



# 敏捷开发方法与实践交流

张邱溪  
2011年4月

[www.asiainfo-Linkage.com](http://www.asiainfo-Linkage.com)



## 了解敏捷

( 为什么要使用敏捷方法? )

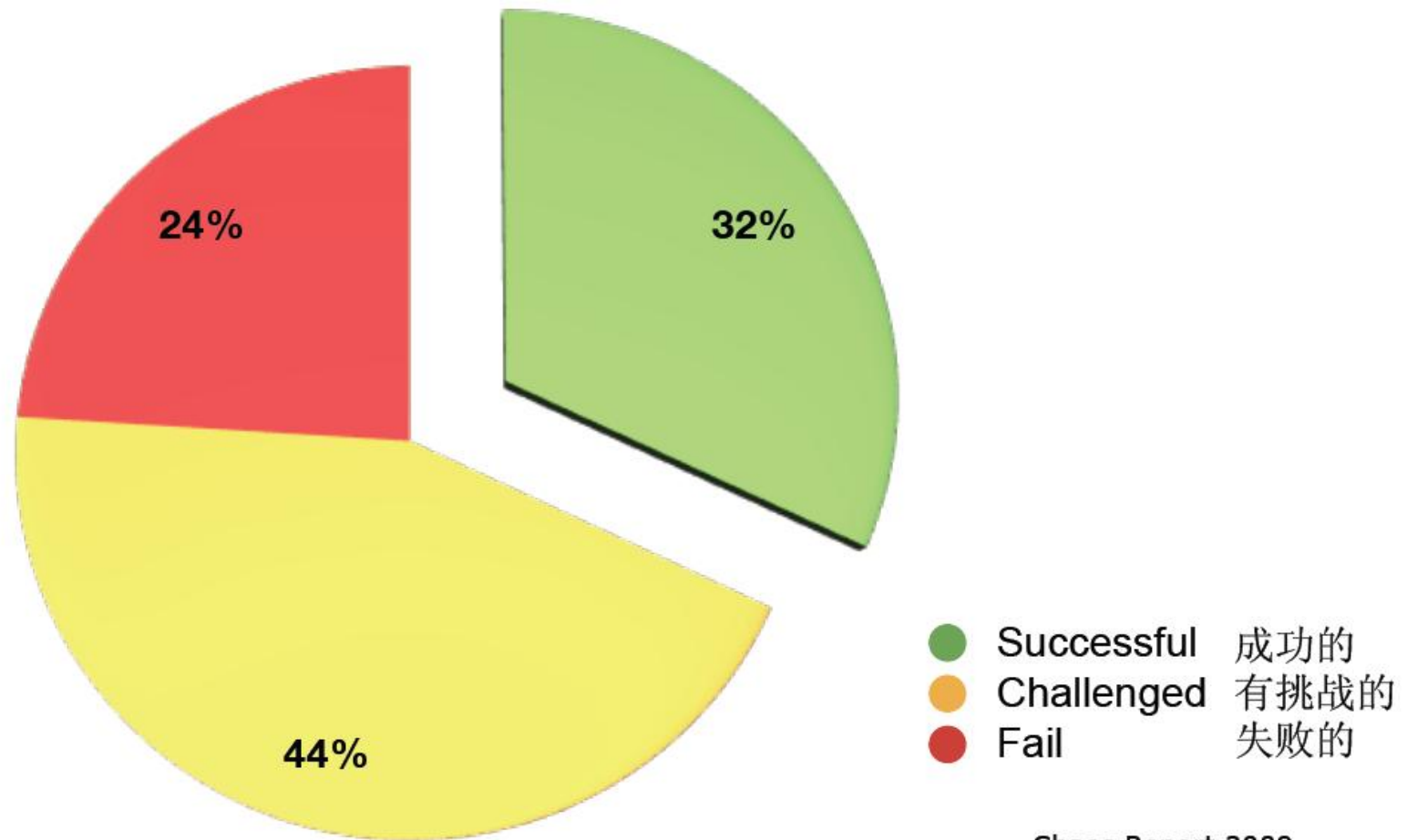
## 实践Scrum

( 什么是Scrum? 我们Scrum了么? )

## 问题探讨

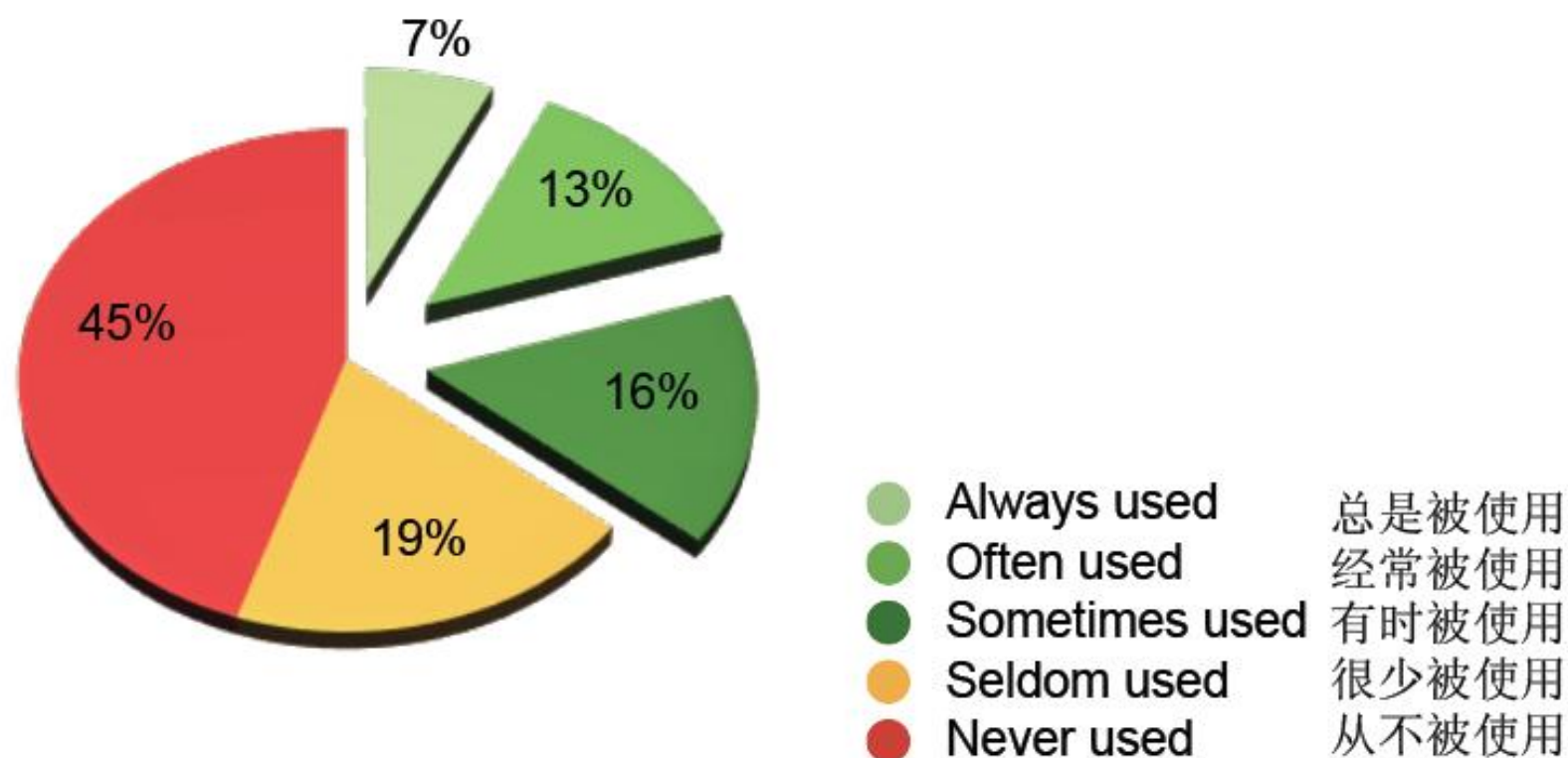
( 寻找属于我们的敏捷之道 )

