

*Scrum*开发模式经验分享

2017年6月 北京数人科技有限公司

邱戈川 (了哥)

公众号: VIPDOCKER

Contents

01. 敏捷Scrum模式概貌

02. 角色分工

03. 团队的日常活动

04. PO的工具/其他

01.敏捷 *Scrum* 模式概貌

现行开发模式、开发过程的痛点

开发过程的各个环节的明确边际，造成信息传递链条过长，沟通成本以及问题反馈效率低

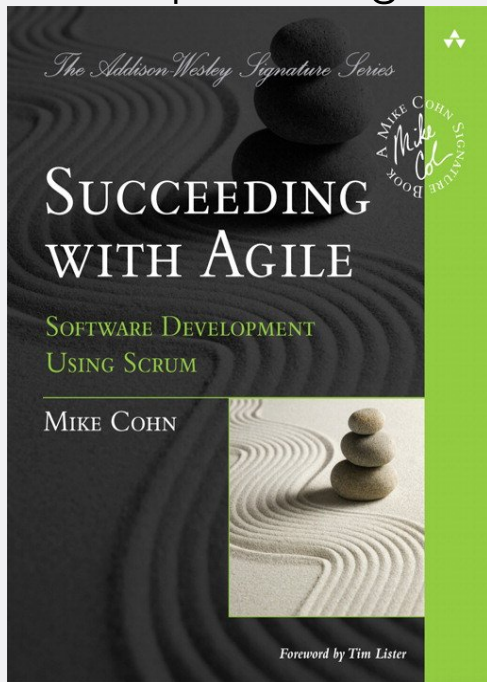
新技术的引入（微服务、容器化），不再适应大军团作战，而是需要匹配的灵活的端到端团队

客户的需求变化已经不再局限于整体产品购买，而是倾向于业务快速迭代，大军团作战已经难以快速交付、快速变更



什么是Scrum

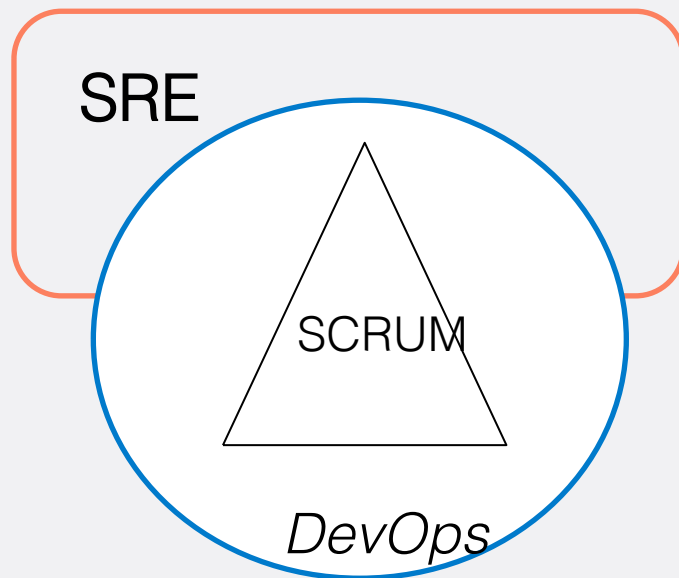
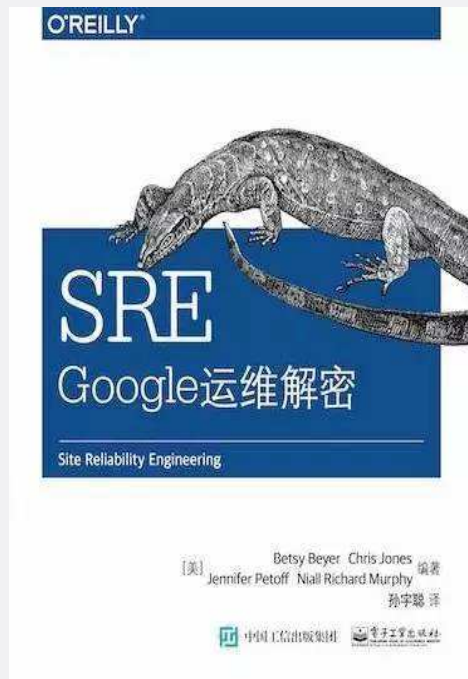
[https://en.wikipedia.org/wiki/Scrum_\(software_development\)](https://en.wikipedia.org/wiki/Scrum_(software_development))



