



从理念到实战

商业银行混沌工程研究和实践白皮书

中国农业银行 华为技术有限公司 联合发布



编写单位及作者

指导单位

北京金融科技产业联盟 混沌工程实验室

编写单位

中国农业银行研发中心 华为2012服务实验室
华为云计算技术有限公司

编写委员会

专家顾问

蔡仕志 闫晓林 聂丽琴 陈屹力 贡佳炜 谢友珉 许国平
罗学平 刘跃光 邓锐 龚毅 贾磊 刘柳明 徐志华 魏坤
孙虎 胡玉海 郭婷 何亮 齐泊淞

主要执笔人

赵勇 帅慧敏 刘浩 彭尚峰 杨易寰 刘福铭 李军 王郁
黄容萍 王海清

编写组成员

李惠婷 徐进晓 邱春丽 刘蕊 张少博 张泽 陈飞云 王嘉良
胡莉莉 刘一童 张慧萍 王宪刚 涂锐 来丽晗 耿鹏 赵欣
何学辉 曹铨 曾剑鹿 王晓华 耿鑫 王振刚 常佳 聂刚
陈鹏飞 张子阳 王耀园 傅晟 刘平 郑云洲 胡志远 梁晓豪
胡晓磊 林丽鑫 曾磊 刘丽彦



前言

二十大报告指出要加快建设数字中国，并对加快发展数字经济做出了战略部署。具体到金融领域，《国务院关于金融工作情况的报告》明确要统筹发展与安全，牢牢守住不发生系统性金融风险的底线，坚定不移走中国特色金融发展之路，实现金融高质量发展。对于商业银行，IT系统作为数字业务的承载，是支撑数字化转型的重要力量。随着金融创新越来越多、产品交付效率越来越高，IT系统规模愈发庞大、链路结构依赖日益复杂，这种结构性的内生复杂与不可预见的用户行为交织在一起，对系统的稳定性提出了更高要求，稳定性建设已成为云原生时代银行IT部门的关注重点。

在此背景下，混沌工程（Chaos Engineering）作为一种以主动注入故障方式识别和修复未知隐患的手段，已成为稳定性提升的有力抓手。本白皮书针对混沌工程在商业银行落地过程中如何构建体系化实践方法论、如何评价实践效果、如何建设企业文化三方面挑战，给出了参考方案。

《从理念到实战——商业银行混沌工程研究和实践白皮书》在北京金融科技产业联盟、混沌工程实验室指导下，由中国农业银行研发中心、华为2012服务实验室、华为云计算技术有限公司联合编制而成，是践行数字化转型、高质量推进业务发展过程中对于混沌工程的研究和实践集成，对同业具有较强的参考和借鉴价值。

目录

第一章	商业银行混沌工程应用背景	04
1.1	IT系统走向分布式时代	04
1.2	混沌工程提供了分布式系统稳定性分析和检验手段	05
1.3	混沌工程在商业银行快速发展	06
第二章	混沌工程落地挑战	07
2.1	如何构建混沌工程实践的体系化方法论	07
2.1.1	打法思路探索	07
2.1.2	支撑平台建设	09
2.1.3	管理机制适配	09
2.2	如何评价混沌工程实践效果	10
2.3	如何建设混沌工程团队文化	10
第三章	混沌工程落地体系研究	11
3.1	混沌工程基本原理研究	11
3.1.1	从风险到事故的“海因里希法则”	11
3.1.2	稳定性风险控制的“奶酪原理”	12
3.2	混沌工程基本原则研究	13
3.3	混沌工程落地体系研究	13
第四章	混沌工程落地体系要点	16
4.1	实践方法体系	16
4.1.1	打法思路	16
4.1.2	技术平台	26
4.1.3	管理机制	27
4.2	评价体系	29
4.3	企业团队文化	30
4.3.1	技术认同感	30
4.3.2	教练人才培养	31
第五章	实践案例	32
第六章	未来展望	37
6.1	技术层面展望	37
6.1.1	自动化故障场景分析	37
6.1.2	超大规模故障注入和回退	38
6.1.3	请求级别的无侵入故障注入	38
6.2	应用层面展望	38
6.2.1	混沌工程信心值模型	38
6.2.2	领域故障场景库建设	39
第七章	总结	39
附录	术语解释	40

第一章

商业银行混沌工程应用背景



1.1 IT系统走向分布式时代

党的二十大报告指出要加快建设数字中国，并对加快发展数字经济做出了战略部署。具体到金融领域，《国务院关于金融工作情况的报告》明确要求统筹发展与安全，牢牢守住不发生系统性金融风险的底线，坚定不移走中国特色金融发展之路，实现金融高质量发展。IT系统作为数字业务的承载，是支撑商业银行数字化转型的重要力量，随着金融创新越来越多，支付效率越来越高，传统单体IT架构已暴露出很多不适应的地方，为此商业银行广泛应用云原生，分布架构等新技术来支撑快速发展。

新技术在具备“高性能，高弹性，低成本”的优势同时，也伴随IT系统规模愈发庞大，暴露出风险点位增多、依赖关系复杂等特征，这种结构性的内生复杂与不可预见的用户行为交织在一起，对系统的稳定性提出了更高要求，稳定性建设已成为云原生时代银行IT部门的关注重点。

相比小型机时代的集中式单体应用，分布式系统由大量PC服务器组成，强调通过横向扩展来提升整体性能，但当前PC服务器的单点可靠性与小型机还存在差距。这意味着为了达到同等甚至更优的稳定性水平，技术路径上须要从依赖硬件可靠性，转为主动在软件层面构建稳定性。

1.2 混沌工程提供了分布式系统稳定性分析和检验手段

稳定性属于IT系统非功能性需求，在实际工作中存在不易验证的特点。由于单体应用故障率相对较低，所以在单体时代，需要在软件层面关注的稳定性问题并不复杂。而在分布式时代，系统规模呈指数级上升、故障发生的频率变高，这种变化驱使软件层面必须提升系统稳定性的建设和运营能力。关于系统稳定性，业界专家积累了很多正向建设经验和原则，但缺乏有效的反向验证手段。

混沌工程作为近些年兴起的一种新理念，其通过设计并执行一系列实验，帮助发现IT系统技术架构（需求、设计、架构、代码、运维等）与运营流程方面的隐藏风险和薄弱环节，从而全面提升系统稳定性¹。混沌工程起源于奈飞公司（Netflix）的一次停机故障，此次故障导致了DVD租赁业务中断三天，多个国家的大量用户受到影响。随后，奈飞的工程师逐步探索创建了一种名为Chaos Monkey的工具，通过随机终止在生产环境中运行的实例可以快速了解正在构建的服务是否健壮，是否有足够的弹性可以容忍计划外的故障。

混沌工程理念一经提出便在技术领域引起了广泛的注意和讨论，相比传统测试手段，混沌工程的价值不仅仅是发现系统的风险或“暗病”，而是综合验证系统和业务健壮性、优化应急预案、锤炼技术运营和应急响应能力、提升对客户响应和舆情攻关能力，最终提升业务、开发、测试、运维全条线人员对业务稳定性的信心。根据《中国混沌工程调查报告》²统计，“混沌工程使用频率”与“组织内产品可用性”是正相关的，体现了混沌工程在分布式系统稳定性提升的价值。

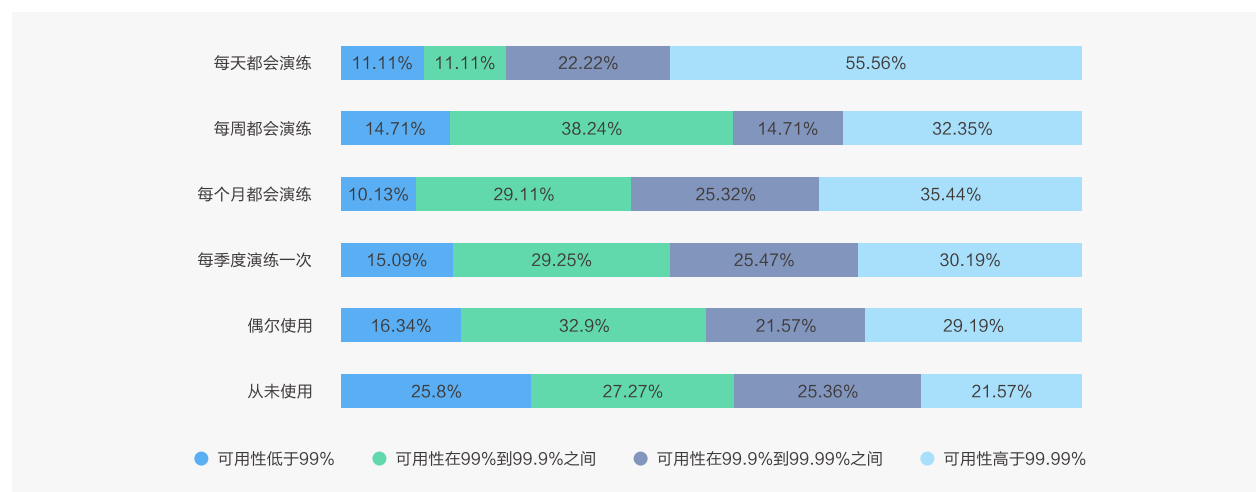


图 1 产品可用性在不同混沌工程使用频率上的分布

稳定性测试非混沌工程首创，但传统的稳定性测试侧重点在于如何防范已知范围内系统缺陷的引入，对于需要特定外界扰动才能触发的故障缺乏识别和修复手段，只能在系统故障发生时对故障进行被动的响应，导致故障应对的进度和成本不可控。同时，传统稳定性测试多需要在代码层面编写用例，效率低下且项目覆盖率无法得到保证。而混沌工程代码侵入性相对较低，很好地弥补了这个短板。

¹《中国混沌工程技术实践指南》中国信通院·混沌工程实验室

²《中国混沌工程调查报告》中国信通院·混沌工程实验室

1.3 混沌工程在商业银行快速发展

近年来随着对IT系统稳定性要求的不断提高，混沌工程在国内呈现快速发展趋势，其中商业银行对于混沌工程的探索和实践处于“头部”位置。混沌工程受到商业银行“青睐”有着必然因素：

一是对于资金安全有着极高要求。提升系统稳定性、防范金融风险是商业银行义不容辞的责任。

二是处于技术转型深水区。技术架构、开发模式、运维模式均在变化，混沌工程作为稳定性提升的重要手段，必然受到更多关注。

三是整体研发实力较强。有着良好的研究和实践基础，更容易落地混沌工程。

商业银行混沌工程发展的过程中，也针对业务稳定性提升积累了大量的特色场景和实践，以下简要列举了4类：

一是业务依赖。对于交易来说，响应时间有着比较高的要求，那么在服务链路日趋复杂的背景下，梳理清服务间的强弱依赖关系、路由关系对于业务健康发展是十分必要的。

二是业务连续性。对于极端情况下业务连续性保障，容灾建设是有效的应对措施，但是这并不是单纯的“堆设备”，需要在平时定期/不定期开展演练，检验从接入层、应用层到数据层的接管能力和时长是否符合预期，应急预案和切换步骤是否可执行、有效果。

三是营销活动。纪念币预约、信用卡营销等典型的营销活动是极具有银行特色的，不仅与单一的交易、业务条线相关，而是整个体系的“压力测试”。演练中可以通过混沌工程手段验证业务逻辑（例如错峰预约策略的合理性）和技术能力（扩缩容、熔断限流能力等），以及人员应急处置能力。

四是资金一致性。围绕着资金安全，对于交易幂等验证、不完整事务发生后的处理策略、差错处理机制等的验证。

特别地，商业银行结合当前国际形势和趋势导向，从实际金融需求出发，以应用场景为牵引推动关键技术持续迭代完善，在多个领域针对新技术、新产品进行攻关，以应用促提升，以应用促创新。在此过程中，混沌工程通过模拟各种故障场景，为新技术、新产品的攻关提供了更为丰富的运行观测数据，加速了经验的积累。



第二章

混沌工程落地挑战



混沌工程的理念非常吸引人，并且经过前期探索实践取得了阶段性成果，但在实践过程中，仍然迫切需要系统性的方法体系发挥指导和引领作用，以便更好、更全面地推广并真正释放技术价值。结合调研分析及自身实践，方法体系需要解决三类挑战，一是如何构建混沌工程实践的体系化方法论，二是如何评价混沌工程的实践效果，三是如何建设团队文化。

2.1 如何构建混沌工程实践的体系化方法论

与较为成熟的开发、测试及运维技术体系相比，混沌工程作为一门“年轻”的技术，在实践环节中尚未形成通用的方法体系或最优实践，主要体现在以下3个方面。

2.1.1 打法思路探索

1. 如何分析故障场景？

混沌工程推广初期，不同工程师对怎样进行混沌工程实践，理解和行动上的差异很大。有的工程师不清楚怎样进行故障实验，是在一个节点注入还是多个节点注入，是注入CPU满还是注入网络丢包，是否需要组合注入，面对灵活的注入工具反而感到迷茫；有的工程师选择“随机注入”，有甚者为制造故障“不择手段”，这种做法确实会导致业务不可用，但现实中显然也很难出现这种故障，脱离了实际，使得混沌工程实践偏离目标，变成为了故障而制造故障。

