



SCRUM敏捷过程概述



敏捷开发已逐渐成为主流趋势

业界敏捷浪潮



- ISO 9000 (09版) 标准将在原来八大原则的基础上新增**敏捷原则**
- 2000年美国军方软件开发标准（DOD 5000.2）推荐**迭代为软件开发优选模式**
- 世界影响最大的美国波多里奇国家质量奖将**敏捷**作为**核心的十一大原则之一**

敏捷诞生的历史背景



软件开发顺应时代变化，从重型过程转向轻量型敏捷

敏捷宣言揭示更好的软件开发方法

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。



- 敏捷宣言（2001年）是敏捷起源的基础，由上述4个简单的价值观组成，敏捷宣言的签署推动了敏捷运动
- 敏捷宣言本质是揭示一种更好的软件开发方式，启迪人们重新思考软件开发中的价值和如何更好的工作

敏捷更符合软件开发规律

传统开发

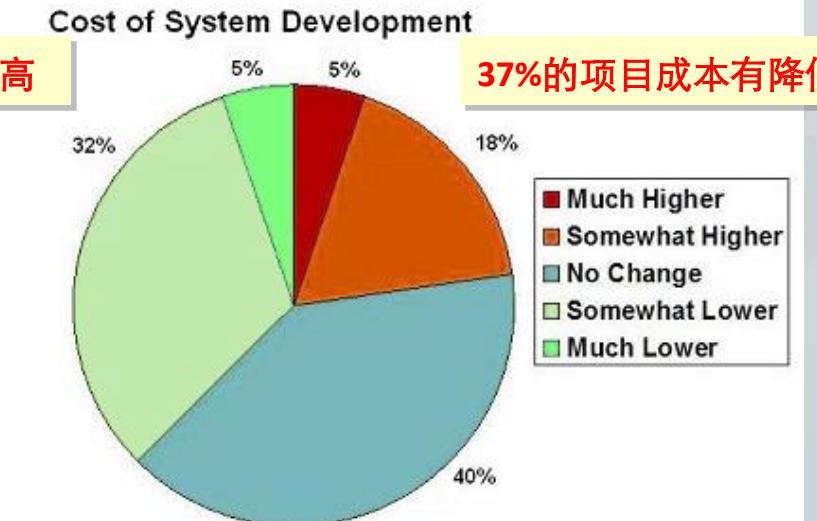
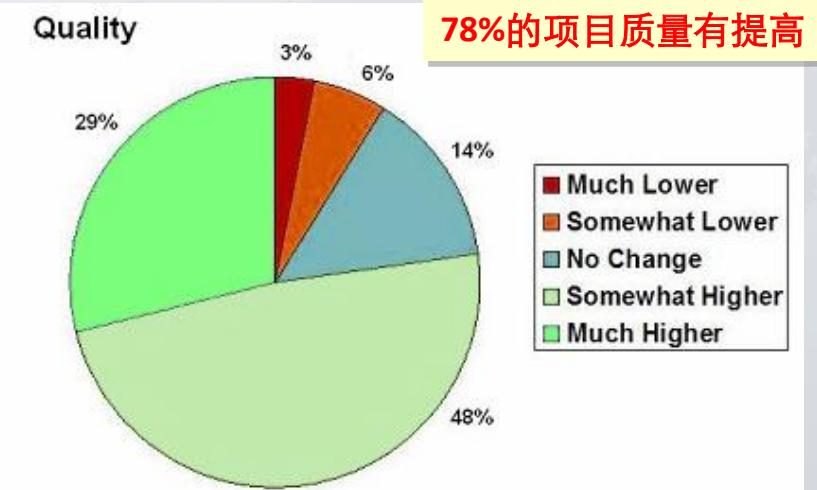
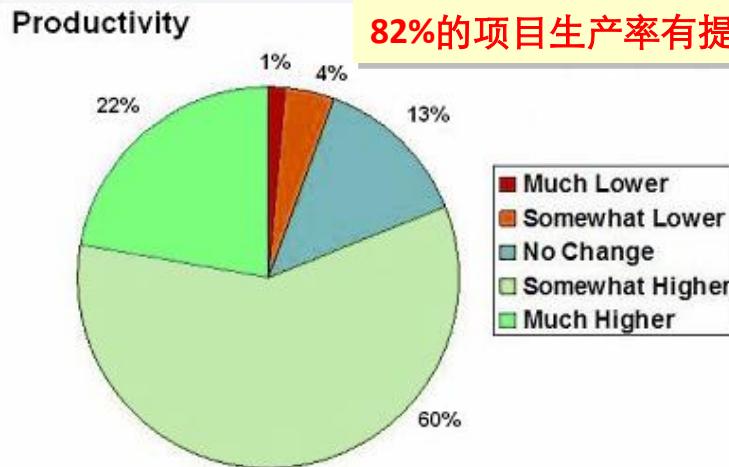


敏捷开发



- 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

敏捷对生产率、质量、满意度、成本有明显改进



Source: Dr. Dobb's Journal 2008 Agile Adoption Survey

Copyright 2008 Scott W. Ambler

* 以上数据来自DDJ 2008由Scott Ambler发起的网上调查结果

CHAOS RESOLUTION BY AGILE VERSUS WATERFALL

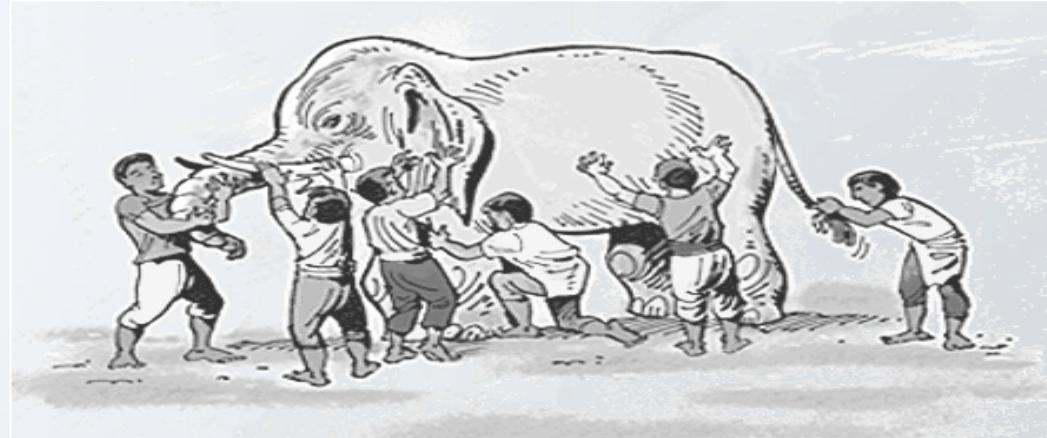
采用敏捷与传统开发的成功率对比

SIZE	METHOD	SUCCESSFUL	CHALLENGED	FAILED
All Size Projects	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Large Size Projects	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small Size Projects	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

The resolution of all software projects from FY2011–2015 within the new CHAOS database, segmented by the agile process and waterfall method. The total number of software projects is over 10,000.

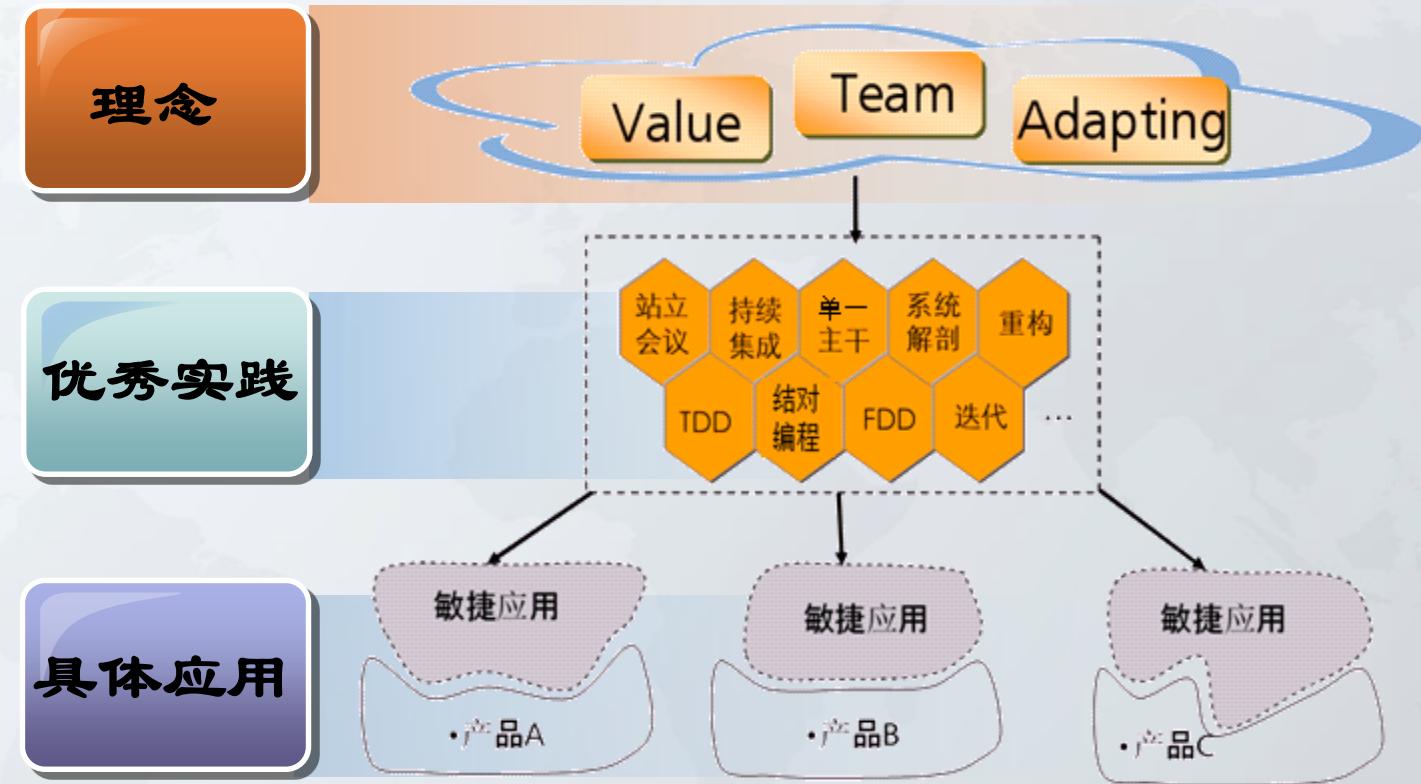
正确认识敏捷

对敏捷的常见误解



- 误解一：敏捷开发意味着可以不需要文档、设计和计划
- 误解二：敏捷只是一些优秀实践，或者是优秀实践的结合
- 误解三：敏捷只适用于小项目开发
- 误解四：敏捷只会对研发产生改变
- 误解五：管理者不需要亲自了解敏捷，只需要管理上支持就可以了
- 误解六：引入敏捷只需要按照既定的步骤去做就可以了
- 误解七：敏捷是CMM的替代品，是另一种流程
- 误解八：敏捷只注重特性的快速交付，在敏捷下架构不重要了

统一认识：敏捷=理念+优秀实践+具体应用



敏捷包括3个层次

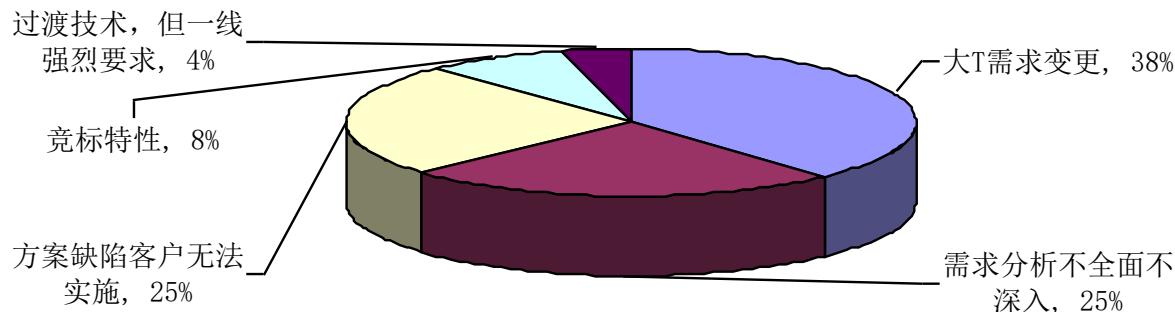
{

- 理念（敏捷核心思想）
- 优秀实践（敏捷的经验积累）
- 具体应用（能够结合自身灵活应用才是真正敏捷）

理念：聚焦客户价值(Value)，消除浪费

华为公司内部统计：研发版本废弃特性

- 07.1-08.6年某产品线所有产品中重要特性无应用的比例达22%（需求变更和分析不足占63%）

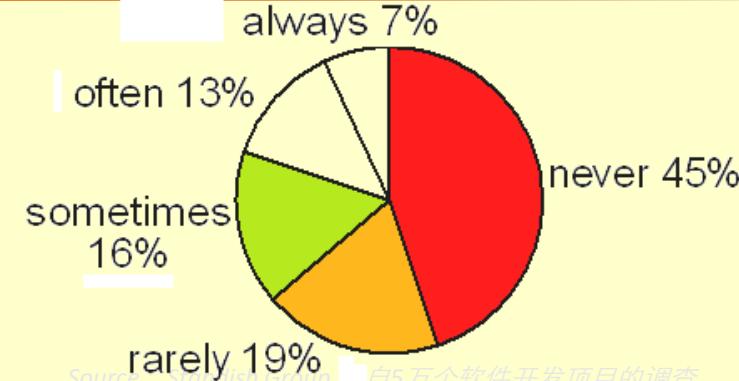


电信业：“电信级”带来的浪费

- 电路交换机的2000多个功能只用了1%
- 路由器网管的告警只有0.01%是有意义的
- 我们提供的上万种业务套餐80%以上使用者不到10个人，浪费了无数昂贵的资源