需求范围自主性？

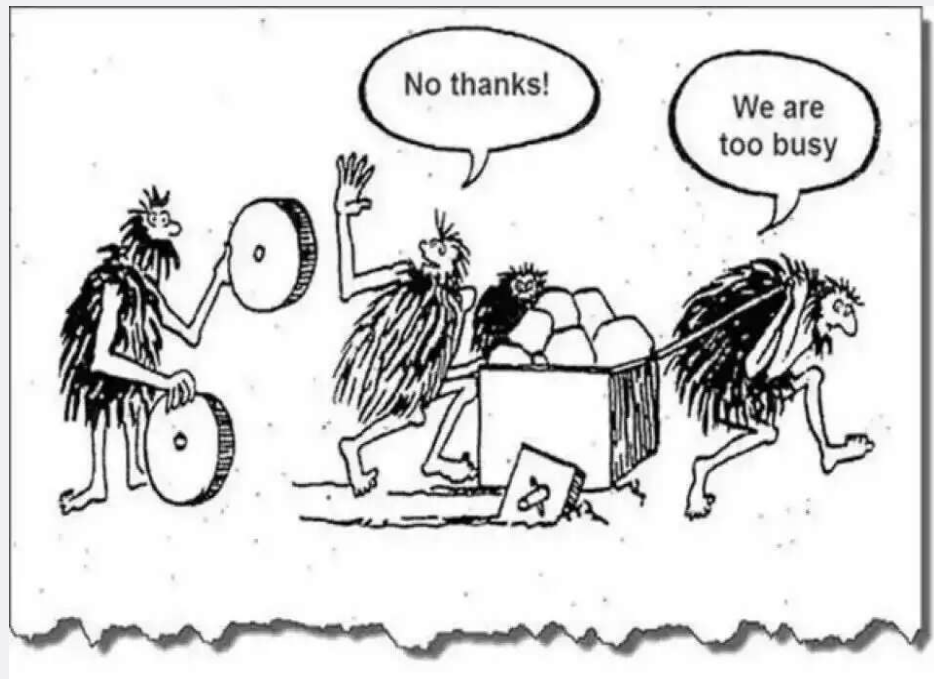
流程主导还是人主导？

颠覆还是改进？

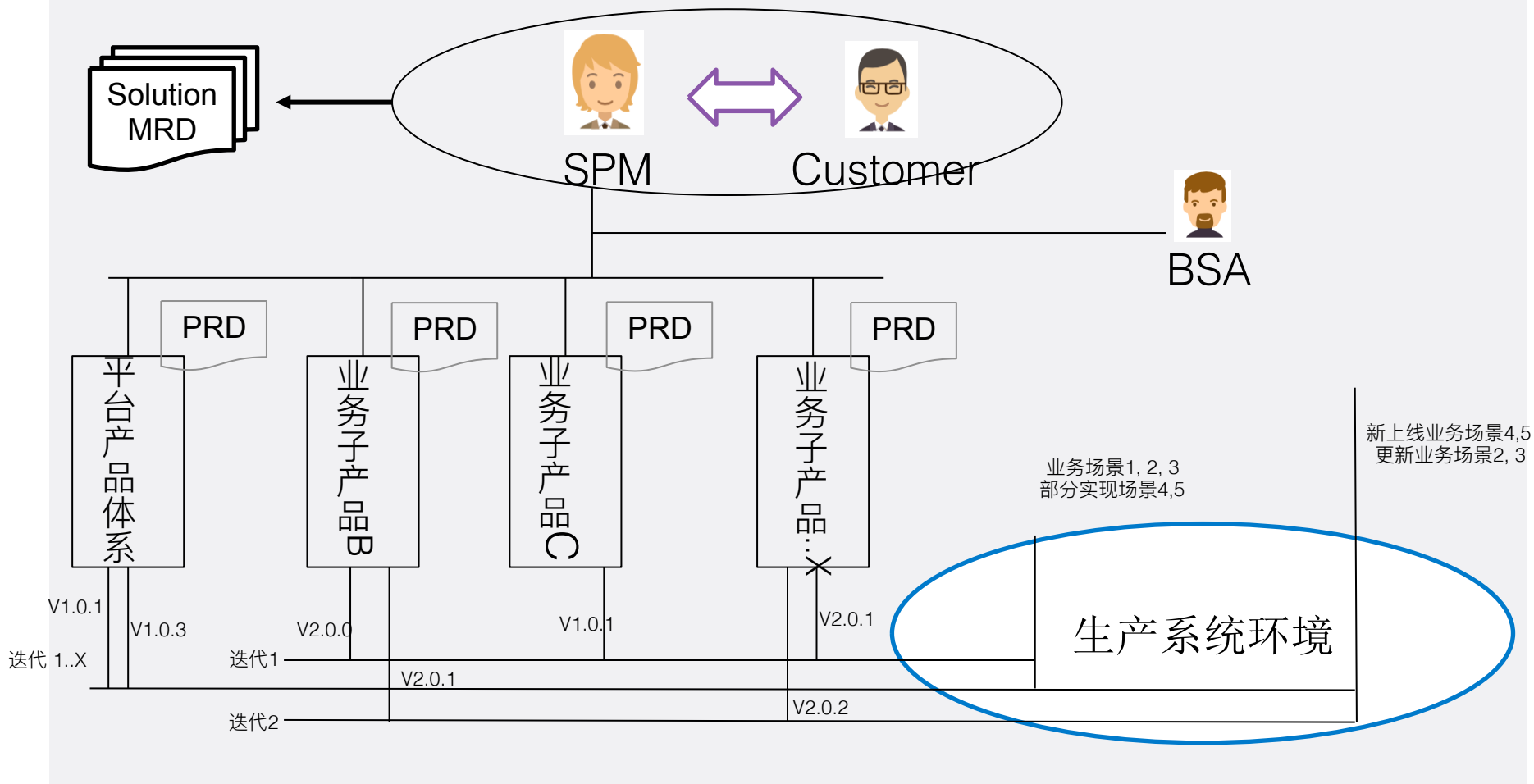


不以敏捷开发为基础的DevOps都是耍流氓

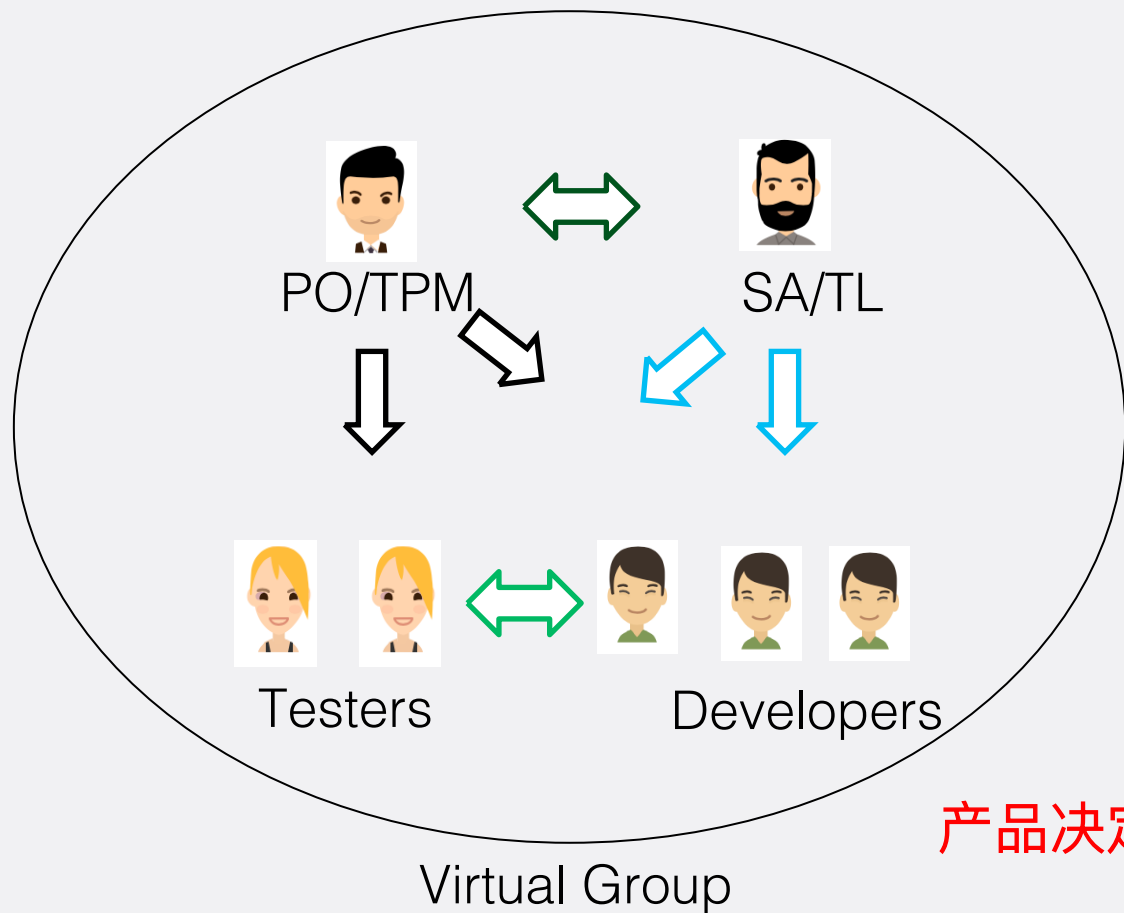
SCRUM != DevOps



自运营/服务模式 Scrum的演进



产品Scrum Team的构成(1-1-3-2)



↔ 范围沟通

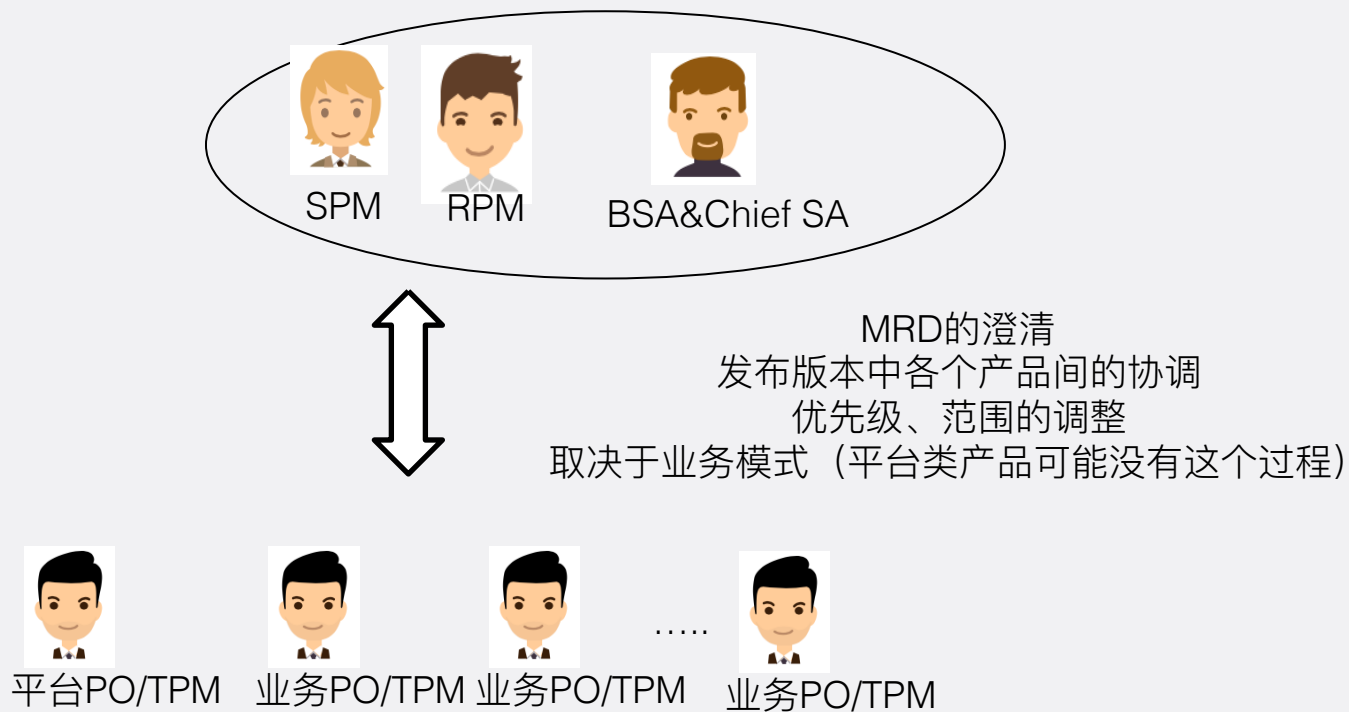
➡ 需求澄清

➡ 实现指导

↔ 开发与测试相互驱动

产品决定做什么，架构决定怎么做！

Scrum 2 Scrum



更多是针对B2B类的业务

02.角色分工

Scrum的构成于职责 – PO/TPM



PO/TPM

PO/TPM主要职责：

- 根据MRD(平台类产品可选)来分析和定义PRD并设计解决方案
- 竞品分析、外部数据收集，自我提出制定产品提升需求
- 并规划产品的Roadmap，制定版本计划
- 选定每个迭代的Scope，对团队内澄清需求
- 组织planning game, 组织产品开发中特性的Demo，组织验收每个版本的交付。组织回顾会议，总结和提升团队能力。
- （平台类）收集业务产品反馈，推动业务对于新特新的使用
- （平台类）运营产品，收集产品运行数据，支撑业务日常使用的方案支持等
- 与架构师配合，规划产品的技术改进
- 。 。 。

