# Scrum敏捷项目管理

# 目录

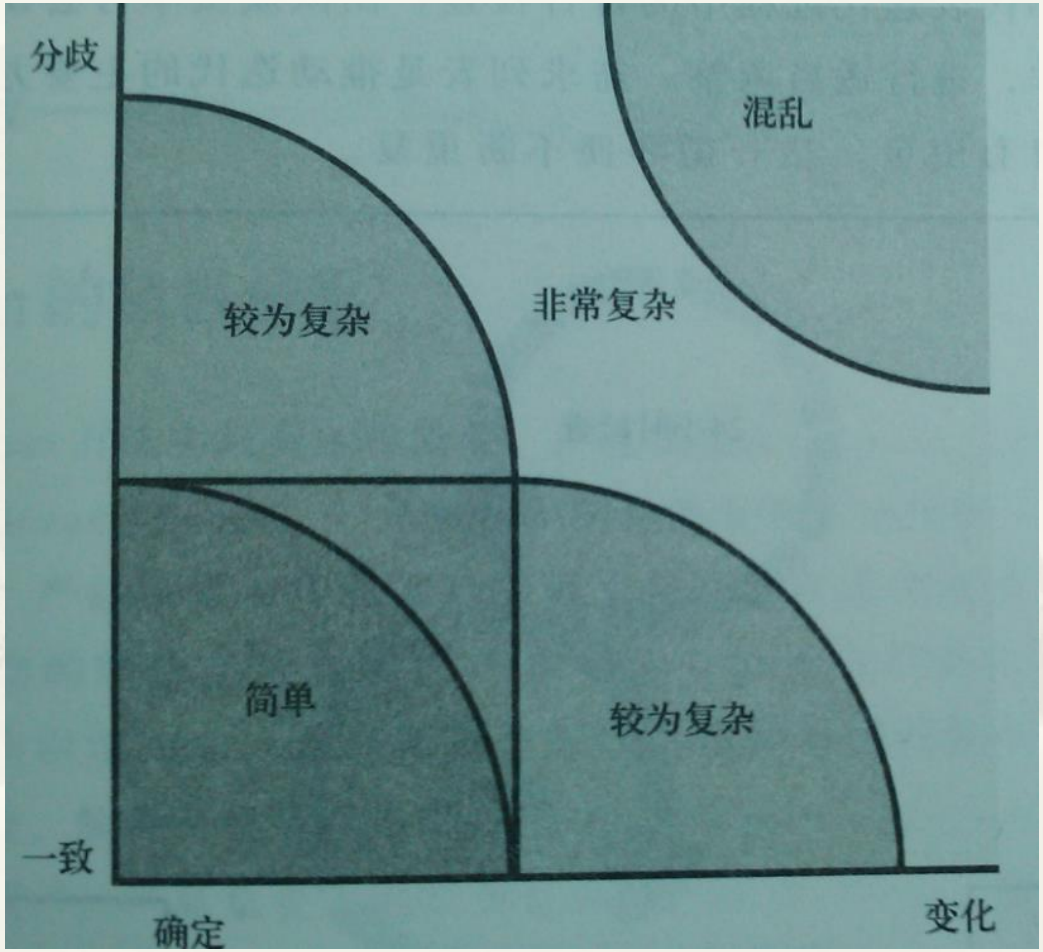

* *敏捷的背景与动机*
* 敏捷宣言及原则
* 敏捷方法是什么？
* 敏捷方法的实践
* Scrum的角色
* Scrum流程和工作产品
* Scrum应用
* 总结


# 敏捷的背景与动机

- 软件危机及软件工程的出现
- 速度是企业竞争致胜的关键因素，软件项目的最大挑战在于
  - 一方面要应付变动中的需求
  - 一方面要在紧缩的时程内完成项目
- 传统的软件工程难以满足这些要求
- 所以软件团队除了在技术上必须日益精进，更需要运用有效的开发流程，以确保团队能够发挥综效。这正是Agile Process（敏捷的软件开发流程)于近年来兴起的主要原因。

# 软件项目的复杂性

- 横轴代表需求的复杂度
- 纵轴表示技术的复杂度
- 还有人力资源的复杂度



# 解决复杂性问题需要采用经验式方式

- 解决问题的两种方式：
  - 预定义过程控制（富士康流水线生产）
  - 经验性过程控制（摸着石头过河）
- 如果复杂度超过预定义方式的能力范围，应该采用经验性方式
- 经验性方式的三大支柱：可见性、检查及适应

# 他山之石

- 互联网时代的出版模式
  - 作者最开始的时候并没有想出一本书，而只是把多年的积累梳理出来写成了博客，凭借博客的成功最后得到了出版商和纸版读者的认可。在写成本书的过程中，作者是渐进式的进行的，每写完一个章节，放到博客上去征求读者的反馈，很多反馈意见在后面的章节或修订中及时地体现出来，这样就形成了与读者之间的良好反馈，在出版之前就锁定了大量的读者。
  - 这就是敏捷开发提倡的“增量迭代、及时交付”的思想。
  - 这种模式能最大程度地不偏离客户需求的本质。
- 精益制造
  - 消除浪费、关注流程、建立无间断流程以快速应变 、降低库存 、一次做对、基于顾客需求的拉动生产 、标准化与工作创新 、尊重员工，给员工授权 等

# 目录

- 敏捷的背景与动机
- ***敏捷宣言及原则***
- 敏捷方法是什么？
- 敏捷方法的实践
- Scrum的角色
- Scrum流程和工作产品
- Scrum应用
- 总结

# 敏捷的历史

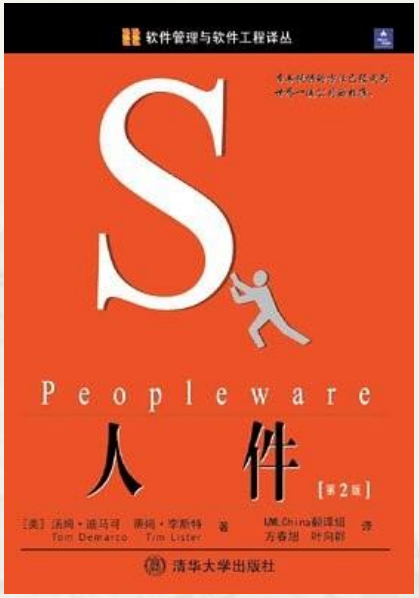
- 敏捷软件开发又称敏捷开发，
  - 从1990年代开始逐渐引起广泛关注的一些新型软件开发方法，是一种应对快速变化的需求的一种软件开发能力。
- 2001年初，因观察到许多的软件团队身陷不断扩大的流程之中的困境，一群业界专家聚集在一起，勾勒出一些能让软件团队迅速工作，以及响应变化的价值观和原则。他们自称为Agile Alliance。
- 之后的七个月里，他们创造具有价值的声明，也就是敏捷软件的开发宣言。
- 十五人中包括：大名鼎鼎的Kent Beck（XP,TDD的创始人,Junit的创始人之一）、Ward Cunningham（Wiki概念的发明者）、Martin Fowler（《企业应用架构模式》作者）、Robert C. Martin、Ken Schwaber

# 敏捷价值观之敏捷宣言


**敏捷开发的核心思想是：以人为本，适应变化。**

# 敏捷价值观之敏捷宣言-1

* 个体和交互胜过过程和工具
  * 人是软件项目获得成功最为重要的因素
  * 合作、沟通能力以及交互能力比单纯的软件编程能力和工具更为重要
  * 方法和工具是死的，人是活的，人要是太“面”或者协作不好，再强大的方法和工具都是白扯；



# 敏捷价值观之敏捷宣言-2

- 可以工作的软件胜过面面俱到的文档
  - 过多的面面俱到的文档往往比过少的文档更糟
  - 软件开发的主要和中心活动是创建可以工作的软件
  - 直到迫切需要并且意义重大时，才进行文档编制
  - 编制的内部文档应尽量短小并且主题突出

# 敏捷价值观之敏捷宣言-3

* 客户合作胜过合同谈判
  * 客户不可能做到一次性地将他们的需求完整清晰地表述在合同中
  * 为开发团队和客户的协同工作方式提供指导的合同才是最好的合同

# 敏捷价值观之敏捷宣言-4

* 响应变化胜过循环计划
  * 变化是软件开发中存在的现实
  * 计划必须有足够的灵活性与可塑性
  * 短期的迭代的计划比中长期计划更有效

# 敏捷开发的12个原则

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。

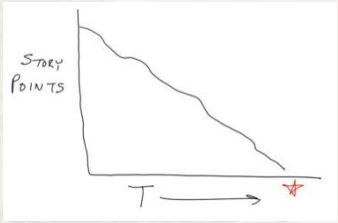

3. 经常性地交付可以工作的软件，交付的间隔可以从几周到几个月，交付的时间间隔越短越好 。
4. 在整个项目开发期间，业务人员和开发人员必须天天都在一起工作。

5. 围绕被激励起来的个人来构建项目。
6. 在团队内部，最具有效果并且富有效率的传递信息的方法，就是面对面的交谈。

# 敏捷开发的12个原则

7. 工作的软件是首要的进度度量标准。


8. 敏捷过程提倡平稳的开发节奏；发起人、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. 简单化是根本（不做过度设计和预测）。

11. 最好的构架、需求和设计出自于自组织的团队。

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思并对自己的行为进行相应调整。

# 目录

- 敏捷的背景与动机
- 敏捷宣言及原则
- ***敏捷方法是什么？***
- 敏捷方法的实践
- Scrum的角色
- Scrum流程和工作产品
- Scrum应用
- 总结

