



敏捷开发方法

清华大学软件学院 刘强





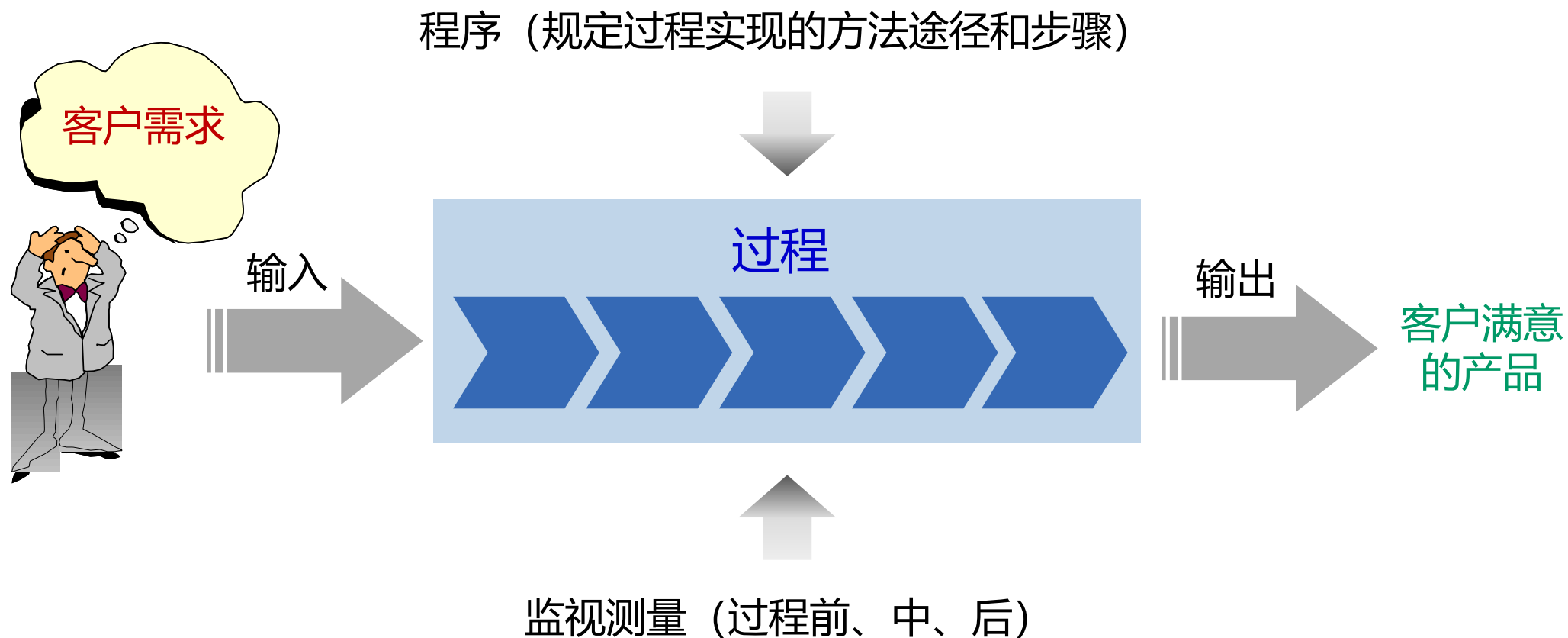
1	软件过程模型
2	敏捷开发方法
3	团队组织与管理

过程的含义

过程是一组将输入转化为输出的相互关联或相互作用的活动。

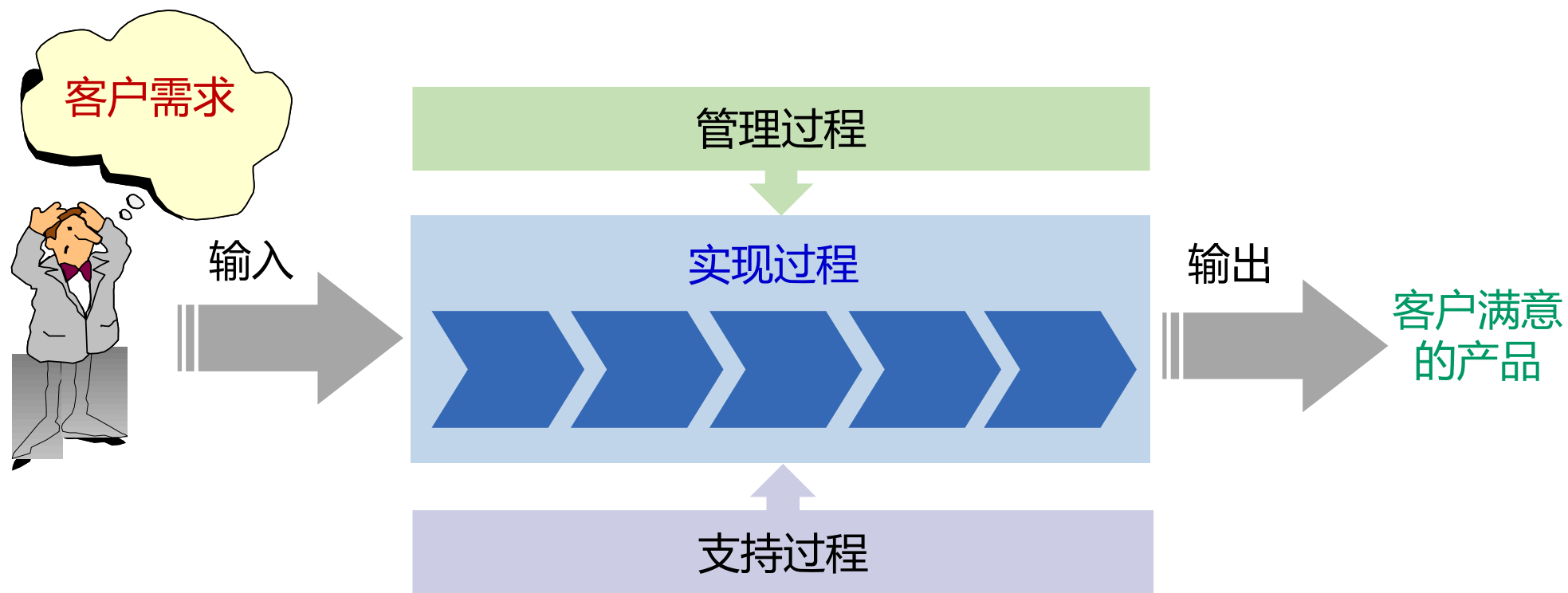


过程的含义

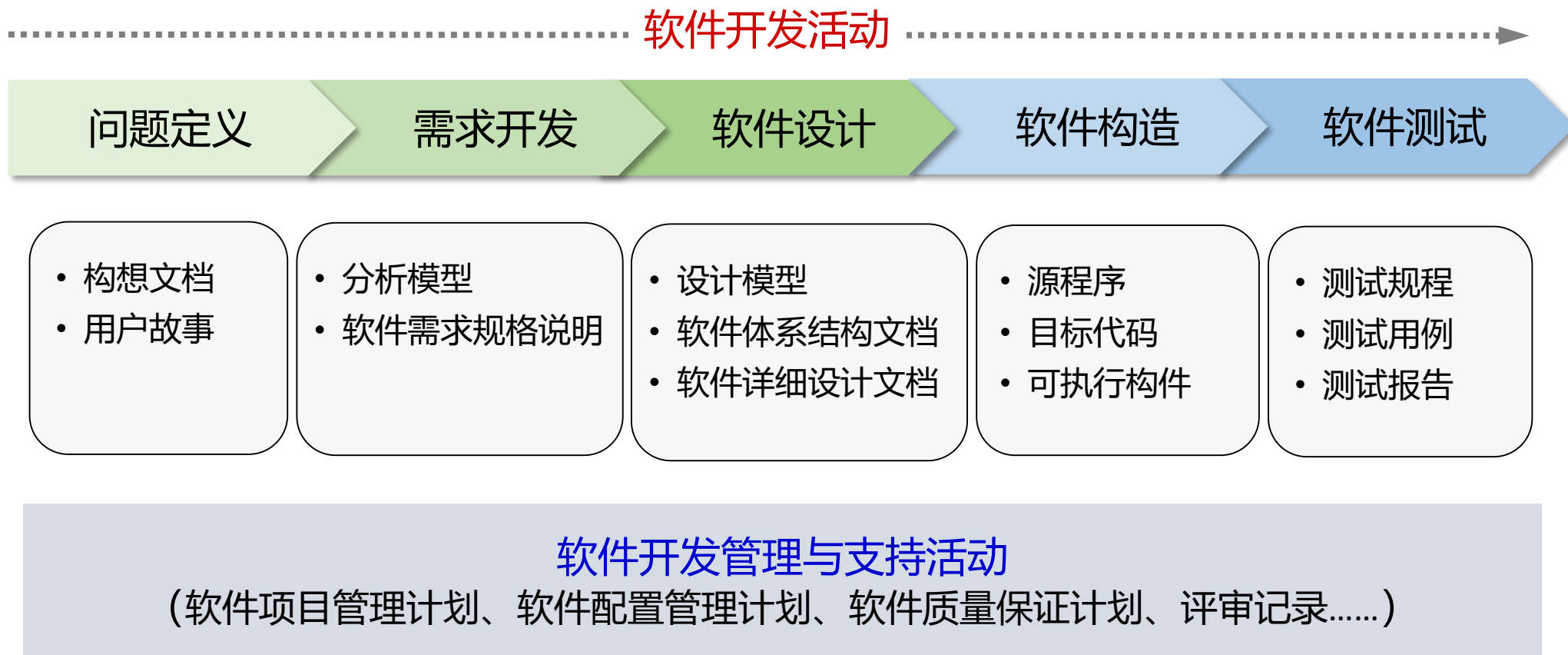


过程方法

过程方法是系统地识别和管理组织内所使用的过程，保证更有效地获得期望的结果。

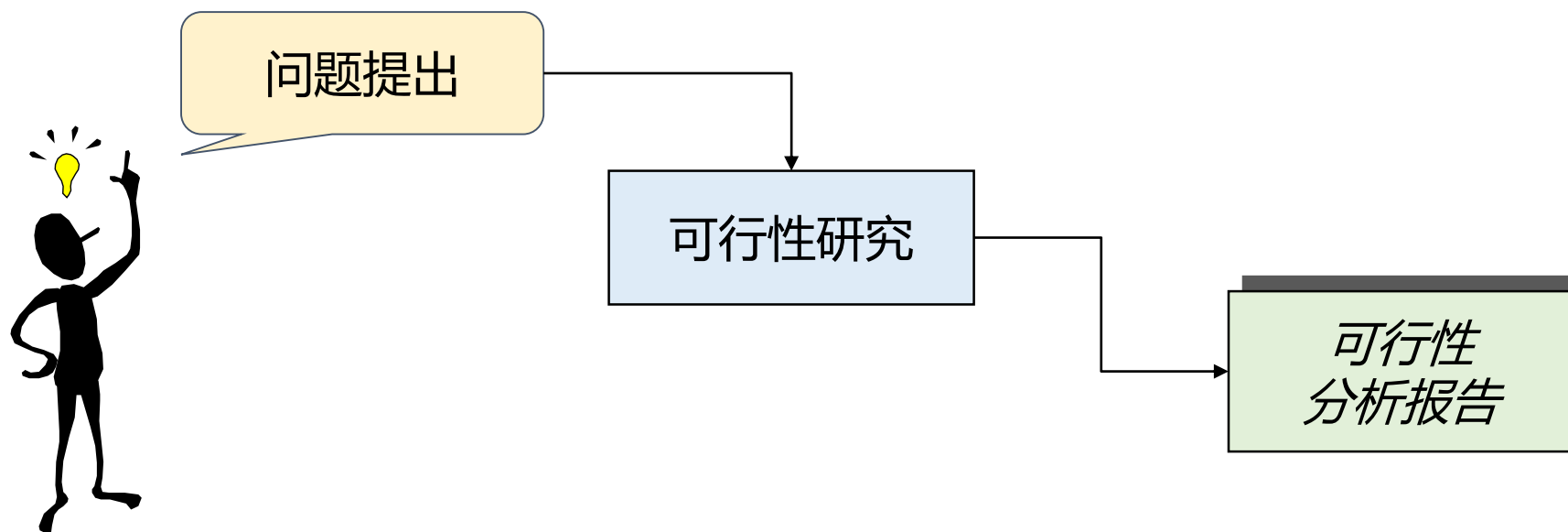


软件过程



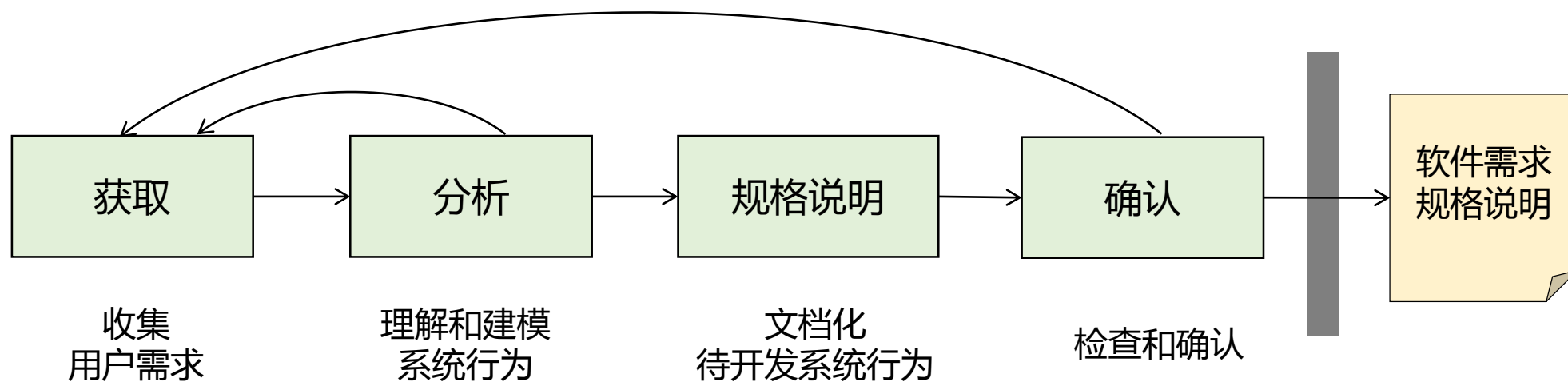
问题定义

问题定义：人们通过开展技术探索和市场调查等活动，研究系统的可行性和可能的解决方案，确定待开发系统的总体目标和范围。



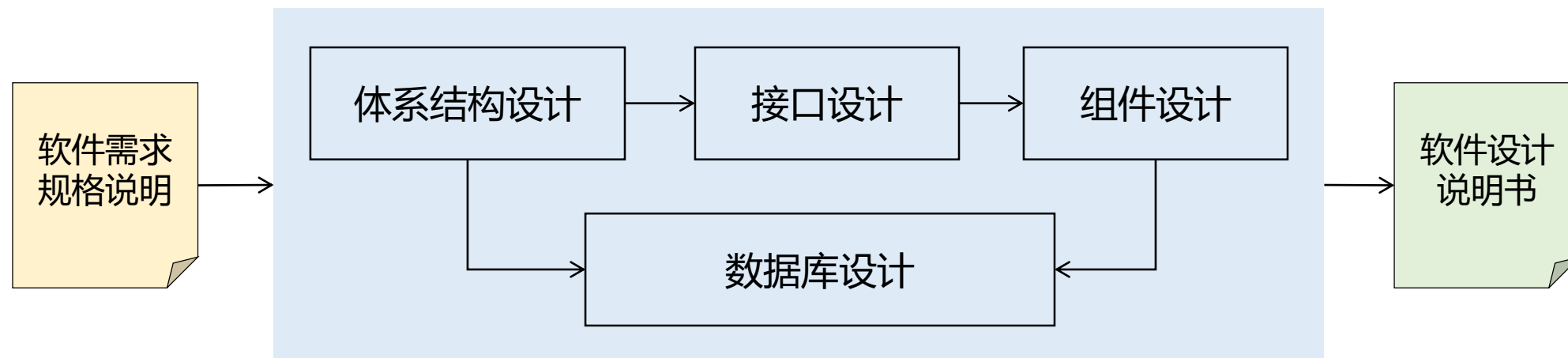
需求开发

需求开发：在可行性研究之后，分析、整理和提炼所收集到的用户需求，建立完整的需求分析模型，编写软件需求规格说明。



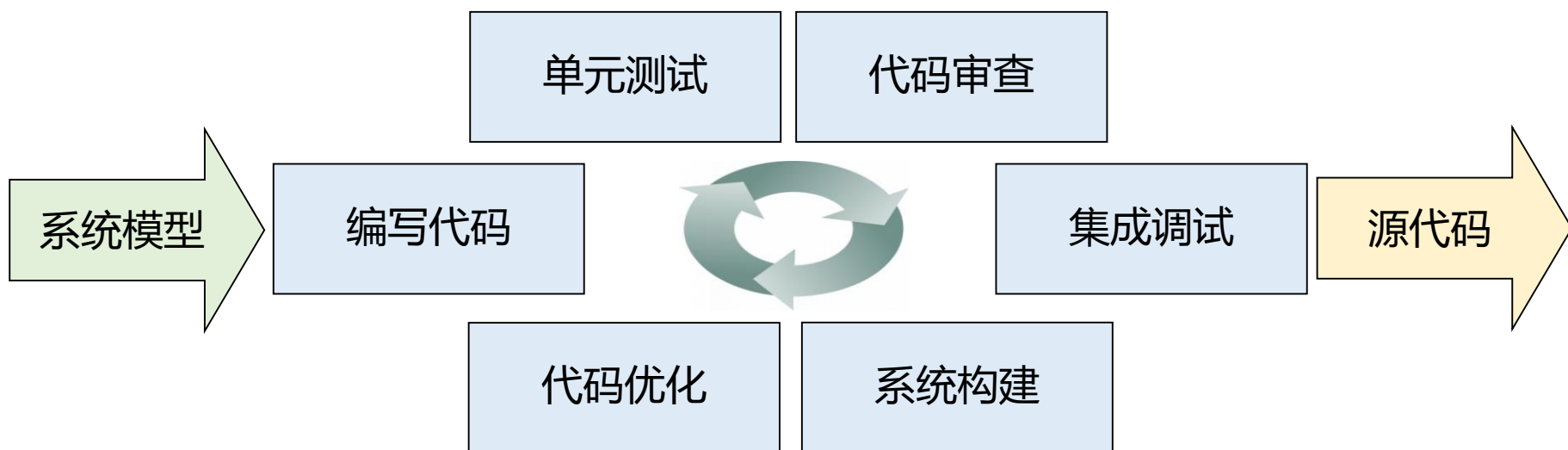
软件设计

软件设计：根据需求规格说明，确定软件体系结构，进一步设计每个系统部件的实现算法、数据结构及其接口等。



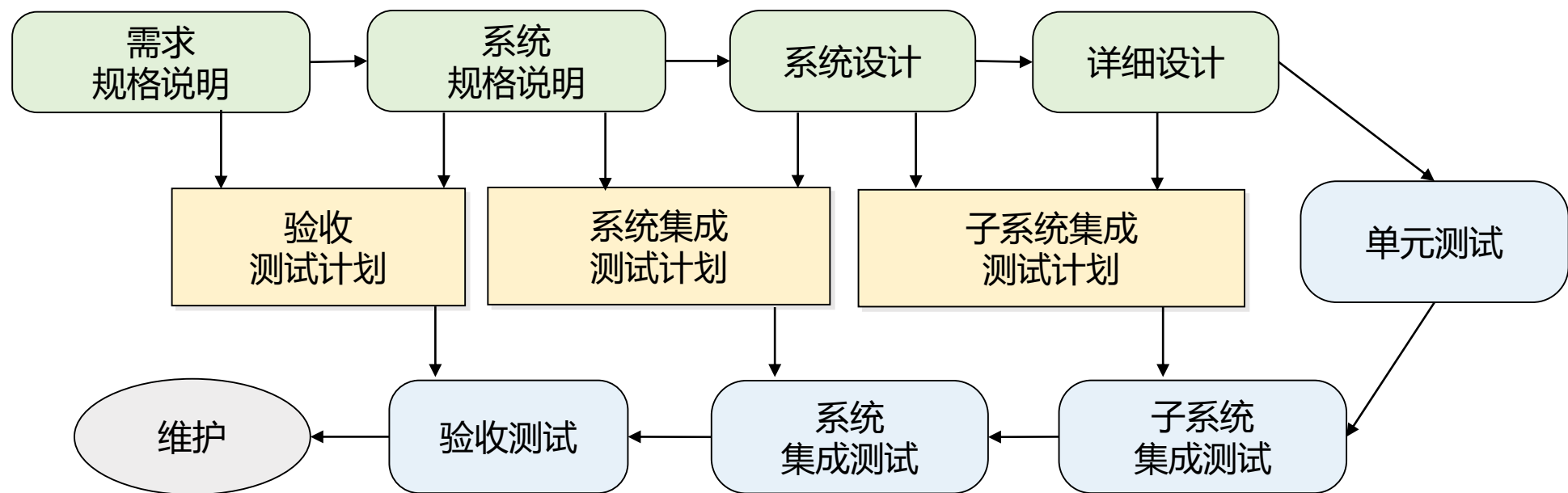
软件构造