PO: Product Owner

TPM: Technical Product Manager

不想做好测试的开发不是好的PO

Scrum的构成于职责 – SA/TL



SA/TL

SA/TL主要职责：

- 产品的技术实现架构的设计
- 指导开发成员的开发
- 指导测试成员的测试设计
- 驱动开测试自动化
- 组织Design Review、Code Review、Test Design Review等
- 与PO配合，规划产品的技术改进
- （平台类）指导业务团队架构师对于产品使用的技术支持
- 。 。 。

Developer



Developer

Developer主要职责：

- 产品的技术实现
- 负责子模块的详细设计
- 负责自动化单元测试以及集成测试的开发
- 对产品的技术持续提升提供建议
- 对测试澄清技术细节、协助测试完善测试设计
- 确保代码即时“可见”：Portal展示/Test Case Pass
- 。 。 。

Tester

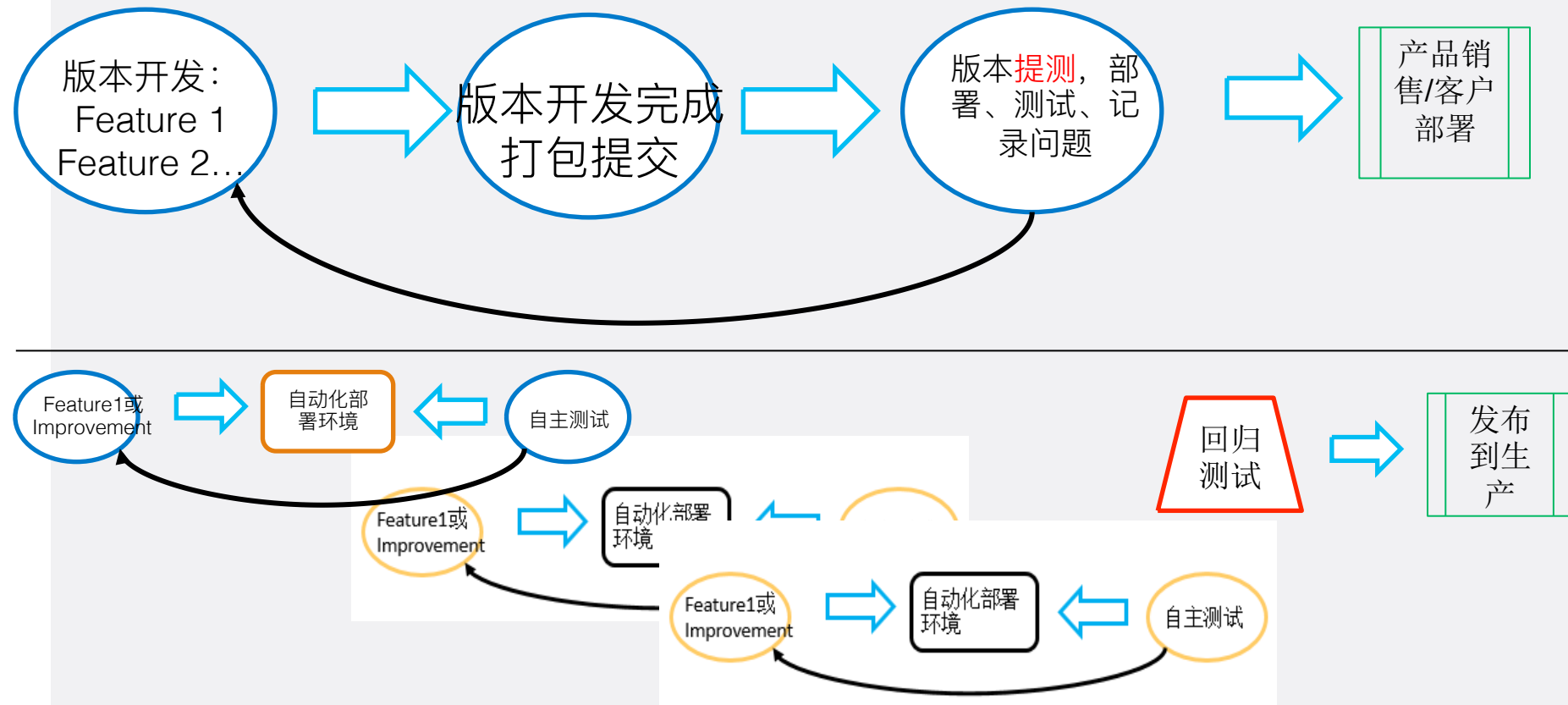


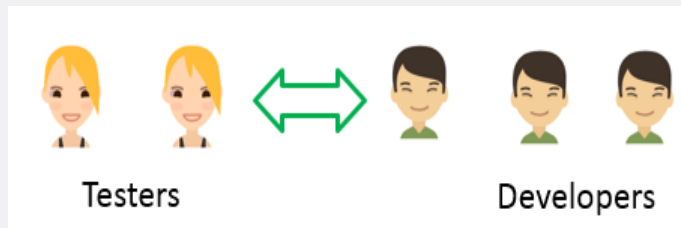
Developer

Tester主要职责：

- 产品的测试设计
- 完善测试自动化工具与流程
- 驱动开发的完善与补充开发思考上遗漏
- 负责端到端的系统级测试、性能测试、稳定性测试
- 为开发定位问题提供便利
- 产品的发布执行
- 。 。 。

