

Certified ScrumMaster Training CSM 认证培训

Vernon Stinebaker, CST

Simple Rules of the game

简单游戏规则

简介 Introduction

- Vernon Stinebaker (史文林)



<http://www.linkedin.com/in/vernonstinebaker>
<http://twitter.com/vernons>

Bill Li (李国彪)



<http://www.linkedin.com/in/scrumchina>

<http://twitter.com/scrumchina>

新浪微博: <http://t.sina.com.cn/scrumchina>

Exercise: 破冰 (Ice-Breaker)

Scrum Alliance Certifications

Foundation-Level



Mid-Level



Professional-Level



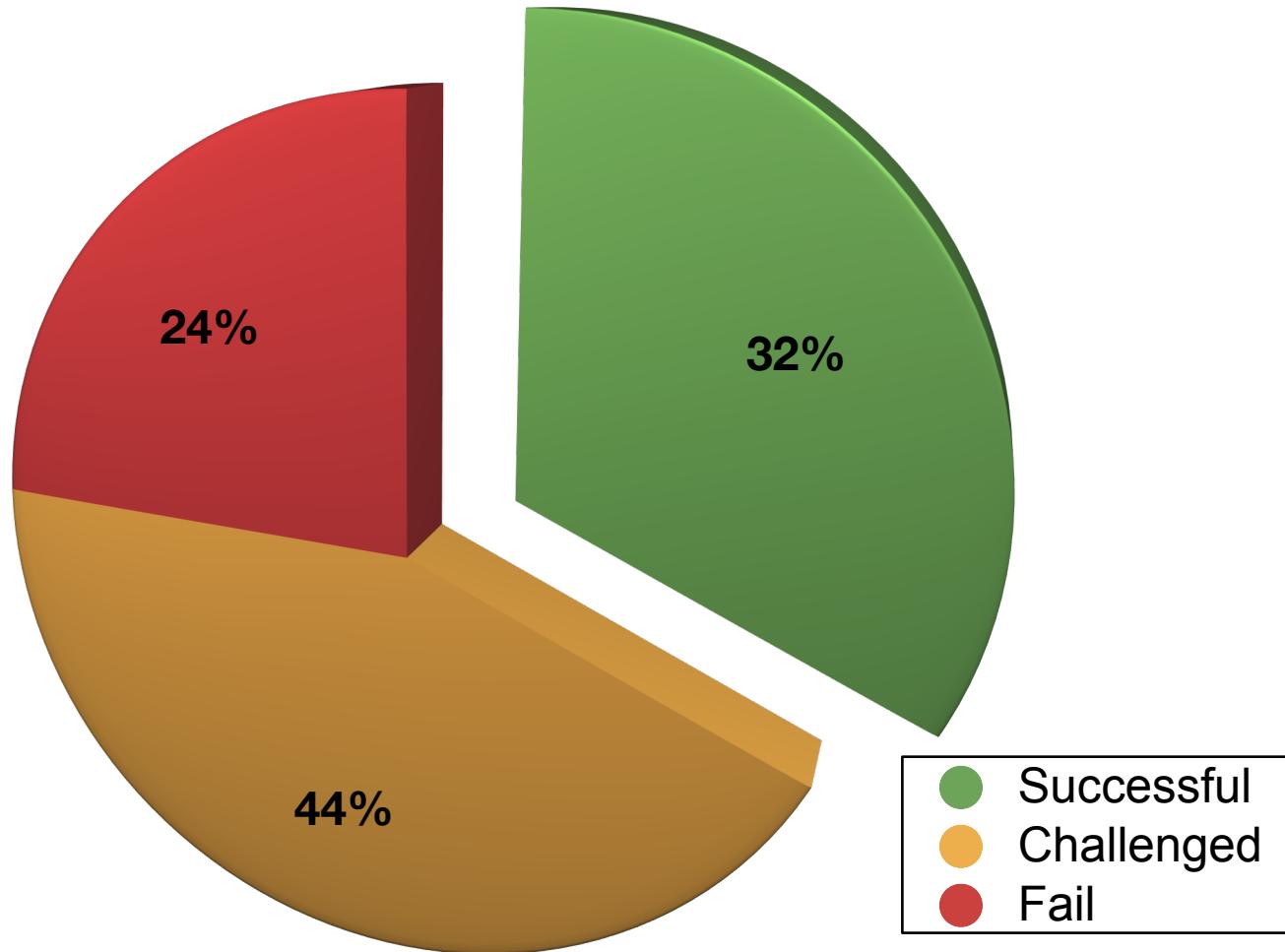
Guide-Level



#FAIL
#失败

Project Success Statistics

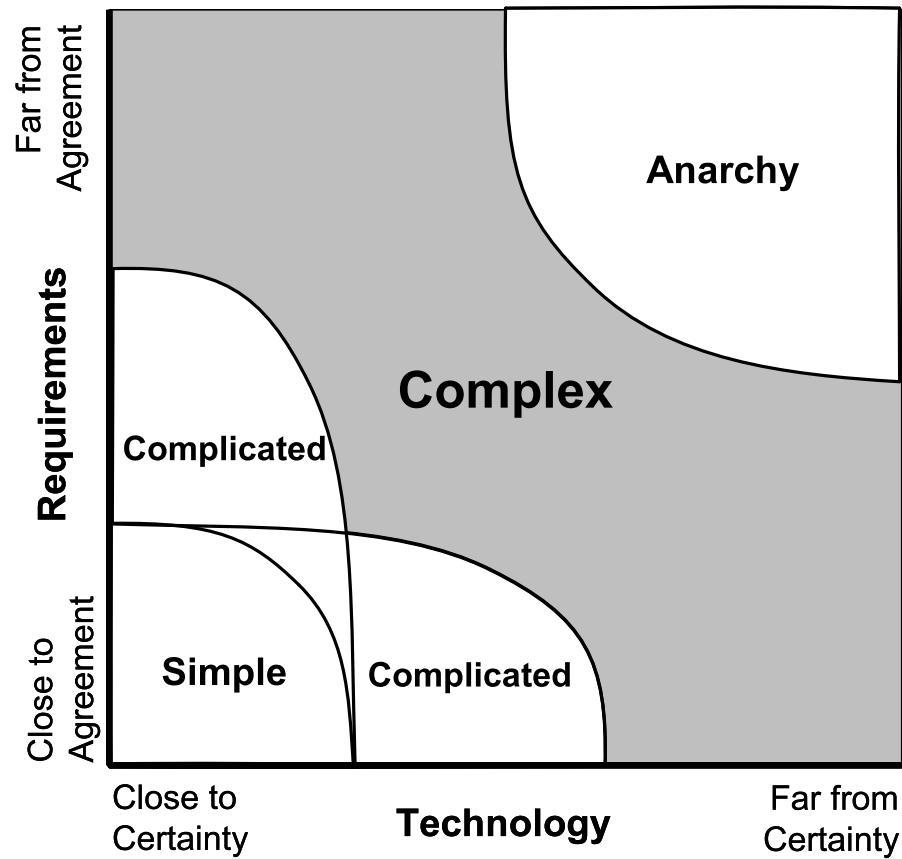
项目成功率统计



成功的
有挑战的
失败的

Chaos Report 2009
The Standish Group

软件开发项目中的复杂性 Playing Field



Graph taken from Ralph Stacey's “Complexity and Creativity in Organizations”

预定义（理论的）过程 vs. 实验性过程

Defined (Theoretical) vs. Empirical Process



- 命令及控制 Command & Control
- 对你预计会发生的进行计划 Plan
- 强制按计划，不管条件的变化 Enforce the Plan
- 使用变更控制 “Control” Change



- 边前进边学习 Learn as we go
- 计划好会变更（需求，工作环境，市场等） Plan to change
- 欢迎变更 Embrace Change
- 使用检视及适应调整 Inspect and Adapt

Thanks to Peter Borsella

预定义（理论的）过程 vs. 实验性过程 Defined (Theoretical) vs. Empirical Process

“When the process is too complicated for the defined approach, the empirical approach is the appropriate choice.”

若过程复杂程度超出预定义方式的能力范围便应选用实验性方式。

▪ *Process Dynamics, Modeling and Control*, Ogunnaike and Ray, Oxford University Press, 1992

Three Legs of the Empirical Process

实验性过程的三大支柱

Transparency 透明

Inspect 检视

Adapt 适应

什么是敏捷? What is Agile?

It is NOT a methodology,
process, or framework

不是一个单方法论、过程或框架

Values and principles

价值观与原则

Values 敏捷宣言价值观

Individuals and interactions over processes and tools

个体和交互 胜于 流程和工具

Working software over comprehensive documentation

可工作软件 胜于 冗长的文档

Customer collaboration over contract negotiation

与客户协作 胜于 合同的谈判

Responding to change over following a plan

对变化响应 胜于 遵循原计划

Principles 敏捷原则

1. 我们最先要做的是通过尽早地、持续地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。敏捷过程利用适应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可以的话尽可能在一起工作。
5. 围绕被激励起来的个体组成团队来构建项目。给他们提供所需的环境与支持，并且信任他们能够完成工作。
6. 在团队内部以及团队之间，最有效果并且最有效率的传递信息的方式，就是面对面的交流。
7. 可以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。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的能力。
10. 简单——使未完成的工作最大化的艺术——是根本的。
11. 最好的架构、需求和设计出自于自组织的团队。
12. 每隔一定的时间，团队会在如何才能更有效地工作方面进行反省，然后相应地调整自己的行为。

Agile Principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Delivery working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity – the art of maximizing the amount of work not done – is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

**Potentially Shippable Product at the
end of each Sprint**

每个Sprint产出潜在可交运的软件

Class Project

课程项目

(Epic 史诗故事)



UPerform Your Agile and Project Performance Partner
Copyright © 2010

Scrum

敏捷与Scrum的起源 Scrum Origin



Scrum in 100 words (Mike Cohn)

- Scrum is an agile process that allows us to focus on delivering the highest business value in the shortest time.
- It allows us to rapidly and repeatedly inspect actual working software (every one weeks to one month).
- The business sets the priorities. Teams self-organize to determine the best way to deliver the highest priority features.
- Every one weeks to a month anyone can see real working software and decide to release it as is or continue to enhance it for another sprint.

- ✓ Scrum是一个敏捷过程，使我们能在最短的时间内专注于交付最高的商业价值
- ✓ Scrum使我们能迅速及不断地检验实在的可用的软件（通常每一周到一个月）
- ✓ 业务人员设定优先级。自组织的团队决定最好的方法与方式来交付最高优先级的产品特性
- ✓ 通常每一周到一个月，所有人都能看到真实的可用的软件，所以能确定是否立即进行发布还是通过下一个Sprint来继续完善它



Thanks to Mountain Goat
Software, LLC



Origins & History 发展历史

- Jeff Sutherland
 - 最早的Scrum项目: Easel Corp 1993
 - IDX – 500+ 的人一起使用Scrum
- Ken Schwaber
 - 与Sutherland一起在 OOPSLA '96年 大会就Scrum发表演讲
 - 后来写了3本关于Scrum的著作
- Mike Beedle
 - 在PLOPD4发表Scrum模式
- Ken Schwaber 及 Mike Cohn
 - 共同创办了 Scrum 联盟, 于2002年

Origins & History

- Jeff Sutherland
 - Initial scrums: Easel Corp 1993
 - IDX - 500+ people doing Scrum
- Ken Schwaber
 - Scrum presented at OOPSLA '96 with Sutherland
 - Three books on Scrum
- Mike Beedle
 - Scrum patterns in PLOPD4
- Ken Schwaber and Mike Cohn
 - Co-founded Scrum Alliance in 2002

Scrum 的一些特征 Scrum Characteristics

- 自组织团队 Self-organizing teams
- 迭代开发，一个Sprint一个Sprint推进 progresses in a series of time boxed “Sprints”
- 项目信息的可见性 Visibility and Transparency
- 一份产品需求清单 “Product Backlog”
- 没有说明任何的工程实践 No specific engineering practices prescribed
- 强调营造一个敏捷的工作环境，并不断改善 Uses generative rules to create an agile environment for delivering projects
- 其中一个敏捷过程 One of the “agile processes”

Transparency 透明性



© Scott Adams, Inc./Dist. by UFS, Inc.

越来越多的组织应用Scrum More and more organizations use Scrum

他们发现SCRUM是解决问题的有效工具。

过去的几年中，一些行业领先公司越来越广泛的应用SCRUM。

They found Scrum as an effective tool. In the past few years, more and more leading companies adopted Scrum.

- Google
- IBM
- Nokia
- Siemens
- Philips
- Accenture
- Sun
- Ubisoft
- Bleum
- SAP
- Virtuos Games
- Microsoft
- Infosys
- Oracle
- Wipro
- Yahoo!
- Schneider
- Agilent
- Double Click
- Huawei
- Motorola
- Perficient
- Tencent
- Cybercom Plenware
- Trendmicro
- Moody's
- EA
- Genesys
- Tieto
- Cdcsoftware
- Ericsson
- Salesforce.com



Thanks to Mountain Goat
Software, LLC



Scrum被应用于 Scrum is being used in

- 大型企业级软件项目 Enterprise Software
- 商业软件产品 Business Software
- 消费者软件项目/大型网站 Consumer Software/Major Websites
- 美国FDA批准的应用于X射线和MRI的软件 Medical Software
- 高可靠性系统 (99.9999%以上) High-reliability Systems
- 财务支付系统 Financial and Payment Systems
- 智能家居项目 Intelligent Household Project
- 联合战斗机项目 The Joint Jet Fighter Project
- 大型数据库应用 Big Database Applications
- 嵌入式电信系统 Embedded Telecom Systems
- 手机项目 Cell Phone Projects
- CMMI5级的组织 CMMI Level 5 Organizations
- 多地点同步开发 Multi-sites Development
- 支撑和维护项目 Maintenance and Support Projects
- 非软件项目 Non-Software Projects
-

Scrum 框架 -- 从开始到结束

Scrum Framework – from Start to Finish

Roles 角色



- Product owner (PO)
- ScrumMaster
- Team 团队

Artifacts 工件



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

Ceremonies 仪式

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

Roles 角色



- Product owner (PO)
- ScrumMaster
- Team 团队

Artifacts

- Product Backlog
- Sprint Backlog
- Burndown Charts

Ceremonies

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

Product Owner (PO)



- 定义产品的Features(功能／特性) Defines the features of the product
- 负责产品的利润 (投资回报) Responsible for the profitability of the product (ROI)
- 根据市场价值对产品特性排优先级 Prioritize features according to market value
- 对发布日期和内容做决策 Decides on release date and content
- 需要时, 每一个迭代调整产品特性和优先级 Adjust features and priority every iteration, as needed
- 接受或者不接受工作的结果 Accept or reject work results

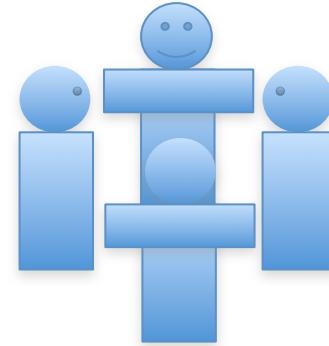
Case Study

The product owner says that he is not going to be available to attend the Sprint Planning meeting this time due to an important golf game with an executive of a partner company, but he doesn't mind if the team goes ahead and does it without him.

