



软件开发中的浪费

陶亮 · 2023年9月16日



目录

CONTENTS

1. 顺势而为
2. 典型浪费
3. 消减浪费
4. Q&A





顺势而为



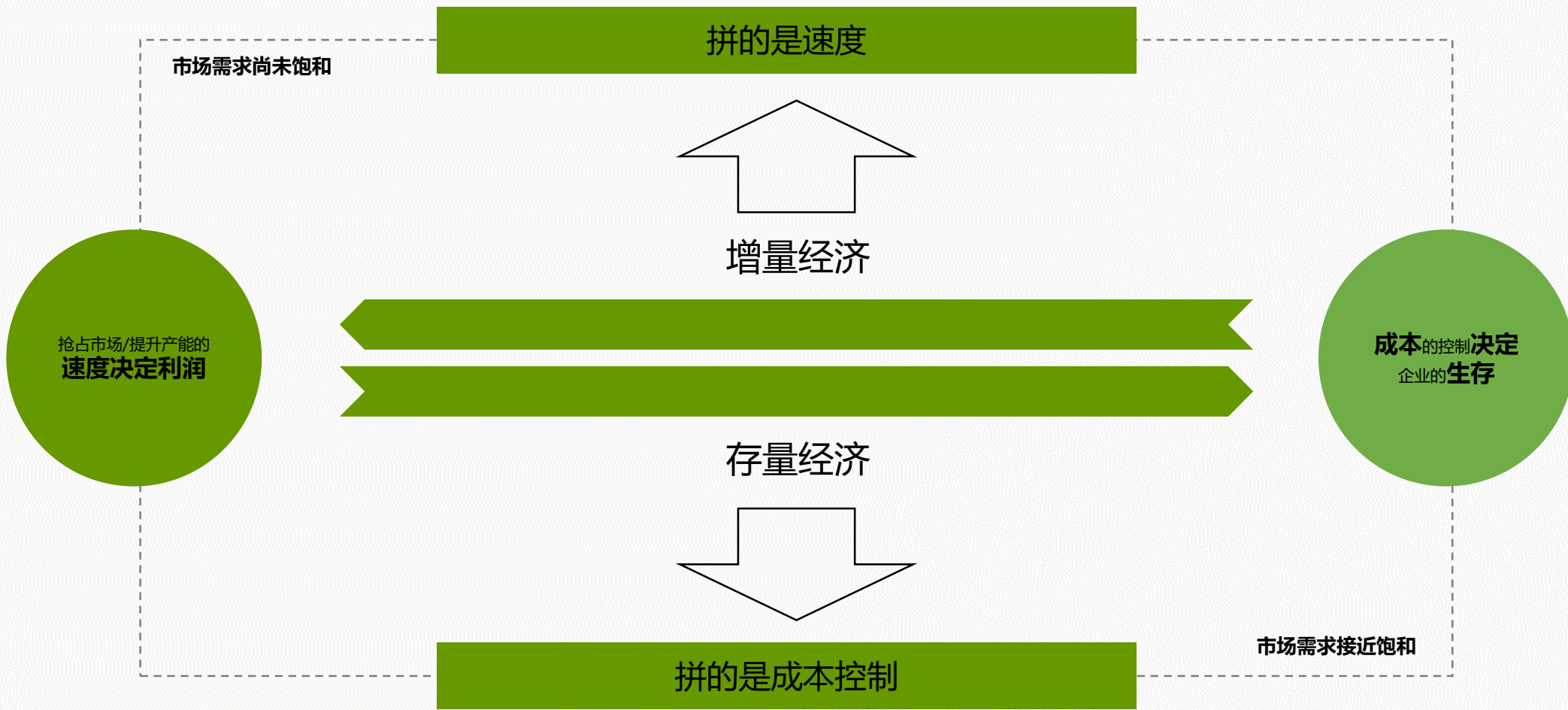
“

今天，存量经济时代已经来临

”