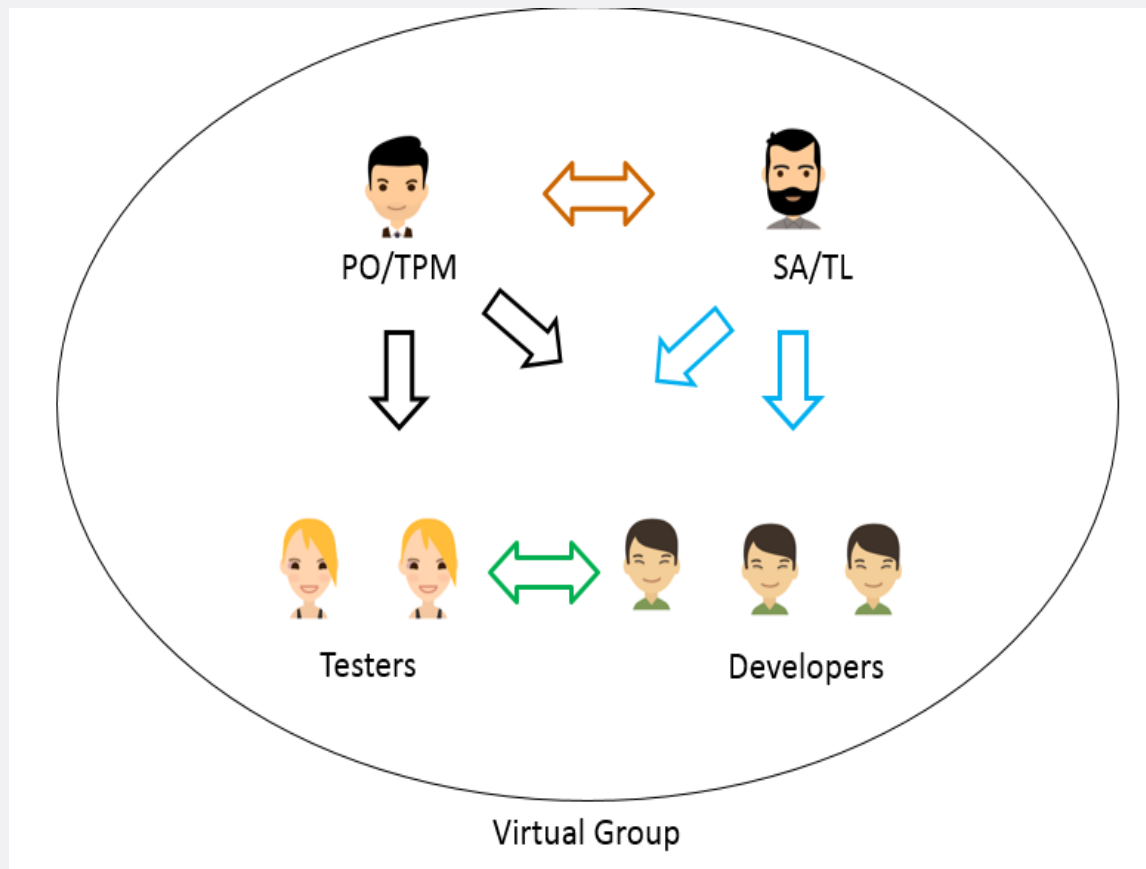
开发与测试的配合 – 模式转变





相互驱动：

- 沟通、沟通还是沟通。
- 开发和测试的信息来源将都来自于产品需求，而不是开发是测试的上游
- 测试的定位的转化。
 - 在Scrum模式下，开发与测试的工作边界将更加模糊。
 - 测试有个最主要的功能是测试点的设计而不是通过复杂的重复性工作来体现测试的价值。
 - 测试应该更好的思考如何提供便利的工具。让定位问题和更加方便
- 开发则应该配合测试一起详细review测试点的设计，并从而思考开发中潜在的问题。