软件构造：概括地说是将软件设计转换成程序代码，这是一个复杂而迭代的过程，要求根据设计模型进行程序设计以及正确而高效地编写和测试代码。



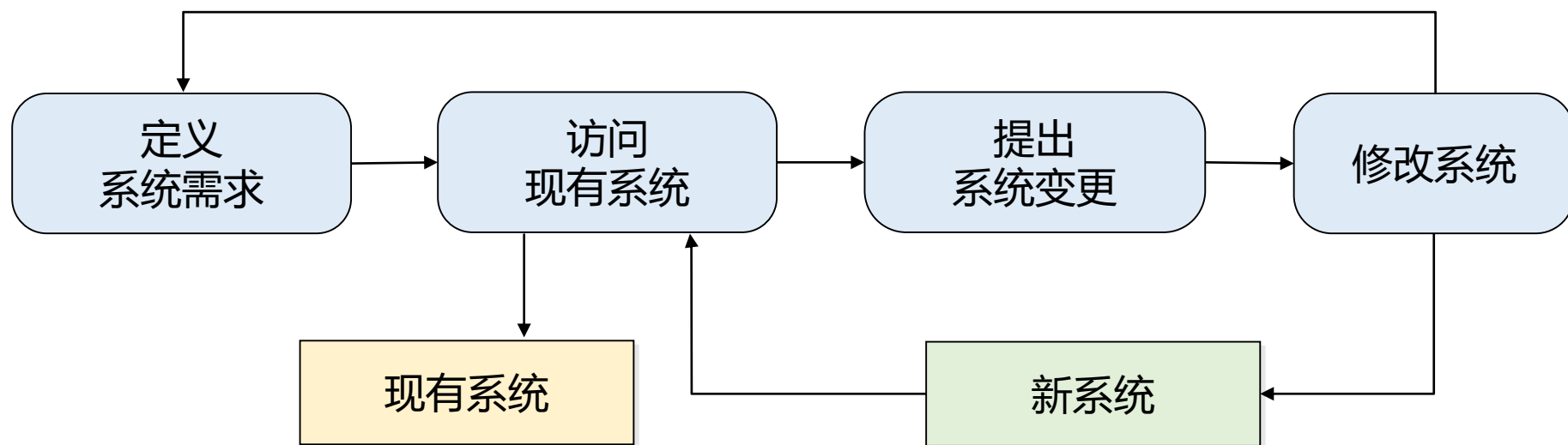
软件测试

软件测试：检查和验证所开发的系统是否符合客户期望，主要包括单元测试、子系统测试、集成测试和验收测试等活动。



软件维护

软件维护：系统投入使用后对其进行改进，以适应不断变化的需求。完全从头开发的系统很少，将软件系统的开发和维护看成是一个连续过程更有意义。



软件项目管理

软件项目管理是为了使软件项目能够按照预定的成本、进度、质量顺利完成，而对成本、人员、进度、质量和风险进行控制和管理的活动。

明确项目的目标，制定项目计划，确定项目所需的资源

计划

领导

指导实施项目计划，检查和评价项目的总体进展情况

组建项目小组，明确成员的分工和职责，配置各种资源

组织

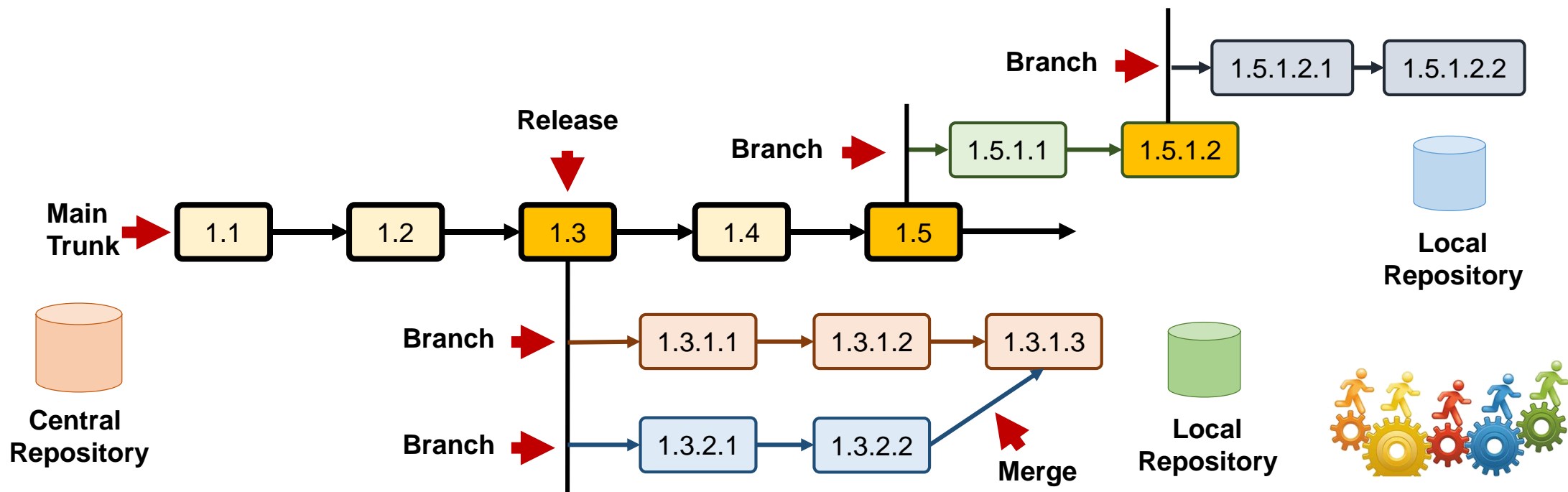
控制

控制项目的范围变更，监控项目进展，发现并纠正问题

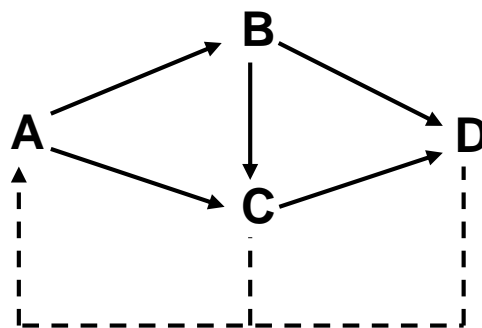
项目
管理

软件配置管理

软件配置管理是通过执行版本控制、变更控制的规程，并且使用合适的配置管理软件，来保证所有产品配置项的完整性和可跟踪性。



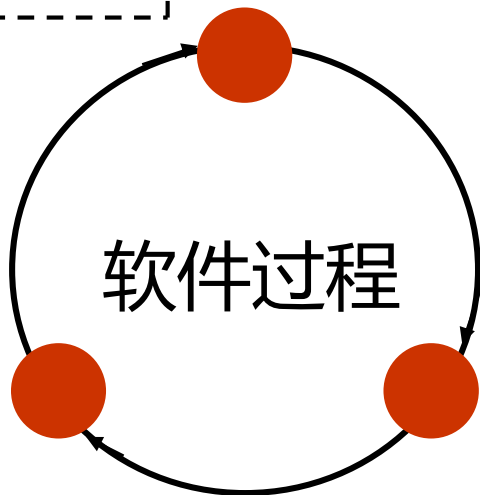
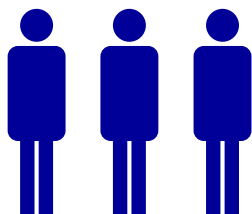
软件过程模型