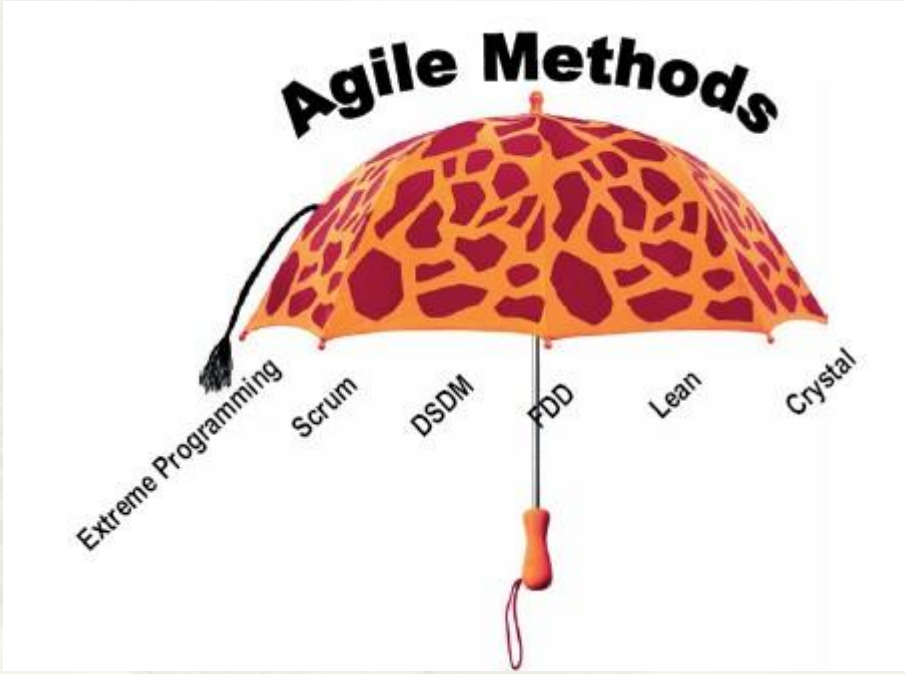
# 什么是敏捷方法？

- 敏捷方法是一类软件开发流程的泛称；
- 敏捷方法是相对于传统的瀑布式软件过程提出的；
- 敏捷方法可以用敏捷宣言（4条）、敏捷原则（12条）来概括；
- 敏捷原则通过一系列的敏捷实践来体现出来；
- 敏捷方法有很多种。

问题1，请举出你知道的2个以上敏捷方法名字来

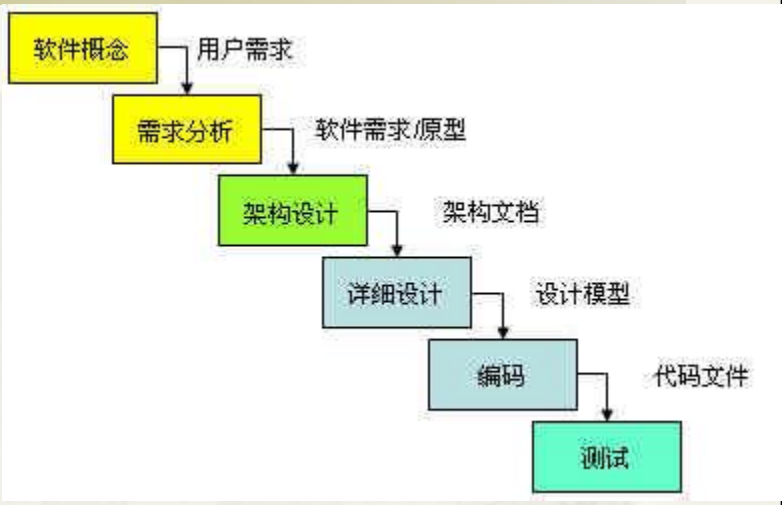
# 敏捷的方法

- Extreme Programming (XP) 极限编程
- Scrum
- Adaptive Software Development (ASD)自适应软件开发
- Crystal Clear and Other Crystal Methodologies水晶方法
- Dynamic Systems Development Method (DSDM) 动态系统开发方法
- 等



# 敏捷方法 VS. 瀑布模型

* 瀑布模型
  * 固定的、没有弹性的。
  * 很困难去达到互动。
  * 假如说需求没有完全的被了解，或是可能需要完全地改变项目的需求，瀑布式的model是比较不适合的。
* 敏捷方法
  * 完整地开发，每少数几周或是少数几个月里可以测试功能。
  * 强调在获得最简短的可执行功能的部分，能够及早给予企业价值。
  * 在整个项目的生命周期里，可以持续的改善、增加未来的功能。



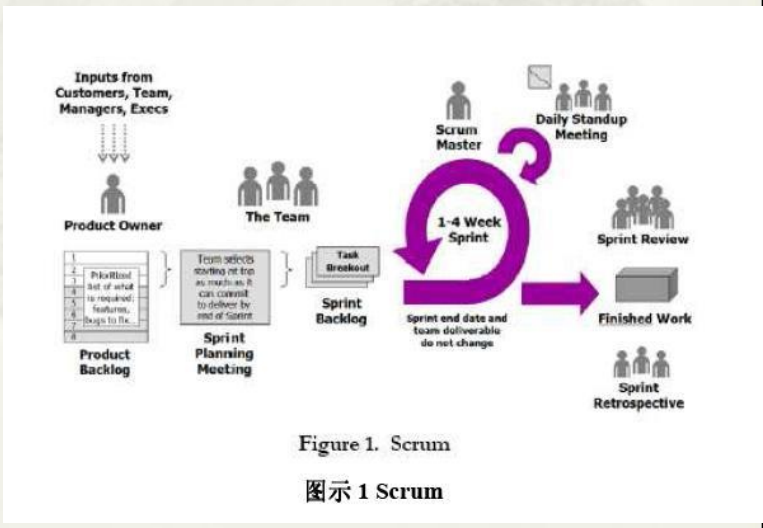



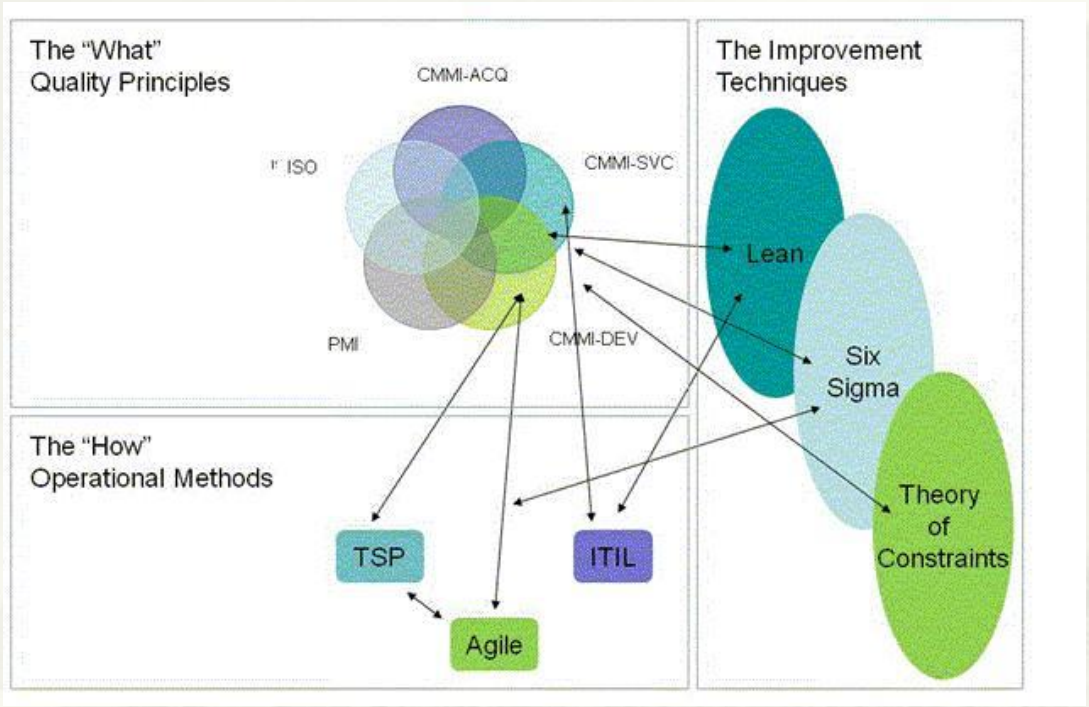

Figure 1. Scrum

图示 1 Scrum

# 敏捷项目管理VS传统项目管理

- 传统项目管理：
  - 事先对整个项目进行估计、计划、分析
  - 反对变更; 变更需要重新估计、重新规划
  - 严密的合同来减少风险, 如果改变需求要走 CR 流程.
  - 项目作为一个“黑盒子”，对客户与供应商的可视性差.
  - 文档和计划驱动的方法.
  - 软件交付时间晚, 意识到风险的时间晚.
  - WBS,甘特图,关键路径分析
- 敏捷项目管理:
  - 对整个项目做一个粗略的估计, 每一次迭代都有详细的计划.
  - 鼓励变化, 客户价值驱动开发.
  - 信任和赋予权力;合约使变更变得简单，增加价值.
  - 客户和开发人员之间是紧密的连续的合作关系
  - 每次迭代都产生可交付的软件
  - 专注于交付软件.
  - 第一次迭代就可交付能工作的版本，风险发现的早.

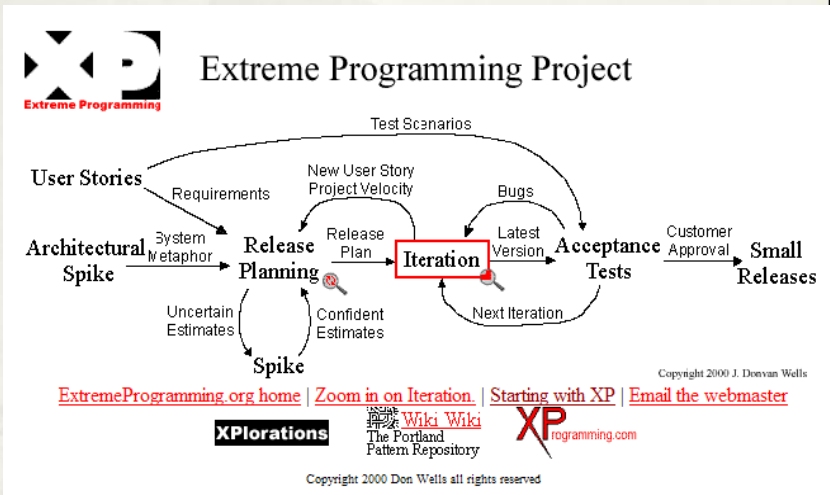
# 敏捷 与CMMI双剑合璧




- CMMI更加关注于流程，敏捷更加关注于人
- CMMI自顶向下，敏捷自底向上
- 敏捷并不排斥必要的文档
- 敏捷的很多实践是对CMMI的一种实现，比如sprint计划会议就是PP的实现，每日例会就是在做PMC
- 很多CMMI4~5级的公司也在应用敏捷，比如说宝信、华为
- 项目级的敏捷实践通过CMMI可以在组织级得以重用

# eXtreme Programming

- XP我们一般称为极限编程，是最轻量级的开发流程。
- 最主要的精神是
  - 『在客户有系统需求时，给予及时满意的可执行程序』，所以最适合需求快速变动的项目。
- 它强调客户所要的是
  - workable的执行码，所以把与撰写程序无关的工作降至最低，并要求客户与开发人员最好以side-by-side的方式一起工作。
- XP的实践包括：
  - 完整团队、计划游戏、客户测试
  - 简单设计、结对编程、测试驱动开发
  - 改进设计、持续集成、集体代码所有权
  - 编码标准、隐喻、可持续的速度