Source: 中国通信工业协会《华为公司工程与技术大会》上的讲话

软件业：45%的软件特性客户没有使用



Source: Steve McConnell, 自5万个软件开发项目的调查

“价值”在“敏捷宣言”中的体现

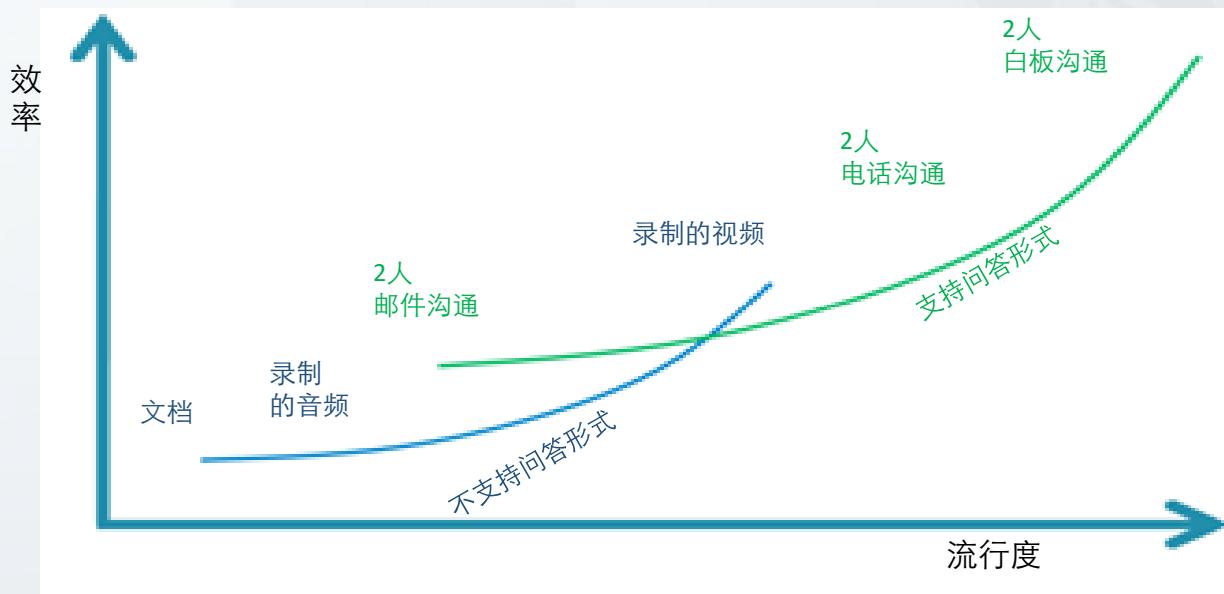
个体和交互	胜过	过程和工具
可以工作的软件	胜过	面面俱到的文档
客户合作	胜过	合同谈判
响应变化	胜过	遵循计划

产品商业成功为目标，聚焦客户价值、围绕价值流消除浪费

理念：激发团队(Team)潜能，加强协作

研究表明面对面的沟通最有效

业界调查：一个50人开发团队，每人平均30%时间用于编码，70%的时间用于与其他成员交流。



“团队”在“敏捷宣言”中的体现

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

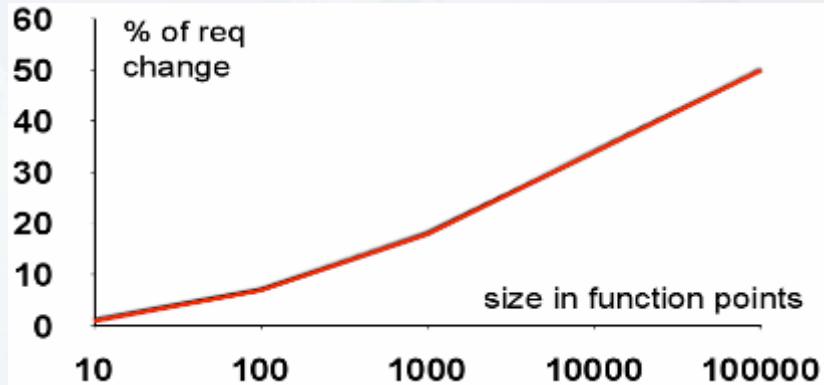
人是软件开发的决定因素

研究表明1981年来自不同公司的优秀程序员生产率之比是7:1，而2007年最新的研究数据，则是40:1。

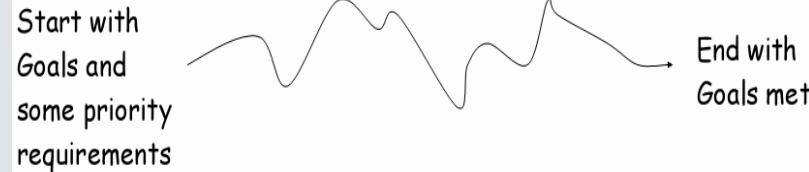
- 团队是价值的真正创造者，应加强团队协作、激发团队潜能
- 软件开发是一种团队活动，首先应做到提升沟通效率降低交流成本