定义任务之间关系和规程和方法

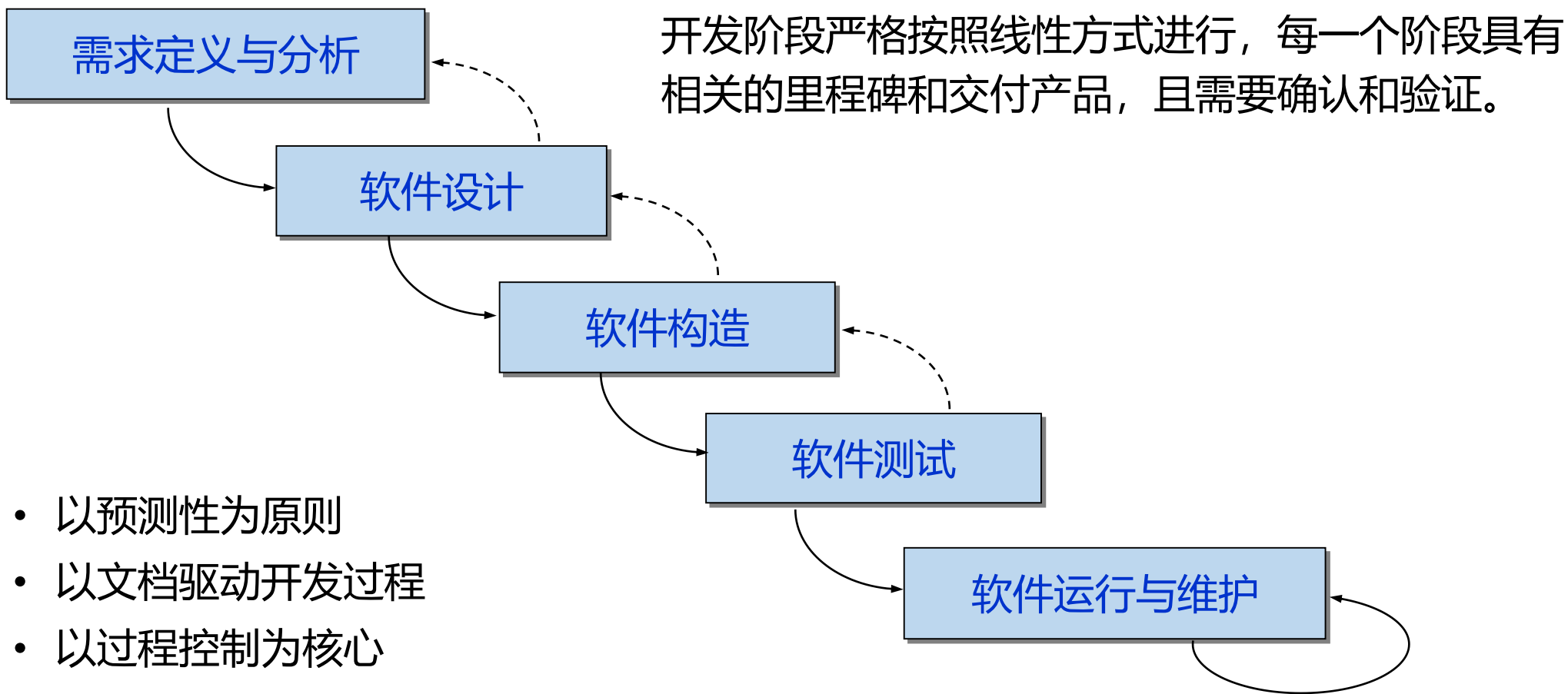
软件过程模型是对软件过程的抽象描述

有技能且受过
训练的人员



工具和设备

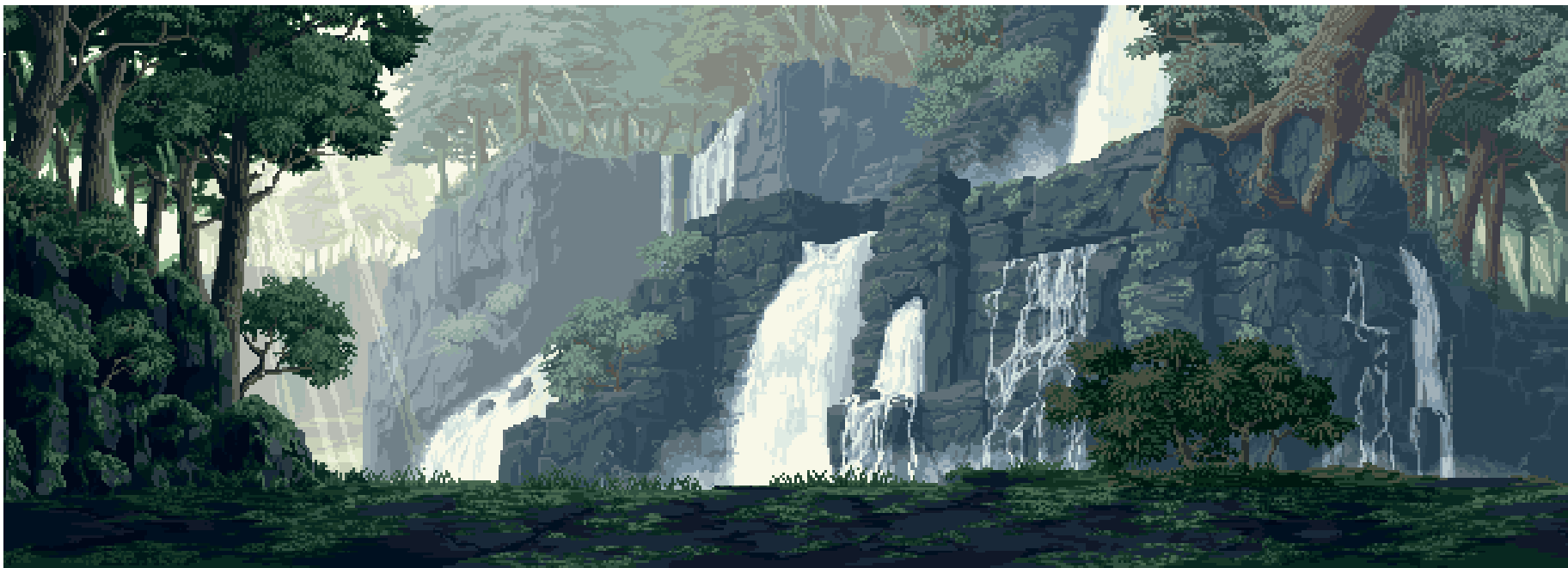
瀑布模型



瀑布模型

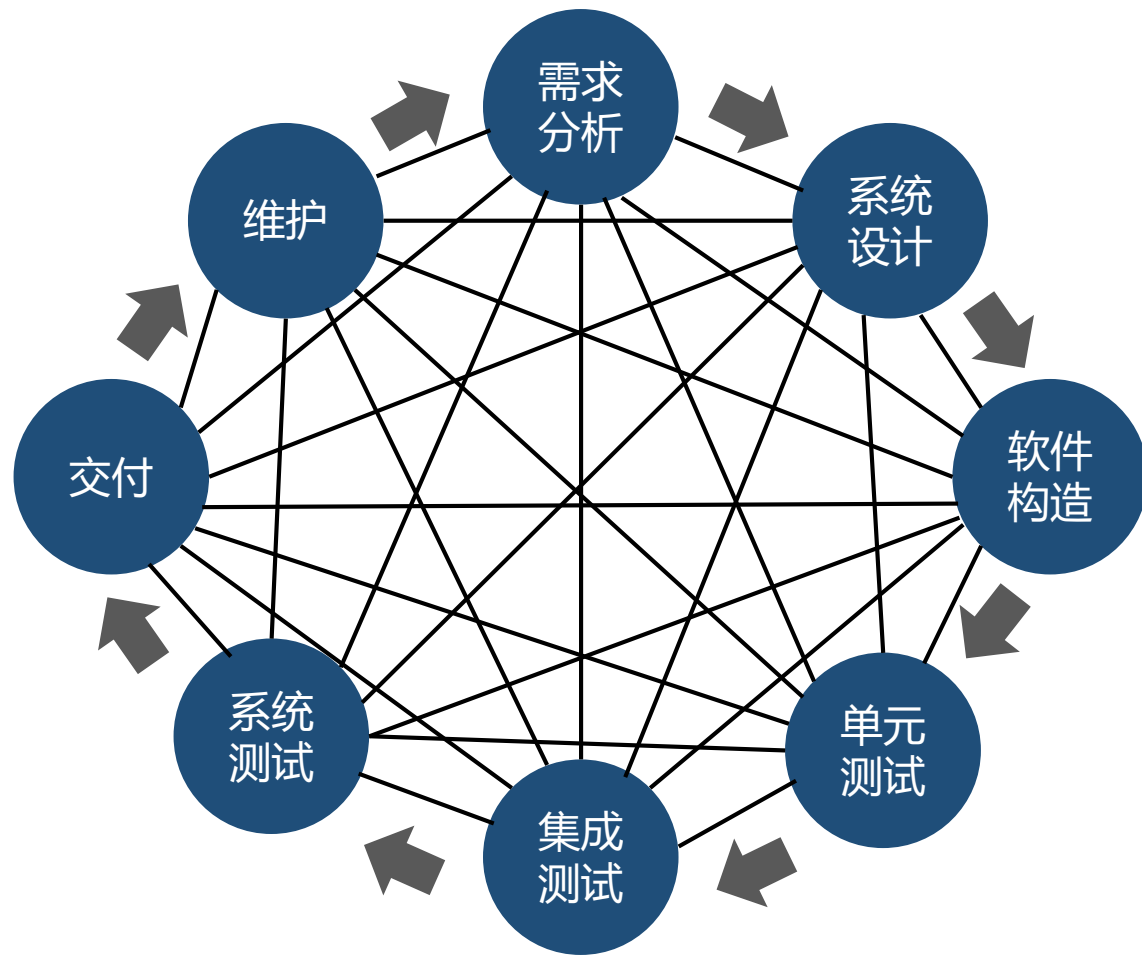
问题

- 瀑布模型是否反映了实际的软件开发过程？
- 软件开发作为一个问题求解过程，应当具备什么特点？



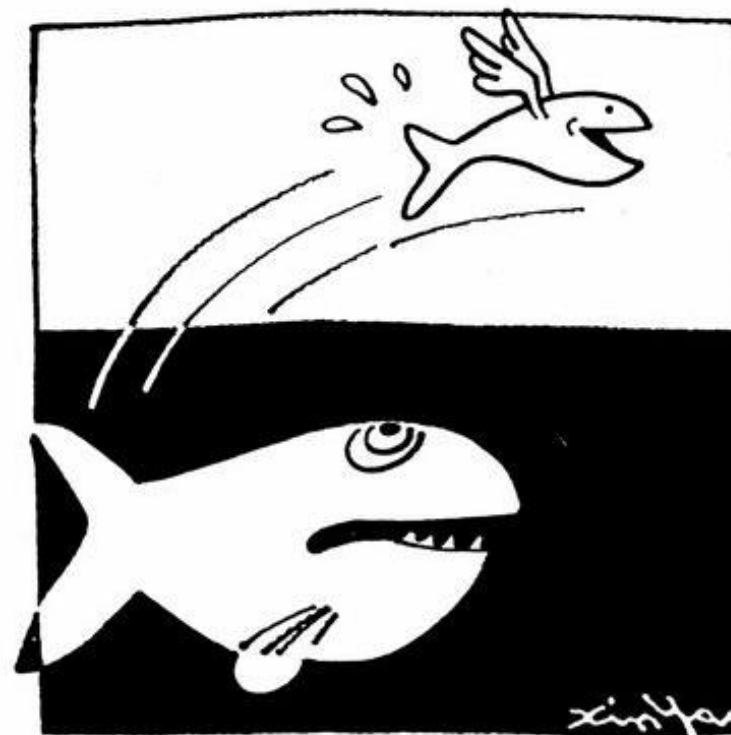
软件开发的迭代性

软件开发具有迭代性，需要不断地反复尝试，通过比较和选择不同的设计，最终确定令人满意的问题解决方案。



互联网产品的开发特点

- 快鱼吃慢鱼
- 版本发布成本很低
- 追求创新
- 需要快速响应用户的变化
- 需求不确定性高
- 关注用户行为



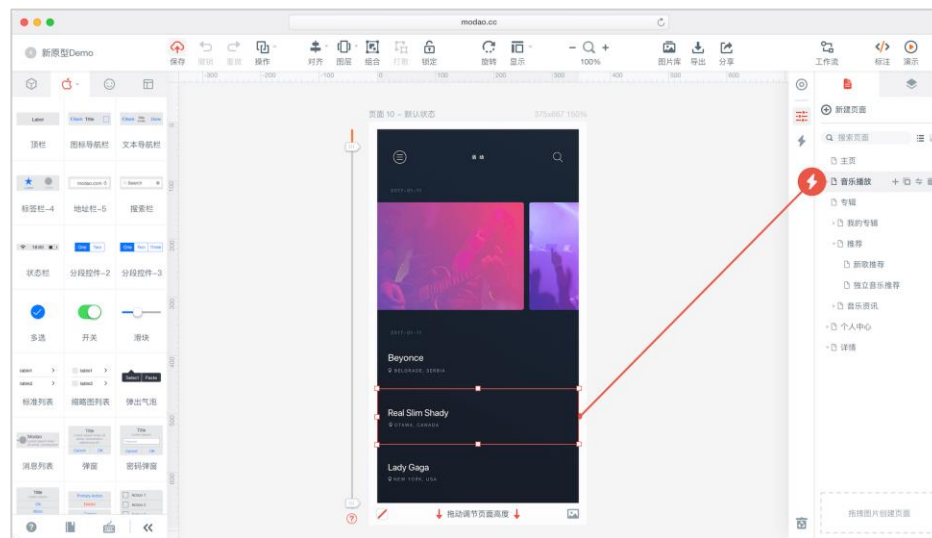
如今不是大鱼吃小鱼,而是快鱼吃慢鱼。

原型化模型

原型法是在实际开发产品之前，先实现该产品的部分原型，并据此征求对需求的反馈意见，它比开发人员常用的技术术语更易于理解。

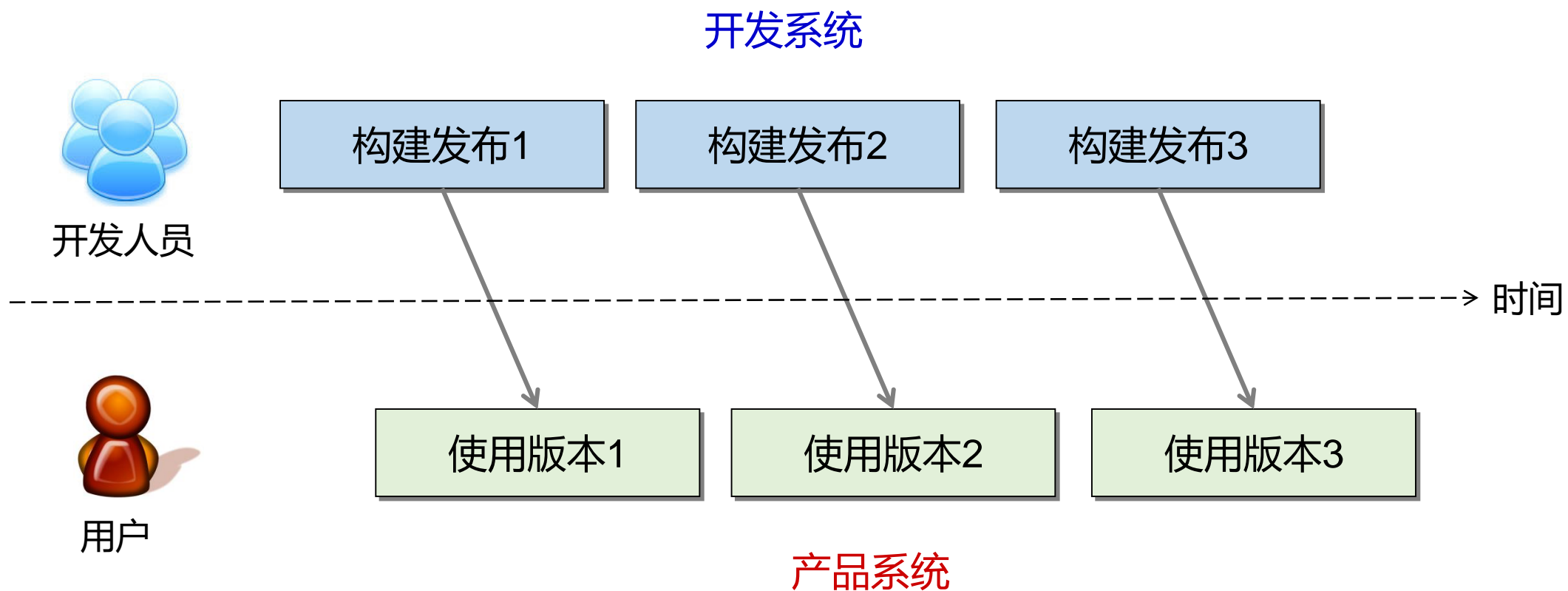


纸上原型设计示例



<https://modao.cc/>

迭代式开发

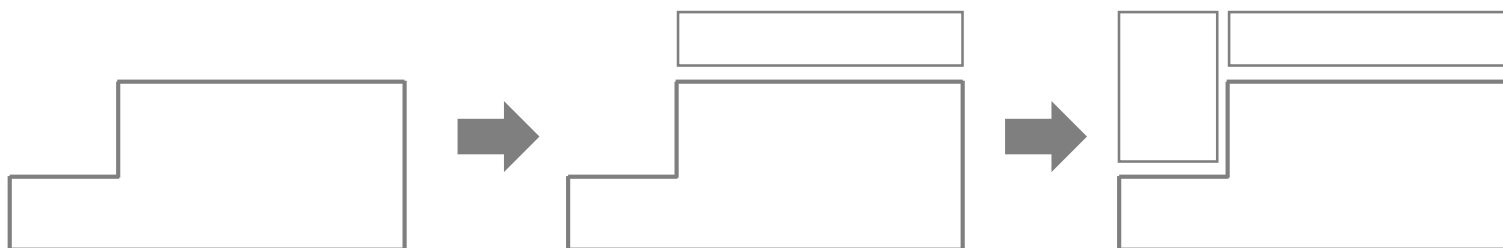


迭代式开发

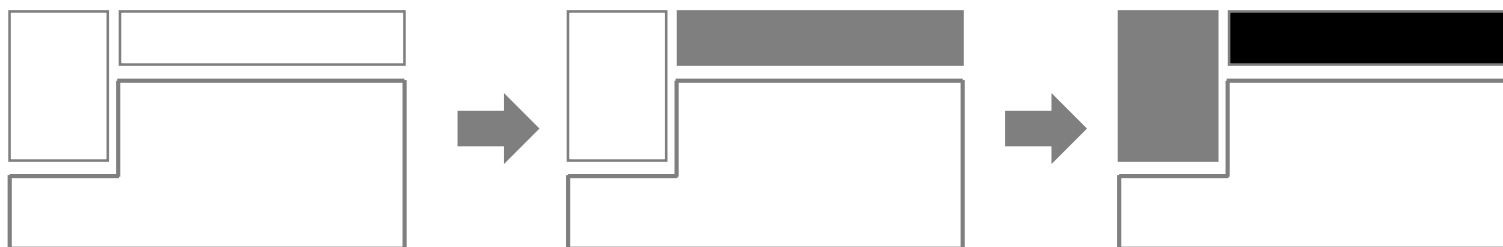
增量模型：在每一个新的发布中逐步增加功能直到构造全部功能。

迭代模型：一开始提交一个完整系统，在后续发布中补充完善各子系统功能。

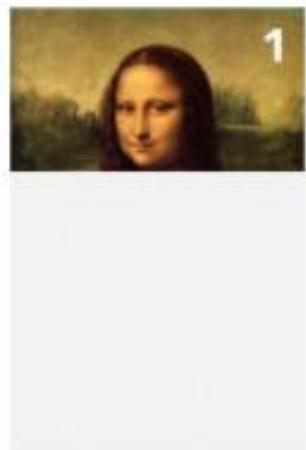
增量模型



迭代模型



迭代式开发



增量模型

迭代模型



选择合适的过程模型



不同的过程



相同的生命周期



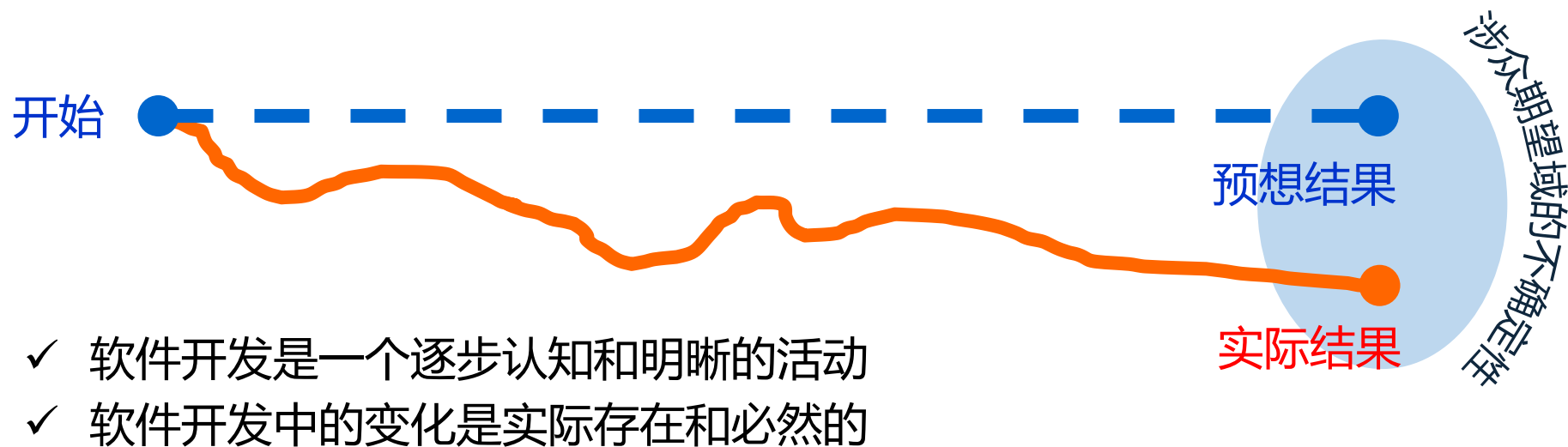


1	软件过程模型
2	敏捷开发方法
3	团队组织与管理

软件开发之道

软件开发是否可以实现一个完整、详尽的计划？软件项目能否预先考虑到所有的风险？

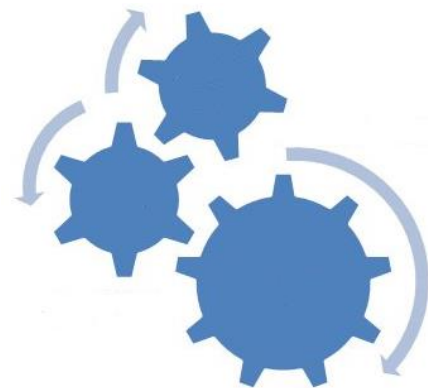
软件项目中难以预知所有的内容和风险！！！！



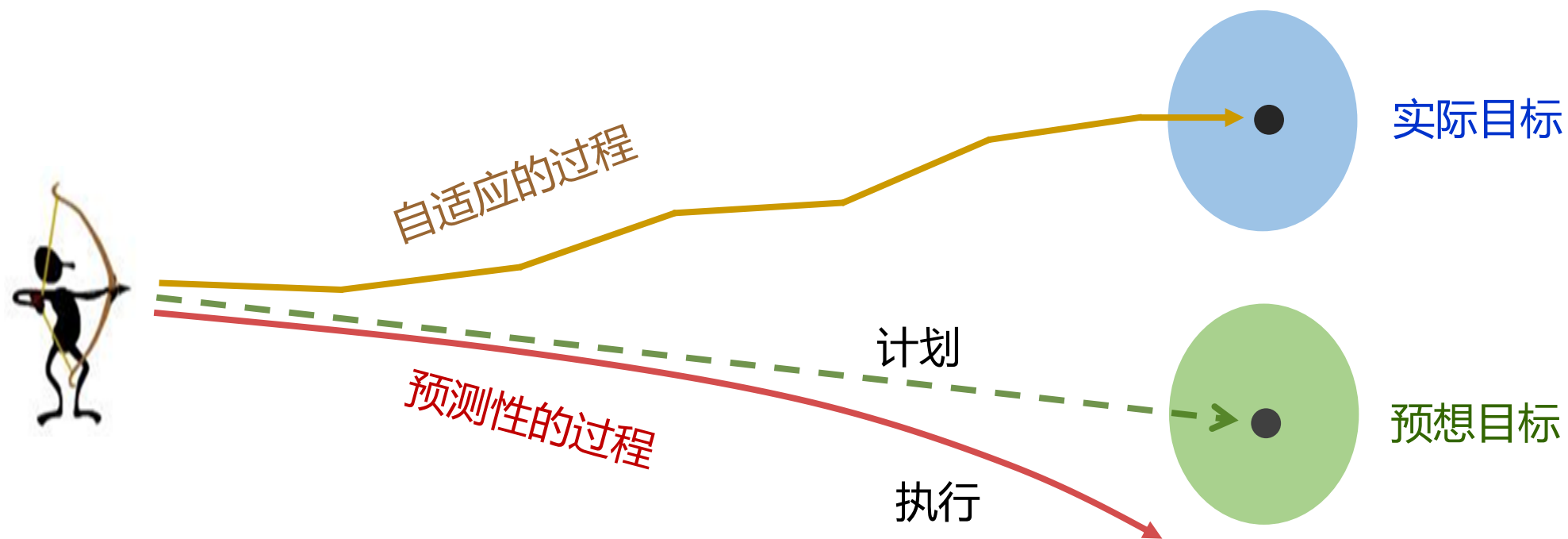
您是想获取一些更有价值的交付产品呢，
还是只想完成进度表!!