存量经济与增量经济



企业的经营方向



增量经济时代

存量经济时代



企业的经营方向



企业的经营思想



垄断企业



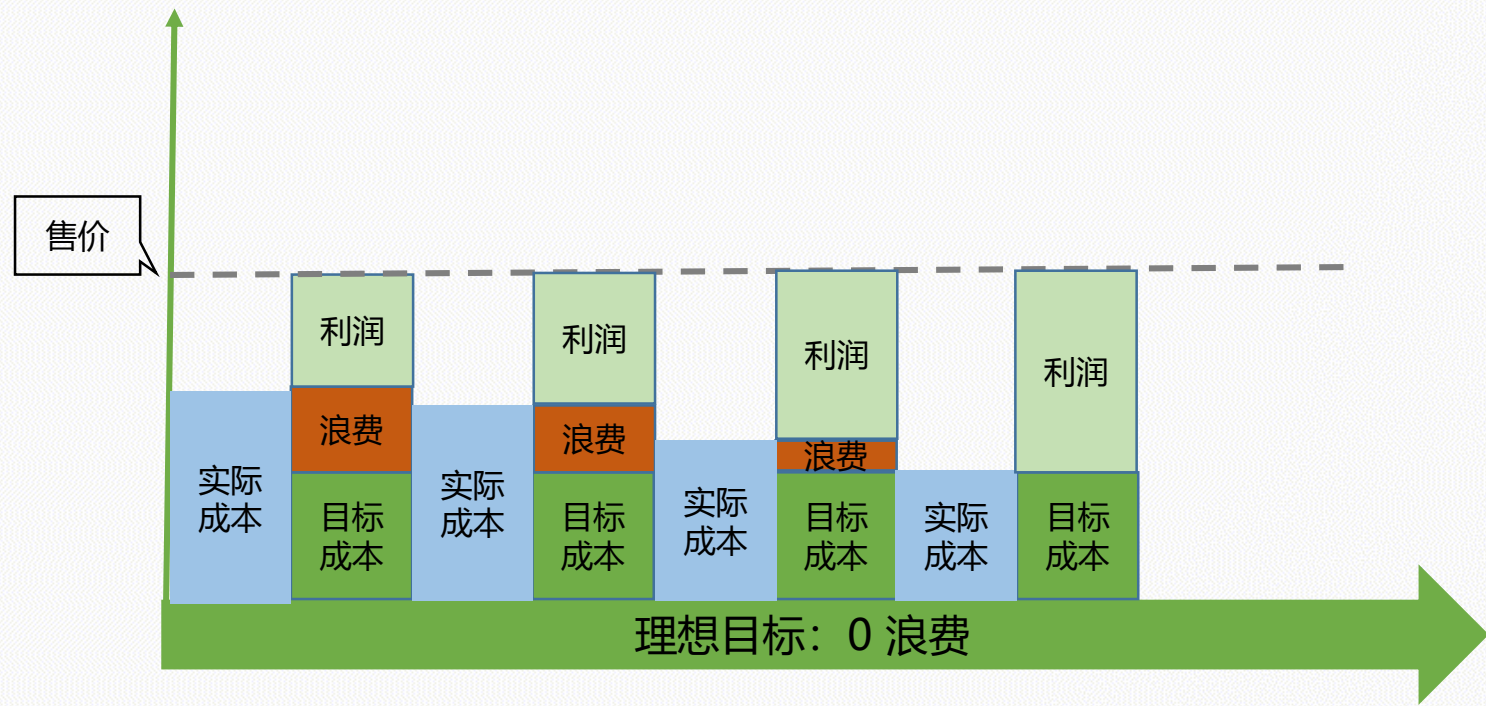
传统管理型企业



精益管理：消除浪费，降低成本



企业的经营思想



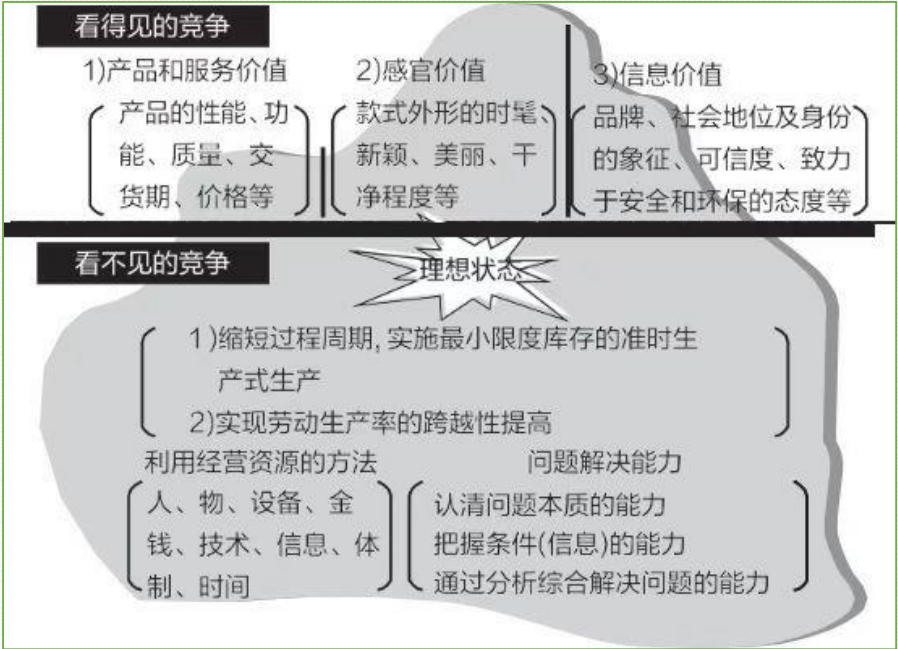
降价只是一个决定，降成本必须是一个能力

这才是竞争的高级模式

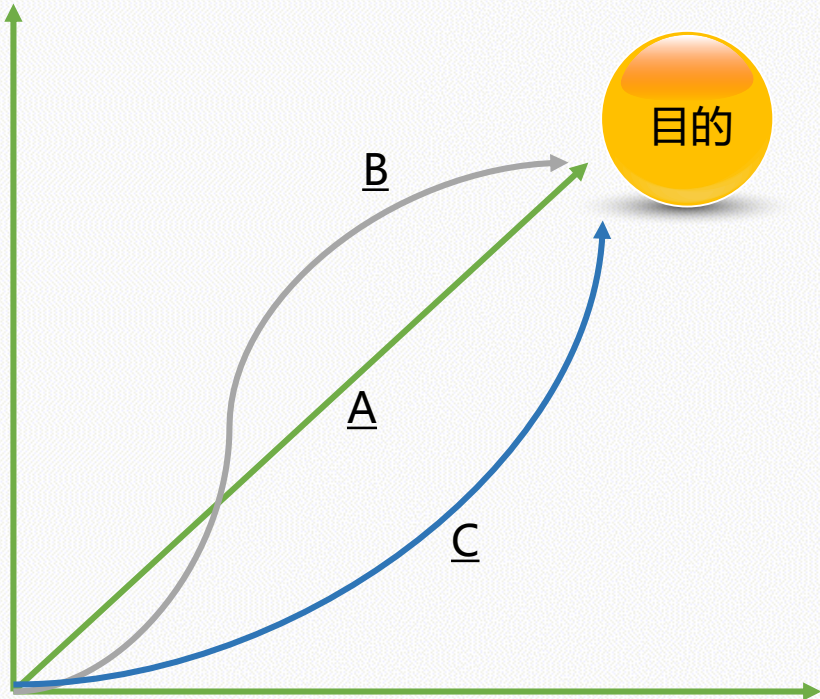
