TechExcel Makes
Agile Development
SCALABLE !

从大团队的计划过程看敏捷生态系统

以网络游戏研发为例

陈勇

2009-9-14

敏捷中国大会

TechExcel

ThoughtWorks®

InfoQ Enterprise Software Development Community

分享：一次实施Scrum的真实经历

- 沉默的Scrum团队

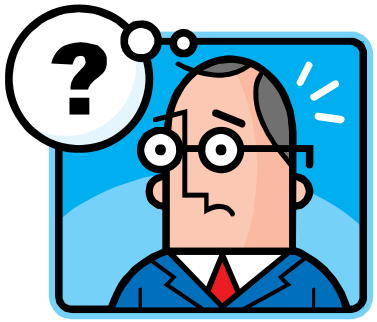
传统计划vs.敏捷Scrum计划

- 角色
  - 高层领导，项目经理，团队
- 团队
  - 分工
  - 个体绩效考核
- 方式
  - 领导估算
  - 经理分派任务
- 动力
  - 领导压力
- 常用语
  - 他最倒霉，分了一个最难的活，20天肯定不可能做完，还导致整个项目延期。

- 角色
  - Product Owner，Scrum Master，团队
- 团队
  - 跨职能团队
  - 团队绩效考核
- 方式
  - 团队估算
  - 自行领取任务
- 动力
  - 同行压力
- 常用语
  - 我们一开始都把事情想简单了，那个关键任务根本不可能20天做完，还好有个次要任务放弃了，三个人一起努力才完成了它。

但是经常听到

- 我们正在实施敏捷Scrum but……
  - Product Owner有时候不来参加计划会……
  - Product Owner讲完需求就走了，我们留下估算……
  - 计划会上人们各自估算各自的……
  - 每日立会上人们各说各的……
  - 迭代中间还是发生了变更……
  - 每次迭代都完不成……
  - 反思会上大家不说话……
- 大型团队中，这些问题更加常见



敏捷生态系统

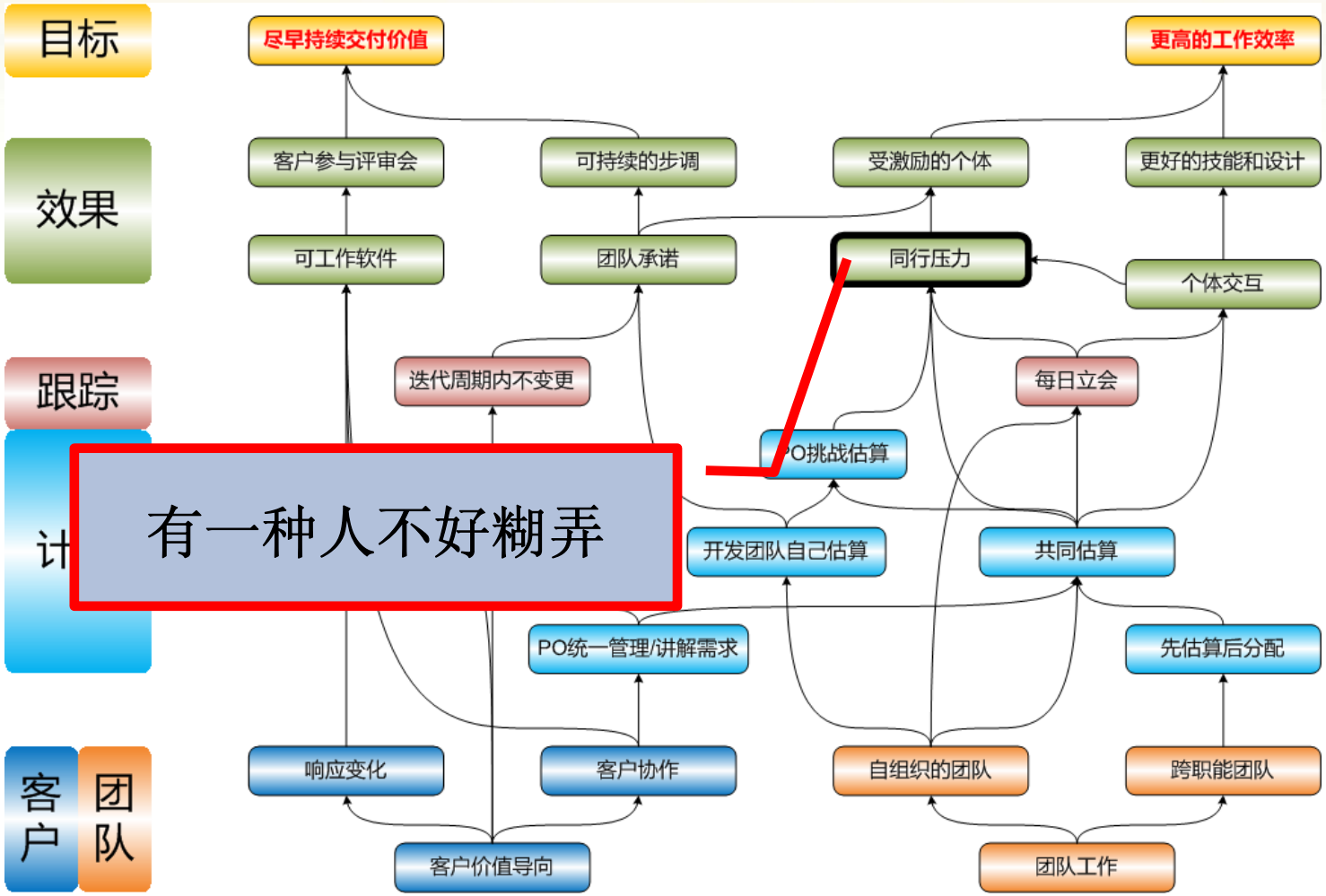
回到起点：为何要敏捷？

- 外因
  - 行业
  - 客户
  - 公司
  - 往往支持敏捷
- 内因
  - 团队规模
  - 文化
  - 经验
  - 纪律
  - 往往反对敏捷