03.团队的日常活动

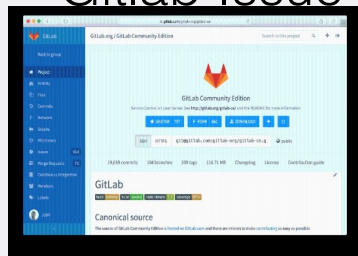
Planning Game

1. 规划版本
2. 规划迭代内容
6. 持续跟踪调整



PO/TPM

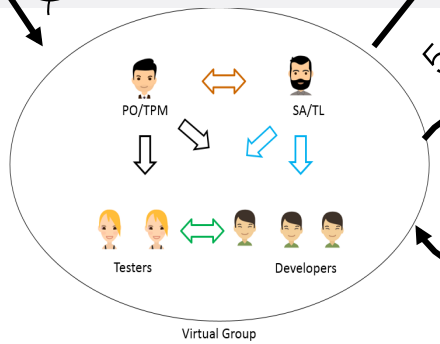
Gitlab Issue



工作白板



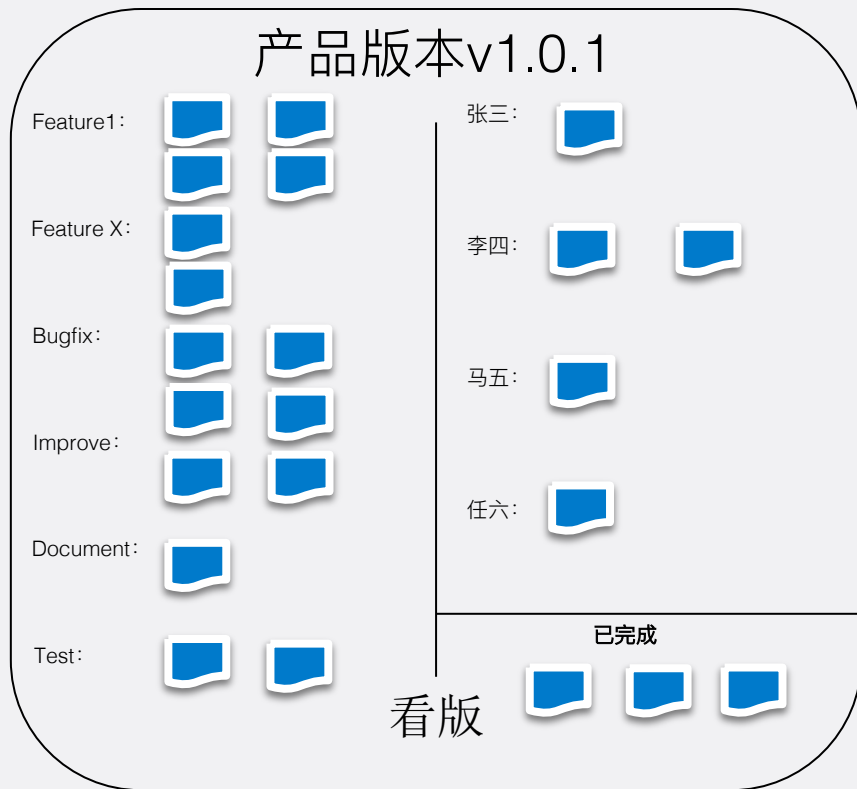
5. 更新迭代内容

4. 团队评估工作量
达成协议

Planning Game的经验:

- 不要一次性选择多个大的 feature 在一个版本中
- 选择一些小功能或改进性功能作为风险控制的缓冲
- 预留技术改进的工作项
- 以版本维度作为计划
- 一个版本中以2个开发迭代（2周）为好
- 需要做发布到生产的，预留一周做发布缓存
- . . .

Standup Meeting



站会的经验：

- 尽可能简短(15min内)，每个人只讲完成情况和遇到的问题，不具体讨论方案
- 需要特意申明需要配合的内容
- 额外增加的内容，需要立即贴出来并由PO判断是否需要做
- 自主领取新的内容，如果有工作依赖或难度的内容，需要TL协调
- 风险需要立即提出
- 尽可能做完一个任务在领下一个任务
- 如果发现任务时间过长，需要做分析和拆解
- 如果觉得任务有难度，开发应该提出要求TL给出设计
- ...

burn down chart?

Review Meeting

随时召开各类Review Meeting：

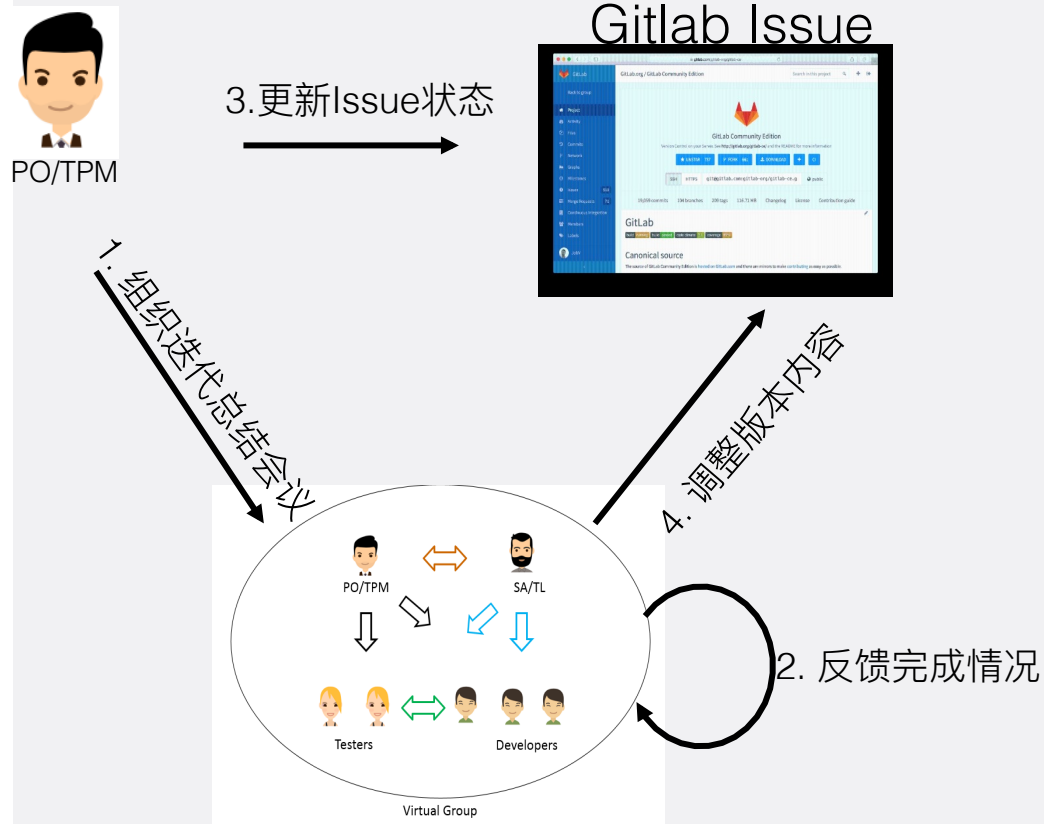
- 由各个主题的Owner自主组织
- 包括：Design Review, Code Review, Test Design Review, Retrospect Review
- 对于比较大或比较复杂的设计可以分阶段Review: 1/3->1/2->final
- Review一定要有最后的完整输出
- 时间长度一定要控制，1个小时为宜，时间过长应该分多次
- 。 。 。



Demo会的经验：

- 目前是团队对于输出的理解是否一致，是否满足PO的想法
- 不要局限在一定完整测试完，只要能端到端跑完场景即可
- 不要局限一定在版本交付前，而是每个feature为维度，时间可长可短，越早越好
- 不要局限一定要有界面，跑测试代码也是一种Demo
- 不是去挑刺，也不是去测试
- ...