软件开发应更关注于交付的价值

- ✓ 高质量的交付物是最重要的
- ✓ 系统不是一次构建而成，而是迭代演进的
- ✓ 基于完整的场景构建计划，并按优先级执行



敏捷方法：适应而非预测



- 需求是不可预测的
- 软件开发应是一个自适应的跟踪过程

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

可以工作的软件

客户合作

响应变化

胜过

胜过

胜过

胜过

过程和工具

面面俱到的文档

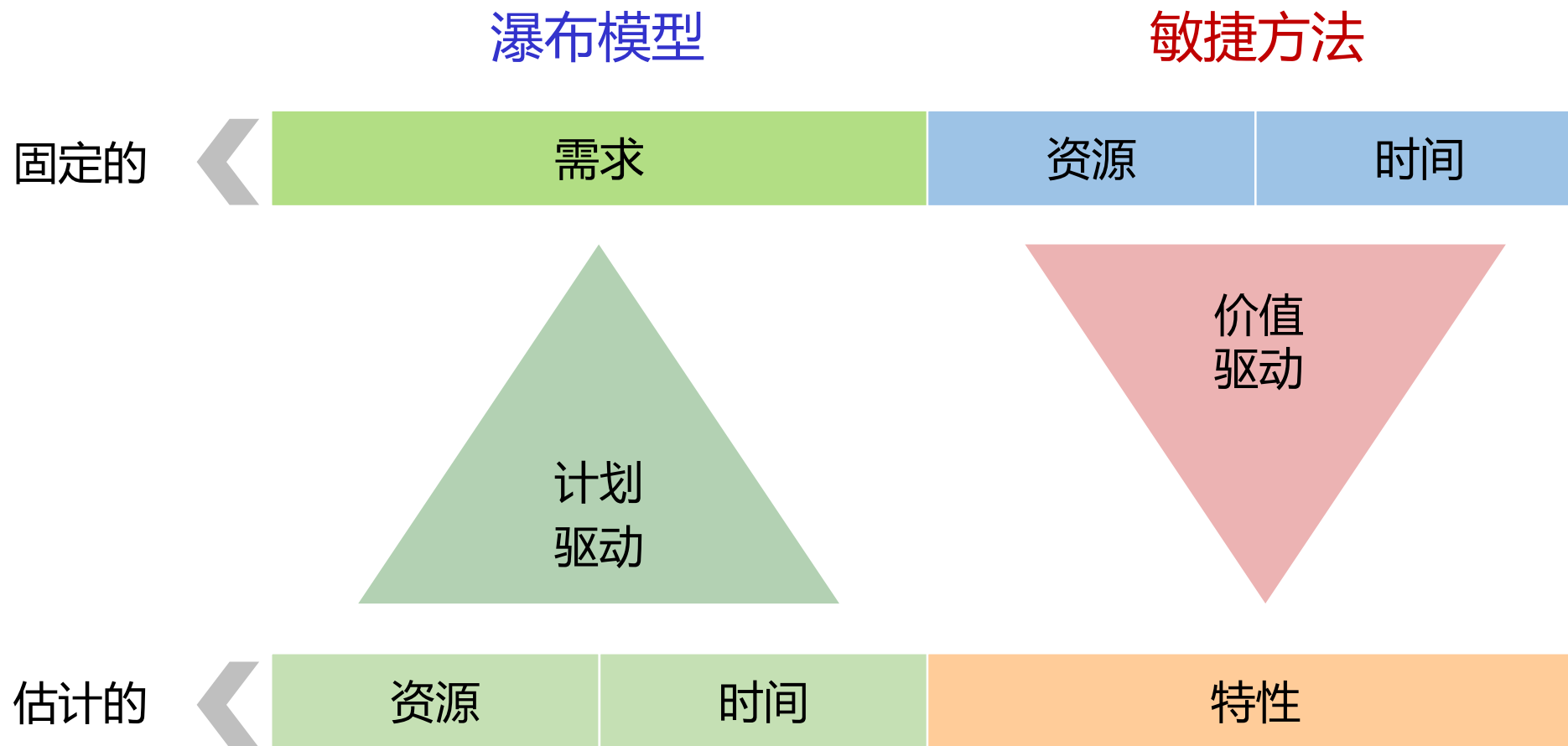
合同谈判

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。



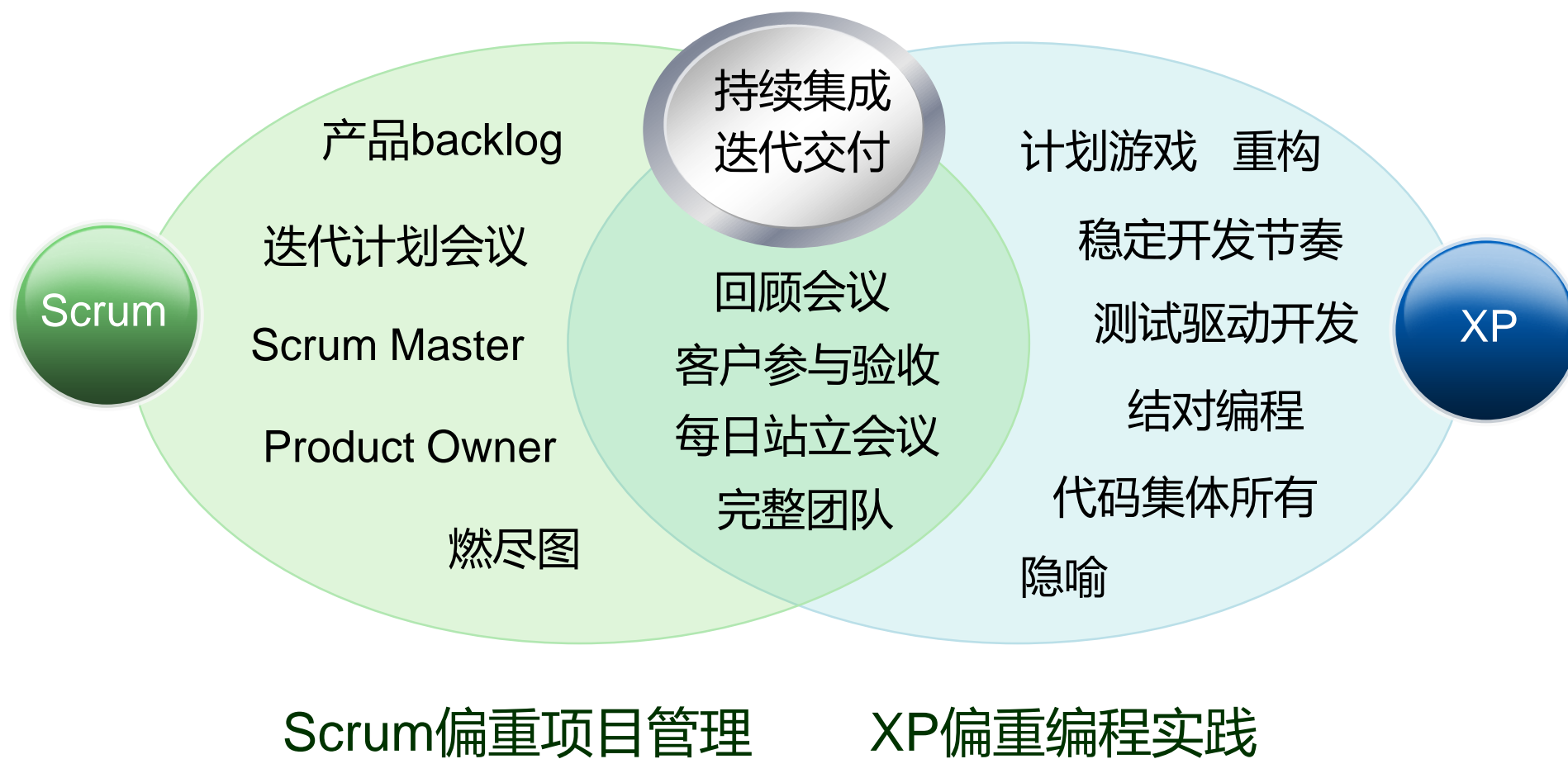
传统开发 vs. 敏捷开发



美国Standish集团的调查报告 (2011-2015)

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%	28%	4%
	Waterfall	44%	45%	11%

敏捷开发方法



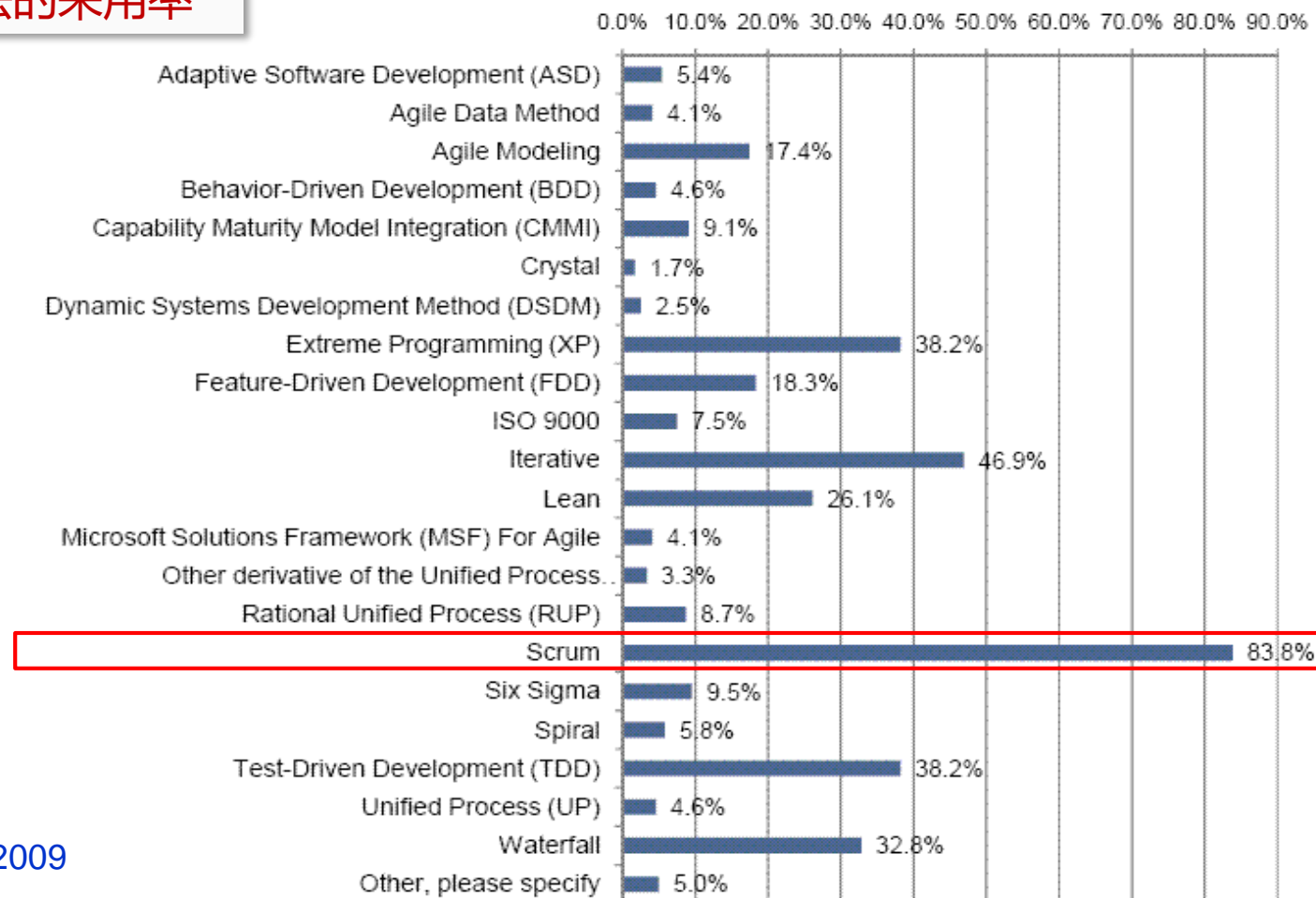
敏捷开发应用



- 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
- 2013年发布的新版PMBOK增加迭代及增量生命周期（即对应敏捷模型）

敏捷开发方法

不同开发方法的采用率



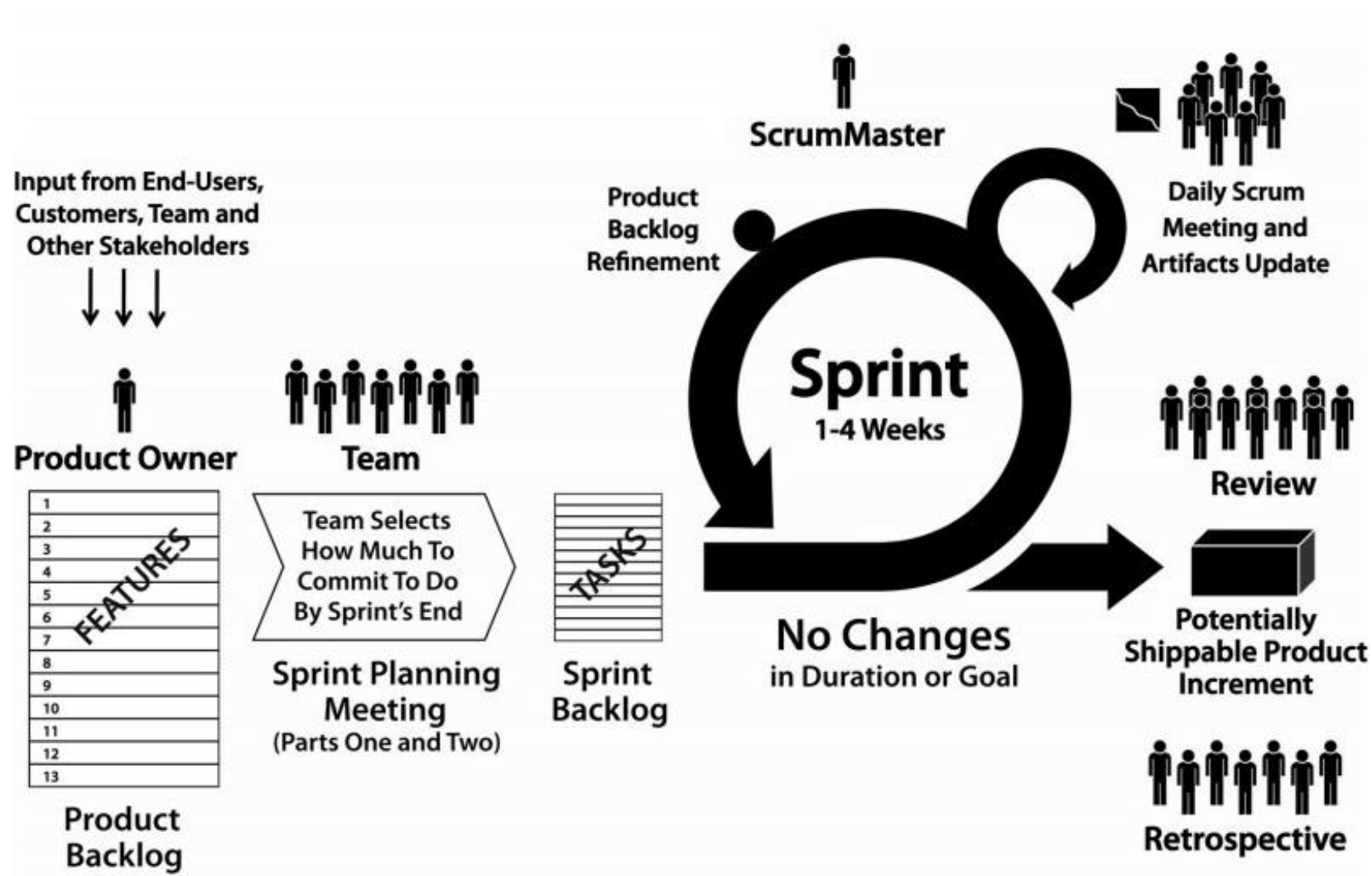
Forrest Research 2009

Scrum方法

Scrum方法是1995年由Ken Schwaber和Jeff Sutherland博士共同提出，已被众多软件企业广泛使用，如Yahoo, Microsoft, Google, Motorola, SAP, IBM 等。