从领导指令到自主工作

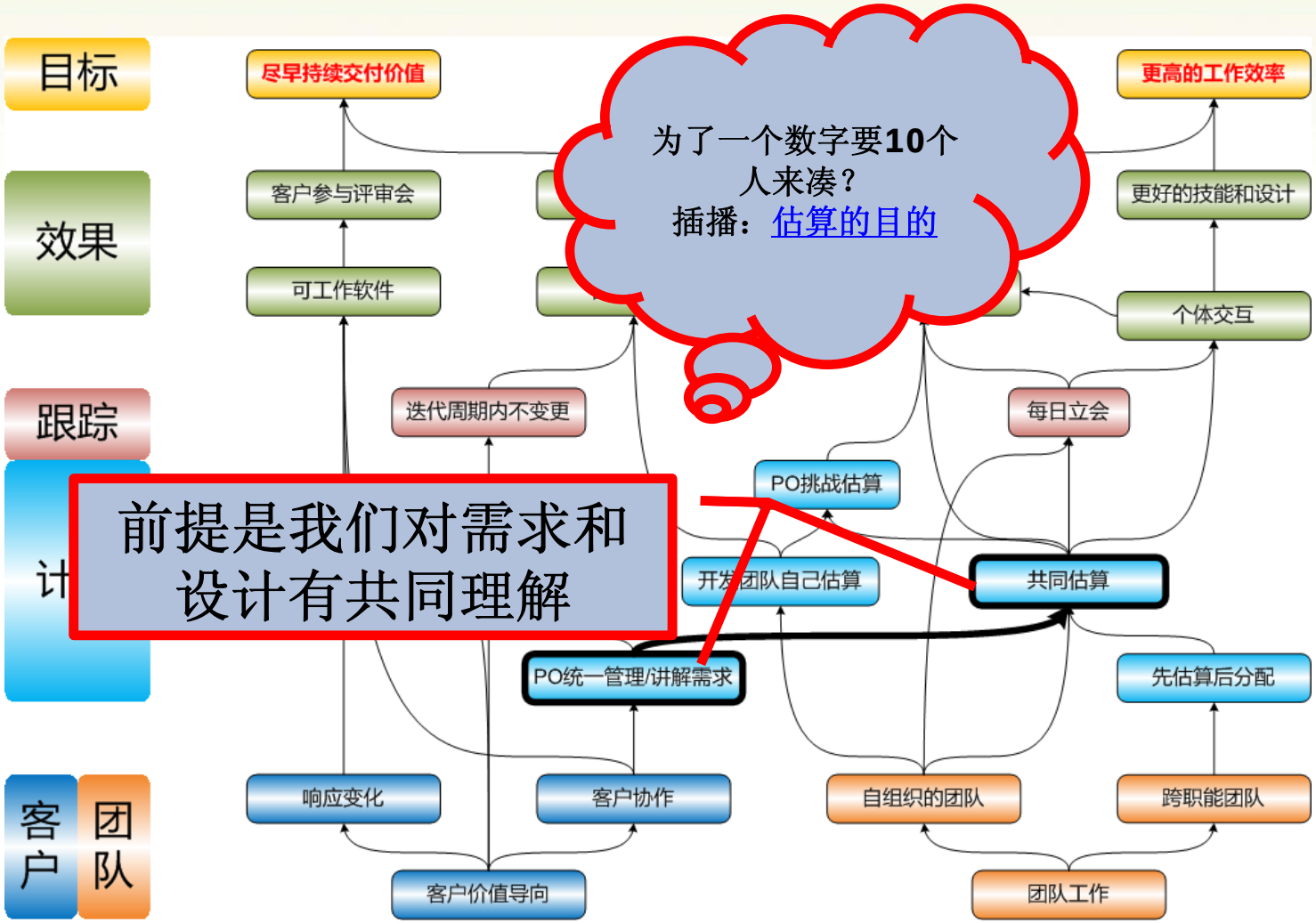
- 偷懒 / 过度悲观
- 激进 / 过度乐观
- 错误的设计实现方法
  - 不知道已经有可复用的模块
  - 误以为某个模块很容易复用
- 过度分工
  - 延期时无人能提出异议
  - 延期时无人能帮忙