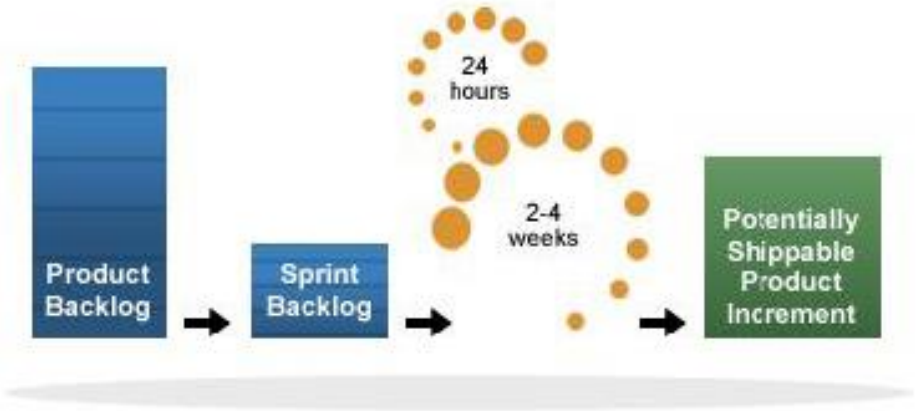


# Scrum开发流程

**一个轻量级的软件开发方法**

Scrum是一个敏捷开发框架，是一个增量的、迭代的开发过程.。在这个框架中，整个开发周期包括若干个小的跌代周期，每个小的的跌代周期称为一个Sprint，每个Sprint的建议长度2到4周。在Scrum中，使用产品Backlog来管理产品或项目的需求，产品backlog是一个按照商业价值排序的需求列表，列表条目的体现形式通常为用户故事。Scrum的开发团队总是先开发的是对客户具有较高价值的需求。在每个Sprint中，Scrum开发团队从产品Backlog中挑选最有价值的需求进行开发。Sprint中挑选的需求经过Sprint计划会议上的分析、讨论和估算得到一个Sprint的任务列表，我们称它为Sprint backlog 。 在每个迭代结束时，Scrum团队将交付潜在可交付的产品增量。

**一个简单的框架**



# 为什么采用敏捷? –预期的收益

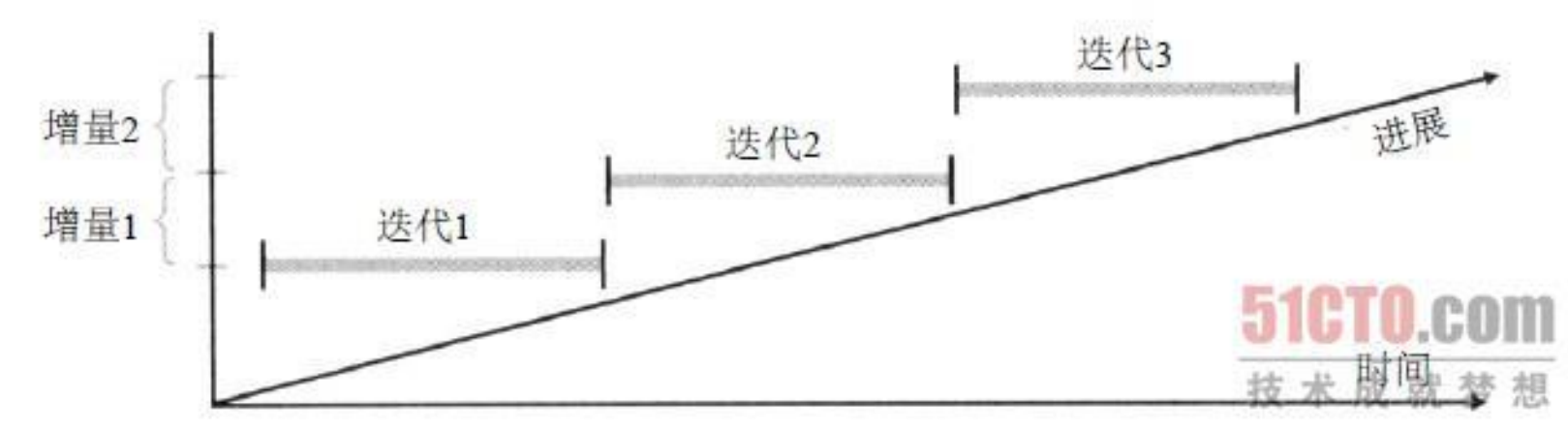
- 采用敏捷方法得当的话，可以：
  - 更加透明; 随时跟踪项目的状态和进展情况，及早发现问题和风险 .
  - 快速交付, 每次迭代都能交付可运行的软件.
  - 最高风险和最高优先级的需求，最优先进行开发.
  - 改善应对变更能力, 减少大量的重计划.
  - 改善项目沟通.
  - 更好的客户参与, 避免错误的假设.
- 总之:
  - 提高了生产率; 减少“浪费”（不需要的文档，重复工作等），项目的每次迭代都有明确的目标.
  - 提高客户满意度; 短期内产生成效, 按预期交付软件, 每次迭代结束产生可以运行的软件.
  - 改善员工的满意度; 团队精神，减少官僚，能够规划和管理自己的工作，减少“恐慌”，稳定的工作量（可持续的步伐）.

# 目录

* 敏捷的背景与动机
* 敏捷宣言及原则
* 敏捷方法是什么？
* ***敏捷方法的实践***
* Scrum的角色
* Scrum流程和工作产品
* Scrum应用
* 总结

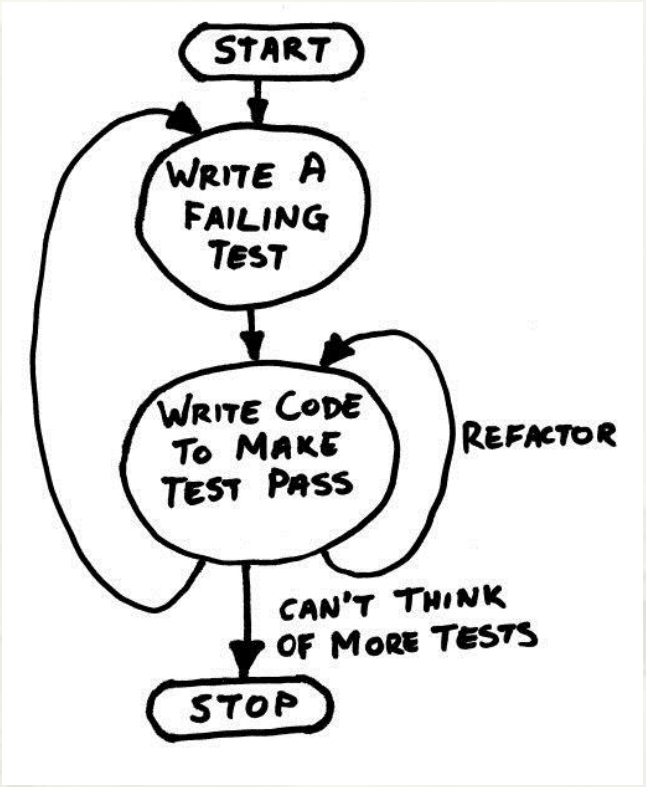
# 敏捷关键实践1——增量迭代

- 每个迭代有一个大约为1～4周的时间框，在SCRUM里称为一次冲刺（超过1个月的详细计划往往偏差很大）
- 每次迭代都应该有明确的目标
- 每次迭代都应该有明确的可演示的工作成果
- 迭代过程中项目团队应该尽量免受打扰
- 迭代可以将项目的压力分解到每个小的阶段，风险也能同时分解



# 敏捷关键实践2——测试驱动开发 TDD

- 什么是测试驱动？
  - 首先创建测试用例，然后开发软件通过测试(在开发代码前，首先编写测试代码)
- 一种设计软件的方法，而不仅仅是一种测试方法
- 所创建的测试用例用来指导和约束项目中的各项工作，对未来的各项工作提供一个安全的保护
  - 不需要测试的工作不需要完成
- 所创建的测试用例通常替代详细的业务和技术需求定
- 测试也有效地驱动设计，使设计更加趋向于可行的设计
- 通常情况下需要自动测试的支持 (EUnit, JUnit etc.).
- 对于UI软件应用TDD方法有一定的困难



# 敏捷关键实践3——持续集成

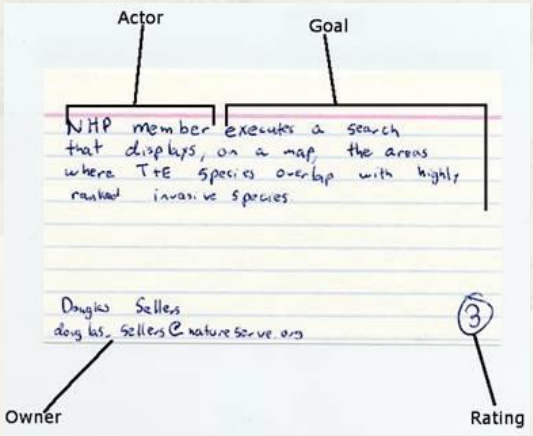
- 极限编程称为“每日构建”
- 持续集成一般利用ANT、MAVEN等工具
- 日构建的好处：
  - 将集成风险降到最低
  - 降低质量风险
  - 提升士气
- 日构建可以看做是项目的心跳，冒烟测试就像是听诊器
- 日构建必须至少：成功编译、打包、发布；不含有任何明显的缺陷；通过冒烟测试

# 敏捷关键实践4——面对面交流

- 虽然如今通讯工具花样繁多，但面对面交流在某些场合下仍然是不可替代的；
- 敏捷开发把交流缺失问题考虑在内，要求团队成员彼此直接协作，尽量创造面对面交流的机会；
- 尤其当业务分析师和软件开发人员一起工作的时候，面对面的交流是很重要的。
- 匿名共享需求文档只会打开曲解和误解之门，更不用说书面信息比口头交流还要慢很多。