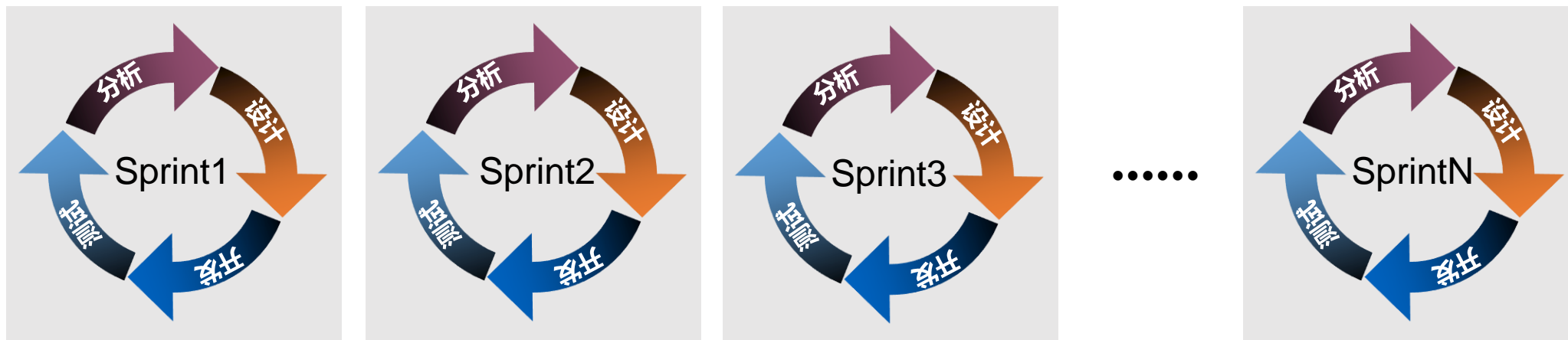
Scrum框架



Scrum框架



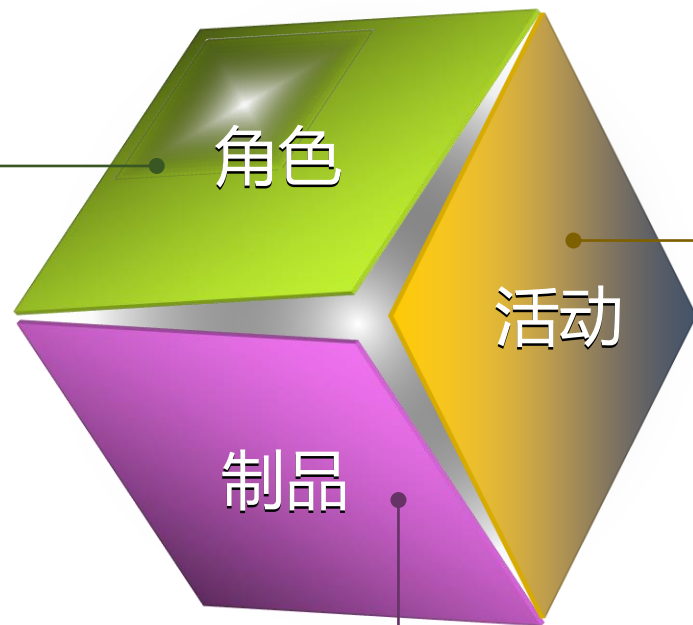
- 一个Sprint是一个1-4周的迭代，它是一个时间盒。
- Sprint的长度一旦确定，将保持不变。
- Sprint的产出是“完成”的、可用的、潜在可发布的产品增量。



需求在一个Sprint内是不允许变化的

Scrum框架

- 产品负责人
Product Owner
- Scrum主管
Scrum Manager
- 团队成员
Team



- 迭代计划会议
Sprint Planning
- 每日站立会议
Daily Scrum Meeting
- 迭代评审会议
Sprint Review
- 迭代回顾会议
Sprint Retrospective

- 产品订单 Product Backlog
- 迭代订单 Sprint Backlog
- 燃尽图 Burndown Charts

Scrum团队

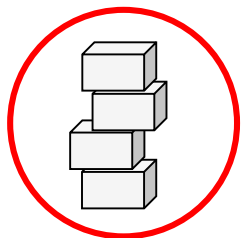
- 一般 6~8 人
- 拥有多技能的、跨职能协作
- 关注于向项目干系人交付价值
- 自指导、自组织、可持续的速度

全功能的整体团队



- 拥有共同目标，分担责任，彼此承诺致力于目标实现
- 拥有足够授权和资源，解决问题，找到自己的成功之路
- 有效的沟通和信息的透明

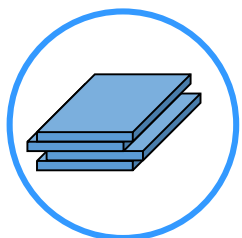
Scrum制品



Product Backlog

产品订单是从客户价值角度理解的产品功能列表。

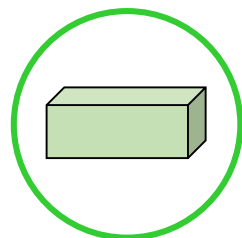
- 功能、缺陷、增强等都可以是产品订单项
- 整体上从客户价值进行优先级排序



Sprint Backlog

迭代订单是从开发技术角度理解的迭代开发任务。

- 简单环境：可直接把产品订单项分配到迭代中
- 复杂环境：可把一个产品订单项分为Web/后台.....软件/硬件.....程序/美工.....等开发任务



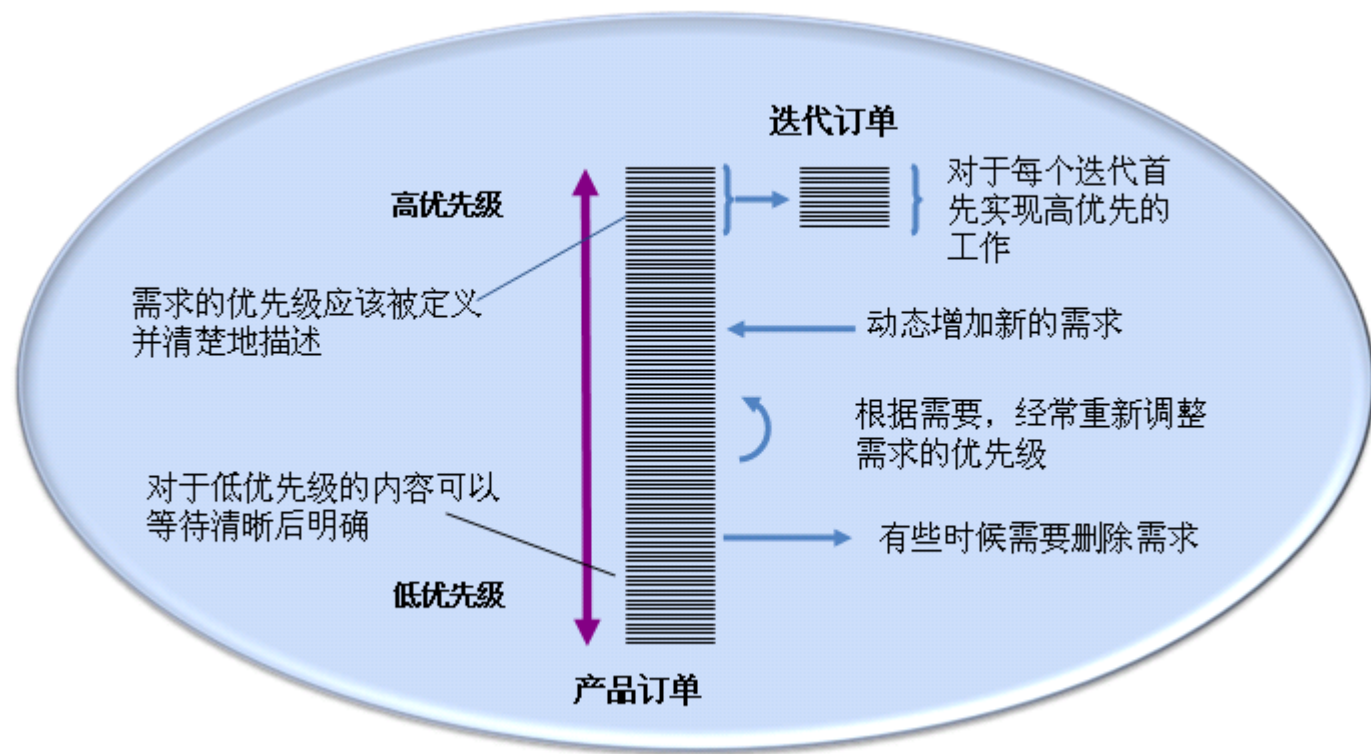
Working Software

可工作软件是可交付的软件产品。

- “可交付”应视不同情况提前设定和选定交付标准。
- 正式产品可能包括使用文档，在新产品开发初期可能只需要交付勉强看到效果的产品。

产品订单

在迭代计划时，产品负责人告诉开发团队需要完成产品订单中的哪些订单项，开发团队决定在下一次迭代中他们能够承诺完成多少订单项。在迭代的过程中，没有人能够变更迭代订单，这意味着在一次迭代中需求是被冻结的。



Scrum之产品规划

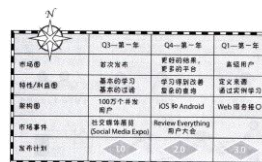
阶段1: 愿景

描述: 产品目标及其与公司战略的一致性
负责人: 产品负责人
频率: 至少每年一次



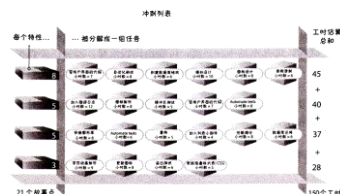
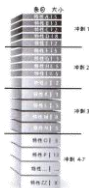
阶段2: 产品路线图

描述: 构成产品愿景的产品特性整体视图
负责人: 产品负责人
频率: 至少每半年一次



阶段3: 发布计划

描述: 特定产品功能的发布时间计划
负责人: 产品负责人
频率: 至少每季度一次



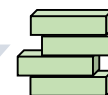
阶段4: 迭代计划 (冲刺计划)

描述: 确定具体的迭代目标和任务
负责人: 产品负责人和开发团队
频率: 每个迭代开始

阶段7: 迭代回顾

描述: 通过改善环境和流程来优化效率
负责人: Scrum团队
频率: 每个迭代结束时

发布产品



阶段6: 迭代评审

描述: 演示可工作产品
负责人: 产品负责人和开发团队
频率: 每个迭代结束时



阶段5: 每日例会

描述: 确定并协调当天任务的优先级
负责人: 开发团队
频率: 每天



定义产品愿景

举例：Smart-Review项目

某公司的核心业务是提供一个交流产品和服务评论的论坛，随着竞争对手的不断发布新特性，该公司需要提供一种新的创新服务来超越竞争对手。

在客户调研过程中，用户反映在公司现有的网站上需要花很多时间来分辨真实评论和可疑评论，同时一个特定产品（如 DVD 播放器）或服务（如大街上的中国餐馆）评论太多，人们很难看完所有评论之后得到一个准确的整体评价。

公司决定启动 Smart-Review 项目，通过一个训练搜索代理，以一种全新的方式来识别、过滤和显示在线评论。

定义产品愿景

产品愿景描述

产品: _____ (产品名称)
为谁: _____ (目标客户)
需求: _____ (客户需要)
提供: _____ (产品服务)
价值: _____ (带来的价值)

你如何定义 Smart-Review 的产品愿景



定义产品愿景

产品愿景描述

产品: _____ (产品名称)
为谁: _____ (目标客户)
需求: _____ (客户需要)
提供: _____ (产品服务)
价值: _____ (带来的价值)

Smart-Review产品将为所有在线用户提供个性化的智能评论服务，可以搜索互联网并找出客观的、无偏见的产品或服务评论信息。

建立产品路线图

- 识别产品需求，整理产品特性
- 大致估算产品特性的开发工作量，并对产品特性进行优先级排序
- 设想一下产品路线图的大致时间框架

Smart-Review 产品特性列表:

- 普通用户可以要求**Smart-Review**过滤指定类型的评论
- **Smart-Review**提供类似谷歌的简单界面进行评论搜索
- 普通用户可以使用**Smart-Review**来监测产品或服务的相关新评论，系统为其进行自动过滤和发送报告。
- **Smart-Review**可以向高级用户提供评论搜索的来源。
- 供货商可以通过**Smart-Review**查看网站上所有产品的评论汇总。

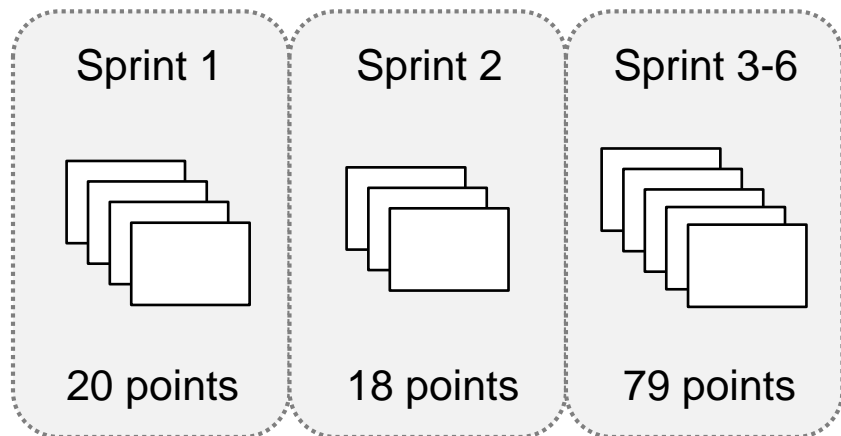
建立产品路线图

- 识别产品需求，整理产品特性
- 大致估算产品特性的开发工作量，并对产品特性进行优先级排序
- 设想一下产品路线图的大致时间框架

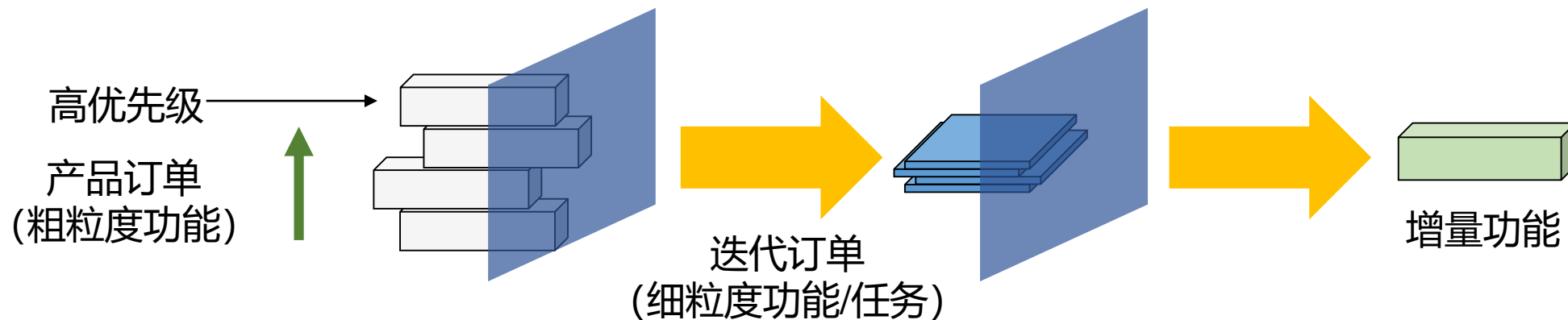
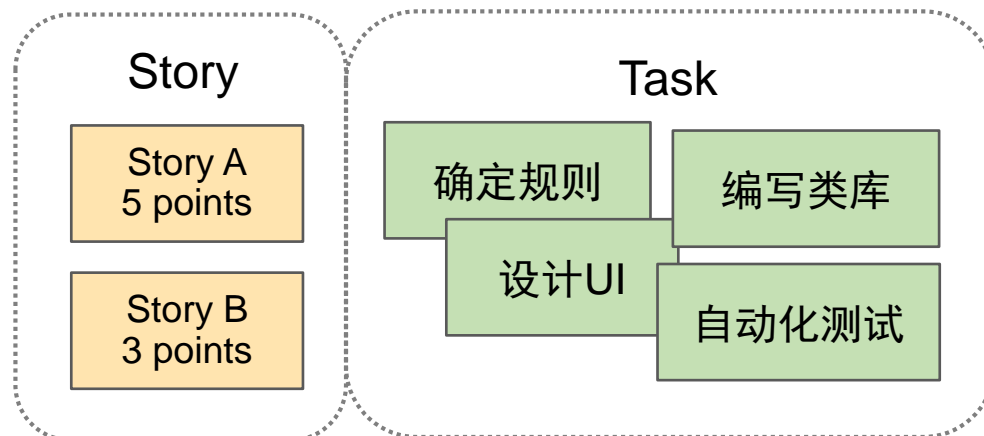
市场策略	首次发布	更好的结果，更多的平台	高级用户
产品特性	基本的学习 基本的过滤	学习得到改善 复杂的查询	定义来源 通过实例学习
	100万并发用户	iOS 和 Android	Web服务接口
市场事件	社交媒体展览	用户大会	—
发布计划	V1.0	V2.0	V3.0

Scrum之迭代计划

发布规划



迭代计划



创建发布计划

- 将产品构想中的概要产品特性细化成用户故事
- 定义发布版本的最小可发布特性
- 在细化和估算的基础上，将一系列订单项映射到每一个迭代中

- 用户可以训练 Smart-Review 过滤指定类型的评论
- Smart-Review 提供类似谷歌的简单界面进行评论搜索



- 作为普通用户，我想要 Smart-Review 忽略含有某些偏见字眼的评论，这样我看到的评论就不会含有那些字眼。
- 作为普通用户，我希望选择一类产品或服务，这样可以帮助 Smart-Review 只关注相关的评论。