# 项目成功率统计



Chaos Report 2009  
The Standish Group

## 即使软件交付了，也不意味着成功了

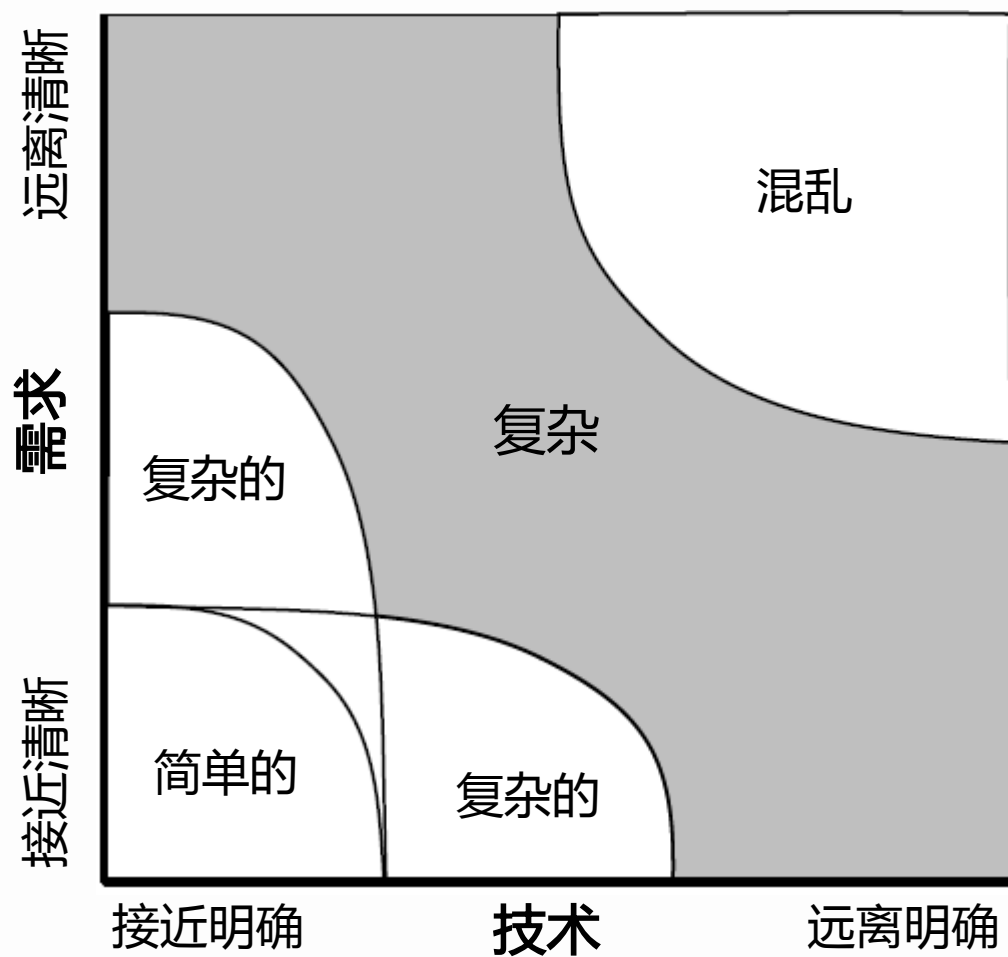


The Standish Group  
XP 2002



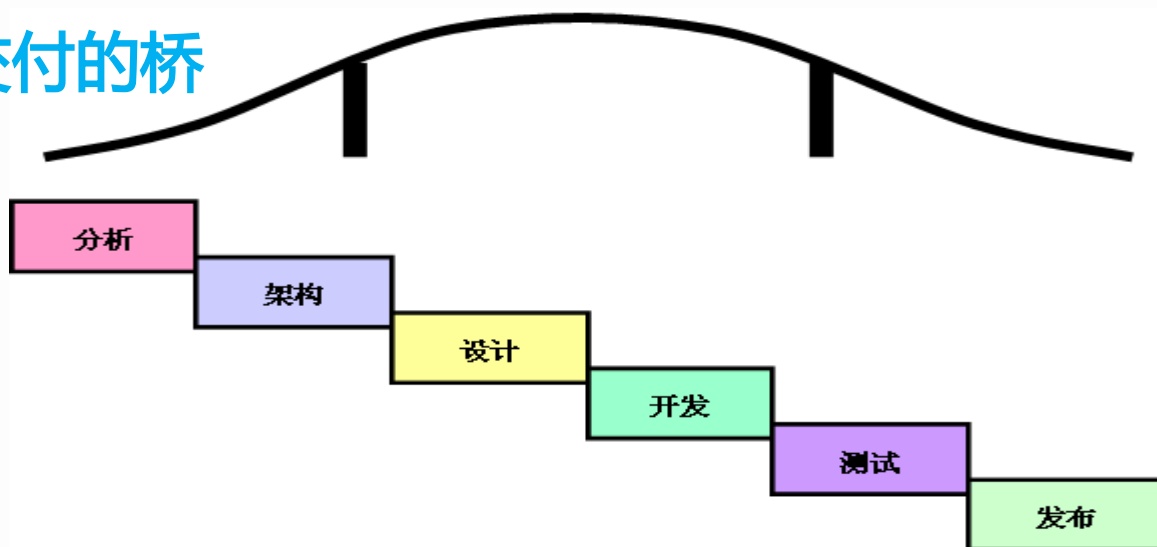
why ?