理念：不断调整以适应(Adapting)变化

随软件规模增长，需求变化呈非线性增长



软件开发是复杂不可预测的经验控制过程



麦当劳是简单可预测生产过



软件开发规律再审视

《人月神话》：软件开发是人类最复杂工作之一，软件具有四个属性：复杂性、一致性、可变性和不可见性。

- 软件开发是不可重复、探索性的、演进的，适应性过程。

“适应变化”在“敏捷宣言”中的体现

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

不断的根据经验调整，最终交付达到业务目标的产品

优秀实践: 业界敏捷优秀实践概览



Scrum偏重项目和过程管理, XP偏重编程实践

■ 具体应用：因地制宜选择适合的敏捷实践

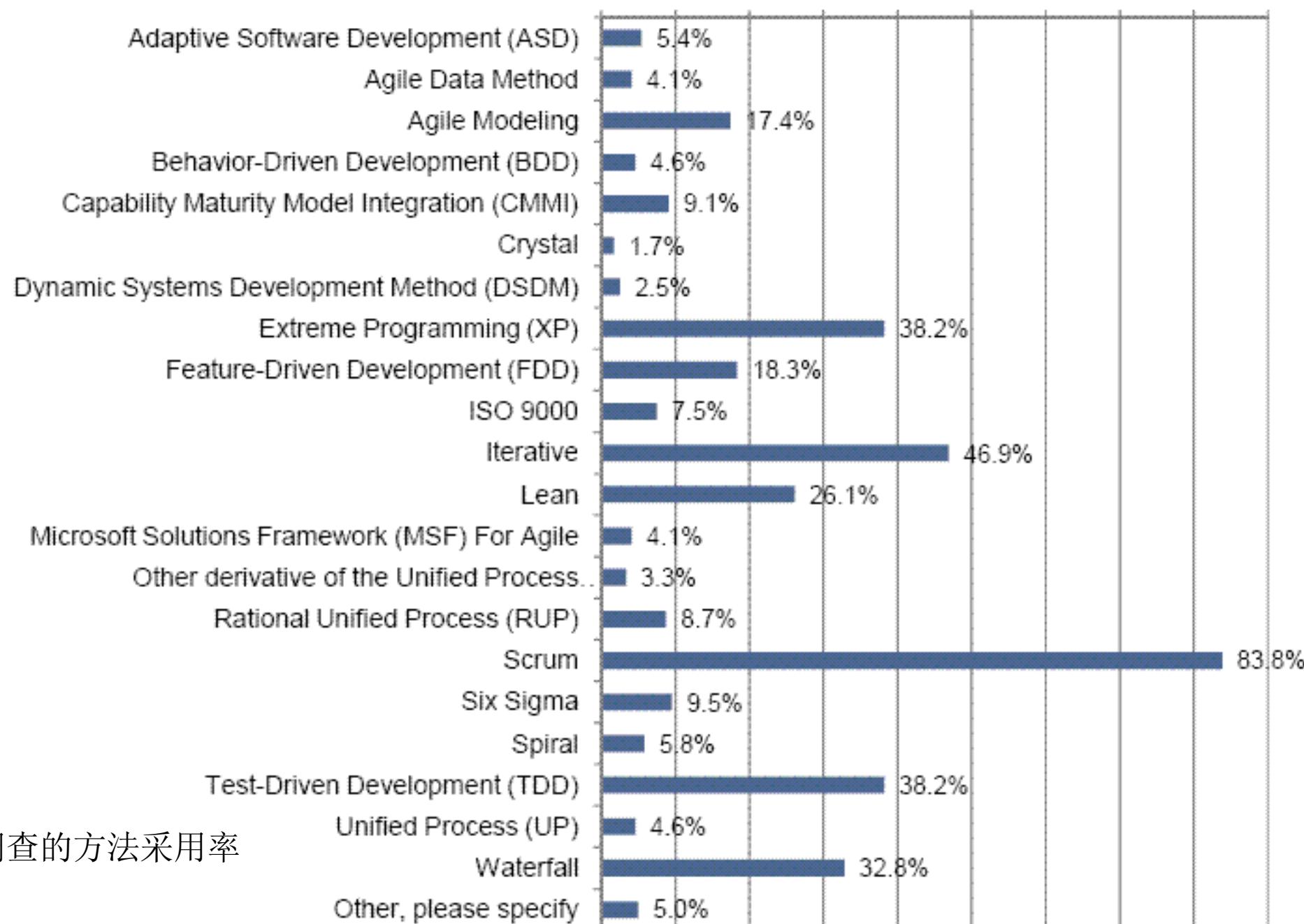


团队在透彻理解敏捷理念的基础上，可以灵活选择最适合自己的实践，避免教条化

敏捷实践(SCRUM)概览

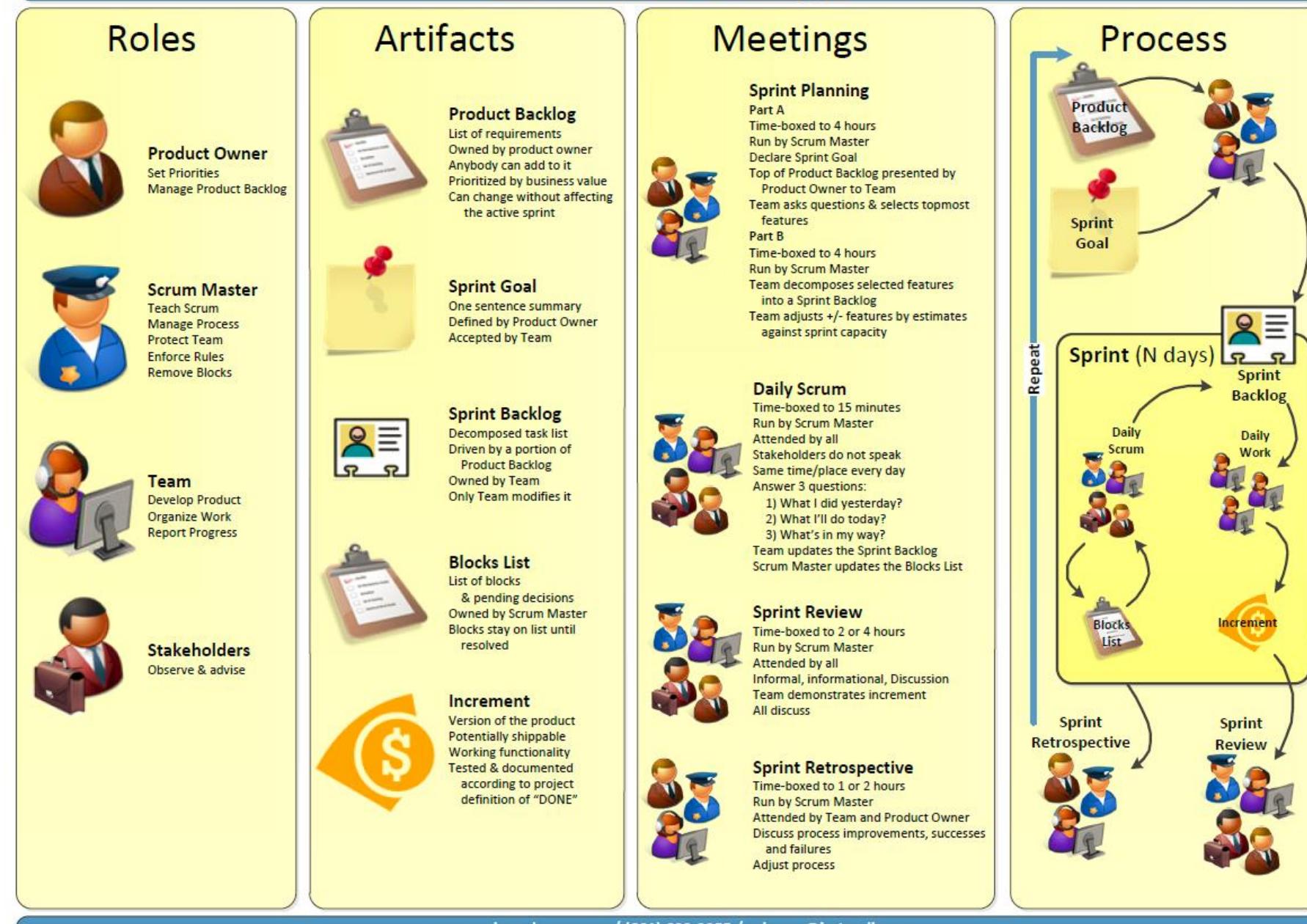
SCRUM是当前最流行的敏捷过程

0.0% 10.0% 20.0% 30.0% 40.0% 50.0% 60.0% 70.0% 80.0% 90.0%



Forrest Research 2009年调查的方法采用率

Scrum on a Page



敏捷软件开发过程



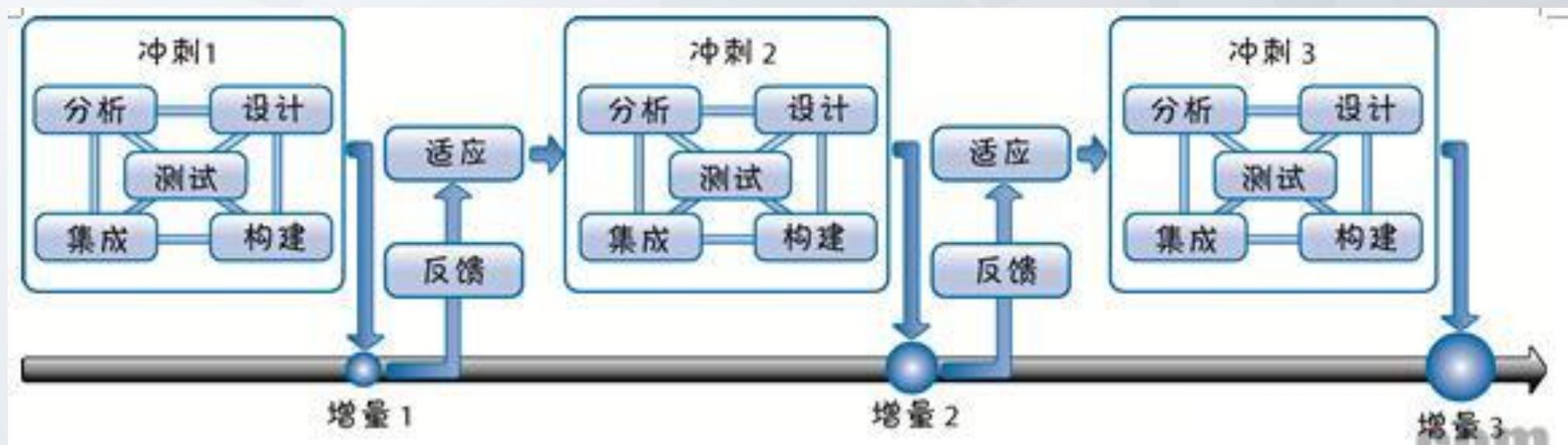
敏捷软件开发核心——迭代增量开发

什么是迭代式开发

- 迭代开发将整个软件生命周期分成多个小的迭代（一般2-4周）
- 每一次迭代都由需求分析、设计、实现和测试在内的多个活动组成
- 每一次迭代都可以生成一个稳定和被验证过的软件版本。

迭代式开发的好处

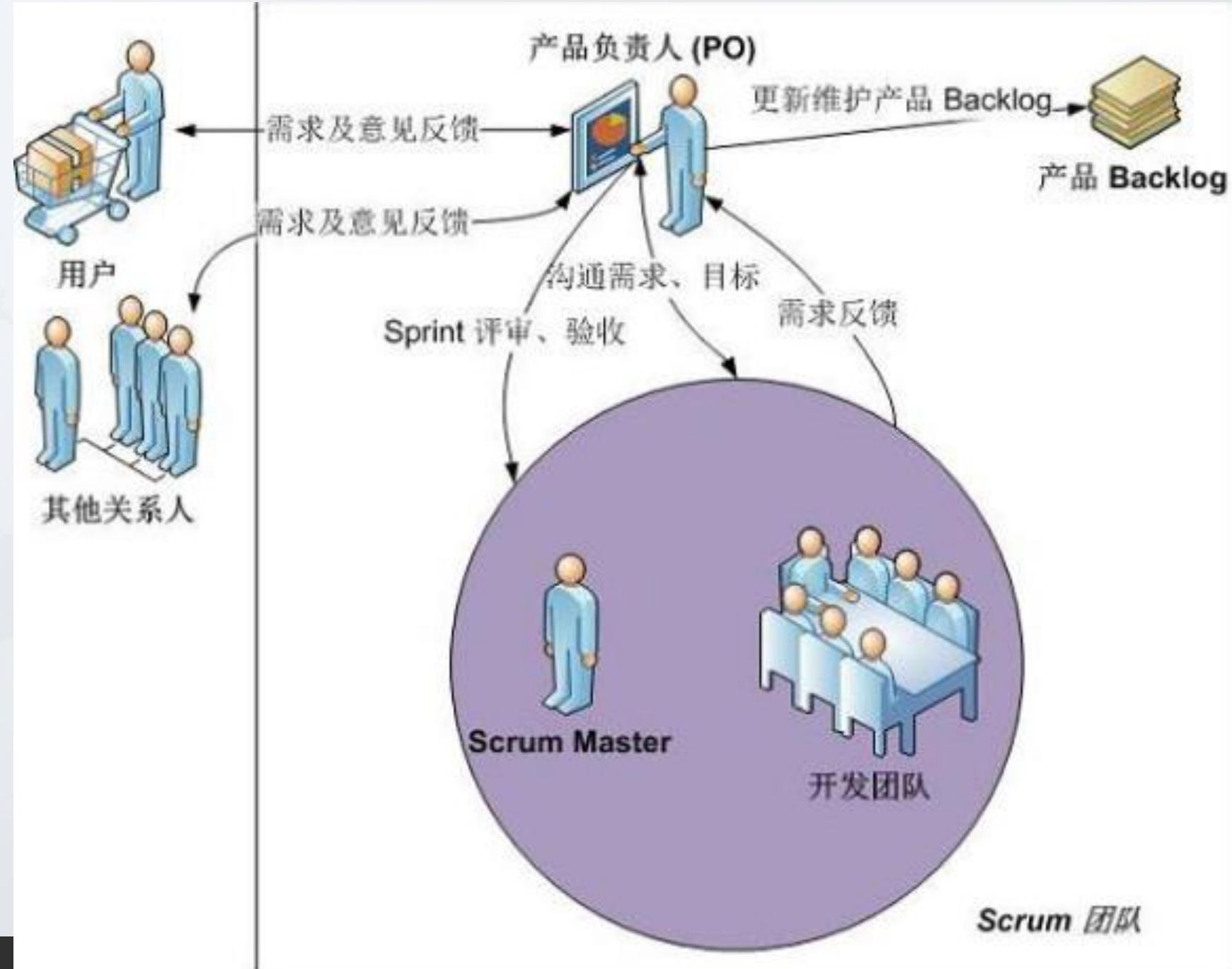
- 通过将高技术风险的需求在早期迭代里实现，有助于尽早暴露问题和及时消除风险
- 通过提供功能渐增的产品，持续从客户获得反馈，根据反馈及时调整，使最终产品更加符合客户的需要
- 通过小批量减少排队，提供更灵活、快速的交付能力
- 平滑人力资源的使用，避免出现瓶颈



迭代开发是有节奏地小步快跑，但建立在坚实的质量基础上

敏捷团队

敏捷团队包括3个核心角色：
PO(Product Owner)、**Scrum Master(Scrum教练)**和
Team(开发产品)

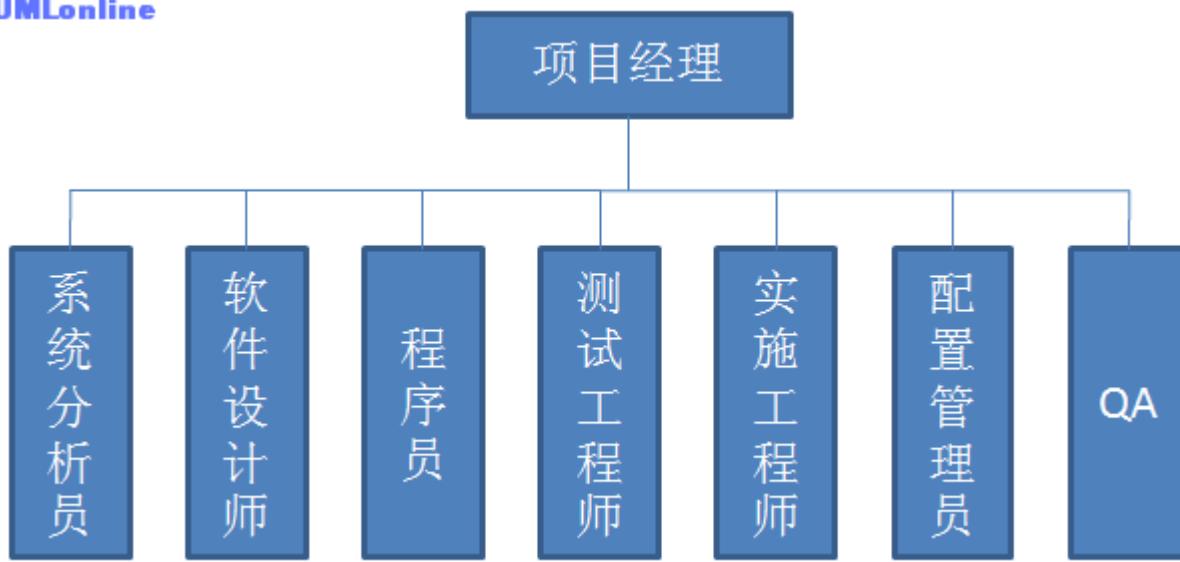


敏捷团队的角色职责

角色名称	角色定义	角色职责	注意事项
Product Owner (产品负责人)	确保Team做正确的事	<ul style="list-style-type: none">代表利益相关人（如用户、Marketing、用服、管理者等），对产品投资回报负责确定产品发布计划定义产品需求并确定优先级验收迭代结果，并根据验收结果和需求变化刷新需求清单和优先级	<ul style="list-style-type: none">除了客户需求之外，内部任务如重构、持续集成环境搭建等也由PO纳入统一管理
Scrum Master (Scrum教练)	确保Team正确地做事	<ul style="list-style-type: none">辅导团队正确应用敏捷实践引导团队建立并遵守规则保护团队不受打扰推动解决团队遇到的障碍激励团队	<ul style="list-style-type: none">不命令和控制Team
Team (开发团队)	负责产品需求实现	<ul style="list-style-type: none">负责估计工作量并根据自身能力找出最佳方案去完成任务且保证交付质量向PO和利益相关人演示工作成果（可运行的软件）团队自我管理、持续改进	<ul style="list-style-type: none">一般由5-9名跨功能领域人员组成坐在一起工作有共同的目标，共担责任团队成员严格遵守团队规则

其它类型项目团队

UMLonline



敏捷团队实践：完整团队（特种兵小组）

什么是完整团队

- 敏捷开发中，以Story为单位的持续交付要求系统组、开发和测试等跨功能团队进行密切协同，相互独立的功能团队难以应对
- 完整团队是跨功能领域（需求分析师、设计师、开发人员、测试人员、资料人员等）的人员组成一个团队，坐在一起工作，团队成员遵循同一份计划，服从于同一个项目经理。

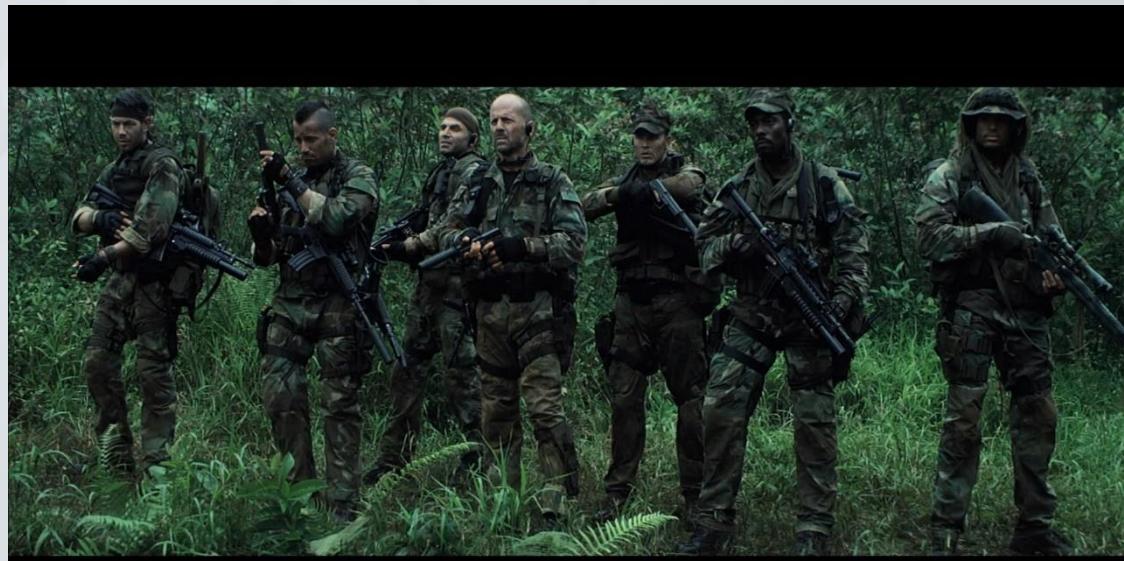
完整团队的好处

- 有助于团队成员形成共同目标和全局意识，促进各功能领域的拉通和融合；
- 通过面对面沟通提升沟通效率。
- 实现团队成员的高度协同，支撑高密度地、持续地、短周期的交付。

完整团队聚焦客户需求交付，提高协作效率

完整团队的关键要点

- 成员来自多功能领域：**团队拥有完成目标所需的各职能成员；
- 坐在一起办公：**团队成员无障碍地沟通；
- 团队保持相对稳定：**临时组建的团队生产效率较低，团队稳定非常关键。
- T型人才，背靠背精神：**每个成员都一专多能，工作上互助互补。



激发团队，敏捷方式下管理者的转变



敏捷方式下对管理者最大的挑战是学会放松“控制”(loose control)

敏捷方式下团队成员的转变

传统方式



团队成员是“听从安排的独立贡献者”：

- 被动等待主管下指令安排工作
- 独立工作为主，协作少
- 以文档和邮件为主要沟通方式
- 只关注个体任务“做完”，不关注团队目标
- 能力相对单一，学习动力不足

敏捷方式

Agile Organization



团队成员是“全方位的积极参与者”：

- 共同参与计划制定和任务安排
- 团队协作贯穿工作始终
- 面对面交流是主要沟通方式
- 关注团队目标，共担责任
- 能力要求更广，主动学习适应岗位要求

从被动到主动的心态转变是团队成员适应敏捷开发的关键

PO的特征



SM的特征

SM的特征

见多识广

善于提问

有耐心

有协作精神

保护团队

公开透明

■ 开发团队的特征

开发团队的特征

自组织

跨职能的多样化和全面化

T型技能

火枪手态度

高带宽沟通（广泛）

透明沟通

团队规模适中

专注、有责任感

工作节奏可持续

团队成员稳定

关于团队的深度思考

1. 谁来担任P0?

