28.一种行之有效的防错策略：在支付系统中实施防呆设计的实践_V20240128

聊个支付人都会碰到的问题：资损防控。做支付如果还没有碰到过资损，那就是做得时间还不够久。资损防控是一个很大的话题，需要开几篇文章才能讲完，今天只从一件小事入手聊一个简单而又行之有效的防错策略：防呆设计的实践。

1. 一个线上事故

曾经处理过一个资损事件，很典型，值得说道说道。

一个研发同学在线下测试环境做测试，为图方便，直接从生产捞取一段日志的参数做为请求参数，不幸的是，线下测试环境竟然配置了外部第三方的生产环境参数，导致真实资金被错误地转入个人账户，造成了平台的资金损失。尽管损失金额不大，但由于操作不规范，事件的性质非常严重。

这类事件出来后，通常会发起所谓的复盘，然后给出一堆的整改措施，流程规范、管理制度、设计优化等。我们今天抛开那些大而全（或者华而不实）的整改措施，单单聊聊如何通过“防呆设计”来预防此类事件的发生。

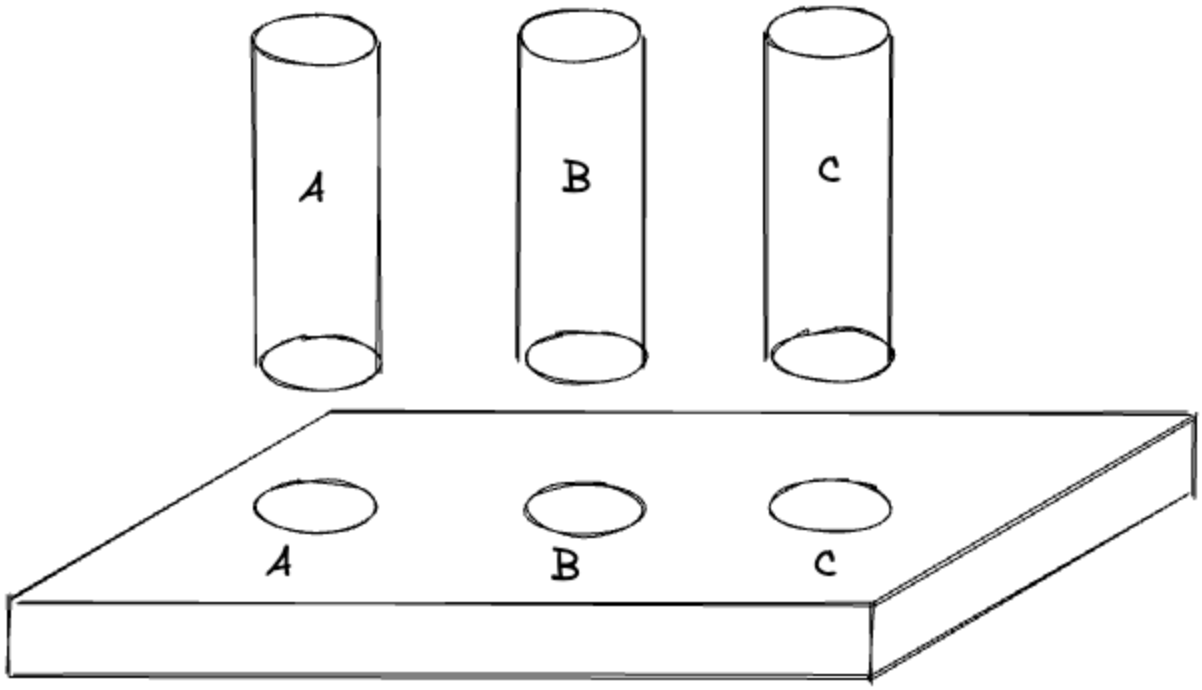
2. 防呆设计定义及在工业界的应用

“防呆设计”（日语：ポカヨケ poka yoke）是一种预防性设计策略，目的是通过限制方法减少错误的发生。用户在无需额外注意力、经验或专业知识的情况下，也能准确无误地完成操作。