# 软件开发项目中的复杂性

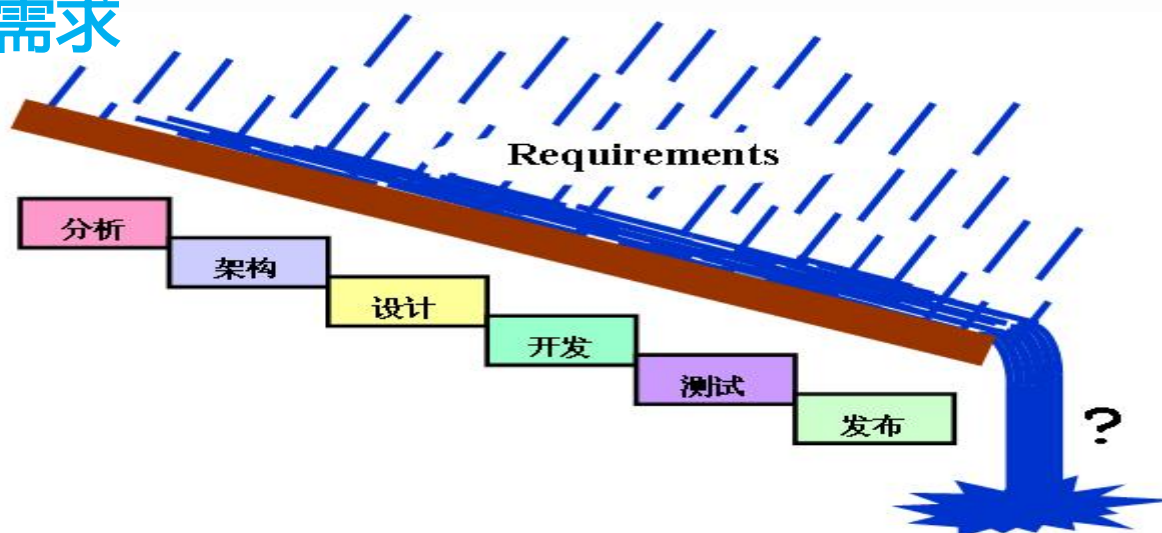


# 传统软件工程的做法

架一座通往按时交付的桥

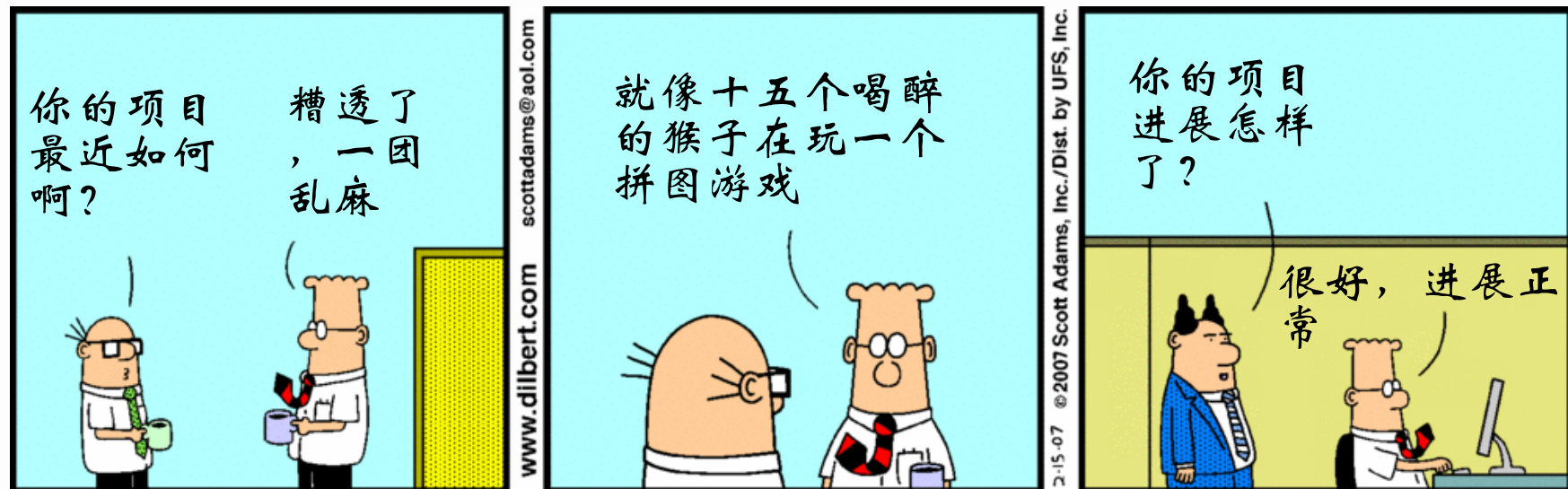


搭一个棚子来拒绝需求



拒绝改变

# 一个小漫画



问题隐藏





如果不能改变它，那就去适应它！

- 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

虽然右项也有价值，但是敏捷开发认为左项具有更大的价值！

也就是说：

自组织团队与客户紧密协作，通过高度迭代、增量式的软件开发过程响应变化，并在每次迭代结束时交付经过编码与测试的有价值的软件

**胜过**

与客户确定合同后在初期制定并遵循基于活动的完整计划，在重型过程和工具指导下，通过完成大量文档进行知识传递，最后交付需求

# 敏捷原则



- | 1、我们最优先要做的是通过尽早的、持续的交付**有价值的软件**来使客户满意。
- | 2、即使到了开发的后期，也**欢迎改变**需求。敏捷过程利用变化来为客户创造竞争优势。
- | 3、**经常性地交付**可以工作的软件，交付的间隔可以从几个星期到几个月，交付的时间间隔越短越好。
- | 4、在整个项目开发期间，业务人员和开发人员必须天天都在**一起工作**。
- | 5、围绕**被激励**起来的**个体**来构建项目。给他们提供所需的环境和支持，并且**信任**他们能够完成工作。
- | 6、在团队内部，最具有效果并富有效率的传递信息的方法，就是**面对面的交谈**。
- | 7、**工作的软件**是首要的进度度量标准。
- | 8、敏捷过程提倡**可持续的**开发速度。责任人、开发者和用户应该能够保持一个长期的、恒定的开发速度。
- | 9、不断地关注**优秀的技能和好的设计**会增强敏捷能力。
- | 10、**简单**是最根本的。
- | 11、最好的构架、需求和设计出于**自组织团队**。
- | 12、每隔一定时间，团队会在如何才能更有效地工作方面进行**反省**，然后相应地对自己的行为进行调整。

# 敏捷关注什么



## 对一系列敏捷项目的分析结果表明

产品上市时间平均减少了**69%**

成本降低了**57%**

工作量降低了**62%**

致命缺陷平均减少了**80%**

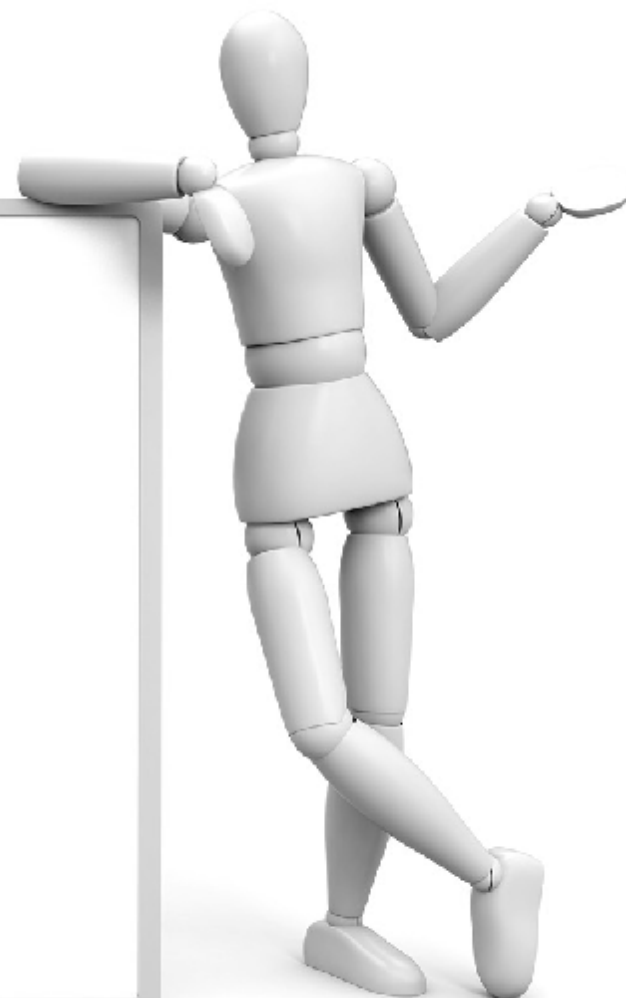
普通缺陷减少了**60%。**

员工士气提升

责任感增加

协调与合作促进

... ..



## 了解敏捷

( 为什么要使用敏捷方法? )



## 实践Scrum

( 什么是Scrum? 我们Scrum了么? )

## 问题探讨

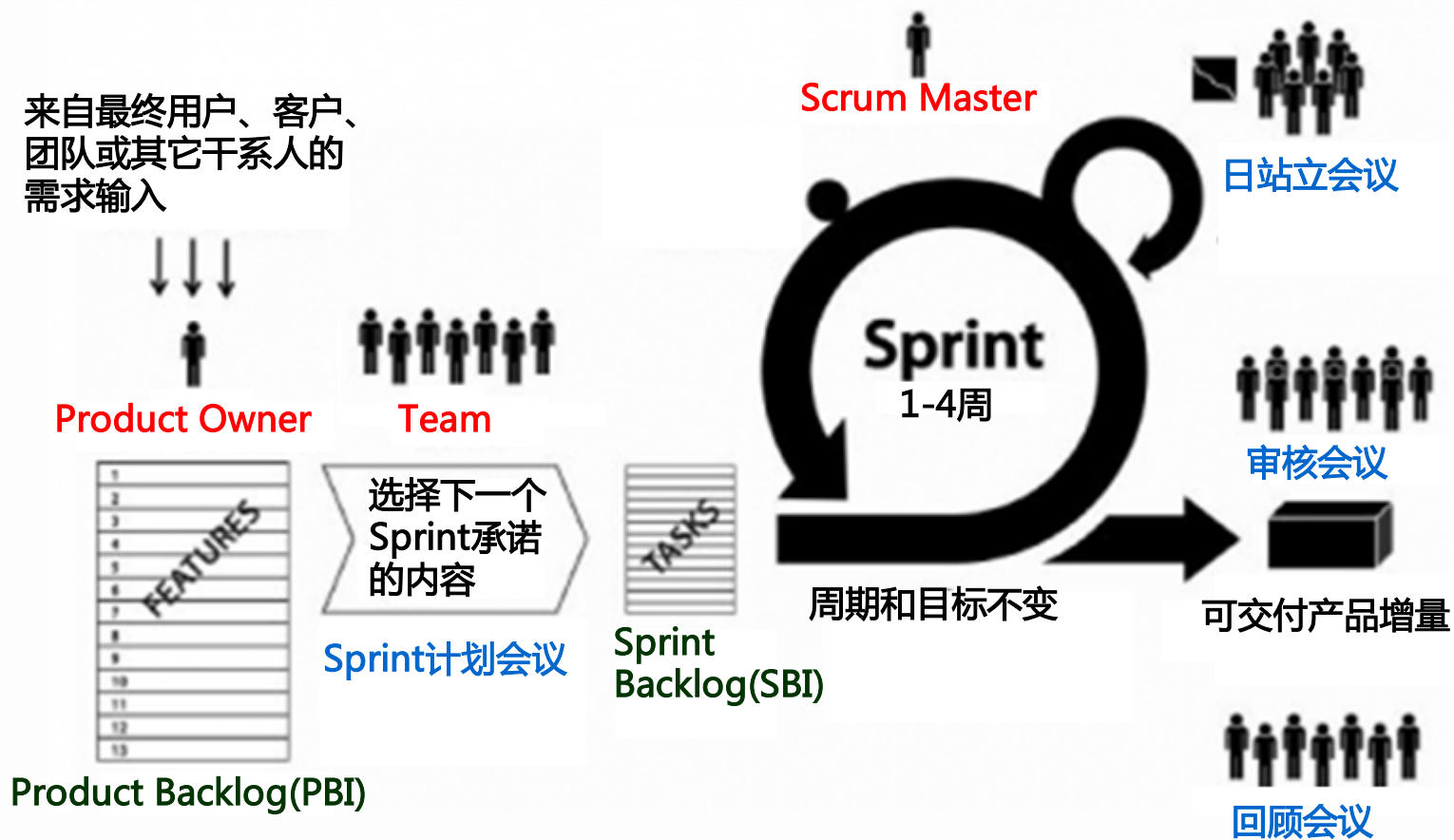
( 寻找属于我们的敏捷之道 )