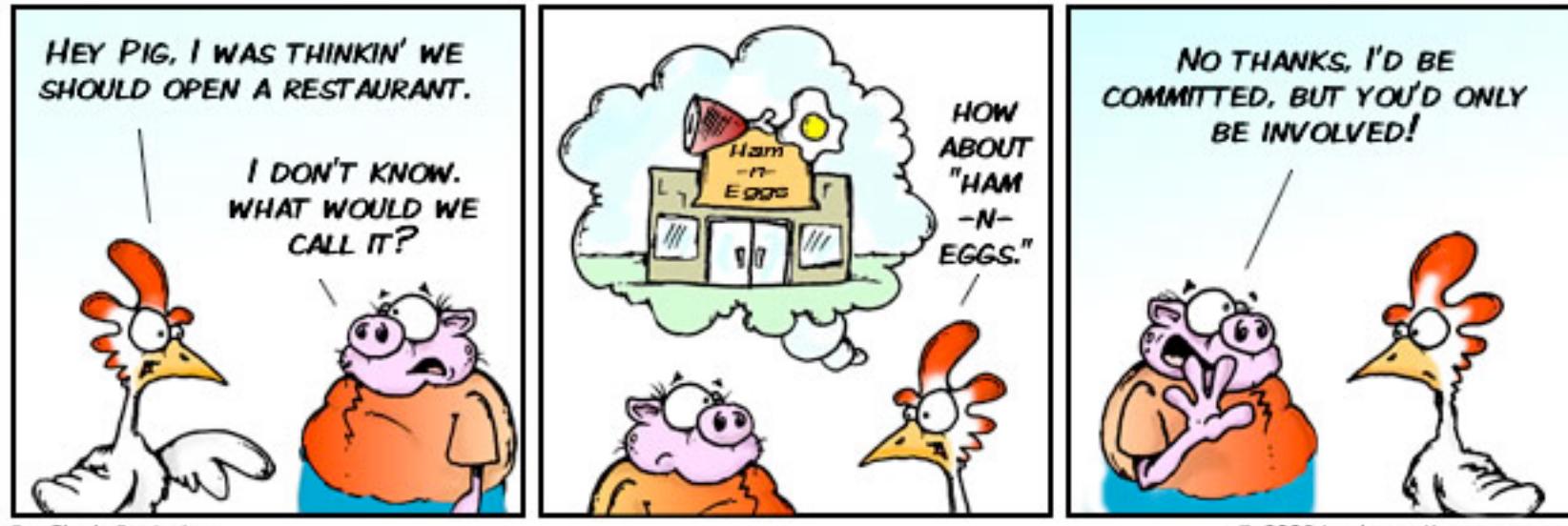
Product Owner对团队说他不能够参加这次的Sprint计划会议，因为他要和一个合作伙伴公司的上层打一场重要的高尔夫球。但他不介意团队按计划独自进行会议。

团队 Team

- 建议通常5—9人 Typically 5-9 people
- 跨职能 Cross-functional:
 - 程序员、测试员、用户体验设计师, 等等 Programmers, testers, user experience designers, etc.
 - 成员最好是全职的 Members should be full-time
 - 可能会有例外 (例如美工) May be exceptions (e.g., graphics artist)
- 团队是自组织的 Teams are self-organizing
 - 理想状况是没有职称之分, 但不常见 Ideally, no titles but rarely a possibility
- 迭代之间成员构成才作变更 Membership should change only between sprints



“鸡”与“猪”的故事 Chicken and the Pig





ScrumMaster

- 面向项目代表管理层 Represents management to the project
- 负责Scrum价值观和过程的实现 Responsible for enacting Scrum values and practices
- 移除障碍 Removes impediments
- 确保团队的生产能力和全速前行 Ensure that the team is fully functional and productive
- 促使所有角色和职能人员间的密切合作 Enable close cooperation across all roles and functions
- 保护团队免于外部干扰 Shield the team from external interferences
- 服务式的领导 Servant-Leader

谁是 ScrumMaster



- 谁是ScrumMaster Who is the ScrumMaster
- ScrumMaster候选人的特征 Characteristics of a ScrumMaster Candidate
- ScrumMaster 作为变革的代理人 ScrumMaster as a Change Agent
- 有效的聆听 Effective Listening
- Scrum的成功有赖于常识 Scrum's success depends on Common Sense

Case Study

In one project, everybody of a 9-member Scrum team seems very busy working on the team commitment for the Sprint. The ScrumMaster seems fairly idle and has not too much things to worry about. Someone suggests to the ScrumMaster: how about you pull some tasks from the Sprint Backlog and work on them.

在一个项目中，一个9个人的Scrum开发团队中所有成员好象都在非常忙碌于实现他们对这个Sprint的承诺。而ScrumMaster好象比较空闲，没太多事情可担心似的。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，和我们一起交付产品特性？

Roles



- Product owner
- ScrumMaster
- Team

Artifacts 工件



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

Ceremonies

- 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 Requirements - a list of all desired work on the project
- 一份合格的产品Backlog应该： A qualified product backlog must be:
 - ✓ 已排序的（按优先级） Ordered (Prioritized)
 - ✓ 有价值的（商业价值） Valued (what is the business value)
 - ✓ 带估算的（需要投入多少工作量） Estimated (how much effort will be required to complete)
- PO负责优先级排序 Prioritized by the product owner
- 在Sprint计划会议前PO可以调整排序 Reprioritized (by PO) before Sprint Planning Meeting

用户故事 User Stories

Card 卡片

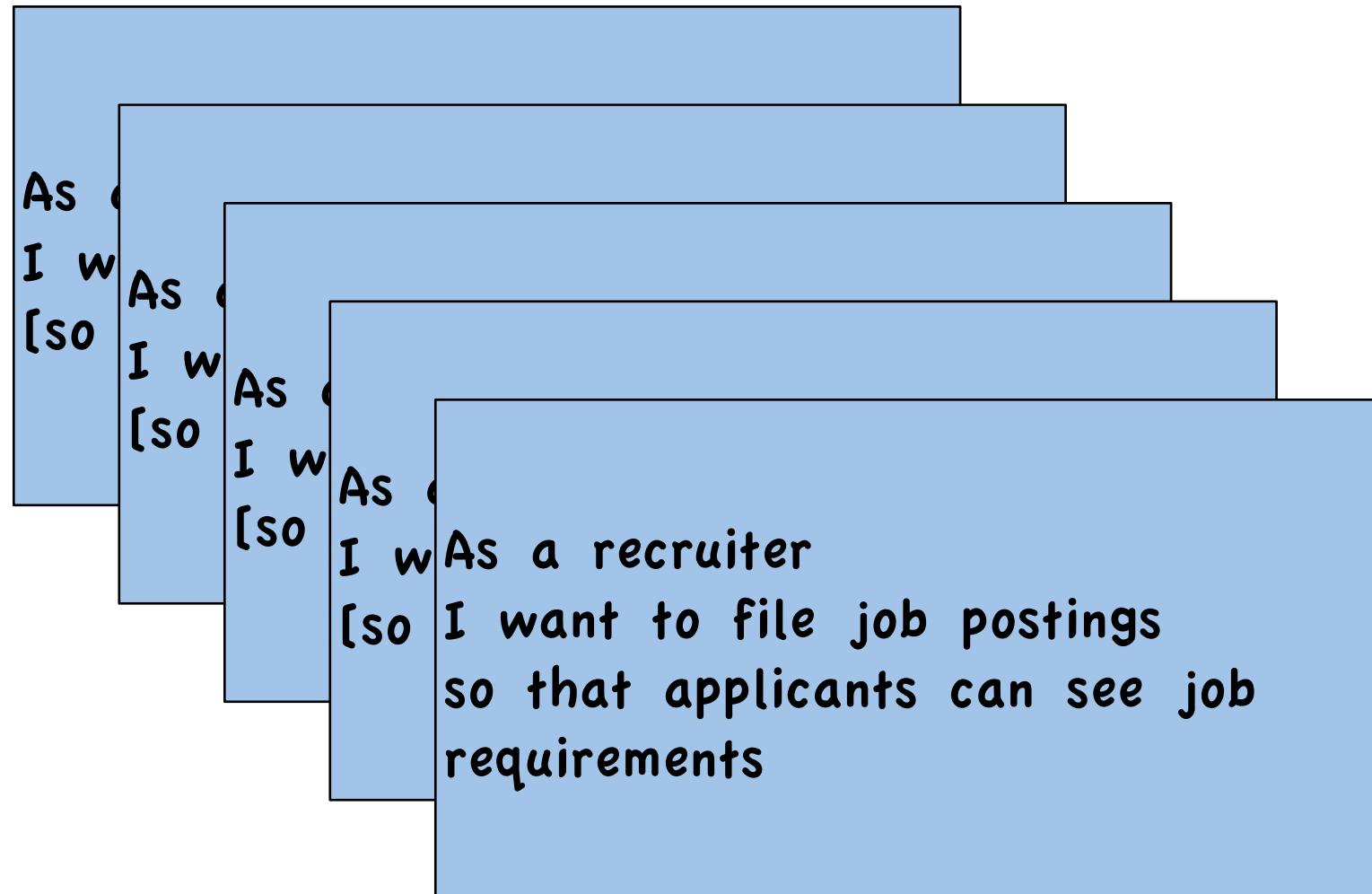
Conversation 交谈

Confirmation 确认

用户故事 User Stories

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]]

产品Backlog Product Backlog



INVEST

IIndependent 独立的

Negotiable 可协商的（可伸缩的）

VAlued 有价值的

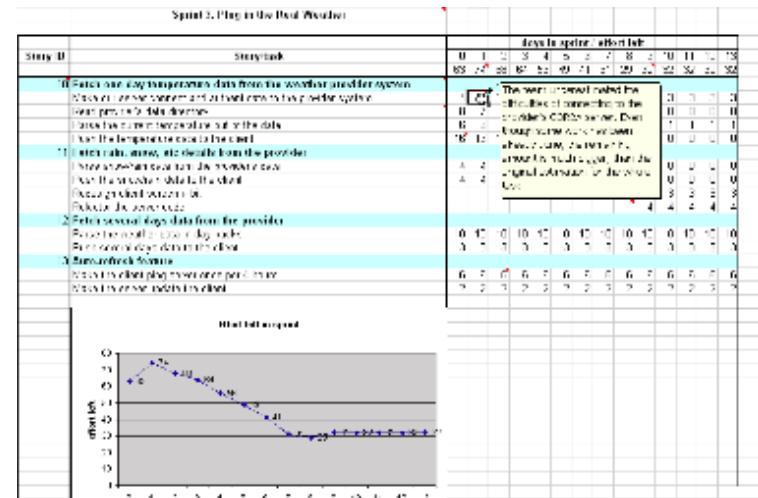
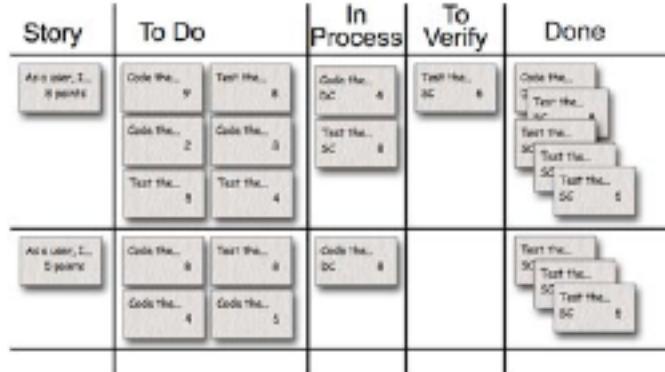
Estimable 可估算的

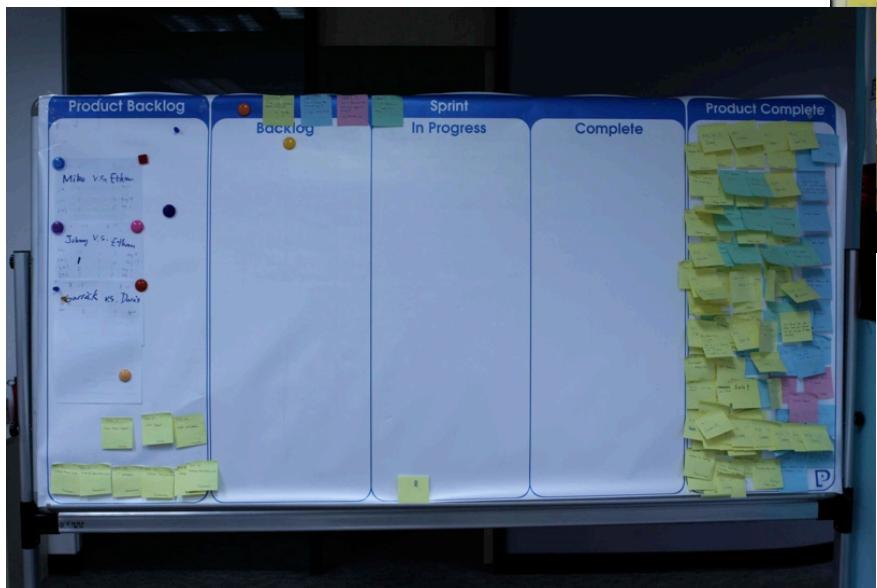
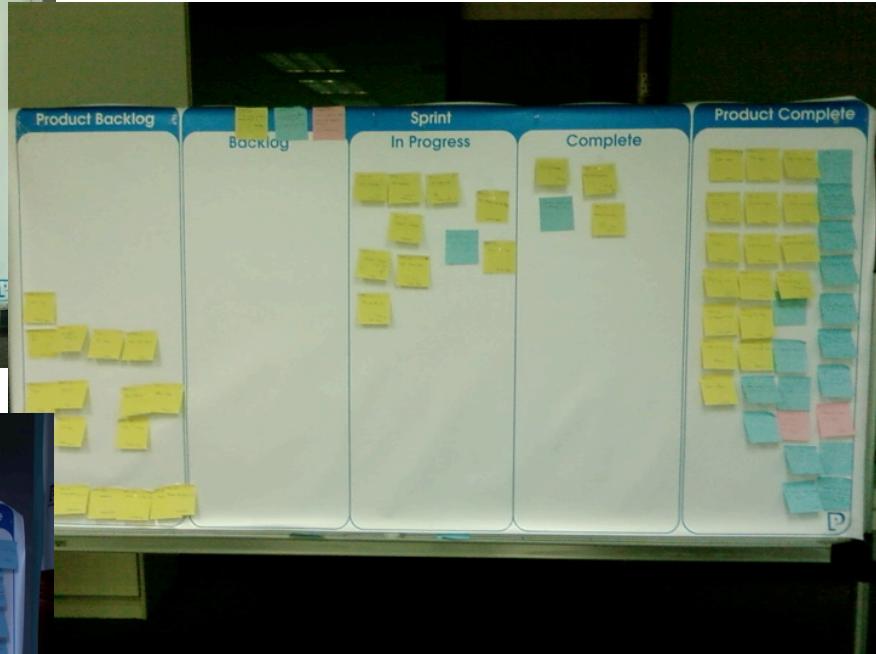
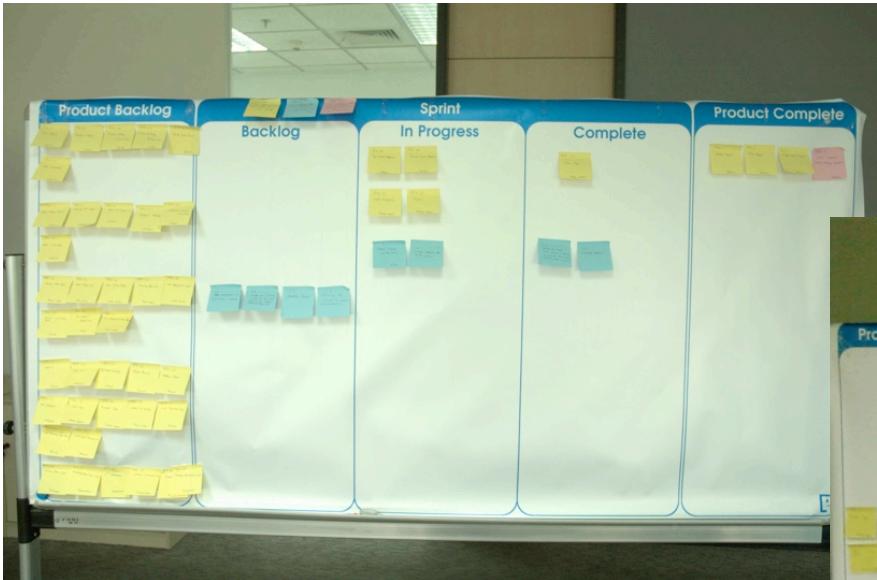
Small 小的