创建发布计划

用户故事地图是敏捷需求规划中一种组织和管理用户故事的方法。

用户故事是从用户角度对功能的简要描述。

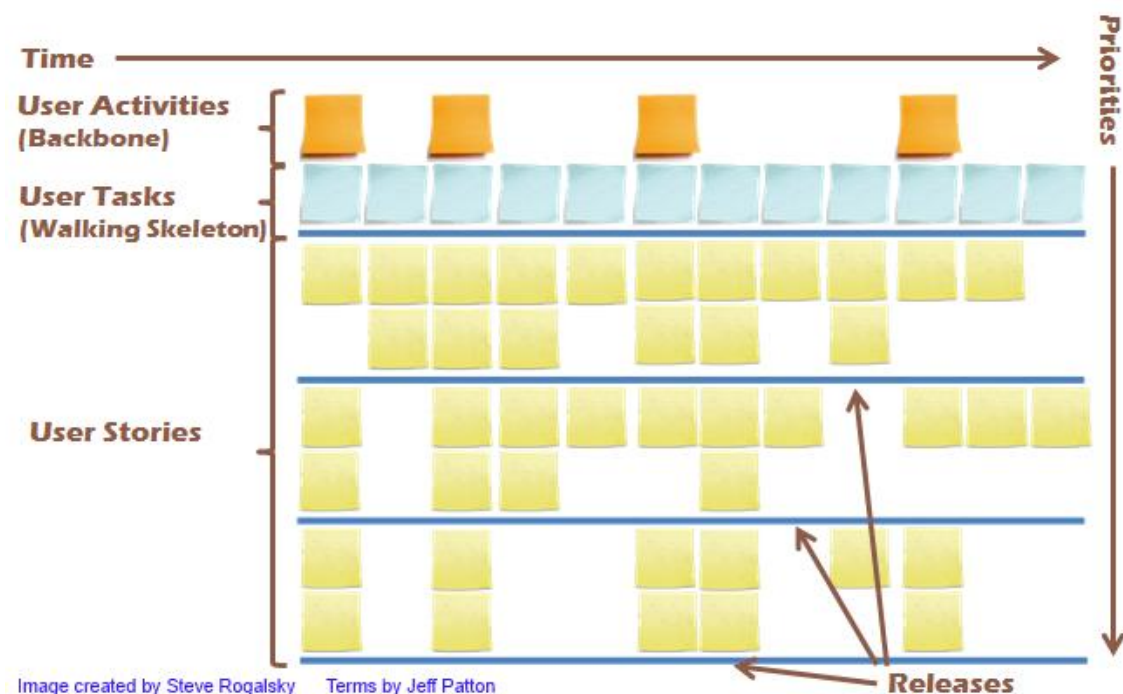
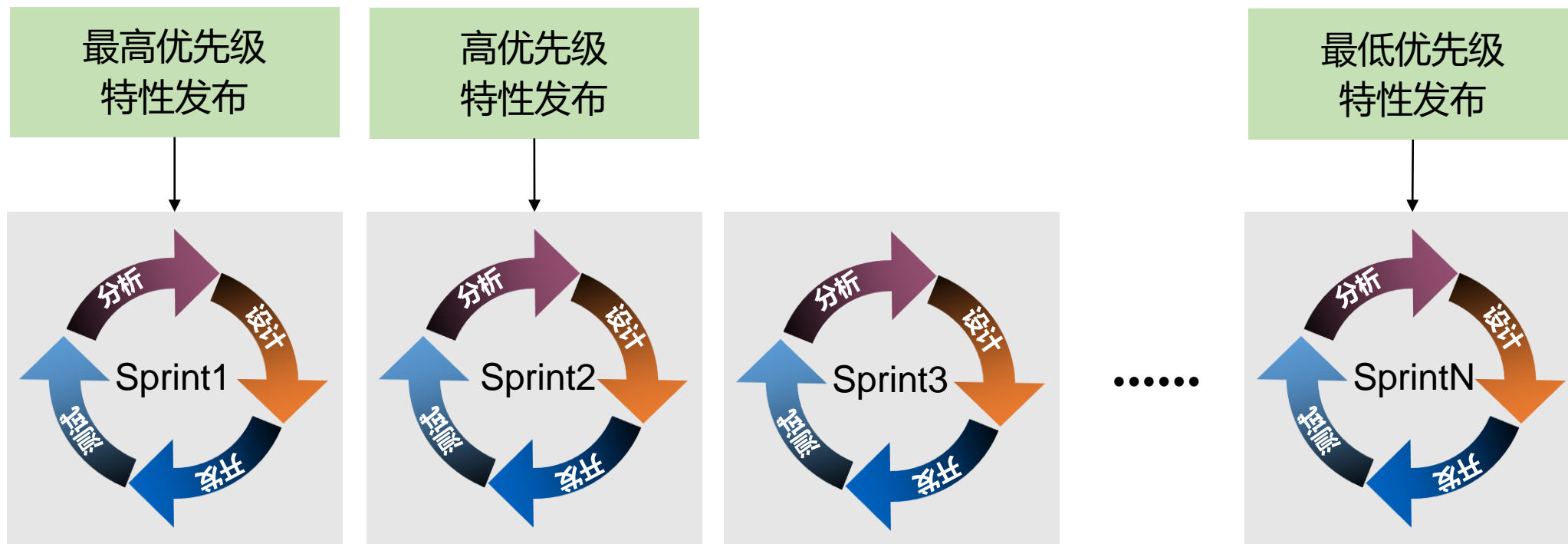


Image created by Steve Rogalsky <http://winnipegagilist.blogspot.com>
Terms by Jeff Patton <http://agileproductdesign.com>

- 每个用户故事地图代表一个完整的业务流程
- User Tasks: 从最重要的用户开始，使用头脑风暴的方法，描述该用户在一天中使用产品的情景，把每一件事都写到一张便利贴中，按照时间顺序从左到右排列
- Backbone: 对所有的便签进行分组，将类似的任务分为一组，并对每个组进行命名
- 在每个用户任务下面添加更加细节的用户故事，借助如Persona和Scenario等方式完成
- 根据用户故事的优先级，划定产品发布计划

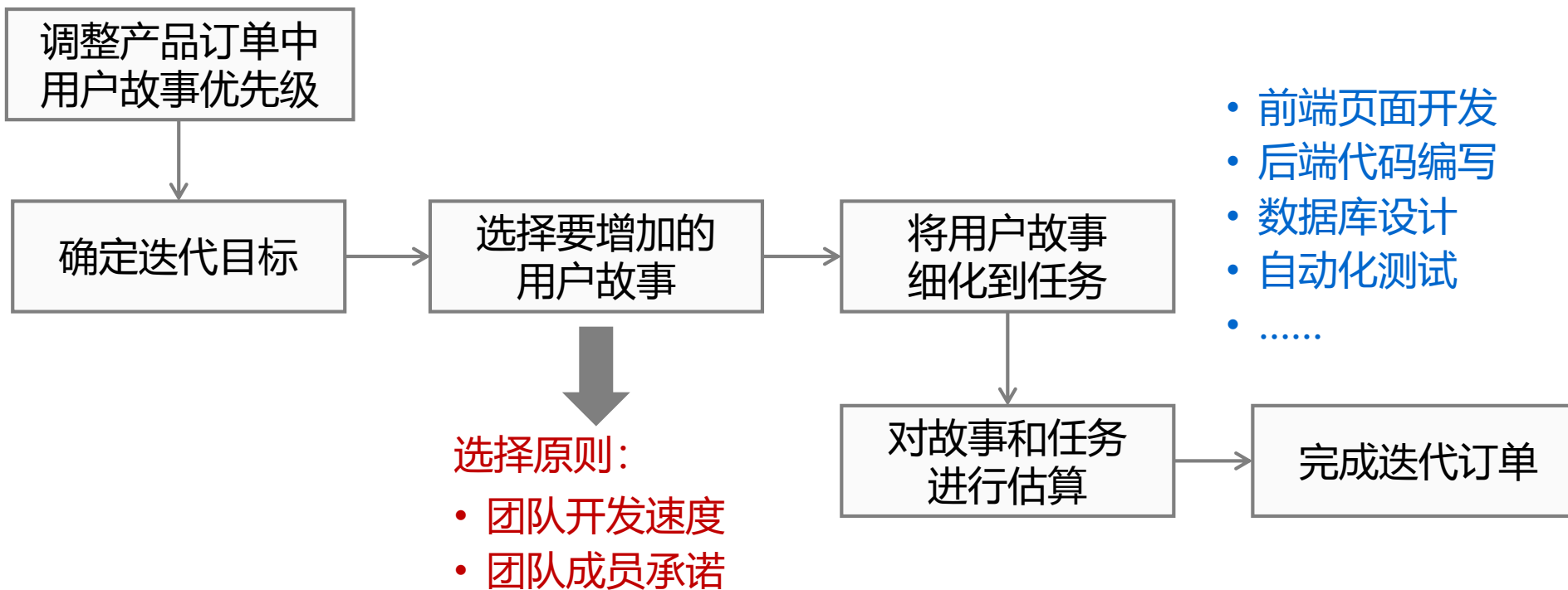
创建发布计划



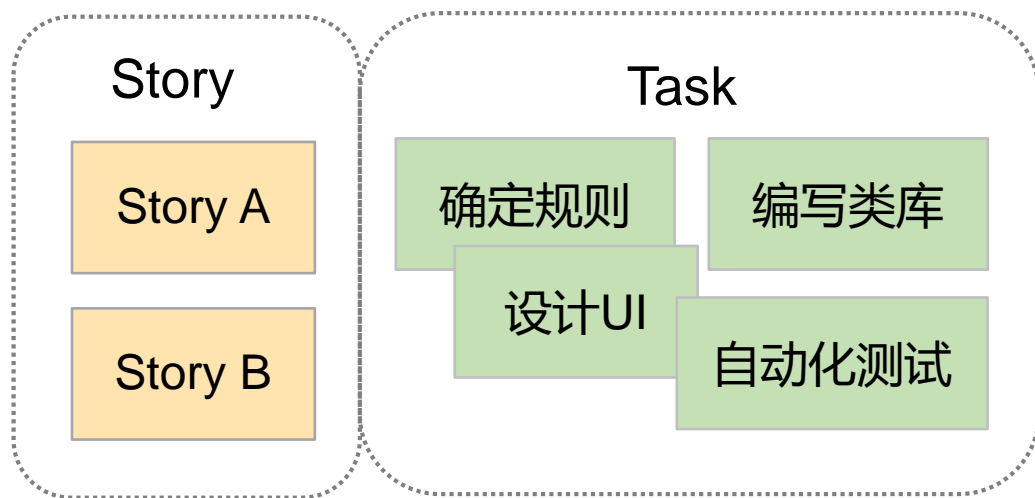
根据团队的开发速率和特性优先级，决定每个迭代应实现的特性

制定迭代计划

迭代计划将产生Scrum中提到的迭代订单，一个迭代的时间是固定的。



用户故事与任务



- 如果故事中的某个任务很难估算，可以将它独立出来。
- 如果某些任务可以分给多个人单独完成，那么把这个任务分割成更小的任务。
- 如果有必要让客户了解故事的某一个部分的完成情况，那么将它们分离出来，成为单独的任务。

用户故事与任务

- **用户故事**是从用户的角度描述用户渴望得到的功能和实现的价值，面向的是具体客户的需求，而不关心具体实现。
- 由用户故事分解而来的**具体任务**，关注于该如何实现该用户故事而必须完成的工作，关注的是具体的技术细节和实现方法。



敏捷软件开发中，团队成员自己主动认领的是用户故事还是被分解成的任务？

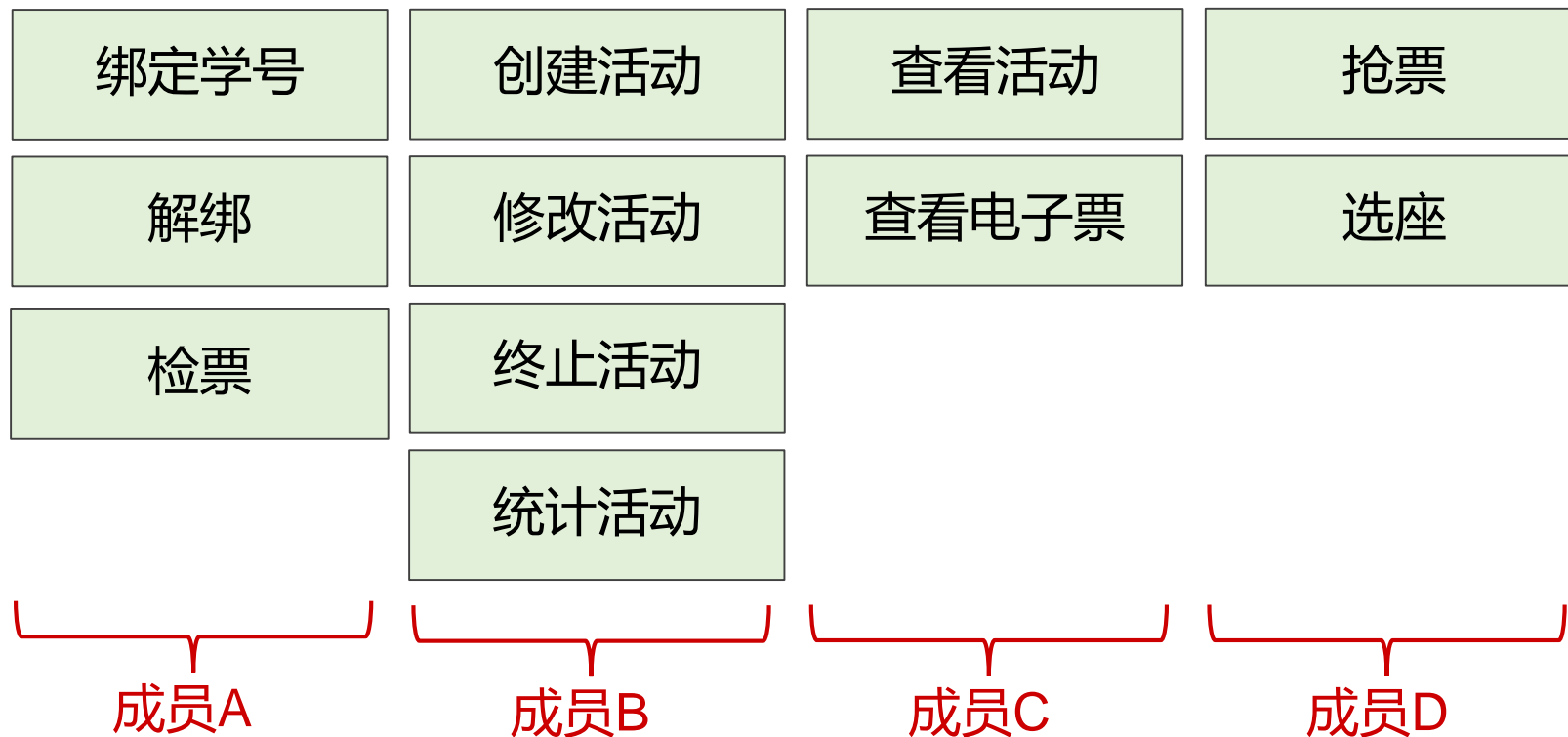
用户故事与任务

- **用户故事**是从用户的角度描述用户渴望得到的功能和实现的价值，面向的是具体客户的需求，而不关心具体实现。
- 由用户故事分解而来的**具体任务**，关注于该如何实现该用户故事而必须完成的工作，关注的是具体的技术细节和实现方法。

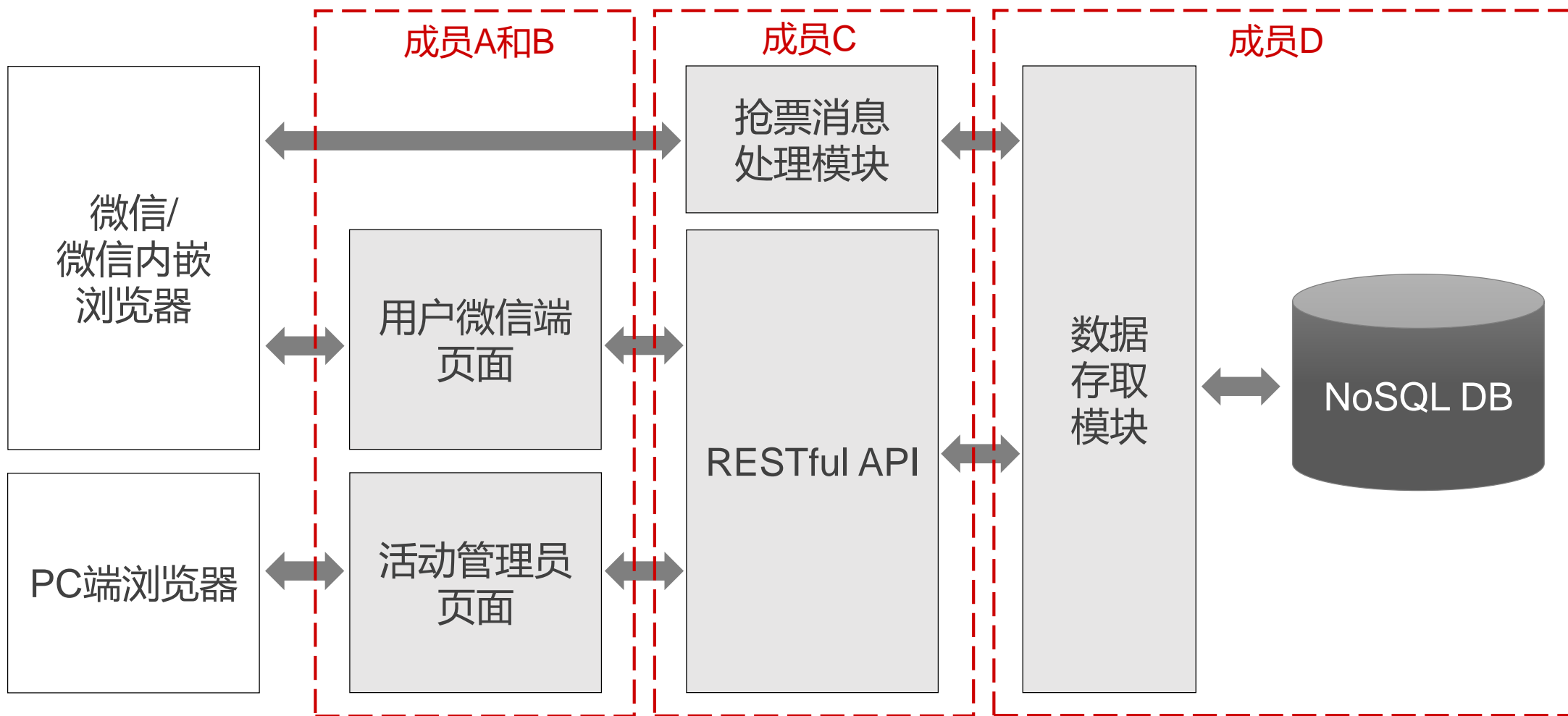


团队成员更倾向于专注于具体的工作任务和自己专长的领域，团队成员主动认领自己擅长的任务并高效、高质量地完成，这是符合敏捷开发原则的最佳做法。

举例：微信抢票应用



举例：微信抢票应用





软件项目估算的挑战是项目的复杂性和不确定性

- 软件规模越大，复杂性越高，不确定性就越大
- 需求的不确定性会对项目估算产生很大影响
- 没有可靠的历史数据使项目估算缺少参照物

没有很好地制定计划是一个项目常见的严重错误之一。

很多技术人员宁愿从事技术工作，也不愿意花费时间制定计划。很多技术管理者没有受过充分的技术管理方面的培训，对于计划能够改善项目成果缺乏信心，既然这两部分人都不想制定计划，因此就不经常制定计划。