赚一块，不如省一块，省则赚，省一块等于赚三块



消除浪费，降低成本，提升竞争力



引用：竞争的冰山示意图



用最简单&合理&优化的方法，达到同样的目的

降本的能力（看不见的竞争），看得见的竞争的创造能力：
以**消除浪费**为基本手段，提高效率，缩短周期时间，最终实现节流的目的





典型浪费



浪费的定义

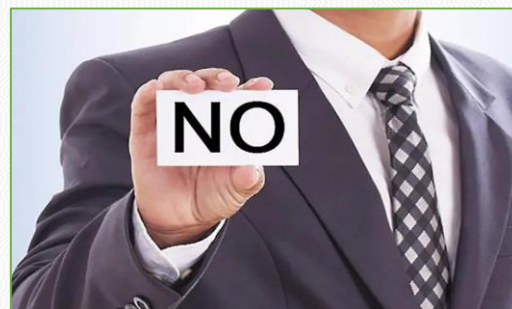
广义的
浪费



狭义的
浪费



是否增值?

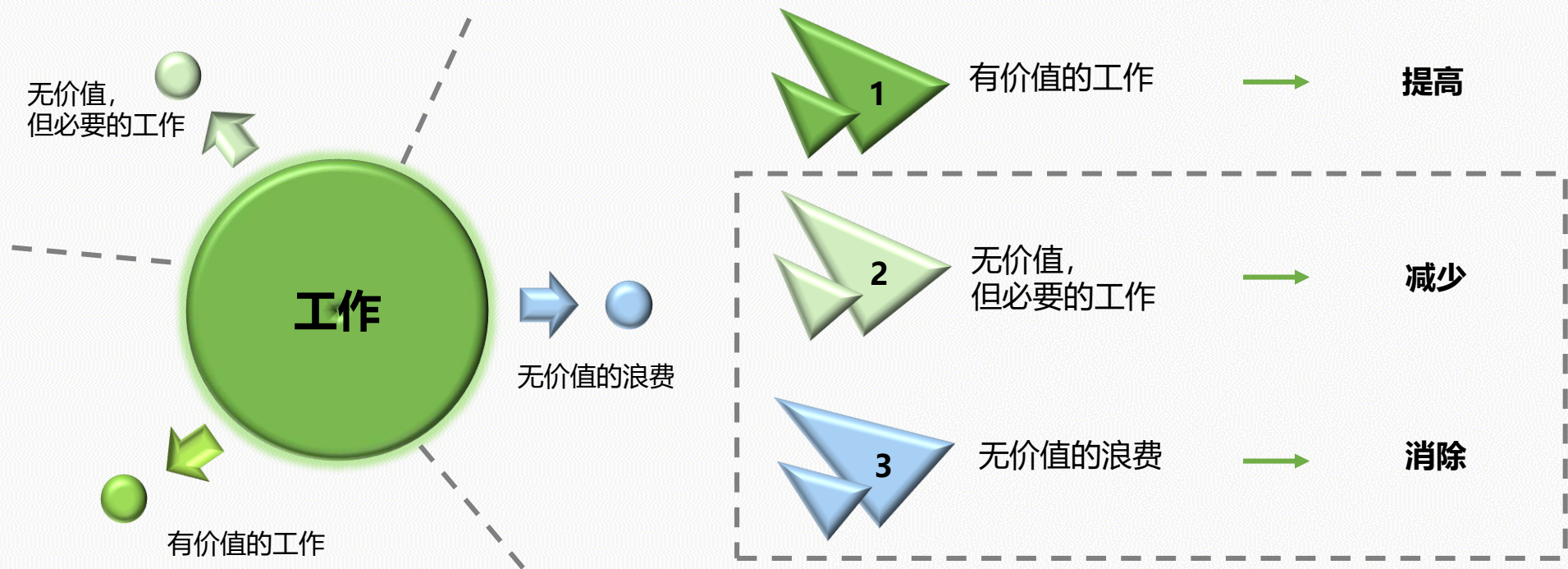


客户是否愿意付钱?



