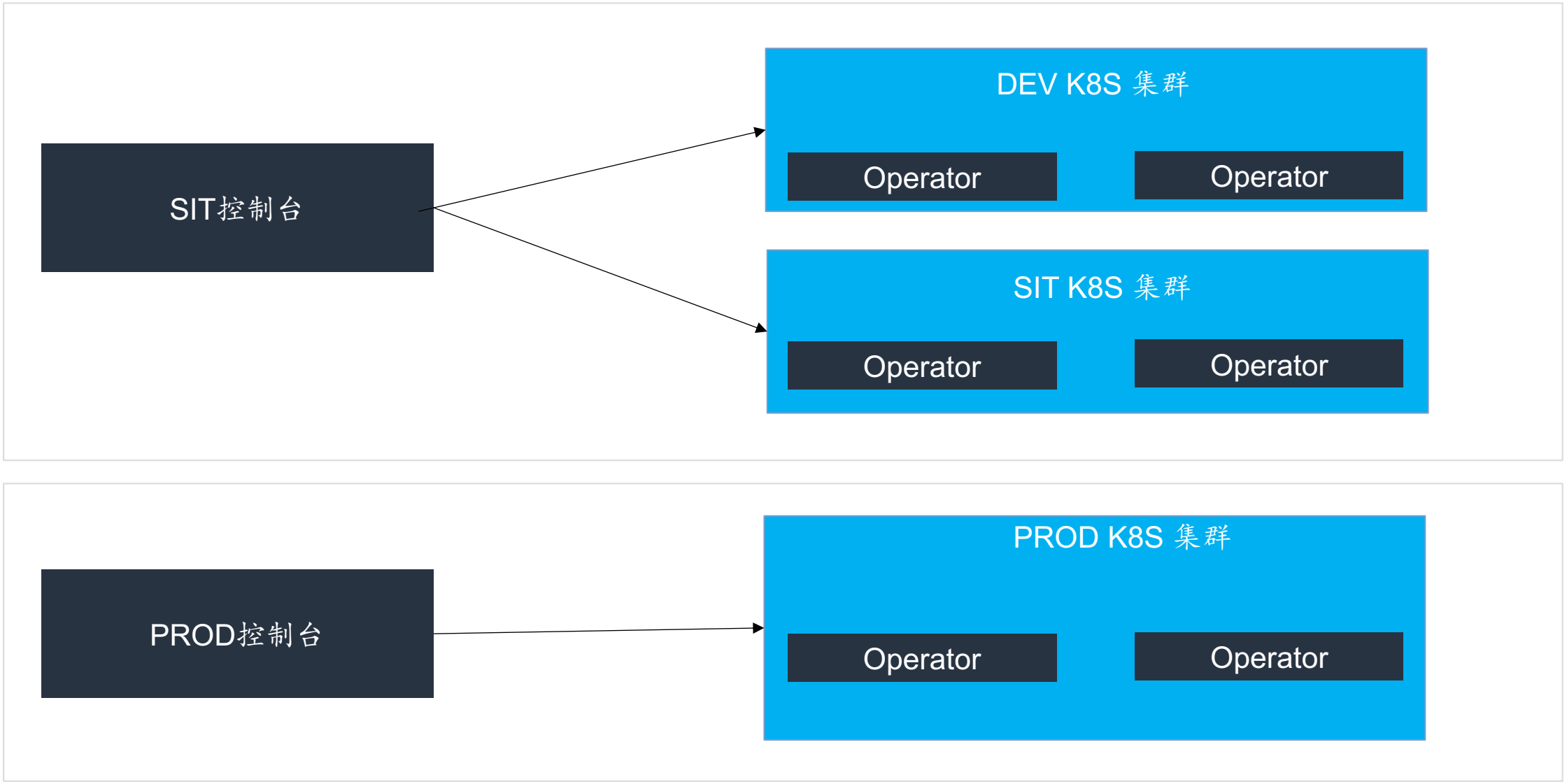
故障演练价值



平均恢复时间 MTTR	• 发现缺失的系统监控、告警，缩短故障发现时间 • 提高技术人员应急效率，缩短故障解决时间
平均无故障时间 MTTF	• 历史故障持续进行回放，避免相同的故障再次发生 • 架构薄弱点持续进行验证，避免架构稳定性衰弱
平均时效间隔 MTBF	• 通过提供平均无故障时间和降低平均修复时间，可以有效提升平均时效间隔

