中国DevOps社区

敏捷&团队协同工作

2023年04月

自我介绍

马建

众安保险 敏捷教练专家

DevOps交付负责人

- SPC 6.0
- 大规模敏捷LeSS
- CSM
- CSP
- CSPO
- PRINCE2

相关经验

- 作为敏捷技术卓越的长期实践者与倡导者，致力于推广有效敏捷工具、思想与敏捷实践；
- 专注于团队发展和敏捷实践转型，擅长敏捷思想与教练技术的融合，激发团队内在潜能，从而提升产品开发的效果与质量，助力企业与组织发展。

AGENDA

01 敏捷介绍

02 众安敏捷实践：促进团队协同工作

03 团队回顾，有效改善团队效能

04 过程度量

敏　　捷　　介　　绍

敏捷宣言

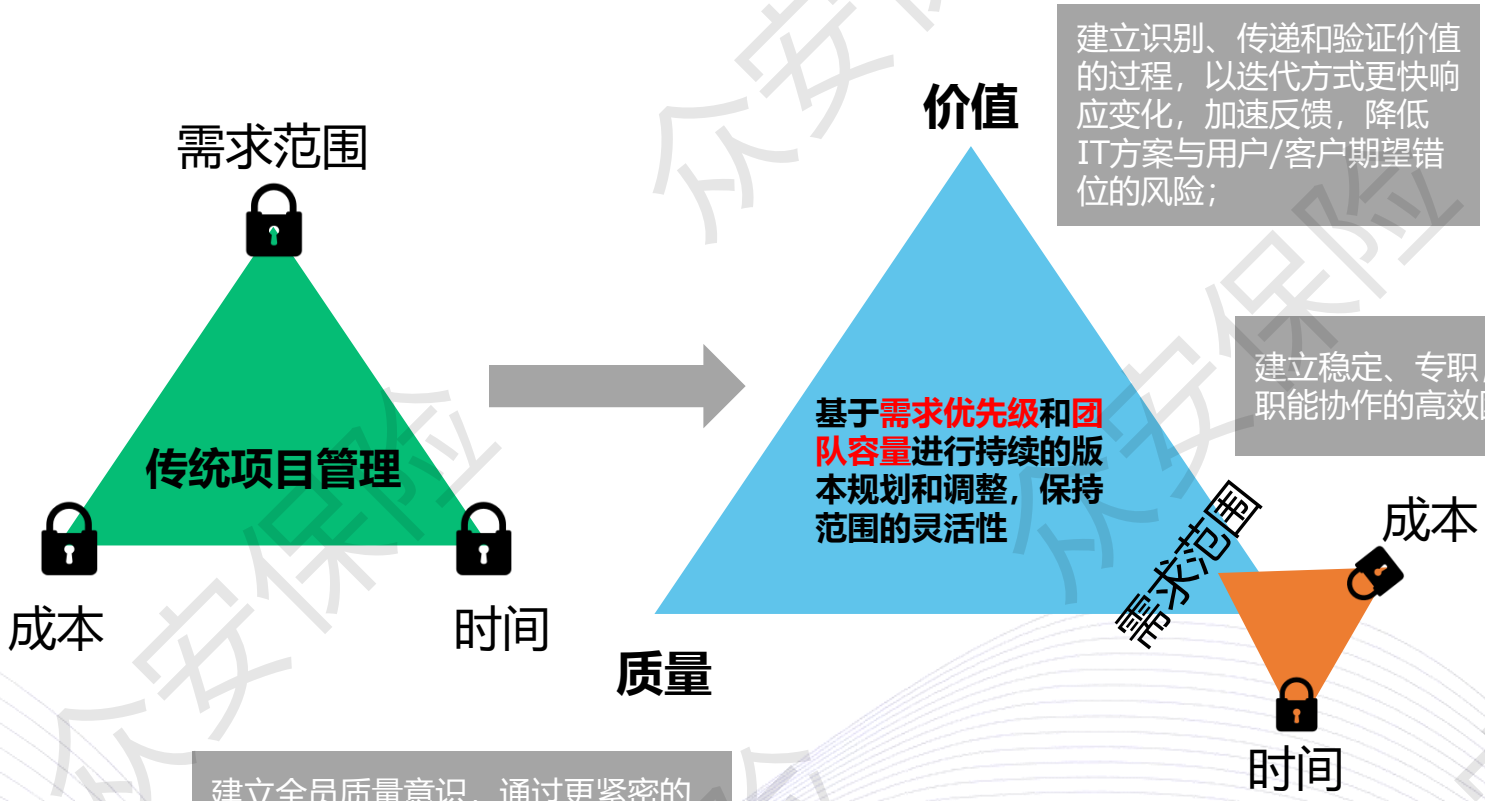
我们一直在实践中探寻更好的软件开发方法，身体力行的同时也帮助他人。由此我们建立了如下价值观：

连接和打通组织的各个职能，以确保协调一致的行动	→	个体与交互	高于	过程和工具
聚焦端到端的价值流动，以快速、灵活地交付真实的客户价值	→	可用的软件	高于	完备的文档
与客户建立共同目标，以最大化业务成果	→	客户协作	高于	合同谈判
有计划和系统地主动试错，以支持有效的学习和创新	→	响应变化	高于	遵循计划

在每组对比中，右项并非全无价值但我们更看重左项价值。

敏捷管理思想的精髓——来自敏捷宣言签署者

通过**适应性计划**（Adaptive Planning），将价值和质量显性出来，放到所有管理活动中心。



- 用具有**弹性的范围**换来变化的及时响应
- 可协商的范围让团队真正有可能将注意力放到价值和质量上
- 用愿景、目标的统一来驱动整个开发过程向业务价值对齐，并带来自主性！