浪费的分类

一切不创造价值的活动，换言之：不能够提高附加价值的部分



精益生产中的浪费



等待浪费：
因材料或作业的等待
或者闲暇时间而产生



过度加工浪费：
当前作业是否真的有必要
为什么要这么做



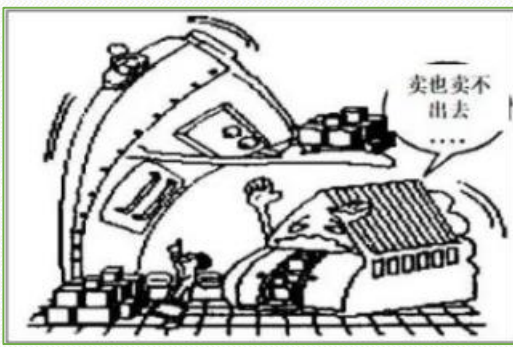
动作浪费：
多余的手、脚、身体的动
作，不能产生价值的



库存浪费：
库存带来资金和场地浪费



不良浪费：
材料不良，加工不良
检查不良 等



过量生产浪费：
生产量超过市场需求量而
产生



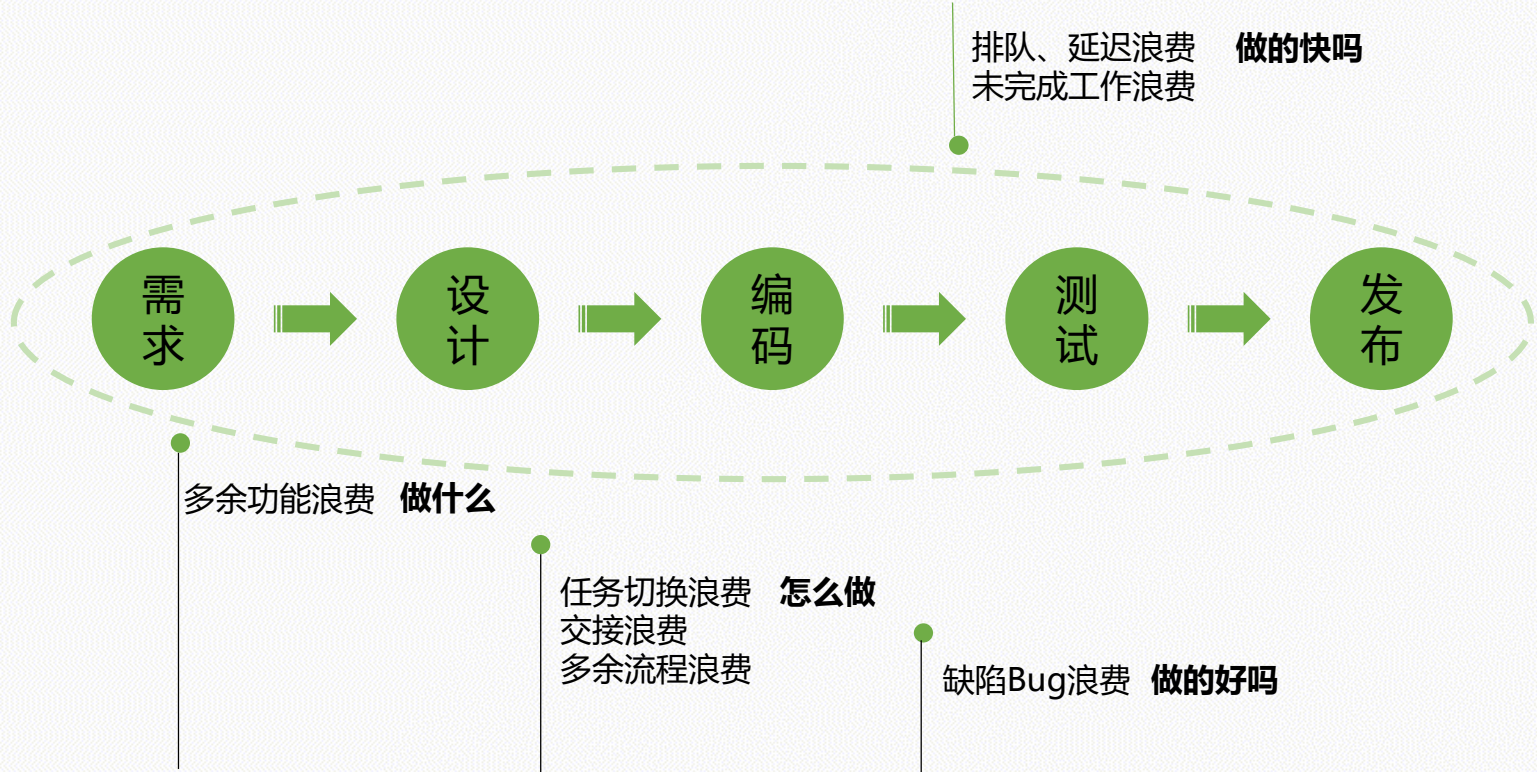
搬运浪费：
为储存、堆积在临时仓库
而进行的搬运所产生

生产过程中的
七大浪费



软件开发中的浪费

精益生产	软件开发
等待浪费	排队、延迟的浪费
过度加工浪费	多余流程的浪费
动作浪费	任务切换的浪费
库存浪费	未完成工作的浪费
不良浪费	缺陷Bug的浪费
过量生产浪费	多余功能的浪费
搬运浪费	交接的浪费