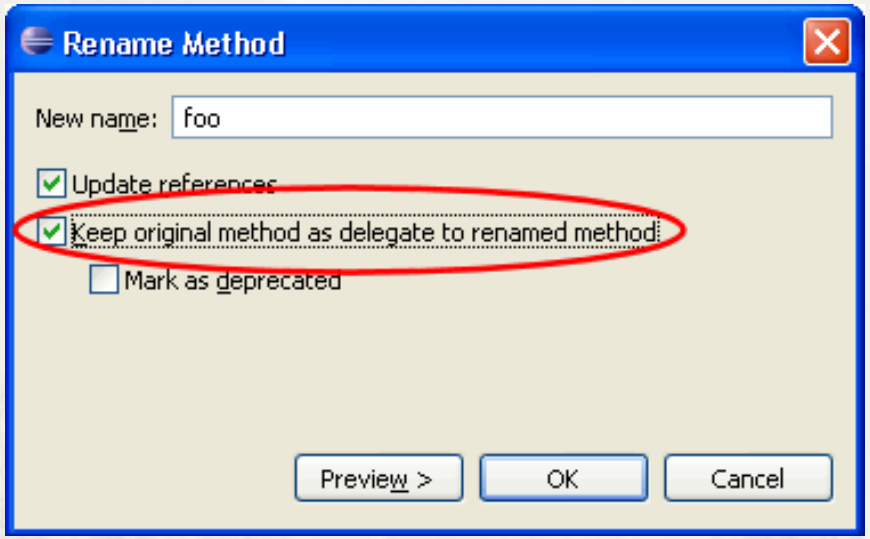
# 敏捷方法的其它实践

* 结对编程
* 每日立会
* 用户故事
* 团队工作室
* 频繁发布
* 自组织团队
* 重构

# 重构——改善既有代码的设计

- Martin Fowler提出
- 代码的坏味道
  - Martin Fowler和Kent Beck列举了22种坏味道：冗余代码、冗长的方法、巨大的类、过多的参数等等
- 重构可以弥补设计的不足
  - 简单设计的思想
- 重构与测试驱动的关系
  - TDD是重构的脚手架
- IDE已经对主要的重构模式提供了自动化支持：Rename, extract method, move field等等
- 简单设计>>测试用例>>实现再说>>（重构>>回归测试）*



# Scrum何时更有效？

- 公司和客户一致认为应当使用敏捷方法，双方都能理解敏捷方法.
- 敏捷方法对需求不完整以及经常变换的项目比较有效.
- 项目可以划分成固定时间间隔的迭代, 并且可以冻结正在进行的迭代的范围
- 公司和客户都有能力担当角色尤其是Product Owner 和 Scrum Master.
- 项目的人员结构能够分成6到10人的团队，最好每个工作地点一个小组.（Scrum of Scrums，Scrum的扩展）
- 团队成员能够以自组织的方式工作.
- 项目的合同允许变更.
  - 固定价格的项目可以使用敏捷，但应当尽量避免。
    - 最好在按时间和材料付费或者按月付费的项目中进行使用、
  - 变更项目的范围不需要高级管理层的批准.

问题2，为什么SCRUM团队人员最好在10人以内？

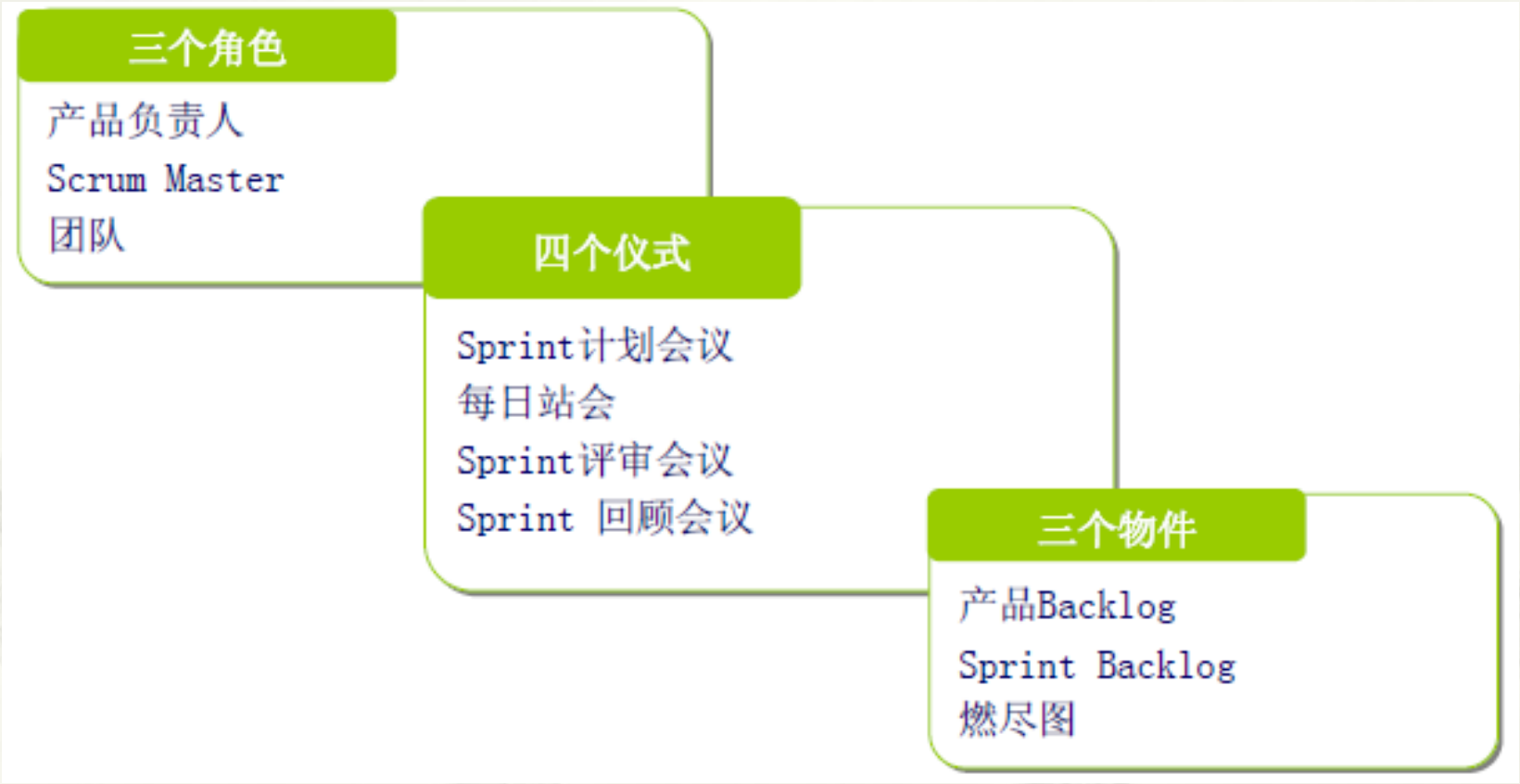
# 目录

* 敏捷的背景与动机
* 敏捷宣言及原则
* 敏捷方法是什么？
* 敏捷方法的实践
* ***Scrum的角色***
* Scrum流程和工作产品
* Scrum应用
* 总结

# 敏捷特别强调人的因素

* 相对于过程与工具，敏捷更强调“人”的因素。
* 诚信是基础
* 没有过程能够对诚信进行有效的约束
* 诚信与否是有效实施敏捷过程的最大限制

# Scrum框架


三个角色
产品负责人
Scrum Master
团队
四个仪式
Sprint计划会议
每日站会
Sprint评审会议
Sprint 回顾会议
三个物件
产品Backlog
Sprint Backlog
燃尽图

# Scrum角色

## 产品负责人

产品负责人是利益相关方的代表，他的工作重点是产品的业务方面。他负责向团队介绍产品远景。他负责给出一份明确的，可度量的，合理的产品 Backlog，并从业务角度出发对 Backlog 中各项问题按优先级排序。

## 团队

团队尽一切可能去完成任务——发布产品。团队需要全面的能力，这意味小组内拥有实现产品的全部技术和技能。团队还需要充分的理解产品负责人所描述的产品愿景以及 Sprint 目标，以更好地支持可能需要进一步开发的产品的发布。

## Scrum Master

Scrum Master 是整个团队的导师和组织者，他负责提高团队的开发效率。他常提出培训团队的计划，列出障碍 Backlog。Scrum Master 控制着检查和改进 Scrum 的周期，他维护这一团队的正常运行，并与产品负责人一起让利益相关方获得最大化投资回报。他关心的是这些敏捷开发思想是否能得到利益相关方的理解和支持。

# Scrum角色之Product Owner

* 产品负责人（Product Owner）的职责如下：
  * 确定产品的功能。
  * 决定发布的日期和发布内容。
  * 为产品的profitability of the product (ROI)负责。
  * 根据市场价值确定功能优先级。
  * 每个Sprint，根据需要调整功能和优先级（每个Sprint开始前调整）。
  * 接受或拒绝接受开发团队的工作成果。

# Scrum角色之ScrumMaster

- 作为Team Leader和Product owner紧密地工作在一起，他可以及时地为团队成员提供帮助。他必须:
  - 保证团队资源完全可被利用并且全部是高产出的。
  - 保证各个角色及职责的良好协作。
  - 解决团队开发中的障碍。
  - 做为团队和外部的接口，屏蔽外界对团队成员的干扰。
  - 保证开发过程按计划进行，组织 Daily Scrum, Sprint Review and Sprint Planning meetings。

# Scrum角色之Scrum Team

- 一般情况人数在5-9个左右
- 团队要跨职能
  （包括开发人员、测试人员、用户界面设计师等）
- 团队成员需要全职。
  （有些情况例外，比如数据库管理员）
- 在项目向导范围内有权利做任何事情已确保达到Sprint的目标。
- 高度的自我组织能力。
- 向Product Owner演示产品功能。
- 团队成员构成在sprint内不允许变化。