众安敏捷实践：促进团队协同工作

研发团队敏捷短迭代开发框架

3

3个角色

产品负责人（PO）
Scrum Master/敏捷教练
跨职能团队

3

3个工件

产品代办事项（PB）
Sprint迭代代办清单
潜在可发产品增量

5

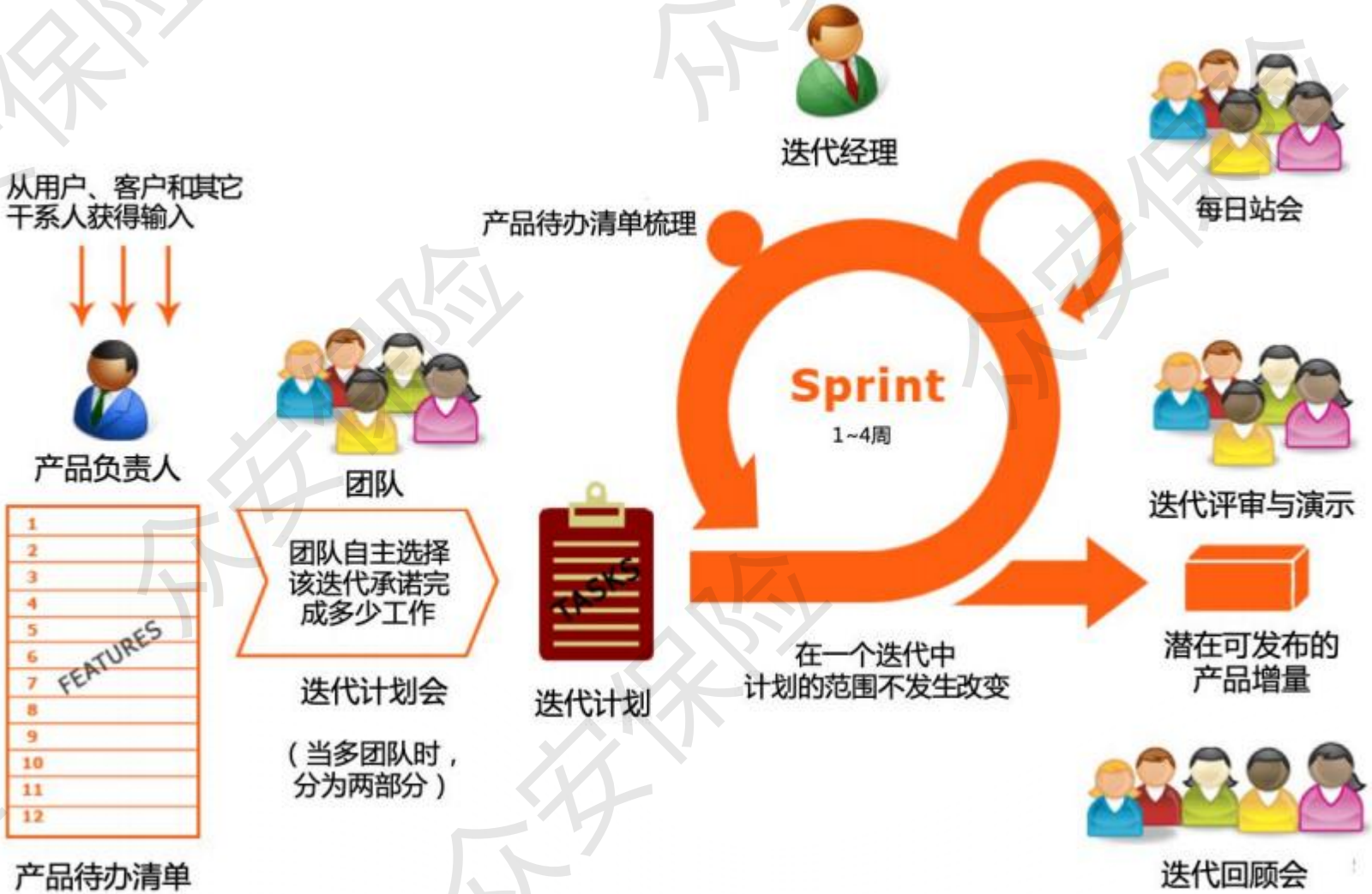
5个活动

冲刺（Sprint）
Sprint 计划会
每日Scrum站会
Sprint 评审会
Sprint 回顾会

5

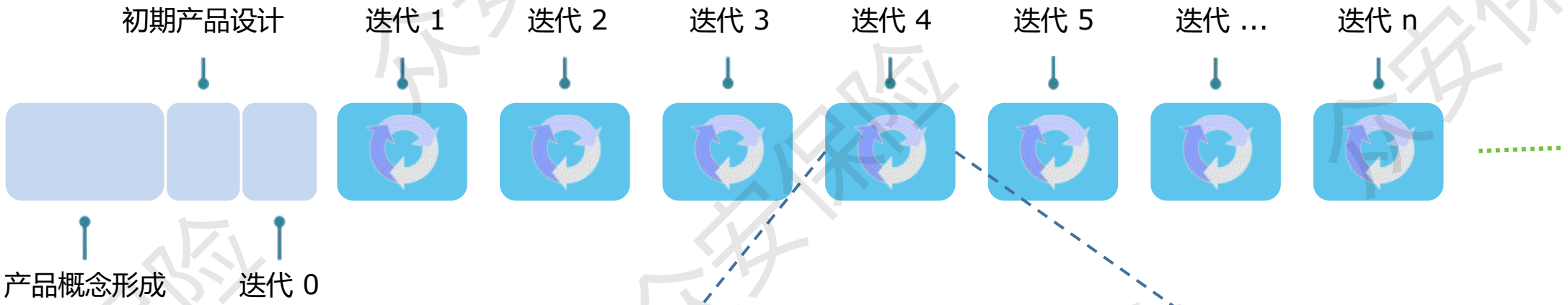
5个价值观

开放
承诺
专注
尊重
勇气



围绕产品迭代式开发的基本流程

项目的周期很短，而产品的生命周期很长，通过一个个迭代长期持续演进。



迭代开始

方案设计 | 故事准备 | 详细设计 | 开发 | 测试

通过建立一个稳定、均衡的生产节拍建立一个可预测式生产过程，增进横向纵向协作。

部门跨团队的敏捷实践落地

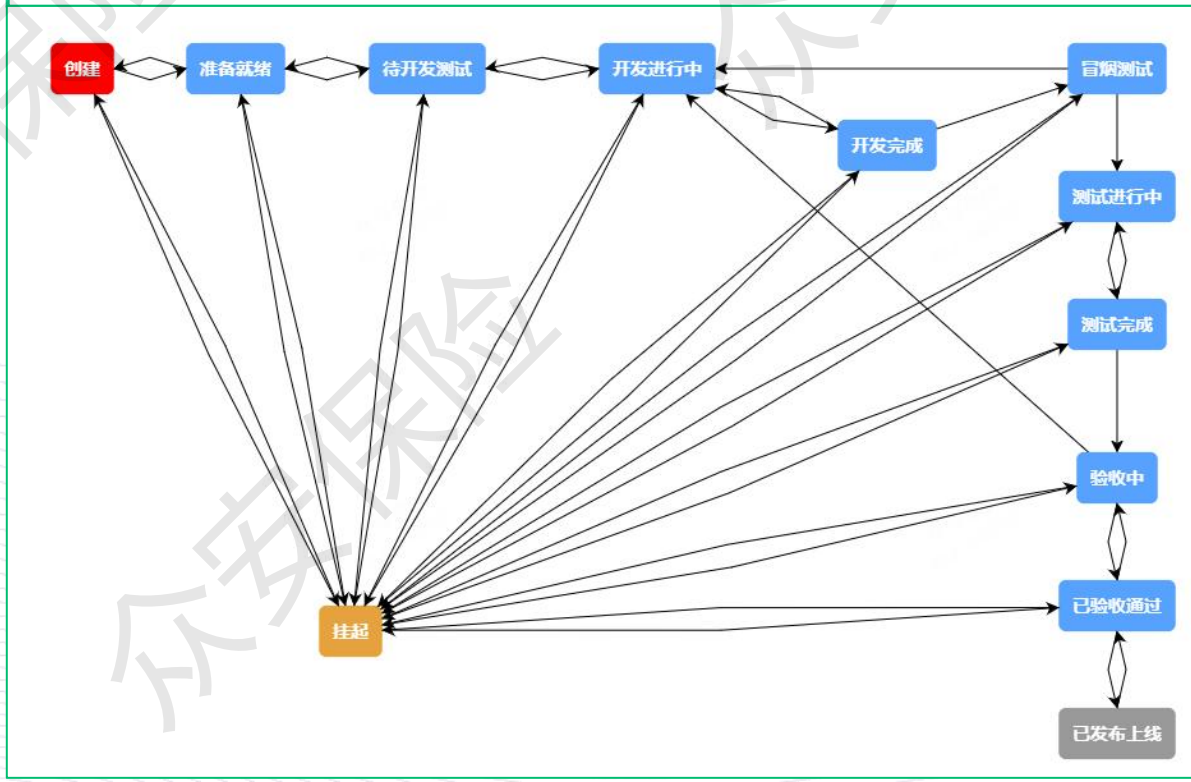
背景：TM、magic、ship、六翼、绿洲团队合并；

问题：各团队管理流程不统一、DevCube使用不规范、缺少统一的度量指标等；

解题思路：统一活动标准、统一团队管理机制；

<table>
<tr><th></th><th></th><th>周一</th><th>周二</th><th>周三</th><th>周四</th><th>周五</th></tr>
<tr><td rowspan="2">第一周</td><td>上午</td><td>Sprint计划会</td><td colspan="4">每日站会(15分钟)</td></tr>
<tr><td>下午</td><td colspan="4">开发测试</td></tr>
<tr><td rowspan="3">第二周</td><td>上午</td><td colspan="4">每日站会(15分钟)</td></tr>
<tr><td rowspan="2">下午</td><td rowspan="2">开发测试</td><td>需求梳理会（1h）</td><td rowspan="2">开发测试</td><td>开发测试</td></tr>
<tr><td>开发测试</td><td>回顾会(1.5h)</td></tr>
</table>