TTestable 可测试的

Examples 例子

| Issue # | Description | Est. | Eff. |
|------------------|--|------|------|
| Very High | | | |
| 1 | Finish database versioning | 16 | KH |
| 2 | Get rid of unneeded shared Java in database | 8 | KH |
| 3 | Add licensing | - | - |
| 3 | Consumer user licensing | 16 | TG |
| 4 | Demo / Eval licensing | 16 | TG |
| 5 | Analysis Manager | - | - |
| 5 | File formats we support are out of date | 160 | TG |
| 6 | Round-Trip Analyses | 200 | MC |
| High | | | |
| 7 | Enforce unique names | - | - |
| 7 | in main application | 24 | KH |
| 8 | in report | 24 | AM |
| 9 | Admin Program | - | - |
| 9 | Delete users | 4 | JM |
| 9 | Analysis Manager | - | - |
| 10 | When items are removed from an analysis, they should show up again in the pick list in lower 1/2 of the analysis tab | 8 | TG |
| 11 | Query | - | - |
| 11 | Support for wildcards when selecting | 16 | TSA |
| 12 | Sorting of number attributes to handle negatives numbers | 16 | TSA |
| 12 | Horizontal scrolling | 12 | TSA |
| 13 | Population Genetics | - | - |
| 14 | Frequency Manager | 400 | TSM |
| 15 | Query Tool | 400 | TSM |
| 16 | Additional Editors (which ones?) | 240 | TSM |
| 17 | Study Variable Manager | 280 | TSM |
| 18 | Haplotype | 320 | TSM |
| 19 | Add icons for v1.1 or 2.0 | - | - |
| 20 | Pedigree Manager | - | - |
| 20 | Variable Derived Variable | 4 | KH |
| Medium | | | |
| 21 | Explorer | - | - |
| 21 | Launch tab synchronization (only show queries/analyses for logged in user) | 8 | TSA |
| 22 | Delete settings (?) | 4 | TSA |





Non-Functional Requirements

非功能性需求

练习 Exercise

为课程项目创建至少10个用户故事（记得
INVEST 模型）

Create at least 10 user stories from
the Class Project Epic
(Remember INVEST)

Sprint Backlog



- 每个人认领Sprint中的任务 Individuals sign up for work of their own choosing
 - 工作不是分配下去的 Work is never assigned
- 每天更新剩余工作的估算 Estimated work remaining is updated daily
- 任何团队成员都可以添加、删除或者更改Sprint Backlog Any team member can add, delete or change the sprint backlog
- Sprint中的工作会涌现 Work for the sprint emerges
- 如果工作不清楚, 定义一个大工作量的Sprint Backlog事项, 留到以后才分解 If work is unclear, define a sprint backlog item with a larger amount of time and break it down later
- 当我们更加了解我们的工作时, 更新剩余的工作 Update work remaining as more becomes known

Sprint Backlog



- 团队从产品Backlog中选择他们可以承诺在此迭代交付的事项 Team selects items from the product backlog they can commit to completing
- 创建Sprint Backlog Sprint backlog is created
 - 识别工作任务并估算工作量 Tasks are identified and each is estimated (1-16 hours)
 - 群策群力, 不是ScrumMaster一个人做 Collaboratively, not done alone by the ScrumMaster
- 考虑概要性的设计 High-level design is considered

Story 故事

(PBI-Product Backlog Item)

Task 任务

(SBI-Sprint Backlog Item)

Case Study

The team is not making any effort to manage itself. During Sprint Planning, they sit in silence until the Manager steps in and does the planning and makes the commitment for them. During the Sprint, they're not very clear on what needs to be done, and they wait for directions from the manager. And at the end of the Sprint, they don't really seem to care whether they met their goal.

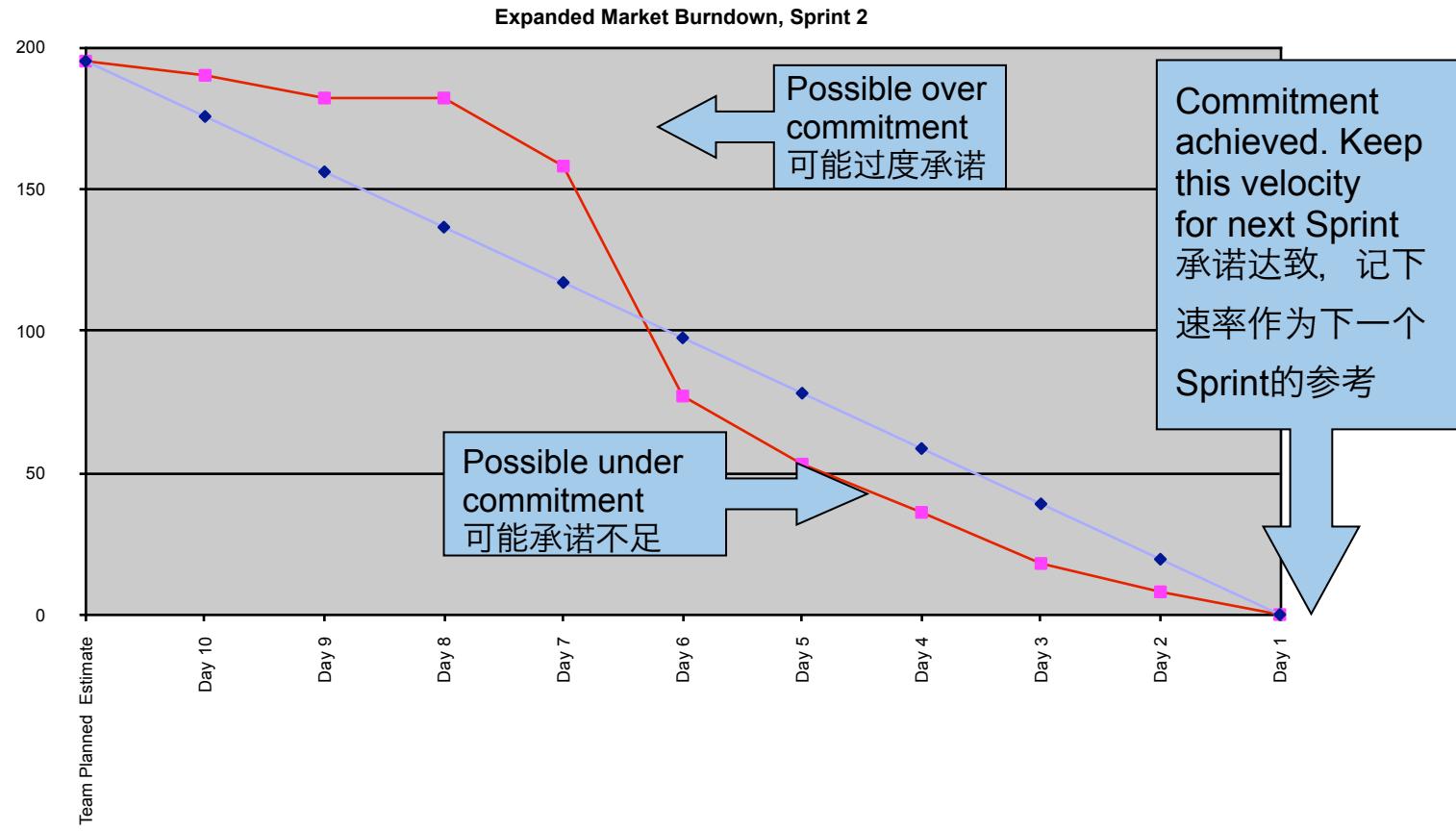
团队并不努力实现自管理。每一个Sprint计划会议，他们都沉默寡言，直到经理插进来做好计划并为他们制定承诺。Sprint进行期间，他们对需要完成什么不太清楚，常常等待经理的指令。而且在每一个Sprint结束时，他们好象并不关心他们是否实现了他们的目标。

Burndown Charts 燃尽图



- 每天更新 Updated daily, usually during the daily stand-up
- 跟踪剩余工作量 Represent the amount of work remaining
- 不同的方式 Different approaches to creating burndown charts
 - 跟踪任务层剩余工作量 Estimated remaining effort
 - 跟踪完成 Tracking Done

Burndown Charts 燃尽图



Roles



- Product owner
- ScrumMaster
- Team

Artifacts



- 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

- Scrum项目通过一个接着一个的Sprint来推进 Scrum projects make progress in a series of “sprints”
 - 和极限编程的迭代是一致的 Analogous to Extreme Programming iterations
- 通常每个Sprint长度为2–4周,最长不超过一个月 Typical duration is 2–4 weeks or a calendar month at most
- 使用一个固定的Sprint长度, 带来节奏感 A constant duration leads to a better rhythm
- 在Sprint中, 函盖产品的设计、编码和测试 Product is designed, coded, and tested during the sprint

Sprint



- 根据你们能够承诺一个Sprint跨度中不做变更来计划你们的 Sprint长度 Plan sprint durations around how long you can commit to keeping change out of the sprint
- 严格的时间箱 Strictly Time-Boxed

Case Study

One of the team members comes to the ScrumMaster and tells him that the Product Owner just asked her to add an Product Backlog item to the current Sprint. Right now, the team is a third of the way through the Sprint.

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

Sprint



- 能否取消一个Sprint ? Can a Sprint be terminated?
- Yes
 - 如果团队觉得他们不能实现Sprint目标, 团队可以取消Sprint Team can cancel the Sprint if they are unable to meet the Sprint goal
 - 如果业务环境变化所需, Product Owner可以取消Sprint Product Owner can cancel the Sprint if business circumstances require
- 重做Sprint计划会议 -- 任何已经开始的工作变成“浪费” Go back to Sprint planning -- any work performed is ‘waste’
- 很少这样做 ! **Very rarely done!**

Short Exercise: Different Types of Sprints? 不同类型的Sprint

1. 有没有一个所谓的“分析” Sprint, 用于收集所有的需求 ?
Is there such thing as an Analysis Sprint used to collect all requirements?
2. 有没有一个所谓的“测试”Sprint ?
Is there such thing as a Testing Sprint?
3. 有没有一个所谓的“稳定 (收尾) Stabilization Sprint” ? 如果有, 可能包括哪些工作 ?
Is there such thing as a Stabilization Sprint?
4. 如果一个项目需要比较多的基础结构及架构工作, 这些工作需要8周来完成, 是否第一个Sprint需要8周 ? 架构本身是否能称作Sprint的一个交付物 ?
If a project needs to do a lot infrastructure and architecture work upfront the work will need about 8 weeks to perform, can the first Sprint 8 be weeks long?
Can architecture be a deliverable of a Sprint?

Sprint Planning Meeting

Sprint 计划会议



Selection 选择

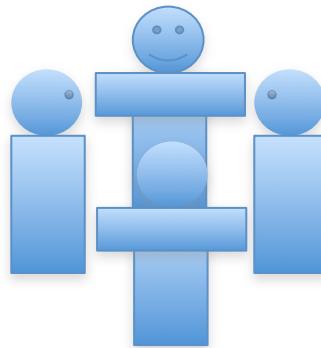
- Define the Sprint Goal 定义Sprint目标
- Select the Product Backlog the team can commit to 选择团队可以承诺交付的产品Backlog事项

Planning 计划

- Decide how to achieve the Sprint Goal 决定大概怎样实现Sprint目标
- Create the Sprint Backlog 创建起始的Sprint Backlog
- Estimate Sprint Backlog Items 估算Sprint Backlog事项

Daily Scrum 每日例（立）会

- 参数 Parameters
 - 每天进行 Daily
 - 15分钟 15-minutes
 - 站立进行 Stand-up
- 不是问题解决会议 Not for problem solving
 - 可以邀请其他人 Whole world is invited
 - 只有团队成员、ScrumMaster、Product Owner可以发言 Only team members, ScrumMaster, Product Owner, can talk
- 帮助避免其他不必要的会议 Helps avoid other unnecessary meetings



3个问题 3 Questions

- 我过去一天完成了什么？ What did I get **done** yesterday?
- 我今天会**完成**什么？ What will I get **done** today?
- 有什么障碍影响我的进展？ Are there any impediments slowing my progress?

这并不是对ScrumMaster的进展状况汇报，是大家在团队面前的承诺

These are **not** status for the ScrumMaster, they are commitments in front of peers

Short Exercise: Team & ScrumMaster

- 情景 1 (Scenario 1): **No-show ScrumMaster** *ScrumMaster在哪里？*

你们是开发团队成员：有一天，现在是9：05，你们的Daily Scrum应该在每天 9:00开始，可是ScrumMaster还没有出现。你们应该怎样做？ (For team members: It's 9:05, and your daily Scrum Meeting was supposed to begin at 9:00, but the ScrumMaster is a no-show. What should the team do?)

- 情景 2 (Scenario 2): **What Time Is It?** *现在是几点？*

你是ScrumMaster，有一位团队成员参加Daily Scrum时经常迟到，她说她的表显示她是准时的。而且，她拒绝支付迟到罚款。你应该怎样做？ (A team member repeatedly arrives late to the daily Scrum, saying that *her* clock indicates she's on time. On top of that, in defiance she won't pay the late fee.)