以上做法与混沌工程内涵是相违背的，一般而言，混沌工程的目标是验证某个故障场景下系统的稳定性。但是到底如何分析并找出有针对性的故障场景，当前并没有很好的参考实践。

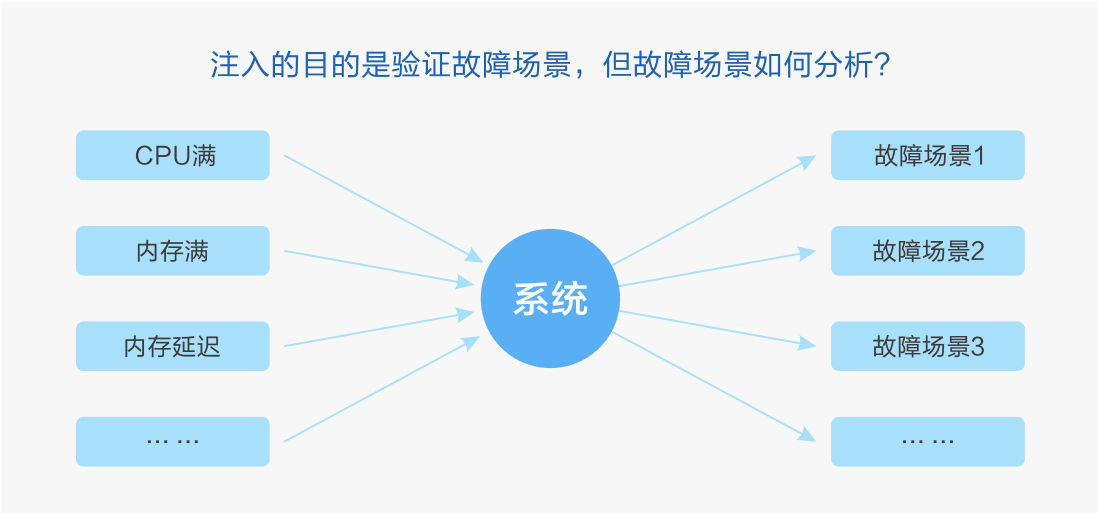


图 2 混沌工程场景分析示例

2. 如何进行故障注入降维？

经过多年发展，当前故障注入工具所能模拟和注入的故障场景非常丰富，少则四五十种、多的达到几百种。工程师在开展混沌工程时常会有疑问：这么多故障场景，是不是每个都要测一遍？IT系统模块、组件、服务、功能的复杂性叠加故障模拟和注入手段的丰富性，会导致“场景爆炸”现象发生。

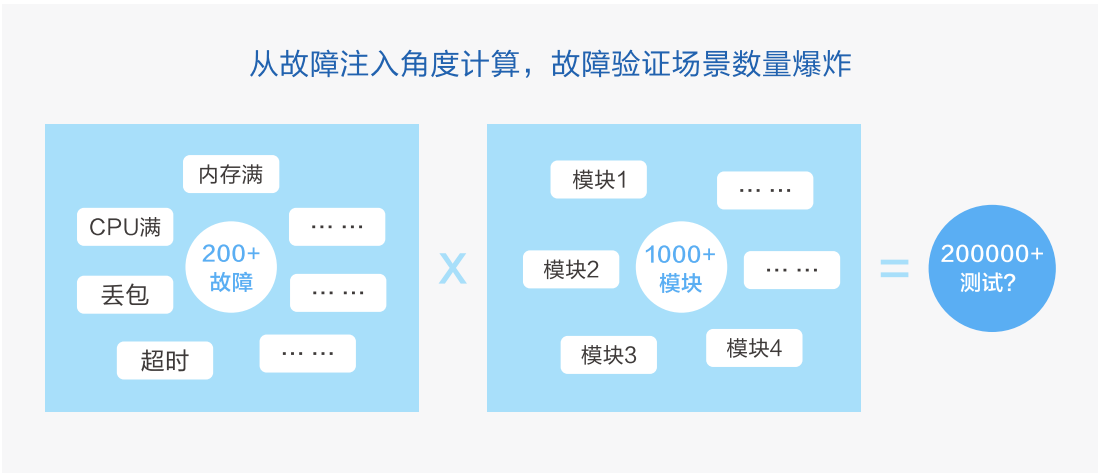


图 3 故障注入“场景爆炸”示例

3. 混沌工程的实验步骤包括哪些环节？

完整的混沌实验应该包括哪些环节，这也是混沌工程在组织中推广面临的挑战之一。标准化的实验步骤可以统一工艺流程、降低推广成本。同时混沌工程的实施具有一定破坏性，标准化的步骤可以控制和规避风险。

4. 以何种业务形式开展混沌工程？

经过多年的发展，工程师们在监控、故障处理等领域已经形成了标准化思路或最佳实践，例如监控的最佳实践是SLI（Service Level Indicators，能体现业务质量的指标）、SLO（Service Level Object，业务期望达到的质量目标）、ErrorBudget（一般称为错误预算，一年允许的不可用时长）。运维工程师实施监控的时候，可按照先分析业务的SLI指标、再商讨SLO、最后确定ErrorBudget的思路开展，已形成标准化的范式。相比传统运维手段，混沌工程属于“新生事物”，以何种形式开展，属于实践层面需要探索的新目标。

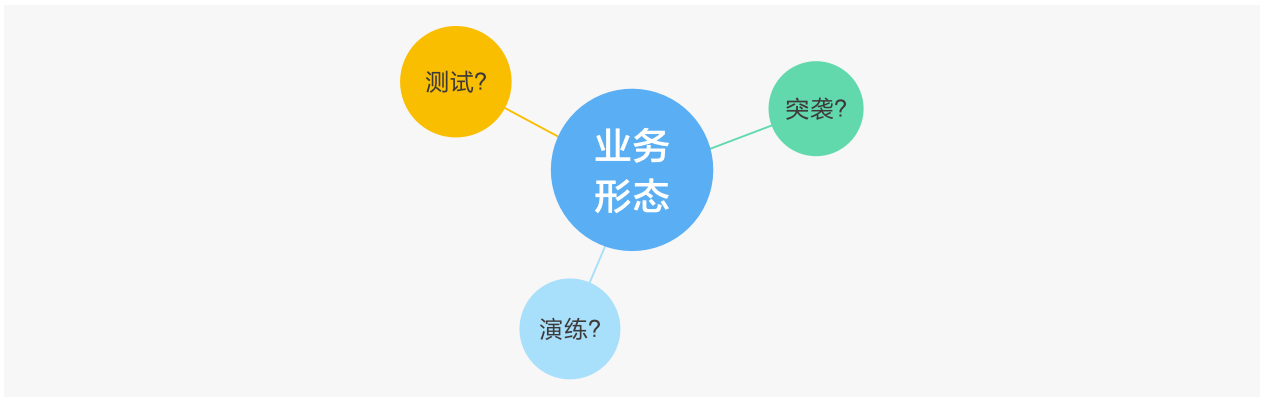


图4 混沌工程开展业务形式举例

2.1.2 支撑平台建设

混沌工程的有效开展离不开技术平台的支撑，但企业级混沌工程实验平台并非单纯是故障注入的工具，而是监控、实验管理、结果分析等能力的集合。对于商业银行而言，一般都具备较为完善的监控、分析能力及相关管理机制。实验平台的建设需要考虑是新建上述能力还是复用已有的能力，如果复用已有能力的话，如何与已有系统打通？

2.1.3 管理机制适配

1. 混沌工程由哪个组织承接、有哪些角色和职责、与关联团队怎样配合？

混沌工程作为新的业务，应该成立一个新团队还是分解到原来的组织承接，这些人应该具备什么样的技能，和组织内原来的开发、测试、运维团队之间怎么配合？

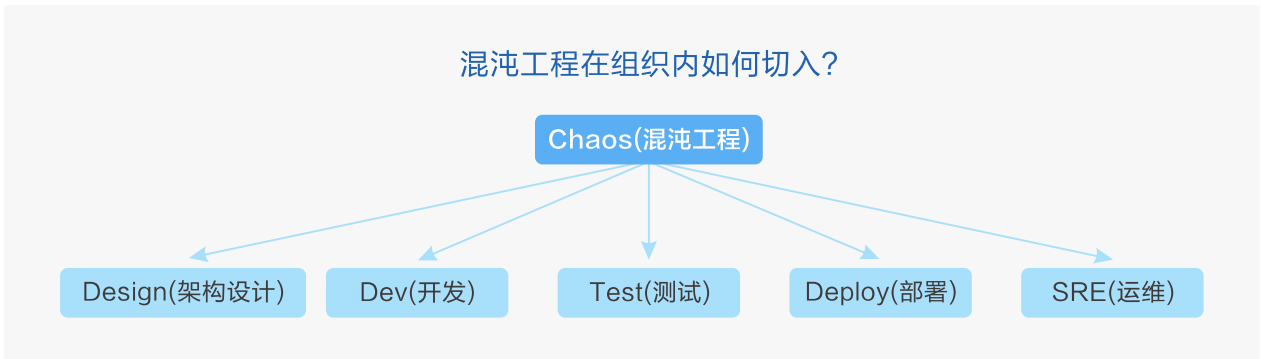


图5 混沌工程如何切入原有组织

2. 混沌工程需要建立什么样的管理机制？

混沌工程类似健身，是一种“反人性”行为。人生病了一定会马上吃药，这是顺应人性的，但健身则不同，即使人人都知道健身有利于保持身体健康，但能始终坚持的是少数。如果我们想长期实施混沌工程，应该构建什么样的管理机制来解决“反人性”的问题，规模化、常态化推进？



2.2 如何评价混沌工程实践效果

混沌工程的落地，不仅需要在实践方法层面给予规范和指引，还需要探索构建可量化的评价体系来牵引混沌工程的健康运营。具体来说，无论是对于个体还是企业，一方面需要了解当前水平、进而提升技术及应用熟练度，另一方面需要通过看到实践价值来获得成就感，形成驱动混沌工程发展的“正向循环”。但“可量化”的内容有哪些，可以从哪些维度切入，评价体系是否就是考核体系？这些问题也需在实践过程中不断探索。

2.3 如何建设混沌工程团队文化

混沌工程落地，管理机制是通过施加外力的方式解决实施动力的问题。除此之外，还应该解决内生动力的问题，提升参与人员的积极性。但混沌工程是一种主动“暴露”风险的技术，需要考虑实践过程中如何引导大家敢于暴露风险。例如，混沌工程暴露风险可能会引起他人不满，导致混沌工程实施会有顾虑，如何解决这种顾虑？混沌工程是否一定要在生产环境实施？类生产环境实施如何达到相同效果？尽管实验前会考虑回退等各种保护措施，但故障注入难免会遇到意外甚至风险（在生产环境开展如果处理不当则会导致生产事故），这种情况下是追责还是鼓励？

此外，混沌工程的推广离不开人才培养。如何在企业内部培养满足混沌工程实践需求的人员，也是混沌工程落地过程中的关键问题。

第三章

混沌工程落地体系研究



第二章介绍了商业银行混沌工程落地过程中面临的三类挑战，本章将从技术风险视角出发，厘清基本原理、正本溯源，并结合基本原理介绍在落地过程中如何体系化解决上述三类挑战。

3.1 混沌工程基本原理研究

3.1.1 从风险到事故的“海因里希法则”

安全工程领域有个“海因里希法则”，即对于企业而言1项重大事故背后必有29项轻度事故和300项潜在的风险。这条法则同样适用于IT系统稳定性领域：想要减少事故，必须从消灭风险入手。

结合稳定性领域进一步分析可以得出这样的结论：风险的潜在性会带来2个问题——“侥幸心理”和“能力退化”。“侥幸心理”指IT系统从单个组件角度看故障发生的概率并不高，而业务长时间不发生故障会使技术人员产生系统很可靠的“侥幸心理”；“能力退化”是指由于故障少，技术人员能力得不到锻炼。

以“侥幸心理”为例。假设1个业务仅部署在1台服务器上，没有做冗余、存在单点风险。按照概率，这台服务器可能今天就出故障、也可能3年不出故障，侥幸心理就会认为服务器3年不出问题，而且业务也下线了。这种做法可能会侥幸成功，但一旦失败就会造成业务不可用、引起损失。

再说“能力退化”。由于故障频率较低，运维人员没有机会见到故障，当遇到真实故障发生时难免紧张。再加上场景陌生，人的心理压力会陡然增大，导致定位和恢复操作变慢，甚至手忙脚乱误操作。

这就是“海因里希法则”在稳定性领域的应用——由于潜在风险转换成故障的概率低加上大家心存侥幸，导致潜在风险难以得到重视。而当风险转化成真实故障时，又由于能力退化导致系统恢复缓慢，进而导致更大损失。因此，为了提升稳定性需在潜在风险挖掘和人员训练2个方面进行努力，而混沌工程在这2个方面提供了有效的手段。

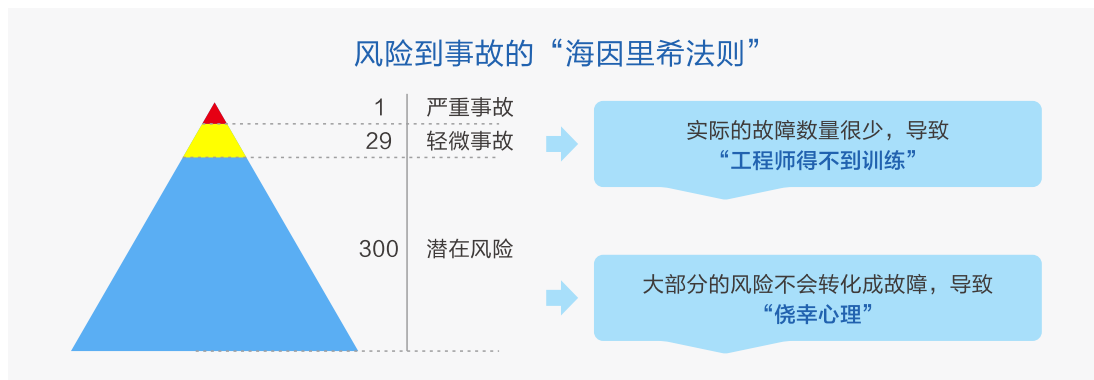


图6 “海因里希法则”示意图

3.1.2 稳定性风险控制的“奶酪原理”

越来越多的企业认识到，IT系统的稳定性提升是伴随全生命周期的，即提升稳定性需要从开发到交付到运维，甚至在产品设计端的每个环节都需要考虑稳定性设计。但这些稳定性设计的要求很容易停留在口头或者纸面检查（checklist）阶段。由于缺乏行之有效的验证手段，风险往往会层层渗透到生产系统，变成由运维人员兜底，这也是运维人员形容自己是“救火队员”的原因，这样的状况持续下去显然会陷入恶性循环。