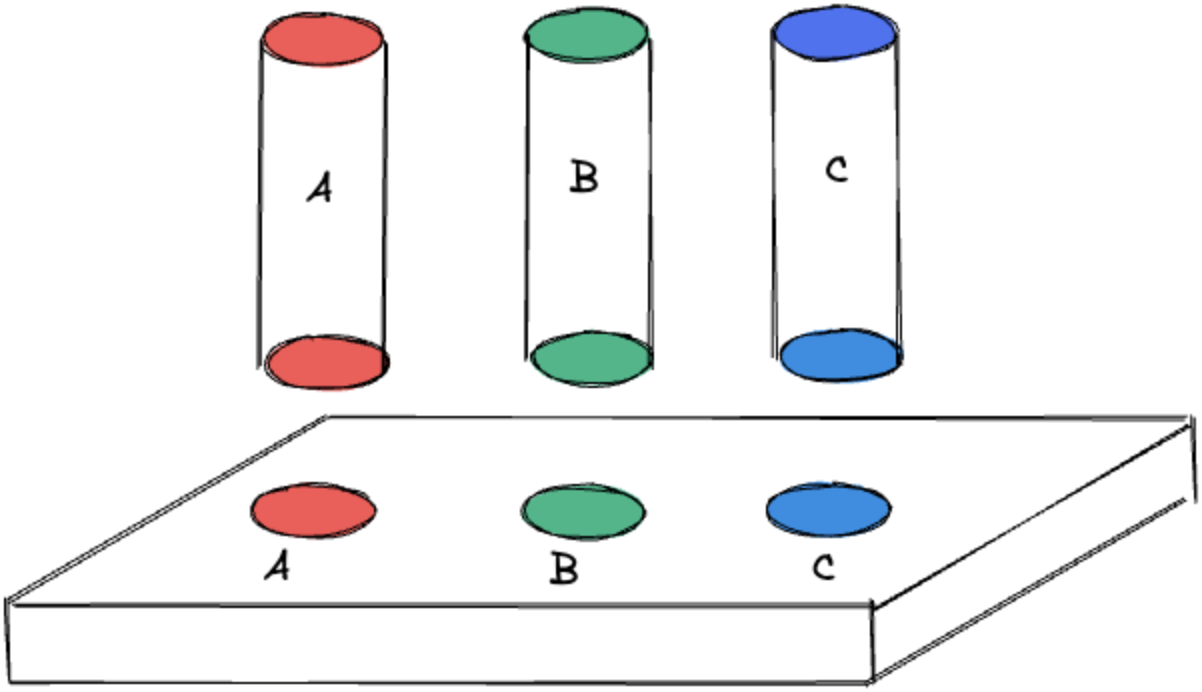
这个概念起源于日本，被广泛应用于丰田汽车的生产过程中，随着时间的推移，已成为全球范围内广泛采用的设计策略。