Thanks to Peter Borsella

Sprint 审核会议



Sprint Review (Demo)

- 团队展示Sprint的成果 Team presents what it accomplished during the sprint
- 通常是新功能的一次演示，或者也展示背后的架构 Typically takes the form of a demo of new features or underlying architecture
- 非正式 Informal
 - 规则：最多2小时准备 2-hour prep time rule
 - 不用幻灯片 No slides
- 全员参与 Whole team participates
- 邀请所有利益干系人和感兴趣的人参与 All stakeholders and interested parties are invited

Sprint回顾会议

Sprint Retrospective

- 持续改进 Continuous improvement
- 15–30分钟 15–30 minutes
- 每一个Sprint结束后进行 After every sprint
- 全员参与 Whole team participates
- 其他感兴趣的人欢迎参与（只有团队成员发言） Other interested parties are welcome (but only team speaks)
- 也可以用3个问题的方式来进行 3 Question format
 - 开始做什么 Start doing
 - 停止做什么 Stop doing
 - 继续做什么 Keep doing

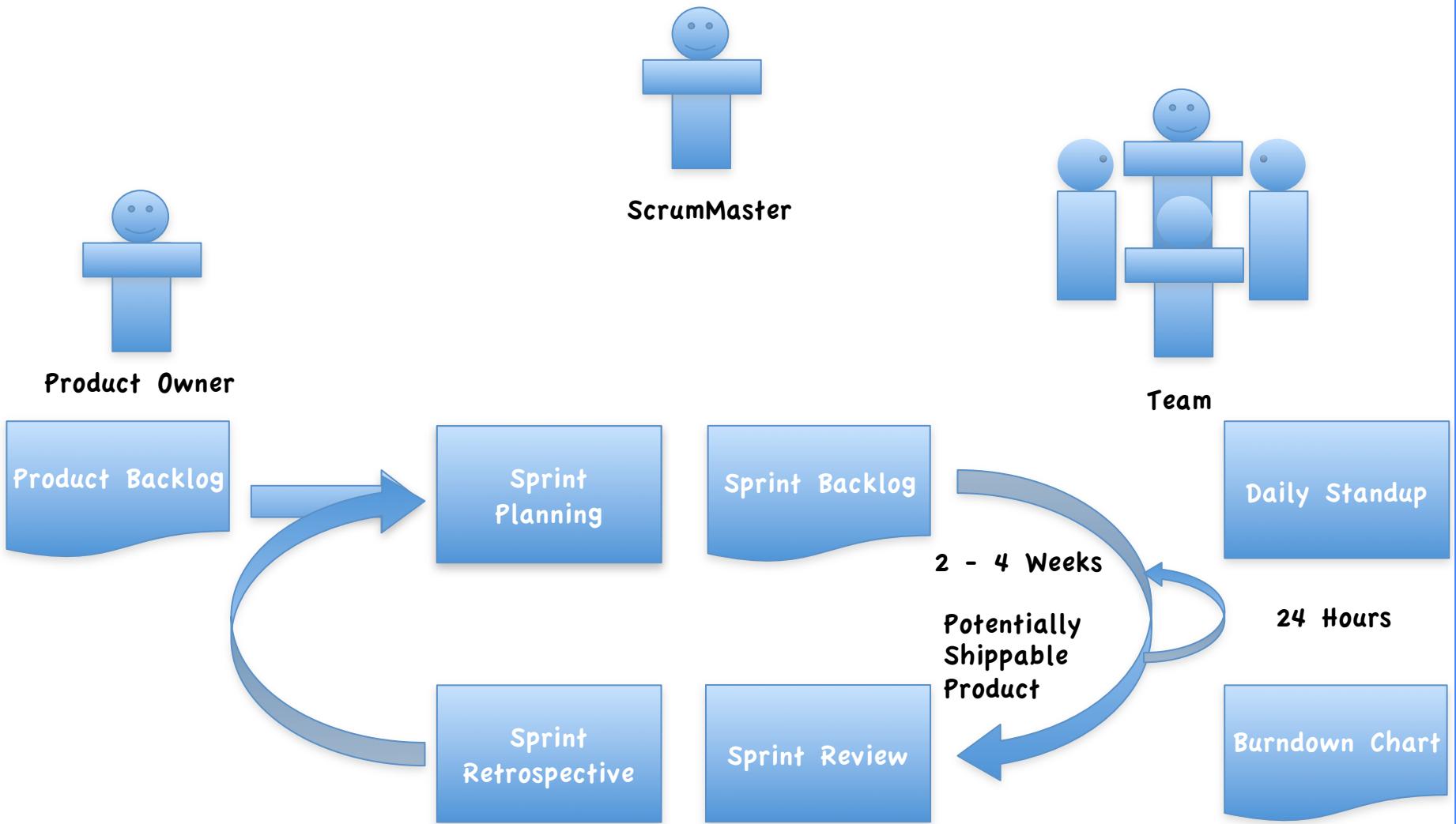
Case Study

A team-member speaks up and says he thinks the retrospective is a waste of time; several other team members murmur in agreement, and someone else suggests that the team stop doing the retrospective.

一个团队成员抱怨说他觉得Sprint回顾会议是在浪费时间。其他几个成员也在附和说有同感。甚至有一位提议干脆停止以后的回顾会议。

Scrum Visualized

Scrum框架视图化

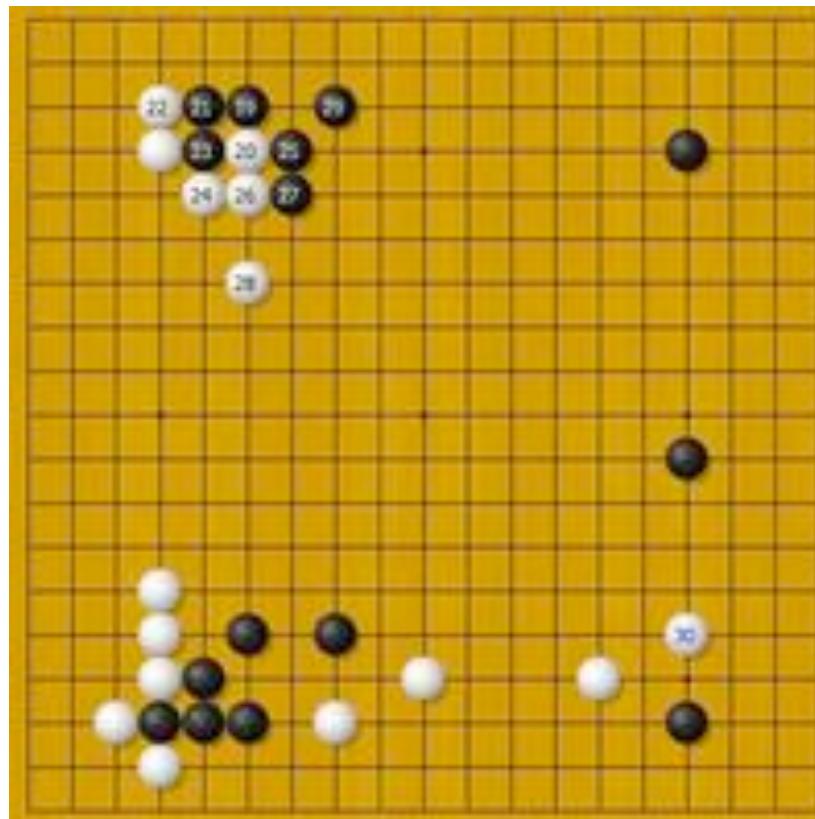


Courage, Openness, Focus, Commitment, Respect

Day 1 Retrospective

回顾

Day 2



Roles 角色



- Product owner (PO)
- ScrumMaster
- Team 团队

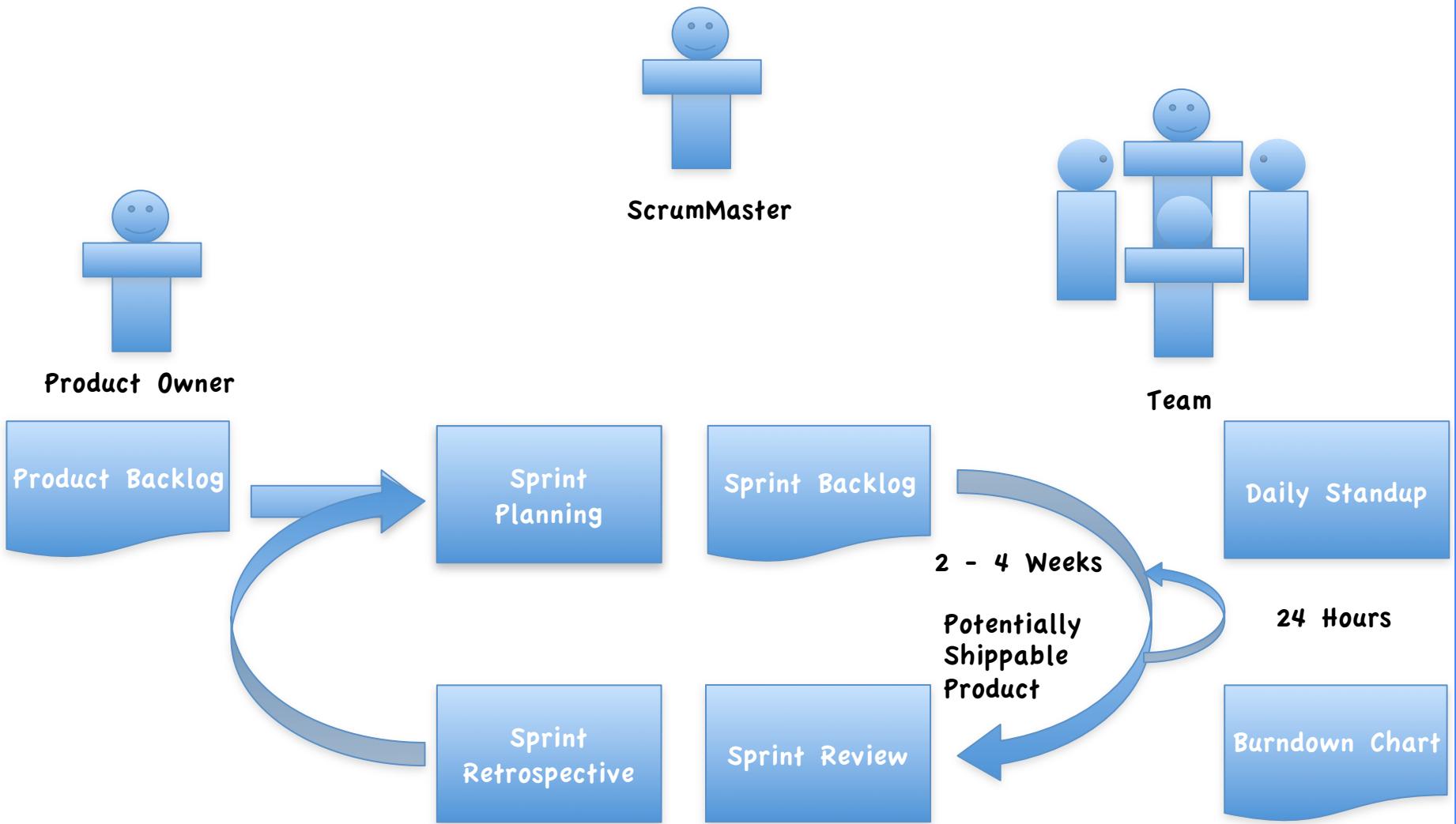
Artifacts 工件



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

Ceremonies 仪式

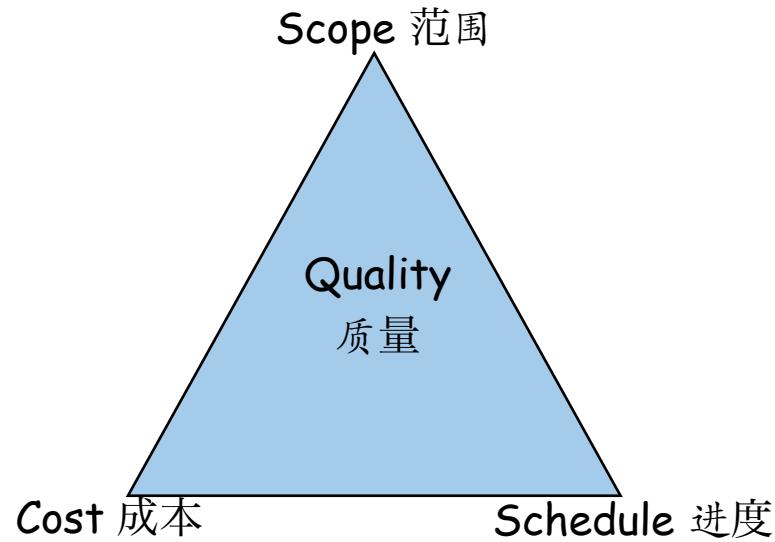
- Sprint
- Sprint Planning 计划会议
- Daily Scrum meeting 每日例会
- Sprint Review 审核会议
- Sprint Retrospective 回顾会议



Courage, Openness, Focus, Commitment, Respect

Project Constraints

项目约束



- Project constraints:
 - Scope 范围
 - Time 时间
 - Cost 成本

- Quality is a fourth dimension 质量是第四个约束
- One side of these cannot be changed without effecting the others 任意一个约束的改变会影响其它约束
- Project management is the process of organizing resources and activities to satisfy these constraints 项目管理就是通过组织资源和活动以满足这些约束的过程

Scope 范围

- Welcome changing requirements 欢迎需求变更
- The most effective method for conveying information is face-to-face 最有效的信息传递方式是面对面的交流
- Working software is the primary measure of success 可用的软件是最主要的成功度量
- Simplicity -- the art of maximizing the amount of work not done -- is essential 简单 ——使不用做的工作最大化 ——是根本的
- There is no scope creep in Scrum! Scrum中没有范围蔓延一说

Schedule 进度

- Our highest priority is to satisfy the customer through the continuous delivery of valuable software 我们的最高优先任务是通过持续的交付有价值的软件来使客户满意
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference for the shorter timescale 频密的交付可用的软件，间隔从2周到一个月，间隔越短越理想

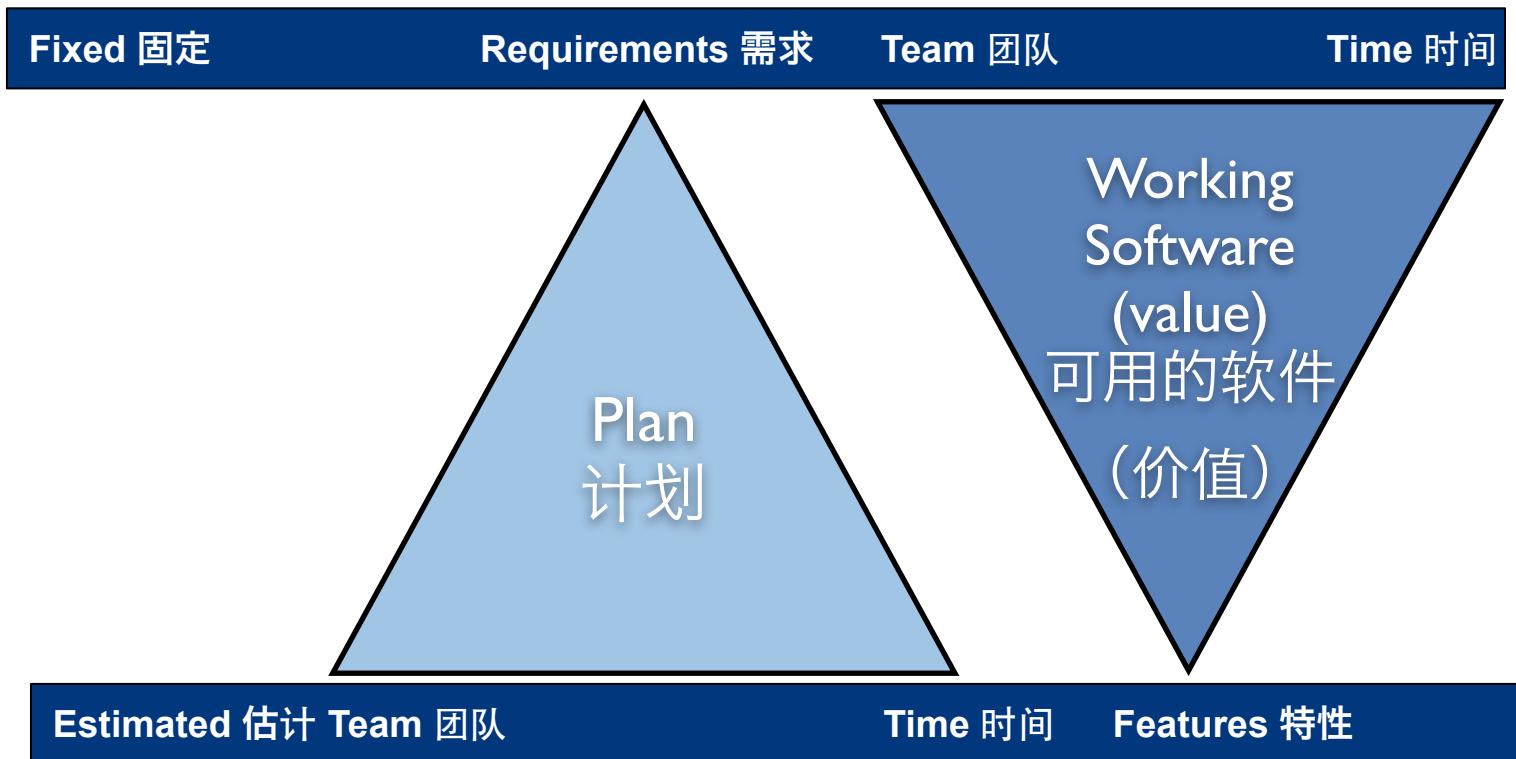
Cost
成本

Quality 质量

- Our highest priority is to satisfy the customer through the continuous delivery of valuable software 我们的最高优先任务是通过持续的交付有价值的软件来使客户满意
- Continuous attention to technical excellence and good design enhances agility 持续的注重技术的优秀性及好的设计增强敏捷能力