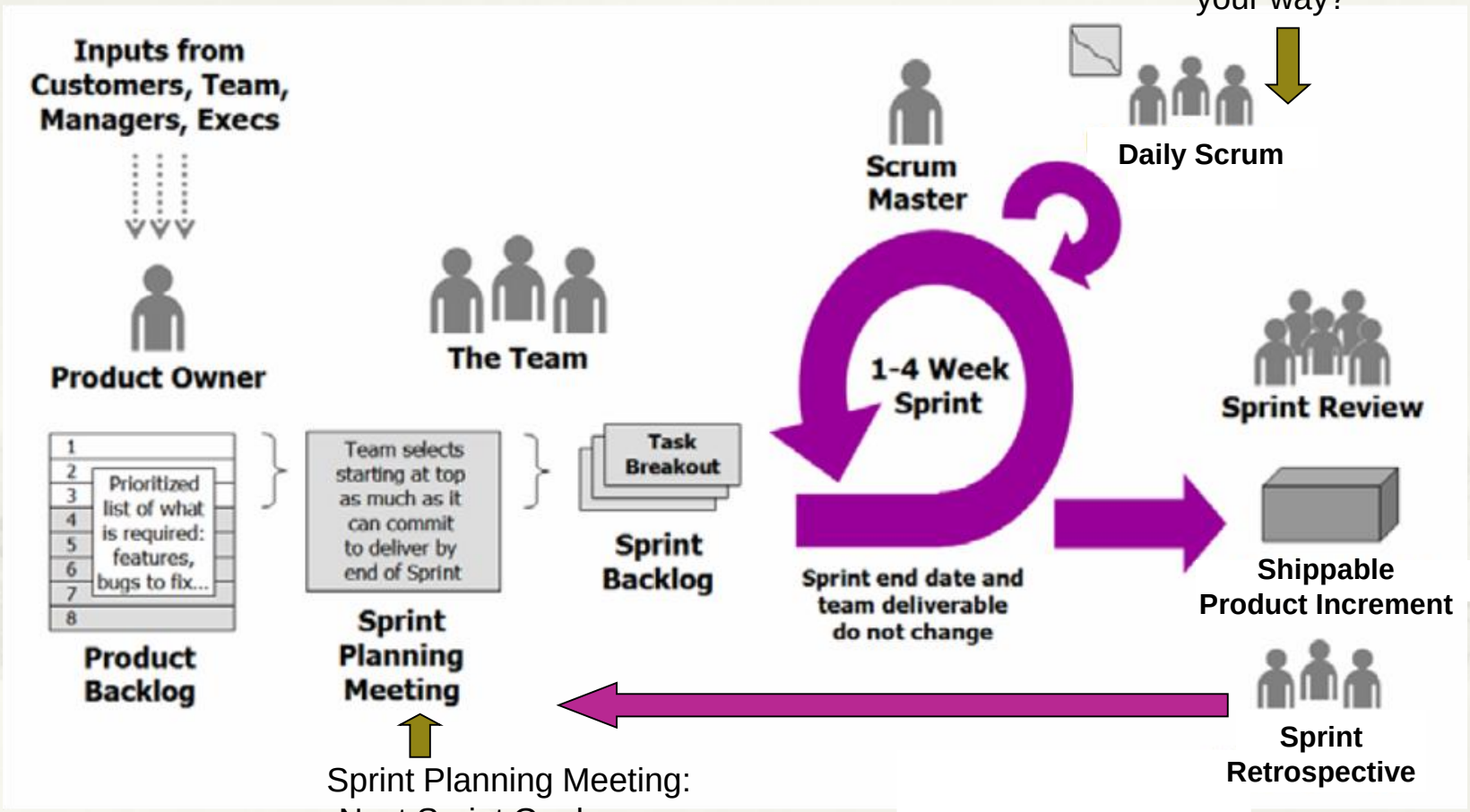
# 目录

- 敏捷的背景与动机
- 敏捷宣言及原则
- 敏捷方法是什么？
- 敏捷方法的实践
- Scrum的角色
- ***Scrum流程和工作产品***
- Scrum应用
- 总结

# Scrum 流程

Daily Scrum meetings :

- What did you do yesterday
- What will you do today?
- What obstacles are in your way?




Sprint Planning Meeting:

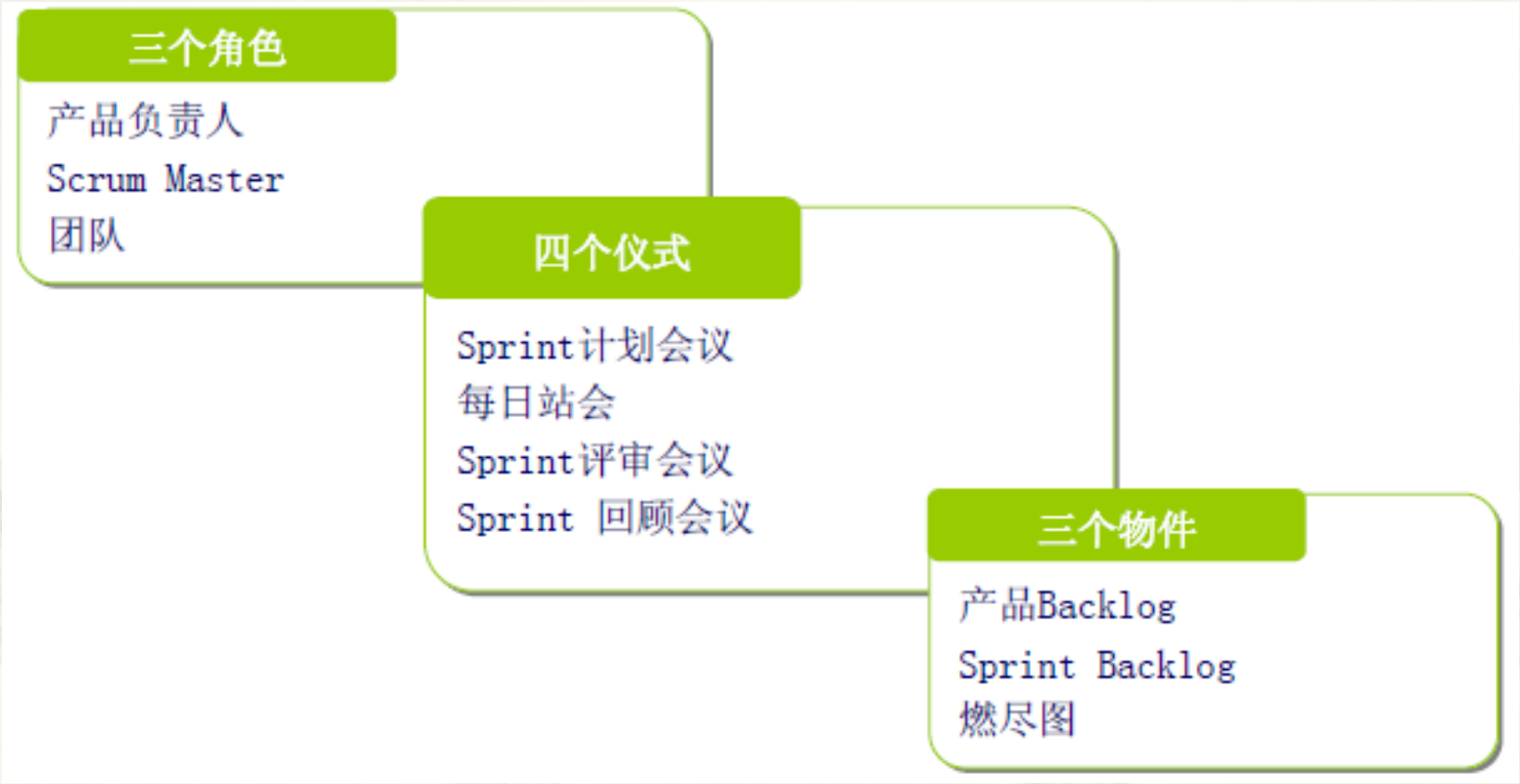
- Next Sprint Goal
- Sprint Backlog
- Updated Product Backlog

Source: http://www.amdika.com/scrum_primer_1_0.pdf

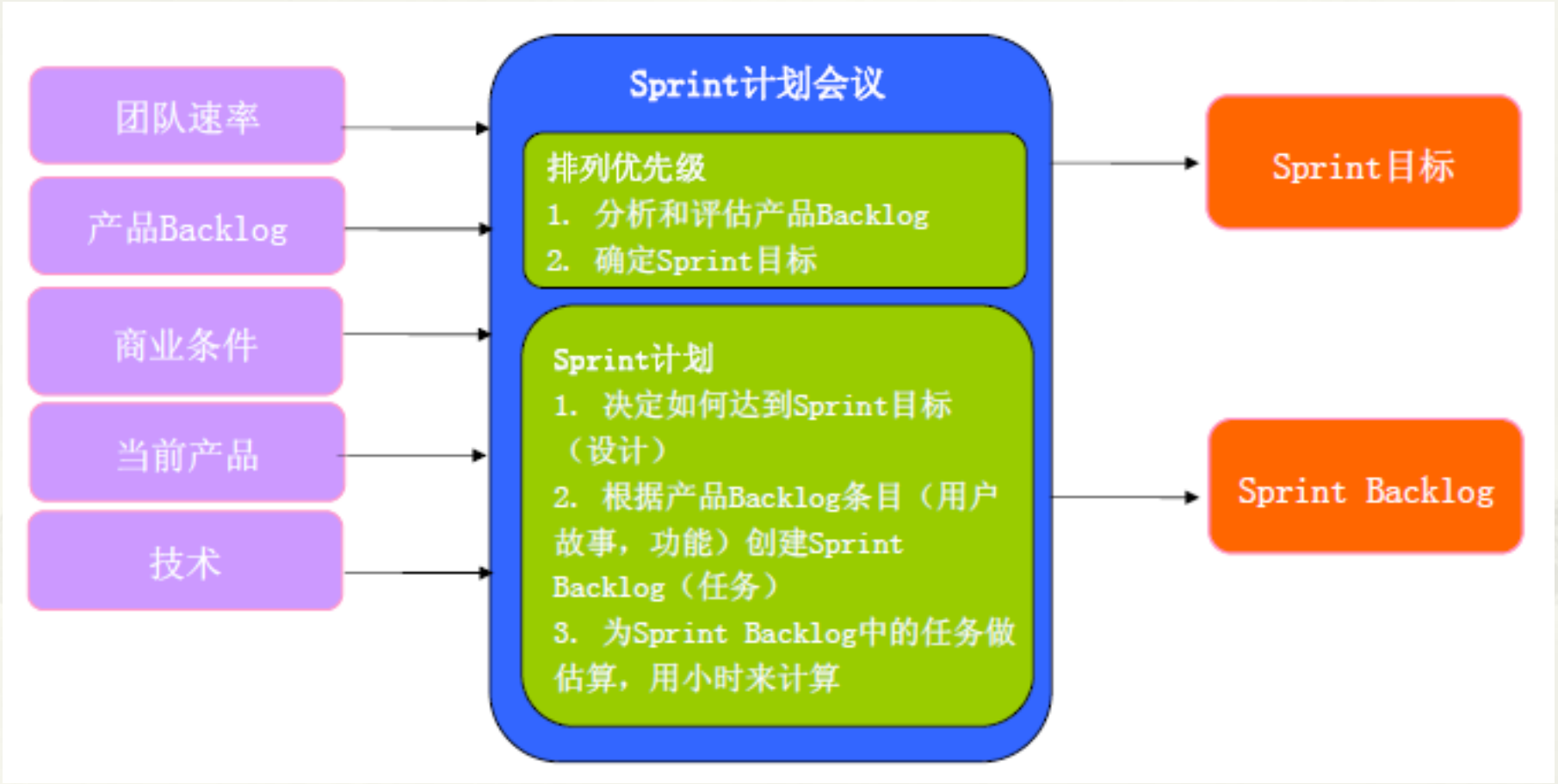
# Sprints(冲刺)