# Scrum的起源



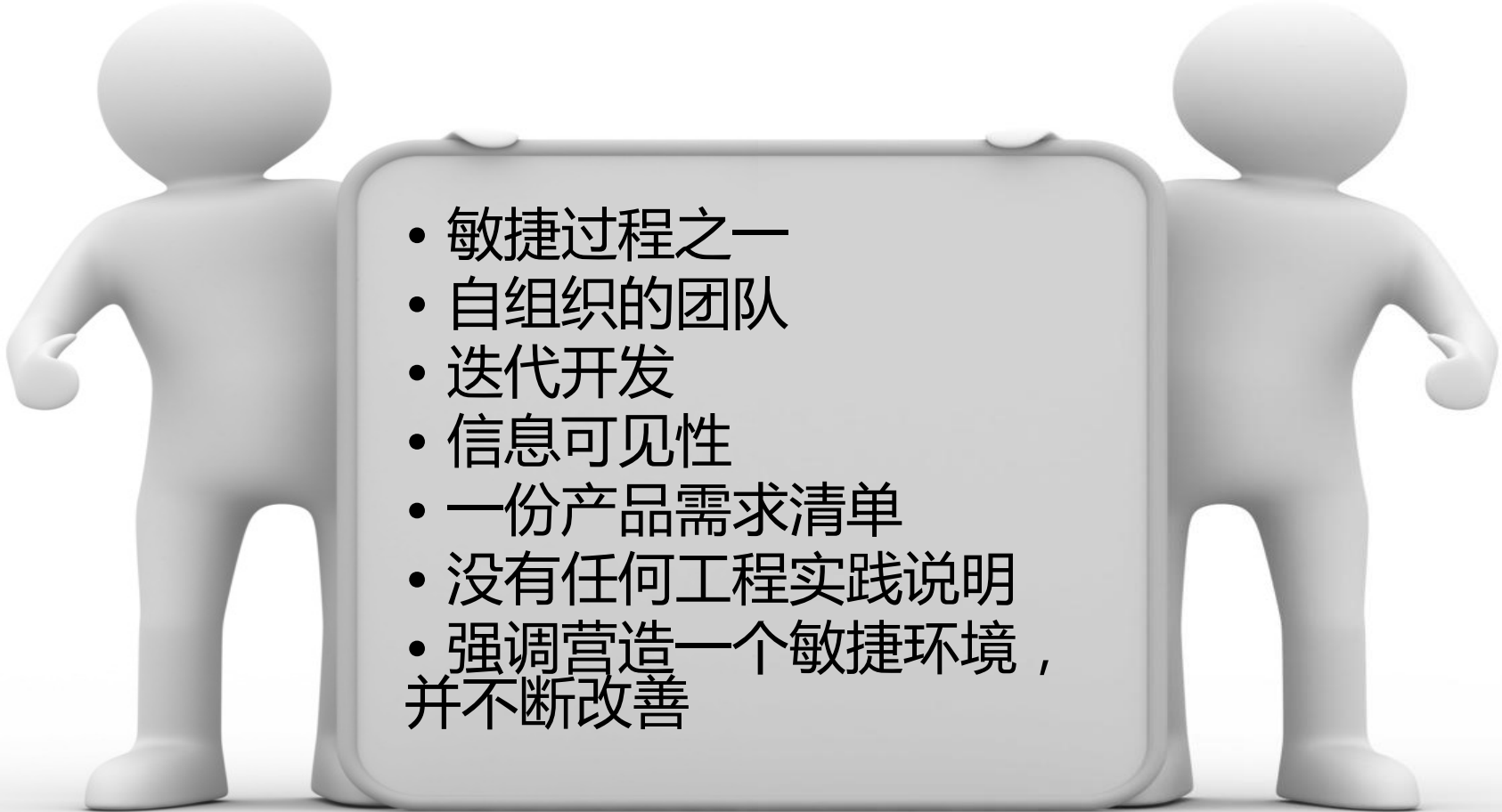


# Scrum的样子

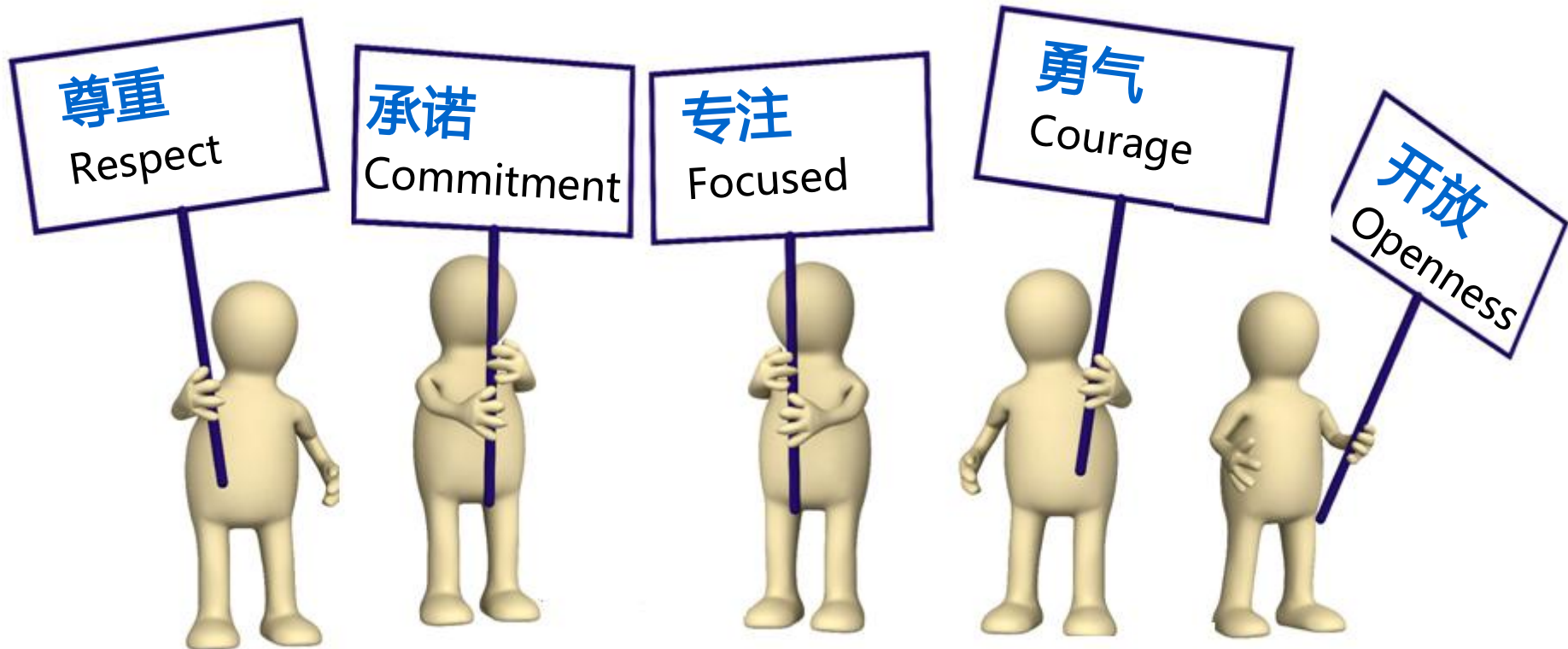




# Scrum的特征

- 
- 敏捷过程之一
  - 自组织的团队
  - 迭代开发
  - 信息可见性
  - 一份产品需求清单
  - 没有任何工程实践说明
  - 强调营造一个敏捷环境，并不断改善

# Scrum的五个价值观



# 越来越多的企业开始用Scrum





## 深入Scrum

# Scrum的3-3-5

## Roles 角色



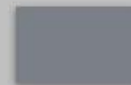
- Product owner (PO)
- ScrumMaster
- Team 团队

## Artifacts 工件



- Product Backlog
- Sprint Backlog
- Burndown Charts

## Ceremonies 仪式



- Sprint
- Sprint Planning
- Daily Scrum meeting
- Sprint Review
- Sprint Retrospective

# Product Owner(PO)



- 丨 定义产品的Feature(功能/特性)
- 丨 负责产品的利润(投资回报)
- 丨 根据市场及业务价值对产品特性排定优先级
- 丨 对发布日期和内容进行决策
- 丨 需要时，每一个迭代调整产品特性和优先级
- 丨 接收或者不接受工作的结果



谁是合适的Product Owner？没有Product Owner可以么？

传统的产品经理、专业的需求分析师可以是PO的人选，但是要完成角色的转变，担负起PO应有的职责，尤其是为用户和客户负责。一个Scrum Team不能没有PO，否则软件的业务价值将无从谈起。





## 案例&问题分析2



Product Owner对团队说他不能够参加这次的Sprint计划会议，因为要和一个合作伙伴公司的上层打一场重要的高尔夫球。他不介意团队计划独自进行会议。这样做可以么？



不可以，因为Sprint计划会议中，需要制定Sprint的目标，向Scrum团队宣贯Product Backlog，解答团队在User Story评估和任务拆分时遇到的业务问题，而这些都是离不开PO的参与的。



