



华为敏捷项目管理实践

王明兰 华为精益高级专家

LEADING NEW ICT



明兰 华为精益高级专家；知名敏捷&精益咨询师，中国最早的精益看板国际认证教练（KCP）&看板认证培训师（AKT）；企业规模化敏捷认证咨询师（SPC4）。

“快鱼吃慢鱼”时代传统研发模式正面临巨大挑战

人工智能



大数据



社交



移动



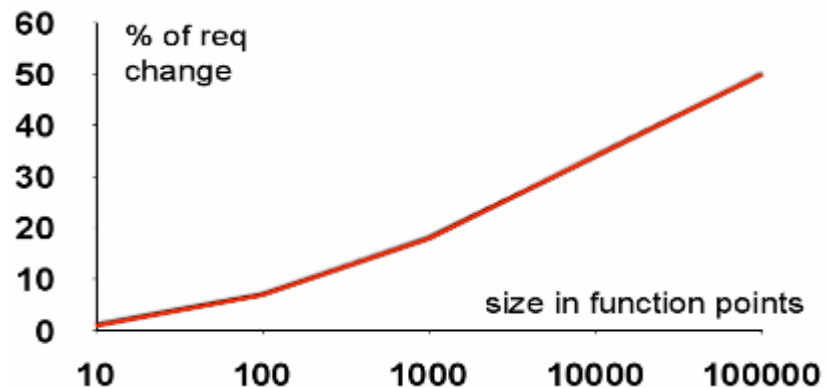
云计算



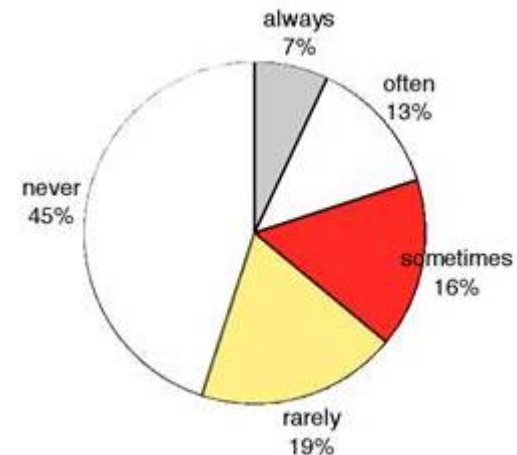
物联网



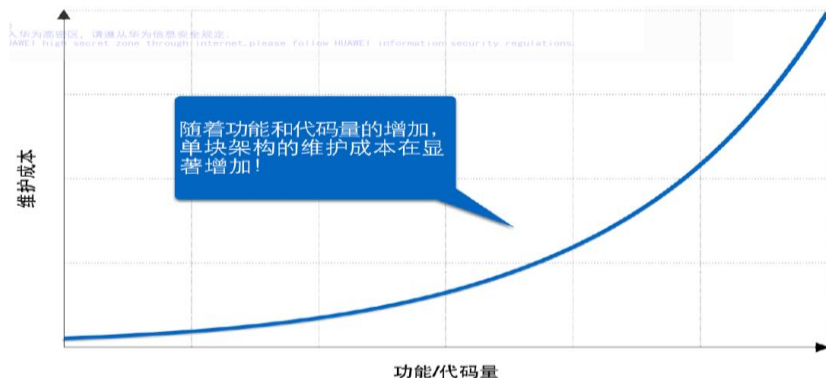
随软件规模增长，需求变化呈非线性增长



45%特性从未被使用，35%很少使用



随软件规模增长，维护成本呈非线性增长



软件开发是复杂不可预测的经验控制过程

Start with Goals and some priority requirements



麦当劳是简单可预测生产过程

Start with Plan and all requirements



业界（20年一轮回）

软件作坊

20世纪60年代



软件危机

70年代



华为（8年一代）

1998年之前

游击队

软件过程控制

80年代



重型控制

90年代



1999~2007年

正规军

敏捷

2001年~今



精益

Devops

2008年~今

特种兵

华为从重型IPO转向DevOps敏捷开发模式

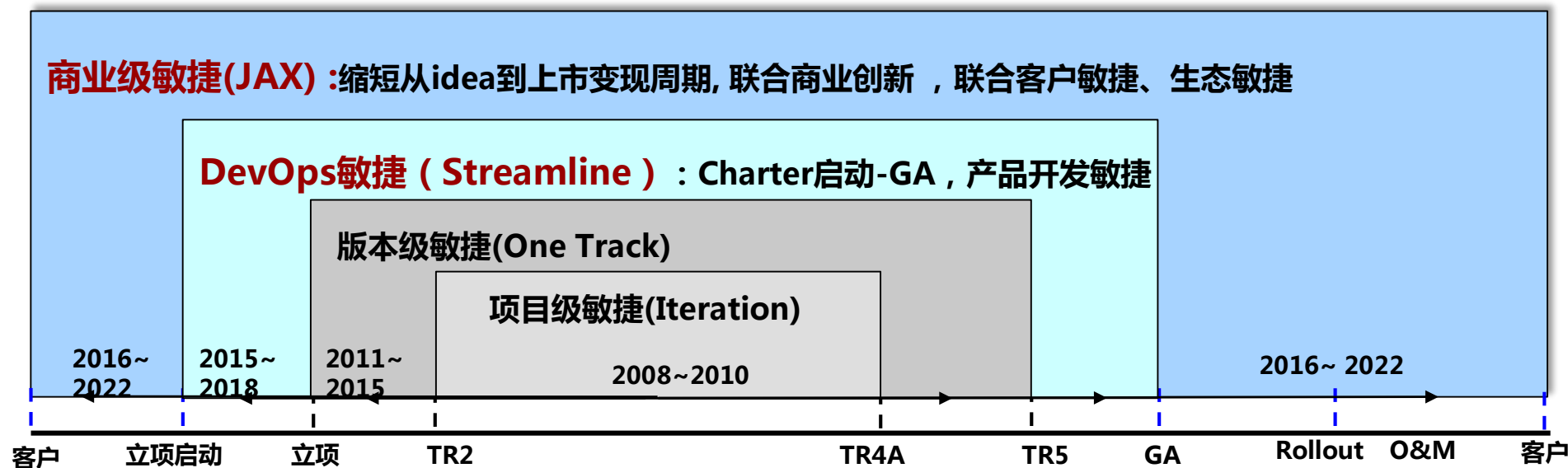
研发变革都是业务驱动，有相应背景，都需要解决某类特定问题：

IPO&CMMI - - 使得华为穿过了青纱帐，从土八路走向了正规军

项目级敏捷 - - 尝试了迭代开发，增强了基础工程能力

版本级敏捷之One Track - - 减少并行开发和维护，减少浪费，提升质量

产品级敏捷 - - 缩短客户界面可感知TTM



- 软件更像一个活着的植物，软件开发是自底向上逐步有序的生长过程，类似于植物自然生长
- 敏捷开发遵循软件客观规律，不断的进行迭代增量开发，最终交付符合客户价值的产品