内部开发：内部业务方代表，例如为市场营销团队开发系统，那么就应该由市场营销团队中得到授权的人当P0；

商业开发：组织内部员工，充当实际客户的代言人，通常是产品管理或营销部门成员；

外包开发：甲方安排P0，乙方安排相应的人对接；

2. 谁来担任SM?

产品经理、项目经理、开发、测试、职业经理人、人力经理……，必须具备前面的6大特征，并愿意掌握SM的技能；

3. SM必须全职吗？

对于成熟的Scrum团队，SM可以兼任其它团队的SM工作；

4. Scrum团队是否需要保持稳定？

尽最大可能稳定，成熟的Scrum团队非常难形成，一旦形成战斗力极强；

敏捷软件开发过程

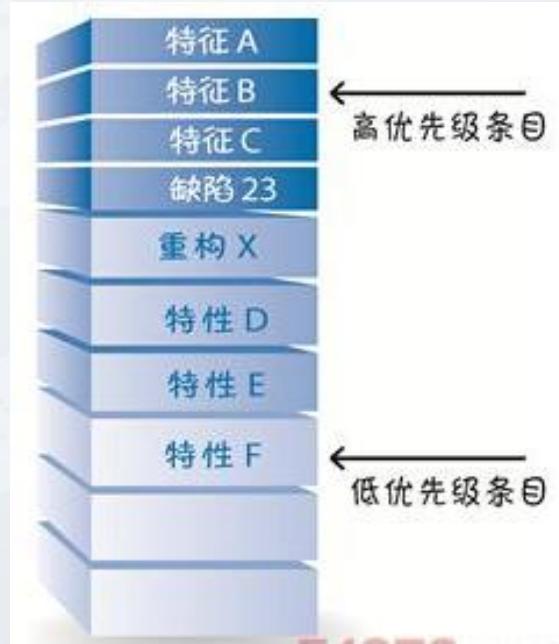


- ① PO和开发团队对产品业务目标形成共识
- ② PO建立和维护产品需求列表（需求会不断新增和改变），并进行优先级排序
- ③ PO每轮迭代前，Review需求列表，并筛选高优先级需求进入本轮迭代开发
- ④ 开发团队细化本轮迭代需求，并按照需求的优先级，依次在本轮迭代完成
- ⑤ 开发团队每日站立会议、特性开发、持续集成，使开发进度真正透明
- ⑥ PO对每轮迭代（2—4周）交付的可工作软件进行现场验收和反馈
- ⑦ 团队内部进行本轮冲刺的过程回顾，发现可改进的方面，指导下一轮迭代

敏捷工作件：产品Backlog

什么是产品Backlog

- 经过优先级排序的动态刷新的产品需求清单，用来制定发布计划和迭代计划。



产品Backlog的好处

- 通过需求的动态管理应对变化，避免浪费；易于优先交付对用户价值高的需求。

产品Backlog关键要点

清楚表述列表中每个需求任务对用户带来的价值，做为优先级排序的重要参考；

动态的需求管理而非“冻结”方式，PO持续地管理和及时刷新需求清单，在每轮迭代前，都要重新筛选出高优先级需求进入本轮迭代；

迭代的需求分析过程，而非一次性分析清楚所有需求（只对近期迭代要做的需求进行详细分析，其它需求停留在粗粒度）。

ID (统一标识符)	Name (标题) – 简短的、描述性的故事名	Story (故事) – 故事内容描述	Priority (重要性) – 产品负责人评出一个数值，指示这个故事有多重要	Initial estimate (初始估计) – 团队的初步估算，表示和其他故事相比，完成该故事所需的工作量	How to demo (如何做演示) – 它大略描述了这个故事应该如何在sprint 演示上进行示范	Notes (注解) – 相关信息、解释说明和对其他资料的引用等等
------------	-------------------------	---------------------	---	---	---	-----------------------------------

产品Backlog是需求动态管理的载体

产品功能列表

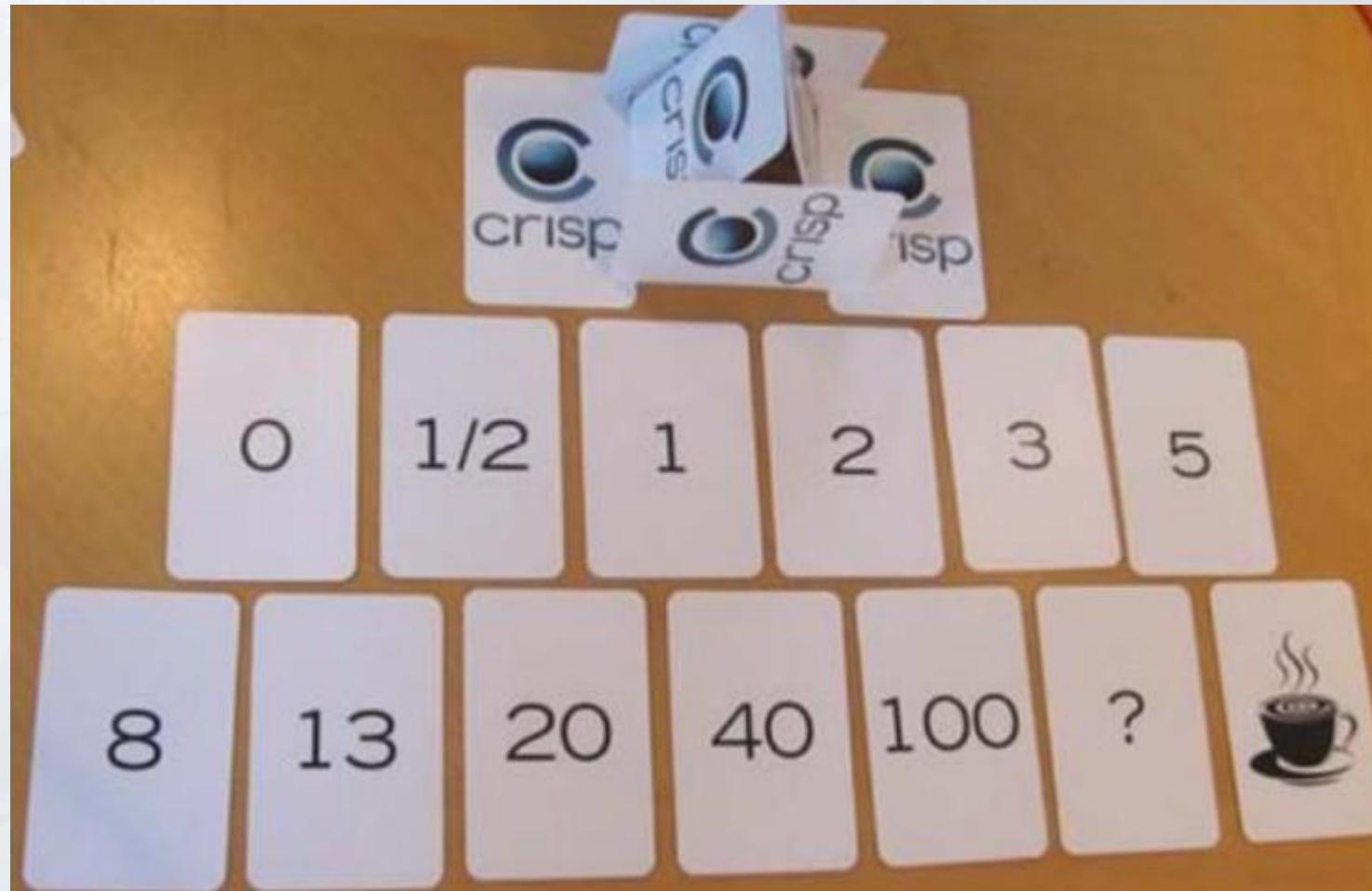
- 按优先级排序的预期产品功能集，通常组成包括
 - 特性
 - 缺陷
 - 技术工作
 - 知识获取

类型	例子
特性	作为客户服务代表，我希望为客户支持问题创建一个记录卡，这样可以记录和管理客户的支持请求
变更	作为客户服务代表，我希望搜索结果在默认情况下按照姓氏排序，不要按记录卡号排序，这样可以更容易查找支持记录卡
缺陷	修复缺陷跟踪系统中的Bug#256，这样可以使客户在搜索项中输入特殊字符时不会出现异常
技术改进	迁移到最新版Oracle数据库管理系统
获取知识	为两种架构同时创建原型或概念验证，执行三次测试，这样可以确定哪种方法更适合产品

好的产品功能列表具备DEEP特征

- 详略得当。马上要做的在顶部，工作量小，内容非常详细，可以在最近的一个冲刺实现。近期不打算做的放在底部，工作量大，内容粗略。
- 涌现的。根据不断涌入、具有经济价值的信息持续更新，目的是适应变化。产生涌的原因：
 - 客户的新想法
 - 竞争对手的行动
 - 意外的技术问题等
- 做过估算。每个条件都做过大小估算（就是完成这个条目需要的工作量，用故事点或理想天数表示）。顶部的详细精确，底部的可以用L、XL等表示
- 排列了优先级。最近几个冲刺，排好顺序，后续的先不排。如果有新条目出现，插入队列（原则上不插入当前冲刺）

使用估算卡片估算PB工作量



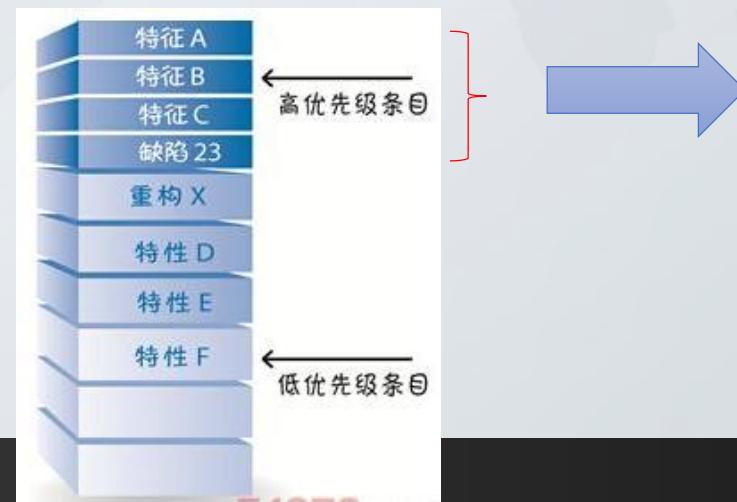
敏捷工作件：迭代Backlog

什么是迭代Backlog

- 迭代Backlog是团队在一轮迭代中的“任务”（Task）清单，是团队的详细迭代开发计划；
- 当团队接收从产品Backlog挑选出要在本轮迭代实现的需求时，召开团队迭代计划会议，将需求转化为具体的“任务”；
- 每项任务信息包括当前剩余工作量和责任人。

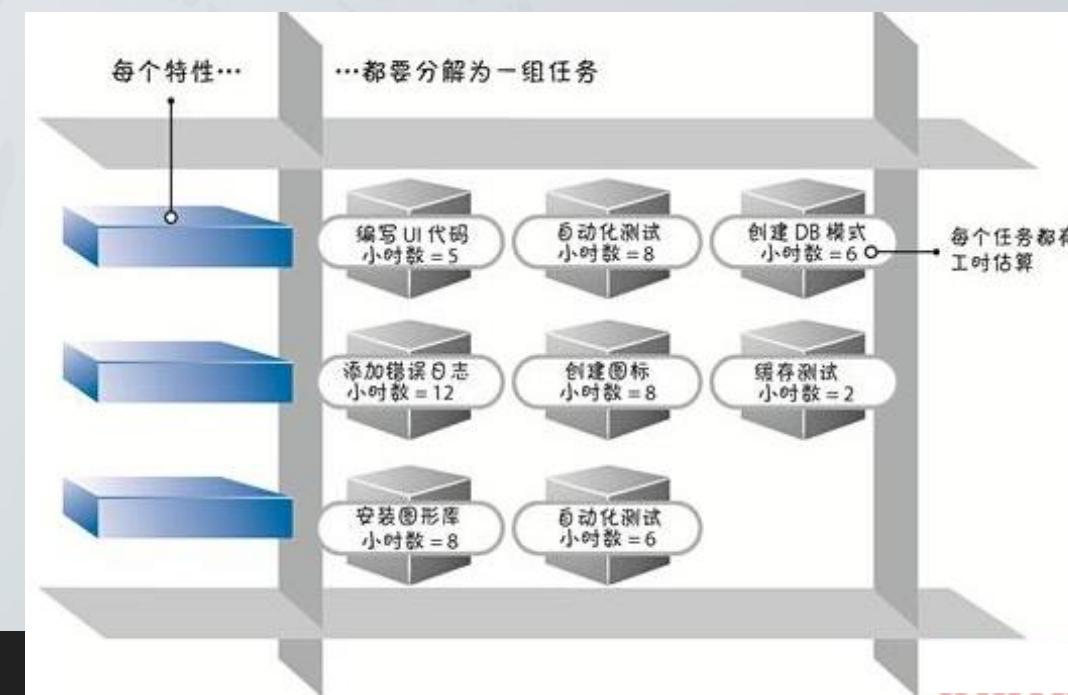
迭代Backlog的好处

- 将需求分解成更细小的任务，利于对迭代内进度进行精确控制；
- 剩余工作量可用来实时跟踪团队当前进展。



迭代Backlog关键要点

- “任务”由团队成员自己分解和定义，而不是上级指派，支撑需求完成的所有工作都可以列为任务；
- 任务要落实到具体的责任人；
- 任务粒度要小，工作量大于两天的任务要进一步分解；
- 用小时做为任务剩余工作量的估计单位，并每日重估计和刷新。