# 团队(Team)

- | 建议通常5-9人
- | 跨职能(团队、个人两个角度)
- | 成员最好是全职的
- | 团队是自组织的，理想是没有职称之分
- | 迭代之间，成员构成才做变更





原先的Team Leader、架构师、界面设计、专职测试职位还有存在的价值么？

理想的Scrum团队，除了Scrum Master、Product Owner和Team外，再没有别的职称之分了，Team中的成员是跨职能的，每个人都可以是架构师、设计师和测试人员。





- | 面向项目代表管理层
- | 负责Scrum价值观和过程的实现
- | 移除障碍
- | 确保团队的生产能力和全速前进
- | 促使所有角色和职能人员间的密切合作
- | 保护团队免于外部干扰
- | 服务式的领导



谁是合适的ScrumMaster？ScrumMaster可以取消么？

ScrumMaster倾向于从原先的开发团队中选择，比如Team Leader，或者是对敏捷有较多认识，较大兴趣的任何Team Member，也可以是传统的项目经理。这是一个需要思想和职责较大转换的角色。ScrumMaster原则上不可以取消，除非Scrum团队达到一个理想的自组织程度，而不再需要敏捷教练的指导。





### ScrumMaster需要有技术背景么？

Scrum框架中并没有要求ScrumMaster必须拥有技术背景，ScrumMaster的职责中更多也是体现协调能力、沟通能力。但是显然，拥有较深的技术背景对发现问题和解决问题，以及与技术出身的开发团队达成融洽方面更有帮助。特别是Scrum团队在技术方面比较欠缺，面临着很多技术问题。否则，如果一个技术成熟的团队，技术背景就不是那么重要了。





## 案例&问题分析6



一个项目中，一个9个人的Scrum开发团队中所有成员好像都在非常忙碌于实现他们对这个Sprint的承诺。而ScrumMaster好像比较空闲，没太多事情可以担心似的。有一位成员向ScrumMaster建议，要不你也在Sprint Backlog上找些开发的活干一下，和我们一期交付产品特性？

对于9个人规模的Scrum开发团队，要发现隐藏的问题、解决出现的问题，协调团队、避免干扰等等事务工作量是巨大的。如果ScrumMaster真的无所事事，要么就是这个Master没有尽职，要么就是团队自发组织的成熟度已经非常高。如果是后者，可以考虑有没有必要继续保留ScrumMaster的角色，否则的话，可以看看后面一页一个6年经验的Master所列举敏捷教练应做的工作内容，我们做到了么，做好了么？



# ScrumMaster应该干些啥？



## 做敏捷这个环境的建设者和守卫者

保持Scrum正常运转 保护团队免受干扰

团队的润滑剂

PO、团队、干系人力量平衡

帮助聚焦

协调与组织 帮助达成Sprint目标

与PO协调工作

帮助PO、团队、管理层和组织不断学习与进步

排除障碍

增加透明度

帮助大家提高自助能力

鼓励和确保自我组织

大力支持内部授权

提高沟通效率和有效性

发现隐藏问题并努力解决

帮助团队经验积累解决



ScrumMaster与PO可以合二为一由一个人承担么？

一般情况不可以，因为ScrumMaster与PO从职责上来讲是有着一定冲突的，PO代表的是最终用户与客户的意愿，要对用户负责，理想情况下，PO就是客户的一员。而ScrumMaster是团队的保护者，避免干扰，保证Sprint的顺利。特殊情况下，如果能把这两个角色都把握的很好，能很好的切换，且能够把两个角色的职责都做的很好，可以同一个人担当，但是不建议。





# Scrum的3-3-5

## Roles 角色



- Product owner
- ScrumMaster
- Team

## Artifacts 工件



- Product Backlog 产品Backlog
- Sprint Backlog
- Burndown Charts 燃尽图

## Ceremonies 仪式



- Sprint
- Sprint Planning
- Sprint Review
- Sprint Retrospective
- Daily Scrum meeting

# 产品Backlog(Product Backlog)



- | 需求-项目想要交付的工作清单
- | 一份合格的产品Backlog应该：
  - ü 已排序(按优先级)
  - ü 有价值的(业务价值)
  - ü 带估算的(需要投入多少工作量)
- | PO负责优先级排序
- | 在Sprint计划会议前PO可以调整排序

Card	卡片
Conversation	交谈
Confirmation	确认

As a [user role]  
I want to [result]  
[so that [reason]]  
作为一个 [用户角色]  
我想要 [结果]  
[因为 [原因] ]

## INVEST

**I**ndependent 独立的

**N**egotiable 可协商的（可伸缩的）

**V**alued 有价值的

**E**stimable 可估算的

**S**mall 小的

**T**estable 可测试的

### 史诗故事Epic VS 可以实施的故事

作为一个招聘者，我想使用  
这个招聘网站为我们的空缺职位  
寻找合适的候选人

作为一个招聘者，我想创建一个工作职位

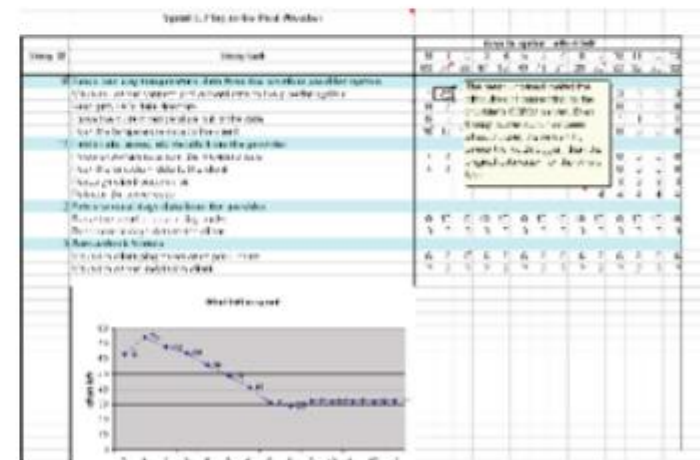
作为一个招聘者，我想审阅对  
工作职位有回应的人的简历

作为一个招聘者，我想在数据库  
中搜索具备合适技能的候选人

# 形式

Item #	Description	Est	By
<b>Very High</b>			
1	Final database structure	30	104
2	User list of spreadsheet shared (not to database)	5	104
3	Add Spreadsheet	-	-
4	Database user licensing	30	105
5	Server - Data Monitoring	30	105
6	Analysis Manager	100	105
7	File Transfer and support on web of data	100	105
8	Reporting Analysis	100	105
<b>High</b>			
9	Business analysis system	-	-
10	in main application	24	104
11	in report	24	104
12	Analysis Program	-	-
13	Details view	4	104
14	Analysis Manager	-	-
15	When data are received from an analysis, they should show up again in the job list in lower 1/2 of the analysis tab	5	105
16	Query	-	-
17	Support for wildcards when searching	10	104
18	Setting of master attributes to handle negative numbers	10	104
19	Horizontal scrolling	12	104
20	Population Overview	-	-
21	Processing Manager	400	104
22	Query Tool	400	104
23	Address of Editors (batch mode)	100	104
24	Query Manager	100	104
25	Reports	120	104
26	Add new file v1.1 to 2.0	-	-
27	Processing Manager	-	-
28	Variables (batch mode)	4	104
<b>Medium</b>			
29	Reports	-	-
30	Analysis Manager (only show system/analysis for report)	5	104
31	Details settings (%)	4	104

Story	To Do	In Process	To Verify	Done
And user, 1, 8 points	Code this, 9 Test this, 8	Code this, 8 Test this, 8	Test this, 8 SC 8	Code this, 8 Test this, 8 SC 8 Test this, 8 SC 8 Test this, 8 SC 8
And user, 1, 8 points	Code this, 8 Test this, 8	Code this, 8 Test this, 8	Test this, 8 SC 8	Test this, 8 SC 8 Test this, 8 SC 8





有团队人员抱怨说，用了用户故事后，需求描述变简单了，开发的时候根本不知道细节，该做成什么样，造成做完的功能不满足要求。是用户故事不合适么？

用户故事有三方面的体现，也就是常说的3C：卡片、交谈和确认。如果只看到卡片，那么用确实是非常简单和概略的，但是更多的，了解用户故事的详细情况是通过不断的交流和沟通，并通过较快得迭代落地实现的。这些内容是由PO所提供的。







- | 每个人认领Sprint中的任务
  - ü 工作不是分配下去的
- | 每天更新剩余工作的估算
- | 任何团队成员都可以添加、删除或者更改Sprint Backlog
- | Sprint中的工作会涌现
- | 如果工作不清楚，定义一个大工作量的Sprint Backlog 事项，留到以后再分解
- | 当我们更加了解我们的工作时，更新剩余的工作

# 创建Sprint Backlog

**Story 故事**  
**(PBI-Product Backlog Item)**

**Task 任务**  
**(SBI-Sprint Backlog Item)**

- 丨 从产品Backlog中选择在新的迭代中承诺完成的事项
- 丨 识别工作任务并估算工作量
- 丨 群策群力，不是Scrum Master一个人在做
- 丨 进行概要性的设计