统一活动标准，统一团队管理机制



需求梳理会（下个迭代）

中国DevOps社区

需求梳理会

目的	1. 提前给团队成员讲解故事，讨论和澄清需求与设计疑问，预留时间解决问题，让正式的需求KT会议更高效、合理 2. 让团队成员了解产品全貌，而非仅关注自己的工作，促进成员间协作 3. 对下个迭代的需求进行需求细节梳理和精化，识别技术风险和依赖，完成估算和优先级排序
时间	迭代第二周的周三（正式计划前，预留几天解决会上解答不清的问题）
参与者	产品经理、SM（或者项目经理）、Team核心成员
活动	1. 产品经理依次讲解下迭代可能期望交付的故事，挨个故事进行2、3活动 2. 团队思考故事要求和设计，提出疑问，产品经理或其他成员澄清疑问（可能包括对技术方案的疑问），直到没有显著影响工作量的问题 3. 确认用户故事的优先级排序
输出	1. 更新的产品待办清单，调整的故事 2. 未解决的需求故事、方案问题，产品经理记录会后解决 3. 澄清后的用户故事的细节信息（原型图和交互流） 4. 用户故事验收标准 5. 用户故事优先级

		周一	周二	周三	周四	周五
第一周	上午	Sprint计划会	每日Scrum站会	每日Scrum站会	每日Scrum站会	每日Scrum站会
	下午	开发测试				
第二周	上午	每日Scrum站会	每日Scrum站会	每日Scrum站会	每日Scrum站会	每日Scrum站会
	下午	开发测试		**需求梳理会(1h)**	code review(1.5h)	Sprint评审会(1h) Sprint回顾会(1.5h)

需求梳理会（下个迭代）—注意事项

会前准备

- ◆产品设计共识（设计未确定的需求不应进入需求梳理会和迭代计划）
- ◆技术概要设计评审通过
- ◆拆分与书写故事，包括验收标准
- ◆产品经理 、SM、核心技术骨干（2～3 人）会前小范围沟通，解决掉显著问题

会中

- ◆ 全员参与，协作
- ◆ 产品经理记录待确认问题
- ◆ SM引导，注意会议效率，对10分钟讨论不清的问题，记下来会后再解决
- ◆ 必要时拆分和新增故事

会后

- ◆正式Sprint 计划会前，确保解决所有待确认问题
- ◆可能安排多次需求梳理会

Sprint计划会

目的	1. 团队明确一个短周期内的承诺的交付范围，符合团队的实际容量 2. 所有工作有明确的责任人
时间	迭代第1天上午（迭代所有工作的正式开端）
参与者	产品经理、SM（或者项目经理）、Team全体成员
活动	1. 产品经理针对需求梳理会准备好的需求进行逐个澄清； 2. 团队针对所有story有疑问的点进行询问，产品经理进行答疑； 3. 团队明确自己本迭代的可用容量大小 4. 全体成员对story进行估算； 5. 团队成员从确定的迭代计划中认领自己负责的工作项
输出	1. 明确的迭代计划（故事列表、优先级） 2. 每个故事的负责人，及合作参与者

		周一	周二	周三	周四	周五
第一周	上午	Sprint计划会	每日Scrum站会	每日Scrum站会	每日Scrum站会	每日Scrum站会
	下午	开发测试				
第二周	上午	每日Scrum站会	每日Scrum站会	每日Scrum站会	每日Scrum站会	每日Scrum站会
	下午	开发测试		需求梳理会(1h)	code review(1.5h)	Sprint评审会(1h) Sprint回顾会(1.5h)

Sprint计划会—注意事项

会前准备

- 故事拆分、定义合理清晰
- 拆分与书写故事，包括验收标准
- 产品经理 、SM、核心技术骨干（2～3 人）会前小范围沟通，解决掉显著问题，风险和外部依赖分析，且可控

会中

- 明确优先级顺序
- 承诺目标和挑战目标
- 清楚团队容量，前期基于计算，几个迭代后参考历史速率，同事考虑以下因素降低可用容量：
  - 期间的年假安排
  - 个别人休假计划
  - 个别人是否兼职工作
  - 生产问题支持的投入预留

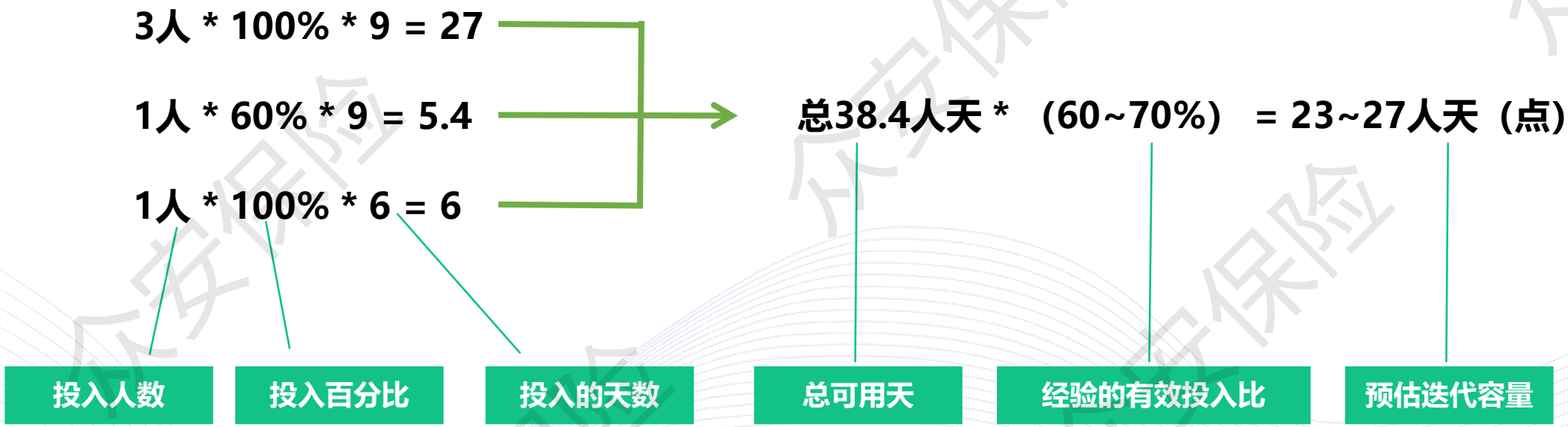
会后

- 重视承诺，努力达成承诺！
- 故事卡片要求：
  - 故事卡和验收标准定义清楚
  - 测试与开发就完整的测试场景达成共识
  - 开发评估给出计划转测日期，记录在卡片上
  - 不要让所有故事都堆在迭代最后几天转测；
- 迭代所有信息上看板

新团队不知道迭代排多少故事合适?

要点:

1. 集体估算，按团队平均能力所需要1人天作为一个故事点;
2. 按以下工时计算粗略迭代容量（人数包括开发与测试）：



迭代容量可视化

Sprint计划会前，确定人员投入情况，根据迭代容量，规划符合团队的故事点数；

如果迭代成员无变化，可以从其他迭代直接导入成员

迭代成员　投入总人数:10　总可用天数:64.50

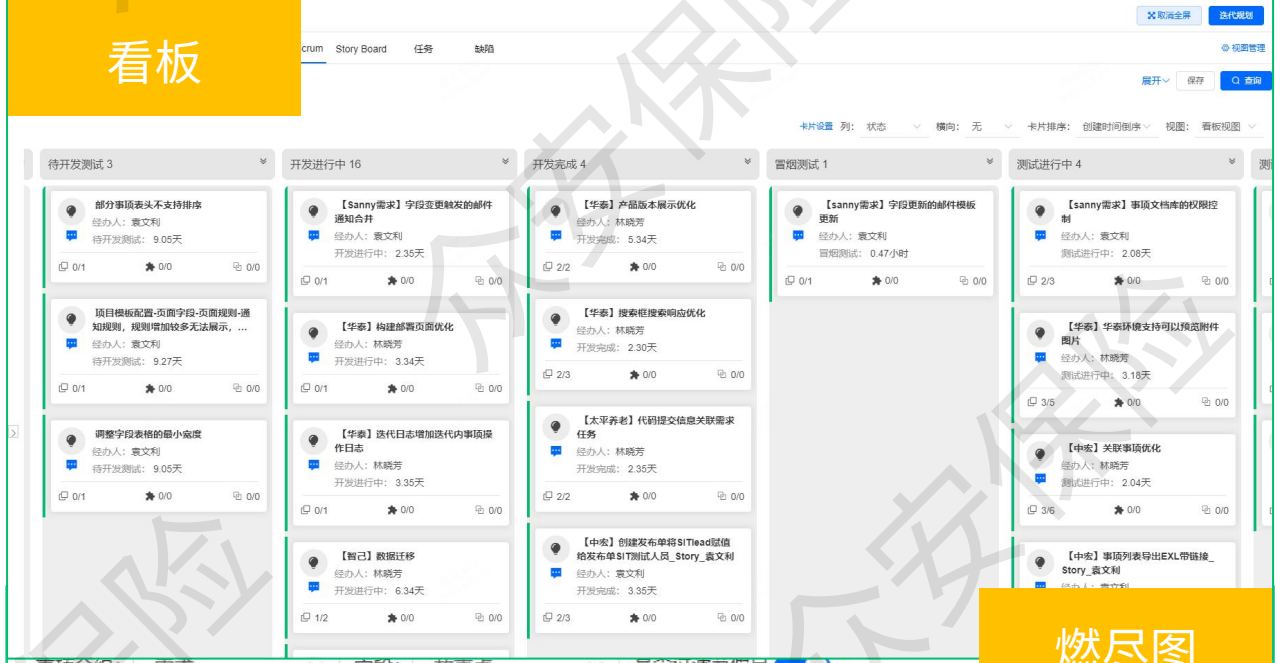
添加成员　导入成员

姓名	占用人时比(人时/人时)	角色	投入百分比	工作的天数	可用人天数	操作
[redacted]	0.0/64.00	[redacted]人员	100%	8	8.00	删除
[redacted]	0.0/24.00	[redacted]管理员	50%	6	3.00	删除
[redacted]	0.0/64.00	[redacted]-前端研发	100%	8	8.00	删除
[redacted]	0.0/24.00	[redacted]-后端研发	50%	6	3.00	删除
[redacted]	0.0/56.00	[redacted]人员	100%	7	7.00	删除
[redacted]	0.0/72.00	[redacted]后端研发	100%	9	9.00	删除
[redacted]	0.0/56.00	[redacted]前端研发	100%	7	7.00	删除
[redacted]	0.0/48.00	[redacted]后端研发	100%	6	6.00	删除
[redacted]	0.0/72.00	[redacted]-后端研发	100%	9	9.00	删除
[redacted]	0.0/36.00	[redacted]-后端研发	50%	9	4.50	删除

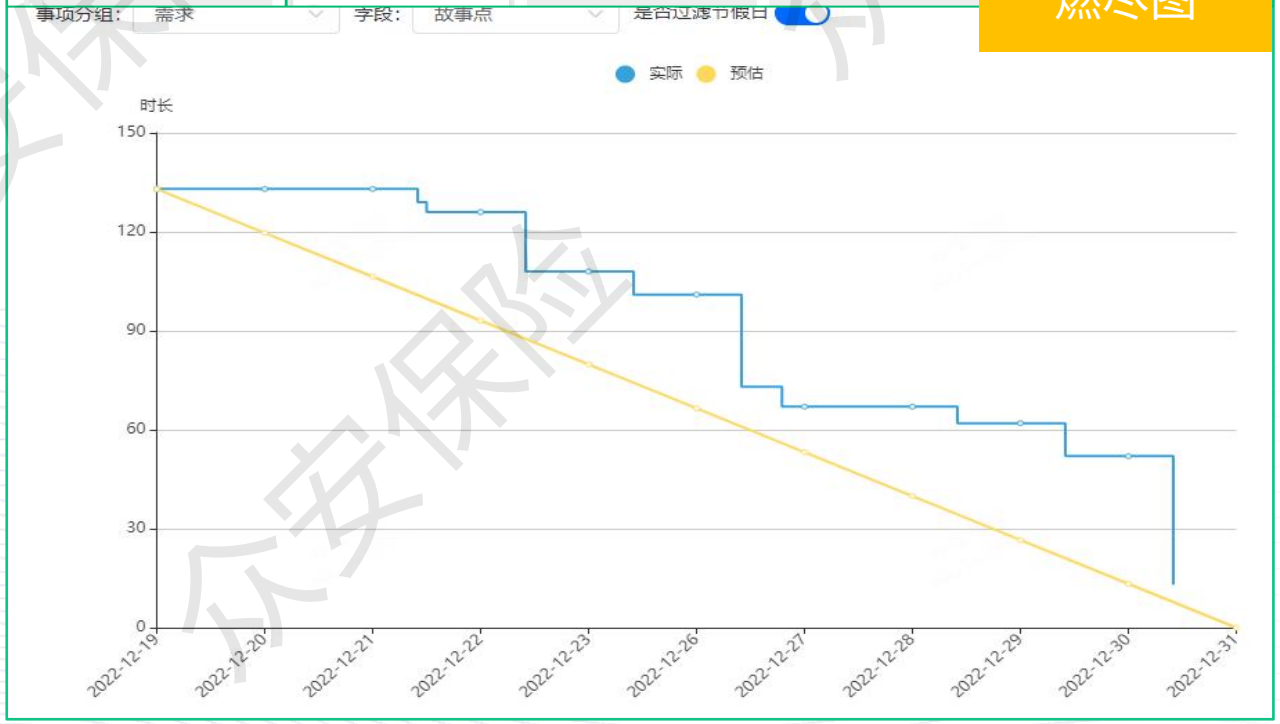
三个举措落实团队管理机制

站会

看板



燃尽图



每日Scrum站会

目的	1. 团队成员沟通工作进展、 问题，促进相互间信息透明、协作
时间	迭代中每天固定时间、固定地点（不超过15分钟）
参与者	PO 、SM 、Team全员参加， 由SM主持
活动	每个团队成员结合看板上自己的工作项，回答三个问题： 1. 我昨天有什么进展？ 2. 我今天计划做什么？ 3. 我遇到了什么阻碍和问题？ 对于看板上标注的需重点关注的工作项，如存在风险、依赖或已进度延迟的，重点更新其状态；SM记录所有人提出的阻碍和问题
输出	1. 更新的看板 2. 团队遇到的阻碍记录

典型问题—如何让站会更高效?