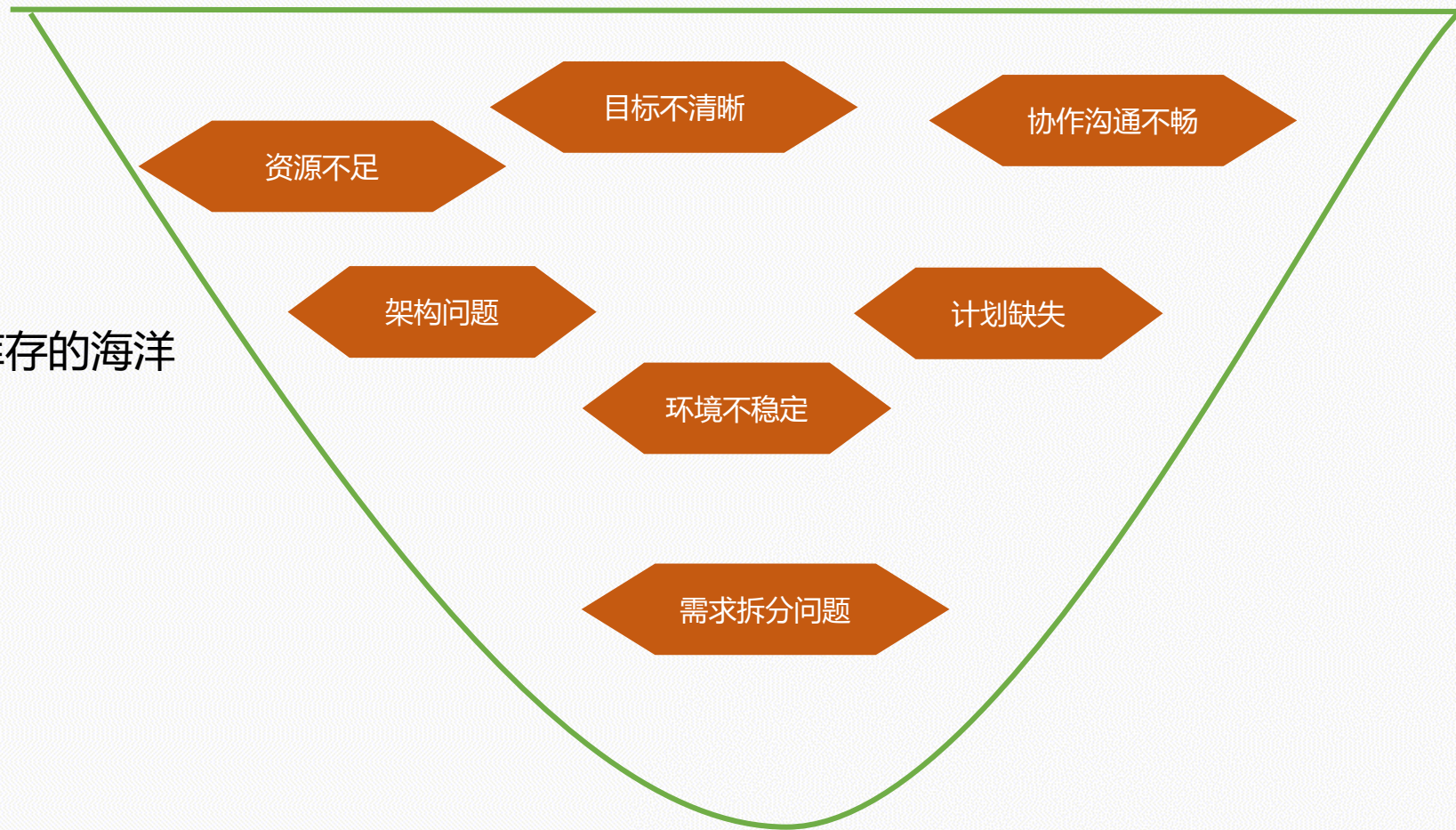
库存浪费



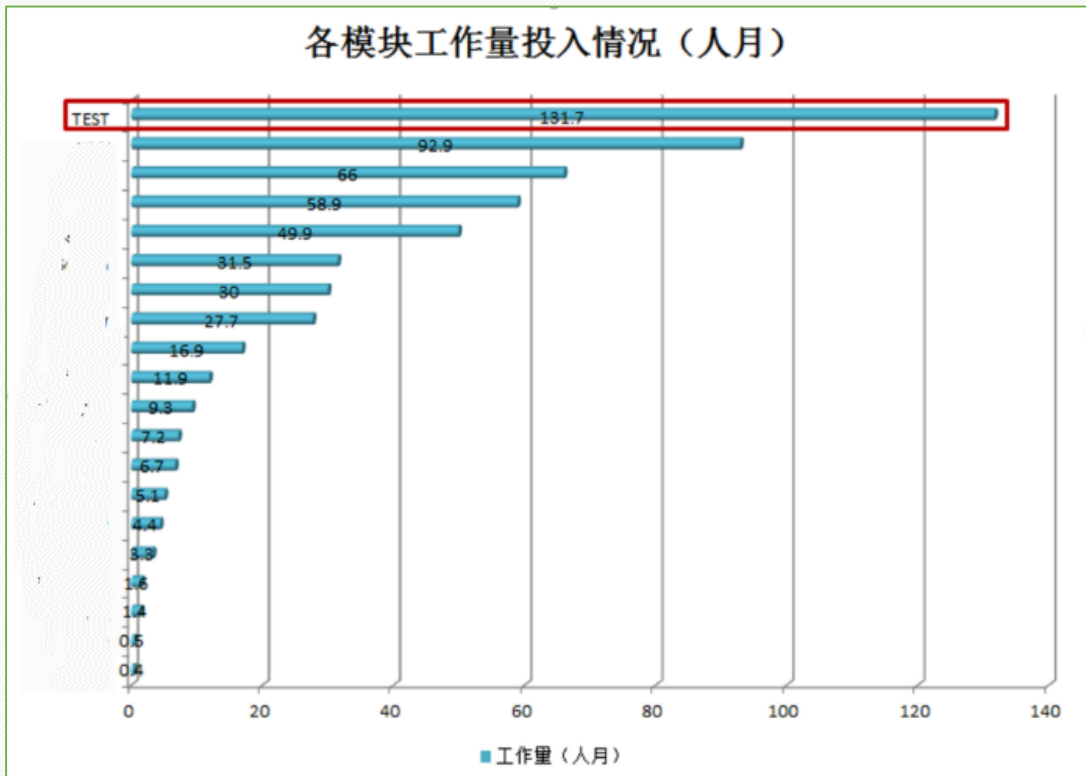
库存浪费

海面

库存的海洋



缺陷浪费

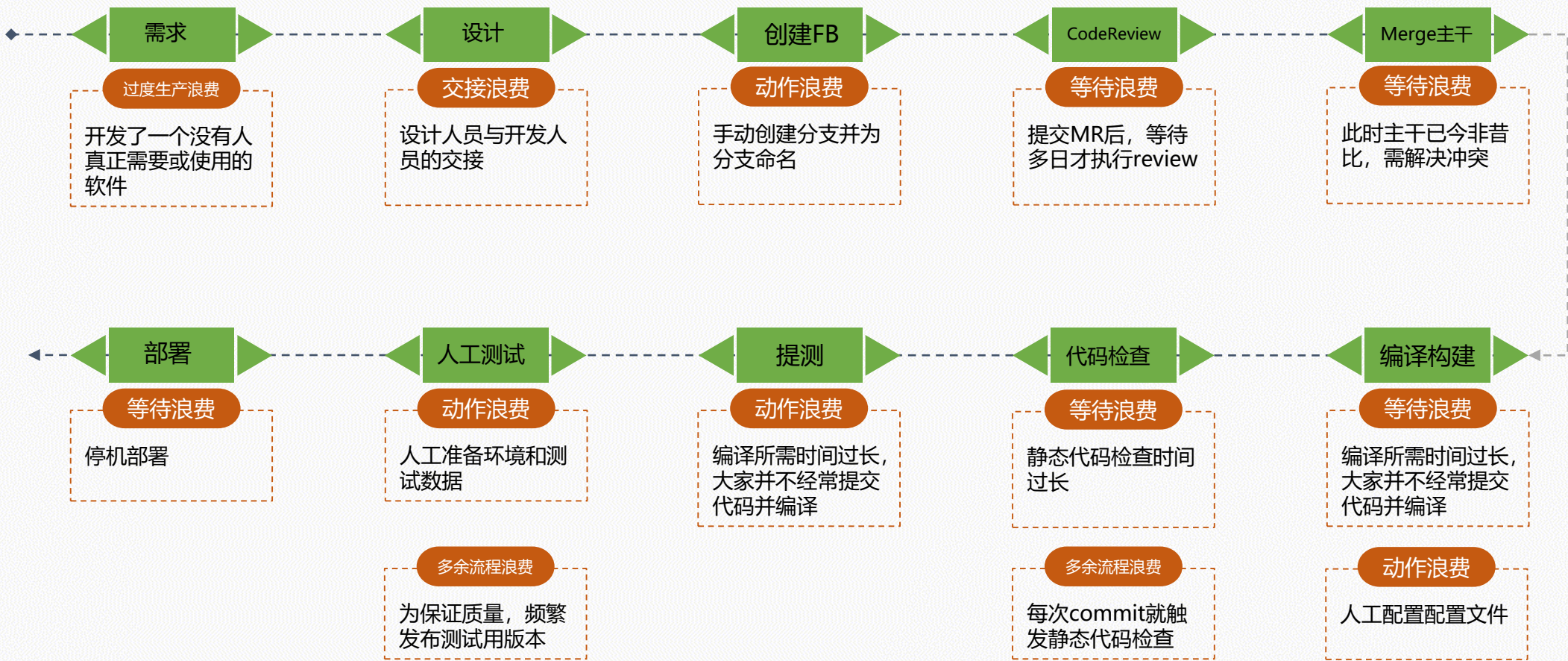


组别	Bug标签下工作量 (人月)	各组总工作量	各组COPQ
开发	17.7	27.7	64.0%
需求	4.3	7.2	59.9%
测试	3.8	6.7	57.3%
部署	16.6	31.5	52.8%
运维	2.3	4.4	52.1%
设计	26.9	66.0	40.7%
架构	28.5	92.9	30.7%
接口	17.7	58.9	30.1%
数据	5.0	16.9	29.7%
安全	7.2	30.0	23.9%
性能	0.7	11.9	6.2%
兼容	4.6	131.7	3.5%
文档	0.1	49.9	0.2%

- 项目整体投入工作量557.3人月，其中TEST投入131.7人月，约占整体1/4
- Bug调查修改用总工作量约 135.4 人月，约占整体1/4
- 上述两条合计，占项目总工作量的 1/2