而混沌工程技术可以作为稳定性风险拦截的手段，通过在不同阶段实施混沌工程，不断挖掘稳定性风险，并通过故障注入反向验证，从而实现“风险左移”，降低生产系统风险。

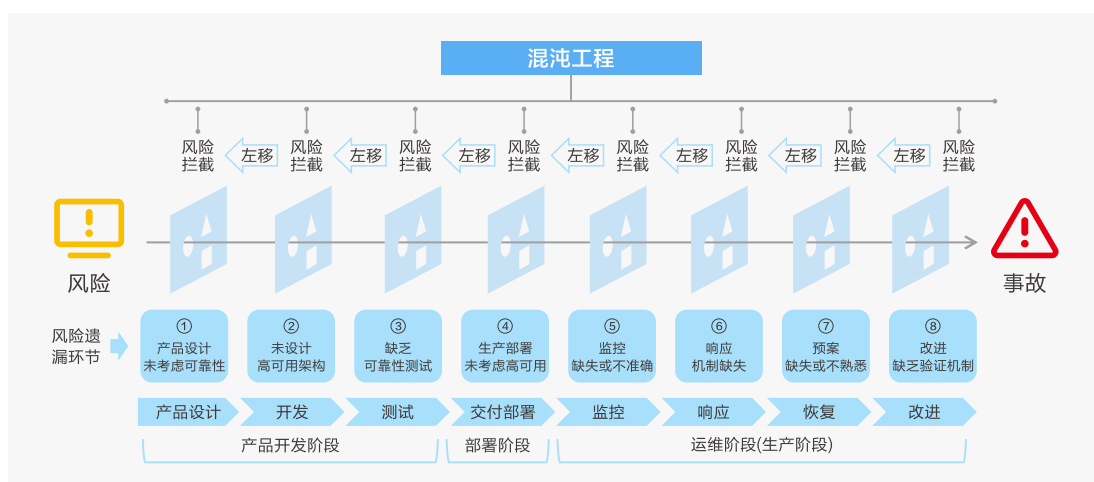


图7 稳定性风险控制的“奶酪原理”示意图

3.2 混沌工程基本原则研究

作为混沌工程的提出者，奈飞公司（Netflix）提出过混沌工程的指导思想，也即俗称的“5大原则”：生产环境运行实验、真实事件、建立稳态假说、自动化运营实验、最小化爆炸半径。该原则通用性很强，但针对性不足。在遵从“5大原则”的基础上，商业银行由于业务连续性高、监管要求高、资金敏感度高、声誉风险大等特点，在落地实践中可以综合考虑研发管理流程、验证成本和风险控制等因素，将通用原则转化为“8大原则”。

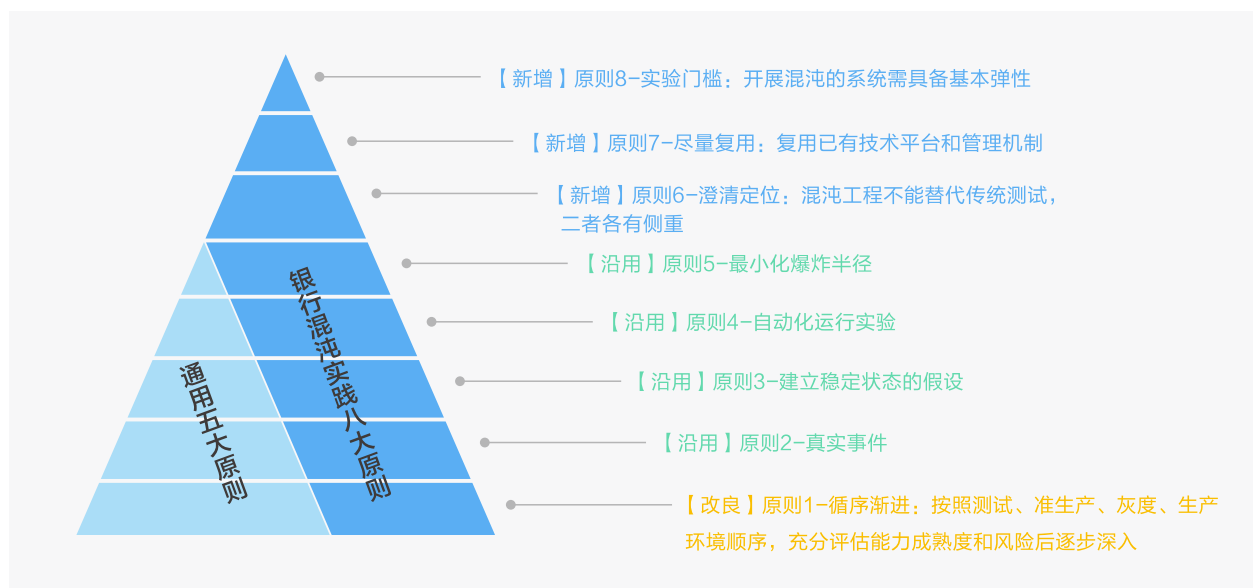


图8 混沌工程参考原则

原则1指出，商业银行由于业务特点，直接在生产环境运行实验并不现实。建议采用测试、准生产、灰度、生产环境递进的方式，在充分评估成熟度后逐步深入。

此外，原则6明确了混沌工程与传统测试的互补关系，不应因为有了混沌工程就放弃了传统测试；为降低混沌工程的推广难度，原则7建议“尽量复用已有技术和管理机制”；尽管混沌工程是提升系统稳定性的有效手段，但并不是“银弹”，对明知有稳定性问题的IT系统，原则8建议首要工作应是提升系统自身水平，而不是先开展故障注入。

3.3 混沌工程落地体系研究

明确了基本原理和实施原则后，更关键的是如何将混沌工程付诸实践。混沌工程并不是简单的工具应用，而是包含了工具、人、流程的综合体系。针对第二章提出的三类挑战，落地体系可拆解为实践方法体系、评价体系和团队文化三个部分。以下结合图9，由“中心到两边”对各部分进行概述。

实践方法体系包括打法思路、技术平台、管理机制3个方面，打法思路主要要解决执行动作标准化的问题，使得组织上下对混沌工程的理解和行为是一致的；技术平台通过将业务平台化，解决技术门槛和效率的问题，使得工程师通过简单的UI界面即可完成复杂的混沌实验；管理机制要明确组织职责，使得混沌工程转变为组织工作的一部分。

评价体系主要解决混沌工程价值如何更好体现的问题，通过有效可量化的指标牵引混沌工程走向正轨。

团队文化为混沌工程在组织内部能更顺利落地营造良好的氛围。



图9 混沌工程落地体系框架

与此同时，混沌工程实践过程中建议采用“统筹规划、分步实施”的路径：以提升业务韧性为核心目标，按“小步快跑”模式分阶段、分步骤完成技术平台建设、机制流程固化、打法思路沉淀、评价体系完善、组织文化确立等工作，逐步深入混沌工程推广落地。

其中，一阶段首要目标是聚焦能力建设，从无到有初步搭建技术支撑平台，使部分系统先具备混沌工程实验能力，解决“从0到1”的问题。一方面围绕能力建设，通过在刚需系统上先行试点逐步建成统一的技术平台，形成标准实验流程与评价体系雏形。另一方面，快速形成混沌工程落地方案，有效赋能系统稳定性验证。

二阶段首要目标为聚焦能力沉淀，在前期试点基础上快速将混沌工程能力规模化拓展到其他系统上，解决“从1到N”的问题。即通过横向的系统规模化推广、纵向的技术深层次应用，拓展链路的实验场景，逐步沉淀技术指引与规范。在深度实验的过程中，可以积累本企业的特色场景，形成特色实验场景库，充分释放技术资产价值。

三阶段首要目标聚焦在能力进阶，在前两阶段建设基本成熟的基础上，继续优化深耕，持续在流程管理等维度构建规范的应用体系，解决“从N到精”的问题。通过前两个阶段的能力建设与规模化推广应用，混沌工程能力已基本完备，进入常态化验证和演练阶段。通过将混沌工程能力纳入项目流程管控，集成到DevOps（研发运维一体化）流水线质量门禁规范化管理，持续锤炼实验能力、优化价值效能。同时，人才队伍建设也是关键一环，需做好人才队伍培养，持续发挥技术价值。



图 10 混沌工程落地参考路径



第四章

混沌工程落地体系要点



4.1 实践方法体系

混沌工程实践方法体系的目标是尽量将混沌工程实施动作标准化、角色职责规范化、业务运作平台化，这也是混沌工程能在商业银行推广的必要条件。

4.1.1 打法思路

打法思路是一系列方法和规范的集合，通过这些方法和规范将执行动作标准化，主要包括故障场景分析、故障注入设计、流程规范、开展形式4个部分。

1. 故障场景分析

如第二章所述，当前混沌工程落地的主要挑战不是缺乏注入工具而是缺乏场景，挑战产生的原因是每个工程师对混沌工程的目标理解不同，即操作层面混沌工程目标、需求是什么、从哪里来。本白皮书站在运维视角，在总结FMEA（故障模式和影响面分析）和头脑风暴两类方法基础上，提出了FT-FMEA方法（容错视角的故障场景分析方法），具有目标清晰、操作性强的优点。

(1) 传统分析方法

FMEA（故障模式和影响面分析）起源于美国国家航空航天局（NASA），主要从业务的功能点出发，列出可能的失效模式、效果和原因、相应的控制手段，配合故障的严重等级、发生概率和可检测性等因素，最后对该模式得出RPN乘积分数，通过该分数可以判断该故障模式的风险等级。

1System

2Subsystem

3Component

4Design Lead

5Core Team

6

7

8

9

10

11

12

13

Cloud Service Availability

Application Layer

Code created by the application dev team

Potential Failure Modes and Effects

Key Date N/A

FMEA Number 1

Prepared By Adrian Cockcroft

FMEA Date 12/5/2018

Revision Date 4/24/2023

Page 1 of 1

Item / Function	Potential Failure Mode(s)	Potential Effect(s) of Failure	Sew	Potential Cause(s)/ Mechanism(s) of Failure	Prob	Current Design Controls	Det	RPN	Recommended Action(s)	Responsibility & Target Completion Date	Action Results						
											Actions Taken	ew	Sew	Onew	DNew	R New	
Authentication	Client can't authenticate	Can't connect application	5	Certificate timeout, version mismatch, account not setup, credential changed	3	Log and alert on authentication failures	3	45									
	Slow or unreliable authentication	Slow start for application	4	Auth service overloaded, high error and retry rate	3	Log and alert on high authentication latency and errors	4	48									
								0									
Client Request to API Endpoint	Service unknown, address un-resolvable	Delay while discovery or DNS times out, slow fallback response	5	DNS configuration error, denial of service attack, or provider failure	1	Customer eventually complains via call center	10	50	Dual redundant DNS, fallback to local cache, hardcoded IP addresses. Endpoint monitoring and alerts								
	Service unreachable, request undeliverable	Fast fail, no response	4	Network route down or no service instances running	1	Autoscaler maintains a number of healthy instances	1	4	Endpoint monitoring and alerts								

图 11 FMEA方法示例³⁾

FMEA给出了面向风险的思考方式，但FMEA中故障发生概率、严重程度、可检测等级的分类等级达到10个，实际实施时难以匹配。另外在实施过程中，工程师对同一个业务的分析差异也很大，容易导致故障模式发散。

头脑风暴法是以小组为单位，分析历史事件或业务本身来挖掘可能的故障因素，例如历史上因为CPU高导致的故障、系统延迟导致的故障等。分析完成之后，对这些故障进行优先级排序，然后围绕这些故障进行实验。

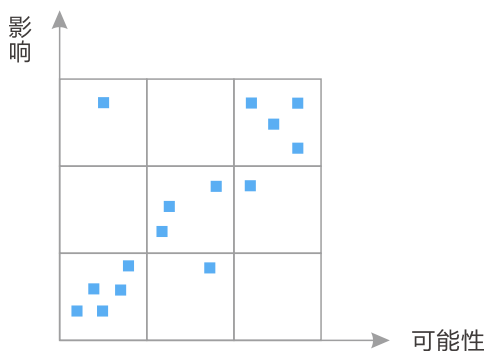


图 12 头脑风暴分析法示例

³ 图片来源: https://github.com/adrianco/slides/blob/master/FMEA_Application.xlsx

头脑风暴可以集众所长、尽量全面分析系统可能出现的故障，但也高度依赖团队经验，存在场景遗漏的可能。此外从故障原因的视角分析也很难控制场景数量、会造成“场景爆炸”，再加上软件工程师不一定理解硬件，硬件工程师也不一定理解软件，也会造成重要场景遗漏的情况。

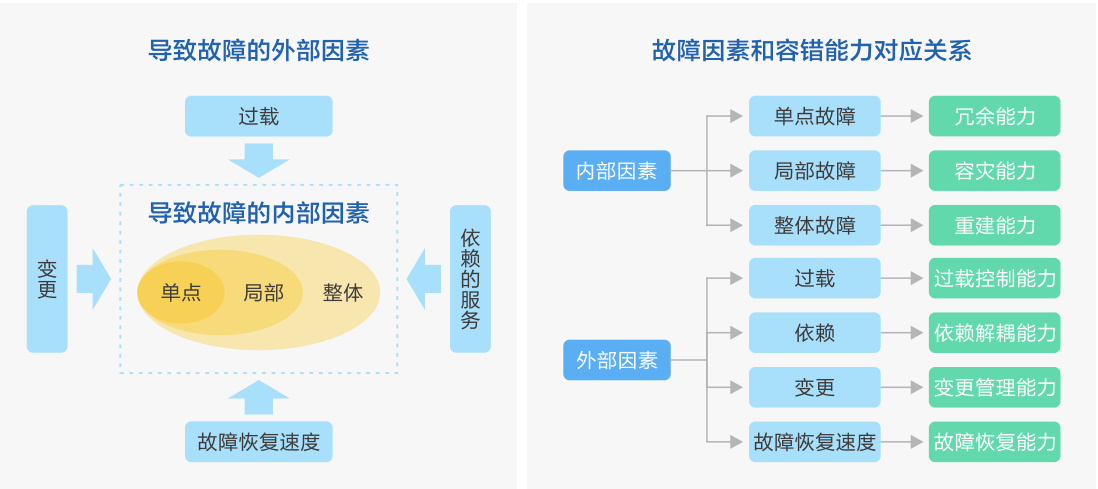
(2) 基于容错视角的故障场景分析方法（FT-FMEA）

传统方法存在分析框架操作性差，容易导致故障场景设计发散或者疏漏问题。此处提出了FT-FMEA方法（容错视角的故障场景分析方法），具有目标清晰、操作性强的优点。FT-FMEA从业务容错视角出发，按照应急手段构建了包含7类的容错视角分析框架，对故障场景进行有效降维；进一步，借鉴SRE思路，将故障与容错、预案、故障检测、恢复等能力建设进行映射，解决了混沌工程实验目的及需求问题。以下展开进行介绍。