敏捷Scrum是怎样解决这些问题的？



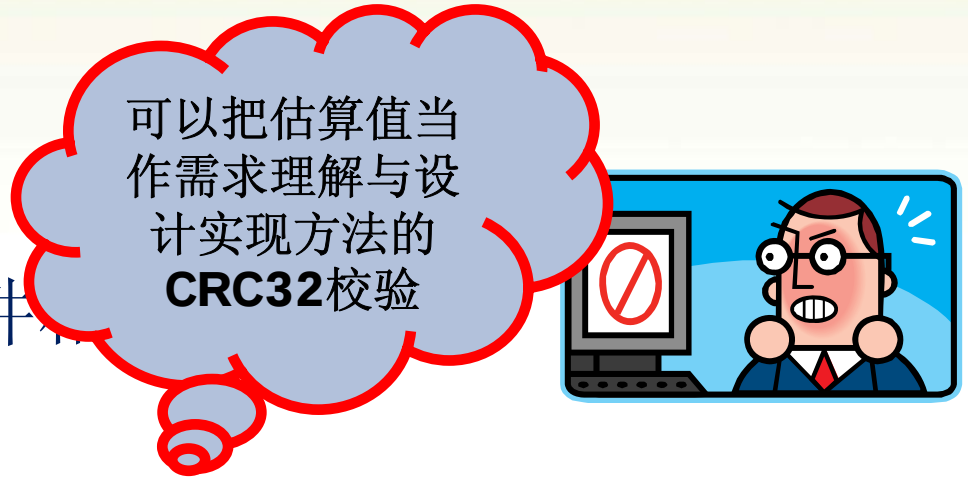
敏捷Scrum是怎样解决这些问题的？

敏捷Scrum是怎样解决这些问题的？

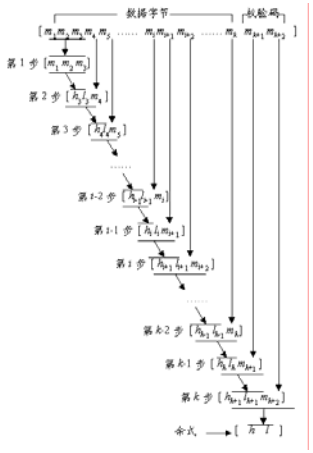


如何知道一个传送过来的文件没有损坏？

- 方法1
  - 先传送一个文件，[illegible]次，如果两个文件[illegible]损坏

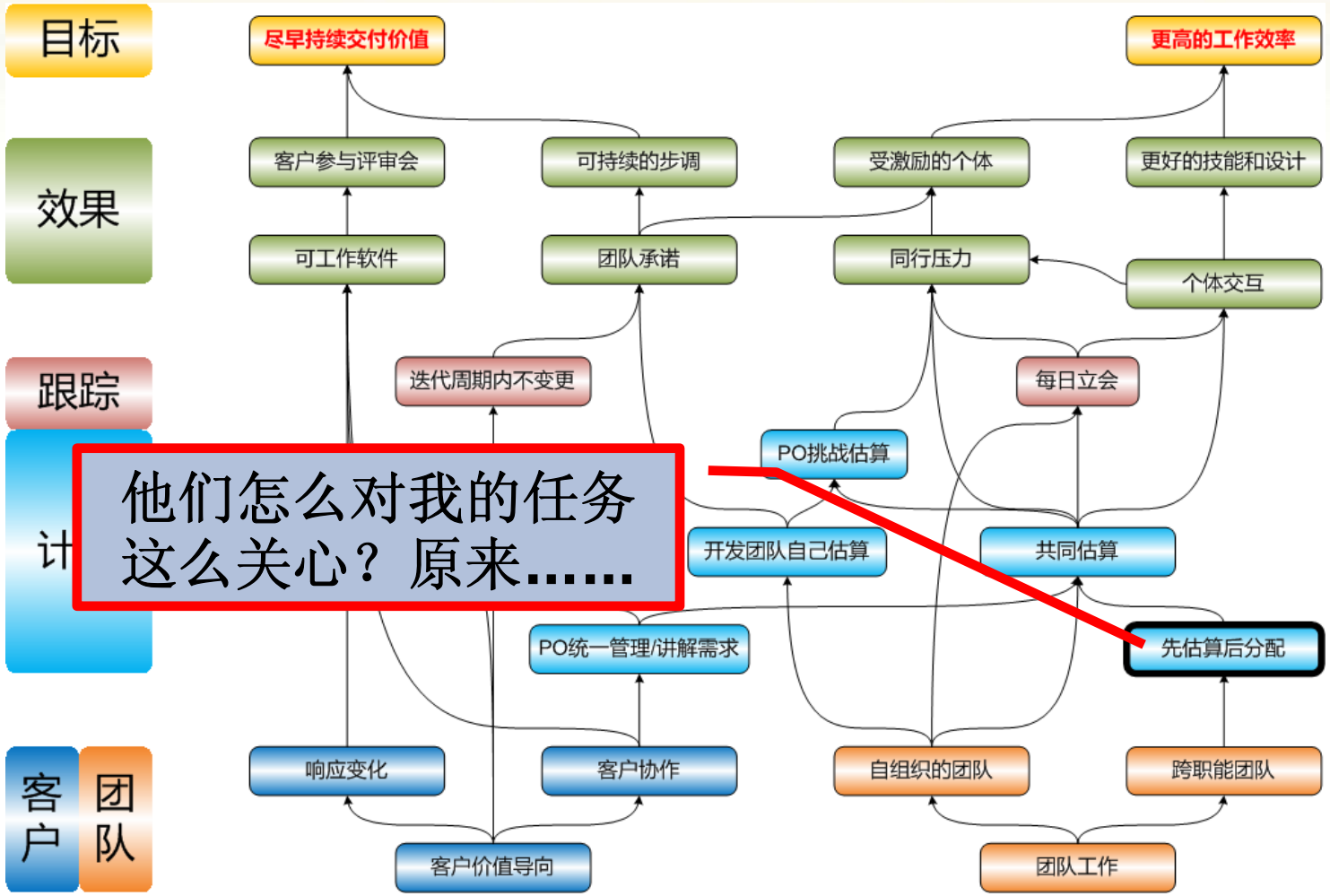