敏捷估算

项目管理者不应该被估算所困扰，勇于面对软件项目估算的挑战，做出一个相对的有价值的估算。估算并不是越准越好，找到合适自己的度最为关键。

故事点

- 故事点是用于表达用户故事、功能或其他工作的总体规模的度量单位，它是一个相对度量单位。
- 使用时，可以给每个故事分配一个点值；点值本身并不重要，重要的是点值的相对大小。

理想日

- 理想日是用于表达用户故事、功能或其他工作的总体规模的另外一种度量单位，它是一个绝对度量单位。
- 理想时间是某件事在剔除所有外围活动以后所需的时间；一般为一天有效工作时间的 60 - 80% 比较合理，但绝不会是全部。

敏捷估算：理想日



估算方法：团队查看每个故事，针对前面介绍的复杂性要素讨论故事，然后估计要用多少理想日可以完成该故事。



- 这种方式是人们平时习惯使用的，容易理解和使用
- 人天生不擅长做绝对估计，很难达成共识
- 不同人的理想日估算是不同的，因每个人的能力和认识的不同而不同
- 每个人的理想日是不一样的，这种估算不能相加，由此产生的计划肯定是不准确的

敏捷估算：故事点

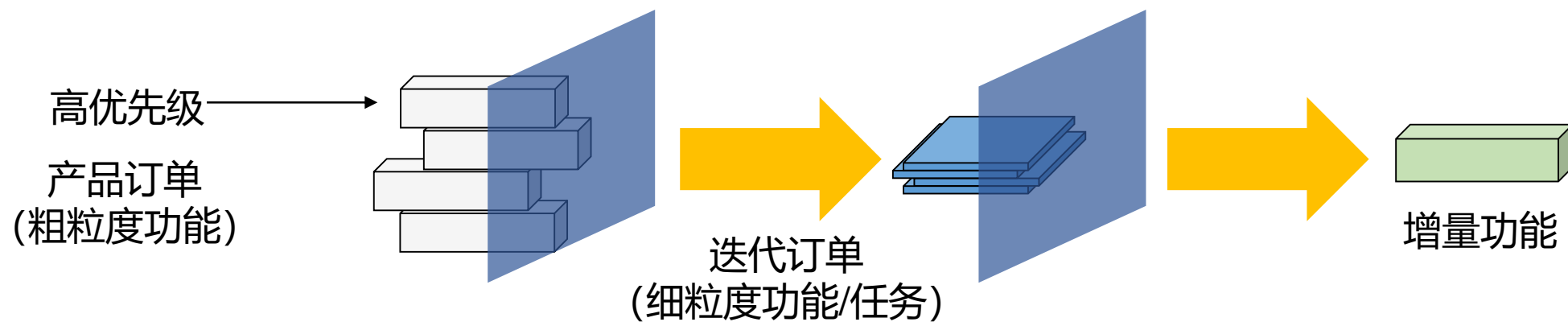
故事点的基本做法：

- 把一些常见的“标准任务”给出一个“标准点数”，形成比较基线；
- 估算时只要是同一类型任务，直接写故事点数而非天数。

使用难点：

- 故事点的项目或产品特征很明显，几乎无法进行跨团队比较；
- 如果没有历史数据，很难设定标准任务；
- 在标准任务没有那么多种类时，很难判断一个新任务到底像哪个标准任务；而太多的标准任务又令人迷惑。

敏捷估算：任务用时

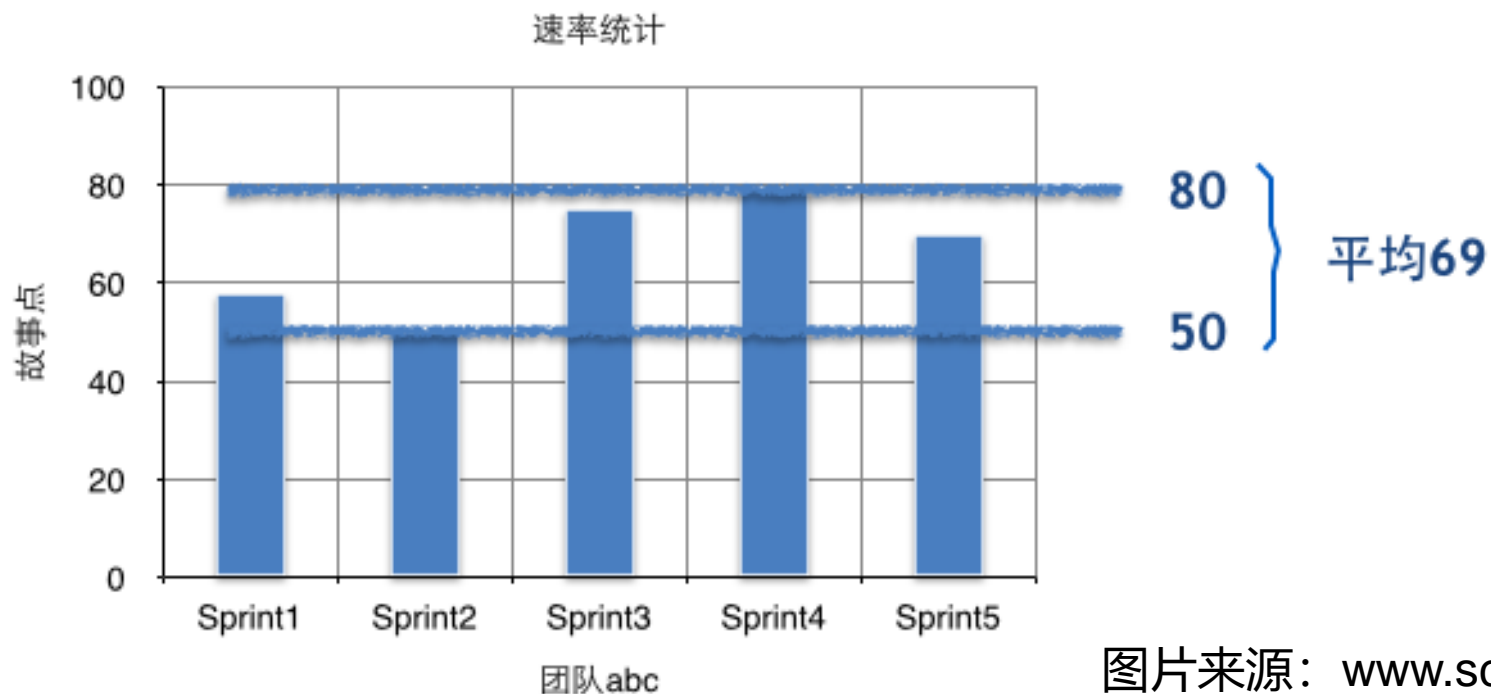


谁承诺任务，谁负责任务，谁估算任务

“做这个事情的那个人是最了解那个任务的人，也是最适合对之做出估计的人”

敏捷估算：团队速率

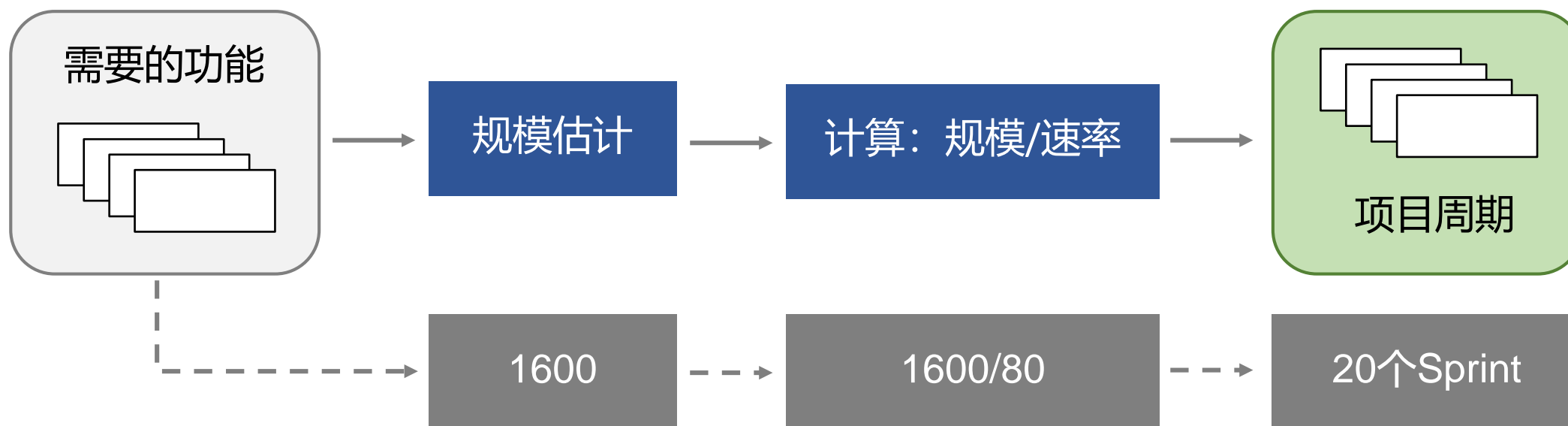
团队速率是一个团队在一个Sprint中实际完成的故事点数。对于新开始的项目或者新团队来说，通常没有初始速度，可以做1-2个Sprint进行测算，将其作为初始速度。



图片来源: www.scrumcn.com

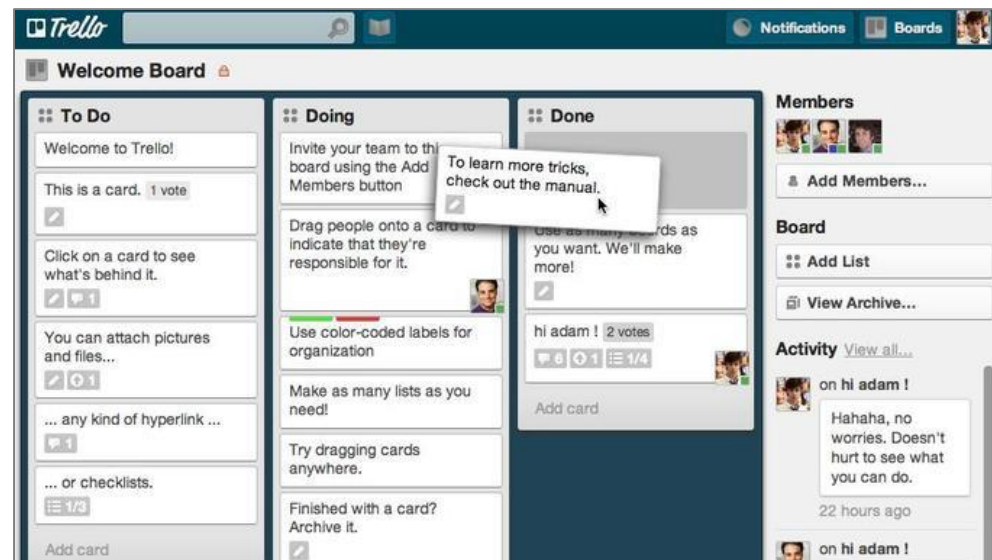
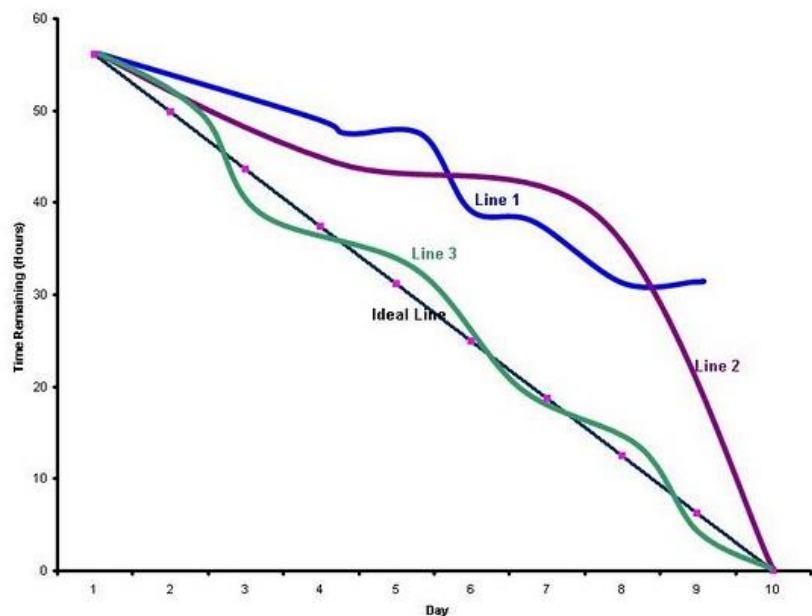
敏捷估算：项目周期

根据产品订单的故事点总数和每个Sprint团队的平均速度，我们可以推算大概需要多少个Sprint可以完成项目，从而得到了项目周期。



敏捷迭代的跟踪与管理

作为敏捷开发中任务完成情况的可视化展示，任务板和燃尽图直观而生动地展示出了项目团队在目前工作上的进展情况和预期进展。





1	软件过程模型
2	敏捷开发方法
3	团队组织与管理

团队的概念

团队是由若干人组成的一个群体，他们具有互补的技能，对一个共同目的、绩效目标及方法做出承诺并彼此负责。



Scrum团队

自组织团队是敏捷软件开发的基本观念，即团队被授权自己管理他们的工作过程和进度，并且团队决定如何完成工作。

- 团队决定谁做什么，即任务的分配
- 团队决定如何做，如何实现目标，即团队做技术决策
- 团队需要在确保目标的前提下制定团队内的行为准则
- 团队有义务保持过程的透明性
- 团队监督和管理他们的过程和进度

Scrum团队角色



用户



其他干系人



产品负责人

- 定义产品需求
- 确定产品发布计划
- 对产品收益负责
- 确定需求优先级
- 调整需求和优先级
- 验收迭代结果



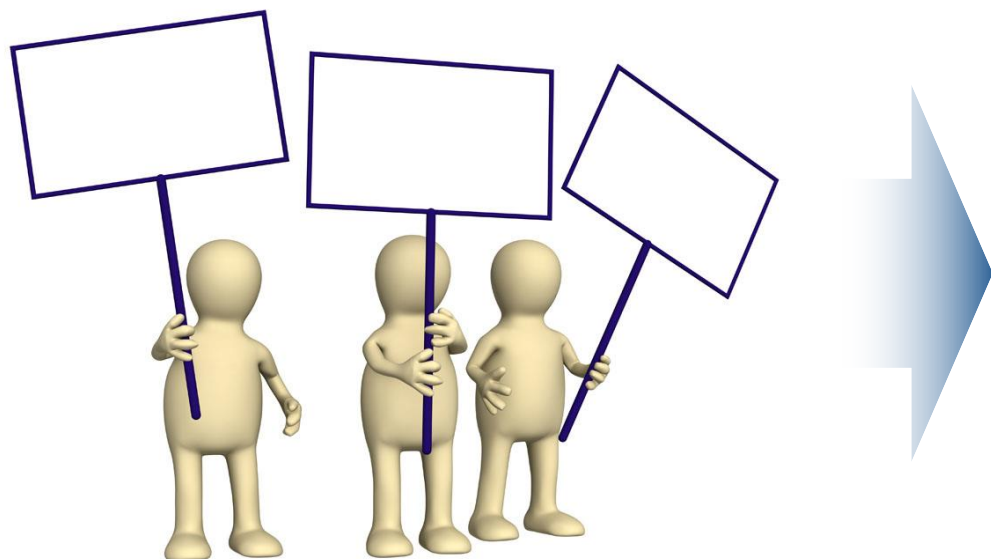
Scrum主管



团队

- 直接管理项目
 - 引导团队正确应用敏捷实践
 - 组织每日站立会议
 - 帮助团队制定冲刺计划
 - 排除团队遇到的障碍
 - 确保团队功能完备富有效率
 - 促进团队紧密协作
 - 保护团队不受打扰
-
- 一般是5-9人
 - 团队是跨职能的
 - 成员都全职工作
 - 自我组织和管理
 - 合作完成冲刺开发工作
 - 保证每一次冲刺的成功