敏捷宣言

我们正在通过亲身实践以及帮助他人实践，揭示更好的软件开发方法。通过这项工作，我们认为：

个体和交互	胜过	过程和工具
可以工作的软件	胜过	面面俱到的文档
客户合作	胜过	合同谈判
响应变化	胜过	遵循计划

虽然右项也具有价值，
但我们认为左项具有更大的价值。

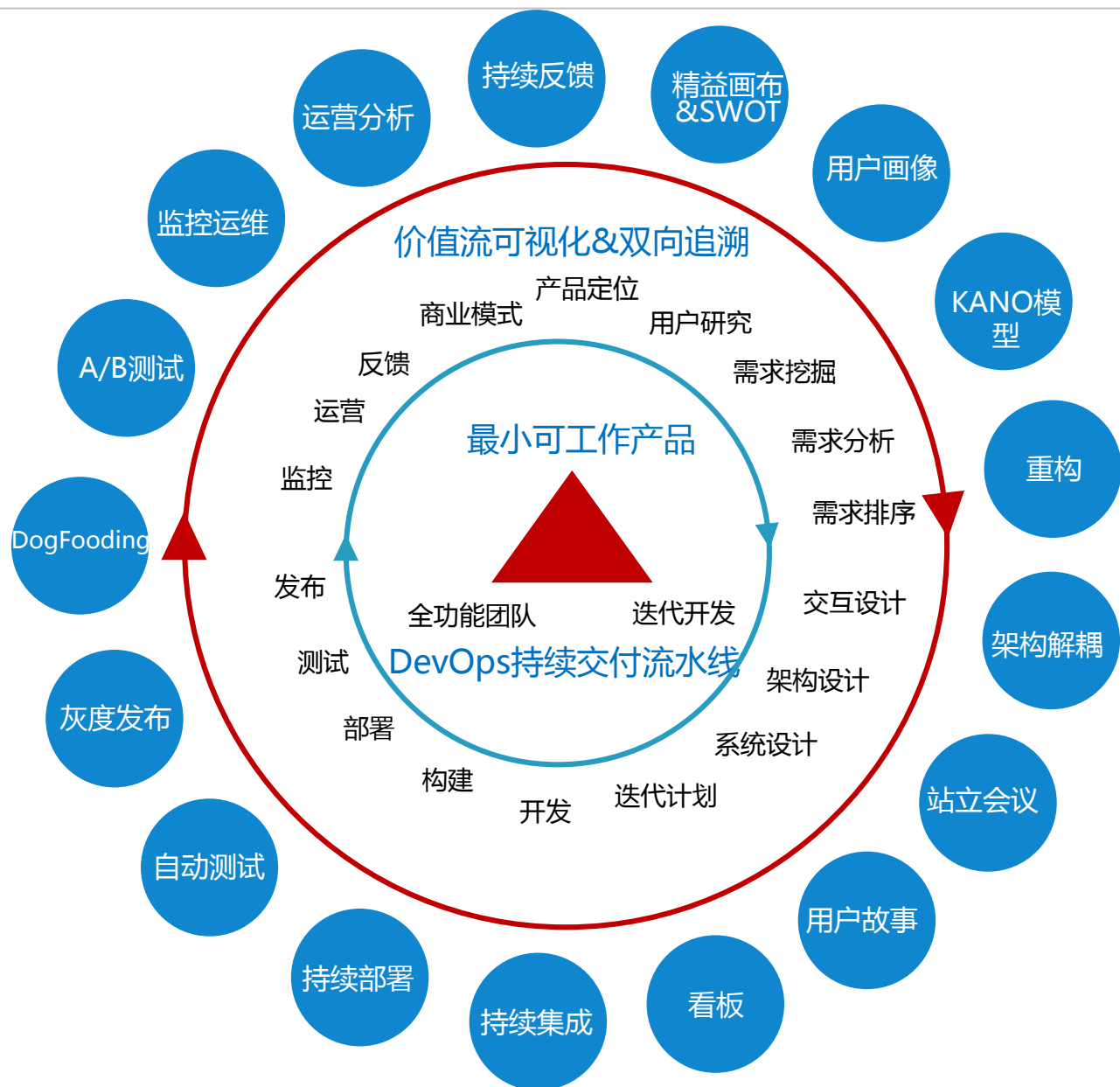
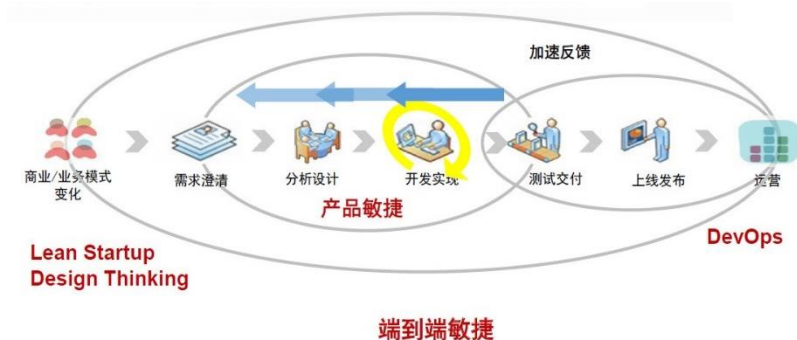


敏捷开发



华为敏捷是一种端到端敏捷

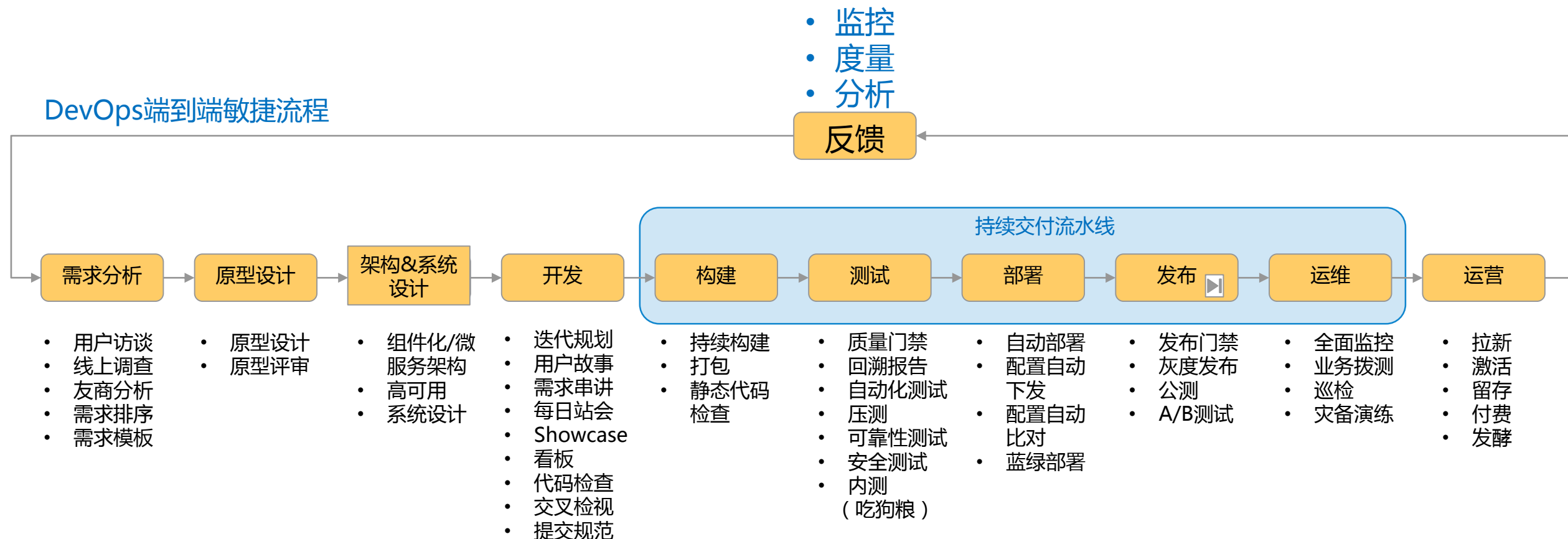
- 华为敏捷项目管理，融合了敏捷、精益、DevOps理念，不只是开发阶段的敏捷，而是从市场，到开发、运维、运营的端到端敏捷。
- 华为1217敏捷项目管理模型：
1个铁三角、2个端到端交付环、17个实践。