Traditional vs. Agile

传统 vs. 敏捷



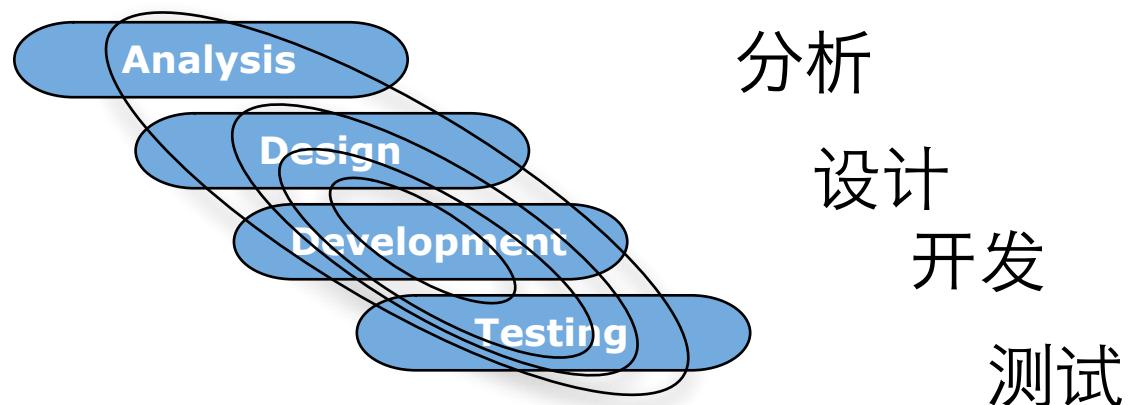
Productive Scrum

高效的 Scrum

Done
完成

Done !

Definition of Done 完成的定义

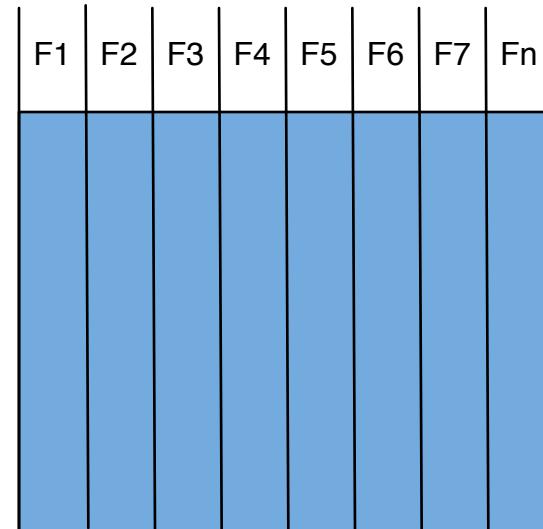
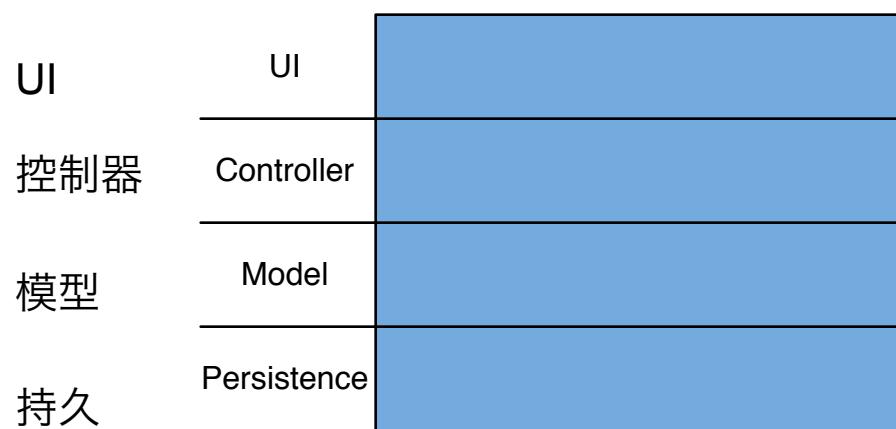


Traditional vs. Agile

Horizontal vs. Vertical

传统 vs. 敏捷

水平 vs. 垂直



Ready
准备好

Self-organized teams

自组织团队



UPerform Your Agile and Project Performance Partner

Backlogs, Planning, and Tracking

Backlogs, 计划及跟踪

Product Owner



- Decides on product vision 确定产品的愿景
- Responsible for the profitability of the product (ROI) 对产品的收益负责 (ROI 投入的回报率)
- Prioritize features according to market value: makes sure highest value items are being worked on first 按照市场价值对需求特性排优先级：确保最高级别的事项先被实现
- Accept or reject work results 接受或者退回产出的结果
- In an 'ideal' agile environment, the PO will develop functional test cases to validate that the requirements have been met 在“理想”的敏捷环境中，PO会创建功能性测试用例用于验证需求是否被满足

Product Backlog

产品Backlog



- Contains a *dynamic* list of all features 一份动态的特性清单
- Can be updated anytime, by anyone on the team 任何人可以在任何时候更新它
- No pre-defined format 没有预定义的格式
- Product Owner owns this artifact PO负责这个工件
- Embrace change 拥抱变化
- No “Scope Creep” in Scrum Scrum中没有“范围蔓延”
- Provides a means for allowing change to be captured and managed 提供了一个机制使变更可以被捕捉和管理
- No single standard way to capture product backlog 没有一个单一的方式来捕获产品Backlog事项

Agile Requirements 敏捷需求管理特征

- Active team (pig) involvement is imperative 活跃的团队（“猪”）参与非常重要
- Teams must be empowered to make decisions 获得信任及授权的团队进行决策
- Requirements emerge as product is developed 随着产品开发进展，需求是演进及涌现的
- Minimum required to continue work 只需要刚好足够的信息就可以前行
- Requirements are small 需求的颗粒度小
- Cooperation, collaboration, and communication are key 合作、配合及频繁的沟通与确认非常关键

Capturing Requirements

敏捷需求的形式

- Software requirements specification (SRS), Use-cases, Features, User Stories, etc. 软件需求规格说明书, 用例, 特性, 用户故事等等
- We like User Stories 我们推荐使用“用户故事”

Capturing Requirements 敏捷需求的形式

用户故事就是简短的特性声明，从某用户的角度说出来，遵从下面的格式：(a short statement from certain user's perspective)

“As a [user role] I want to [result] [so that [reason]].”

“作为一个（用户类型），我想（目标），因为（原因）”

- Who (user role)
- What (result)
- Why (reason)

Where are the details? 细节在哪里？

Epics vs. Implementable Stories 史诗故事 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

As a recruiter I want to
use the recruiting website
to find candidates for
open positions.

作为一个招聘者，我想创建一个工作职位
As a recruiter I want to be able
to create a job posting.

作为一个招聘者，我想审阅对
工作职位有回应的人的简历
As a recruiter I want to be able
to view the resumes of people
who have responded to a job
posting.

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
As a recruiter I want to be able
to search the database for
candidates by skill.

怎样编写更好的用户故事？ How to write better stories?

INVEST 模型

User Stories are not Use Cases

用户故事 不是 用例

A user story is to a use case as a gazelle is to a gazebo. (就如：gazelle 与 gazebo 两个词之间的同与异)

- Alistair Cockburn



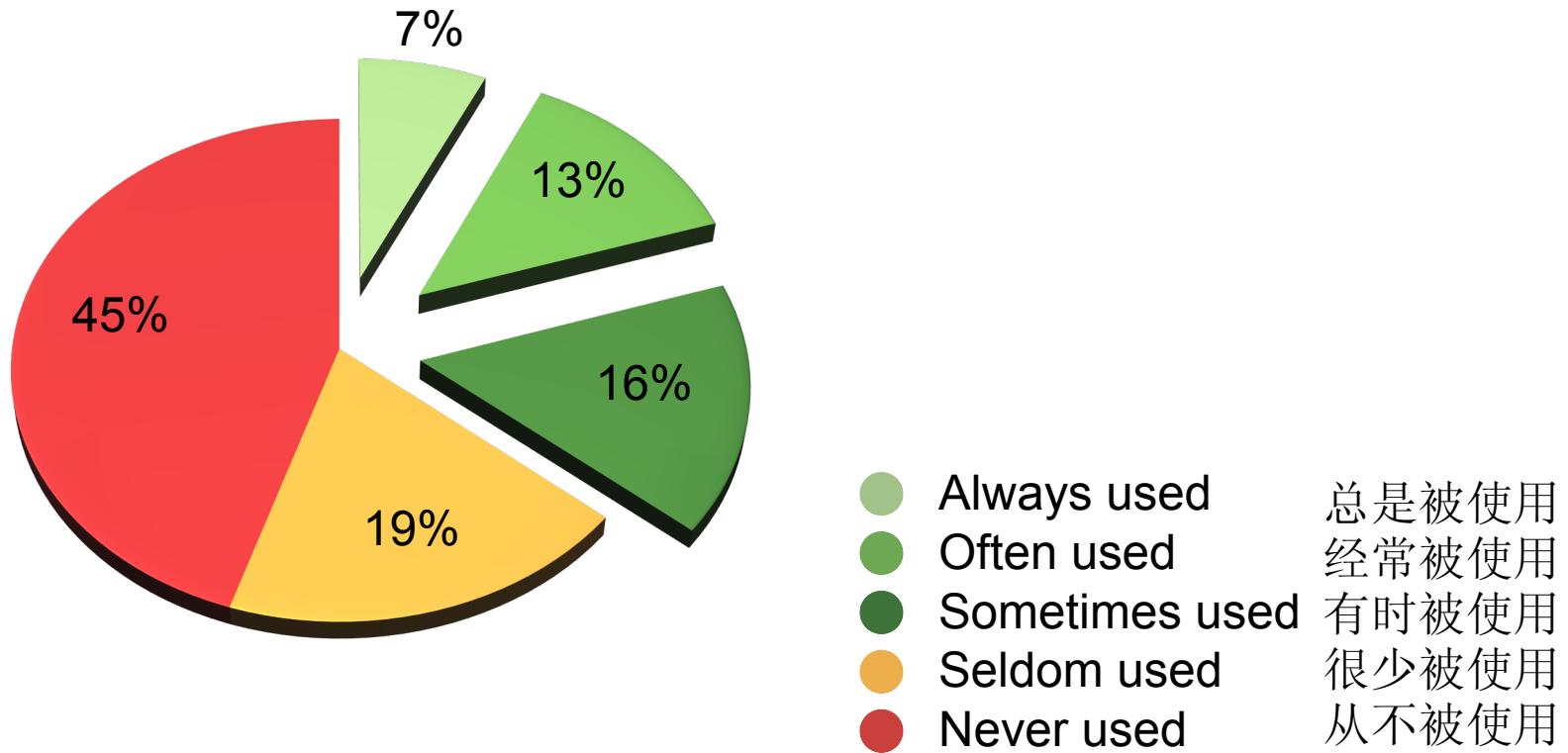
<http://animal.discovery.com/mammals/gazelle/pictures/gazelle-picture.jpg>



<http://www.gazebo-plans.com/Beachside%20Gazebo.jpg>

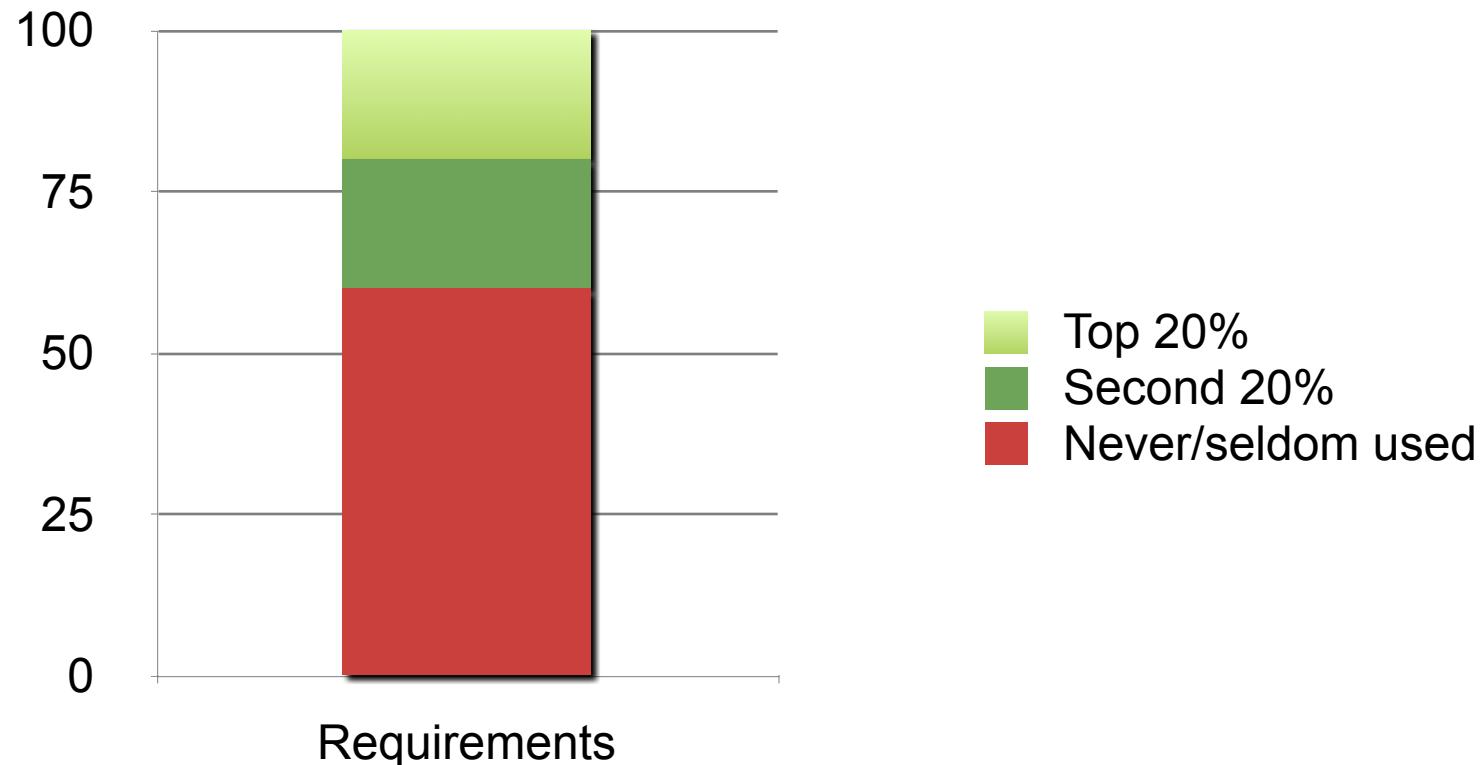
Quality Starts from 质量从何开始 --

Requirements 需求



The Standish Group
XP 2002

Pareto Principle 柏拉图原则 (80/20 rule)