L3	L2	L1	L0
1小时	2小时		
2小时			
0.5小时	1小时	2小时	
1小时	2小时		
2小时			
0.5小时	1小时	2小时	
1小时	2小时		
2小时			
5分钟	0.5小时	1小时	2小时
0.5小时	1小时	2小时	4小时
1小时	2小时	4小时	
5分钟	0.5小时		
0.5小时	1小时		
1小时	2小时		
0.5小时	1小时	2小时	
1小时	2小时	3小时	
2小时			
0.5小时	1小时	2小时	3小时
1小时	2小时	3小时	4小时

平台落地环境


SIT控制台
DEV K8S 集群
Operator
Operator
SIT K8S 集群
Operator
Operator
PROD控制台
PROD K8S 集群
Operator
Operator

平台落地实战

版本组合搭配

Chaosblade v1.5.0

Chaos Box v0.4.2

优点：

- 版本发布周期长，稳定
- 集群接入比较简单

平台落地实战

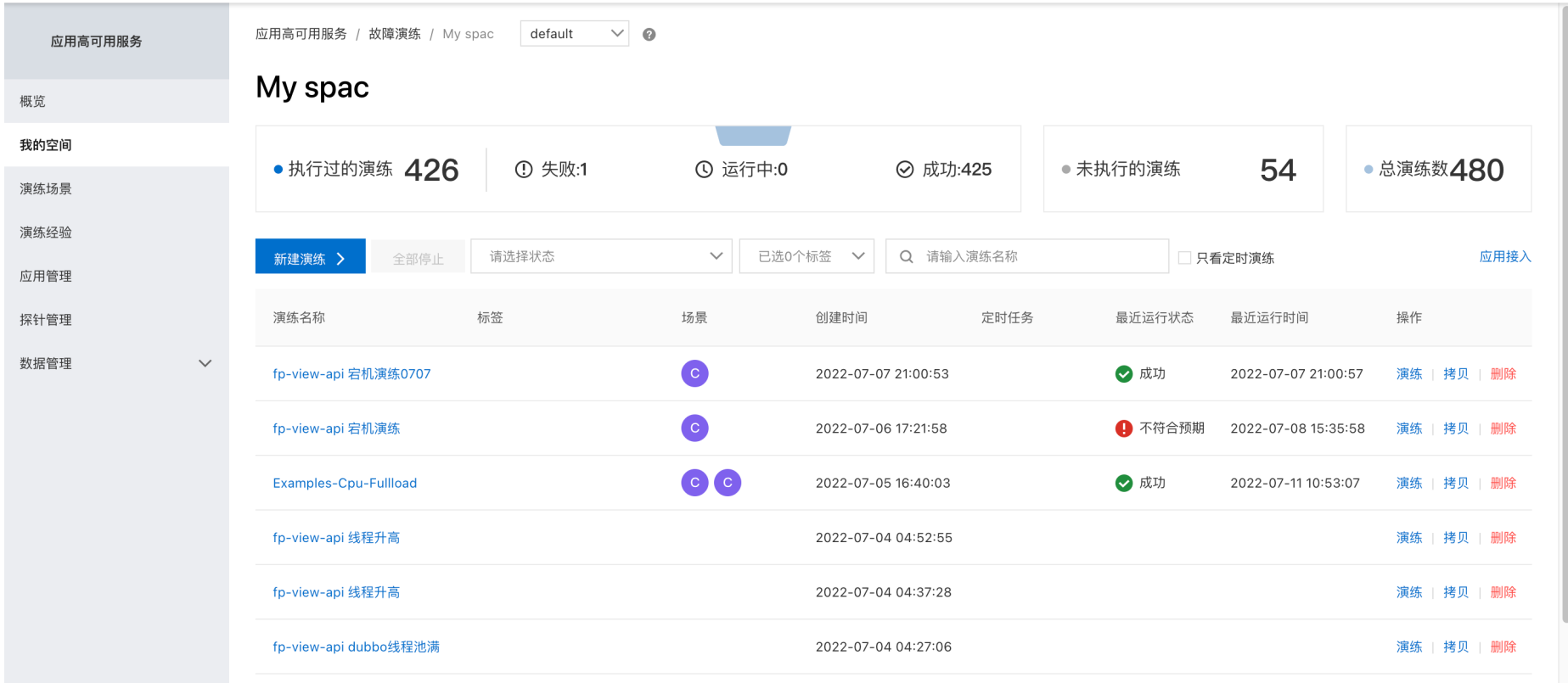
版本组合搭配

Chaosblade v1.6.1

Chaos Box v1.0.1

优点：

- 商业化控制台体验
- 演练库经验能力沉淀
- 租户化环境数据隔离能力



业务落地实战

实战案例：互联网前端业务线故障演练

故障演练组织核心阶段：

- 规划演练计划
- 制定演练目标清单
- 确认故障演练参与相关方角色
- 确认演练业务线配合时间、地点
- 组织演练执行
- 复盘演练结果

实战案例-制定演练目标清单

演练SOP模板地址： https://docs.qq.com/sheet/DS1JPblFibUNRWGV3?tab=BB08J2

演练模板Sheet功能介绍如下：

演练模板填写说明：这里面介绍故障演练具有哪些能力，演练发起人可以参考这些能力指定演练目标；也定义了演练清单里面里面每个字段的填写说明；

演练计划与任务事项：这个sheet里面把演练计划和任务事项进行公开，可以根据实际情况进行填写；

演练参与人员名单：故障演练是一个团队参与的活动，需要很多角色参与配合；这个里面把每个人的工作安排给明确行下；