- Scrum的项目过程有一系列的Sprint组成。
- Sprint的长度一般控制在2-4周。
- 通过固定的周期保持良好的节奏。
- 产品的设计、开发、测试都在Sprint期间完成。
- Sprint结束时交付可以工作的软件。
- 在Sprint过程中不允许发生变更。

# Scrum框架

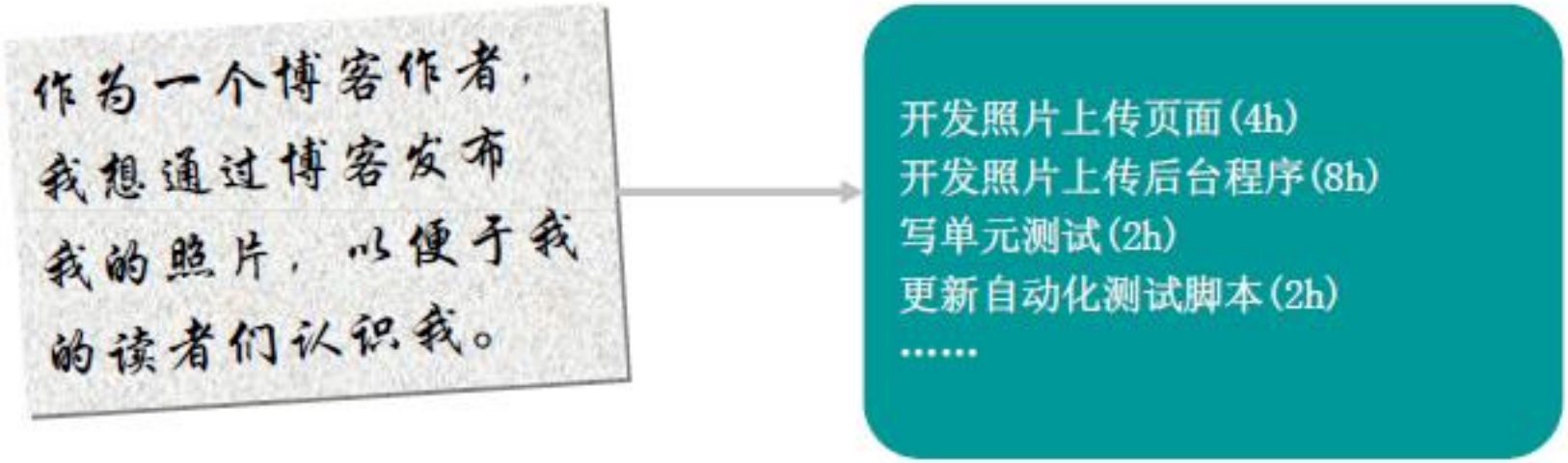

三个角色
产品负责人
Scrum Master
团队
四个仪式
Sprint计划会议
每日站会
Sprint评审会议
Sprint 回顾会议
三个物件
产品Backlog
Sprint Backlog
燃尽图

# Scrum仪式之Sprint计划会议



# Scrum仪式之Sprint计划会议

- 团队从产品backlog中挑选他们承诺完成的条目。
- 创建Sprint Backlog
  - 标识具体的任务并为任务做估算
  - 由团队协作完成，而不是Scrum Master
- 考虑了高层设计

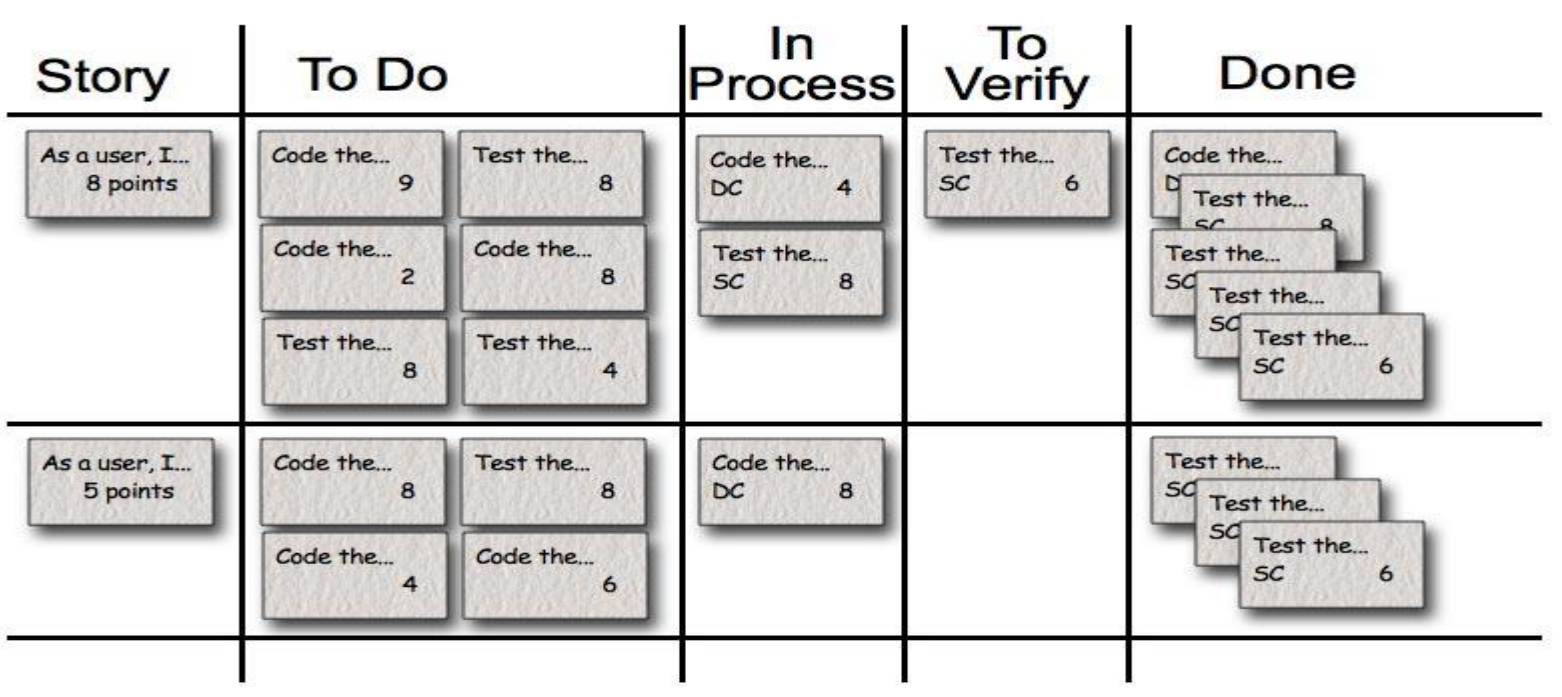


# Scrum仪式之每日Scrum会议(Daily Scrum)

- 每日Scrum会议，即团队每日例会，条件允许的话，每天都应该在同样的时间和地点，组织所有成员站立进行。
- 最好是每天早晨开，一般15分钟左右，时间比较短，也有利于团队成员安排好当天的工作。
- 只有团队成员可以在例会上发言，其他人员有兴趣可以参加，但只能旁听，不能发言。（小猪和小鸡的故事）
- 每日Scrum会议由Scrum Master主持， Scrum团队所有成员轮流回答以下3个问题：
  - 昨天我完成了什么工作？
  - 今天我打算做什么？
  - 我在工作中遇到了什么困难？

# Scrum 任务板(Task Board)

任务板（墙）展现了在Sprint过程中所有要完成的任务。在Sprint过程中我们要不断的更新它。如果某个开发人员想到了一个任务他就可以把这个任务写下来放在任务墙上。 无论每日站会过程中或者之后，如果估计发生了变化，任务会根据变化在任务墙上做相应的调整。通常的任务板是下面这个样子:



# Scrum仪式之Sprint评审会议

- Sprint评审会用来演示在这个Sprint中开发的产品功能给Product Owner. Product Owner会组织这阶段的会议并且邀请相关的干系人参加。
  - 团队展示Sprint中完成的功能
  - 一般是通过现场演示的方式展现功能和架构
  - 不要太正式
    - 不需要PPT
    - 一般控制在2个小时
  - 团队成员都要参加
  - 可以邀请所有人参加

# Scrum仪式之Sprint回顾会议

- 团队的定期自我检视，发现什么是好的，什么是不好的。
- 一般控制在15-30分钟
- 每个Sprint都要做
- 全体参加
  - Scrum Master
  - 产品负责人
  - 团队
  - 可能的客户或其它干系人

Sprint回顾会议上，全体成员讨论有哪些好的做法可以启动，哪些不好的做法不能再继续下去了，哪些好的做法要继续发扬。

# Scrum物件之产品订单(Product Backlog)

- 一个需求的列表。
- 一般情况使用用户故事来表示backlog条目
- 理想情况每个需求项都对产品的客户或用户有价值
- Backlog条目按照商业价值排列优先级
- 优先级由产品负责人来排列
- 在每个Sprint结束的时候要更新优先级的排列

# Scrum物件之产品订单(Product Backlog)

一个产品Backlog的例子

| Backlog 条目 | 估算（故事点） |
| --- | --- |
| 作为一个博客作者，我想设置我发布文章的背景图片，以便于我的读者阅读的时候感受到文章的意境。 | 8 |
| 作为一个博客作者，我想让我的读者对我的文章进行评价，以便于收集读者反馈，日后改进。 | 10 |
| 作为一个博客作者，我想通过博客发布我的照片，以便于我的读者们认识我。 | 20 |
| ..... | 30 |
| ..... | 50 |
| | |
| | |

# Scrum物件之冲刺订单(Sprint Backlog)

Sprint backlog定义了Sprint的目标，明确了Sprint过程中具体需要完成的任务

下面是一个Sprint backlog的例子：

| 任务 | 2 周的Sprint | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 开发照片上传页面(4h) | 4 | 2 | 0 | | | | | | | |
| 开发照片上传后台程序(8h) | 8 | 2 | 3 | 0 | | | | | | |
| 写单元测试(6h) | 6 | 3 | 0 | | | | | | | |
| 开发文章背景图片设置页面(16h) | 16 | 8 | 4 | 0 | | | | | | |