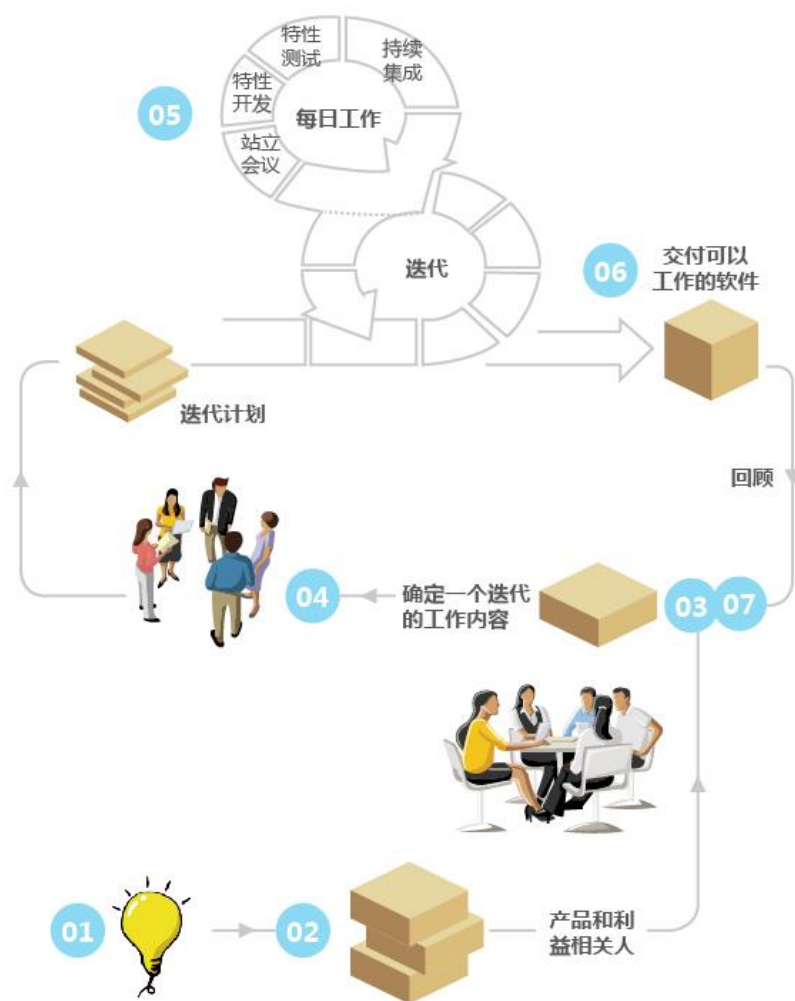


端到端敏捷实施依赖于全自动化的DevOps持续交付流水线

敏捷团队可以将精力聚焦于业务的分析、设计、开发，后续的构建、测试、部署、发布、运维均由自动化工具实现持续交付与反馈。

DevOps端到端敏捷流程





区别点：

1. 传统敏捷模式强调持续构建CI（持续构建、测试）
2. 融合了DevOps的新型敏捷模式强调持续交付CD（包含持续构建、测试、以及持续部署、发布、反馈）。加速了产品推向真实用户，并及时获得海量反馈的决策数据源，比仅仅依靠PO或者少数粉丝用户反馈的决策准确率大为提升。

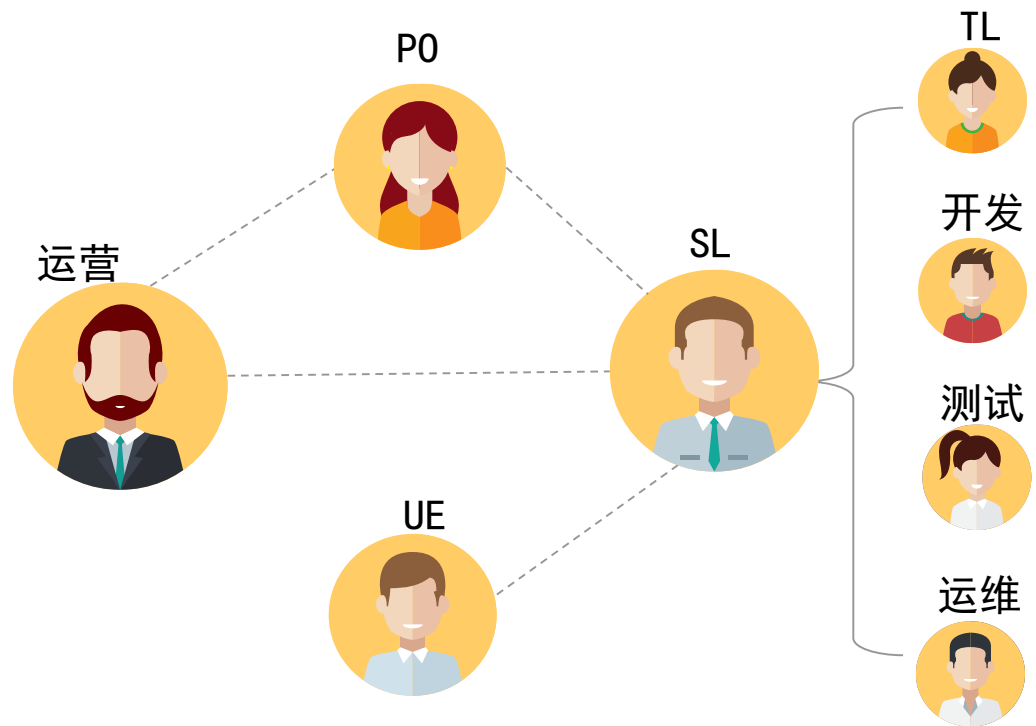
敏捷4阶段循环图，包括准备、计划、开发、回顾：

- ① PO和开发团队对产品业务目标形成共识
- ② PO建立和维护产品Backlog并进行优先级排序（需求会不断新增和改变）
- ③ PO在每轮迭代前，Review需求列表，并筛选高优先级需求进入本轮迭代Backlog
- ④ 开发团队细化本轮迭代需求，并按照需求的优先级，依次在本轮迭代完成
- ⑤ 开发团队每日站立会议、特性开发、持续集成，使开发进度真正透明
- ⑥ PO对每轮迭代（1 - 4周）交付的可工作软件进行现场验收和反馈
- ⑦ 回到第③步，开始下一轮迭代