Sprint进行当中，Sprint Backlog可以发生变化么？

可以，因为Sprint Backlog是一个个的任务Task，而随着开发进程的深入，有可能一些初期不明确的Task会浮出水面，或者一些冗余的Task可以丢弃。



# 燃尽图(Burndown Charts)

- | 每天更新
- | 跟踪剩余工作量
- | 不同的方式
  - | 跟踪任务层剩余工作量
  - | 跟踪完成情况



# 团队的速率(Velocity)

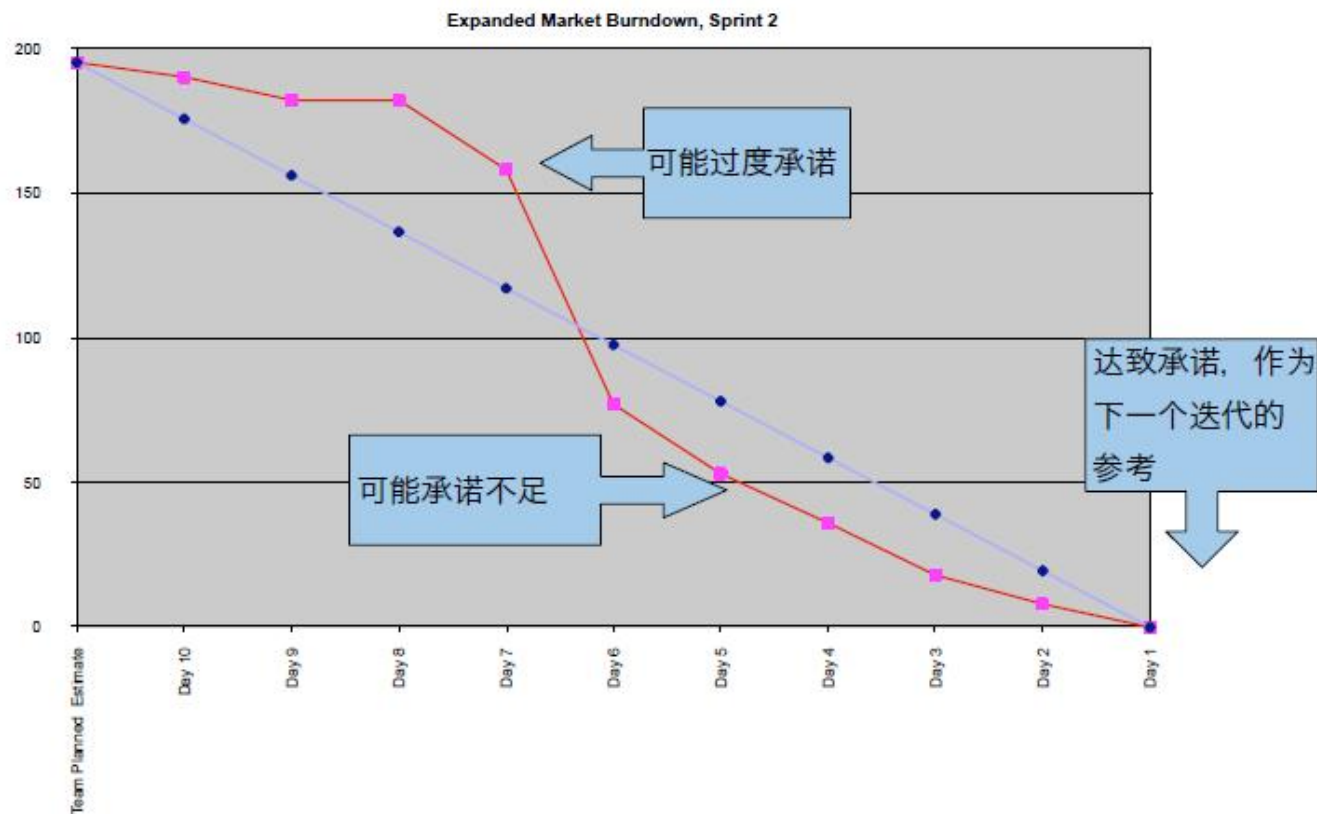
- | 团队 速率描述及预测我们一个迭代能完成多少工作
- | 你的团队迭代速率是多少？
- | 最理想的是历史数据
- | 开始时(第一个Sprint)更多是一个估算
- | 当一个Sprint完成，就有了历史数据
- | 使用真正的完成
- | 使用相对工作量

燃尽图燃烧的是什么？





## 案例&问题分析11



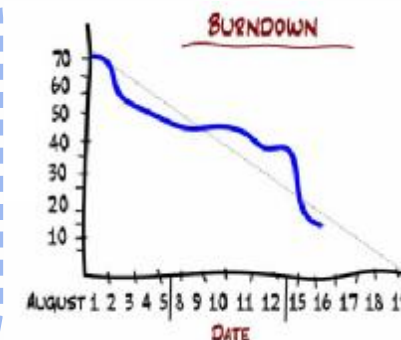


## 案例&问题分析12



一个团队的燃尽图看起来很完美，可是快到Sprint结束时，团队还没有交付任何可用的User Story，Why？

另外一个团队，虽然也按照交付了东西，可是用户现场反馈问题太多，根本没法上线。又出现了什么问题？



第一个团队的燃尽图纵坐标不是剩余时间就是剩余任务点，而这些都不代表UserStory的真正完成，比如Sprint中承诺实现10个UserStory，每个Story都只剩下最后的测试任务了，好像燃尽图接近尾声，实际任何Story都没交付。

第二个团队则是没有定义出什么是真正合理的“Done”，团队认为简单的测试就是交付了，实际上离“可用”还差的很远。





# Scrum的3-3-5

## Roles 角色



- Product owner
- ScrumMaster
- Team

## Artifacts 工件



- Product Backlog
- Sprint Backlog
- Burndown Charts

## Ceremonies 仪式



- Sprint
- Sprint Planning 计划会议
- Sprint Review 审核会议
- Sprint Retrospective 回顾会议
- Daily Scrum meeting 每日例会

# Sprint

- 丨 Scrum项目通过一个接着一个的Sprint来推进
- 丨 通常每个Sprint长度为2-4周，最长不超过一个月
- 丨 使用一个固定的Sprint长度，带来节奏感
- 丨 在Sprint中，涵盖产品的设计、编码、测试
- 丨 根据能够承诺一个Sprint跨度不做变更来计划长度
- 丨 严格的时间箱，成功 or 失败的标准





一位团队成员告诉ScrumMaster说Product Owner刚刚来请求她在当前的Sprint中加入一个产品Backlog事项。目前，这个Sprint大概进行了1/3。

在一个还未完成的Sprint阶段中，是不允许加入产品Backlog的。如果因为市场原因，必须调整纳入该Sprint的产品Backlog，则可以取消当前Sprint。重新进行一轮。另外，如果Sprint承诺的Backlog提前完成，且时间较长，可以考虑增加合适大小的产品Backlog。





能否取消一个Sprint ?



可以！如果商业上、业务上或者其它特殊情况出现，使得不得不调整一个Sprint已经承诺的Product Backlog，则可以取消一个Sprint，然后开始一个新的。



## 案例&问题分析15



有没有所谓的一个分析Sprint？  
有没有所谓的一个测试Sprint？  
有没有一个所谓的“稳定(收尾)”Sprint？  
如果一个项目需要比较多的基础架构工作，需要8周来实现，是否需要第一个为期8周的Sprint？

没有  
没有  
没有

还是没有！即使是基础架构的Sprint，也应该进一步拆分，在更短的Sprint里交付架构功能。把架构开发的风险降到最低。更好的选择是将架构开发与提供业务功能合并在一个Sprint中。





- | 所有Scrum成员都要参加
- | 必须有一个已划分优先级的Product Backlog
- | 第一部分，选择承诺：
  - ü 制定并宣贯Sprint目标
  - ü 对于不清晰的事项进行重新评估
  - ü 根据团队速率选择承诺Product Backlog事项
- | 第二部分，创建Sprint Backlog：
  - ü 拆分Product Backlog，形成Sprint Backlog
  - ü 回顾先前的承诺，看是否需要调整
- | 两周的Sprint，计划会议控制在4小时内

# 怎样做估算

小时 vs 故事点



计划扑克





计划会议究竟多长时间合适？我们团队一个计划会议要开两天，这正常么？计划会议计划出的Sprint Backlog总是估计不准，一天的变两天，两天的变一周...