① 7维容错视角分析框架

为了让工程师们的故障场景分析范围可控、维度可控，本白皮书提出一种实践层面更友好的分析方法——“容错视角的故障场景分析法”。提出这种方法的理论基础是虽然故障表现千差万别，但从业务容错视角来看，很多故障的处理手段是类似的。即从应急处置的视角反向来看，许多故障可以合并成同一类故障。例如服务器硬盘故障、服务器CPU满、服务器宕机，某个应用节点返回错误码5XX，这些故障原因不同，但都可以归类为单点故障，因为从容错视角来看，容错手段都是把故障节点隔离，从而保证业务整体不出问题；再比如系统发生局部范围故障的时候，原因可能是网络故障、可能是机房掉电、还可能是应用本身出现问题。但应急处置中为了快速恢复，往往也不会多花时间精准定位到具体的故障根因，而是大概定位到是哪个集群，把流量迅速切换到业务正常的集群。

所以，从容错的视角可以将多种故障场景合并同类项，大幅减少要分析的场景数量。基于上述思路，再结合分布式稳定性领域积累的经验，本白皮书总结出7种故障类型，对应最常见的7种容错能力。基于容错视角的故障场景分析法既可避免思维过于发散，也避免了遗漏。



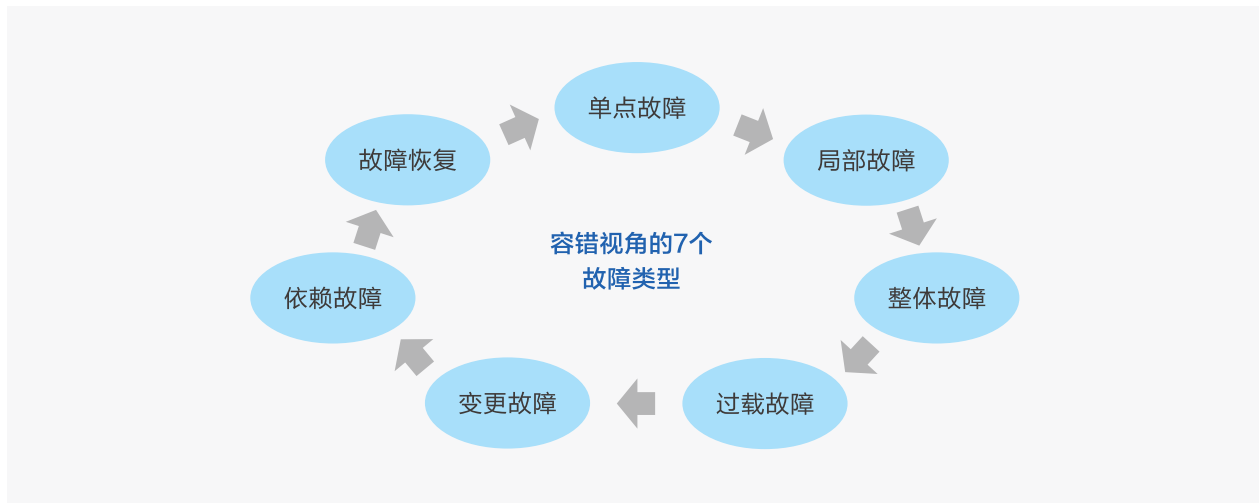


图 14 容错视角的7个故障类型

② 结合SRE视角的进一步优化

此外，借鉴SRE思路，可进一步将FMEA方法中增加故障类型、容错能力评估、预案、TTD、RTO、RPO等字段。综上，在实际操作中，首先从故障归类引导工程师从7个维度进行故障场景分析，而容错能力、预案、TTD、RTO、RPO则代表混沌测试的目的——检验系统容错能力、预案能力、故障检测能力、恢复能力等。改进的FMEA解决了混沌测试目的和需求的问题，可称为FT-FMEA（基于容错视角的故障场景分析方法）

表 1 基于容错视角的故障场景分析方法示例

(示例) LoginService服务故障场景分析 (FMEA)														分析人: XXX 时间: 2022-10-20	
产品/服务	微服务	组件	功能	故障场景名称	故障类型	发生概率	应急预案	故障检测方式	目标 TTD (秒)	目标 RTO (分钟)	目标 RPO (分钟)	容错能力评估	发生故障会造成什么业务影响	验证程度	风险等级
用户管理	loginService	loginService	用户登录/退出登录	loginService发生单点故障	单点故障	高	缺乏应急预案	1.节点连通性检测，连续3次连通失败认为节点故障 2: 节点返回码(4xx/5xx)检测，返回码错误率达到10%，认为节点故障	3	1	0		登录服务不可用，关联造成无法使用所有后台管理功能	P2	高
用户管理	loginService	LoginService	用户登录/退出登录	loginService发生局部故障	局部故障	中	切流	1.loginService QPS监控，loginService AZ1 QPS升降，AZ2QPS正常，则认为AZ1故障	180	5	0		登录服务短暂不可用，5分钟流量切换完成后，业务恢复	P3	中
用户管理	loginService	LoginService	用户登录/退出登录	loginService发生整体故障	整体故障	低	全部重启	XX	3	10	0		登录服务不可用，关联造成无法使用所有后台管理功能不可用	P4	低
用户管理	loginService	LoginService	用户登录/退出登录	loginService过载故障	过载故障	中	事先配置好限流阈值，如果过载是正常行为，扩容解决	缺乏监控	60	5	0		部分用户登录服务不可用，返回429，扩容后故障恢复	P3	低
用户管理	loginService	LoginService	用户登录/退出登录	loginService依赖的xxxService不可用	依赖故障	低	loginService自动启动服务降级，xxxService恢复后，loginService重试机制可以自动关闭降级	缺乏监控	30	5	0		loginService尝试5次失败后，自动进行降级，loginService本身不受影响	P2	低
用户管理	loginService	LoginService	用户登录/退出登录	loginService版本发布失败	变更故障		回滚						loginService不可用	P3	
用户管理	loginService	LoginService	用户登录/退出登录	loginService配置变更失败	变更故障		回滚/默认值			X	0		loginService不可用		
用户管理	loginService	MySQL主库	数据库读写服务	MySQL主库发生单点故障	单点故障	P2	mysql HA机制检测到主库故障后，自动切换到从库	通过MHA组件检测mysql主库连通性，5次检测失败认为主库故障	60	5	1		LoginService不可用，mysql切换到从库后恢复		高

2. 故障注入设计

故障注入设计是混沌工程落地面临的难点之一，核心问题是故障注入技术如何选择。前文提到，现有的故障模拟和注入技术少则几十种、多则数百种，如果每个系统都要做上百次故障注入，显然不可接受。

为了解决这个问题，可借鉴容错视角故障分类的思维，将故障注入也合并同类项。从运维实践来看，往往会忽略故障的根因，首先根据监控指标来做初步定位，进而快速恢复。例如单点故障，无论是CPU高导致业务故障、宕机导致业务故障、OS异常导致业务故障，还是其他单点故障，从监控表现来说要么是中断、要么是延迟、要是返回错误，因此在模拟故障时，完全不需要模拟那么多的故障类型，主要围绕网络中断、网络延迟、返回错误码就能覆盖绝大部分的故障场景。

3. 混沌实验流程规范

标准化的实验步骤可以统一工艺流程、降低推广成本。同时，混沌工程本身是一种“破坏性”实验，如何控制实验本身带来的风险也是实际操作中需要重点考虑的因素。结合前期实践，本白皮书将流程初步归纳为以下7个步骤：



图 15 混沌工程实验步骤

下面以系统登录服务（LoginService）为例，从工作内容、建议承担角色以及能力要求三个维度依次介绍7个步骤：

步骤1：故障场景分析

内容：根据应用在生产系统的部署架构（注意，尽管最终实验环境不局限于生产环境，但是分析一定要面向生产环境），参照故障场景分析指南得出本系统的故障场景。

角色：系统架构师或SRE团队。

能力：熟悉系统架构和生产部署情况。

表 2 故障场景分析示例

(示例) LoginService服务故障场景分析（FMEA）

分析人：XXX

产品/服务	微服务	组件	功能	故障场景名称	故障类型	发生概率	应急预案	障检测方	目标（TTD）（秒）	目标 RTO（分钟）	目标 RPO（分钟）	容错能力评估	故障造成什么业务影响	严重程度	风险等级
用户管理	loginService	loginService	用户登录/退出登录	loginService发生单点故障	单点故障	高	xx	xx	x	x	x	xx	xx	P3	高
用户管理	loginService	loginService	用户登录/退出登录	loginService发生单点故障	局部故障	中	xx	xx	x	x	x	xx	xx	P3	中
用户管理	loginService	loginService	用户登录/退出登录	loginService发生单点故障	整体故障	低	xx	xx	x	x	x	xx	xx	P3	低
用户管理	loginService	loginService	用户登录/退出登录	loginService发生单点故障	过载故障	中	xx	xx	x	x	x	xx	xx	P3	中
用户管理	loginService	loginService	用户登录/退出登录	loginService发生单点故障	依赖故障	低	xx	xx	x	x	x	xx	xx	P3	低

步骤2：风险等级评估和实施优先级排序

内容：评估该场景的风险等级，同时评估该场景的实施可行性以及优先级。

角色：目标系统的技术团队（包括开发、测试、SRE等）。

能力：熟悉本系统功能和技术实现细节。

表 3 优先级评估示例

(示例) LoginService服务故障场景分析（FMEA）																优先级 评估
																分析人：XXX
产品/服务	微服务	组件	功能	故障场景名称	故障类型	发生 概率	应急预案	障检测方	目标 (TTD) (秒)	目标 RTO (分钟)	目标 RPO (分钟)	容错能力 评估	故障造成 什么业务 影响	严重 程度	风险等级	优先级
用户管理	loginService	loginService	用户登录/退出登录	loginService发生单点故障	单点故障	高	xx	xx	x	x	x	xx	xx	P3	高	低
用户管理	loginService	loginService	用户登录/退出登录	loginService发生单点故障	局部故障	中	xx	xx	x	x	x	xx	xx	P3	中	低
用户管理	loginService	loginService	用户登录/退出登录	loginService发生单点故障	整体故障	低	xx	xx	x	x	x	xx	xx	P3	低	高
用户管理	loginService	loginService	用户登录/退出登录	loginService发生单点故障	过载故障	中	xx	xx	x	x	x	xx	xx	P3	中	高
用户管理	loginService	loginService	用户登录/退出登录	loginService发生单点故障	依赖故障	低	xx	xx	x	x	x	xx	xx	P3	低	中

步骤3：故障注入和爆炸半径设计

内容：设计本场景的故障注入方式，并评估实验的爆炸半径，实验异常时的及时终止。

角色：SRE或专门负责混沌工程的技术人员。

能力：熟悉故障注入工具，熟悉系统部署架构。

(示例) LoginService服务故障场景分析 (FMEA)																优先级评估		故障注入和爆炸半径设计			
分析人: XXX																					
产品/服务	微服务	组件	功能	故障场景名称	故障类型	发生概率	应急预案	检测方法	目标 (TTD) (秒)	目标 RTO (分钟)	目标 RPO (分钟)	容错能力评估	故障造成什么业务影响	严重程度	风险等级	优先级	故障注入设计	爆炸半径	故障紧急中止方法	回退时间评估 (分钟)	
用户管理	loginService	loginService	用户登录/退出登录	loginService发生单点故障	单点故障	高	xx	xx	x	x	x	xx	xx	P3	高	低	使用xx故障注入工具，注入网络中断故障（使用xx故障注入工具，注入网络超时故障，超时3s）（使用xx故障注入工具，注入返回5xx故障）	仅YY服务ZZ节点（节点级，机房级，服务器级，服务级）	使用xx故障注入工具直接取消故障	1	
用户管理	loginService	loginService	用户登录/退出登录	loginService发生单点故障	局部故障	中	xx	xx	x	x	x	xx	xx	P3	中	低	xx	xx	xx	1	
用户管理	loginService	loginService	用户登录/退出登录	loginService发生单点故障	整体故障	低	xx	xx	x	x	x	xx	xx	P3	低	高	xx	xx	xx	1	
用户管理	loginService	loginService	用户登录/退出登录	loginService发生单点故障	过载故障	中	xx	xx	x	x	x	xx	xx	P3	中	高	xx	xx	xx	1	
用户管理	loginService	loginService	用户登录/退出登录	loginService发生单点故障	依赖故障	低	xx	xx	x	x	x	xx	xx	P3	低	中	xx	xx	xx	1	

步骤4：稳态假设

内容：稳态假设是实验前对系统处于稳定状态的定义。一般可以从监控指标入手（监控指标本身就是为了反映监控系统是否正常），如有明确的SLO监控，则以直接使用SLO是否告警作为稳态定义；如果没有，则可以根据VALET⁴原则给系统制定合适的SLI监控指标，并制定异常监控策略。

角色：SRE或监控工程师。

能力：熟悉系统功能和监控设计。

表 5 稳态假设示例

(示例) LoginService服务故障场景分析 (FMEA)															优先级 评估		故障注入和爆炸半径设计				稳态假设	
分析人: XXX																						
产品/服务	微服务	组件	功能	故障场景名称	故障类型	发生 概率	应急预案	障检测方	目标 (TTD) (秒)	目标 RTO (分钟)	目标 RPO (分钟)	容错能力 评估	故障造成 什么业务 影响	严重 程度	风险等级	优先级	故障注入设计	爆炸半径	故障紧急中 止方法	回退时间评 估 (分钟)	稳态假设	
用户管理	loginService	loginService	用户登录/退出登录	loginService发生单点故障	单点故障	高	XX	XX	X	X	X	XX	XX	P3	高	低	使用xx故障注入工具，注入网络中断故障（使用xx故障注入工具，注入网络超时故障，超时3s）（使用xx故障注入工具，注入返回5xx故障）	仅YY服务ZZ节点（节点级，机柜级，机房级，服务级）	使用xx故障注入工具直接取消故障	1	SLO无警告	
用户管理	loginService	loginService	用户登录/退出登录	loginService发生单点故障	局部故障	中	XX	XX	X	X	X	XX	XX	P3	中	低	XX	XX	XX	1	xx指标下跌不超过5%	
用户管理	loginService	loginService	用户登录/退出登录	loginService发生单点故障	整体故障	低	XX	XX	X	X	X	XX	XX	P3	低	高	XX	XX	XX	1	XX	
用户管理	loginService	loginService	用户登录/退出登录	loginService发生单点故障	过载故障	中	XX	XX	X	X	X	XX	XX	P3	中	高	XX	XX	XX	1	XX	
用户管理	loginService	loginService	用户登录/退出登录	loginService发生单点故障	依赖故障	低	XX	XX	X	X	X	XX	XX	P3	低	中	XX	XX	XX	1	XX	