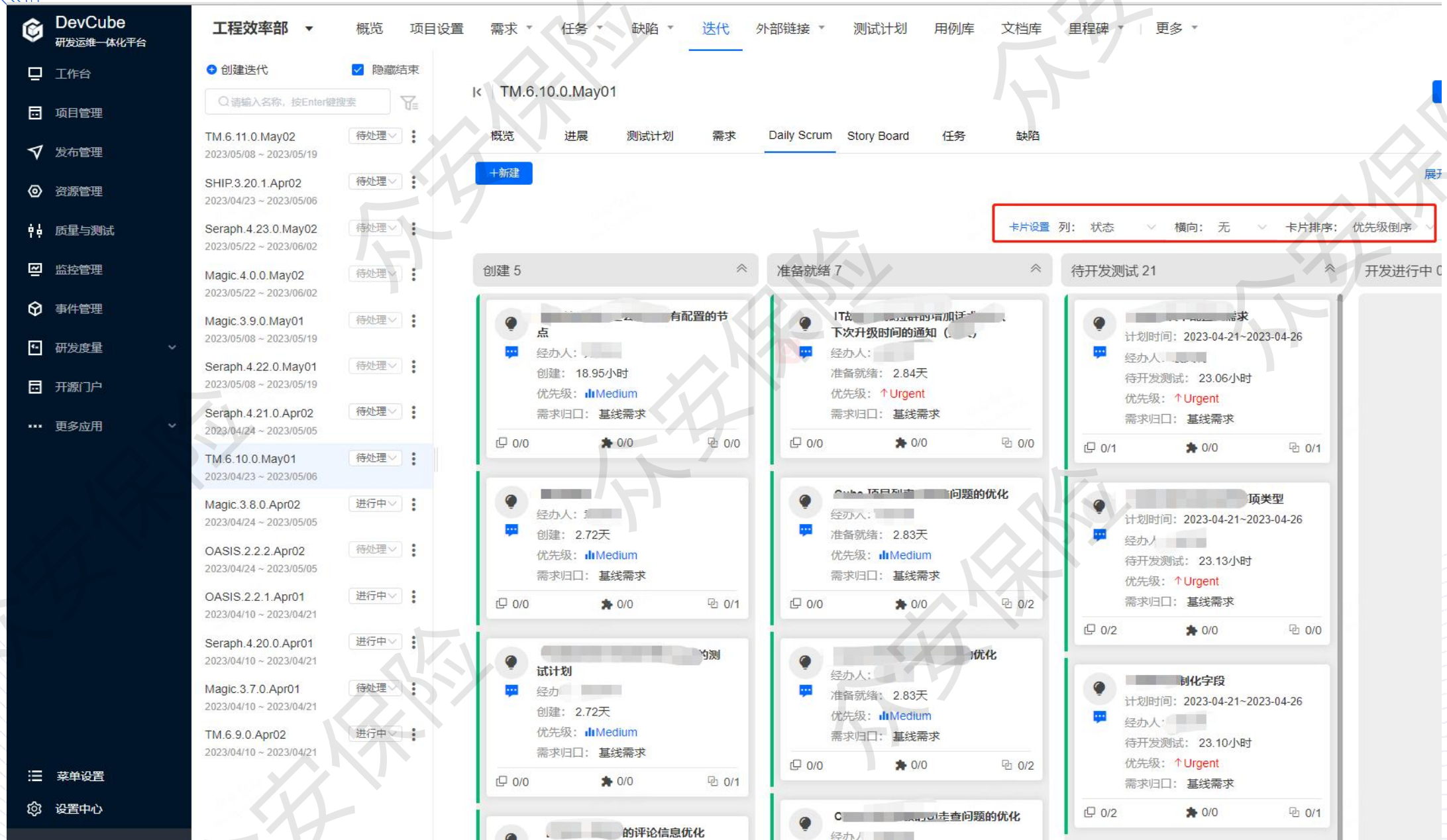
站会低效，浪费时间?

1. 站会的人数不能太多，建议不能超过10人（人 数太多必须拆分敏捷团队）
2. 会前，每个人应提前准备好要讲什么
3. 专注三个问题，不要花过多时间讨论，SM记下 问题会后再讨论
4. 不要把两组做完全不相关工作的人拉到一起开站会，意味着团队划分有问题

站会起不到信息拉通、识别问题的目的?

1. 不是给产品经理和SM做汇报，而是要对着所有成员讲 （站到看板前）
2. 站成一圈，不能太散，每个人都能看到其他所有人，讲话的声音要能让每个人听到
3. 若成员羞涩，表达不充分，SM要调动每个人讲问 题
4. **站会要充分利用起看板**，每人讲的工作要指着看板上对应的卡片讲，及时更新信息

工欲善其事必先利其器：多维度组合看板

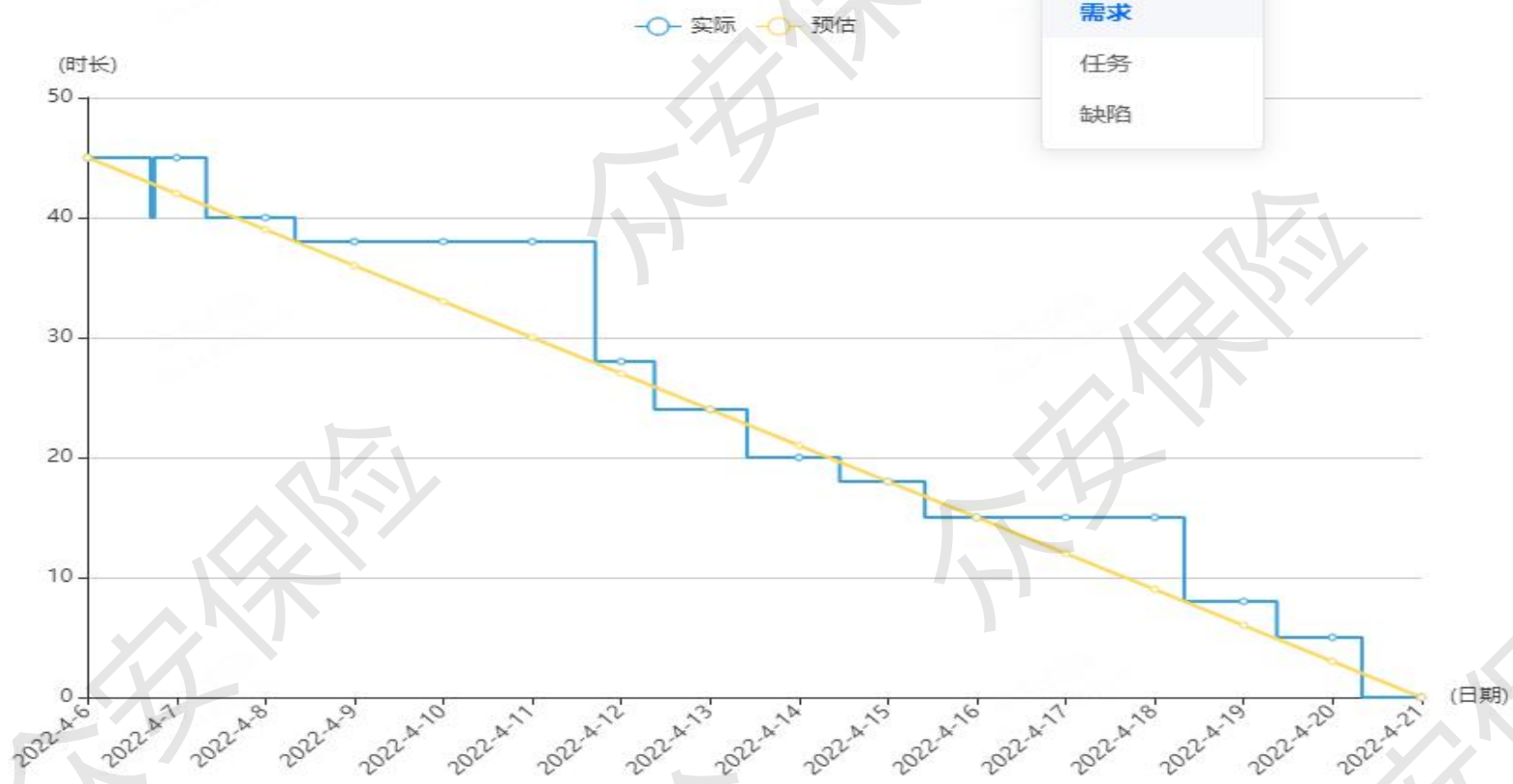


多种组合形式燃尽图

中国DevOps社区

迭代燃尽图

事项分组：需求　字段：故事点　是否过滤节假日：否　保存　刷新



迭代事项总览

开发任务
未开始　进行中　已完成　挂起

Story
未开始　进行中　已完成　挂起

缺陷
未开始　进行中　已完成　挂起

超期
Story

Sprint评审会

目的	1. 演示成果听取反馈，从而及时调整，真正交付业务和用户满意的产品 2. 反向驱动团队重视交付质量，不能演示出问题 3. 团队获得成就感，长期持续能够建立业务信任与满意度
时间	迭代结束前（通常在迭代的最后一天，也就是周五）
参与者	产品经理、SM 、Team全员参加， 由产品经理主导演示过程
活动	1. PO或Team成员给业务/用户代表介绍本迭代进展（计划、完成） 2. ——演示本迭代完成的特性，每个特性演示后暂停听取反馈，再继续下 一个 3. 总结并讨论对主要反馈的处理方式，立即修改还是后续迭代处理
输出	1. 反馈或问题记录

		周一	周二	周三	周四	周五
第一周	上午	Sprint计划会	每日Scrum站会	每日Scrum站会	每日Scrum站会	每日Scrum站会
	下午	开发测试				
第二周	上午	每日Scrum站会	每日Scrum站会	每日Scrum站会	每日Scrum站会	每日Scrum站会
	下午	开发测试		需求梳理会(1h)	code review(1.5h)	**Sprint评审会(1h)** Sprint回顾会(1.5h)

典型问题—如何让迭代演示会更高效?