一般情况，两周的Sprint计划会议4个小时，4周的8小时左右。如果会议时间太过冗长，可能是PO先前的工作没有准备到位，也可能是进行任务拆分时太过细化，也依赖于Scrum Master对会议节奏的控制。

Sprint Backlog估计不准，大多数情况是由于按照绝对时间估计，且开发人员的估计都相对乐观，考虑不够周全，可以改用相对故事点的方式估计。另外需要持续加强对估算能力的训练。





# 每日例(立)会 Daily Scrum

## I 执行

- ü 每天进行
- ü 15分钟
- ü 站立进行

## I 原则

- ü 不是问题解决会议
- ü 可以邀请他人，但只有Team、ScrumMaster、PO才能发言

## I 主题

- ü 昨天做了什么
- ü 今天要做什么
- ü 遇到什么障碍

## I 帮助避免其它不必要的会议



你们的Daily Scrum时间定的是每天9点，已经9点05了，你们的Scrum Master还没出现，你们应该怎样做？

应该按时召开每日立会，而不是等Scrum Master来组织，因为团队自组织是一个Scrum团队的特性。且通常情况下，Scrum Master是不应有太多发言的。



# 审核会议(Sprint Review)

- | 团队展示Sprint的成果
- | 不应该有长的准备时间，最多2小时
- | 不用幻灯片
- | 全员参与
- | 邀请所有利益干系人和感兴趣的人参加



# 回顾会议(Sprint Retrospective)

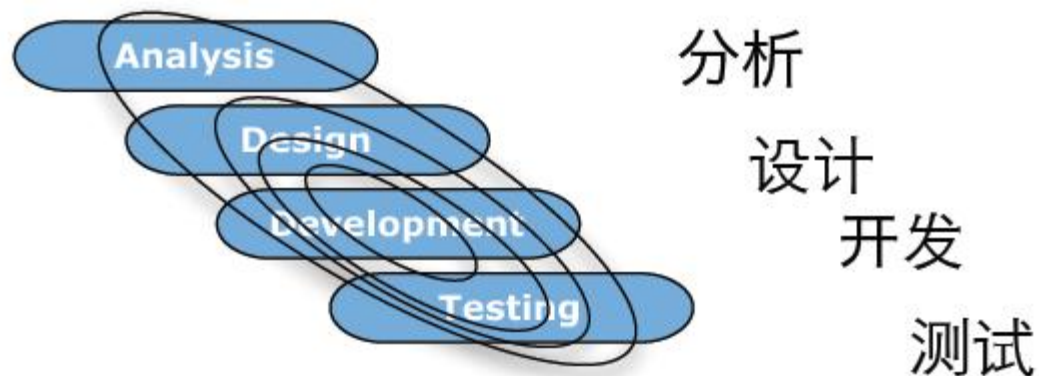


- | 持续改进
- | 简短
- | 每一个Sprint结束后进行
- | 全员参与
- | 可邀请感兴趣的人参与，但只有团队发言
- | 三个建议内容
  - ü 开始做什么
  - ü 停止做什么
  - ü 继续做什么