⁴ VALET指Volume（容量）、Availability（可用性）、Latency（延迟）、Error（错误）、Ticket（工单）5个维度的监控，是SRE常用监控指标。

步骤5：计划与周知

内容：明确本次混沌工程实验的执行计划、形式、参与人，并周知关联方。

角色：实验申请方（开发、测试、SRE）。

能力：善于项目管理和沟通。

表 6 混沌实验计划与周知示例

混沌实验计划与组织表	
类型	混沌测试/红蓝攻防/生产突袭
时间	2023.03.02 16:00-17:00
角色与参与人员	XXXX, XXXX

步骤6：执行实验

内容：执行故障注入，观测系统状态，异常情况的应急处理。

角色：SRE、开发、负责本次实验的工程师（红蓝攻防情况分蓝军和红军，蓝军注入故障，红军负责观测和应急处理）。

能力：具有系统变更权限。

步骤7：复盘和改进

内容：对本次混沌实验进行复盘，复盘可以参考真实事件的复盘流程，或参考如下简化的AAR（After Action Review，复盘分析）复盘形式。复盘完成后完成实验报告编写，记录本次实验发现的问题，跟踪改进。

角色：混沌工程运营工程师。

能力：项目管理能力，沟通能力。

表 7 复盘分析表示例

AAR (After Action Review)复盘表	
我们原先期望发生什么？	XXXX
实际发生了什么？	XXXX
造成差异的原因是什么？	XXXX
怎么改进？	XXXX

表 8 改进项管理表示例

改进项管理				
序号	改进项	责任人	计划完成时间	当前状态
1	对xx服务进行跨机架部署改造	xxx	2023.4.20	已完成
2	对xx服务补充监控项	xxx	2023.4.20	未完成
3				
4				

4. 混沌工程开展形式

混沌工程是一种主动行为，不像事件、报警，是不得不处理的。混沌工程的开展需要配套一些活动运营，通过专人主动组织才能有效开展。参考业界的GameDay（活动日）、DiRT测试（Disater Recovery Testing，灾难恢复测试）等混沌工程开展方式，商业银行可以通过可靠性测试、红蓝攻防和生产突袭三种业务形式开展。其中，可靠性测试频率最高，在系统内部1个工程师就可以自行开展；红蓝攻防则是以团队的形式开展；而生产突袭则是由混沌工程小组统一筹划。这3种形式对应不同的目的，效果上则是层层递进，例如红蓝攻防可以部分检验可靠性测试的质量，而生产突袭则全面检验混沌工程建设结果以及系统真正面临突发事件的表现。

(1) 可靠性测试

可靠性测试一般限定在系统范围内，通常情况下1个工程师就可以完成，实施人员可以是SRE或测试人员。一般SRE面向生产环境，测试人员面向测试环境。

可靠性测试的特点是主要检验业务的高可用能力，故障场景和发生时间都是透明的，没有真实故障带来的压力感。可靠性测试可以由各系统自行组织，也可以在组织级别统一以专项的形式开展，例如今年我们认为容灾问题最重要，那就专门成立容灾专项，拉通各组织统一行动。通过这种专项，可以地毯式的对某个场景的风险进行“扫荡”，大面积暴露并消减风险。

表 9 专项运作的可靠性测试示例

以专项形式运作		
容灾专项	2021.05-2021.12	项目经理:XXX
过载专项	2022.01-2022.12	项目经理:XXX
XX专项		

(2) 红蓝攻防

红蓝攻防通常是一种团队活动，与可靠性测试的区别是红军不清楚蓝军要构造什么样的故障，除了知道故障发生时间，其他都更接近于真实故障，有一定压力感。红蓝攻防活动相比可靠性测试更有仪式感，可以配有标志性的服饰、邀请领导或同组成员参加，国外一般称这种活动为“GameDay”（活动日）。红蓝攻防在互联网领域混沌工程应用成熟度较高的企业每年大约组织几十次，在商业银行目前实践较少。

小型的红蓝攻防可以邀请本领域领导参加，作为第三方观察者；大型的红蓝攻防（例如机房断网掉电演练）则可邀请高层领导参与，会后再发布战报宣传，进一步提升红蓝攻防的影响力。红蓝攻防可在更大范围组织内提升“面向失败的设计意识”，一次红蓝攻防可以影响到几十到上千名工程师。

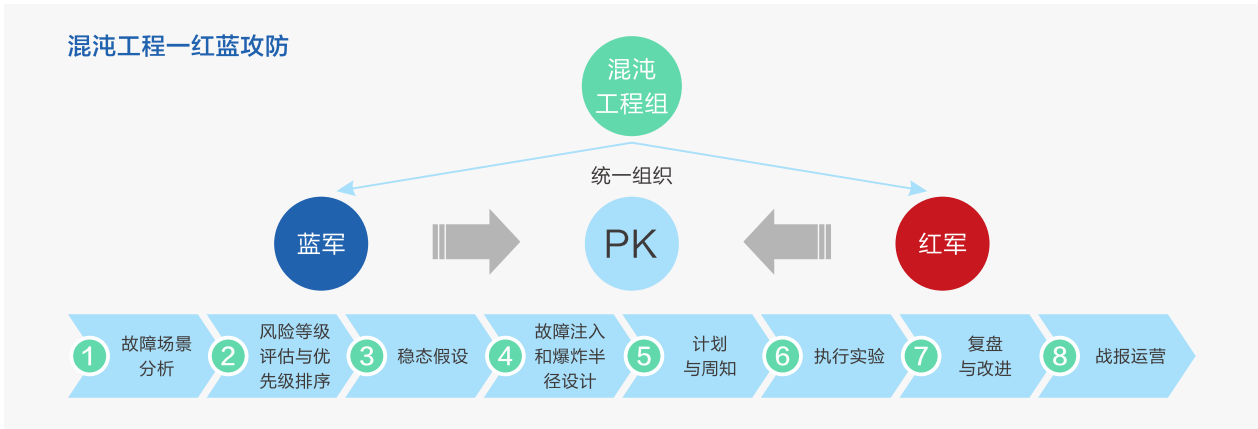


图 16 红蓝攻防流程示例

(3) 生产突袭

生产突袭一般由混沌工程小组策划、高层参与，甚至可以由高层承担实际角色、亲自注入故障。整个过程完全保密，与真实故障几乎没有区别，紧急情况可以中止。生产突袭可以完整地检验整个业务的稳定性能力、SRE和一二线故障处理能力。生产突袭会对提升稳定性运营能力产生特殊的作用，因为所有人都不知道这个故障是真是假，为了应对这种突袭，只能在平时就好这种准备，自然而然就提升了稳定性。生产突袭在互联网领域混沌工程应用成熟度较高的企业每年大约组织几次，在商业银行目前实践很少。

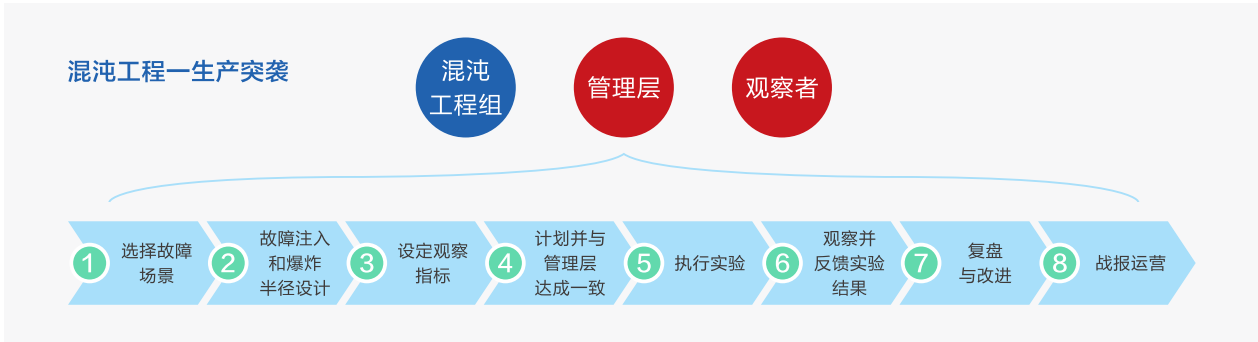


图 17 生产突袭示例

4.1.2 技术平台

为满足企业级混沌工程实践需求，商业银行多采用自研或采购与自研相结合的方式建设技术平台。典型技术平台应屏蔽底层基础设施差异、适配多种环境和技术栈、提供多维度多类型的一站式故障注入、实验管理、实验观测、故障场景建设等能力，保障“混沌不混”，为稳定性提升赋能，提升应用质效。

图18为一体化混沌工程实验平台的示例，整体平台架构分为服务端和客户端，其中服务端（Server）划分为四个中心：混沌实验中心、故障场景管理中心、实验观测中心和混沌运营中心。同时宜具备灵活的扩展能力，打通内部管理和运维工具，使混沌平台能够与内部开发测试、运维、业务连续性等体系相结合，复用内部数据和流程，减少重复建设。

客户端（Client）主要提供故障模拟和注入能力，在建设过程中可参考、集成或扩展国内外优秀混沌故障注入工具，与服务端功能结合，提供多维度多类型的故障场景，使得平台既能兼容已有资产，又能面向其他特色领域和未来架构。

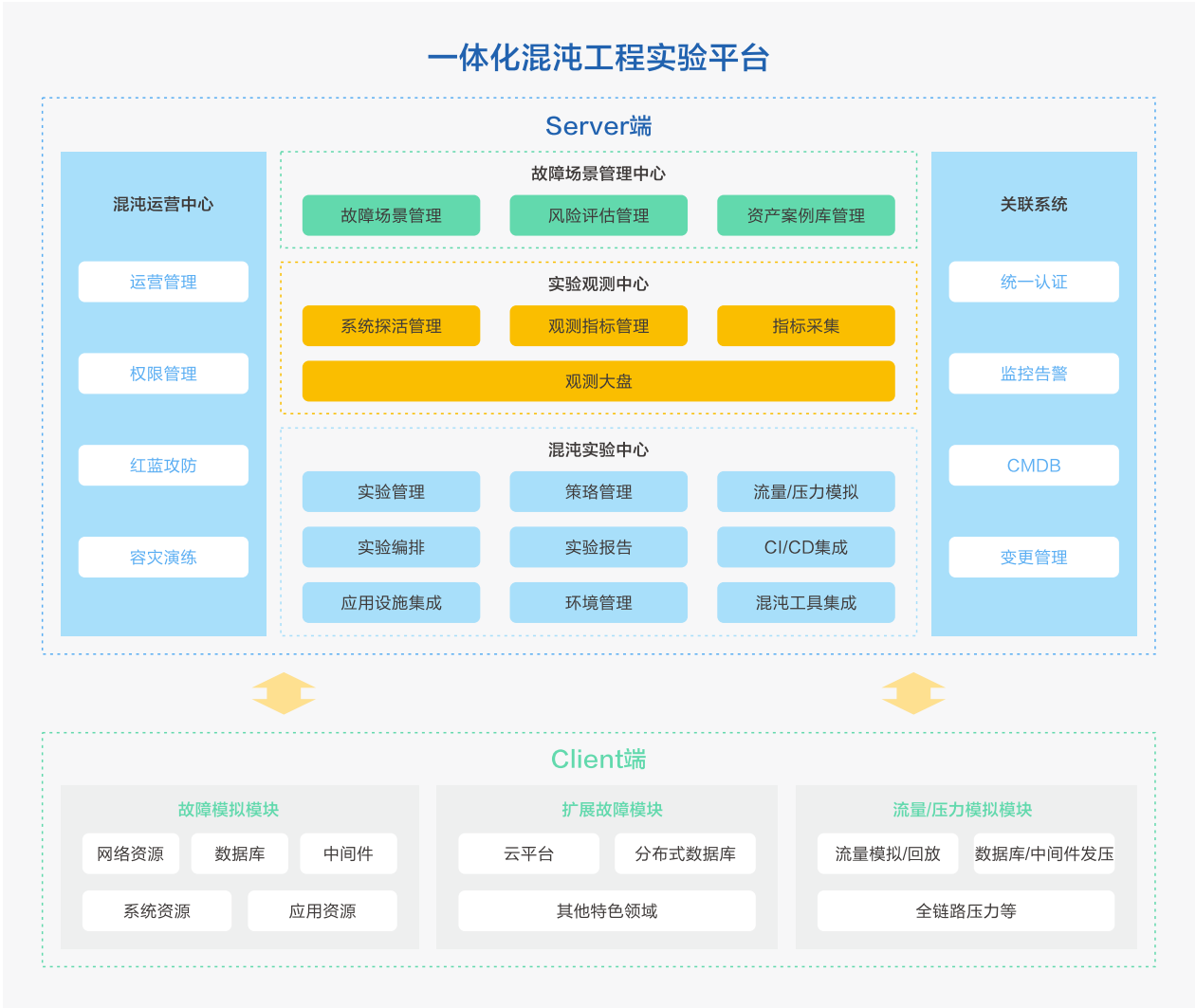


图 18 混沌工程技术平台示例

4.1.3 管理机制

管理机制包括组织形式和配套机制两部分，主要明确混沌工程业务承载的组织、职责和机制。

1. 组织形式

混沌工程的实施需要有组织支撑，可按照孵化、常态化实施两个时期设立相应的组织形式。其中，孵化时期对应于第三章所述建设路径的一、二阶段，常态化实施时期对应三阶段及以后。

孵化时期的组织机制，主要目标是保障技术能力建设、快速捋顺机制流程，并逐步明确适合的工作分工，可包含以下6部分：

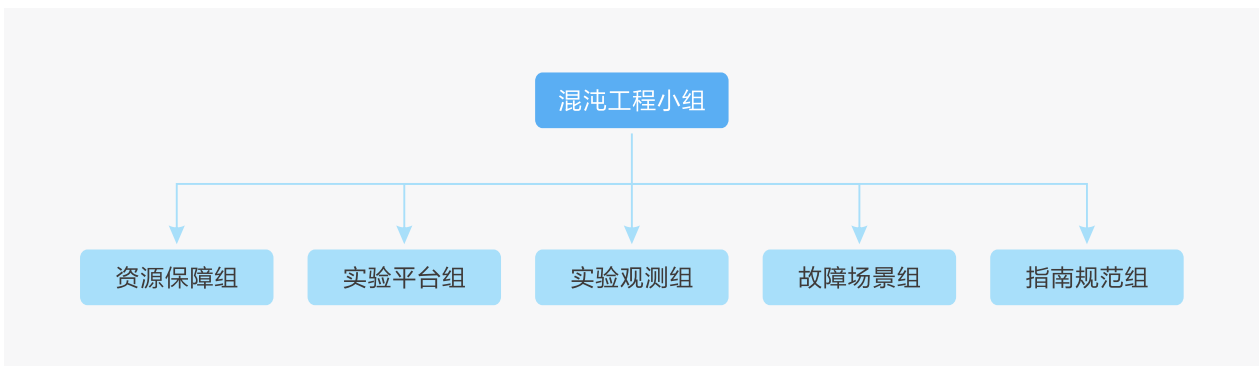


图 19 孵化时期组织形式

混沌工程小组：负责整体统筹并制定实施规划，跟踪计划落地实施；负责组织外部交流；负责组织内部交流和学习，促进技术能力共享；负责适时优化项目和质量管理流程；负责标准实验流程的制定和优化；负责组织实验结果的评价和优化。