等待、动作、交接、过度加工浪费





消减浪费



Devops 三步工作法



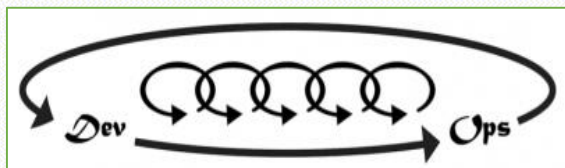
加速流动
可视化、限制在制品、小批量

实现开发到运维的工作快速的从左向右移动



建立反馈
建立度量、安全部署、AB测试

越早反馈，越早发现并解决问题，成本越低

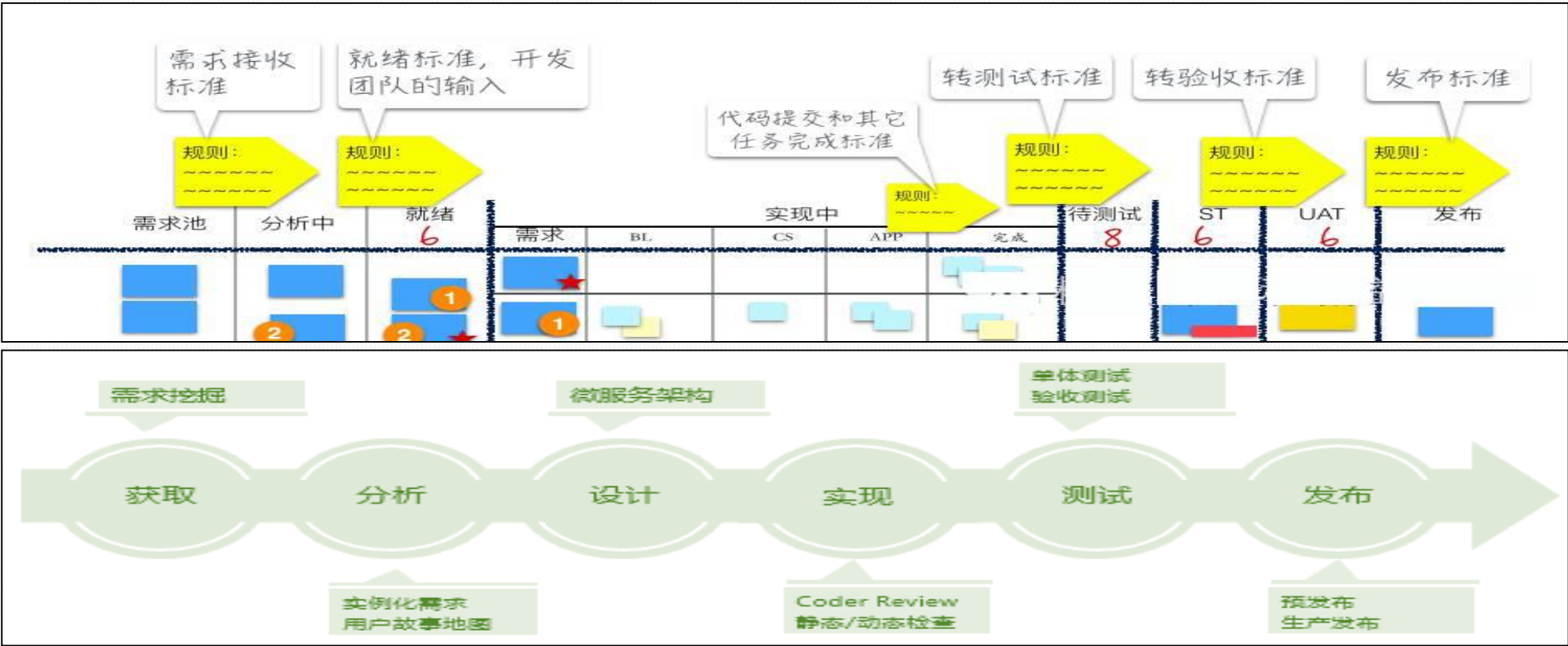


持续实验和学习
根因分析、经验共享、学习型组织

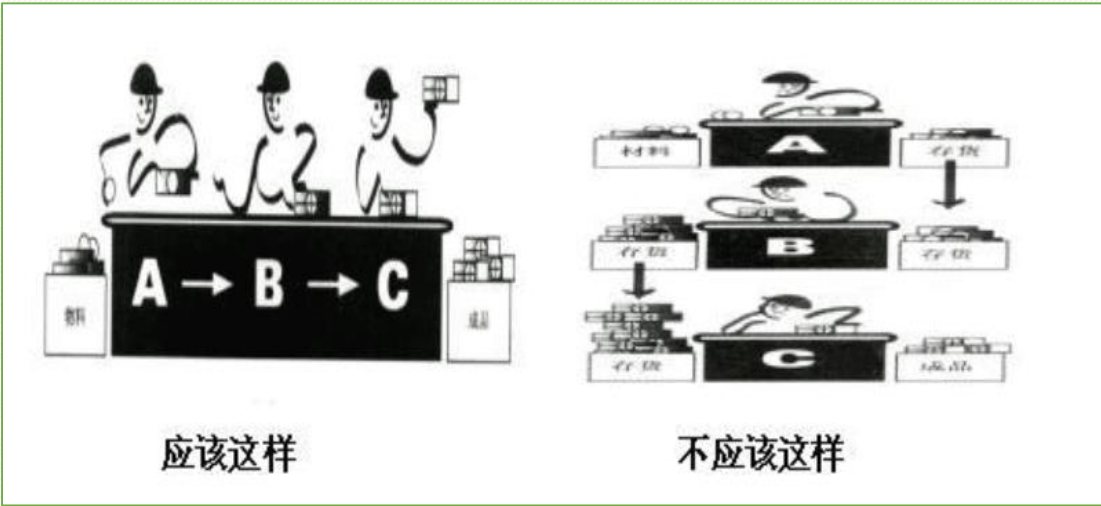
建立具有创意和高可信度的文化



使工作可视化