- 方法2
  - 先传送文件，再传送文件的数据校验和（或CRC32），如果计算后两者无误，就没有损坏

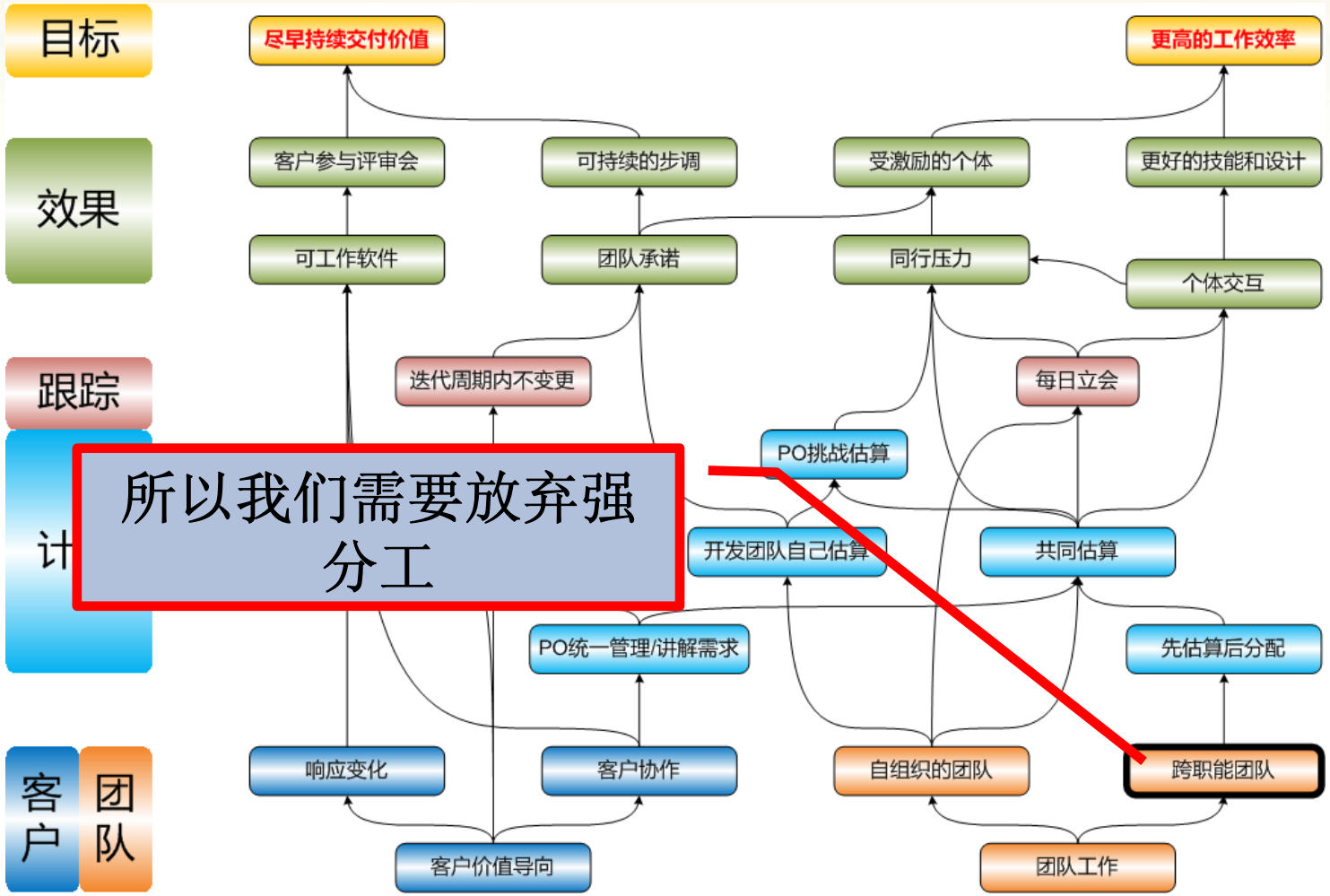


最高效的敏捷计划方式：敏捷扑克

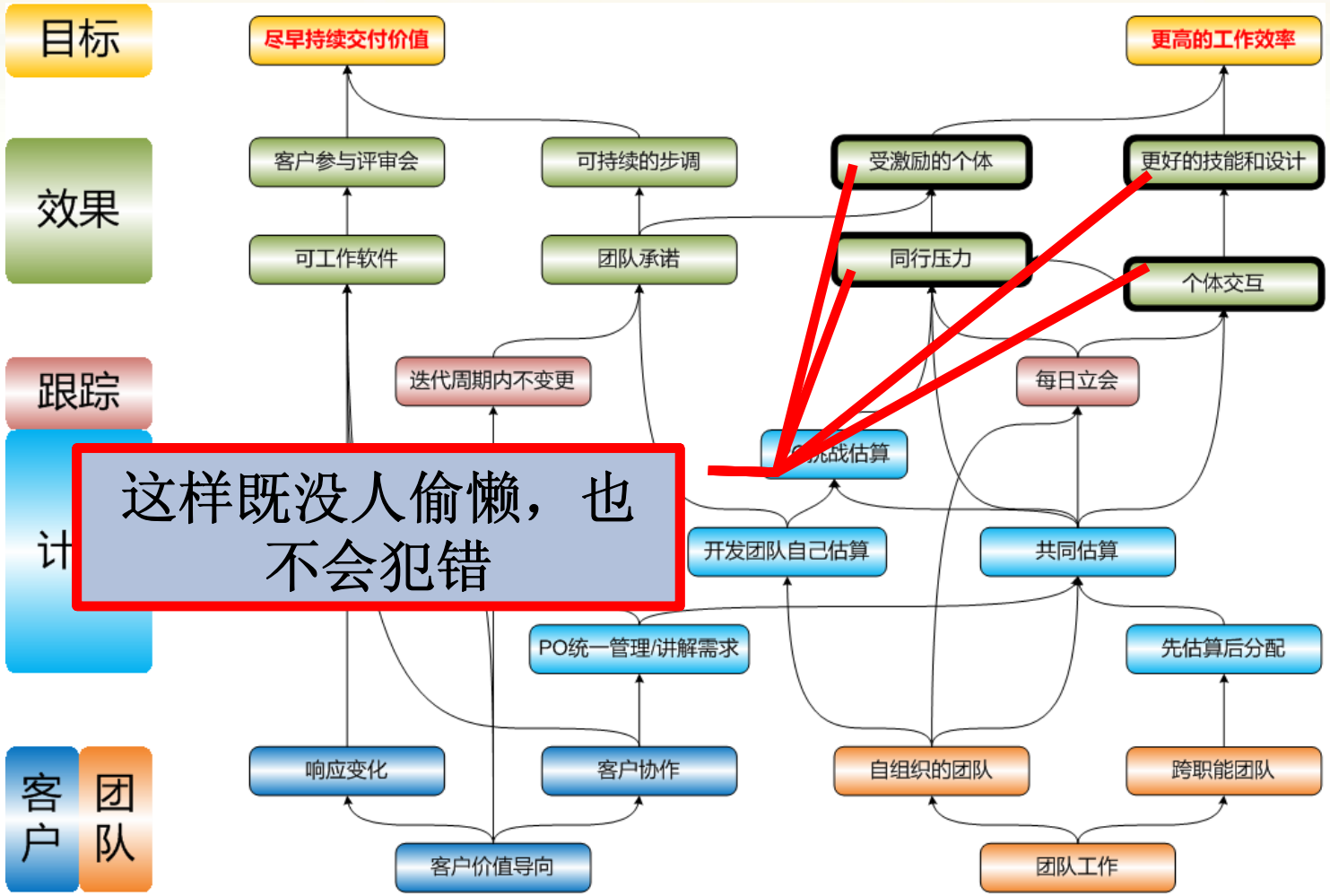
敏捷Scrum是怎样解决这些问题的？



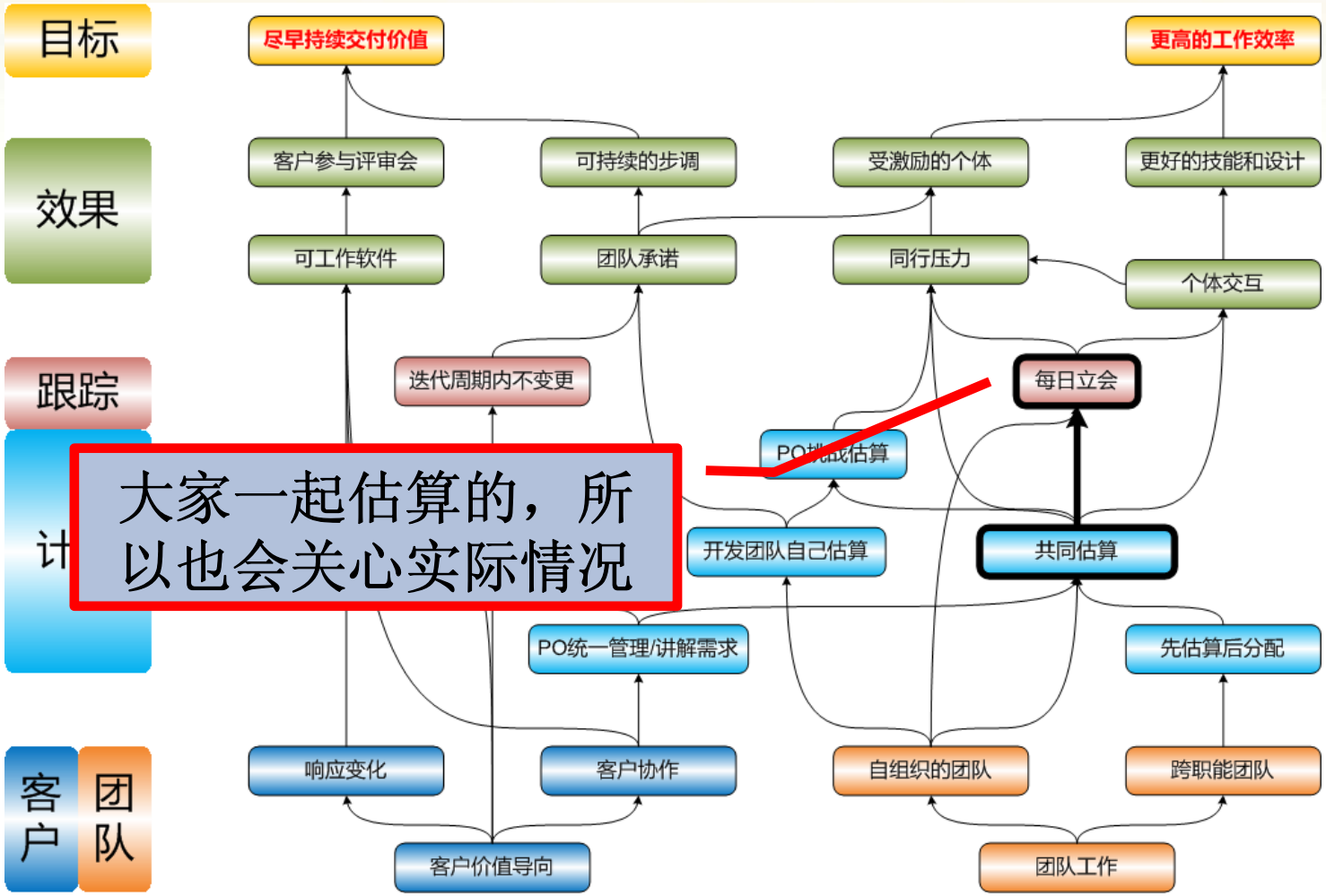
敏捷Scrum是怎样解决这些问题的？