Definition of Done (DoD Meeting)



DoD的经验:

- 以迭代为维度作为DoD
- 版本中的第一个迭代的DoD要为下一个迭代做好风险控制, 及时调整版本范围
- 不要只关注代码是否已经完成, 要同时关注自动化程度、代码质量程度 (Jenkins状态, Sonar状态), 文档情况 (Doc, Wiki), 测试结果, 是否已经CodeReview, 是否已经做了代码清理工作(如扫描ToDo)等
- ○ ○ ○

Retrospect Meeting



Retrospect的经验:

- 需要先回顾上一次回顾会的执行情况
- 每项内容每个成员各写3个认为是重点的内容，写在便利贴上
- 不是批斗会也不是拍马屁会
- 需要总结后制定Action Plan
- 尽可能把氛围调整在一个轻松的环境如咖啡屋
- 频率可以是两个版本一次(8-10周)
- 。 。 。

好的，需要保持的

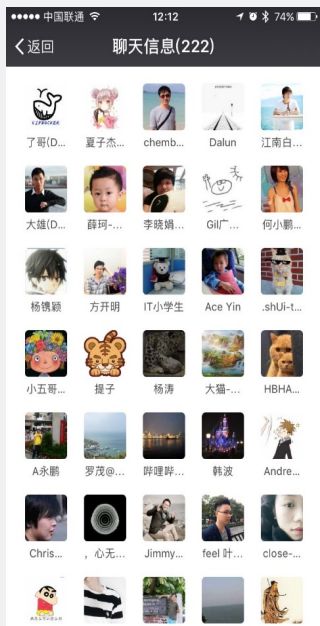
3

不够，需要
提高的

3

遗漏，没做好的

3



用户支持群



运营数据分析



对内对外分享

04.PO的工具

迭代版本范围的选择原则

以一句话概括这个版本的价值和目标！

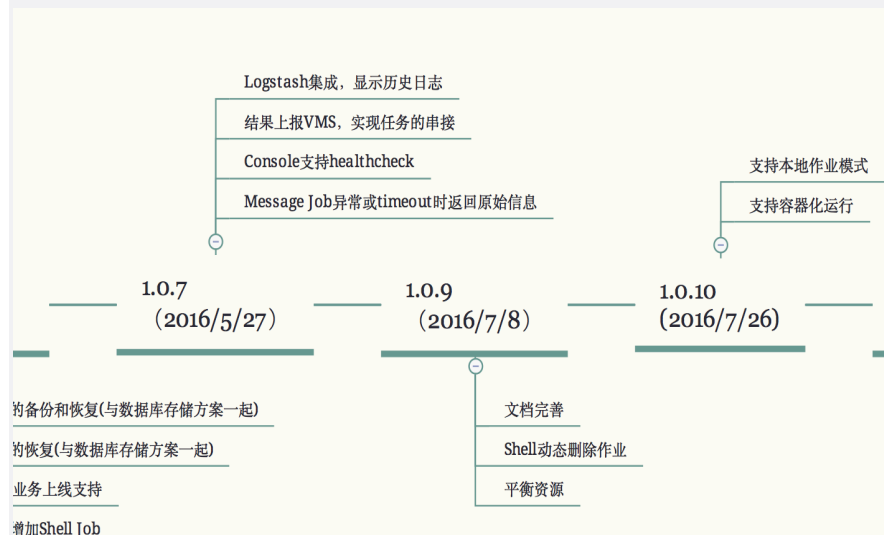
满足物流系统容器化部署，节约机器数量以及快速扩容

调度系统v2.0.1

满足业务在无需升级框架的基础上，提供按照来源IP的方式灰度上线的能力

服务化框架v2.5.1

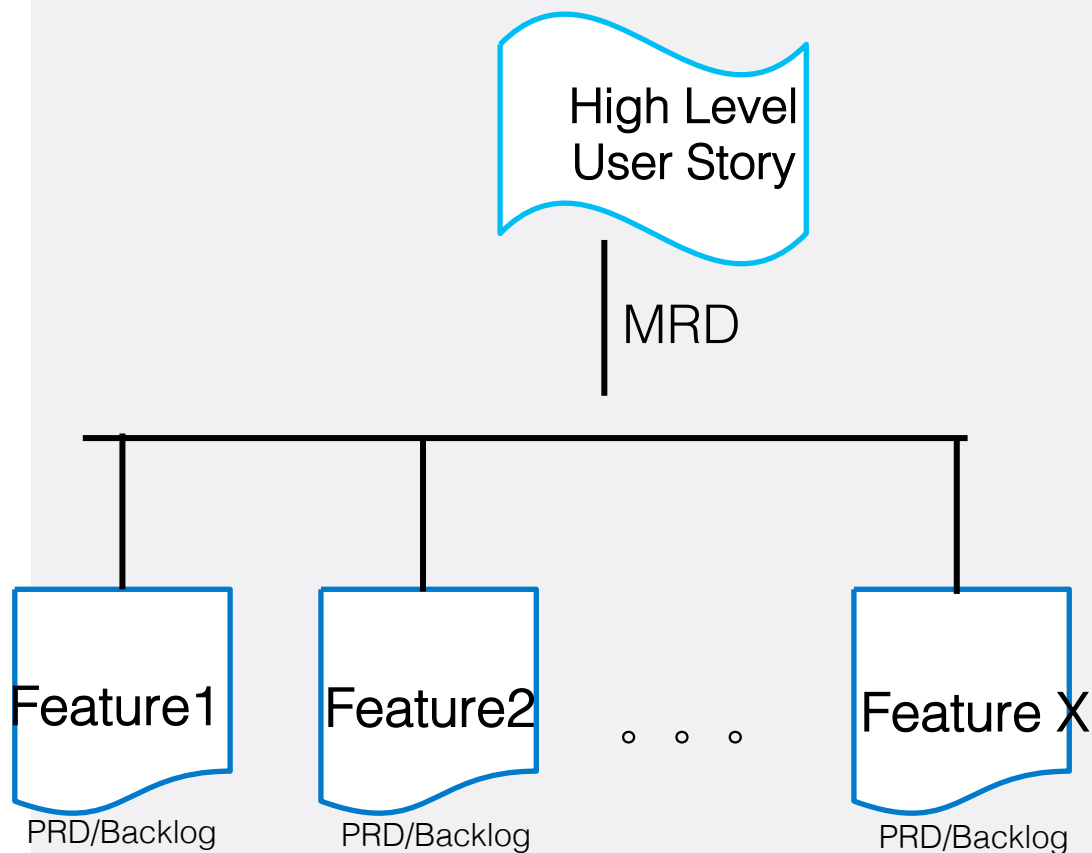
版本细节定义(xmind)