演练清单：里面每一行介绍了本次演练一次动作编排，和演练预期行为需要补充完善；

演练复盘：演练结束后针对本次演练清单的结构进行复盘，针对问题需要给出跟进人和修复时间；

实战案例-规划演练计划


标题
投入
十一月 17日
十一月 18日
十一月 19日
十一月 22日
十一月 23日
十一月 24日
十一月 25日
十一月 26日
十一月 29日
十一月 30日
十二月 1日
十二月 2日
十二月 3日
1) 方案设计 2 周 3 天
1.1) 方案设计 4 天
演练发起人; 业务研发
1.2) 目标方案评审 3 天
业务研发; 业务负责人; 业务测试
1.3) 演练计划评审 3 天
业务研发; 业务负责人; 业务测试
1.4) 演练参与人员确认 3 天
业务研发; 业务负责人; 业务测试
2) 方案准备 4 周 4 天
2.1) 资源信息梳理 1 周 3 天
业务研发; 业务测试; 演练发起人; 演练负责人
2.2) 监控渠道梳理 1 周 3 天
业务研发; 业务测试; 演练发起人; 演练负责人
2.3) 故障场景编排 1 周 3 天
业务研发; 业务测试; 演练发起人; 演练负责人
3) 故障正式演练 2 周 4 天 6.75 小时
3.1) 相关人员准备事项确认 2 天 7.75 小时
3.2) 故障发起 2 天 7.75 小时
演练发起人
3.3) 故障业务测试 2 天 7.75 小时
业务测试
3.4) 故障演练结论评估 2 天 7.75 小时
业务负责人
3.5) 故障演练恢复 2 天 7.75 小时
演练发起人
4) 故障演练复盘 1 周
4.1) 故障演练结论复盘 1 周
业务研发; 业务测试; 演练发起人; 业务负责人; 演练负责

实战案例-组织演练执行

将相关演练场景在故障平台上面提前编排好，确认参与同事都已经上线可以支持；

发起演练故障植入，通知业务测试人员对业务进行回归测试得出业务实际表现；

业务测试人员对业务进行回归测试得出业务实际表现；

研发人员观察业务监控、日志监控或者微信渠道是否及时通知相关负责人同事；

执行恢复故障操作，检测业务是否恢复正常，是否有对应自愈能力生效；

记录故障演练关键信息方便进行有效复盘操作；

Q&A

融合共享 成于至善