Prioritization 优先级排序

Techniques 技巧

Exercise:

Prioritize your Product Backlog using the 100 Point Technique

使用100分技巧把产品Backlog作优先级排序

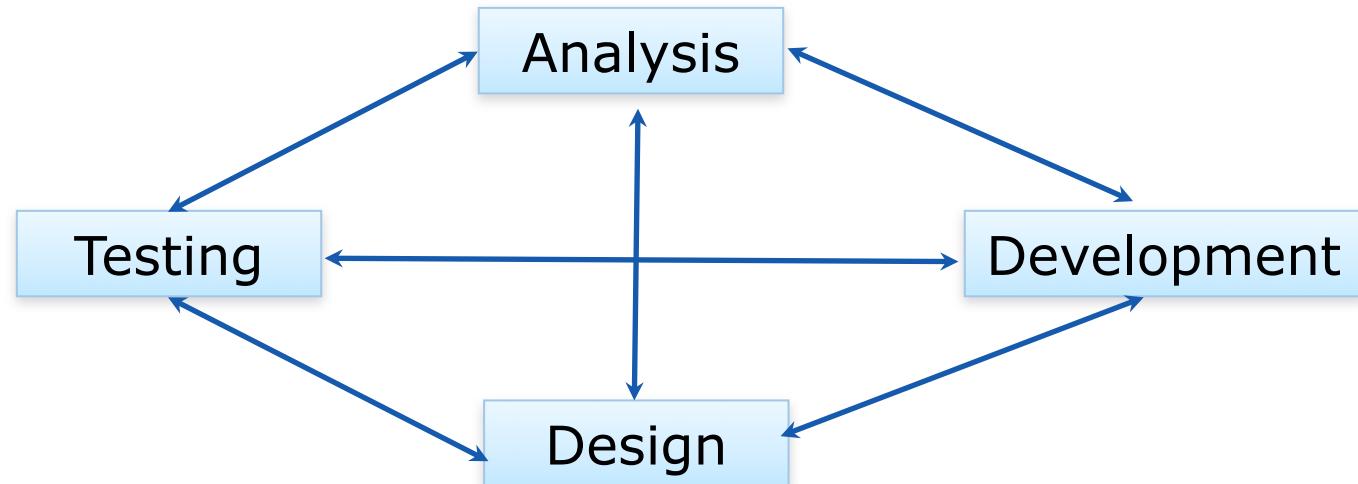
Quality

质量

关注测试及质量

Pay Attention to Testing & Quality

- 敏捷过程非常重视测试及质量（敏捷原则） Agile processes place very strong emphasis on quality and verification through testing (Agile Principles)
- 需求也可以是测试驱动的，质量也源自需求（“T” in INVEST） Quality, and testing, starts with requirements – Look at the “T” in INVEST
- 敏捷中测试是一个持续的活动，并不是一个阶段 In Agile, testing is a continuous activity, not an end-of-project ‘phase’
- 我们期望所有的工作产出都有测试能够验证,否则不能确认其完成 An activity is never complete unless there is a test for it



Test Driven Development

测试驱动开发

- 先创建测试, 然后交付工作以通过这些测试 Create tests, then create work to pass the tests
 - 接受测试驱动开发 Acceptance Test Driven Development (ATDD)
 - 测试驱动开发 Test Driven Development (TDD)
- 最短的反馈周期 Achieve shortest feedback loop
- 促使最简单的, 刚好足够的设计与实现 Urge simplest, just enough design and implementation
- 所有的工作产出都有测试能够验证, 否则不能确认其完成 No work should be done if there is no test to validate it
- 测试影响及驱动设计, 既是设计活动也可以是需求分析活动 Tests imply and drive design – testing is also a design activity

Benefits 好处

- Very high (perfect) test coverage 非常高的（完美的）测试覆盖
- 100% pass rates 100%通过率
- High quality 高质量
- The ability to move very quickly 有能力快速前行
- Better design 更好的设计
- Confidence to refactor 有信心去重构
- Low accumulation of Technical Debt 低的技术债务积累

Exercise:

Reconsider your User Stories relative to Testability and Quality, and develop one or more test cases for the top 5 User Stories.

考慮你手上的故事的可测试性及质量，为头5个用户故事写出1个或者多个测试用例

Did you change/improve any of your User Stories to make them more testable?

在这个过程中，有没有修改／改进任何一个故事，使之更可测试？

小案例 Case Study

In a product company, a team is finishing every Sprint with a large number of defects remaining. After 5-6 Sprints, the number of defects is so large that the system becomes very unstable, and the team has to stop and spend an entire Sprint doing nothing but fixing bugs.

在一个产品公司，一个开发团队每一个Sprint结束时都遗留很多缺陷。过了5—6个Sprint之后，缺陷的数量多得令系统非常难以捉摸，团队只好停下开发工作，要花整整一个Sprint的时间来修补问题。

Thanks to Pete Deemer

Agile Estimation

敏捷估算

Exercise: Relative Size

相对大小——国家面积

Estimating - Hours vs. Story Points

估算 —— 小时数 vs. 故事点

- No 'standard' or agreement on which to use 没有“标准”或者协议指定用什么单位
- A key feature of Story Points is that they can be estimated 故事点的重要特点是
可以被估算
 - These are a relative measure of effort 使用的是相对的工作量规模
 - Team based 'planning poker' is a common method for building the estimate 基于团队估算的“计划扑克”方法是敏捷团队中比较常见的
 - Planning poker increases correctness of estimate 计划扑克增加估算的正
确度
 - Provides a forum to socialize the story across the team 提供一个方式使
团队一起理解故事
- Some (unreferenced) studies indicate people are better at estimating effort
than they are at estimating time 有一些研究表明人们通常更善于估算相对工作量,
相比估算绝对值时间而言

敏捷估算 Agile Estimation

另外，使用相对工作量规模估算：

- 允许交付条件的不同（比如谁来完成工作等）
- 允许自修正的开发速度（比如对工具用熟练时速度变快，但故事的规模应该是不变的，只是团队速率高了）
- 通常比用估算绝对数值更快更容易

Using relative effort estimation:

- Allows for change in delivery condition (ie. who to complete the work)
- Allows for velocity to self-correct (ie. competence improves on certain tool as it goes)
- Normally faster and easier than using absolute values

Planning Poker 计划扑克

团队里的每一人会用计划扑克（或普通的扑克牌）来代表相对投入的大小：

Each member of the team uses Planning Poker card to represent relative size of effort

| | | |
|-------|---|----|
| Ace | = | 1 |
| Deuce | = | 2 |
| Three | = | 3 |
| Five | = | 5 |
| Eight | = | 8 |
| Jack | = | 13 |
| Queen | = | 20 |
| King | = | 40 |



(For you math wizards, this is a distorted Fibonacci series)

Exercise:

Play planning poker to estimate the effort required to implement the common Product Backlog items

使用团队计划扑克方法，对每一个产品Backlog的事项进行工作量估算

Velocity 团队速率

- What is our Velocity 你的团队迭代速率是多少？
 - Ideally this comes from historical data 最理想的是历史数据
 - Initially it may come from an estimate 开始时可能只能是一个估算
 - Once we have complete one Sprint, we have historical data 当我们完成一个Sprint, 就有了历史数据
 - Velocity prescribes and predicts how much work we can perform 团队速率描述及预测我们一个迭代能完成多少工作
- Can be units of effort or (ideal) hours 可以用工作量单位或者（理想）小时
- Measuring ‘done’ for velocity 速率中度量“完成”
 - Do you only count completed stories? 你是否只包含完全完成的故事？
 - How do you account for hours? 你如何用小时？

小案例 Case Study

A team has been doing Scrum for 6 months, and it has never gotten to 100% done at the end of a Sprint; it only gets to 80-90% done every Sprint.

一个开发团队已经采用Scrum 6个月了。但是他们从来没有一个Sprint实现100%的Sprint目标的交付。每一个Sprint他们只达到80 – 90%的交付承诺。

Thanks to Pete Deemer



[http://bucf.files.wordpress.com/2008/09/
general_of_the_army_dwight_d_eisenhower_1947.jpg](http://bucf.files.wordpress.com/2008/09/general_of_the_army_dwight_d_eisenhower_1947.jpg)

Plans are nothing;
planning is everything
计划本身不重要；
持续做计划才是硬道理

– Dwight D. Eisenhower
艾森豪威尔

Planning 做计划

- Product Backlog prioritized by Product Owner PO对产品 Backlog进行优先级排序
- Stories are estimated by team 开发团队对故事规模进行估算
- How much planning? 花多少力气做计划？
- Just-in-time JIT 适时地
- Release planning 发布计划

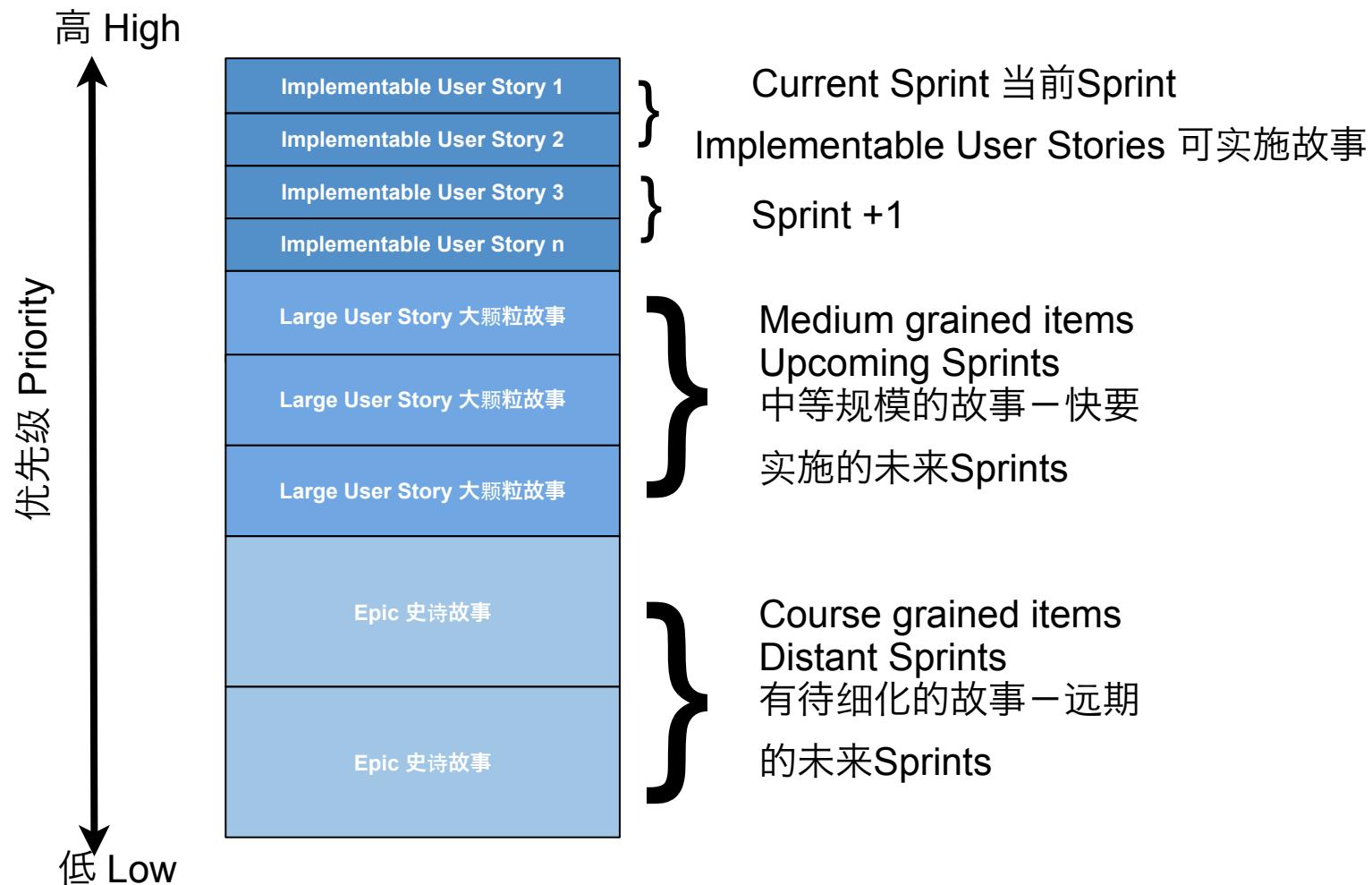
Release Planning – Issues

发布计划 —— 潜在问题

- Availability of historic Velocity 是否有交付速率历史数据
 - For **THIS** team “这个” 团队的
- User stories are high level 用户故事是概要性的
- Requirements will change 需求会变更

Product Backlog Planning

计划你的产品Backlog



Exercise :

- Using your smallest User Story estimate your Velocity 用你最小的用户故事估算你的团队速率
- Using your estimated velocity, create a release plan for all of your User Stories 利用这个估算的速率，为你手上的所有故事创建一个发布计划
- How many Sprints will be required to complete all of your user stories? 一共需要多少个Sprint来完成所有的故事？

Sprint Planning Sprint计划

- Tasks are added to turn product backlog into working product functionality 添加任务以将产品Backlog转变成产品功能
- Any team member can add, delete or change the Sprint backlog items 任何团队成员可以添加、删除或者更改Sprint Backlog事项
- No need for PO to approve 不需要PO的批准
- If work is unclear, define a larger item and break it down later 如果对工作了解不够, 定义一个较大规模的事项, 以后再把它分解
- Rule of Thumb: between 50 and 200 tasks per Sprint 经验法则 : 每一个Sprint分解出来50到200个任务

Tasks 任务

- Tasks are pulled, not assigned 任务是被拉动的，而不是分配下去
- A team member should work on only one task at a time 一个成员任何一个时间只做一个任务
- Tasks are estimated in relative units of effort 任务也应该用相对工作量规模估算
- Larger tasks are broken down when someone starts working on them 较大的任务，当有人开始开展其工作时进行分解
- Tasks are granular enough to be completed by one person in a single work period 任务颗粒度足够小，以使一个人能在单一时间段内完成
- Tasks that take longer than a single work period become obstacles 花了超过单一时间段的任务成为了进展的障碍

Sprint Backlog -> Tasks 任务

- Team needs to ask Product Owner questions to identify tasks to be performed 识别任务时，团队需要向PO提问
- Consider engineering, pre-requisite, environment, etc. to add relevant tasks 考虑工程技术、前提条件、环境等因素，添加相关的任务
- Teams (especially newly formed teams) often fail to ask enough questions to adequately define tasks 团队（特别是新组建的团队）经常提问不足，未能充分地识别任务
- Tasks should be estimated 应该估算任务的工作量
- Compare task estimates to story estimates -- is there (significant) gap? 比较任务估算和故事估算 —— 是否存在（大的）差异
- If so, note it immediately -- transparency is essential to success 如果有，立刻记录 —— 成功依赖于透明性
- Do you need to go back and reassess what you are committing to this sprint? 你需要返回重新评估你对这个Sprint的承诺吗？

Exercise :

Break down two or three of your User Stories into Tasks
(remember to start with the most important ones first –
these would be the ones you would put into your Sprint
backlog).