减小批量，聚焦完成，暂缓开始



大批量策略

- 1、先折10张纸
- 2、统一装入信封
- 3、统一封口
- 4、统一贴邮票

完成第一封信：310s

小批量策略

第一封信做完4个步骤
再开始做第二封信
以此类推

完成第一封信：40s

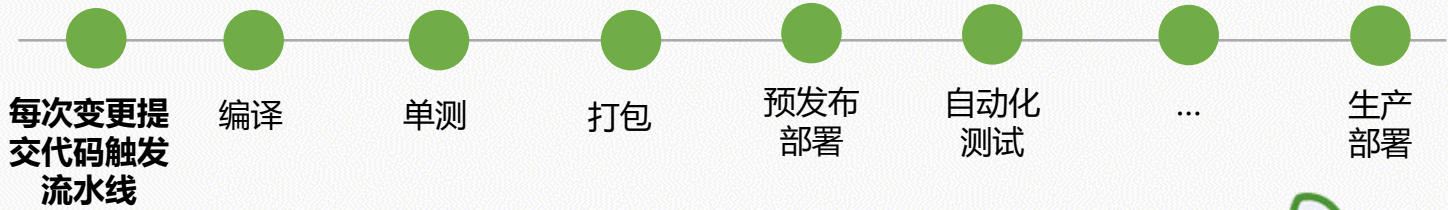
VS



小批量开发的好处

- 更少的 WIP
- 快速发现错误
- 更少的等待
- 更少的返工

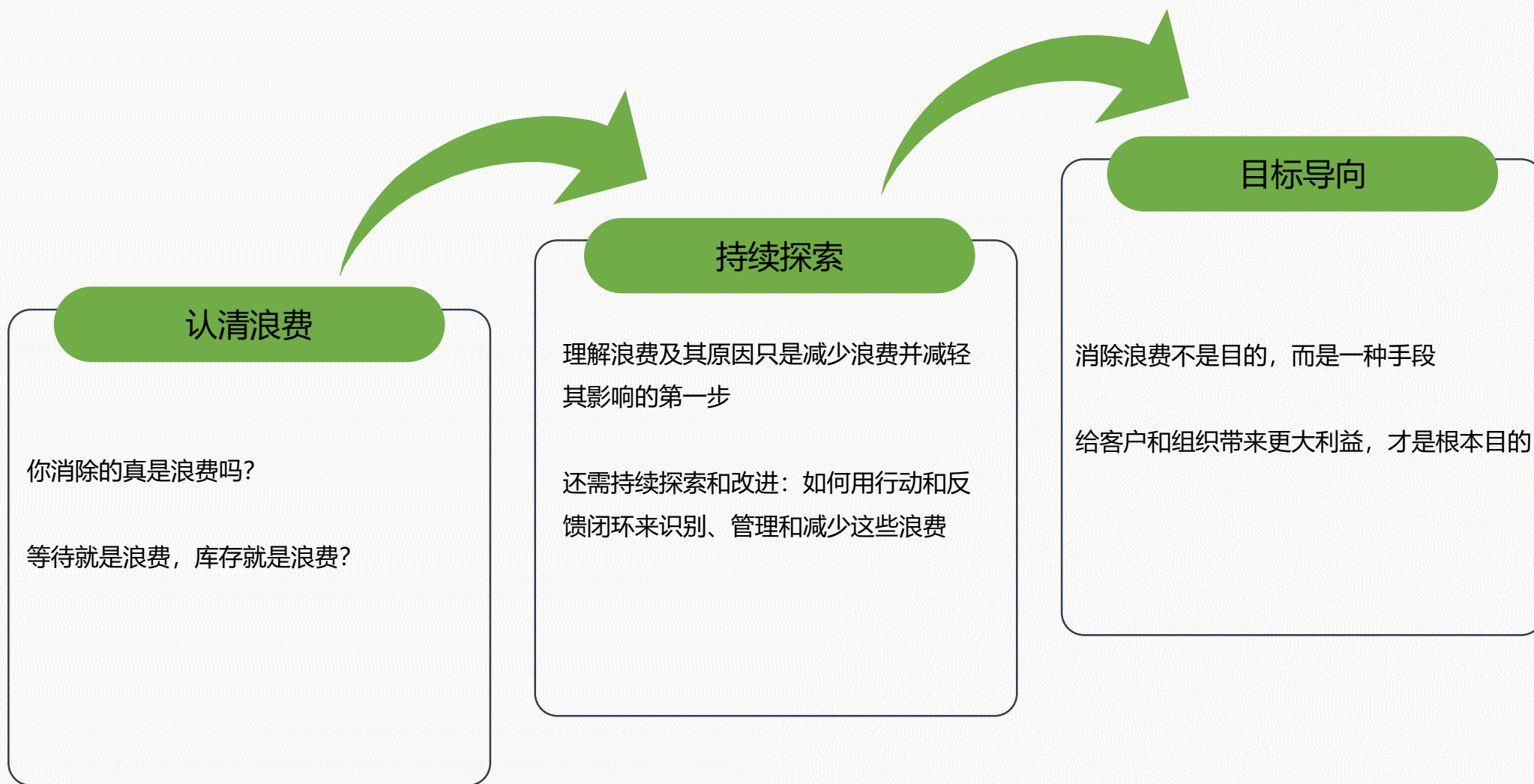
单件连续流 – 软件



建立快速反馈



结束语





提问时间

谢谢大家





关注社区公众号
了解更多活动