# Sprint Backlog 示例

Sprint goal

Persons working on the task

Effort estimate

**Sprint 1 Backlog** — **Goal: deliver working version of web page**

| 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 |
| | | | | 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 | | 4 | Not started |
| 5 | 2 | Create graphics & banners | 5 | Create Hotel logo | | 2 | Not started |
| | | | | Create animated advertisment banner | Erik | 2 | Impeded |
| | | | | Create background images | | 6 | Not started |
| | | | | Review graphics with customer | | 4 | Not started |
| | | | | Agree on colours used | | 4 | Not started |
| | | | | Make photos of the hotel | John | 8 | Done |

Description of the task

Meets the definition of done

Task blocked by an impediment

# Scrum物件之冲刺订单(Sprint Backlog)

- 管理Sprint的backlog:
  - 团队成员自己挑选任务，而不是指派任务
  - 对每一个任务，每天要更新剩余的工作量估算
  - 每个团队成员都可以修改Sprint backlog，增加、删除或者修改任务

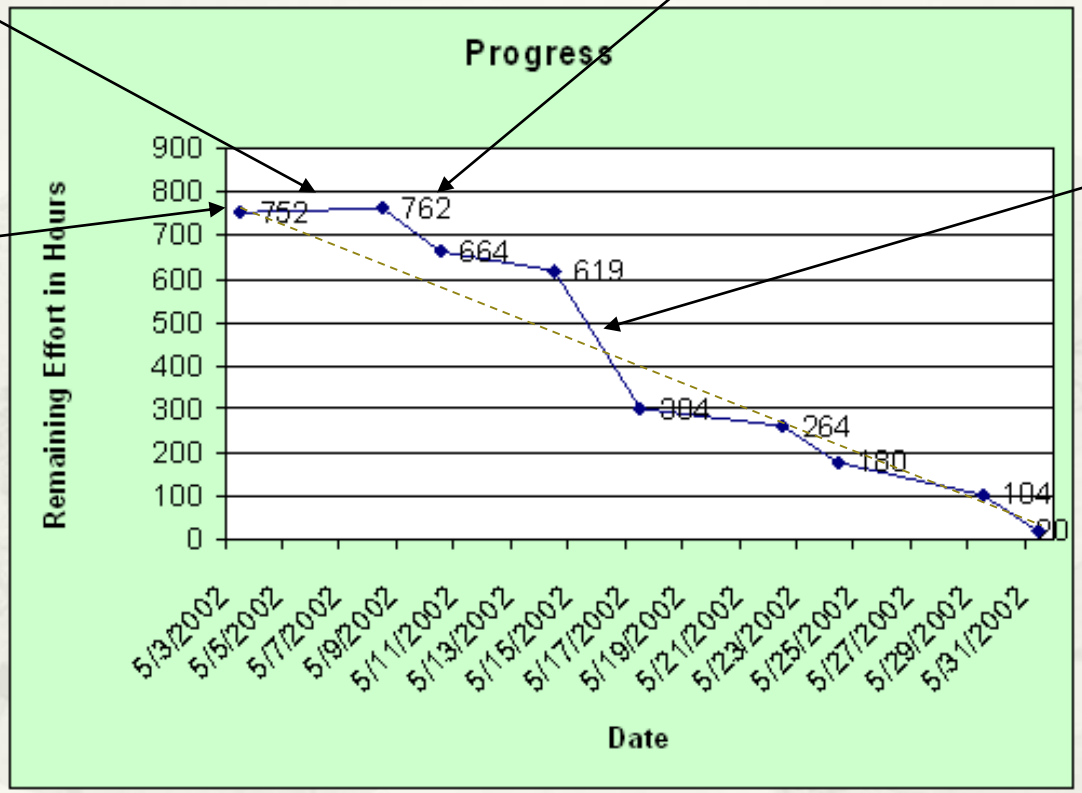
# Scrum物件之燃尽图(Burn Down Chart)

Remaining work increasing →Tasks underestimated and/or work remaining not updated.

Sum of remaining work [h] for all tasks in the Sprint Backlog on a particular day.

Initial estimate (752 h) In the beginning of the Sprint

Tasks removed from the Sprint Backlog to meet Sprint Goal → faster decline.




Ideal burndown.

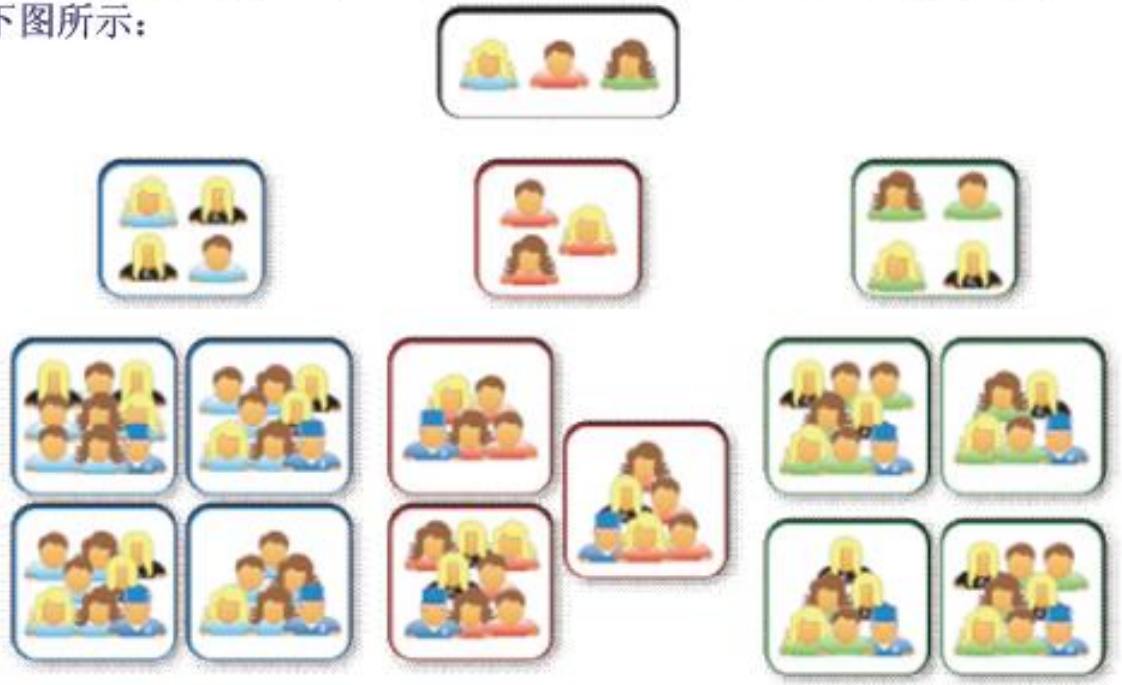
Actual burndown.

# 扩展Scrum

- 一般情况一个团队的人数控制在5-9人
  - 大型项目可以采用多团队，通过team of teams来扩展Scrum。
- 影响扩展的因素
  - 团队规模
  - 项目类型
  - 项目周期
  - 团队分布
- Scrum曾被用于超过1000人团队规模的项目。

# Scrum Of Scrums

Scrum of Scrums是把Scrum扩展到大型项目团队一个重要的实践。Scrum of Scrums是跨团队的沟通与交流。一般做法是由各个Scrum团队的代表参加Scrum of Scrums会议，会议同样要采用固定的频率和时间箱机制，频率可以有团队根据自身情况确定，一般可以一周2到3次，不一定要每天。对于会议长度，Ken Schwaber的建议是控制在15分钟，会上提出的问题应及时组织相关人员处理。Scrum of Scrums是多层次的，如下图所示：



# Scrum项目之估计

- Scrum团队对产品需求清单的每一项的规模提供初步的估计，通常采用事件点作为单位Story Points (模糊的).
  - 也可采用人天或者人小时作为单位，但容易混淆： a) 实际的规模 b) 时间的单位.
  - 精确的估计值可以在Sprint 规划时给出, 当前阶段没有足够的信息.
  - 规模的相对值才有意义.
  - 这个估计值有助于确定优先级;
  - 可以采用估算扑克

产品规模

团队速度

所需时间

# 完成的定义

- 当迭代任务清单上的任务都完成时，变为“已完成”状态
- 定义“已完成”的含义是非常重要的, 例如:
  - 如何记录软件的变化.
  - 使用什么样的代码分析工具，发现的问题应当如何处理.
  - 进行了什么样的测试, 结果是如何记录的, 通过标准（如覆盖率、修正的错误）是什么.
- 定义“已完成”意味着定义质量上的需求.
- “已完成”是0/1变量：完成或者未完成. 所有的任务（task)都完成了迭代任务才算完成.
- 在第一个迭代开始之前应该定义好，因为它会影响工作量, 而且必须文档化，这样团队和产品所有者的理解是一致的.

# 完成的定义 - Example

* 完成的定义
  * 遵循编码规范
  * 能在模拟器上演示
  * 使用PCLint 进行静态代码分析
  * 具有EUnit 测试套件的通过率 和执行率.
  * 或者使用结对编程，或者进行代码走查

# 障碍

- 基本上，任何阻止团队正常工作的，都可称之为障碍，例如:
  - 无法访问信息系统.
  - 所需要的信息不能及时提供或者提供的不正确，如界面规格或者其它软件模块不到位或不正确
  - 开发环境或者原型系统出现问题
  - 其他的任务分配：培训，售前支持
  - 缺乏必要的信息或者相应的知识
- 对于团队提出的各项障碍，Scrum Master要以列表形式进行记录，

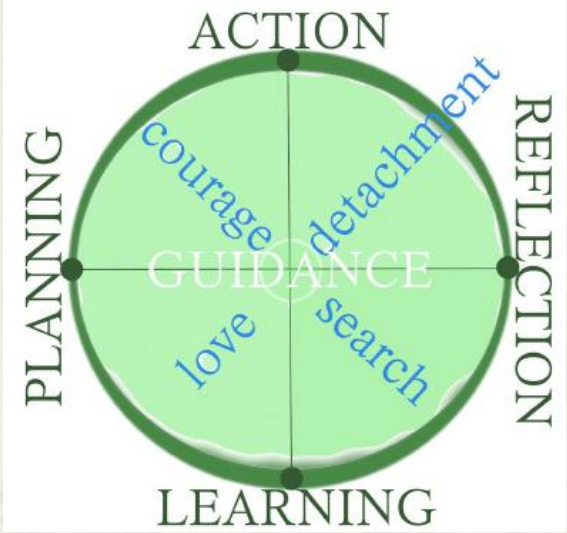
# 谁来清除障碍？

* 每个人
  * 自我管理、自我组织的团队
  * Scrum Master
  * 产品所有者
  * 管理层
  * 其他相关的干系人
* Scrum Master 负责确定障碍已经清除，不一定亲自自己清除