迭代 Backlog 示例

Sprint 1 Backlog

Item #	Priority	Product Backlog Item	Size	Task	Owner	Est. [h]	Status
3	1	Design web page look and feel	2	Check requirements with customer	John	1	Done
任务描述				Discuss on page layout	Erik, Pavel, Marian, John	8	Done
				Create web page template	Pavel	6	In progress
				Create design document	Marian	3	In progress
				Review design with customer	Erik, John	4	Not started
5	2	Create graphics & banners	5	Task	Owner	Est. [h]	Status
				Create Hotel logo	Erik	2	Not started
				Create animated advertisement banner		2	Impeded
				Create background images	John	6	Not started
				Review graphics with customer,an	4	Not started
				Agree on colours used	el,		
				Make photos of the hotel	John	4	Not started
					John	8	Done

迭代目标

任务执行者

预估工作量

任务描述

任务状态

任务阻塞

敏捷工作件：完成标准（Definition of Done）

什么是完成标准

- 基于“随时可向用户发布”的目标制定衡量团队工作是否已完成的标准，由团队和PO形成共识；

完成标准的好处

- 共同协商的完成标准是团队的自我承诺，团队会更认真；
- 用于准确评估团队工作进展；
- 清晰和明确的完成标准保证了每次迭代是高质量的。

完成标准确保团队每一步前进都奠定在坚实的质量基础之上

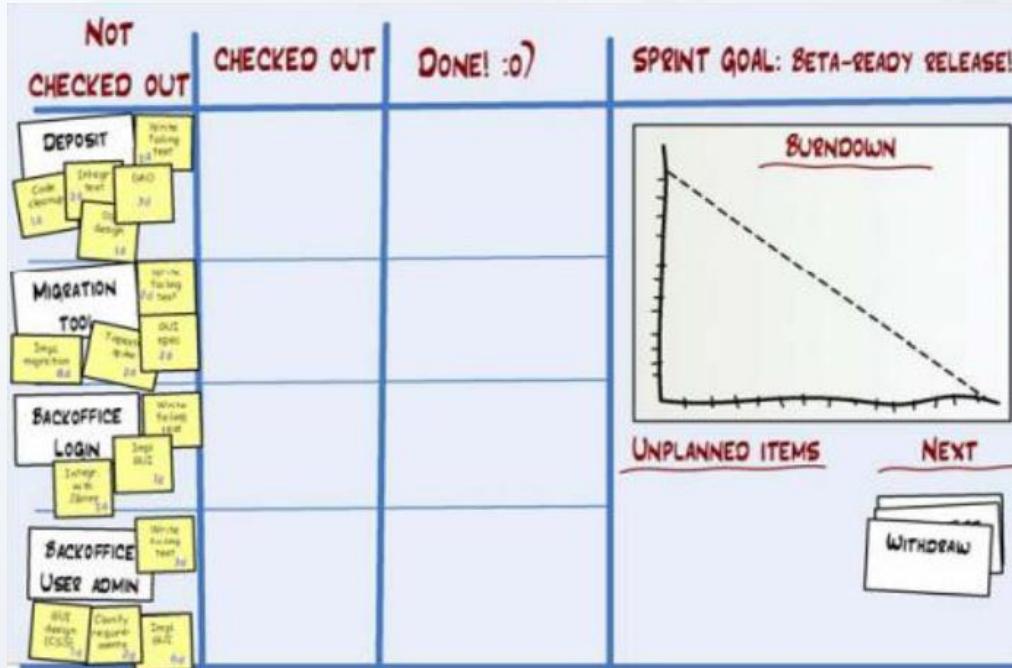
完成标准的关键要点

- 团队自协商：团队根据项目实际情况来定义完成标准，并严格遵守；
- 有层次：一般分为三个层次：Story级别，迭代级和发布级，每个级别都有各自的完成标准。

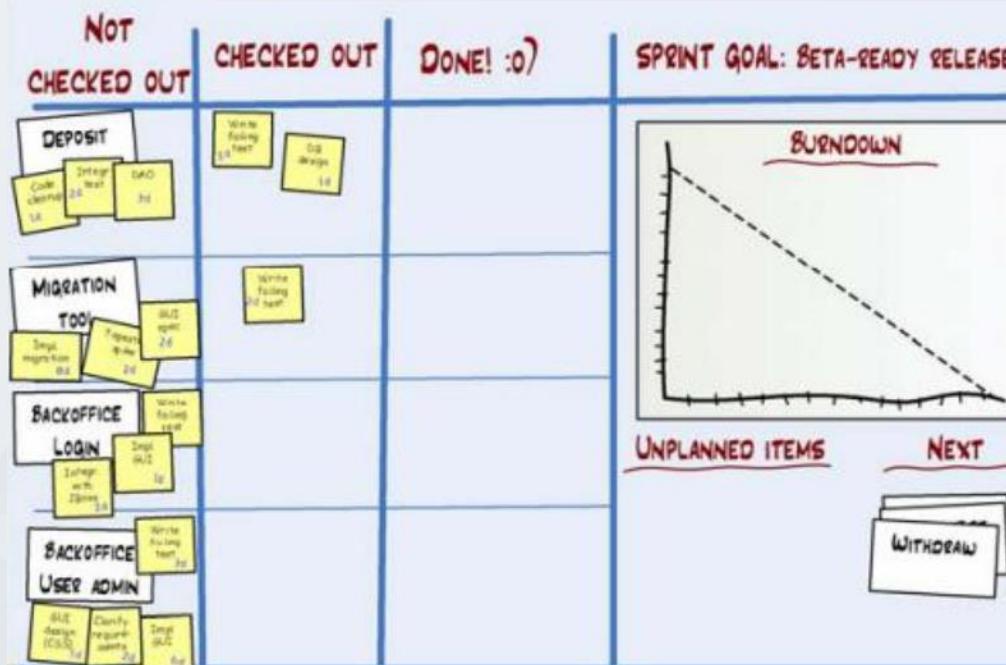
完成标准的样例

Story完成 标准样例	代码合入主干	持续集成无错误
	代码100%检视	代码100%通过单元测试
	代码符合规范	通过验收测试
迭代完成标 准样例	所有Story完成	系统测试用例100%通过
	所有缺陷解决	通过迭代验收
	通过回归测试	通过性能测试
发布完成 标准样例	更新配套资料	

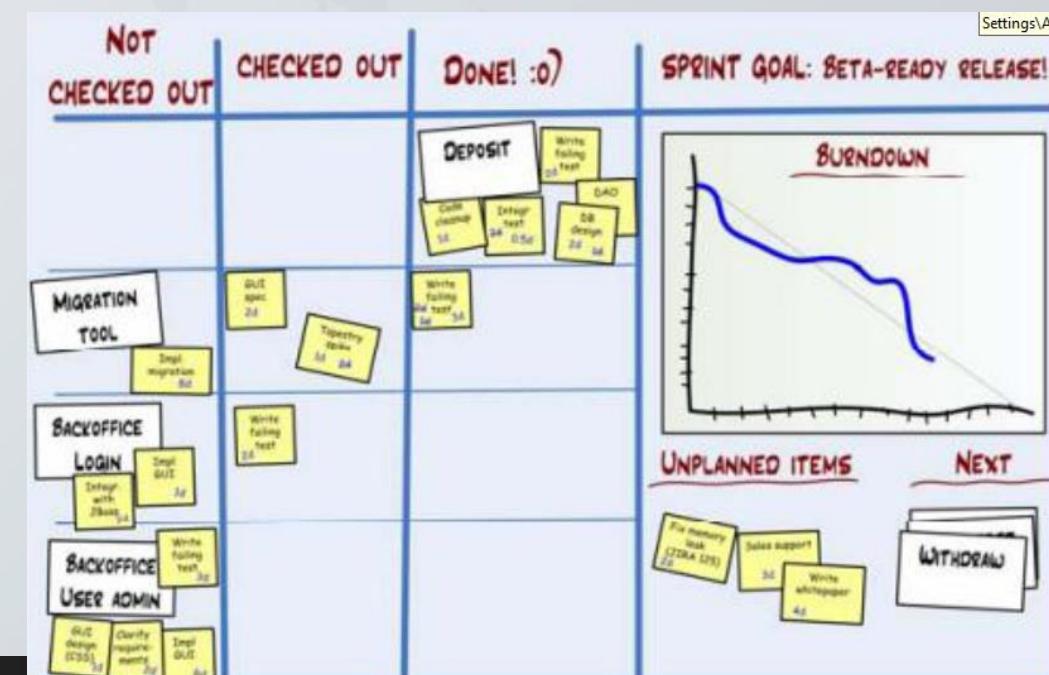
敏捷工作件：任务看板



1. 迭代计划会以后，任务板可能是这样



2. 在首次每日例会以后，任务板可能会变成这样



3. 几天以后，任务板可能会变成这样

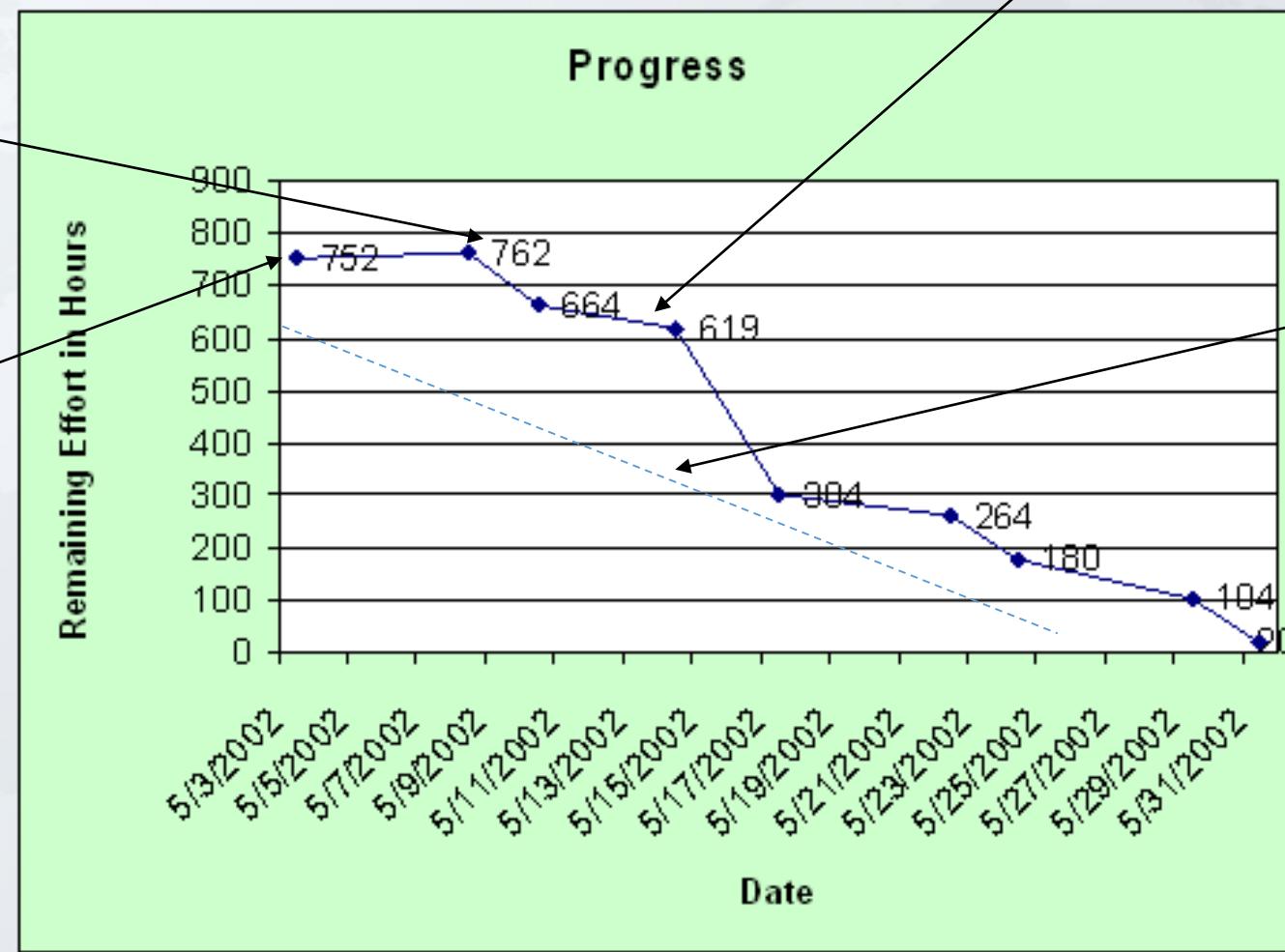
敏捷工作件：燃尽图(Burn Down Chart)