资源保障组：负责实验环境管理、统筹资源申请、审批、争议调解。

实验平台组：负责混沌实验平台建设、故障场景平台建设和技术支持；根据使用需求制定平台优化和演进规划。

实验观测组：负责实验观测能力建设和技术支持；制定监控分析能力优化和演进规划；制定红蓝攻防等大规模演练计划并统筹实施。

故障场景组：负责故障场景流程标准的制定、审核、维护。

指南规范组：负责配套规范指南的规划、编制、审核、更新。

常态化时期的组织仍然由混沌工程小组牵头，但职责与分工与孵化时期有较大变化，这个时期的主要目标是通过不同频度的持续实验提前发现系统风险，提升稳定性，可分为4个部分：

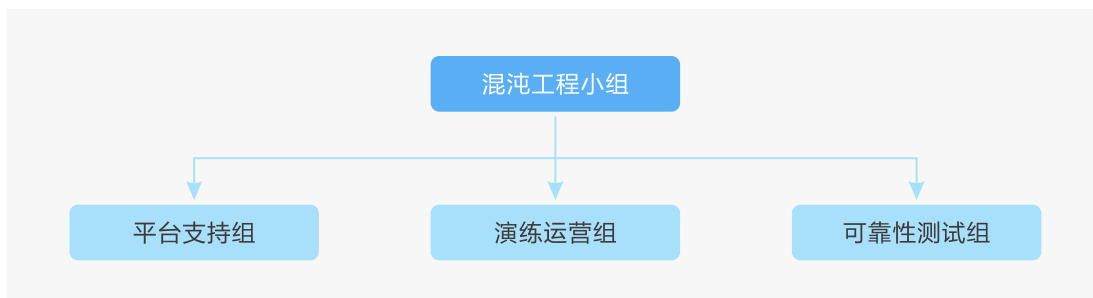


图20 常态化时期组织形式

混沌工程小组：对混沌工程整体进行业务规划和运营管理，通过实施混沌工程降低生产系统的可靠性风险。在常态化运营时期，混沌工程工作可与SRE建设或者稳定性提升工程紧密关联。

平台支持组：负责混沌工程实验平台开发支持工作，通过设计体验优秀的混沌工程平台，提升混沌工程的实施效率。

演练运营组：负责中低频的演练活动组织工作，开展常态化容灾验证、红蓝攻防、生产突袭等系统可靠性和应急预案有效性验证工作，提升业务可用性能力。

可靠性测试组：负责高频的故障注入工作，在测试阶段围绕故障场景随系统或项目开展混沌实验，并逐步沉淀故障场景和规范指南，提出管理流程优化建议。

2. 配套机制

相对于其它新技术，混沌工程对于开发、测试、运维环节均有一定侵入性，因此本文第三章参考原则8建议“尽量复用已有技术和管理机制”，比如环境管理、监控要求、变更管理、问题管理等，以降低组织层面落地成本。在此原则下，混沌工程实践过程中仍然不可避免地需要对管理机制做适应性调整。结合可靠性测试和红蓝攻防（包括生产突袭）两类组织形式，要点如下：

可靠性测试由测试人员或SRE为主，可以在“项目”和“系统”两个维度开展。其中“项目”维度是指随项目生命周期，在非功能测试等环节中，对系统做投产前的稳定性验证测试。“系统维度”是指可靠性测试与项目周期解耦，不受限于研发效率的考核，在系统全生命周期内开展。混沌工程实践早期（如第三章建设路径中提到的一、二阶段），宜按照“系统”维度开展实践，避免研发效率考核成为实验人员开展混沌实验的顾虑。待具备较为完善的自动化实验能力及经验后，再酌情将一些高频、高危且高自动化的故障场景纳入“项目”维度，将测试案例集成到质量门禁，做为稳定性测试准出项。

红蓝攻防及生产突袭一般为组织级行为，是对平日积累的稳定性运营能力的全面综合检验，宜由横向组织牵头开展。对于此类活动的组织，可以和SRE工作相结合，安排专人对接“混沌工程小组”，负责本系统处置、分析和复盘工作。

4.2 评价体系

参考中国信通院《混沌工程成熟度模型》，对于混沌工程小组的工作评价，在实践中可以从技术熟练度以及实践价值两方面进行评价。其中每一方面又可分解为牵引指标和评价流程两个维度，牵引指标用来设定优化方向，评价流程用来验证和优化指标、提升预期，两者形成闭环，迭代优化混沌工程实践水平。

技术熟练度主要评价混沌工程实践过程中相关人员的操作能力水平。牵引指标主要围绕能力体系中故障场景分析、设计、流程规范程度等开展，以提高相关人员能力水平为主要目标。因能力水平相对主观，评价流程宜采用“教练辅导”机制，即通过指引规范宣贯、优秀实践分享、教练团队辅导等方式，以点带面提升人员技术水平。

实践价值可从过程目标和结果目标两个视角切入，过程目标主要关注可靠性风险的发现与拦截，结果目标则是故障数量和恢复时长。实验价值的牵引指标可以参考下表。对于实践价值的评价过程，可以随着可靠性测试在常态化的测试验证环节中进行统计，也可以随着红蓝攻防、生产突袭等组织级演练中进行统计。

需要注意的是，无论是针对技术熟练度还是实践价值的指标，均是探索混沌工程工作如何量化的一种尝试，属于“牵引”范畴，鼓励大家提升能力水平、提前发现和避免风险，并非绩效考核指标。

表 10 混沌工程实践价值牵引指标参考

类型	价值说明	举例
管理价值	提供了一种系统潜在风险评估的手段，从被动运维走向主动运维	构建潜在稳定性风险地图，可视化风险分布和状态
过程指标	风险发现的个数与等级	年度主动发现*级别风险*个
	风险消减的个数，等级，类型（完善了多少监控，完善了多少预案等）	1.风险降级*个，例如*级降到*级 2.风险消减*个 3.增加预案*个 4.增加监控项*个
	故障场景的混沌实验的覆盖率	*个场景，覆盖率多少，例如：容灾场景覆盖*%，依赖场景覆盖*%
	已执行混沌实验场景的恢复速度提升率	*个场景经过混沌实验训练，平均恢复速度从*个小时提升到*分钟
	混沌实验参与人员覆盖率	共*名工程师参与混沌实验，人员覆盖率*
结果指标	故障数量相比上年减少个数	例如*级别故障降低到多少个，总恢复时长从*缩短到*
	故障恢复时长相比上年降低比率	

4.3 企业团队文化

混沌工程的开展除了要构建技术能力和管理机制外，形成一定的文化基础也很有必要，通过技术认同感能够进一步调动混沌工程开展的积极性。对于具体实践中的一些常见问题如何进行正确引导，例如在生产环境实施混沌导致系统故障是否追责，实施人员对于暴露系统风险会不会有顾虑等，以下结合商业银行特点进行初步分析。



4.3.1 技术认同感

1. 混沌工程并不是刻意制造故障

这个问题本质上是一次注入一种故障还是多种故障的问题。实际实施的时候，有人同时注入多个故障，刻意把系统搞瘫，这种做法没有实际意义，也不是从真实故障出发。原因是从历史真实的故障看，虽然故障现象表现为多种，但实际触发因素往往就一种，其他故障现象往往是单个触发因素引起的关联故障。只有从真实故障出发，才能真正验证系统可能面临的风险。因此在实际操作过程中，建议一次只注入同一种类型的故障（可以通过编排间隔注入，但始终都是一种）。

2. 混沌工程不一定要在生产环境实施

混沌工程的通用原则是强调在生产环境实施（见第三章），但并不代表一定要在生产环境实施。由于银行业的强稳定性要求，在生产环境直接实施有很高的业务和监管风险，从实践经验来看，分阶段实施同样能够良好达到“风险左移”效果。例如在测试环境测试可以初步发现部分产品可靠性能力方面的问题。测试完成后可以晋级到准生产环境，在准生产环境进行大量混沌测试，进一步拦截风险。一般而言，准生产环境与生产环境是对等架构建设，同时会镜像真实生产流量、配套完整的监控告警配置。在此环境开展实验能够非常接近生产但又尽量避免贸然在生产环境上开展实验可能导致的风险。

3. 系统实施混沌工程造成故障不一定要追责

这个问题其实没有标准的答案，每个公司的处理办法不同，本质上还是公司对于故障的处理态度问题。业界一般有2种做法，一种是认为故障不是某个人的问题，而是系统、流程、机制整体的问题，所以不追责；另外一种则认为故障是必须追责的，否则大家就不在乎故障了。很难评价这两种做法的好坏，根本还是技术文化不同。国内的普遍做法一般强调追责，但直接追责显然容易打击积极性。互联网企业的实践经验是给出一定的故障预算（免责数量），比如每年多少个低级别的故障以内可以不进行追责，这样可以降低在生产环境中操作的心理负担。商业银行主要在测试环境或准生产环境开展混沌实验，这方面的压力小一些。在测试或者准生产环境的故障预算，可以由商业银行内部自行制定。

4. 如何打消暴露风险的顾虑

混沌工程最终的目的是暴露系统风险，但暴露风险可能会引起系统负责方不适甚至引起争议，如果处理不当，则会形成实施人员不愿意暴露风险的局面，导致混沌工程效果打折。

一般认为，只要能在风险真正发生前暴露并解决，对于安全生产都是有积极意义的。具体实施的时候宜对暴露风险采用正向鼓励的方式，例如将分析和测试出的风险划分风险等级，并且将发现风险的功劳等同于处理同等级故障，打消暴露风险的顾虑。

4.3.2 教练人才培养

同任何一种新技术一样，混沌工程的推广离不开人才队伍的培养。商业银行历来重视金融科技发展，因此IT系统全生命周期中一般有明确职责分工的开发、测试、运维人员。在混沌工程孵化时期（对应第三章建设路径一、二阶段）建议由上述人员兼职担任。与此同时，银行内部最开始引入、孵化和试点混沌工程实践的人员可以作为原始教练团队成员，承担培训试点系统混沌工程人员的职责。随着混沌工程的逐步推广，试点系统的工程师逐步加入到教练团队中，可以对各自系统内、部门内其他技术人员进行培训和宣贯，以点带面逐步扩大队伍。

在常态化实施时期（对应第三章所述三阶段及以后），可结合银行内部组织架构规划，设立专职混沌工程师或结合SRE、稳定性提升等工程，将混沌工程纳入SRE人员日常工作，形成专职工作团队。



第五章

实践案例

结合第四章所述方法，本章以商业银行典型系统及服务——信用卡快捷支付服务为例进一步进行说明。需要注意的是，本章所展示图表中内容仅用于帮助理解方法步骤，非真实生产信息及数据。

步骤1：故障场景识别

表 11 故障场景分析示例

产品/服务	微服务	组件	功能	故障场景名称	故障归类	该场景能覆盖的故障类型	恢复手段	发生概率	应急预案	故障检测方式	目标 TTD (秒)	目标 RTO (分钟)	目标 RPO (分钟)	容错能力评估	该故障发生会造成什么业务影响	严重程度	风险等级
信用卡	快捷支付	Qpay	银联快捷支付 银联快捷退货 网联快捷支付 网联快捷退货	Qpay发生单点不可用	单点	机柜掉电，机柜网络中断，服务器故障，OS故障，应用单点故障	冗余/隔离	中	通过容器集群管理机制自动检测服务状态，自动隔离故障节点并重建新容器	1.有业务交易量的成功率监控，成功率分集群监控，通过该监控指标可以判断故障节点 2.通过容器集群liveness监控机制探测服务状态	60	2	0	冗余能力评估项： 1.是否跨机架部署（leader选举模式至少要跨3个机架） 是(o) 否 不涉及 不清楚 2.机架故障后，剩余容量是否充足 是(o) 否 不涉及 不清楚	有冗余措施，单点故障不会造成业务影响	P4	低
信用卡	快捷支付	Qpay	银联快捷支付 银联快捷退货 网联快捷支付 网联快捷退货	Qpay发生机房级故障	局部	机房掉电，高温，机房断网（东西向，南北向，孤岛），批次性硬件故障，应用局部故障	容灾	低	通过容灾的方式切换到其他园区	有业务交易量的成功率监控，成功率分园区监控，通过该监控指标可以判断故障机房	60	同城2 异地10	0	容灾能力评估项： 1.是否进行了跨机房容灾部署 是(o) 否 不涉及 不清楚 2.服务是否是否进行异地容灾部署 是(o) 否 不涉及 不清楚 3.是否进行cell化设计且流量单元内收敛 是(o) 否 不涉及 不清楚 4.同城容灾节点是否承担生产流量 是(o) 否 不涉及 不清楚 5.异地容灾节点是否承担生产流量 是 否(o) 不涉及 不清楚 6.容灾切换工具在灾难情况下是否保证可用 是(o) 否 不涉及 不清楚 7.如容灾节点不承担生产流量，是否有有效性检验机制 是(o) 否 不涉及 不清楚	双活角度： 50%的客户交易失败持续2分钟 异地容灾角度： 100%的客户交易失败持续10分钟	同城：P4 异地：P4	低
信用卡	快捷支付	Qpay	银联快捷支付 银联快捷退货 网联快捷支付 网联快捷退货	Qpay发生支付请求过载	过载	主要考虑网络流量过载，链接数过载，请求量过载	过载保护	高	事先配置好限流阈值，如果过载是正常行为，扩容解决	触发流控会有对应的错误码监控	60	0	0	过载保护能力评估项： 1.系统是具备限流能力 是(o) 否 不涉及 不清楚 2.是否配置了合适的流控规则 是(o) 否 不涉及 不清楚 3.是否具备熔断降级能力防止下游雪崩 是(o) 否 不涉及 不清楚 4.是否具备横向扩容能力 是(o) 否 不涉及 不清楚	被限流的客户交易失败，部分客户交易客户的响应时间会增大	P4	中