在工业设计中，防呆设计的例子比比皆是。例如，USB接口的设计确保了只有正确方向才能插入，而Type-C接口则进一步简化，支持双面插入。

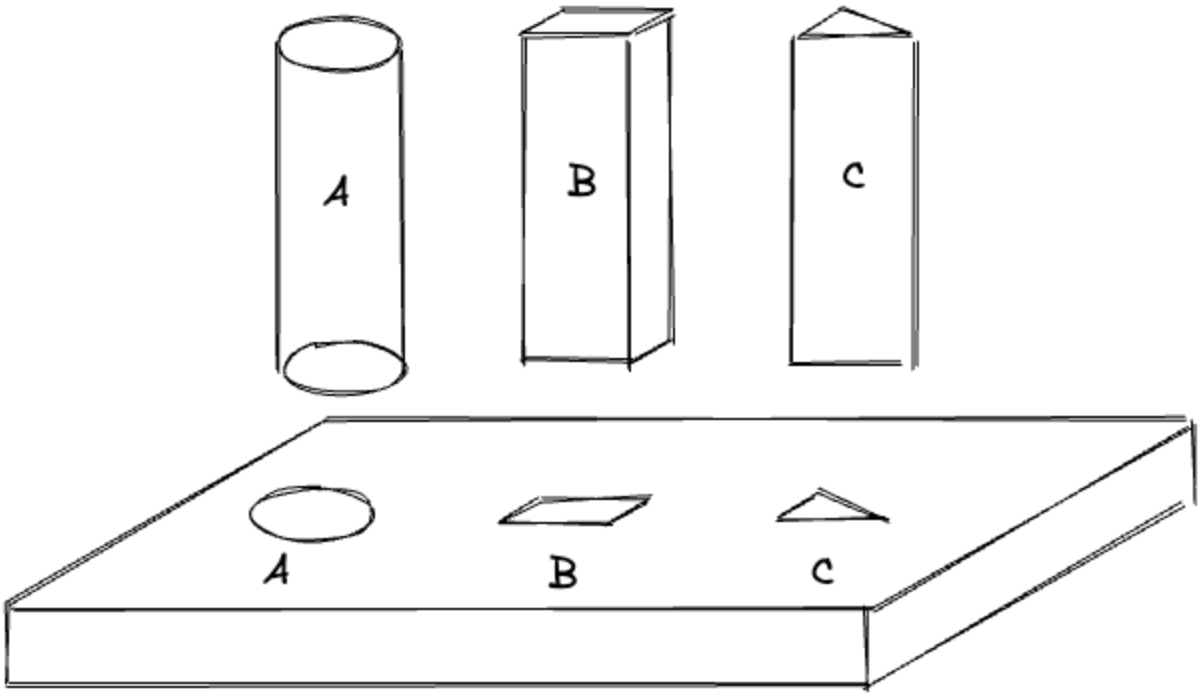
第一版：使用字母标识，容易出错。



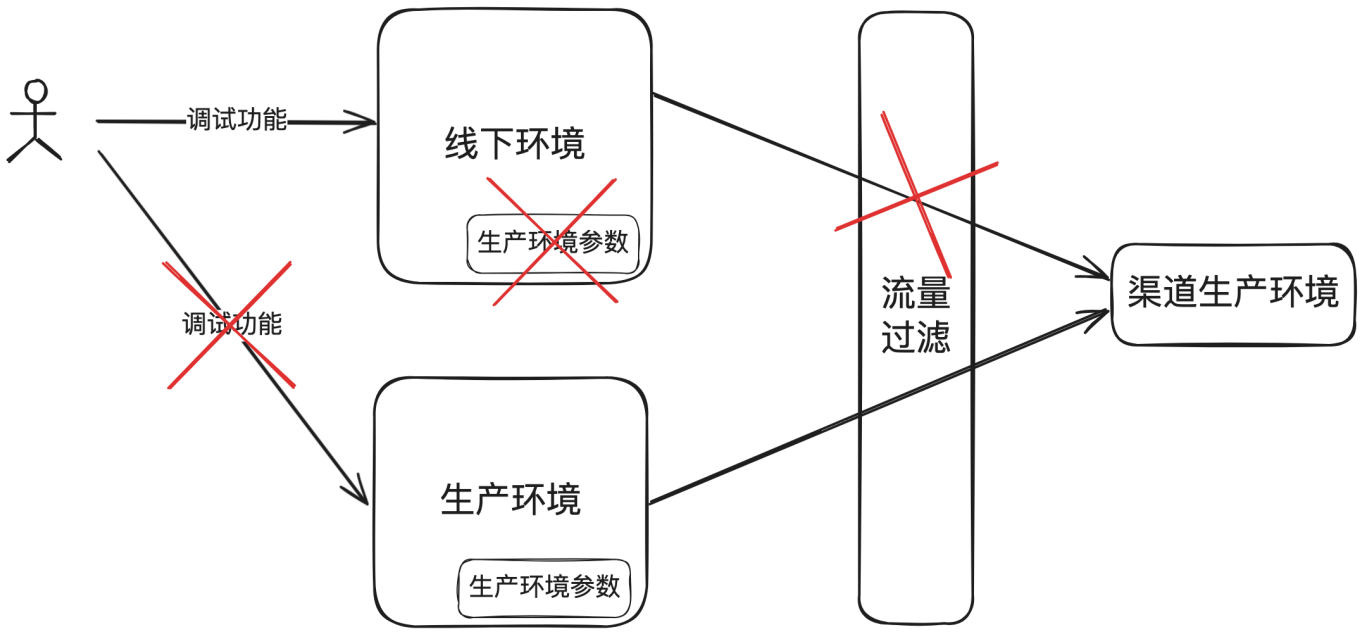
第二版：使用颜色标识，减少出错。



第三版（防呆）：不同形状只能插到不同的位置，想错都错不了。



3. 针对事故的防呆设计措施



再说回到开头所说的故障，我们应该使用“防呆”设计的思想，彻底阻断使用者犯错的可能性。具体怎么做呢？如下：

1. **环境隔离**：确保所有测试环境均配置为外部渠道的测试参数，防止测试操作影响真实交易环境。
2. **流量控制中间件**：在系统中集成一个流量控制中间件，维护一个外部渠道的生产环境白名单。通过中间件筛选请求来源，若来源于测试环境，则自动阻断请求。
3. **禁用线上环境的直接操作**：关闭线上环境调试、手动直接调用能力。
4. **增加操作审核**：对线上环境的操作引入审核机制，由具备风险控制能力的团队成员负责审核。

通过实施上述防呆措施，即使是经验不足的新员工也能避免类似的资损风险。防呆设计并非只是解决问题的手段，而是一种预防性策略，**旨在通过系统设计减少人为错误的可能性**。

4. 支付系统中防呆设计的应用

在支付行业中，防呆设计随处可见，举几个例子：