准备阶段01：组建全功能团队，实现快速自我决策

华为由“集团军作战”转变为“班长的战争”，按照特性/微服务组建<10人的全功能团队，俗称“2披萨团队”，可以实现业务快速开发、决策与上线：

- Sponsor：项目干系人
- PO：Product Owner，负责产品规划、设计、分析
- 运营：负责产品运营
- UE：UCD工程师，负责用户研究、交互设计、美工、视觉
- SL：微服务/特性经理Service Leader，对应敏捷模式Scrum Master，带领团队进行开发
- TL：技术leader，对应于系统工程师SE，负责架构、系统设计
- 开发：负责代码实现
- 测试：负责测试验证
- 运维：负责部署、发布、运维、监控



准备阶段02：DevOps敏捷项目管理平台创建项目环境

1. 华为内部采用DevCloud软件开发云平台，进行敏捷项目管理和DevOps开发
2. 软件开发云自身采用DevCloud on DevCloud自产自用模式，践行“吃狗粮”文化，所有特性自用一个迭代后再上线公网发布。

华为软件开发云对DevOps敏捷项目管理的支撑包括两大服务：

- 包含标准Scrum流程和精简流程两种敏捷项目管理模型，精简流程适合小、微和个人开发者，Scrum流程适合小、中、大型团队。
- 流水线服务可以灵活调度包括代码分析、构建、测试、部署、发布等DevOps端到端服务，实现持续交付。



新建项目

* 项目名称：可以输入中英文，数字和符号(空格，!@#\$%^&*~`-),长度介于3~128之间

* 开发流程：

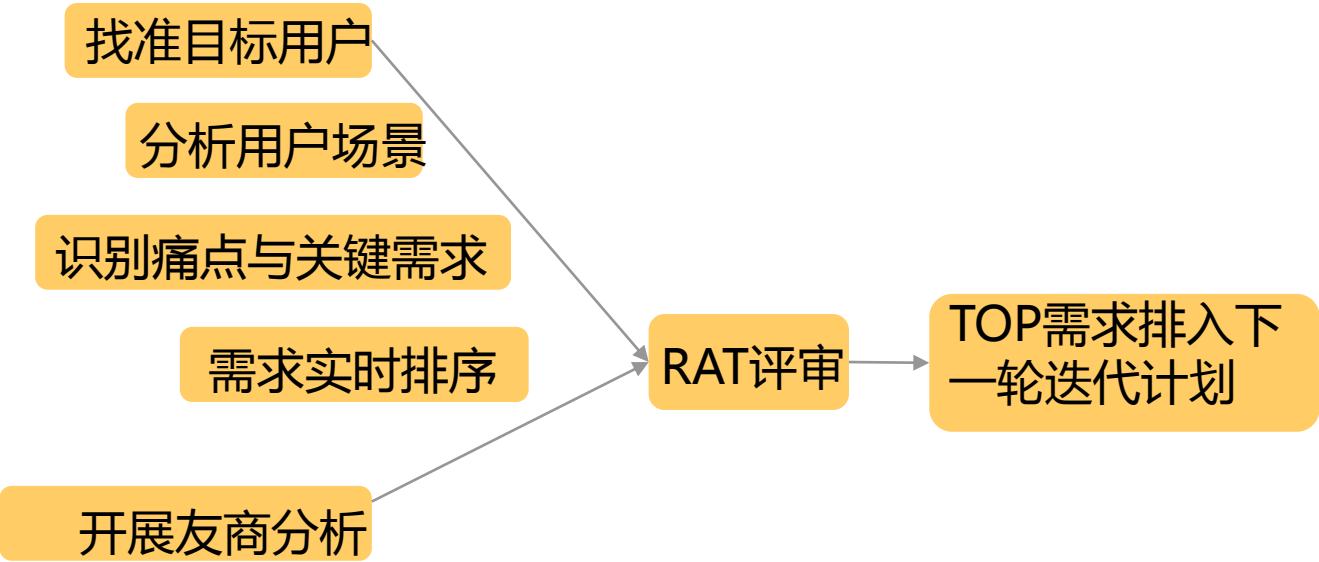


计划阶段01：两级计划是消除浪费的首要环节

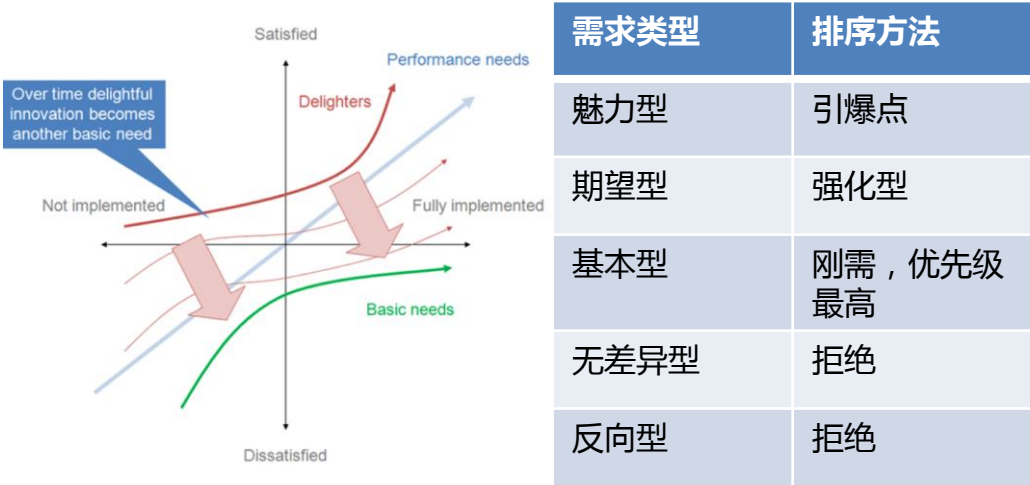
- 计划阶段最核心的做好需求分析，舞好龙头，才能够最大程度消除浪费，交付最小却最大程度满足客户需求的产品。
- 1. 发布计划：基于Epic和Feature优先级排序，整理出来的发布计划。借助精益画布、SWOT、KANO模型辅助梳理。
 - 2. 迭代计划：每轮迭代开始前开展需求重新排序，确保落入迭代的是最高优先级特性