将你的2–3个故事分解成任务（记得：你应该从最重要的故事开始选择放入最近一个Sprint Backlog）

Don't forget to estimate your tasks. 不要忘记对任务进行估算

Did your task estimation roll-ups shed any new light onto
your User Story estimates? 你的任务估算的总和有没有对你的
用户故事估算提供新的信息？

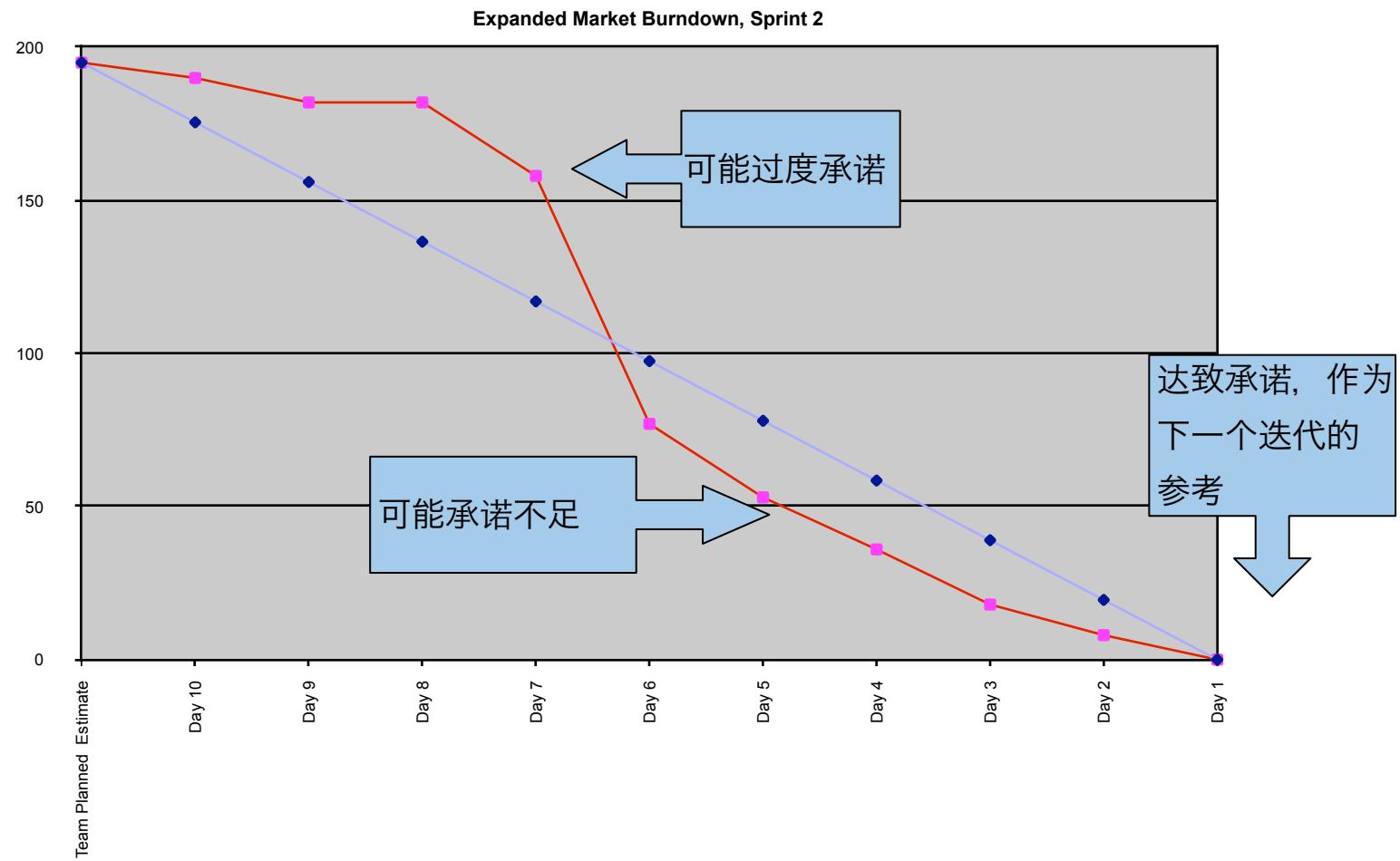
Tracking 跟踪

- Teams are measured by meeting goals 用是否达致目标来评估团队进展
- Measure remaining work 考量剩余的工作
- Measure by “Done” 考量“完成”
- Measure every day 每天都考量
- Post tracking results – transparency 公开跟踪的结果 – 透明性
- Use tracking to identify and address possible issues (as early as possible) 利用跟踪来发现和解决可能的问题 (越早越好)

Tools of the Trade 工具

- Visibility and transparency are critical 可见性和透明性是关键的
 - This should be large and posted for everyone to see 信息板
应该很大，公布出来所有人都很容易看到
- Use spreadsheet or other tool for complex projects (or to share across distributed organizations) 对于复杂的项目，可以使用电子表格或者其他工具（或者用于分布式开发的信息分享）
 - Visibility needs to remain high – post if possible 要保持高度可见性 —— 信息尽量公布出来

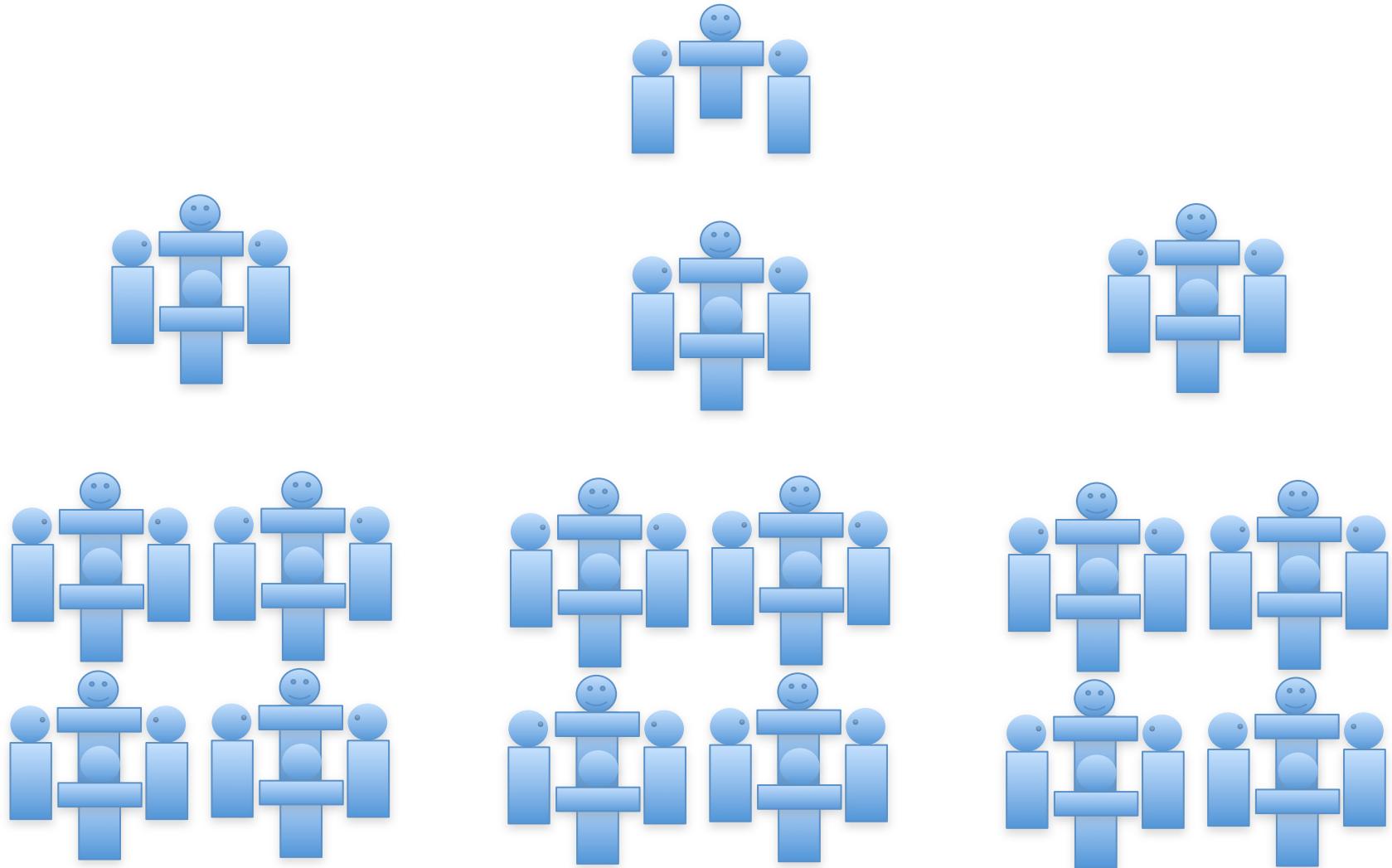
Burndown Charts 燃尽图探讨



Scaling Scrum 扩展

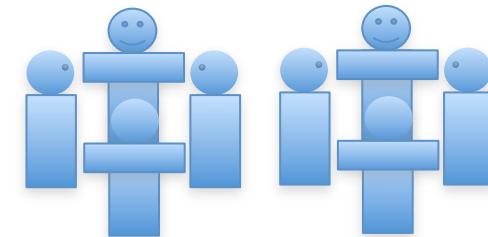
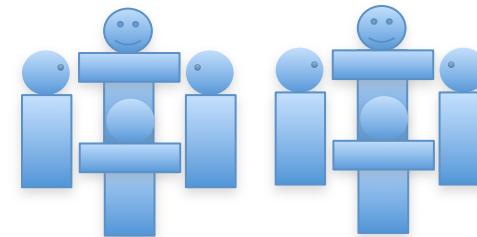
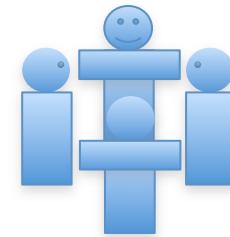
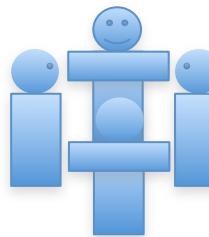
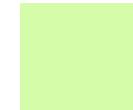
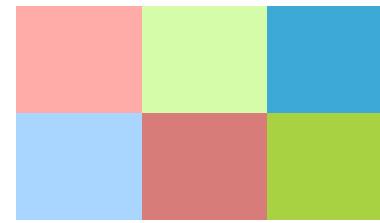
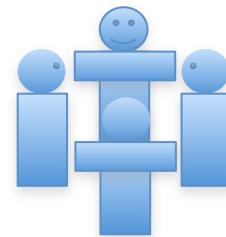
- Typical team is 7 ± 2 people 典型的团队人数是 7 ± 2
- Scalability comes from teams of teams 数个团队组成大团队来实现扩展
- Factors in scaling 扩展中的因素
 - Type of application 应用的类型
 - Team size 团队规模
 - Team dispersion 团队分布
 - Project duration 项目工期
- Scrum has been used on multiple 500+ person projects 有规模超过500人的一些项目使用Scrum

团队扩展——Scrum of Scrums



Integration Scrum Team

集成Scrum团队



Exercise:

The art of the possible
可能性的艺术

Review

重温

Roles 角色



- Product owner (PO)
- ScrumMaster
- Team 团队

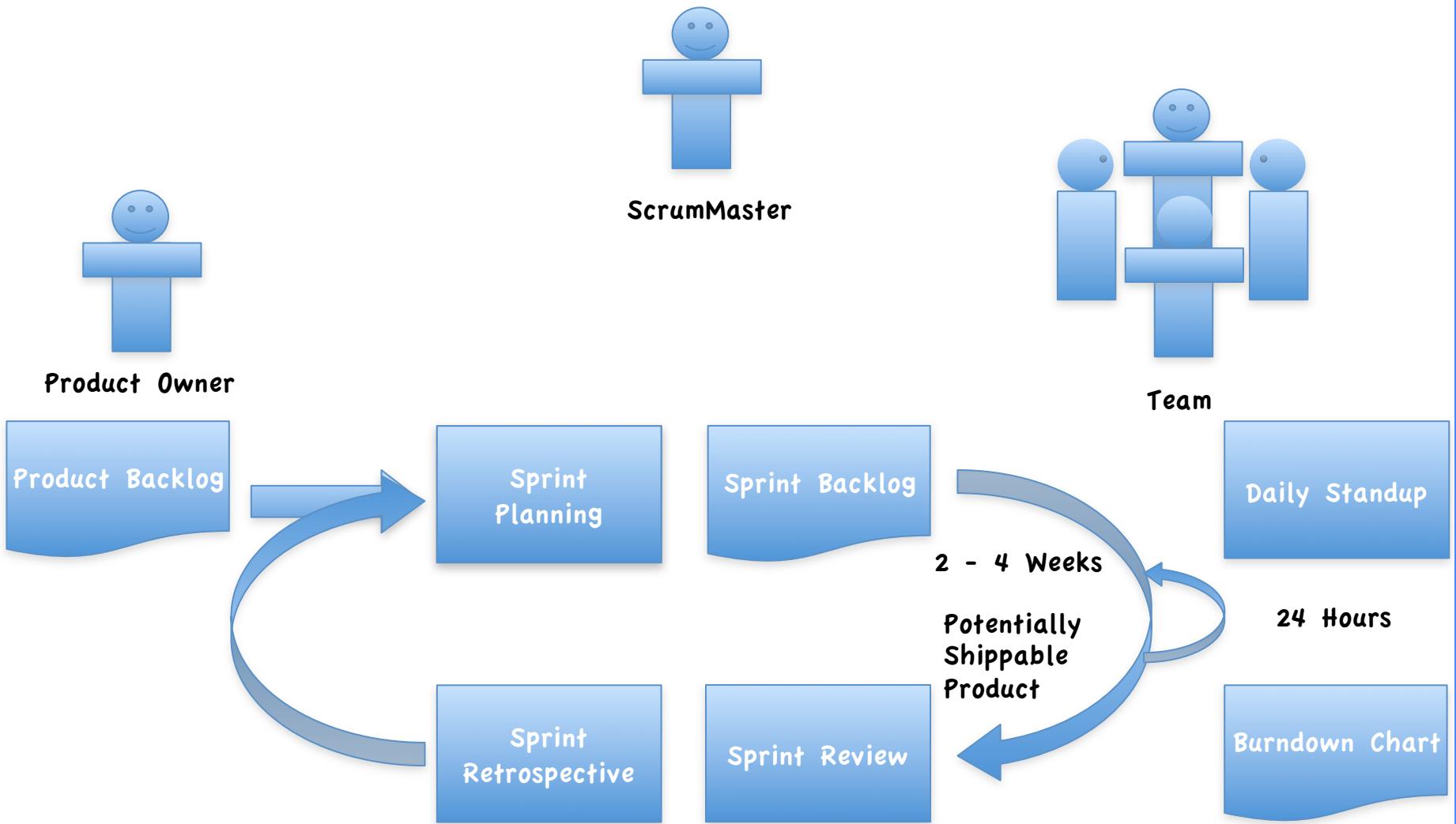
Artifacts 工件



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

Ceremonies 仪式

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



Courage, Openness, Focus, Commitment, Respect

#FAIL Revisited

重访——#失败

Next Steps

下一步

Invite you to join the following 2 Groups in Linkedin

邀请您参加下面2个Linkedin讨论组

- UPerform Alumni - 优普丰学友 (Private group for UPerform students to collaborate and share)
- 敏捷中国-Agile China (Open group for anyone to collaborate and share)