实际工作时所有任务剩余的工作时间 (h) 线

估计不足导致的任务工作量增加

迭代开始预估时间(752 h)

完成任务的最快耗时线



----- 理想燃尽
—— 实际燃尽

敏捷管理实践：迭代计划会议

什么是迭代计划会议

- 每轮迭代启动前，团队共同讨论本轮迭代详细开发计划的过程，输入是产品Backlog，输出是团队迭代Backlog
- 多团队迭代计划会议要分层召开
 - 版本迭代计划会议：将产品Backlog（需求）分配给团队
 - 团队迭代计划会议：将选取的产品Backlog需求转换成迭代Backlog（任务），分配给团队成员；
- 迭代计划会议内容：
 - 澄清需求、对“完成标准”达成一致
 - 工作量估计、根据团队能力确定本轮迭代交付内容
 - 细化、分配迭代任务和初始工作计划。

迭代计划会议由团队共同确定迭代交付内容和完成标准

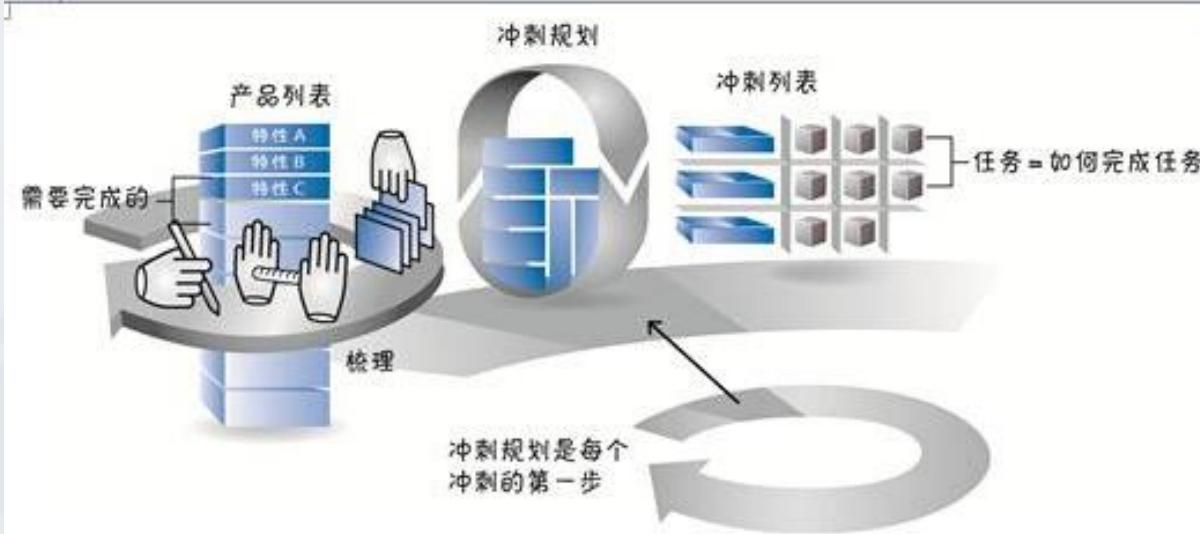
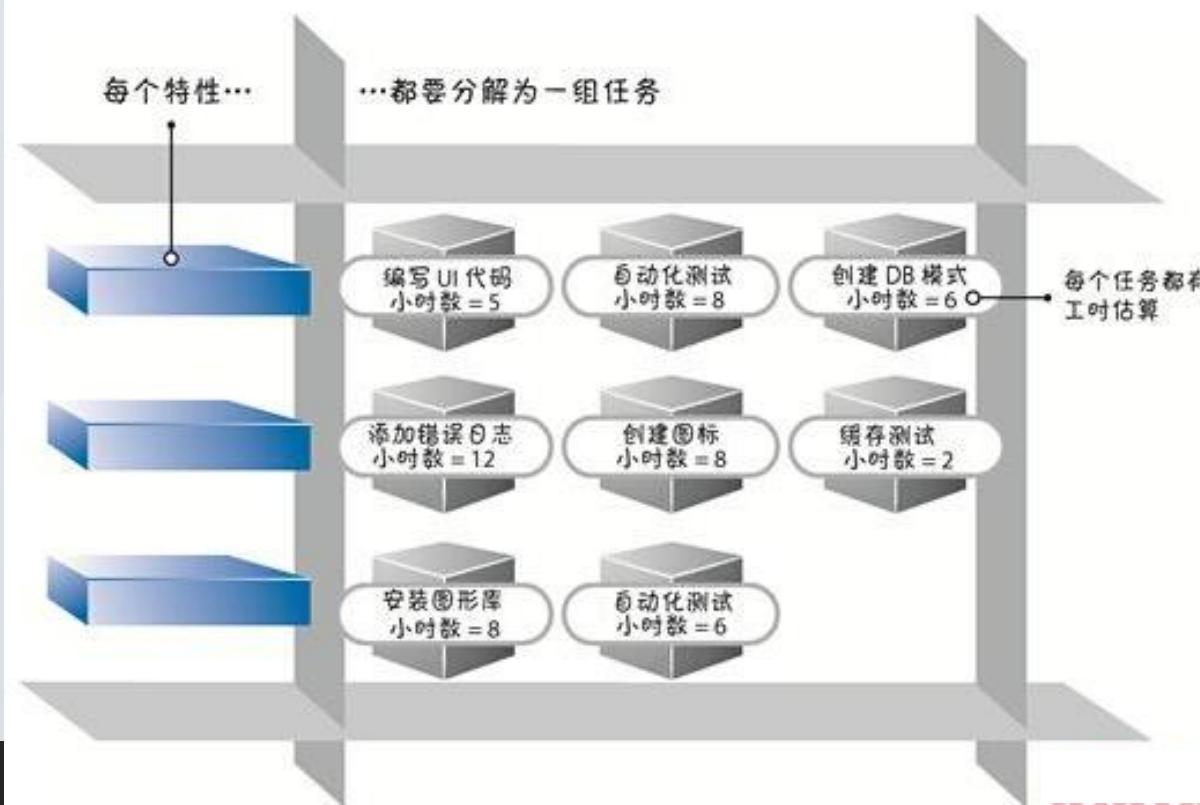


图 2.8 冲刺规划



敏捷管理实践：迭代计划会议

迭代计划会议的好处

- 通过充分讨论，使团队成员对任务和完成标准理解一致
- 团队共同参与，促进团队成员更认真对待自己的承诺。



迭代计划会议的关键要点

充分参与：Scrum Master确保PO和Team充分参与讨论，达成理解一致；

相互承诺：Team承诺完成迭代Backlog中的需求并达到“完成标准”，PO承诺在短迭代周期不增加需求（2-4周）；

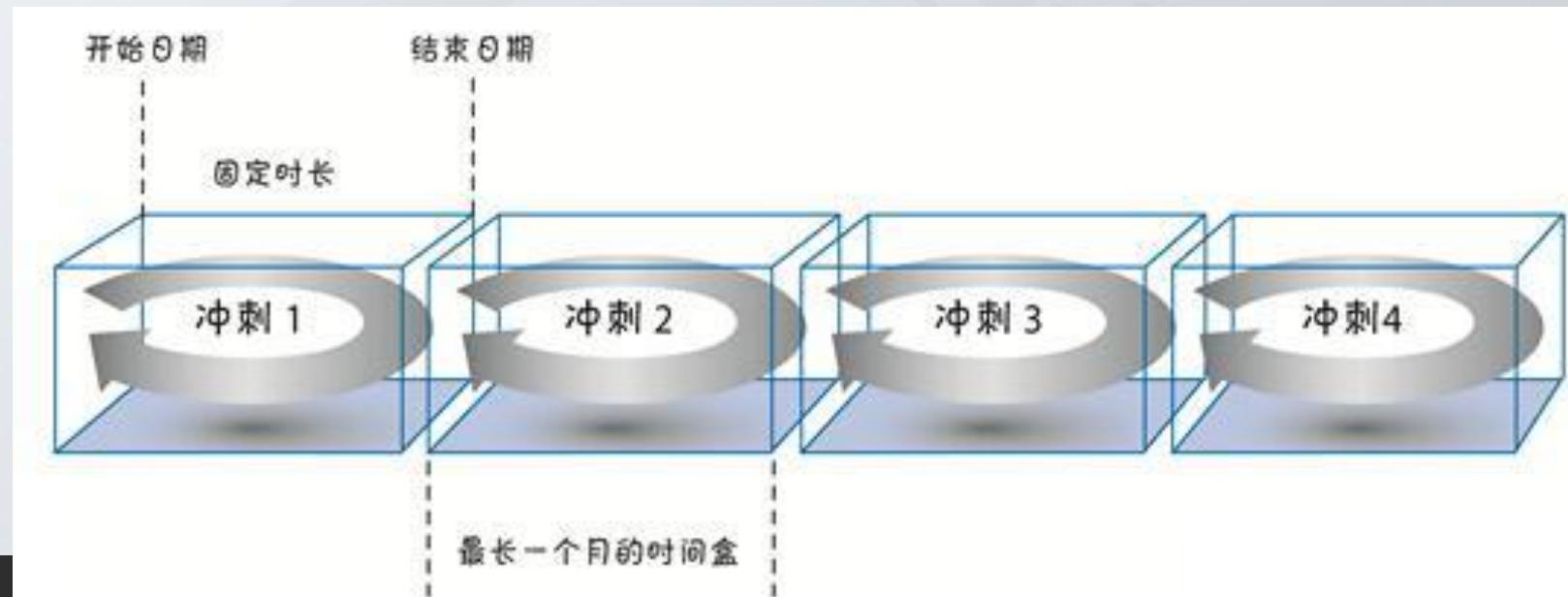
确定内部任务：Team和PO协商把一些内部任务放入迭代中（例如重构、持续集成环境搭建等），由PO考虑并与其他外部需求一起排序。

迭代 Backlog 规划

每个迭代的周期通常为2-4周， 对应的规划时间为4-8小时

参与者：

- P0：分享本迭代的目标，展示排好序的PBI, 回答团队提出的问题
- 团队：确定可交付哪些特性，在冲刺规划结束时做出承诺
- SM：观察规划活动，提出深入细节的问题，引导并帮助团队确保有成果



迭代 Backlog 规划步骤



某团队开展两周（**10工作日**）的迭代，全部时间都用于执行工作吗？

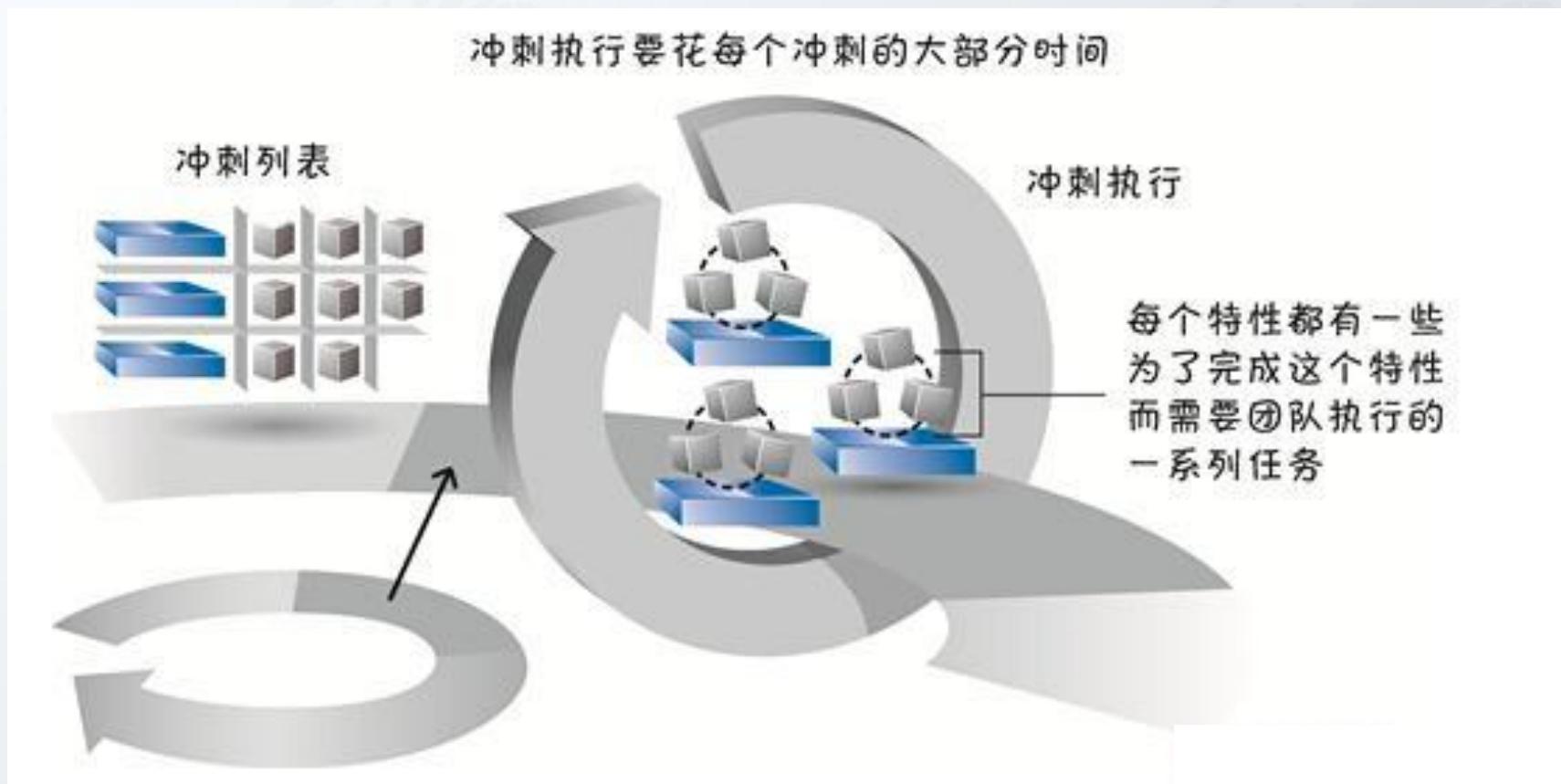
- 1.SCRUM活动；
 - 2.支持、维护；
 - 3.日常事务：会议、邮件...
 - 4.休假！！
- 剩下的才是真正的生产能力