- 注意：
- 特性规划工作量80%，预留20%用于重构及持续改进；
 - 迭代开始前，组织开发团队全员开展需求串讲（对应迭代计划会议）。
 - “交付刚刚好的系统”，不是不精益求精，而是做好取舍，只做客户当前阶段最需要的特性！

需求分析流程：



根据是否必须，以及使用时长*频率排序，并借助KANO模型定义需求类型：



常规版本：

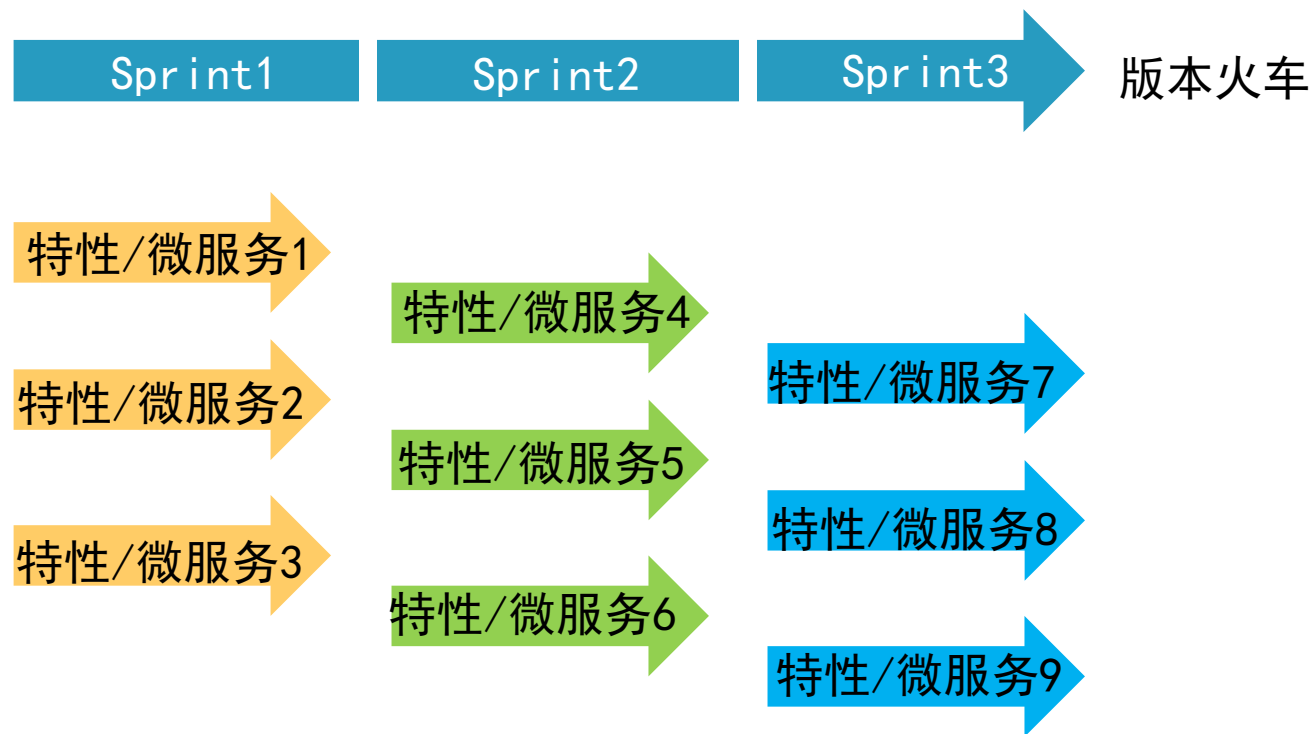
1. 确定合适的版本火车节奏，通常定义迭代周期为1~4周
2. 特性、微服务可随版本火车发布，也可以选择 not 发布，特性有完全的自主权

快速上线：

1. 缺陷、小特性1天上线公网。

注意：

迭代速度受限于DevOps持续交付流水线的自动化程度。如果自动化程度低，而迭代周期越短，团队交付吞吐量会急剧下降，团队容易陷入疲劳作战且绩效极差的境地。



基于迭代需求清单，每轮迭代之初，召集全员，开展需求串讲，让团队所有成员均明确清晰了解为什么（需求场景）、是什么（用户故事）、怎么做（视觉、系统设计方案）。

类型	编号	标题	迭代	模块	处理人	优先级	重要程度	状态	操作
需求	79215	【Scrum】富文本编辑器要跟精简模式保持一致,同时保...	3.0.6	Scrum		高	一般	新建	
需求	79214	【Scrum】工作项页面增加最大工作项条数设置功能	3.0.6	Scrum		高	一般	新建	
需求	79213	【Scrum】点击工作项ID,能新开页面显示工作项详情	3.0.6	Scrum		高	一般	新建	
需求	79212	【Scrum】批量导入工作项功能	3.0.6	Scrum		高	一般	新建	
需求	79211	【Scrum】工作项要有附件功能	3.0.6	Scrum		高	一般	新建	
任务	79144	【DocMan】代码重构	3.0.6	DocMan		高	一般	新建	
需求	79143	【DocMan】欧拉系统切换整改	3.0.6	DocMan		高	一般	已关闭	
需求	79033	【ProjectMan】描述里的富文本框可以全屏、预览和打...	3.0.6	ProjectMan		高	一般	新建	
需求	78687	【ProjectMan】在项目页面里面也可以点关注项目按钮	3.0.6	ProjectMan		高	一般	新建	
需求	78686	【ProjectMan】欧拉系统切换整改	3.0.6	ProjectMan		高	一般	已关闭	