Final Q&A

Graduation Photo

合影

Invite you to join the following 2 Groups online

邀请您参加下面1个新浪微群及1个Linkedin讨论组

优尚会—UClub, Sina 新浪微群 : <http://q.t.sina.com.cn/982097> (Private group for UPerform students to collaborate and share)

LinkedIn: 敏捷中国-Agile China (Open group for anyone to collaborate and share)

References

- www.uperform.cn
- www.mountaingoatsoftware.com/scrum
- www.scrumalliance.org
- www.controlchaos.com
- scrumdevelopment@yahoogroups.com

Scrum 阅读推荐清单 Recommended Reading List

- 敏捷迭代开发管理者指南 *Agile and Iterative Development: A Manager's Guide* by **Craig Larman**
- Scrum敏捷项目管理 *Agile Project Management with Scrum* by **Ken Schwaber**
- 敏捷估计与规划 *Agile Estimating and Planning* by **Mike Cohn**
- 用户故事与敏捷方法 *User Stories Applied for Agile Software Development* by **Mike Cohn**
- Scrum敏捷软件开发 *Succeeding with Agile* by **Mike Cohn**
- Agile Retrospectives (敏捷回顾) by **Esther Derby and Diana Larsen**
- 敏捷软件开发生态系统 *Agile Software Development Ecosystems* by **Jim Highsmith**
- Agile Software Development with Scrum (敏捷软件开发 (使用Scrum过程)) by **Ken Schwaber and Mike Beedle**
- Scrum敏捷项目管理实战 *Scrum and The Enterprise* by **Ken Schwaber**
- Agile Testing: A Practical Guide for Testers and Agile Teams (敏捷测试) by **Lisa Crispin & Janet Gregory**
- The Software Project Manager's Bridge to Agility by **Michele Slinger & Stacia Broderick**
- Lots of weekly articles at www.scrumalliance.org

For Your Reference :

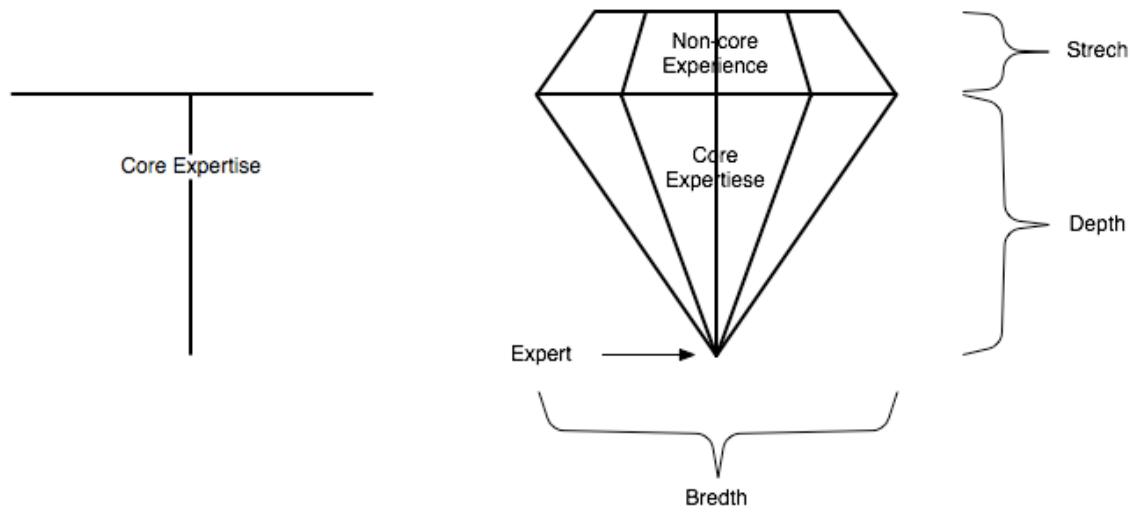
Teams &

Solutions to Common Problems

A team is a **small** number of people with **complementary skills** who are committed to a **common purpose, performance goals**, and an **approach** for which members hold themselves **mutually accountable**.

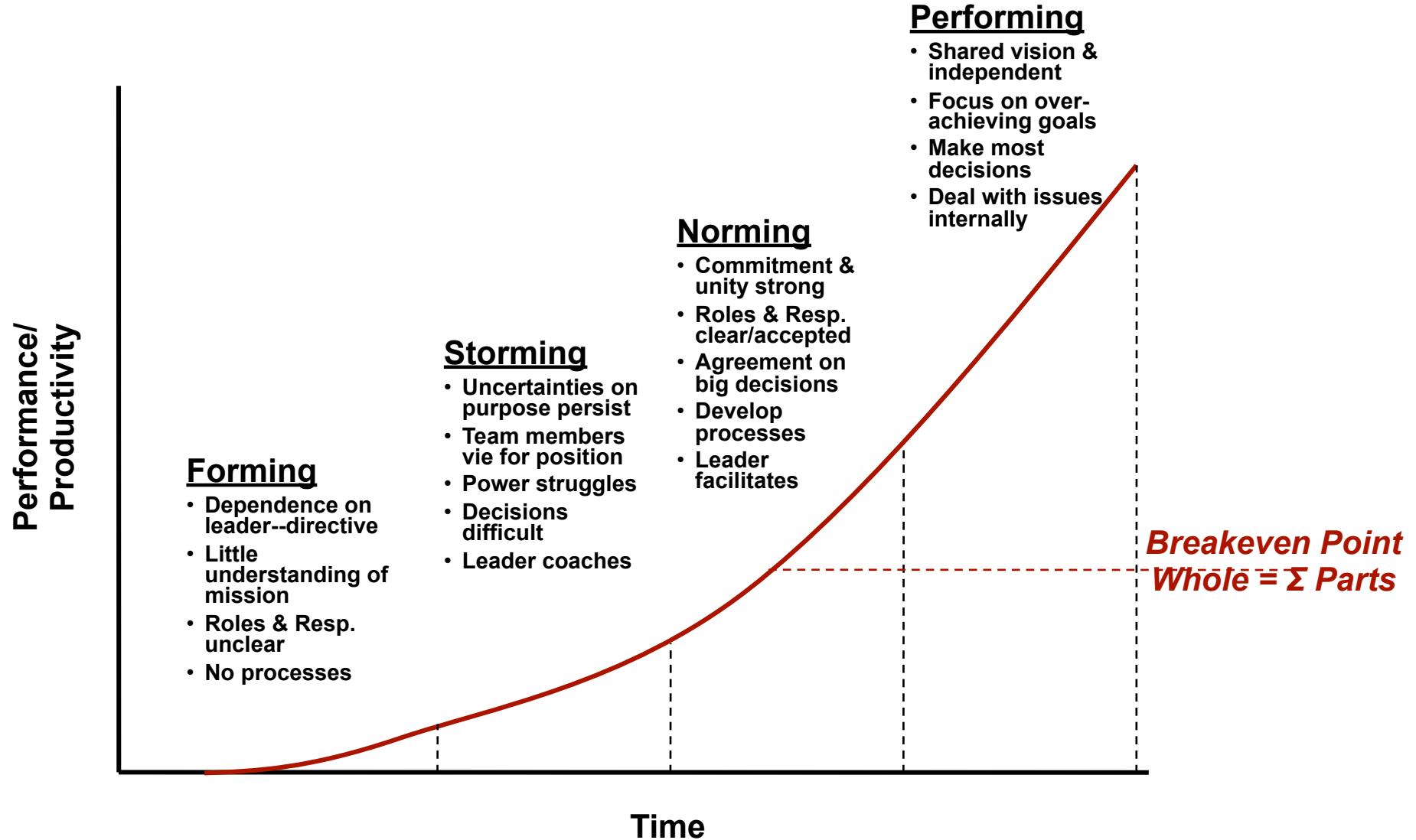
Source: Katzenbach and Smith, [This Wisdom of Team](#)

Diamond Principle



- Great teams start with great individuals
- The leader's job is to cut and polish
- Discuss:
 - How does the Diamond principle relate to Agile?
 - Do you have a team/culture of diamonds?

Tuckman's Team Performance Model



Scrum Teams - Forming

- Team is inundated with information:
 - Scrum framework
 - Project requirements
 - New people and relationships
- Can be overwhelming, and people will react differently. Expect:
 - Wait and see
 - Rebellion/Reversion
 - Passive-aggressive
 - Excitement
- This is an exciting and tiring stage. Be prepared to invest.
- Towards the end of this stage the team will start to note improvements in communication and teamwork, be pleased with the progress they have made, and start to raise issues.

Scrum Teams - Storming

- In early iterations the team will move to storming
 - This intentionally happens early in agile methods
 - This is normal and healthy -- we have to move through this stage to progress
 - It's possible this stage may be fairly mild
- It can be frightening and disheartening for this to happen early in a project. Realize it is normal and expected and DON'T GIVE UP.
- If the team makes it through this stage, it will continue to grow and improve
- If the team gives up things will revert back to the way they were, and more optimal performance will never be recognized.

Scrum Teams - Norming

- After several iterations the team will establish a comfortable pace:
 - process and practices will be institutionalized within the team
 - key impediments to achieving results will have been removed
 - individuals will be comfortable in performing their role
 - communications will be efficient and effective
- Results will likely be similar to pre-Scrum process, but some tangible benefits will also manifest themselves:
 - Delivery of potentially shippable product at the end of each sprint
 - good communication between stakeholders and the team
- Now is not time to become complacent: many challenges remain

Scrum Teams - Performing

- Teams can advance to performing:
 - Quickly or slowly
- The definition of "done" will develop so that the team can deliver a potentially shippable product every iteration
- There may be temptation to break the team up to take advantage of the new team members and to spread the concepts through the organization. DON'T!
 - Any change results in the team having to go through all stages again, resulting in loss of the benefits achieved by finally reaching the performing stage
 - It's as easy to develop an entirely new team and bring them through the stages
 - This is like splitting up a working relationship to create two to create two questionably viable ones. *A bird in the hand is worth two in the bush.*

Scrum Teams - Discussion

- Where are your teams relative to Tuckman's model?
- What is the role of the ScrumMaster relative to Tuckman's Model?

5 Dysfunctions of a Team - Scrum Context

The Five Dysfunctions of a Team

Dysfunction #1: Absence of Trust

Strategy for Overcoming:

- Identify and discuss individual strengths and weaknesses
- Spend considerable time in face-to-face meetings and working sessions

Dysfunction #2: Fear of Conflict

Strategy for Overcoming:

- Acknowledge that conflict is required for productive meetings
- Understand individual team member's natural conflict styles, and establish common ground rules for engaging in conflict

Dysfunction #3: Lack of Commitment

Strategy for Overcoming:

- Review commitments at the end of each meeting to ensure all team members are aligned
- Adopt a "disagree and commit" mentality—make sure all team members are committed regardless of initial disagreements

Dysfunction #4: Avoidance of Accountability

Strategy for Overcoming:

- Explicitly communicate goals and standards of behavior
- Regularly discuss performance versus goals and standards

Dysfunction #5: Inattention to Results

Strategy for Overcoming:

- Keep the team focused on tangible group goals
- Reward individuals based on team goals and collective success

From: *The Five Dysfunctions of a Team* by Patrick M. Lencioni; Jossey-Bass, 2002



Scrum Teams - Discussion

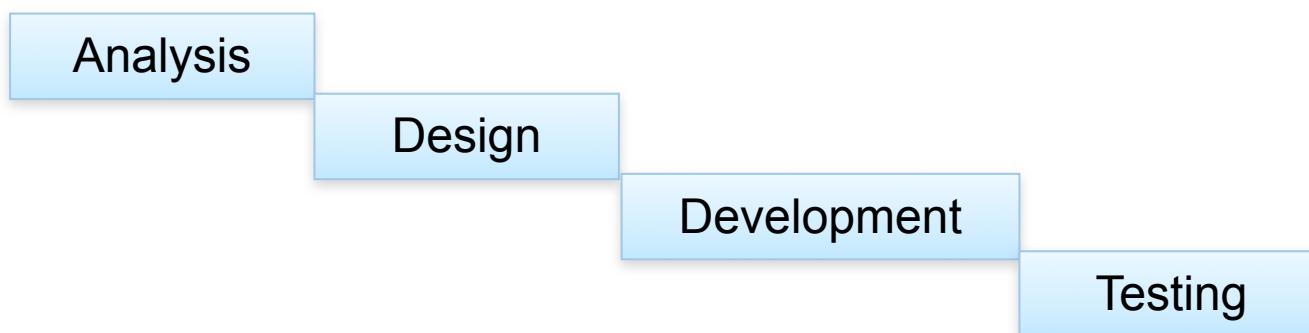
- Are any of the 5 Dysfunctions of a Team present now in your teams?
- If so, how are you going to address them?

Scrum Teams - Cross-Functional

- What is cross-functional?
- Alternative views/approaches

Agile vs. Waterfall

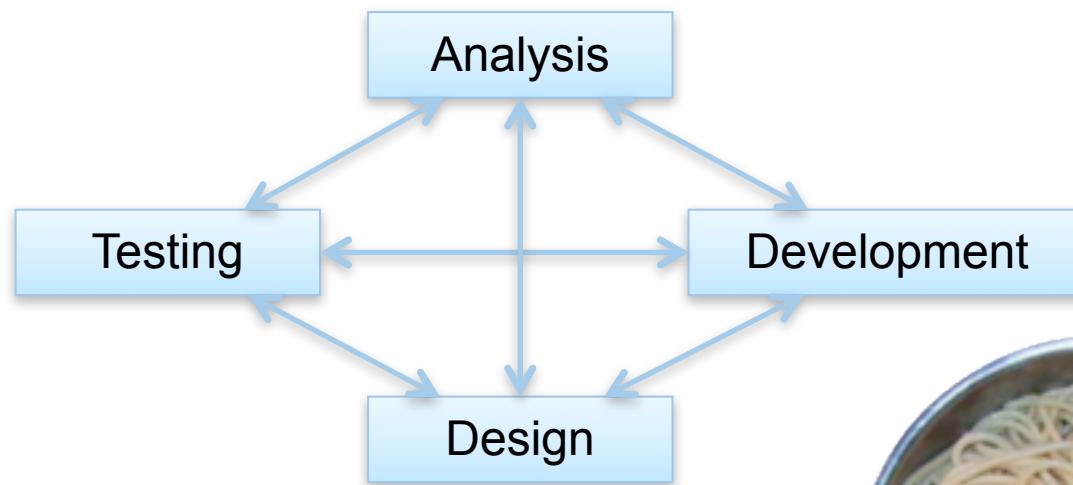
Waterfall



Still Waterfall



Agile



Scrum Solutions to Common Problems

Communication Breakdown

- Co-location
- Daily stand up
- Transparency
- “Team”

Poor user input

- Agile state of mind - embrace change
- Co-location
- Iterative development
- Demos
- Retrospectives

Stakeholder conflicts

- “Team”
 - Chickens and Pigs
- Product Owner
- Backlog management
- Iterative development

Vague requirements

- Co-location
- Iterative development
- Potentially shippable product demos

Poor cost and schedule estimates

- Backlog estimates
- Sprint planning
- Velocity
- Sprints
- Inspect/adapt

Hidden costs of going “lean and mean”

- Transparency
- Burndown charts
- Potentially shippable products at the end of each sprint

Failure to plan

- Backlog management
- Sprint planning
- Burndown charts

Poor architecture

- Simple/Iterative design (borrowed from XP)
 - YAGNI (you ain't gonna need it)

Late failure warning signals

- Sprints
- Burndown charts
- Potentially shippable product
- Commitment and “done”
- Transparency

Skills that don't match the job

- The Diamond principle
- Cross-functional teams