步骤2：风险等级评估与优先级排序

表 12 风险评估示例

产品/服务	微服务	组件	功能	故障场景名称	故障归类	该场景能覆盖的故障类型	恢复手段	发生概率	应急预案	故障检测方式	目标TTD(秒)	目标RTO(分钟)	目标RPO(分钟)	容错能力评估	该故障发生会造成什么业务影响	严重程度	风险等级	优先级排序
信用卡	快捷支付	Qpay	银联快捷支付 银联快捷退货 网联快捷支付 网联快捷退货	Qpay发生单点不可用	单点	机柜断电，机柜网络中断，服务器故障，OS故障，应用单点故障	冗余/隔离	中	通过容器集群管理机制自动检测服务状态，自动隔离故障节点并重建新容器	1.有业务交易量的成功率监控，成功率分集群监控，通过该监控指标可以判断故障节点 2.通过容器集群liveness监控机制探测服务状态	60	2	0	冗余能力评估项： 1.是否跨机架部署（leader选举模式至少要跨3个机架） 是(o) 否 不涉及 不清楚 2.机架故障后，剩余容量是否充足 是(o) 否 不涉及 不清楚	有冗余措施，单点故障不会造成业务影响	P4	低	中
信用卡	快捷支付	Qpay	银联快捷支付 银联快捷退货 网联快捷支付 网联快捷退货	Qpay发生机房级故障	局部	机房断电，高温，机房断网（东西向，南北向，孤岛），批次性硬件故障，应用局部故障	容灾	低	通过容灾的方式切换到其他园区	有业务交易量的成功率分园区监控，通过该监控指标可以判断故障机房	60	同城2 异地10	0	容灾能力评估项： 1.是否进行了跨机房容灾部署 是(o) 否 不涉及 不清楚 2.服务是否是否进行异地容灾部署 是(o) 否 不涉及 不清楚 3.是否进行cell化设计且流量单元内收敛 是(o) 否 不涉及 不清楚 4.同城容灾节点是否承担生产流量 是(o) 否 不涉及 不清楚 5.异地容灾节点是否承担生产流量 是 否(o) 不涉及 不清楚 6.容灾切换工具在灾难情况下是否保证可用 是(o) 否 不涉及 不清楚 7.如容灾节点不承担生产流量，是否有有效性检验机制 是(o) 否 不涉及 不清楚	双活角度： 50%的客户交易失败持续2分钟 异地容灾角度： 100%的客户交易失败持续10分钟	同城：P4 异地：P4	低	高
信用卡	快捷支付	Qpay	银联快捷支付 银联快捷退货 网联快捷支付 网联快捷退货	Qpay发生支付请求过载	过载	主要考虑网络流量过载，链接数过载，请求量过载	过载保护	高	事先配置好限流阈值，如果过载是正常行为，扩容解决	触发流控会有对应的错误码监控	60	0	0	过载保护能力评估项： 1.系统是具备限流能力 是(o) 否 不涉及 不清楚 2.是否配置了合适的流控规则 是(o) 否 不涉及 不清楚 3.是否具备熔断降级能力防止下游雪崩 是(o) 否 不涉及 不清楚 4.是否具备横向扩容能力 是(o) 否 不涉及 不清楚	被限流的客户交易失败，部分客户交易客户的响应时间会增大	P4	中	高

步骤3：故障注入和爆炸半径设计

表 13 爆炸半径分析示例

产品/服务	微服务	组件	功能	故障场景名称	故障归类	该场景能覆盖的故障类型	恢复手段	发生概率	应急预案	故障检测方式	目标TTD(秒)	目标RTO(分钟)	目标RPO(分钟)	容错能力评估	该故障发生会造成什么业务影响	严重程度	风险等级	优先级排序	故障注入设计	爆炸半径
信用卡	快捷支付	Qpay	银联快捷支付 银联快捷退货 网联快捷支付 网联快捷退货	Qpay发生单点不可用	单点	机柜掉电，机柜网络中断，服务器故障，OS故障，应用单点故障	冗余/隔离	中	通过容器集群管理机制自动检测服务状态，自动隔离故障节点并重建新容器	1.有业务交易量的成功率监控，成功率分集群监控，通过该监控指标可以判断故障节点 2.通过容器集群liveness监控机制探测服务状态	60	2	0	冗余能力评估项： 1.是否跨机架部署（leader选举模式至少要跨3个机架） 是(o) 否 不涉及 不清楚 2.机架故障后，剩余容量是否充足 是(o) 否 不涉及 不清楚	有冗余措施，单点故障不会造成业务影响	P4	低	中	选择qpay服务的一个pod，注入网络中断故障	节点级
信用卡	快捷支付	Qpay	银联快捷支付 银联快捷退货 网联快捷支付 网联快捷退货	Qpay发生机房级故障	局部	机房掉电，高温，机房断网（东西向，南北向，孤岛），批次性硬件故障，应用局部故障	容灾	低	通过容灾的方式切换到其他园区	有业务交易量的成功率监控，成功率分园区监控，通过该监控指标可以判断故障机房	60	同城2 异地10	0	容灾能力评估项： 1.是否进行了跨机房容灾部署 是(o) 否 不涉及 不清楚 2.服务是否是否进行异地容灾部署 是(o) 否 不涉及 不清楚 3.是否进行cell化设计且流量单元内收敛 是(o) 否 不涉及 不清楚 4.同城容灾节点是否承担生产流量 是(o) 否 不涉及 不清楚 5.异地容灾节点是否承担生产流量 是 否(o) 不涉及 不清楚 6.容灾切换工具在灾难情况下是否保证可用 是(o) 否 不涉及 不清楚 7.如容灾节点不承担生产流量，是否有有效性检验机制 是(o) 否 不涉及 不清楚	双活角度： 50%的客户交易失败持续2分钟 异地容灾角度： 100%的客户交易失败持续10分钟	同城：P4 异地：P4	低	高	同城容灾：通过k8s关停A机房所有qpay服务	服务级
信用卡	快捷支付	Qpay	银联快捷支付 银联快捷退货 网联快捷支付 网联快捷退货	Qpay发生支付请求过载	过载	主要考虑网络流量过载，链接数过载，请求量过载	过载保护	高	事先配置好限流阈值，如果过载是正常行为，扩容解决	触发流控会有对应的错误码监控	60	0	0	过载保护能力评估项： 1.系统是具备限流能力 是(o) 否 不涉及 不清楚 2.是否配置了合适的流控规则 是(o) 否 不涉及 不清楚 3.是否具备熔断降级能力防止下游雪崩 是(o) 否 不涉及 不清楚 4.是否具备横向扩容能力 是(o) 否 不涉及 不清楚	被限流的客户交易失败，部分客户交易客户的响应时间会增大	P4	中	高	使用压测工具对4个接口发压5倍流量	服务级

步骤4：稳态假设

由于Qpay服务针对服务可用性，延迟，错误率等设置了紧急告警。对于此类情况可认为无紧急告警即为稳态。

表 14 信用卡快捷支付服务稳态设置示例

产品/服务	微服务	组件	功能	故障场景名称	故障归类	该场景能覆盖的故障类型	恢复手段	发生概率	应急预案	故障检测方式	目标 TTD (秒)	目标 RTO (分钟)	目标 RPO (分钟)	容错能力评估	该故障发生会造成什么业务影响	严重程度	风险等级	优先级排序	故障注入设计	爆炸半径	故障紧急中止方法	回退时间评估 (分钟)	稳态假设
信用卡	快捷支付	Qpay	银联快捷支付 银联快捷退货 网联快捷支付 网联快捷退货	Qpay发生单点不可用	单点	机柜掉电，机柜网络中断，服务器故障，OS故障，应用单点故障	冗余/隔离	中	通过容器集群管理机制自动检测服务状态，自动隔离故障节点并重建新容器	1.有业务交易量的成功率监控，成功率分集群监控，通过该监控指标可以判断故障节点 2.通过容器集群liveness监控机制探测服务状态	60	2	0	冗余能力评估项： 1.是否跨机架部署（leader选举模式至少要跨3个机架） 是(o) 否 不涉及不清楚 2.机架故障后，剩余容量是否充足 是(o) 否 不涉及不清楚	有冗余措施，单点故障不会造成业务影响	P4	低	中	选择qpay服务的一个pod，注入网络中断故障	节点级	中止故障注入，恢复网络，如果无法恢复，则重启服务器进行恢复	1	无紧急告警
信用卡	快捷支付	Qpay	银联快捷支付 银联快捷退货 网联快捷支付 网联快捷退货	Qpay发生机房级故障	局部	机房掉电，高温，机房断网（东西向，南北向，孤岛），批次性硬件故障，应用局部故障	容灾	低	通过容灾的方式切换到其他园区	有业务交易量的成功率监控，成功率分园区监控，通过该监控指标可以判断故障机房	60	同城2 异地10	0	容灾能力评估项： 1.是否进行了跨机房容灾部署 是(o) 否 不涉及不清楚 2.服务是否是否进行异地容灾部署 是(o) 否 不涉及不清楚 3.是否进行cell化设计且流量单元内收敛 是(o) 否 不涉及不清楚 4.同城容灾节点是否承担生产流量 是(o) 否 不涉及不清楚 5.异地容灾节点是否承担生产流量 是 否(o) 不涉及不清楚 6.容灾切换工具在灾难情况下是否保证可用 是(o) 否 不涉及不清楚 7.如容灾节点不承担生产流量，是否有有效性检验机制 是(o) 否 不涉及不清楚	双活角度： 50%的客户交易失败持续2分钟 异地容灾角度： 100%的客户交易失败持续10分钟	同城：P4 异地：P4	低	高	同城容灾：通过k8s关停A机房所有qpay服务 服务级	同城容灾：通过K8S重新启动A机房Qpay服务	1	无紧急告警	
信用卡	快捷支付	Qpay	银联快捷支付 银联快捷退货 网联快捷支付 网联快捷退货	Qpay发生支付请求过载	过载	主要考虑网络流量过载，链接数过载，请求量过载	过载保护	高	事先配置好限流阈值，如果过载是正常行为，扩容解决	触发流控会有对应的错误码监控	60	0	0	过载保护能力评估项： 1.系统是具备限流能力 是(o) 否 不涉及不清楚 2.是否配置了合适的流控规则 是(o) 否 不涉及不清楚 3.是否具备熔断降级能力防止下游雪崩 是(o) 否 不涉及不清楚 4.是否具备横向扩容能力 是(o) 否 不涉及不清楚	被限流的客户交易失败，部分客户交易客户的响应时间会增大	P4	中	高	使用压测工具对4个接口发压5倍流量 服务级	停止压测	1	无紧急告警	

步骤5：计划与周知

表 15 实验计划示例

产品/服务	微服务	组件	功能	故障场景名称	故障归类	该场景能覆盖的故障类型	恢复手段	发生概率	应急预案	故障检测方式	目标TTD (秒)	目标RTO (分钟)	目标RPO (分钟)	该故障发生会造成什么业务影响	严重程度	风险等级	优先级排序	故障注入设计	爆炸半径	故障紧急中止方法	回退时间评估 (分钟)	稳态假设	实验日期	参与人
信用卡	快捷支付	Qpay	银联快捷支付 银联快捷退货 网联快捷支付 网联快捷退货	Qpay发生单点不可用	单点	机柜停电, 机柜网络中断, 服务器故障, 应用单点故障	冗余/隔离	中	通过容器集群管理机制自动检测服务状态, 自动隔离故障节点并重建新容器	1.有业务交易量的成功率监控, 成功率分集群监控, 通过该监控指标可以判断故障节点 2.通过容器集群liveness监控机制探测服务状态	60	2	0	有冗余措施, 单点故障不会造成业务影响	P4	低	中	选择qpay服务的一个pod, 注入网络中断故障	节点级	中止故障注入, 恢复网络, 如果无法恢复, 则重启服务器进行恢复	1	无紧急告警	2023/4/18	混沌工程小组
信用卡	快捷支付	Qpay	银联快捷支付 银联快捷退货 网联快捷支付 网联快捷退货	Qpay发生机房级故障	局部	机房停电, 高温, 机房断网 (东西向, 南北向, 孤岛), 批次性硬件故障, 应用局部故障	容灾	低	通过容灾的方式切换到其他园区	有业务交易量的成功率分园区监控, 通过该监控指标可以判断故障机房	60	同城2异地10	0	双活角度, 50%的客户交易失败持续2分钟 异地容灾角度: 100%的客户交易失败持续10分钟	同城, P4 异地, P4	低	高	同城容灾: 通过k8s关停A机房所有qpay服务	服务器级	同城容灾: 通过K8S重新启动A机房Qpay服务	1	无紧急告警	2023/4/18	混沌工程小组
信用卡	快捷支付	Qpay	银联快捷支付 银联快捷退货 网联快捷支付 网联快捷退货	Qpay发生支付请求过载	过载	主要考虑网络流量过载, 链接数过载, 请求量过载	过载保护	高	事先配置好限流阈值, 如果过载是正常行为, 扩容解决	触发流控会有对应的错误码监控	60	0	0	被限流的客户交易失败, 部分客户交易客户的响应时间会增大	P4	中	高	使用压测工具对4个接口发压5倍流量	服务器级	停止压测	1	无紧急告警	2023/4/18	混沌工程小组