1. 拆分输入框：信用卡绑卡页面将有效期拆分为两个输入框，分别用于输入月份和年份，从而减少输入错误。把用户的First Name和Last Name分开两个输入框。（我还真见过有支付平台提供的**First Name和Last Name使用一个输入框，导致信用卡绑卡成功率持续很低**）。
2. 自动超时退出：在线支付系统在用户一段时间内无操作时，会自动退出，以防止未经授权的使

用

3. 限制尝试次数：支付系统会限制密码或验证码输入错误的次数，超过限制会暂时锁定账户，以防试图破解。
4. 防止重复提交：为防止用户多次点击造成重复交易，支付系统通常会在首次点击后禁用提交按钮
5. 可视化的交易流程：针对新用户第一次使用时给出指引页面，通过图形化界面展示交易流程，帮助用户轻松理解每一步操作。

5. 结束语

从业多年，见过太多的线上故障，得到一个朴素的道理：“人都是不可靠的，如果能通过系统解决的，就一定不要依赖人或流程来解决。”多引入一些防呆设计，让再“呆笨”的人都没有出错的可能性，那么系统就是健壮的，也就没有那么多的线上应急和复盘。

这是《百图解码支付系统设计与实现》专栏系列文章中的第（28）篇。和墨哥（隐墨星辰）一起深入解码支付系统的方方面面。

欢迎转载。

Github（PDF文档全集，不定时更新）：https://github.com/yinmo-sc/Decoding-Payment-System-Book

公众号：隐墨星辰。

有个小群不定时解答一些问题或知识点，有兴趣的同学可先加微信（yinmo_sc）后进入，添加微信请备注：加支付系统设计与实现讨论群。