我隐身啦

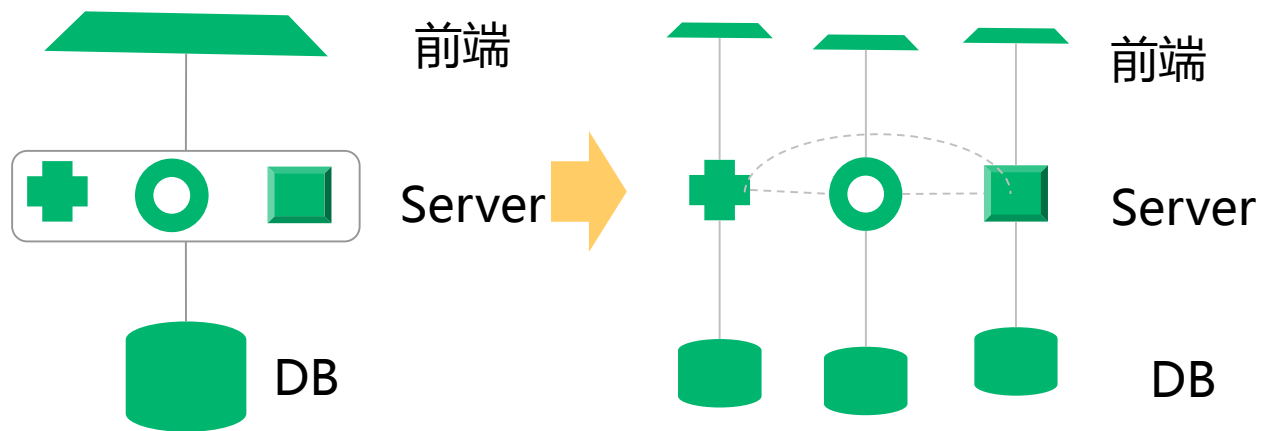
开发阶段02：架构解耦，最小可行产品是敏捷的保障

1. 架构与系统解耦，做到组件化，乃至微服务化。实现松耦合，可并行开发、构建、测试、部署、运行的最小可运行产品/特性。
2. 需求分解两个原则：
 1. 需求分解遵循小步快跑，同一个特性可以由多个迭代Story逐步演进，从简单可用、到功能完善、再到智能化的理想程度。
 2. Story足够小（比如2天），在一个迭代内开发完成。

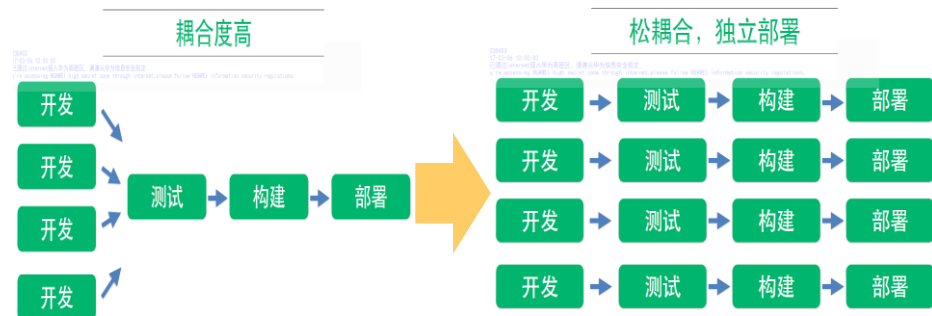
软件开发云团队Story分解：

1. Epic：文档在线托管服务（替代本地文件服务）
2. Feature：文档在线预览->所有格式文档上传->版本管理->在线编辑->断点续传->支持4G视频文件。。。
3. Story：Word系列文档在线预览->POF/TXT->图片->视频->压缩文件。。。
4. Task：由Story开发者个人进行分解