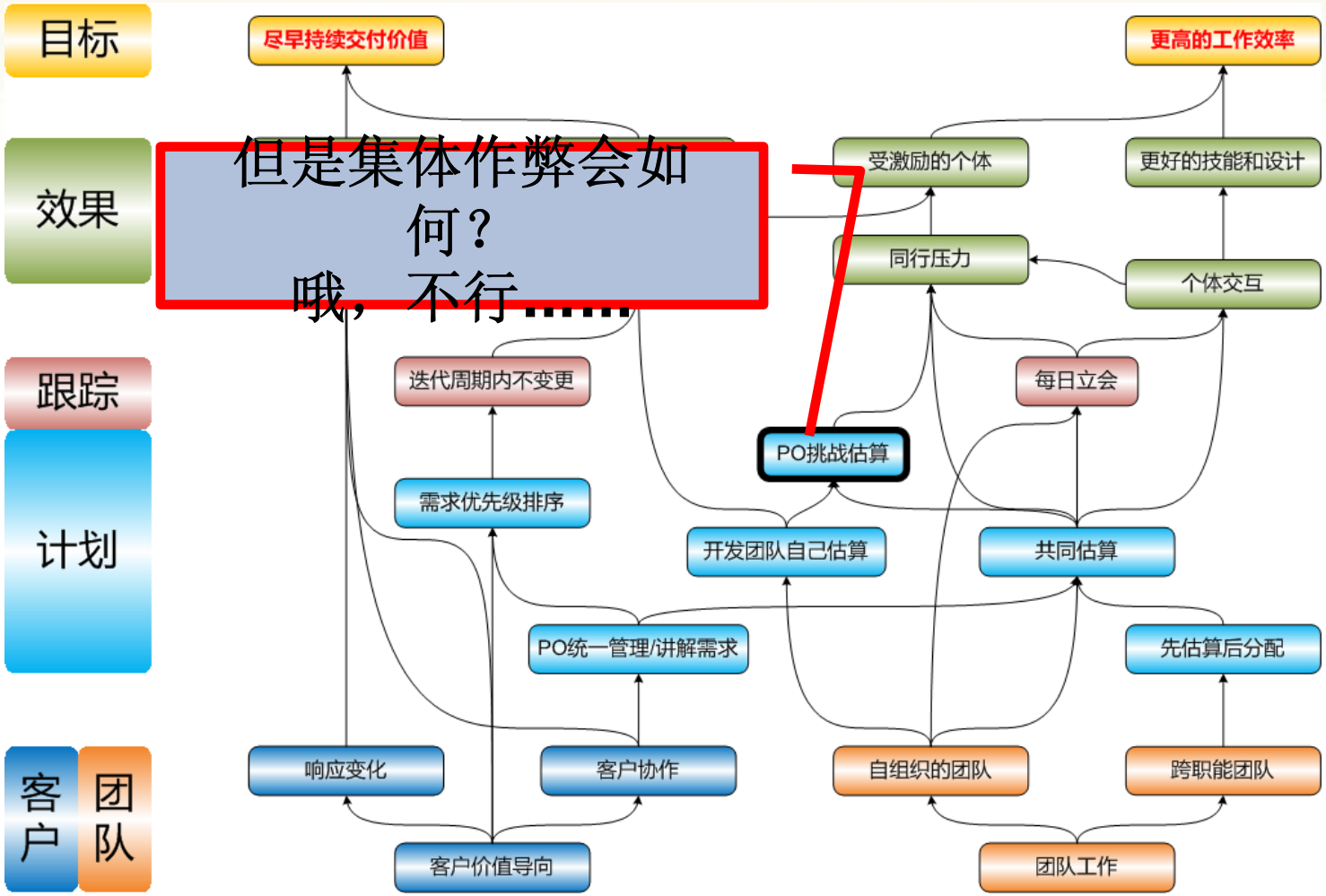
敏捷Scrum是怎样解决这些问题的？



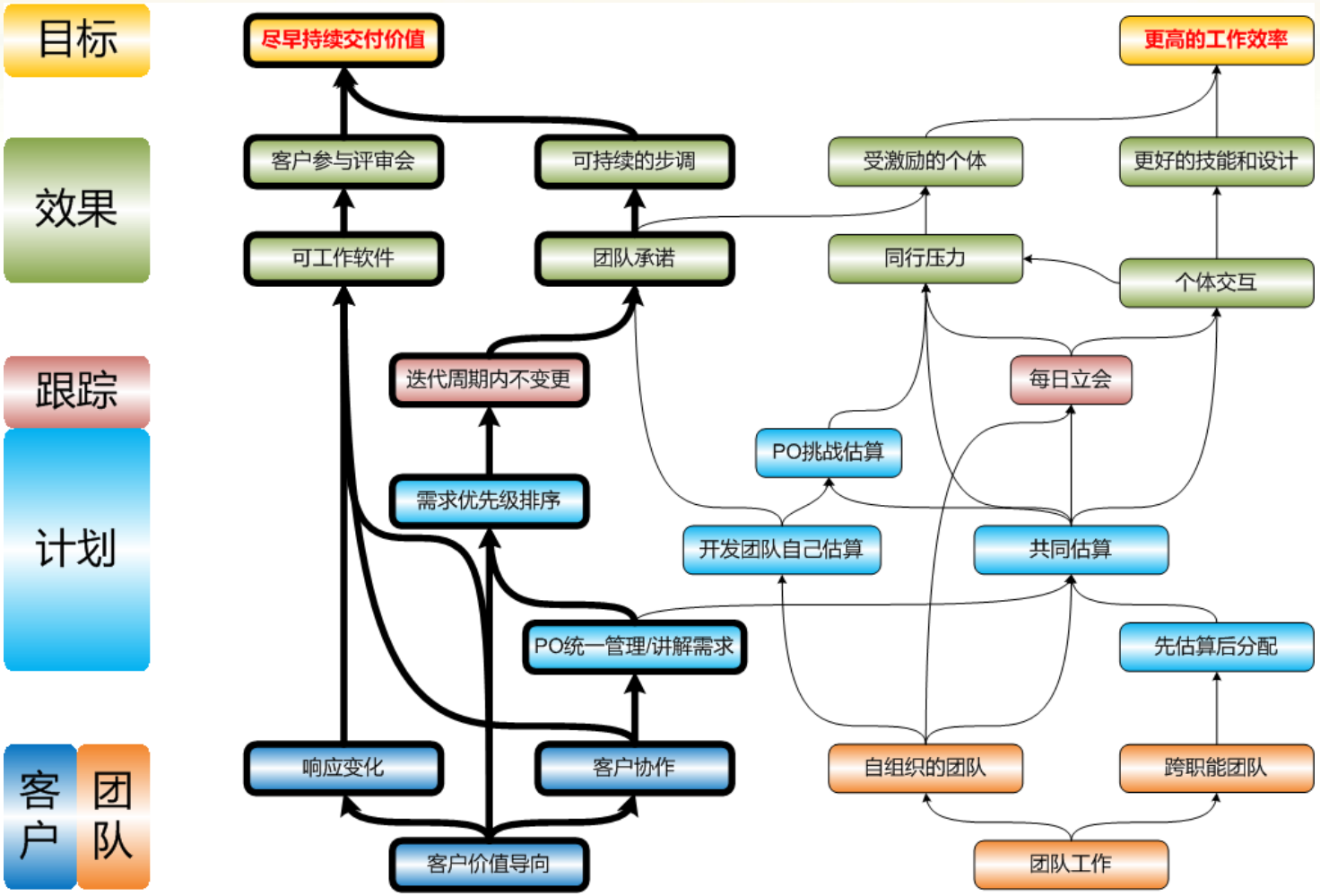
敏捷Scrum是怎样解决这些问题的？



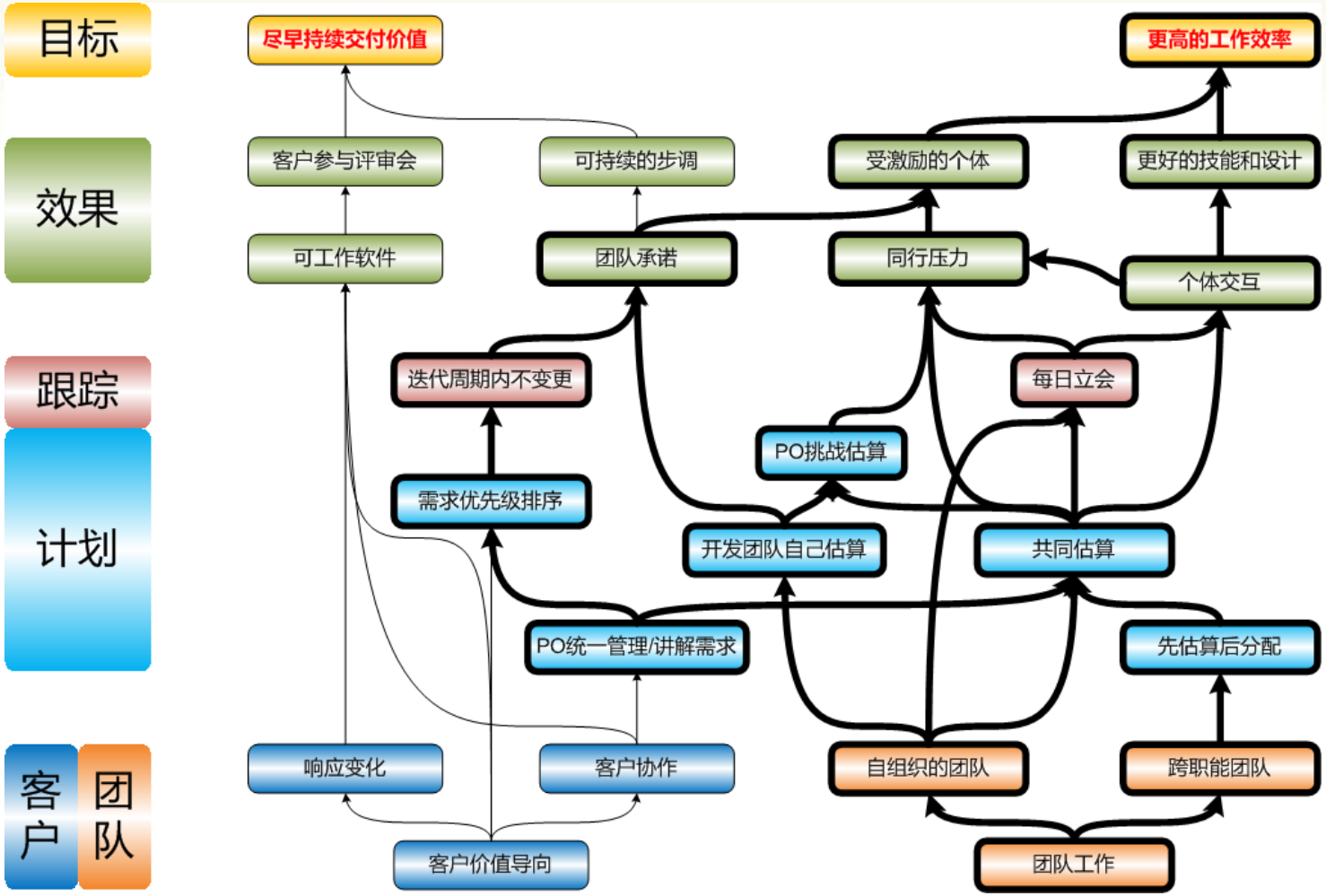
敏捷Scrum是怎样解决这些问题的？



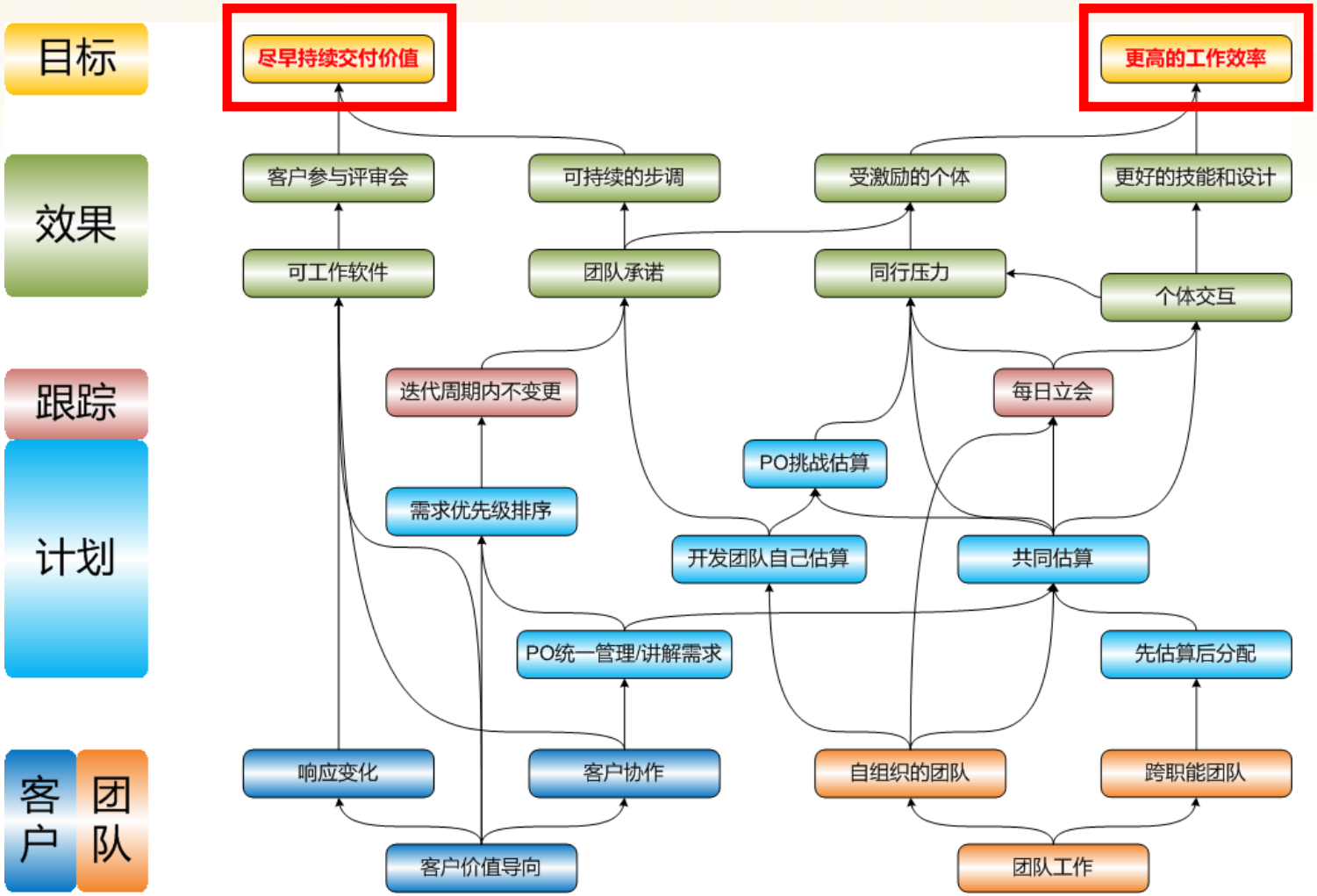
敏捷Scrum计划生态全貌：需求管理



敏捷Scrum计划生态全貌：计划跟踪