何时需要Sprint评审会?

两个场景:

1. 新产品MVP阶段或大版本，需要数个迭代后 才会版本上线，需要尽早演示，尽早得到反馈 并修正（而不是上线前UAT或上线后才抱怨一 堆问题）

2. 尽管每个迭代都上线，但有一些大的特性可能要跨多个迭代才能完整交付用户，需要迭代演 示，尽早得到反馈

对那些小的，持续每个迭代立即上线的功能不一定需要迭代演示

如何开好Sprint评审会?

1. 团队高度重视交付质量，稳定和测试通过
2. 演示前准备（构建配置、部署、服务器与网络环境、演示数据！）
3. “演示必败”的警惕。面对挑战，持续坚持
4. 演示主要特性，不演示小的缺陷修复
5. 演示中讲解做成什么样了，不讲解怎么做的， 业务不关心实现

讲场景化问题，演示场景化问题的解决方案

Sprint回顾会

目的	识别本迭代的开发过程中存在的工作方式、方法问题，并确定下迭代改进计划。
时间	迭代的最后一天下午，1.5小时以内
参与者	产品经理、SM 、Team全员参加， SM组织团队开展
活动	1. 团队每个回顾上个迭代中碰到的障碍、问题写在卡片上，按照Keep doing、more of 、stop doing、less of 、start doing五个维度分类卡片； 2. 团队总结上个迭代的目标达成问题，有哪些用户故事未能完成或者存在质量问题。 3. 团队使用5Why法，分析目标达成问题与碰到的障碍、问题之间的因果关系，补充新识别的障碍、问题。 4. 团队投票决定，哪些障碍或问题是要解决的TOPn问题。 5. 团队确定问题的改进方案，下个迭代的行动计划。
输出	1. 要解决的Top N问题（一般最多三个） 2. 解决问题的方案

		周一	周二	周三	周四	周五
第一周	上午	Sprint计划会	每日Scrum站会	每日Scrum站会	每日Scrum站会	每日Scrum站会
	下午	开发测试				
第二周	上午	每日Scrum站会	每日Scrum站会	每日Scrum站会	每日Scrum站会	每日Scrum站会
	下午	开发测试		需求梳理会(1h)	code review(1.5h)	Sprint评审会(1h) **Sprint回顾会(1.5h)**

团队回顾，有效改善团队效能

回顾会的首要原则

无论我们发现了什么，考虑到当时的已知情况，个人的技术水平和能力，

可用的资源，以及手上的状况，我们理解并愿意相信：

每个人对自己的工作都已经全力以赴

回顾会的安全度检查

- 为了达到回顾会的目的和积极效果，会议组织者有必要在会议开始之前做一次”检查“，确保团队所有成员真正认为在这个会议上的任何发言都不会影响到个人声誉与绩效，不会收到鄙视等，即”在这个会议上发言是安全的“
  - 我什么都不准备说，我觉得不安全；
  - 我不会说太多，我会更多的让别人说；
  - 我会主动说一些，但是大部分话题难以启齿；
  - 我几乎可以无所不谈，但是个别话题会有所保留；
  - **完全没有问题，我可以无所不谈；**

- **对于一个成熟的团队来说，一段时间后做一次“安全度检查”是非常必要的。对于一个不成熟的团队，最好每次都做安全度检查；**

典型回顾会的流程

第一步：跟进以往回顾会制定的Action执行情况，直到关闭

- 可用红、黄、绿灯贴纸跟进改进状态；
- **绿灯**：改进完成；**黄灯**：改进中；**红灯**：还没有开始改进；
- 连续积攒了三个红灯，要引起高度重视，为什么？

第二步：本次回顾

- 每个人写条贴条，发散问题
- 引导者对所有问题归类，相同相似的放到一起，不明白的让写的人快速讲解；
- 团队对问题排优先级，挑出最优先希望得到解决、且有可能解决的三个问题（可以通过投票）

第三步：讨论后续改进措施及负责人

- 分别就最优先希望得到解决的问题一一讨论切实可行的改进措施，具体行动，团队内达成共识；
- 明确每个改进措施的负责人；
- 将改进措施及负责人可在团队看板上可视化，持续跟踪。

回顾会的多种形式

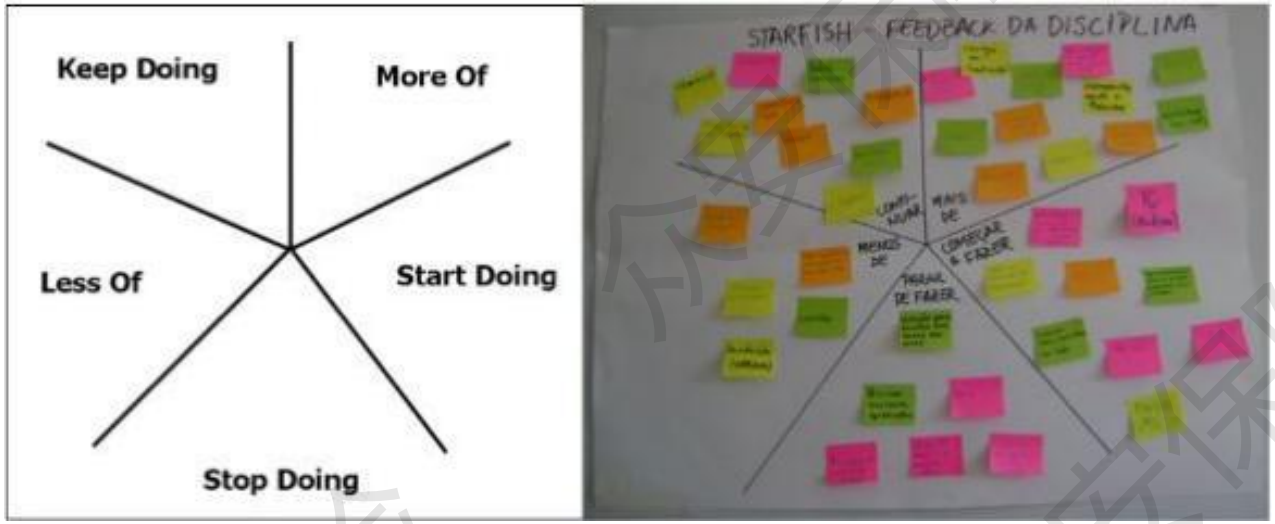
Well/Less Well/Actions

重点改进专题分析

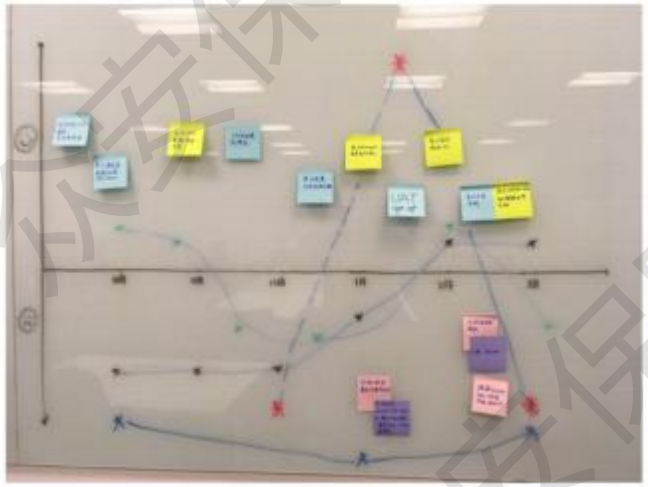
1. 问题陈述
2. 当前状态
3. 期望目标状态
4. PDCA的分步骤措施
5. 每一步的评估反馈机制
6. 负责人