架构解耦：由横向分层的大系统，向纵向解耦的小系统演进。



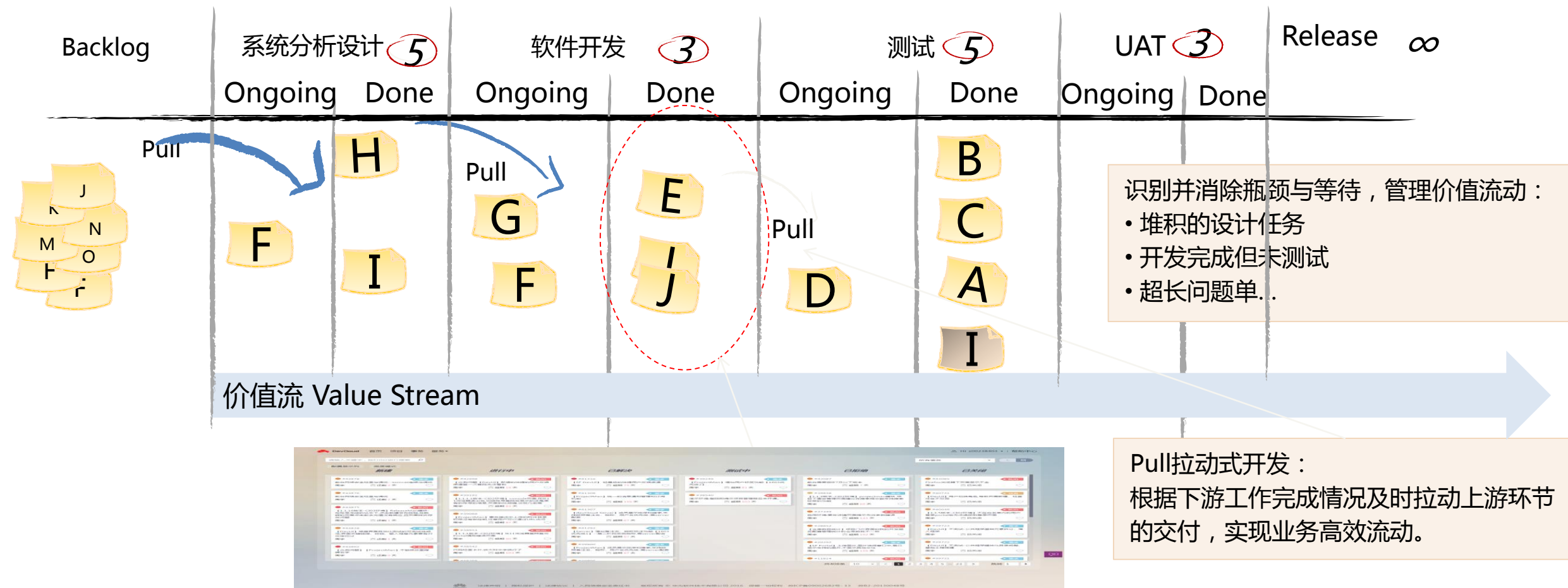
各个微服务/特性，可由独立团队并行开发交付，甚至各微服务技术栈可不一样。



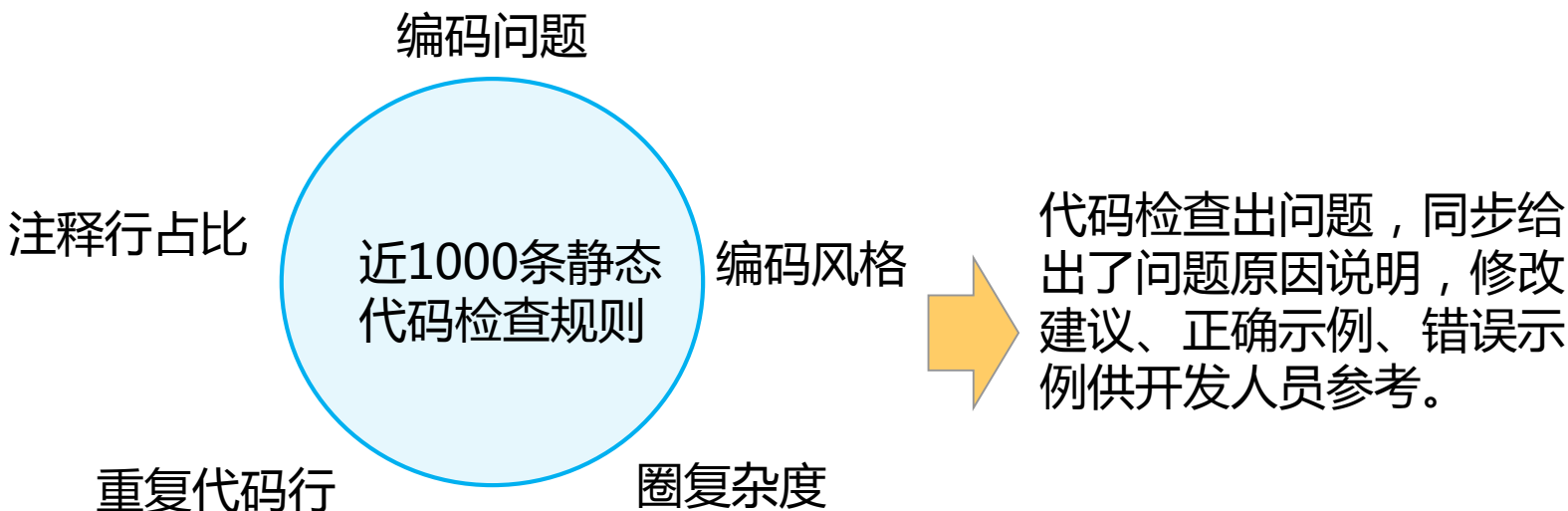
开发阶段03：每日站会，通过Kanban梳理并消除堆积

看板方法:

一种增量和渐进的产品开发和组织运营的方法。它的核心机制是限制在制品数量的拉动式系统，通过它暴露系统流程中的问题，并激发协作以改进系统。



- 通过持续的自动化静态代码检查，确保所有开发人员代码质量达标。
- 代码注释、代码提交规范模板，确保源代码可读性和传承性。

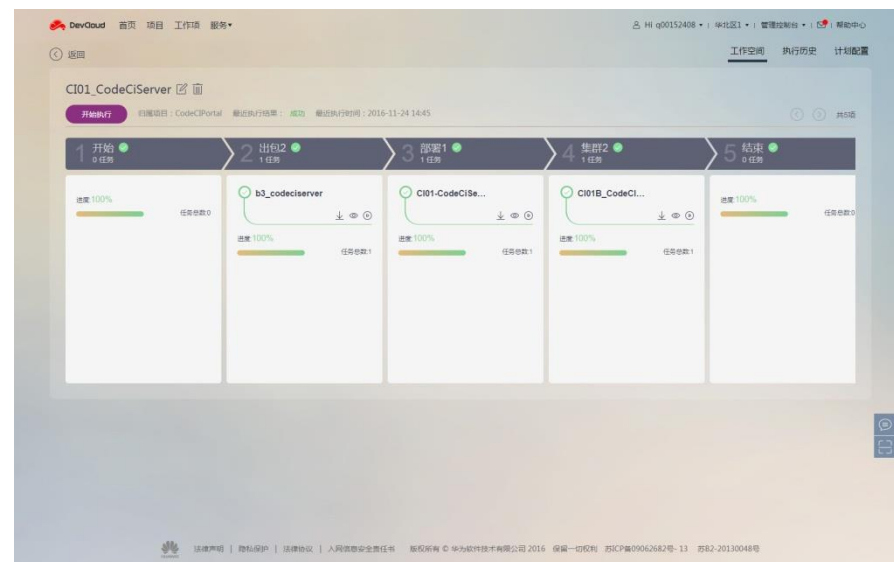


The screenshot shows the IDE interface with a list of code snippets on the left and a detailed view of a snippet on the right. The snippet is a Java method for getting a connection. The right panel shows a "正确示例" (Correct example) and a "错误示例" (Incorrect example) for the same method, highlighting the difference in handling null values. Below the examples, there is a "修改建议" (Modification suggestion) section with a list of rules and their descriptions.

- 流水线支持调度Dev&Ops各环节服务，以及子流水线，实现个人、模块、项目等多级流水线。
- 与传统敏捷模式强调CI持续构建不同的是，融合了DevOps理念的新型敏捷模式，通过云端自动化的持续交付流水线，实现：
 - 持续构建
 - 持续测试（功能、接口、性能、可靠性、安全等自动化）
 - 持续部署（蓝绿部署、脚本自动下发、比对）
 - 持续发布（灰度发布）
 - 持续反馈（AB测试、运营分析），可将Ops端手工操作的时间缩短80%，减少团队等待和修复手工错误带来的等待和浪费
- 全功能团队可以聚焦于业务分析、开发交付及运营上，显著提升效率和产品质量。