敏捷Scrum开发的最终目标



如此完美的生态系统到底出了什么问题？

大型团队实施敏捷的挑战

大团队计划问题：沉默的大团队

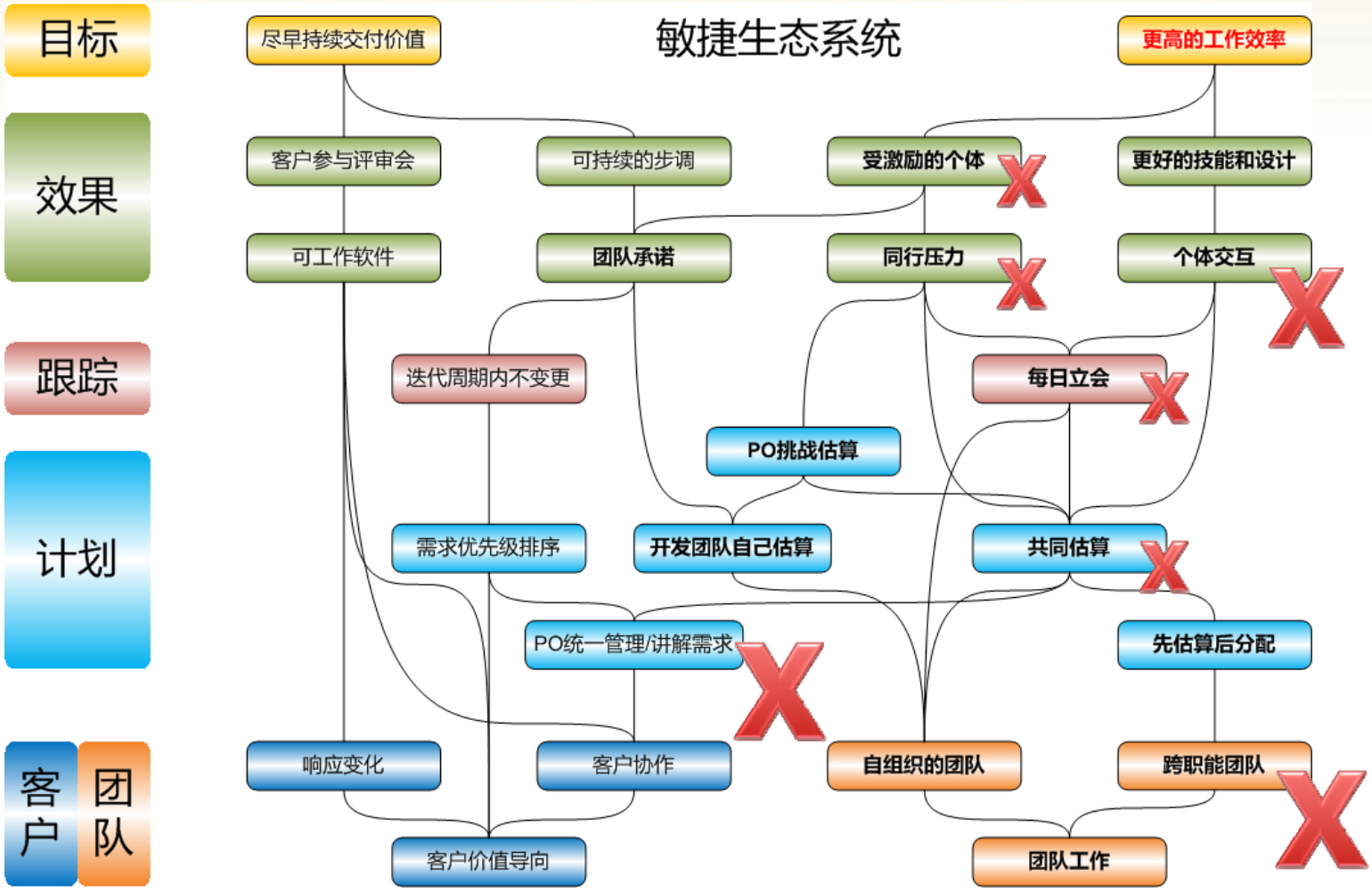
- 大团队效应
  - 新人得到沉默的机会
- 强分工效应
  - 孤独的计划者

案例：游戏研发中的敏捷和反敏捷因素

- 敏捷因素
  - 快速响应/反馈
    - 快速响应市场/玩家的需求变化
    - 快速响应竞争对手的变化
    - 快速修复平衡/质量等问题
  - 客户价值观
  - 投资安全性