- 1.有迭代目标的按目标选择
- 2.无目标的按排好序的优先级选择
- 3.绝不选择一个迭代做不完的PBI

选取PBI的总工作量小于团队的迭代速率，同时考虑当前迭代团队的生产能力

团队给出承诺，在本迭代周期结束时完成所选的PBI

敏捷管理实践：迭代执行



敏捷管理实践：每日站立会议

什么是每日站立会议

- 每日工作前，团队成员的例行沟通机制，由Scrum Master组织，Team成员全体站立参加
- 聚焦在下面的三个主题：
 - 我昨天为本项目做了什么？
 - 我计划今天为本项目做什么？
 - 我需要什么帮助以更高效的工作？

每日站立会议的好处

- 增加团队凝聚力，产生积极的工作氛围
- 及时暴露风险和问题；
- 促进团队内成员的沟通和协调。

每日站立会议的关键要点

- 准时开始：**按计划会议制定的时间地点开会，形成团队成员的自然习惯；
- 高效会议：**会议限时15分钟，每个人都保持站立，依次发言，不讨论与会议三个主题无关的事情（如技术解决方案等）；
- 问题跟踪：**Scrum Master应该记录下所有的问题并跟踪解决；



每日站立会议促进团队沟通协调，及时暴露问题

敏捷管理实践：迭代评审会

什么是迭代验收

- 每次迭代开发结束时举行，通过演示可工作的软件检查需求是否满足客户要求；
- 由Scrum Master组织， PO和用户代表（外部或内部利益相关人）负责验收、Team负责演示可工作软件。

迭代验收的好处

- 通过演示可工作的软件来确认项目的进度，具有真实性；
- 能尽早的获得用户对产品的反馈，使产品更加贴近客户需求

迭代验收的关键要点

- **展示“真实” 的产品：** Team 应在真实环境中展示可运行的软件，判断是否达到“完成”标准；
- **收集反馈：** PO 根据验收情况及客户反馈意见，及时调整产品Backlog。



迭代验收尽早演示可工作的软件，收集反馈意见

迭代评审会准备工作

迭代评审准备工作

确定邀请谁参加

团队、内外部干系人

安排活动日程

最好固定时间和地点，形成规律
通常2-4小时

确认迭代工作已经完成

由PO确认

为演示做准备

以演示成果为主，减少PPT演讲
对于无界面的成果如何演示？

确定谁做什么

通常SM组织、PO主持，团队代表演示
每一次让不同的团队成员演示

迭代评审会的典型内容

PO 概要说明
迭代目标
中哪些完
成，哪些
没完成

唱成 演示完成
的增量

讨论产品
当前状态

团队 调整产品
未来方向

- 1.如果完成的成果与目标不符，团队要给出解释。但评审的目的是通过对当前目标的探讨，确定最佳前进路线。所以不要成为互相指责的场合。
- 2.如果评审时，一位级别高的干系人不同意接收产品，他认为PBI没有完成，怎么办？

敏捷管理实践：迭代回顾会议

什么是迭代回顾会议

- 在每轮迭代结束后举行的会议，目的是分享好的经验和发现改进点，促进团队不断进步；
- 围绕如下三个问题：
 - 本次迭代有哪些做得好
 - 本次迭代我们在哪些方面还能做得更好
 - 我们在下次迭代准备在哪些方面改进？

迭代回顾会议的好处

- 激励团队成员；
- 帮助团队挖掘优秀经验并继承；
- 避免团队犯重复的错误；
- 营造团队自主改进的氛围。

迭代回顾会议是促进团队持续改进的最有效手段

迭代回顾会议的关键要点

- 会议气氛：**Team全员参加，气氛宽松自由，畅所欲言，头脑风暴发现问题，共同分析根因；
- 关注重点：**Team共同讨论优先级，将精力放在最需要的地方（关注几个改进就够了）；
- 会议结论要跟踪闭环：**可以放入迭代backlog中。



迭代回顾会准备工作

迭代回顾准备工作

定义回顾重点

选择练习活动

收集客观数据

安排回顾日程

可以是对整个过程的回顾，也可以有所侧重点，例如：如何提高团队的TDD能力

- 1.建立并挖掘迭代事件的时间线
- 2.通过头脑风暴获得见解
- 3.将获得的见解进行分组并投票表决

- 1.未完成的PBI数量
- 2.燃尽图特征
- 3.....

最好固定时间和地点，形成规律
通常2-4小时

迭代回顾会典型内容

营造氛围

识别见解

结束活动

共同背景

确定措施

安全感
人人参与
积极发言

通过客观数据
尽可能减少主
观分歧

做的好
做的不好
可以改进

团队共同认可
的有价值的措
施，纳入后续
工作中改进
(集体投票)

确认此次活
动的成果，尤
其是强调要改
进的措施

敏捷工程实践技术

- 用户故事
- 结对编程
- TDD（测试驱动开发）
- 持续集成
- CodeReview
- 发布规则

敏捷工程实践：用户故事（user story）

什么是用户故事

- 用户故事是站在用户角度描述需求的一种方式；
- 每个用户故事须有对应的验收测试用例；
- 用户故事是分层分级的，在使用过程中逐步分解细化；
- 典型的描述句式为：**作为一个XXX客户角色，我需要XXX功能，带来XXX好处。**

故事样例

初始需求：1.作为网络规划人员，我想要配置一个媒体网关，因为想要增加网络容量和服务

初次分解：1.1作为网络规划人员，我想把媒体网关参数上传到管理系统
1.2作为网络规划人员，我想从管理系统下载媒体网关参数

再次分解：1.2.1作为网络规划人员，我想用文件方式从管理系统下载媒体网关参数

用例：用户在管理系统上选择以文件方式下载媒体网关参数，执行成功后，检查文件是否正确下载到本地且内容正确

1.2.2作为网络规划人员，我想用MML结构方式从管理系统下载媒体网关的参数

用例：.....

用户故事的关键要点

- I – Independent，可独立交付给客户
- N – Negotiable，便于与客户交流
- V - Valuable，对客户有价值
- E - Estimable，能估计出工作量
- S - Small，分解到最底层的用户故事粒度尽量小，至少在一个迭代中能完成
- T - Testable，可测试

用户故事的好处

- 用户故事站在用户视角便于和客户交流，准确描述客户需求；
- 用户故事可独立交付单元、规模小，适于迭代开发，以获得用户快速反馈；
- 用户故事强调编写验收测试用例作为验收标准，能促使需求分析人员准确把握需求，牵引开发人员避免过度设计。

用户故事便于团队站在用户角度分解细化需求并制定验收标准

用户故事特征

- 体现客户（用户）价值，轻量级的点位符
- 3C特质
 - 卡片（Card）：作为XX，我希望XXX，这样可以XXX
 - 对话（Conversation）：不描述到细节，由团队通过持续对话细化，激发大家的深度理解
 - 确认（Confirmation）：有明确的验收标准

用户故事大小级别

- 史诗故事（1-2个月）
- 特性故事（1-2周）
- 冲刺故事（1-2天）
- 任务（几小时）：可分工执行

用户故事INVEST标准

- 独立：故事之间松耦合，依赖小，否则无法进行优先级排序
- 可协商：开始仅用于做占位符，逐步细化
- 有价值：客户付费、用户使用（支付时间）
- 可估算：设计、开发、测试团队可估算工作量和成本
 - 不可估算的原因：太大需要分解，或信息不全需要进一步探索
- 大小合适：冲刺中的故事应足够小（例如两周冲刺，故事就应该是2天以内的）
- 可测试：有相应测验收标准（史诗级的可以没有）

用户故事约束（验收条件）

- 作为用户故事的约束体现（Card背面）

上传文件
作为一个维基用户，
我想上传一个文件
到维基，这样我可以和同事们分享。

满意条件
验证txt和doc文件
验证jpg,gif和png文件
验证小于等于1GB的mp4文件
验证无数字版权管理限制的文件

非功能性需求如何表达？

国际化
作为用户，我想要可以支持英语、一种罗曼语与一门复杂语言的接口，这样可以有很高统计概率应对需要支持的所有30种语言。

网络浏览器支持
系统必须支持IE8、IE9、FireFox6、
FireFox7、Safari5和Chrome15

知识获取性故事如何表达？

- 一般偏重后台实现，例如：原型、概念验证、学习、探针等
- 必须向PO说明价值，并且不宜出现太多

迁移至新版Oracle
作为开发人员，我想迁移系统到最新版Oracle数据库管理系统上去，这样可以避免在即将退役的Oracle版本上运营。

自动构建
作为开发人员，我想让构建在我提交代码时自动运行，这样可以避免在引入回归错误的时候就发现它。

如何收集用户故事

- 用户故事写作研讨会，产生首批用户故事
 - 参与者：P0、SM、Team、内部干系人、用户
 - 方法：头脑风暴、人物角色、思维导图、故事地图
 - 时间：几小时到几天
- 故事地图
 - 史诗故事按时间流横向排开
 - 纵向按优先级排序

敏捷工程实践：结对编程

什么是结对编程

- 两位程序员在一台电脑前工作，一个负责敲入代码，而另外一个实时检视每一行敲入的代码；
- 操作键盘和鼠标的程序员被称为“驾驶员”，负责实时评审和协助的程序员被称为“领航员”；
- 领航员检视的同时还必须负责考虑下一步的工作方向，比如可能出现的问题以及改进等。



结对编程的关键要点

- 程序员应经常性地在“驾驶员”和“领航员”间切换，保持成员间平等协商和相互理解，避免出现一个角色支配另一个角色的现象；
- 开始一个新Story开发的时候即可变换搭档，以增进知识传播；
- 培养团队成员积极、主动、开放、协作的心态能够增进结对编程效果；
- 实施初期需要精心辅导，帮助团队成员克服个性冲突和习惯差异。

结对编程的好处

- 有助于提升代码设计质量；
- 研究表明结对生产率比两个单人总和低 15%，但缺陷数少 15%，考虑修改缺陷工作量和时间都比初始编程大几倍，所以结对编程总体效率更高 (*source: The Economist*)；
- 结对编程能够大幅促进团队能力提升和知识传播。

结对编程提高代码质量和工作效率

敏捷工程实践：测试驱动开发（TDD）

什么是测试驱动开发

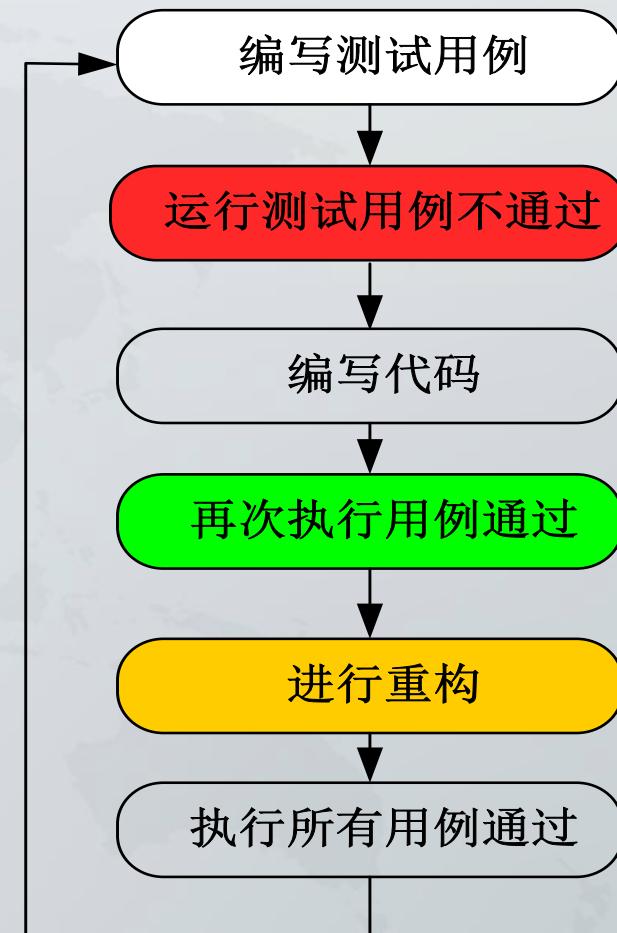
- TDD以测试作为编程的中心，它要求在编写任何代码之前，首先编写定义代码功能的测试用例，编写的代码要通过用例，并不断进行重构优化；
- TDD要求测试可以完全自动化运行。

测试驱动开发的关键要点

- 测试代码和源代码一样都需要简洁，可读性好；
- 测试用例的设计要保证完备，覆盖被测单元的所有功能；
- 每个测试用例尽量保持独立，减少依赖，提高用例的可维护性；
- 当功能单元较大时，为降低难度，可分解为多个更小的功能单元，并逐一用 TDD 实现。