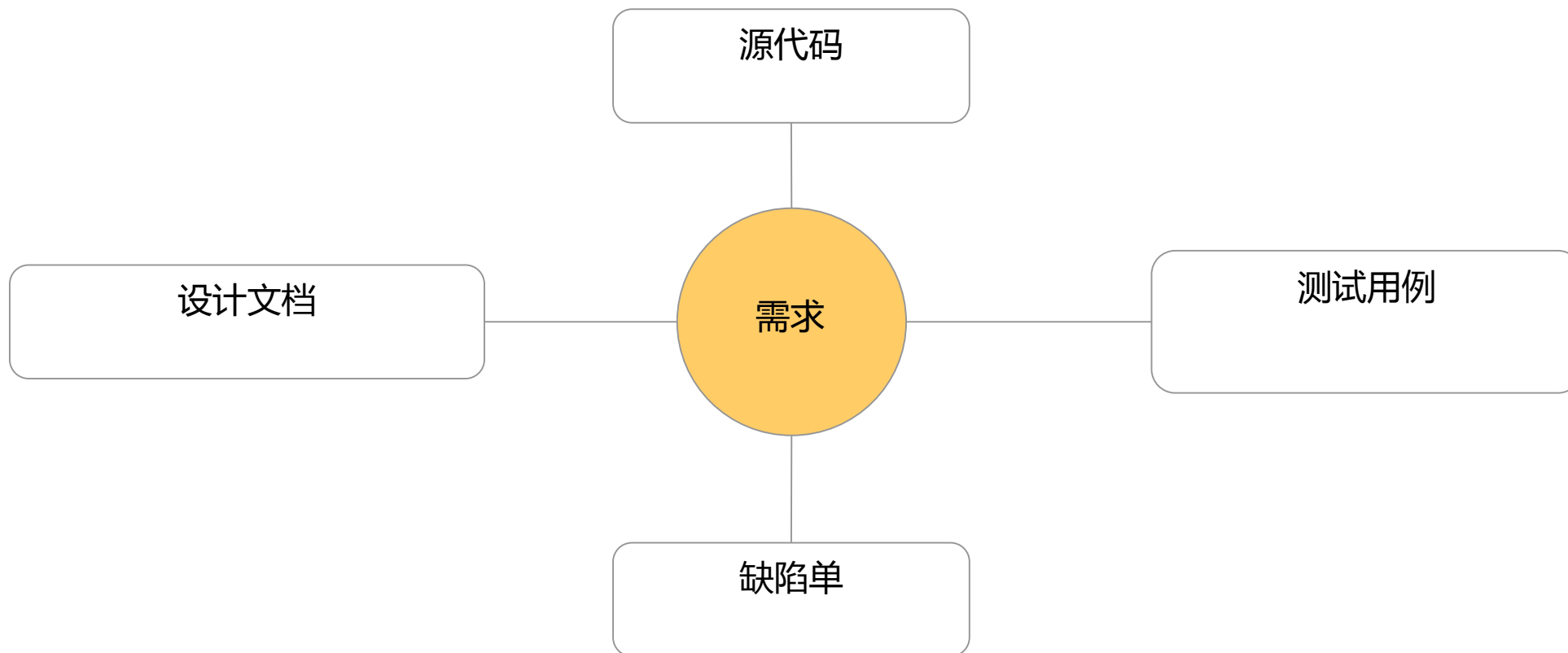


流水线



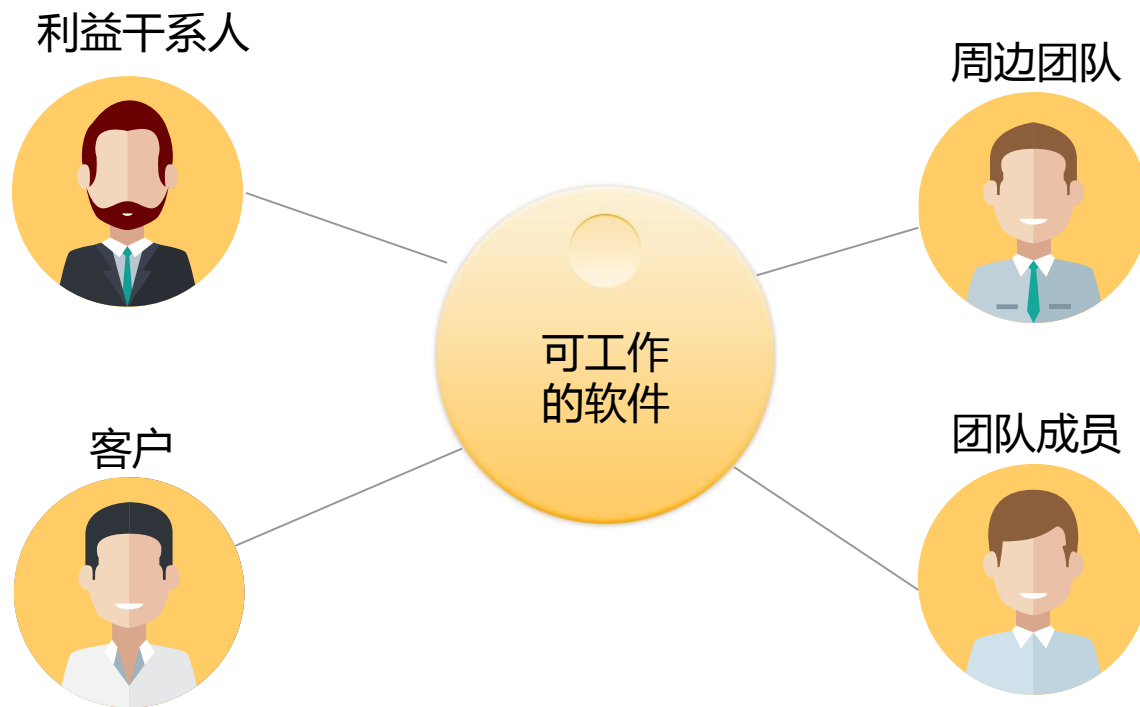
开发阶段06：“需求-设计-代码-用例-缺陷”双向追溯系统

- 低保真、高保真、系统设计文档等材料可作为附件上传与需求关联；
- 代码提交时按照规范备注需求ID，即可实现代码与需求的关联
- 测试用例和缺陷，也需关联需求
- 可针对需求开展讨论，讨论结论可追溯

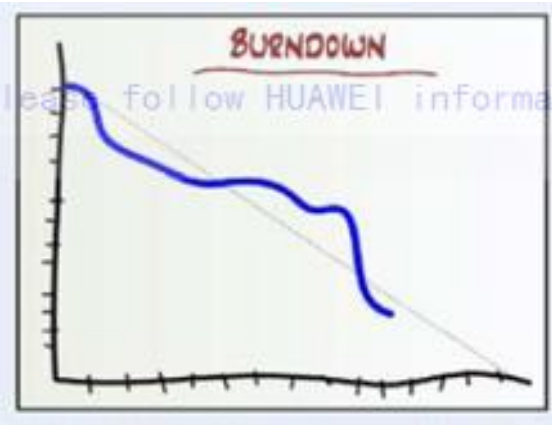


- 每轮迭代转测前集中ShowCase，由产品经理验收，开发、测试、UCD等关键角色参与

注：ShowCase应该是团队内、团队间、周边沟通、客户沟通、利益干系人沟通的主要载体，消除PPT演示，减少材料写作与汇报工作量，提升工作效率。



- 1. 通过迭代需求统计报表和燃尽图，查看需求交付进展。
- 2. 迭代遗留缺陷报告呈现每个模块/服务质量情况。可结合质量门禁，单服务遗留严重及以上级别问题，或者总遗留DI值（遗留缺陷密度）>x分，则服务质量不达标，不允许发布。



项目Story统计 项目Bug统计							
模块	新建 ▾	进行中 ▾	已解决 ▾	已拒绝 ▾	已关闭 ▾	总计 ▾	详情
总计	2	0	0	0	0	2	查看
未配置模块	2	0	0	0	0	2	查看

项目Story统计 项目Bug统计							
模块	关键 ▾	重要 ▾	一般 ▾	提示 ▾	总计 ▾	DI值 ▾	缺陷详情
总计	0	0	1	0	1	1	查看
未配置模块	0	0	1	0	1	1	查看

华为质量回溯会议，对应于敏捷迭代回顾会议：

- 转测试回归不通过问题、现网问题，均需由Scrum Master辅导问题责任人牵头进行回溯，并召开整个团队及涉及该问题的周边团队开展质量回溯会议。会议重点在于分析问题根因，并识别出管理、流程、技术、工具上可落地的改进点。
- 这些改进点每一个都必须符合SMART原则，可落地、可执行，不能出现大话空话套话。
- 问题要求最晚在下一个迭代中执行落地，避免问题再次出现。

质量回溯是华为持续改进的实践瑰宝。其要义是塑造整个团队对事不对人、勇于直面问题、只要有方法有措施下次改进不再重犯错的“从泥坑里爬起来就是圣人”的文化氛围，培养全员持续改进的意识。





THANK YOU

Copyright©2016 Huawei Technologies Co., Ltd. All Rights Reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.