## 总结

## 什么是完成-Done ?



## 水平 VS 垂直

UI	UI	
控制器	Controller	
模型	Model	
持久	Persistence	

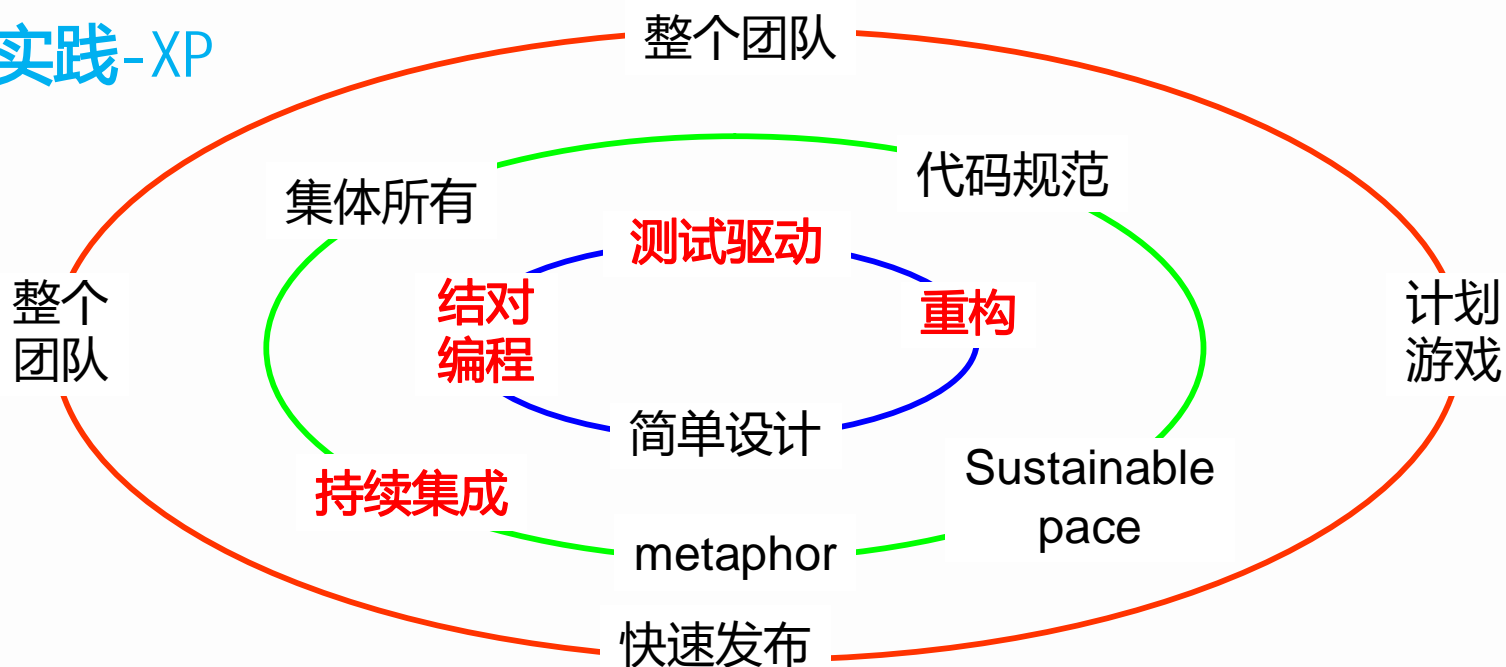
F1	F2	F3	F4	F5	F6	F7	F <sub>n</sub>

# Scrum 与 XP

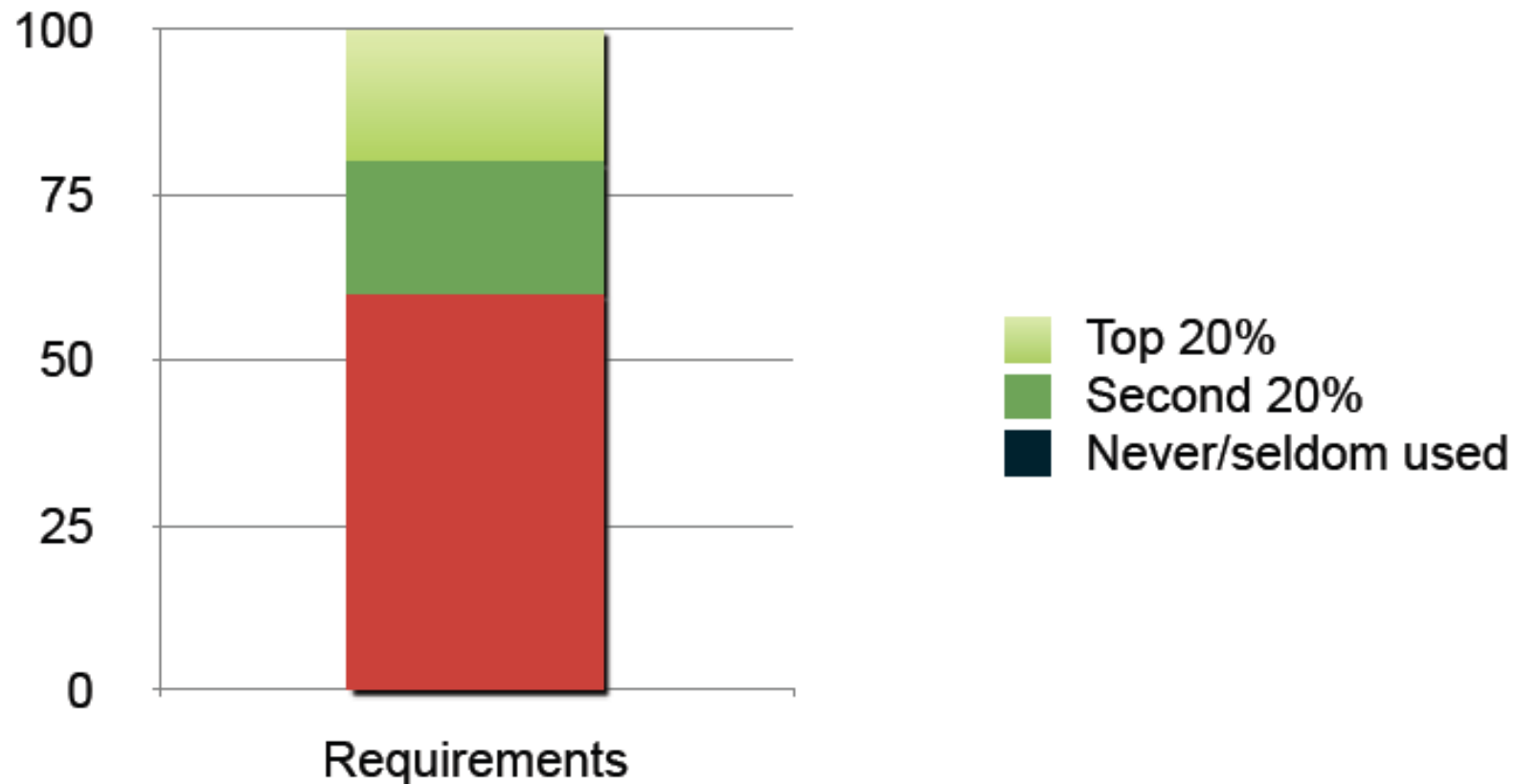
## 管理框架-Scrum



## 工程实践-XP

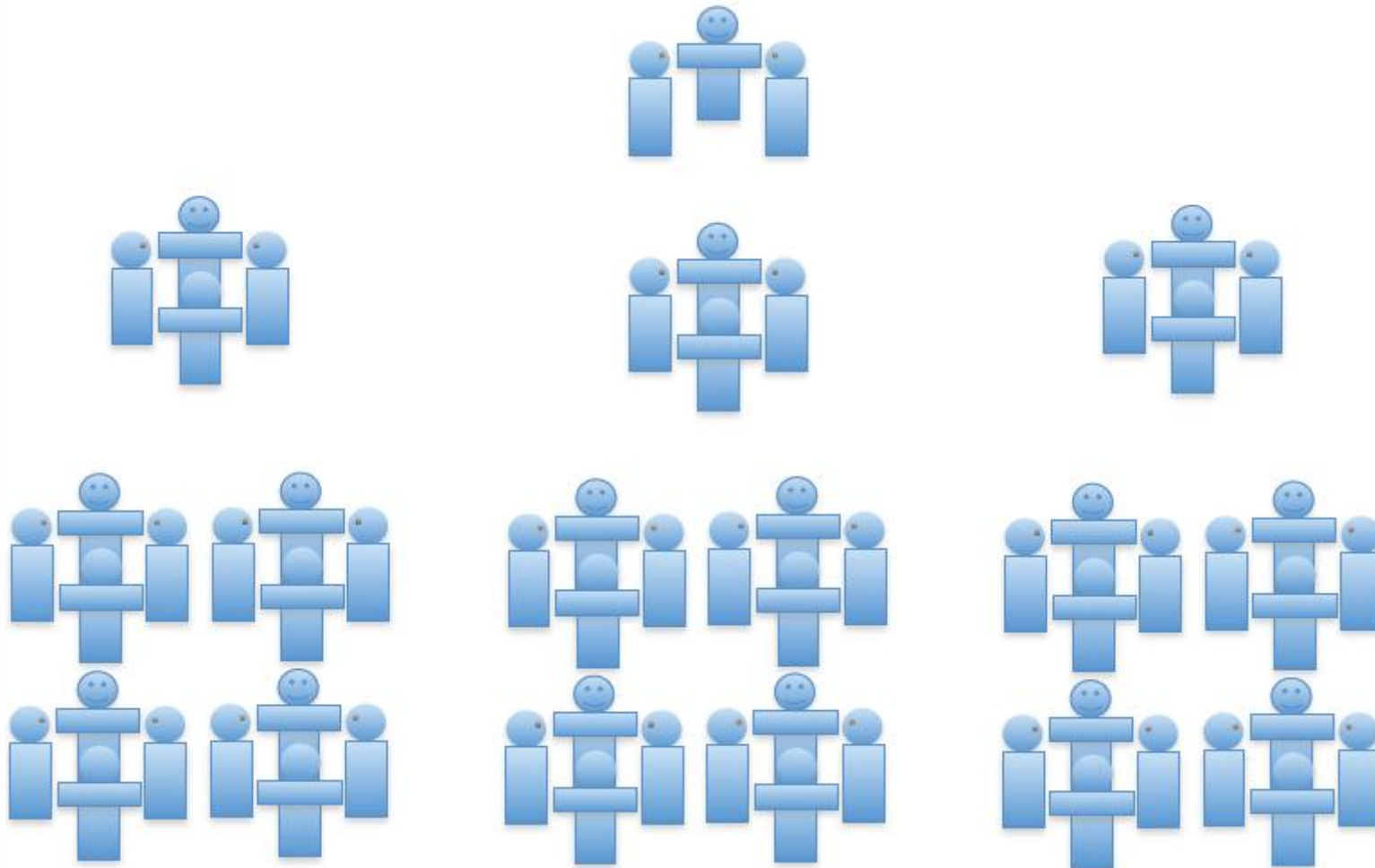


# 柏拉图原则(80/20 Rule)





# 大型团队Scrum Of Scrum



# 敏捷是关于人的

Courage, Openness, focus, Commitment, Respect

角色转换，而不是人员变更

我要做，而不是要我做

更有智慧，而不只是更有知识



# Scrum认证

Foundation-Level

Mid-Level

Professional-Level

Guide-Level



## 了解敏捷

( 为什么要使用敏捷方法? )

---

## 实践Scrum

( 什么是Scrum? 我们Scrum了么? )

---



## 问题探讨

( 寻找属于我们的敏捷之道 )

# 反馈的问题1



如果把敏捷开发比喻成盖房子的话，思想是蓝图，技术是基石，管理是具体工序。没有蓝图，就没有秩序。基石不好，房子很快会坍塌。

一些团队进行敏捷开发时，往往只有敏捷管理这一方面，而非常重要的思想和技术则不涉及，并且管理也只是流于形式。你们遇见过这个问题么？如何解决的？

首先，非常赞同这个比喻。提出的问题也是目前大多数敏捷团队所出现的。

关于敏捷的思想，我想是需要通过自下而上的兴趣、学习、交流与自上而下的支持、变革、推广相结合。同时，也是一个逐渐的过程，与每个人的努力都息息相关。

关于敏捷的技术，尤其是TDD、持续集成、自动测试等敏捷开发必备技术，在这里也强烈建议引入使用，否则，距离真正的敏捷恐怕还有很远的距离。





在敏捷开发过程中，如何处理外来干扰对敏捷过程的干扰，比如现场的工程需求和工程故障，这些外来干扰往往是不可预知的。



作为Scrum Master，很大的工作任务就是避免外界干扰对Scrum团队的影响，从部门乃至公司来讲也应该给Scrum团队一个相对稳定的环境。理想情况，严重的缺陷应该在上一个Sprint解决而不会出现在当前Sprint，工程需求应该是纳入下一个Sprint处理。实际情况如果没法避免，如果影响工作量太大，建议重新开始Sprint。但这绝不应该成为常态。

## 反馈的问题3



在产品敏捷开发过程中，前期控制比较稳定，但是当产品进入集成测试阶段，往往过程很难控制，有可能导致产品延期发布。

正如前面的案例所分析，燃尽图在Sprint前期乃至接近尾声都一切正常，但是到发布日期了，却交付不出可用的特性。一方面，集成测试不应该是Sprint中的一个阶段，而是一个特性做好了就要立刻进行测试，另一方面，也需要提高单元测试、集成测试的能力及自动化程度。使得问题尽早暴露。





产品在敏捷开发过程中，如果出现需求不清，或者设计不到位的情况，导致需要返工，这种情况如何处理。



需求不清楚，最主要的原因是PO职责不到位，没有在Sprint开始之前把用户故事及其优先级别准备妥当。另外也需要团队成员在有疑问时及时与PO沟通交流。

设计不到位，则需要提高团队设计能力和健全设计评审机制，应该与敏捷方法关系不大。





在当前迭代无法完成的需求，是不是可以转移到下迭代完成，如果不能，怎样避免出现这种无法完成的需求？

可以，而且在不成熟的Scrum团队，可能会经常出现。但这种情况下，意味着承诺的Sprint目标以及User Story没有完成，也就是Sprint失败了。





如何处理在迭代过程中不断补充或者增加的TASK，这些任务往往是迭代开始无法明确的，随着开发的深入，会逐步显现。



这也是一个常见的现象，也是一个正常的情况。但是为了使Sprint计划会议时对用户故事以及拆分后Task任务的评估更加准确性，还是需要整个团队提高问题的分析能力，Scrum Master的问题发现能力，尽量提前发现隐藏的任务点，以加强Sprint的可控性。

# 开放式讨论





AsialInfo Linkage

Thanks

[www.asiainfo-linkage.com](http://www.asiainfo-linkage.com)