步骤6：执行实验

表 16 实验执行示例

产品/服务	微服务	组件	功能	故障场景名称	故障归类	该场景能覆盖的故障类型	恢复手段	发生概率	应急预案	故障检测方式	目标TTD (秒)	目标RTO (分钟)	目标RPO (分钟)	该故障发生会造成什么业务影响	故障注入设计	爆炸半径	故障紧急中止方法	回退时间评估 (分钟)	稳态假设	实验日期	参与人	实验TTD (秒)	实验RTO (分钟)	实验RPO (分钟)	结果是否达标
信用卡	快捷支付	Qpay	银联快捷支付 银联快捷退货 网联快捷支付 网联快捷退货	Qpay发生单点不可用	单点	机柜停电, 机柜网络中断, 服务器故障, 应用单点故障	冗余/隔离	中	通过容器集群管理机制自动检测服务状态, 自动隔离故障节点并重建新容器	1.有业务交易量的成功率监控, 成功率分集群监控, 通过该监控指标可以判断故障节点 2.通过容器集群liveness监控机制探测服务状态	60	2	0	有冗余措施, 单点故障不会造成业务影响	选择qpay服务的一个pod, 注入网络中断故障	节点级	中止故障注入, 恢复网络, 如果无法恢复, 则重启服务器进行恢复	1	无紧急告警	2023/4/18	混沌工程小组	60	1	0	监控能力: 达标 恢复能力: 达标 容错能力: 达标
信用卡	快捷支付	Qpay	银联快捷支付 银联快捷退货 网联快捷支付 网联快捷退货	Qpay发生机房级故障	局部	机房停电, 高温, 机房断网 (东西向, 南北向, 孤岛), 批次性硬件故障, 应用局部故障	容灾	低	通过容灾的方式切换到其他园区	有业务交易量的成功率分园区监控, 通过该监控指标可以判断故障机房	60	同城2异地10	0	双活角度: 50%的客户交易失败持续2分钟 异地容灾角度: 100%的客户交易失败持续10分钟	同城容灾: 通过k8s关停A机房所有qpay服务	服务器级	同城容灾: 通过K8S重新启动A机房Qpay服务	1	无紧急告警	2023/4/18	混沌工程小组	60	2	0	监控能力: 达标 恢复能力: 达标 容错能力: 达标
信用卡	快捷支付	Qpay	银联快捷支付 银联快捷退货 网联快捷支付 网联快捷退货	Qpay发生支付请求过载	过载	主要考虑网络流量过载, 链接数过载, 请求量过载	过载保护	高	事先配置好限流阈值, 如果过载是正常行为, 扩容解决	触发流控会有对应的错误码监控	60	0	0	被限流的客户交易失败, 部分客户交易客户的响应时间会增大	使用压测工具对4个接口发压5倍流量	服务器级	停止压测	1	无紧急告警	2023/4/18	混沌工程小组	60	0	0	监控能力: 达标 恢复能力: 达标 容错能力: 达标

步骤7：复盘与改进

表 17 复盘示例

AAR复盘表	
我们原先期望发生什么？	系统同城容灾切换时间(RTO)2分钟
实际发生了什么？	容灾切换2分钟完成
造成差异的原因是什么？	暂无差异
如何改进？	符合要求

第六章 未来展望

商业银行数字化转型的大趋势不可逆，在享受分布式、云原生等技术带来“高性能、高弹性、低成本”的优势同时，如何保持服务“高可用、高标准、低风险”将是大家共同面临的挑战。混沌工程作为商业银行IT系统稳定性提升的重要抓手，在技术、应用层面都将迎来高速发展期。



6.1 技术层面展望

6.1.1 自动化故障场景分析

类似于业务上线前的PRR（Production Readiness Review，生产准备审核），故障场景分析的目的是将某个系统中的风险挖掘出来。当前用的是FMEA的方式由人工进行分析，未来随着对业务架构、组件逻辑关系动态采集和汇总，结合企业内部的CMDB系统可将稳定性项目规则化，最终实现自动化的故障场景分析。

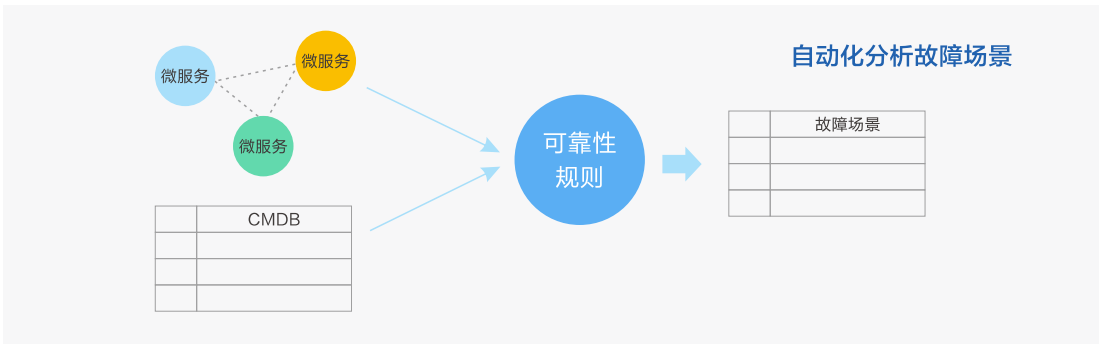


图21 自动化故障场景分析示意

6.1.2 超大规模故障注入和回退

当系统规模超大时，会面临着数以万计的服务器或容器同时注入故障的情形，但这并不是一个简单的“任务传达”。例如对一个机房做断网演习，但机房内一部分业务还不支持容灾、无法参与演练，因此机房断网这种手段就无法使用，需要做到IP粒度的断网。此外，如何保证迅速回退故障，也是待研究的问题。

6.1.3 请求级别的无侵入故障注入

关键业务的性能以及可用性是稳定性保障的重中之重，在混沌工程实践中通常有着更高的任务优先级。但现有的方法暂时无法做到请求级别的故障注入，即在复杂流量下无法精确控制到只对特定的请求进行故障注入。解决此问题的初步思路是可以基于eBPF（extended Berkeley Packet Filter，直译为扩展的伯克利数据包过滤机制）技术，以非侵入的方式，实现针对特定请求的延迟、错误、丢失、超时等多种故障类型模拟和注入。

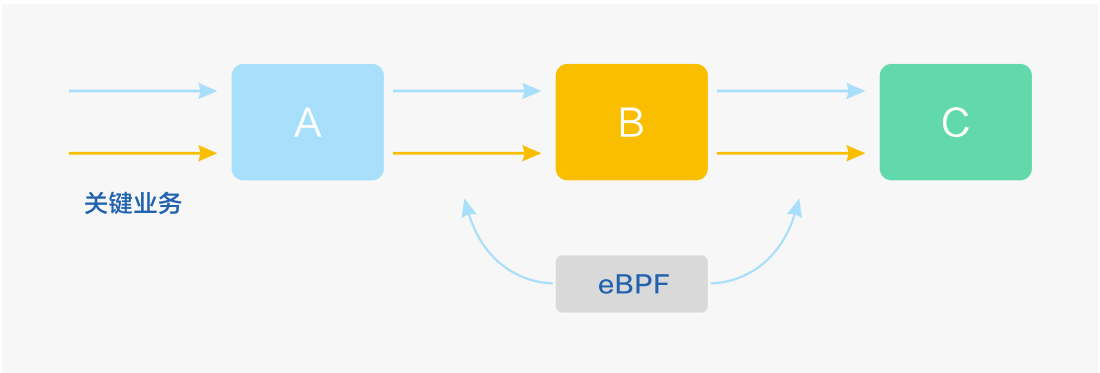


图22 基于eBPF的无侵入故障注入示意

6.2 应用层面展望

6.2.1 混沌工程信心值模型

商业银行历来重视系统稳定性建设，也对系统稳定性水平有着多种评估方法。是否都能结合混沌工程的实施以及相关风险的逐步解决，进一步给出系统的信心值模型？初步思路是可以综合系统稳定性状态、工程师训练熟练度、实验频率、风险挖掘和验证数量、风险处理比例等指标得出。例如某个场景经过生产突袭，恢复时间达标，那该服务就会增加得分，信心值可以作为混沌工程的价值指标，将各种过程指标、结果指标收敛成一个指标，更好地衡量混沌工程实施效果。

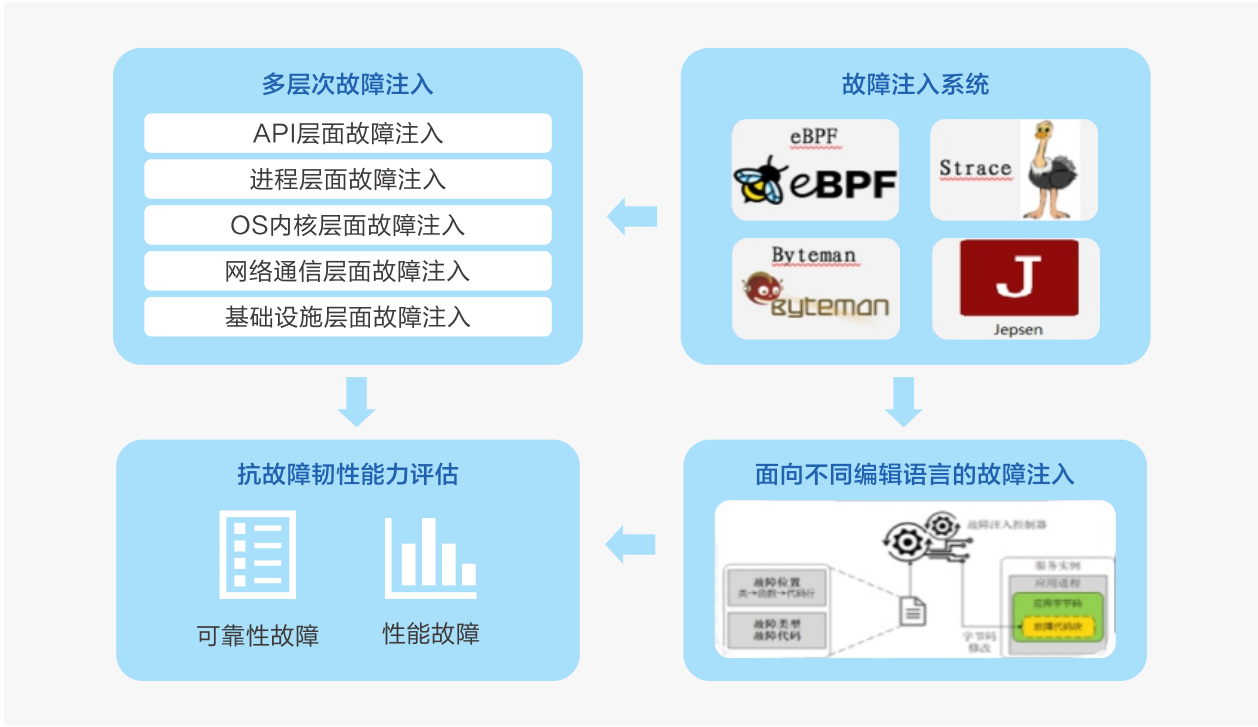


图 23 通过混沌工程评价稳定性信心值

6.2.2 领域故障场景库建设

商业银行IT系统稳定性提升是行业共同面临的挑战，随着混沌工程的逐步深入，可以结合行业特点、业务特性，建设商业银行领域甚至金融领域级别的故障场景库，针对行业特点固化和共享实验场景，共促行业生态建设，共同收益。

第七章 总结

本白皮书是在研究了大量业界案例和优秀企业实践的基础上，结合商业银行特点总结出的体系化落地理论和实践。不仅对“重工具、缺场景”“重理论、缺实践”的情况进行了探讨，也给出了大规模实施混沌工程涉及的组织和管文化问题、平台建设中的思考。这些方法及经验的总结，希望能对商业银行混沌工程的运用和推广提供借鉴参考，帮助商业银行用好混沌工程，最大程度释放其在稳定性建设方面的价值。也期望通过白皮书获取同行反馈、互相交流，共同丰富和提升商业银行稳定性建设能力。

附录

术语解释

表 18 术语解释

名词	解释
FMEA	故障模式和影响面分析，是对一个系统进行故障场景分析的方法。
FT-FMEA	基于容错视角的故障场景分析方法，在FMEA的基础上融入7维故障分析框架和SRE要素修改得出。
稳态假设	是对系统稳态的定义，一般包含监控和用户报障两个方面。
故障场景分析/ 故障假设	根据目标系统的架构和部署情况，通过FT-FMEA分析法得出该系统的故障场景列表。
故障注入	指通过技术手段对目标系统进行操作，使目标系统产生故障效果。
故障域	故障的影响的范围，可以从硬件角度划分（例如服务器级、机柜级、机房级等）或软件角度划分（例如节点级、服务级等）。
爆炸半径	故障注入的范围，与故障域可以同等理解。
故障注入设计	针对一个“故障场景”，设计故障模拟的方法。
SLI	Service Level Indicators，指某个能体现业务质量的指标，例如请求延迟，可用性，错误率等。
SLO	Service Level Object，指一个业务期望达到的质量目标，也就是常说的3个9或4个9，SLO是内部研发、运维、产品共同制定的结果。
ErrorBudget	ErrorBudget一般称为错误预算，即一年允许的不可用时长，一般业务和运维通过这个指标来权衡开发功能和做稳定性的冲突，超过错误预算，主要工作为稳定性提升，不超过业务预算，主要工作为功能研发。
DiRT	灾难恢复测试。
VALET	指Volume（容量）、Availability（可用性）、Latency（延迟）、Error（错误）、Ticket（工单）5个维度的监控，是SRE常用监控指标。
eBPF	一种能够在内核运行沙箱程序的技术，提供了一种在内核事件和用户程序事件发生时安全注入代码的机制，使得非内核开发人员也可以对内核进行控制。

中国农业银行

北京东城区建国门内大街69号

客服电话：95599

邮政编码：100005

门户网站：www.abchina.com

华为技术有限公司

深圳龙岗区坂田华为基地

客服电话：+86 755 28780808

邮政编码：518129

门户网站：www.huawei.com

商标声明

 中国农业银行， “中国农业银行” 以及其它任何用来标识农业银行和农业银行各项业务的文字、图形及其它可视性标识，其知识产权都归属于我行。

他人或他方未经许可不得修改、复制或以其它方式使用上述商标和标识；任何试图淡化或丑化上述商标和标识的行为都是违法的，我行保留追究其法律责任的权利。

 HUAWEI, HUAWEI,  是华为技术有限公司商标或者注册商标，在本手册中以及本手册描述的产品中，出现的其它商标，产品名称，服务名称以及公司名称，由其各自的所有人拥有。

版权说明

本手册版权属于中国农业银行与华为技术有限公司，受法律保护。转载、摘编或其它使用本手册文字、观点、图片、图表或其他信息内容的，应注明“来源:中国农业银行 华为技术有限公司”。违反上述声明者，将追究其相关法律责任。

免责声明

本手册可能含有预测信息，包括但不限于有关未来的财务、运营、产品系列、新技术等信息。由于实践中存在很多不确定因素，可能导致实际结果与预测信息有很大的差别。因此，本手册信息仅供参考，不构成任何要约或承诺，中国农业银行与华为技术有限公司不对您在本手册基础上做出的任何行为承担责任。中国农业银行与华为技术有限公司可能不经通知修改上述信息，恕不另行通知。