# 清除障碍

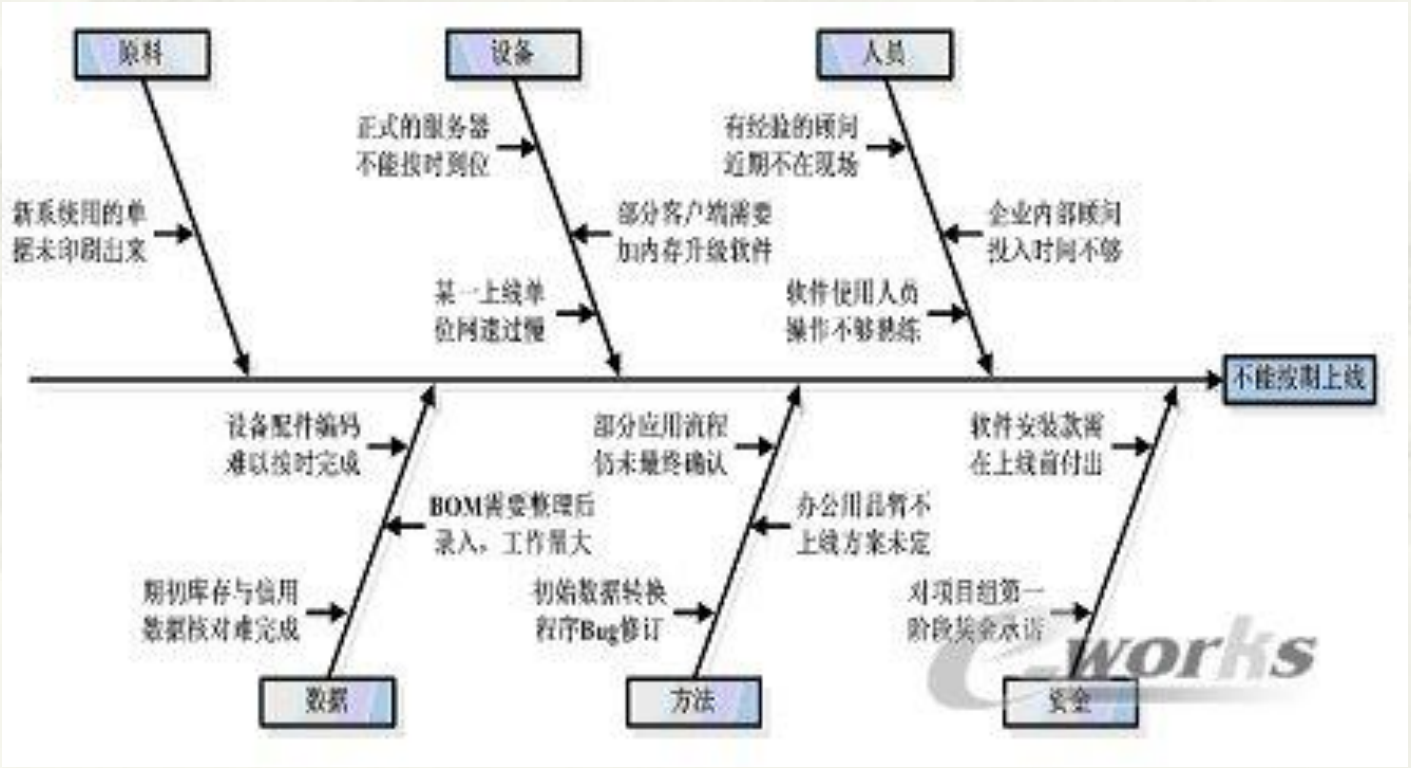
* 某些障碍是浪费
  * 部分地完成工作
  * 额外的过程
  * 额外的功能
  * 任务转换
  * 等待
  * 缺陷

* 清除障碍的过程是团队和组织学习的过程



# 浪费产生的原因

- 多问几个“为什么”
  - 对于每个标识的障碍或者浪费，问一问“为什么”浪费会存在
  - 多问几个“为什么”，找到造成浪费的根本原因



# 目录

- 敏捷的背景与动机
- 敏捷宣言及原则
- 敏捷方法是什么？
- 敏捷方法的实践
- Scrum的角色
- Scrum流程和工作产品
- ***Scrum应用***
- 总结

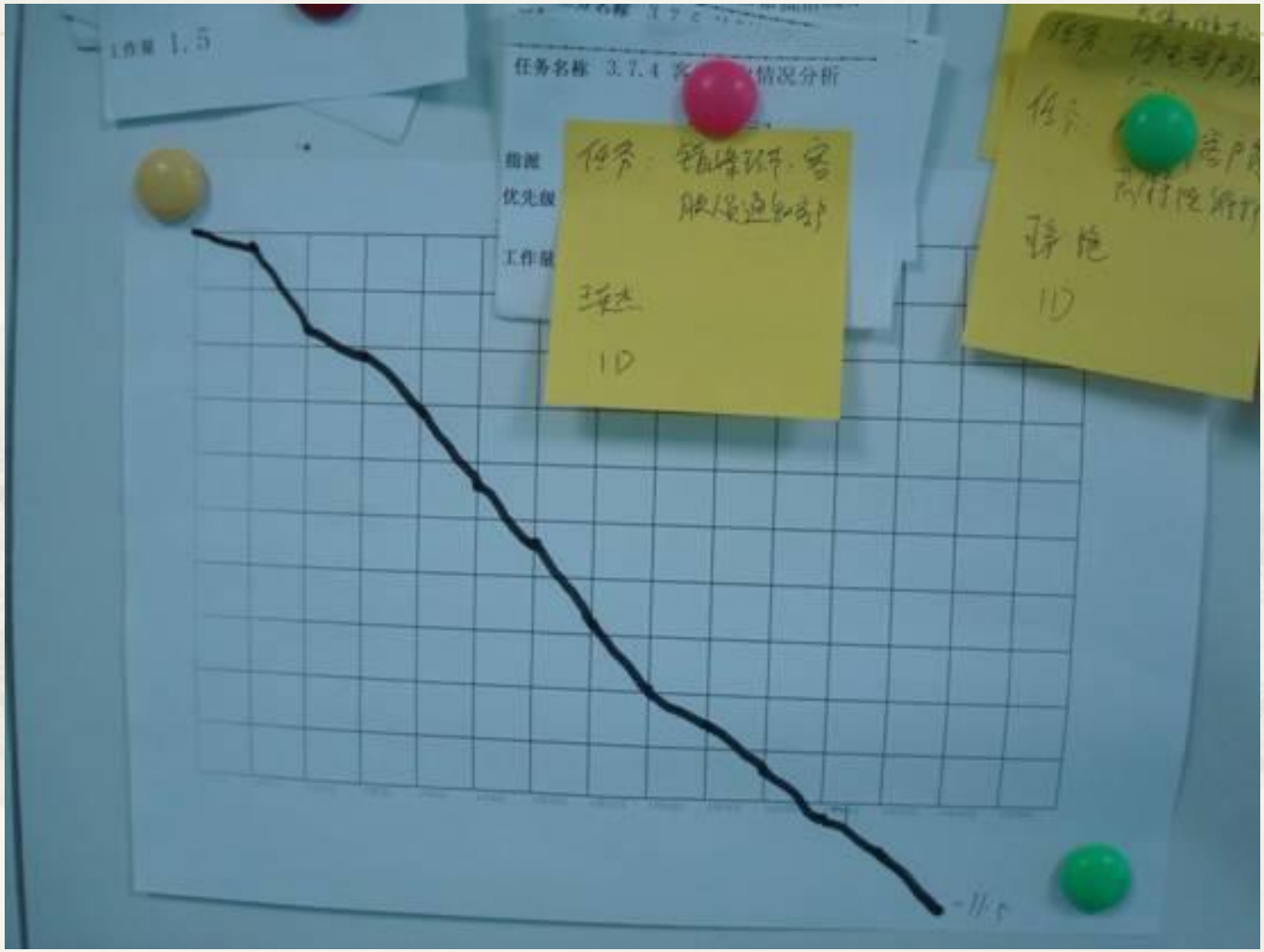
# SCRUM实践

- 研发部2009年开始在几个项目当中进行了SCRUM项目管理的尝试：
  - 营销综合停电系统开发
  - FLEX-ADP开发
  - 海颐OA项目

  等

# SCRUM看板

# SCRUM燃尽图



# SCRUM带来的改善

- 项目的计划性更强了，将项目按SPRINT进行分解，每个SPRINT要进行计划和总结，每天也有立会来进行简短的总结和计划；

- 引入SCRUM以后，项目团队的沟通比以往更有效，项目看板为项目团队沟通提供了一个统一的项目视图，每日立会是项目团队沟通的有效通道；

- 项目的阶段性比以前更明确，通过SPRINT将项目划分成阶段，通过SPRINT演示等活动将项目整体的压力分解到每个SPRINT，这样可以有效降低项目的整体风险。

# 目录

- 敏捷的背景与动机
- 敏捷宣言及原则
- 敏捷方法是什么？
- 敏捷方法的实践
- Scrum的角色
- Scrum流程和工作产品
- Scrum应用
- *总结*

# 一些常见的误解

- 敏捷是拯救任何项目的银弹.
  - 敏捷方法只有运用得当才有效果.
- 敏捷意味着 ad-hoc hacking，不需要任何文档.
  - 敏捷是有严格要求的，也是面向质量的
  - 根据沟通的需要产生相应的文档.
- 敏捷只是开发者的问题
  - 基本的开发方法与传统相比有显著不同, 影响项目的各个方面: 合同, 角色, 定价模型, 项目管理等.
- 采用敏捷方法的开发组/项目不需要制定计划
  - 敏捷项目需要经常制定计划，但是不需要试图超前制定项目计划，通常这也是不可能的.
- 敏捷项目的范围可以随时改变.
  - 变更可以等到下一次迭代开始，当前正在进行中的迭代不能变更
- 只对小项目适用
  - 在中型和大型的项目中一样取得了成功

# 总结

- Agile Software Development是软件开发所强调的一个精神，而不是一个方法。
- 遵循Agile Alliance所提的四个价值观与12个原则。
- 最常见的开发方式
  - XP
  - SCRUM
- 敏捷开发过程是一个艰苦的过程，重在实践
- 即使非敏捷的项目中也可以应用敏捷的实践经验
- CMMI应该与敏捷实现融合，双剑合璧

问题3，在SCRUM中能够直观展现冲刺的进度的图形是什么图？

# Thank You !