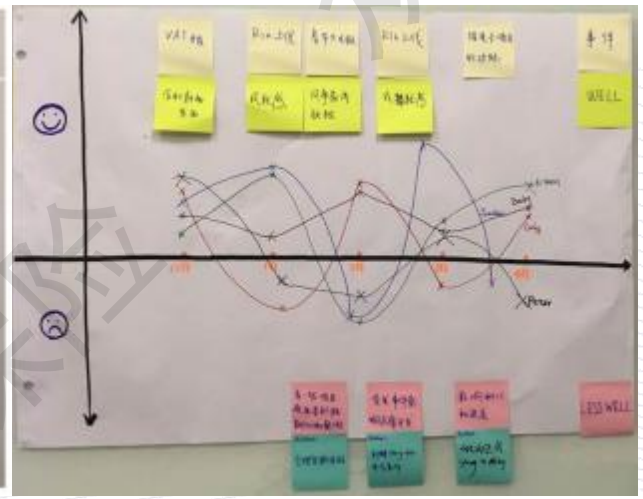
回顾会的多种形式

海星图



心情曲线





回顾会的多种形式

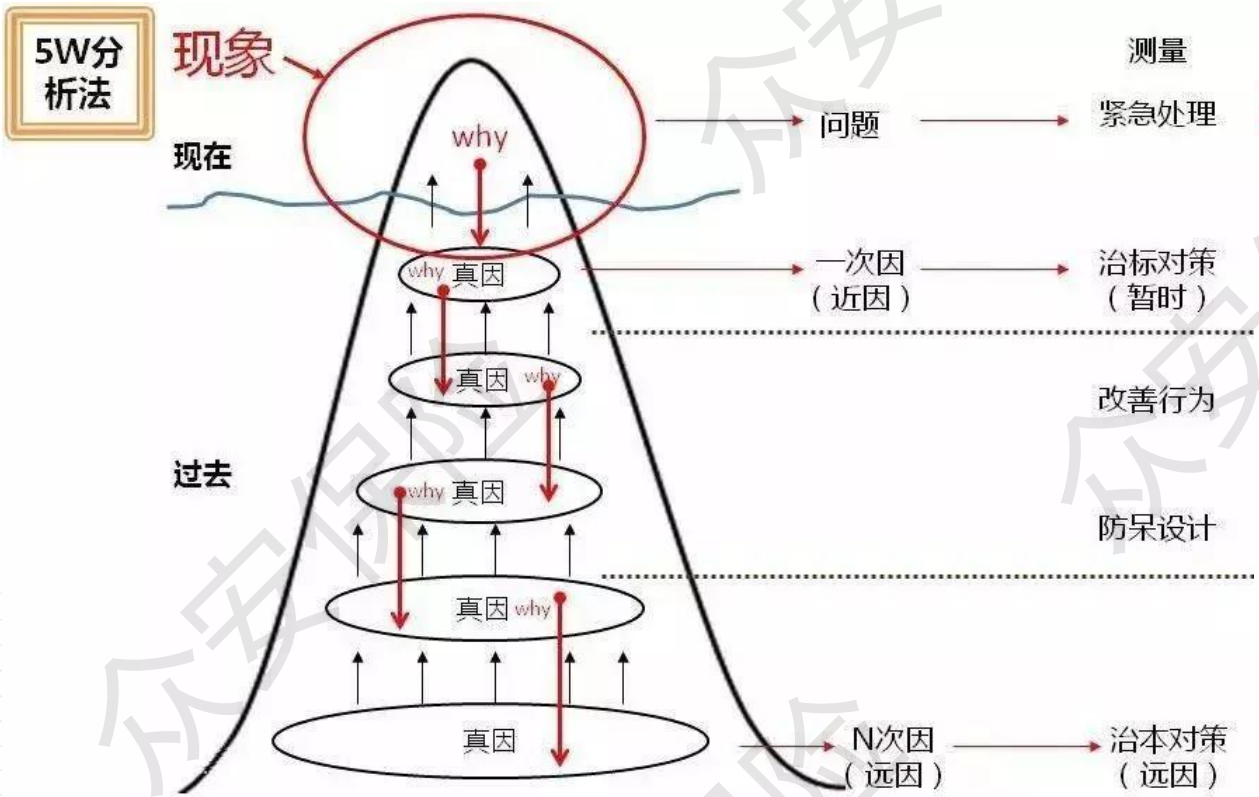
专题讨论

例如专门讨论如何开发与测试加强合作、讨论单、主干开发如何执行等（成熟度达到一定程度后，可以穿插这种形式）

鱼骨图/Root Cause Analysis

例如组织对最近发生的严重生产问题进行根因分析，从而制定改进措施；

回顾会根因分析法



5WHY分析的运用形式图

判定：OK -- 要因事项确认后，不是引起问题的原因，认为没有问题不必对此

NG -- 要因事项确认后，判定是问题引发原因，需要进行再发防止对策

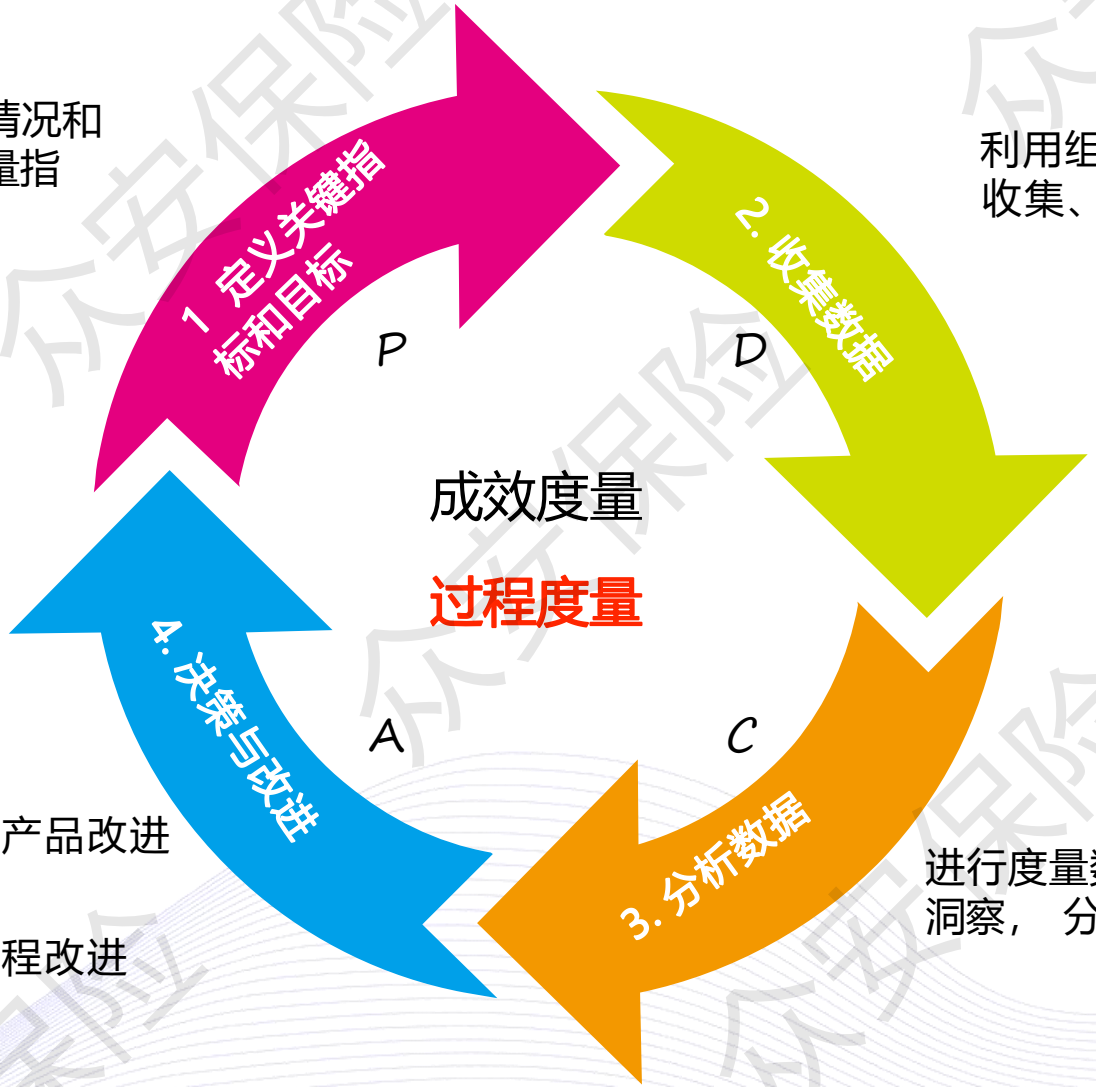
解决方案，如何具体执行

改进项	负责人	具体措施	解决时间

过　　程　　度　　量

度量的PDCA

组织或团队， 根据实际情况和现状能力， 设计关键度量指标， 并确定改进目标

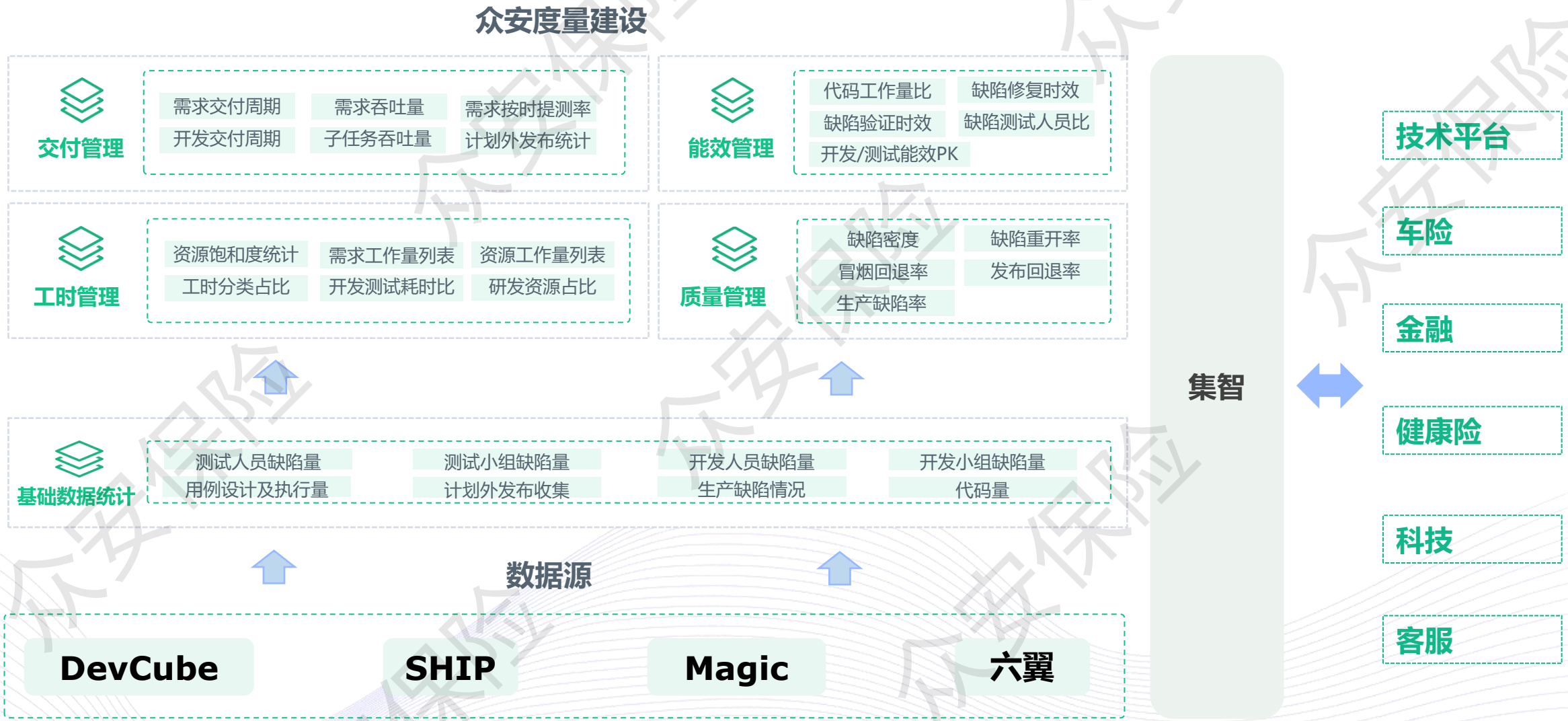


利用组织或团队提供的度量工具，收集、统计与展示度量数据

- 基于成效度量进行业务与产品改进决策， 或调整指标本身
- 基于过程度量进行研发过程改进

进行度量数据分析， 从数据中获得洞察， 分析问题， 识别改进机会

过程度量的指标体系建设



关键过程度量指标

端到端交付周期

每一个需求，从方案确认到首个版本交付用户的周期时间

衡量IT团队对客户、市场机会的响应力，核心关键指标！周期越短，响应力越高。

需求实现周期