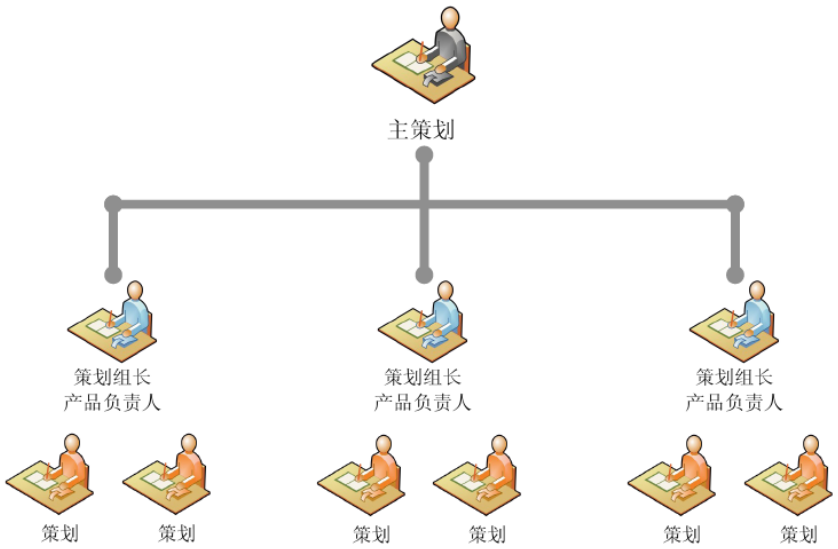
- 反敏捷因素
  - 团队庞大
    - 30～200人
  - 分工明确
    - 策划，文案，脚本
    - 程序，程序测试
    - 美工，2D/3D/原画
    - 黑盒测试
  - 变更频繁
    - 常常有非改不可的缺陷或漏洞

大团队/强分工下易受影响的生物



问题的解决：网游团队敏捷方法

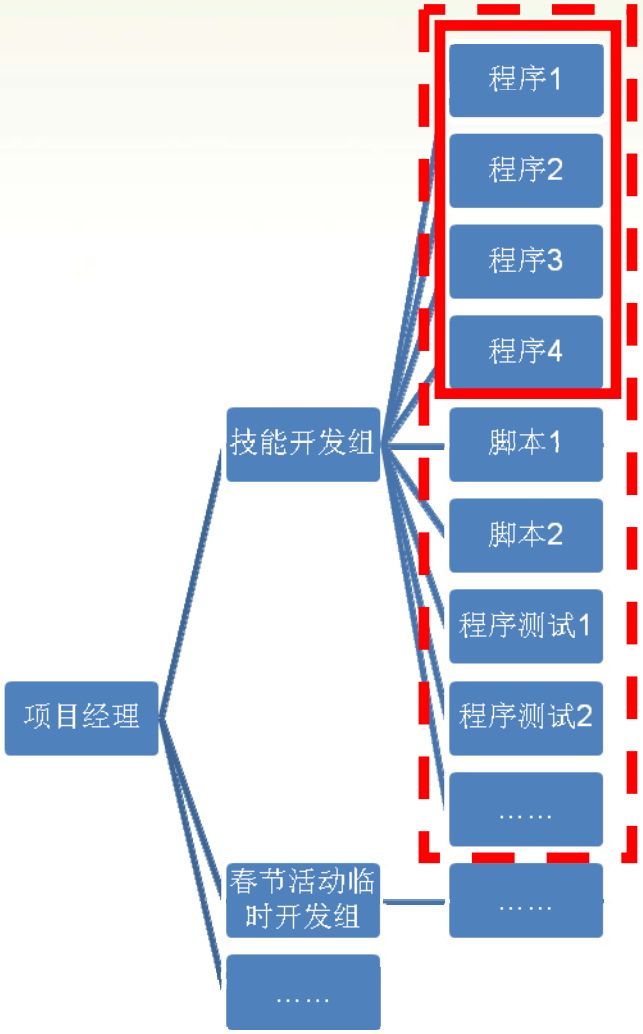
大型需求团队：Product Owner组



- 策划组成为Product Owner组
  - 主策划负责优先级设定
  - 策划组长负责需求解释
  - 国外游戏公司报告了将发行商邀请到PO组以帮助确认需求方向的实践

需求决策与需求细节的平衡

大型开发团队：基于功能的分组

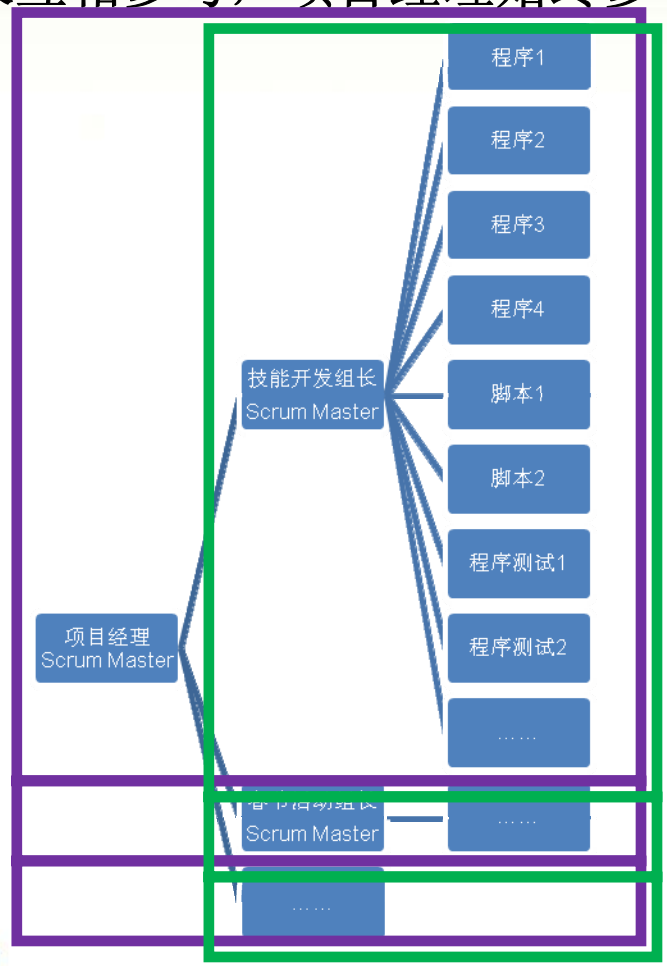


- 将开发组按功能分为几个功能组
  - 每个组都有其策划/程序/测试，可以独立工作
  - 每个功能组有各自的组长（Scrum Master）
  - 每个工种尽量两人以上
    - 共同计划与跟踪
    - 方便资源调配
  - 利于PO集中讲解功能