定义版本的经验:

- 需要在团队内充分的沟通, 并达成一致
- 团队成员充分理解版本内的具体内容
- 需要有配套的运营思路/计划 (如平台产品推动)
- ○ ○ ○

Backlog/PRD管理 (Wiki)

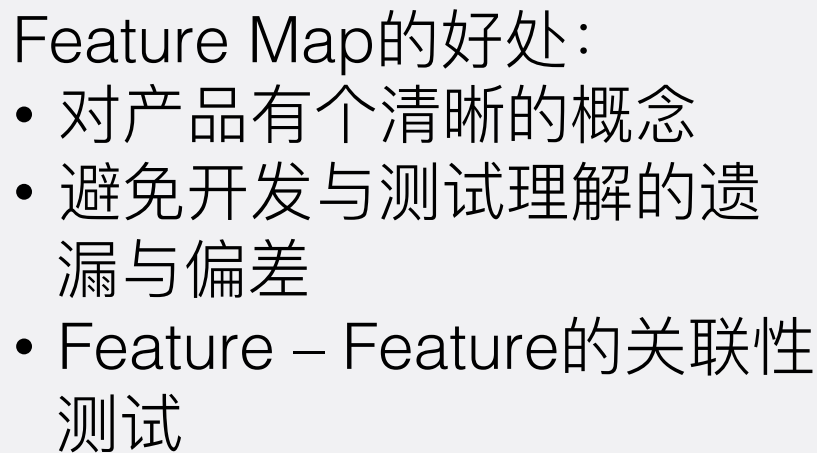


Backlog的概念：

- Backlog是MRD的细分和拆解，特现在Feature描述的颗粒度
- Feature不等同于Task，但是依然能端到端演示
- Backlog需要定期与SPM Review，在优先级上达成一致(最好在版本迭代开始前)
- . . .

Backlog的定义：

- id: 需求编号，便于追踪
- title: 标题
- description: 需求详细描述
- priority : 优先级
- feasibility study: 是否需要需求分析
- solution: 是否需要架构 / 方案设计
- review: 是否需要评审需求分析或方案
- status: 当前状态
- version: 计划的实施的版本




任务拆解与管理 (Gitlab)

<input type="text" value="Search or filter results..."/>	Last created ▾
Comp Calculator rounds too strongly #1204 · opened 2 days ago by Ernst van Nierop	2 updated 2 days ago
Get all reps to a point where they can confidently deliver a sales demo #1203 · opened 2 days ago by Amara Nwaigwe Sales Enablement 0 of 8 tasks completed	5 updated 2 days ago
Add audit user to EE Premium features #1201 · opened 3 days ago by Mike Bartlett GitLab EE	0 updated 3 days ago
Blog post - Research study on issue boards #1200 · opened 4 days ago by Taurie Davis blog post	0 updated 4 days ago
Project deletion takes weeks #1199 · opened 5 days ago by Matthew Prasinov	0 updated 5 days ago
Send event on account signup to Google Analytics #1197 · opened 6 days ago by Mitchell Wright	0 updated 6 days ago
Better search tool on the handbook? #1196 · opened 6 days ago by Ernst van Nierop	7 updated 2 days ago
Demo: CI/CD with GitLab #1195 · opened 6 days ago by Rebecca Dodd Mar 10, 2017 blog post	2 updated 5 days ago
Add Buy Now Button to Home Page #1194 · opened 6 days ago by Mitchell Wright UX Website Redesign	2 updated 5 days ago
Add import/export to GitLab vs GitHub comparison page #1193 · opened 6 days ago by Sean Packham (GitLab)	0 updated 6 days ago

基于Gitlab Issue的任务管理与Bug管理：

- 轻量级管理模式
- 任务拆解成2-3颗粒度的Issue
- 将任务归类
- 将任务归版本
- 与代码关联
- 直接团队讨论(需求细节澄清、问题日志...)
- ○ ○ ○

Saturn 2.0.4

 IvyLee-cn released this 18 days ago · 1 commit to master since this release

New Features

- [#91](#) add group attribute for jobs which can be used for group filter
- [#90](#) add job dependency attribute for jobs and do dependency check when starting/stopping

Improvements

- [#92](#) add Executor version information in the registry page

Bugfix

- [#96](#) fix ConcurrentModificationException when stopping Executor
- [#83](#) fix IllegalMonitorExcetion when handling job timeout strategy
- [#94](#) fix concurrency issue of sharding content when enabling hundreds of jobs

- 任何一个子系统都是一个独立的产品，无论大小
- 任何一个产品，都应该有自己的生命周期和版本规划
- 任何一个产品的发布不应该强依赖其他产品

内部版本定义规则

2

大版本号，不
向下兼容

1

Feature版本
号，向下兼容

8

Patch版本
号，向下兼容

-RCx/Snapshot

测试标识，用于回归测
试、性能测试等

总结

人是活的，流程
是死的

没有明文禁止
的，都是应该可
以改变的