测试驱动开发的好处

和代码同步增长的自动化测试用例，能为代码构筑安全网，保证代码重构的质量；
TDD有助于开发人员优化代码设计，提高代码可测试性



测试驱动开发保证代码整洁可用（Clean code that works）

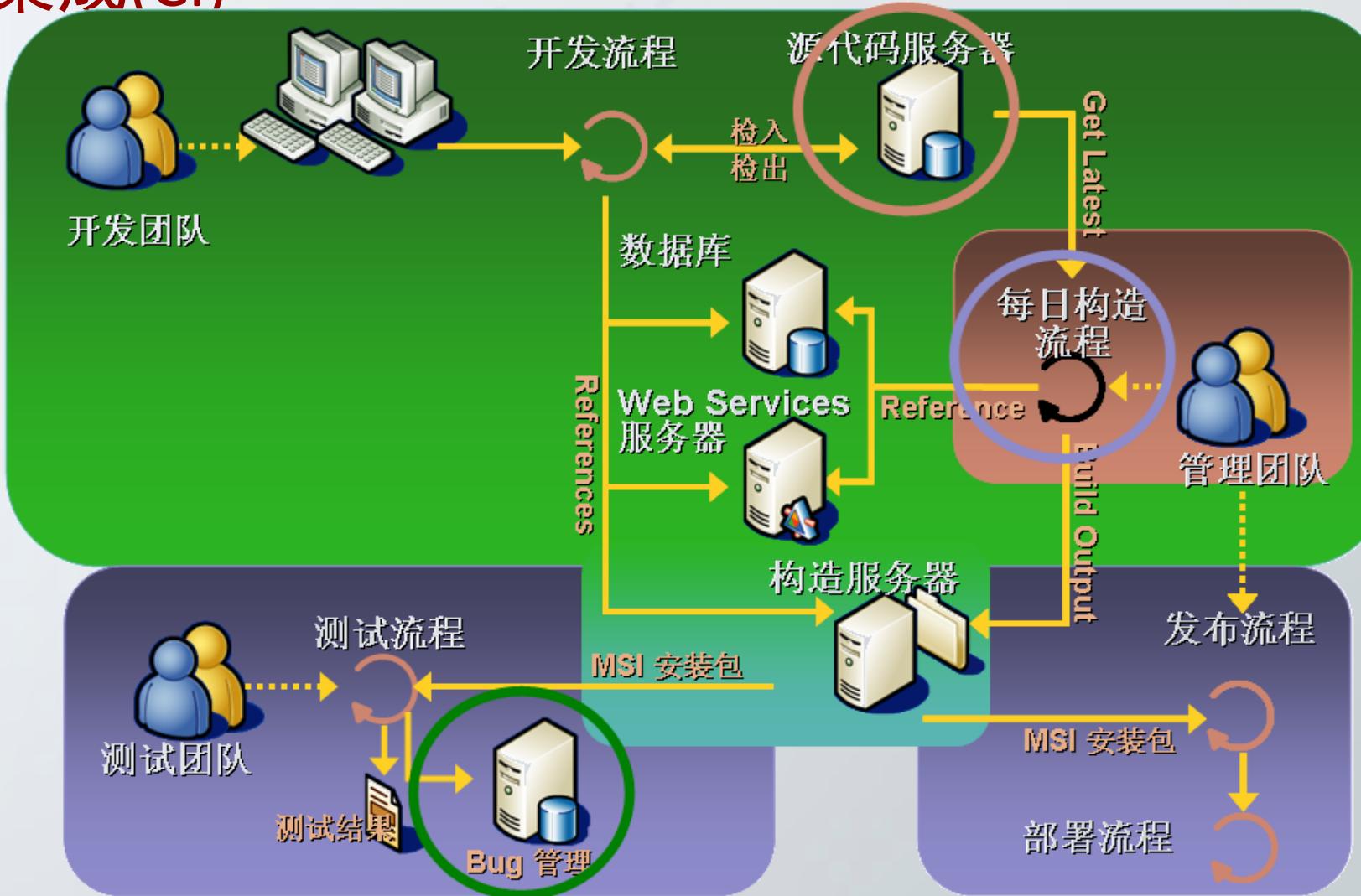
敏捷工程实践：持续集成(CI)

什么是持续集成

- 持续集成 (CI) 是一项软件开发实践，其中团队的成员经常集成他们的工作，通常每人每天至少集成一次，每次集成通过自动化构建完成。

持续集成的好处

- 大幅缩短反馈周期，实时反映产品真实质量状态；
- 缺陷在引入的当天就被发现并解决，降低缺陷修改成本；
- 将集成工作分散在平时，通过每天生成可部署的软件；避免产品最终集成时爆发大量问题。

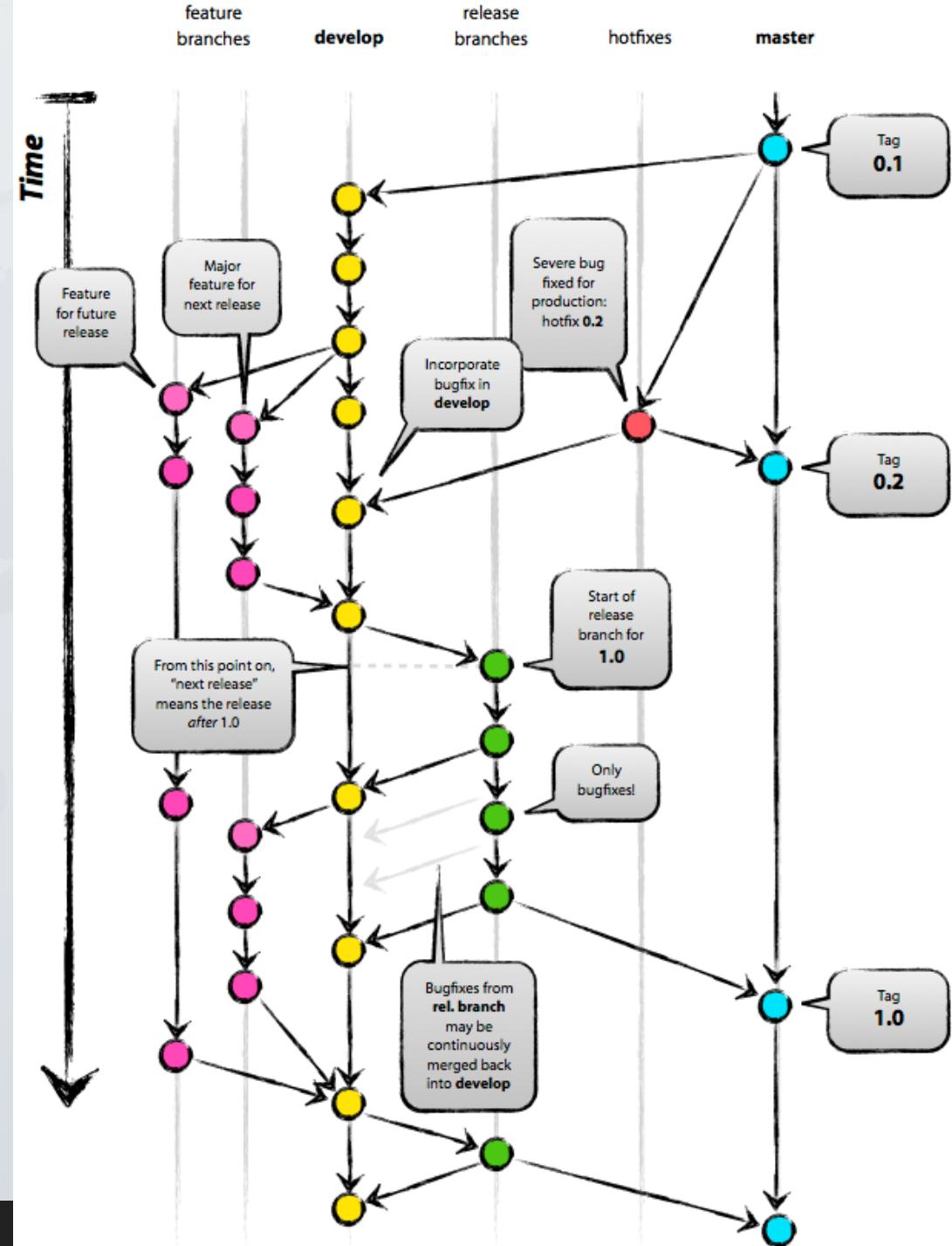


持续集成提供产品质量的快速反馈，保证随时拥有可工作的软件

敏捷工程实践：持续集成(CI)

持续集成的关键要点

- 持续集成强调“快速”和“反馈”，要求完成一次系统集成的时间尽量短，并提供完备且有效的反馈信息；
- 自动化测试用例的完备性和有效性是持续集成质量保障；
- 修复失败的构建是团队最高优先级的任务；
- 开发人员须先在本地构建成功，才可提交代码到配置库；
- 持续集成的状态必须实时可视化显示给所有人；
- 大系统持续集成需分层分级，建立各层次统一的测试策略。

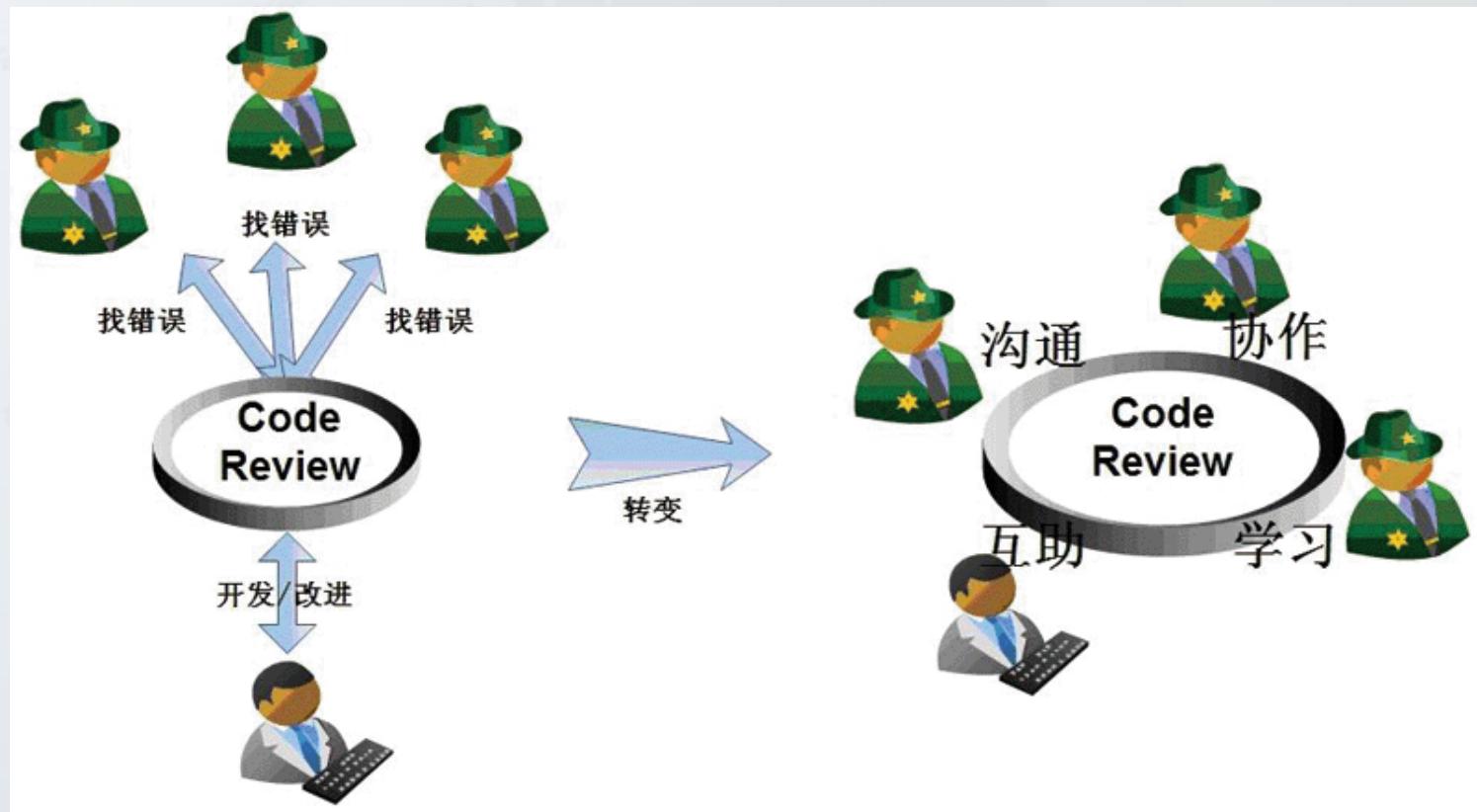


敏捷工程实践：Code Review

目的：持续的提高开发团队的工作质量

好处：

- 代码复查者(reviewer)能从他们的角度来发现问题并且提出更好的解决方案。
- 确保至少你团队的另一个其他成员熟悉你的代码，通过给新员工看有经验的开发者的代码能够某种程度上提高他们的水平。
- 公开reviewer和被复查者的想法和经验能够促进团队间的知识的分享。
- 能够鼓励开发者将他们的工作进行的更彻底，因为他们知道他们的代码将被其他人阅读。



在项目开发的过程中，25%的时间应该花费在code review上

敏捷工程实践：产品发布规则

每次迭代发布，移动互联网项目，自主性研发产品.....



多次迭代发布，传统项目、大型项目



训练要求

- 和身边的同学组成4-5人的项目小组，确定一位PO；
- 每个小组构想一款面向在校大学生用户群体的产品
 1. 定义至少10条以上产品特性的PB列表，每项特性以用户故事的格式表达；
 2. 对所有PB中的特性定义优先级和估算工时（人天数）；
 3. 以2周为迭代周期，定义第一个迭代周期的迭代PB，并将每项特性分解为任务，估算各项任务的工时（人时数），并分工到团队成员上；
 4. 模拟2-3天的例会进度，画出和跟踪燃尽图的变化；
 5. 以界面原型为工作成果，在评审会上演示工作成果；
 6. 召开回顾会，探讨流程中的可改进点（集体发言，并最终归类为：避免犯的错误，可以做的更好的经验，继续发扬的好做法）



THANKS