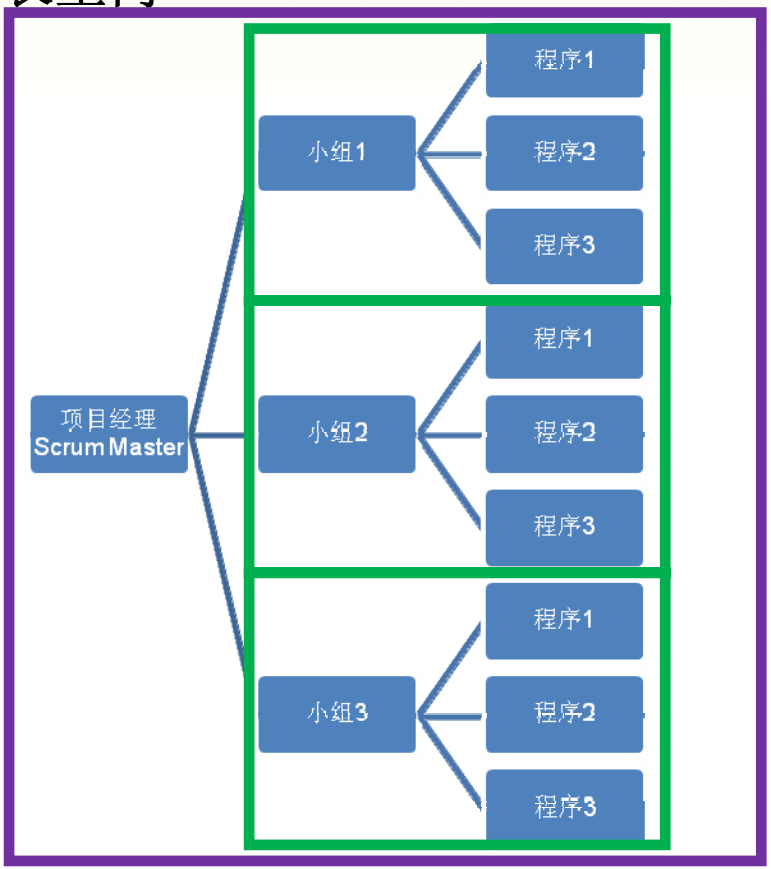
功能组更容易对成果产生认同感

大型开发团队：两种划分方式

大型团队：分别开会，分别估算，组长互相参与，项目经理始终参与



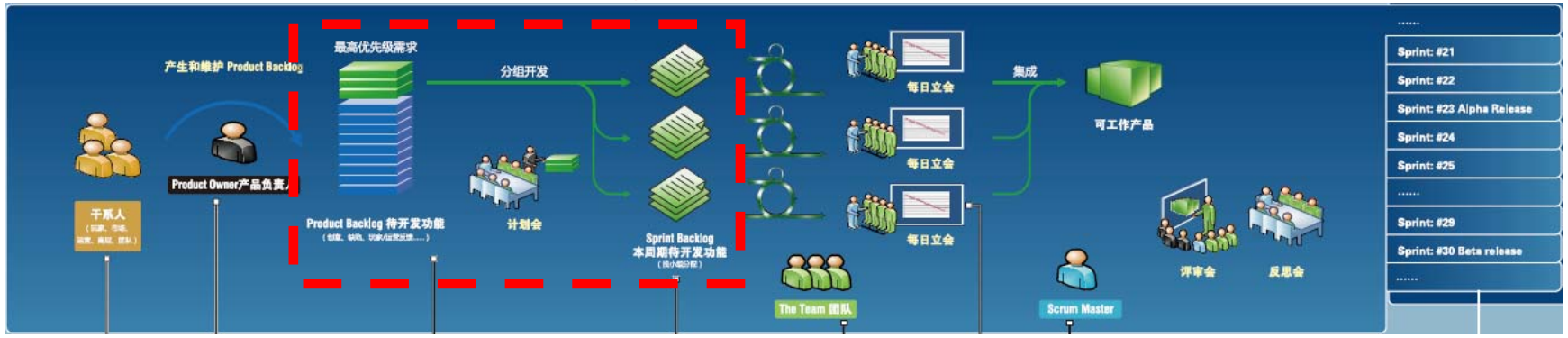
中型团队：一起开会，分别估算，组长主问



大型开发团队：计划

- 几个功能小组分别计划
  - 小组内部分别召开计划会
    - 小组内的同一工种或相关工种共同估算
  - 组间沟通
    - Product Owner是同一
    - Scrum Master参加每

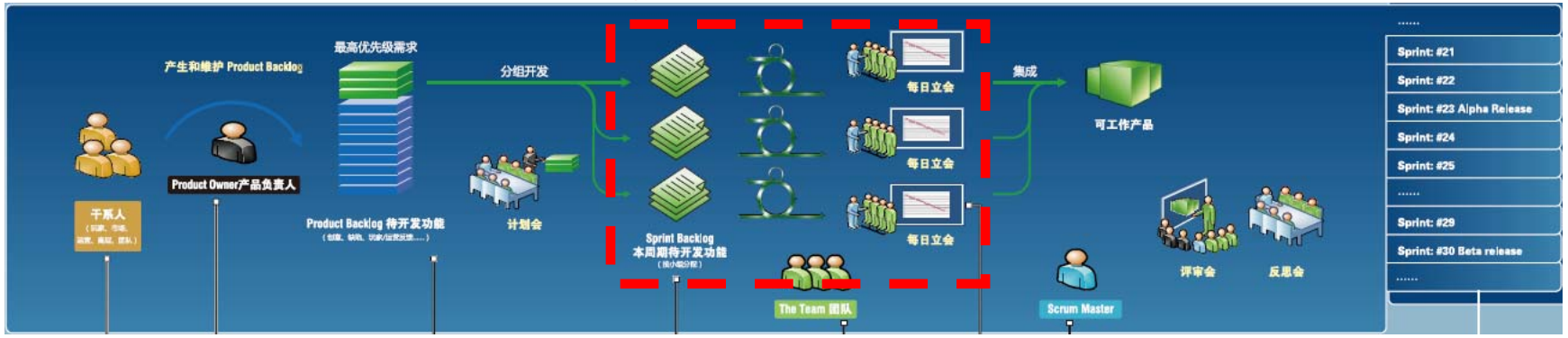
PO / SM实现了组间的沟通



大型开发团队：跟踪

- 几个功能小组分别计划
  - 小组内部分别召开每日立会
    - Product Owner指定PO团队中的“跟进人”跟进单个需求
    - 跟进人跨小组跟进
  - 在迭代期间持续集
    - 跟进人持续Review

跟进人增进了组间沟通



长周期开发：里程碑

- 在迭代之外，嵌套一个与游戏研发生命周期相吻合的版本计划
  - 不同阶段的计划截然不同
- 展示两个信息
  - 各里程碑在做什么
  - 各功能组在做什么

防止只见树木，不见森林

	Name	Duration	状态	完成百...
1	逍遥武林Online	31 days	-	0%
2	线上系统维护	31 days	-	0%
3	[迭代 #1] 122308_Release	5 days	完成	100%
5	[迭代 #2] 123008_Release	5 days	完成	100%
7	[迭代 #3] 011309_Release	7 days	进行中	90%
8	21 [程序]: 春节促销活动	5 days	进行中	70%
9	21 [美工]: 春节促销活动	3 days	完成	100%
10	21 [策划]: 春节促销活动	4 days	完成	100%
11	Bug修复 (0/0/0H)	5 days	-	0%
12	[迭代 #4] 012009_Release	5 days	未开始	0%
14	[迭代 #5] 012709_Release	6 days	未开始	0%
16	逍遥武林-西域传奇	436 days	进行中	70%
17	策划阶段	70 days	完成	100%
21	可行性测试阶段	48 days	完成	100%
24	小规模制作阶段	46 days	完成	100%
27	大规模制作阶段	135 days	完成	100%
33	内测阶段	64 days	进行中	40%
34	[迭代 #1] Sprint1	22 days	进行中	80%
35	服务器结构调整 (3/3/...	7 days	进行中	70%
36	41 [程序]: 服务器...	1 day	完成	100%
37	40 [程序]: 单台服...	2 days	进行中	50%
38	42 [程序]: 服务器...	5 days	进行中	41%
39	聊天系统 (7/11/6W 2...	16 days	进行中	20%
51	生活技能 (4/9/8W 2D)	21 days	进行中	40%
	师门任务 (4/5/1W 1D...	8 days	进行中	20%
	武功系统 (6/6/2W 2D...	9 days	进行中	30%
	Bug修复 (0/0/0H)	7 days	未开始	0%
	[迭代 #2] Sprint2	20 days	进行中	10%
	[迭代 #3] Sprint3	22 days	未开始	0%
	公测阶段	69 days	未开始	0%
	[迭代 #1] Sprint1	22 days	未开始	0%
94	[迭代 #2] Sprint2	23 days	未开始	0%

总结：大团队的敏捷生态

- 容易受到破坏的生物
  - 跨职能团队，个体交互，PO统一管理/讲解需求
- 一些有效的方法
  - **PO团队管理和跟进需求**
  - 宏观上PO/SM要跨小组计划/跟踪
  - 微观上每个小组要形成同一工种共同估算和跟进
  - 里程碑/各小组的Sprint Backlog信息集成展示
- 最终目标
  - 在一定程度上维护生态系统不被破坏

分享：故事的结尾

- 沉默的Scrum团队

谢谢

- Q&A