每个故事从第一次进入迭代计划的迭代开始日期，到该故事测试 通过的周期时间

衡量交付团队对需求变化的响应速度，周期越短，响应越快，变化越灵活

迭代速率

每个迭代周期内实际完成（测试通过）的故事总点数

衡量团队的迭代实际容量，用以参考制定更合理的迭代计划

关键过程度量指标

迭代完成率	每个迭代实际完成（测试通过）的故事总点数，除以迭代计划的 总故事点数
	衡量团队迭代计划及其执行的有效性、合理性，70～80%比较合适
技术债务率	Sonar工具扫描评价的质量指标，综合了代码问题、复杂度和重 复率等因素
	衡量软件的内在质量
生产缺陷数	产品一段时间内（如每个月）在生产环境报告的缺陷数
	衡量团队所交付产品的最终质量（用户感受质量）

实践中的关键过程度量指标

哪里浪费，度量哪里！！

效能度量的效果

问题/现状：

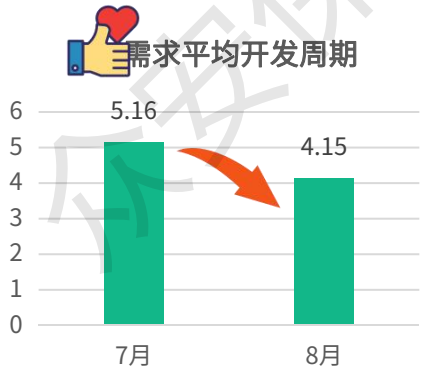
- 需求交付过程中，平均开发周期较长，交付效率待提升
- 需求bug未能及时处理及修复

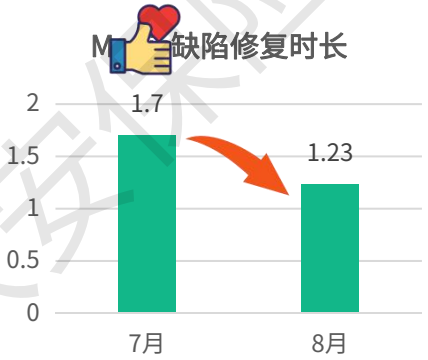
实施策略：

- 拆分需求，story颗粒度平均小于5个故事点，降低需求复杂度，限制在制品数量，实现小批次开发，提升效率
- 推动团队及时关注需求状态，做到需求及时KT，及时开发并提测，缩短交付时长
- 定期进行缺陷优先级确认，及时清理
- 定期复盘迭代执行情况，优化改进

成果产出：

- 团队：
  - 实际开发周期下降24%
  - 缺陷处理等待时长下降了27%
- ic团队：
  - 实际开发周期缩短34%
  - 缺陷修复时长下降38%
- 团队 缺陷处理等待时长下降61%





Q&A