人员的选择



应用领域经验

平台经验

编程语言经验

教育背景

解决问题能力

沟通能力

适应性

工作态度

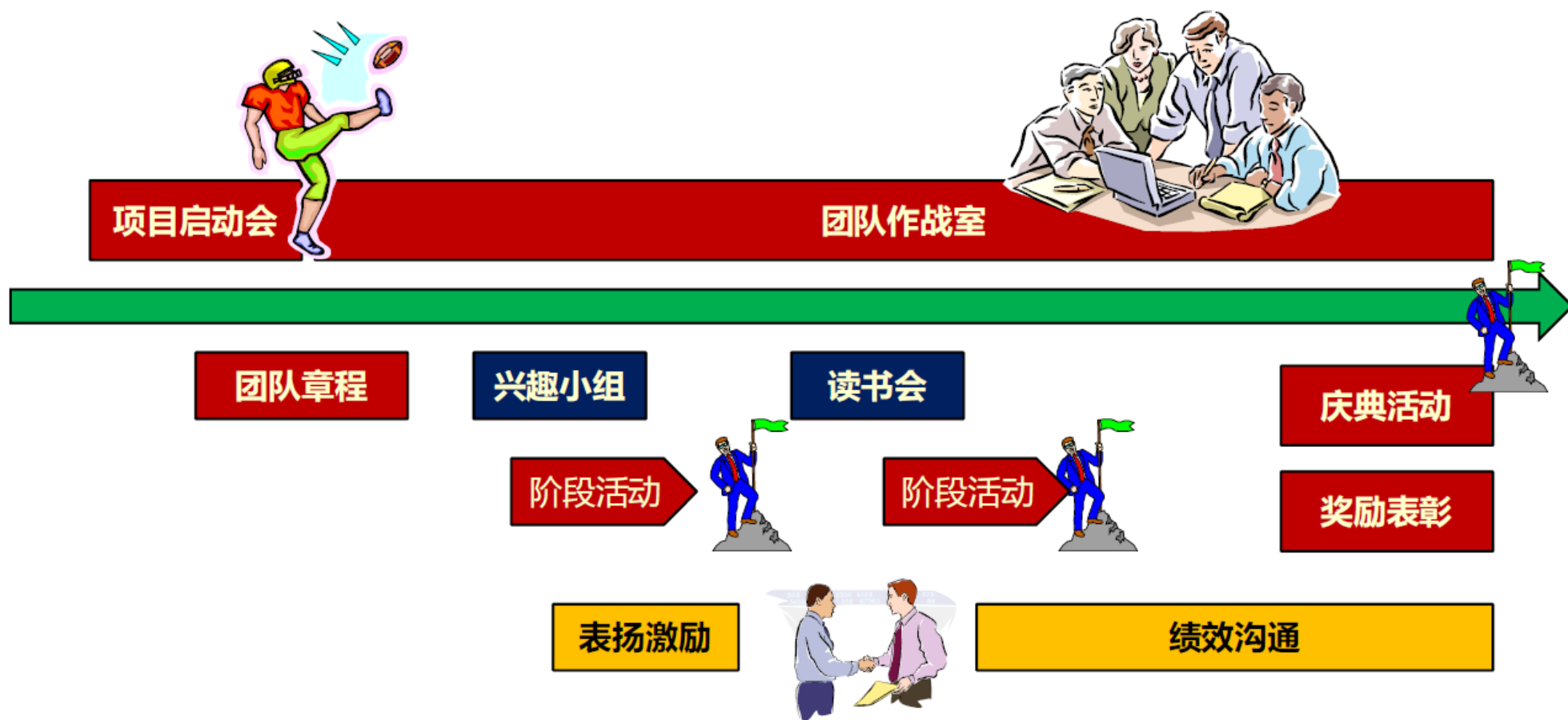
个性

人员的选择



- 应该考虑团队中的技术、经验和个性是否整体均衡。
- 选择性格互补的成员组成的团队可能比仅仅根据技术能力选择成员的团队更有效率。
- 团队的领导力来自于成员的尊重，而不是名义上的头衔。

团队建设活动



XXX 开发团队章程

（一）团队开发

1. 每周一晚8:00召开小组成员讨论会，地点在紫荆公寓XXXX。周例会主要总结上周的课程内容和开发进展，商讨本周软件工程项目开发情况。要求每位组员表达个人的学习心得，交流自己遇到的问题。
2. 每周三晚和周五上午组织团队集中开发，分享结对编程的成果。根据调研的已有产品，完善自己的产品设想和创意。
3. 每次集中开发之前举行一次10分钟的站立会议，组员报告自己的开发情况和遇到的困难，以便及时了解项目的进展情况。
4. 所有组员每周应完成规定的个人任务，并发布在配置系统中。

（二）团队管理

1. 全体组员必须参加每周例会和集中开发，组长最好提前一天进行提醒；如发生变更，由组长提前通知大家。
2. 采用计分方式记录每个成员对团队开发的参与度，参加一次记1分，迟到和早退者需要为团队开发贡献水果。
3. 周例会应简短有效，整个团队应事先认真准备，组长负责主持会议，组员轮流做会议纪要。
4. 团队成员应主动承担开发任务，并对任务做出合理的承诺，按照承诺的时间完成规定的开发任务。
5. 团队成员应积极进行自学和互学，提倡与其他团队的技术交流，努力提高个人的技术能力和团队的综合实力。
6. 组长应定期与老师和助教进行沟通和交流，准确了解项目的要求，及时汇报团队的项目进展情况，必要时寻求一定的帮助。

（三）团队建设

1. 每名组员在每周要有一定的体育锻炼时间，为一周的繁重学业任务储备体能。
2. 加强组员之间的交流，分享学习成果和学习材料，鼓励大家坚持撰写技术博客。
3. 在每次实验项目阶段检查达到预定目标时，组长负责组织全体组员举行一次 Pizza Party，共同庆祝项目开发取得阶段性成果。



要求：

- 讨论并确定你们团队的章程
- 每周应该有集中例会和开发时间
- 章程重在执行，团队成员要遵守规定
- 每两周和项目的用户代表见面交流

项目沟通的重要性

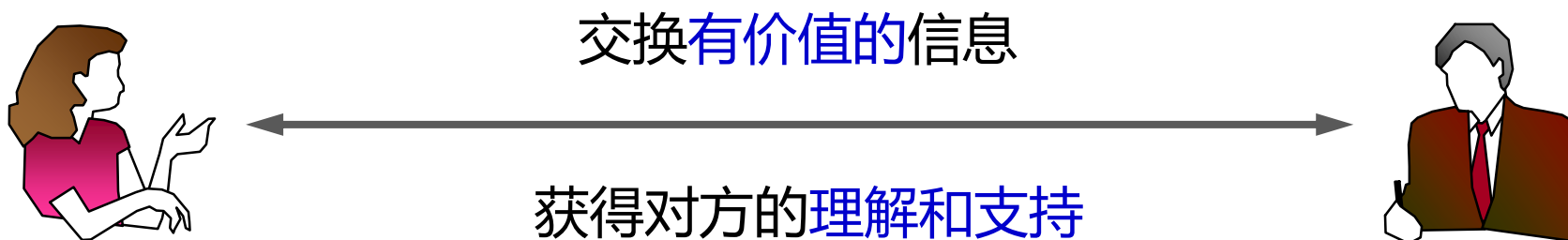


普林斯顿大学研究发现：“智慧、专业技术、经验三者只占成功因素的25%，其余75%决定于良好的人际沟通。”

美国著名成功学专家卡耐基认为：“一个成功的企业家只有15%是靠他的专业知识，而85%是靠他的人际关系和处世技巧。”

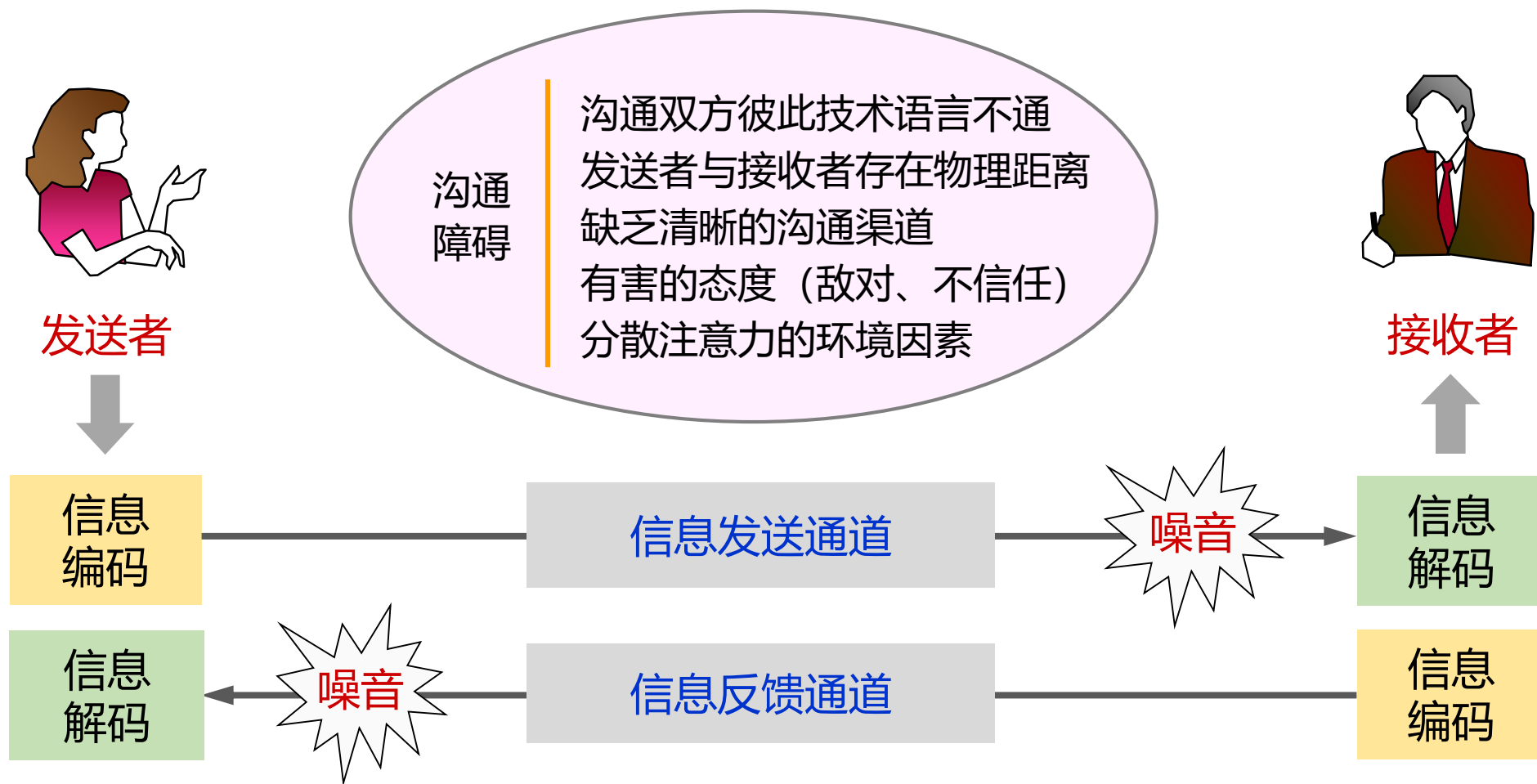
沟通的概念

沟通是为了达到一定的目的，将信息、思想、情感在个人或群体之间进行传递或交流的过程。



沟通是你被理解了什么而不是说了什么

沟通的模型

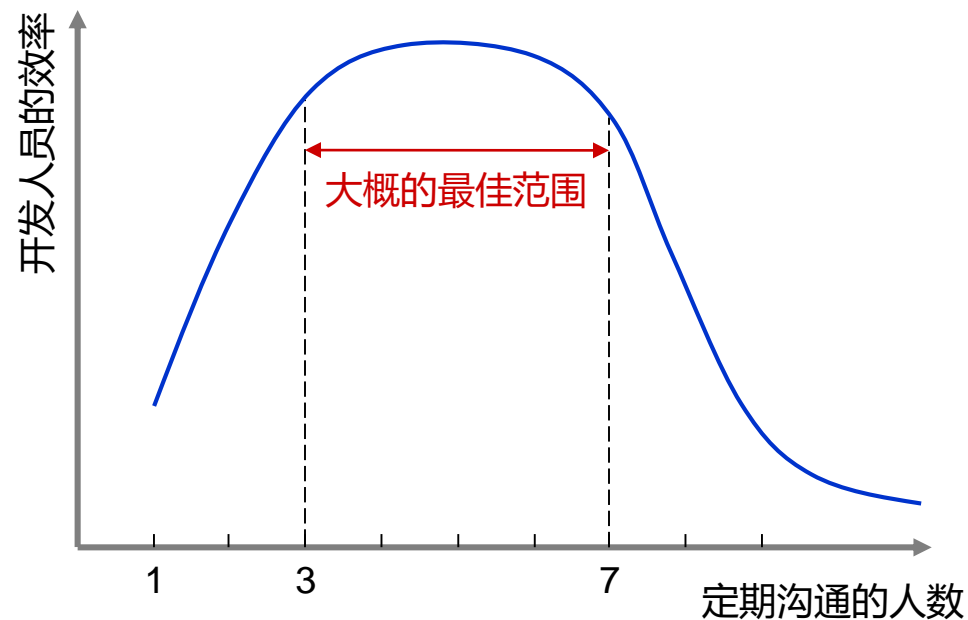
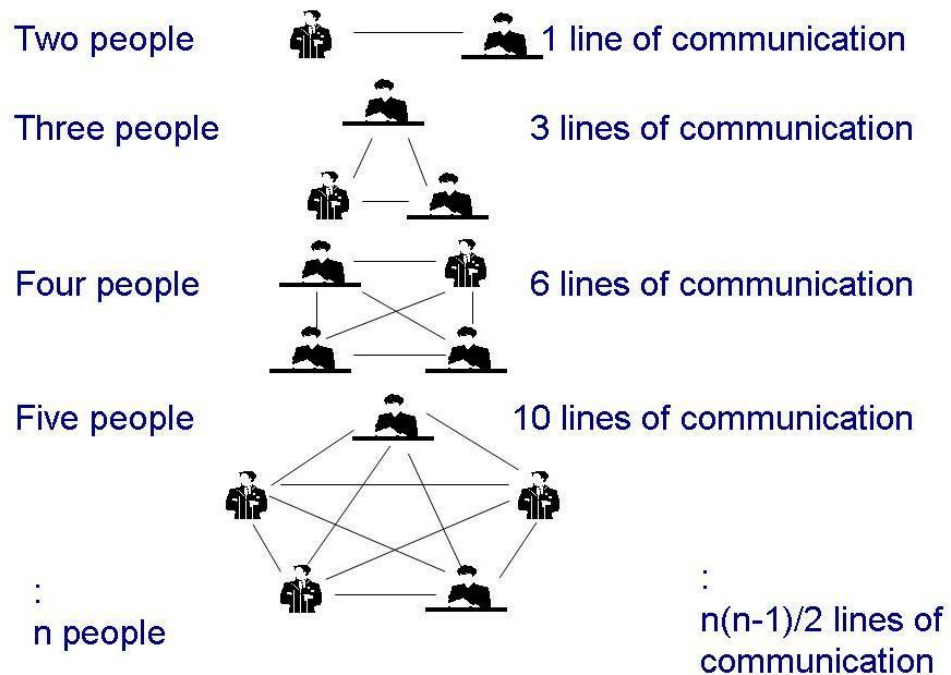


沟通的复杂性



沟通的复杂性

Brooks法则： 向一个进度延迟的软件项目中增加人员可能会使其进度更加推迟。



常用沟通方式



与管理层和客户的沟通



- 和谁进行沟通？为什么？
- 需要什么类型的信息？
- 详尽程度和频率如何？
- 沟通的目标是什么？
- 采用何种方式沟通比较好？

项目团队内部的沟通



沟通需求：

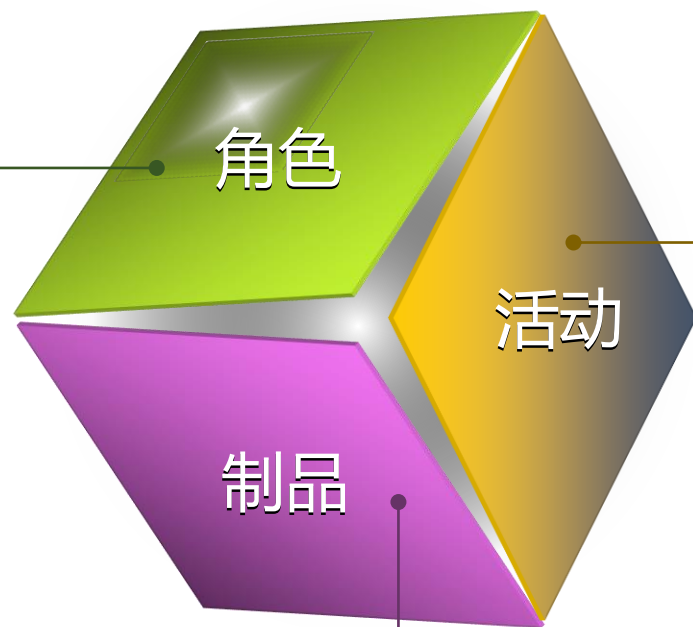
- 职责、协调、状态、授权
- 任务分配清晰

工作会议：

- 项目启动会、成员进度汇报、项目进展会
- 设置沟通期望
- 及时、公开、恰到好处

项目会议

- 产品负责人
Product Owner
- Scrum主管
Scrum Manager
- 团队成员
Team



- 迭代计划会议
Sprint Planning
- 每日站立会议
Daily Scrum Meeting
- 迭代评审会议
Sprint Review
- 迭代回顾会议
Sprint Retrospective

- 产品订单 Product Backlog
- 迭代订单 Sprint Backlog
- 燃尽图 Burndown Charts

迭代计划会议

迭代计划会议在每次迭代（或冲刺）开始时召开，一般是2~4小时，目的是选择和估算本次迭代的工作项。

- 第一部分：以需求分析为主，选择和排序本次迭代需要实现的订单条目
- 第二部分：以设计为主，确定系统设计方案和工作内容



每日站立会

会议目的：

- 团队在会议中做计划，协调其每日活动，还可以报告和讨论遇到的障碍。
- 任务板帮助团队聚焦于每日活动上，应在这个时候更新任务板和燃尽图。

每个团队成员需要回答三个问题：

- 上次例会后完成了什么？
- 遇到了什么困难（或障碍）？
- 下次例会前计划做什么？



迭代评审会议

会议目的：

- Scrum团队在会议中向最终用户展示迭代的工作成果，团队成员希望得到反馈，并以之创建或变更 Backlog 条目。

基本要求：

- 由团队展示有可能发布的产品增量
- 允许所有参与者尝试由团队展示的新功能
- 用户对团队演示的产品功能进行反馈



迭代回顾会议

每一次迭代完成后，都会举行一次迭代总结会，会上所有团队成员都要反思这个迭代。举行迭代总结会议是为了进行持续过程改进，时间限制在1小时左右。

迭代回顾会议的关键要点：

- 会议气氛：团队全员参加，气氛宽松自由，畅所欲言，发现问题和分析原因；
- 关注重点：每次仅就1-3个关键问题做出可行的解决方案；
- 跟踪闭环：可以放入下一次迭代订单中执行改进。

团队绩效评估

绩效评估是通过对团队成员工作绩效的考察与评价，反映团队成员的实际能力和业绩以及对某种工作职位的适应度。



个人工作量？

个人工作时间？

团队平均主义？

团队绩效评估



		团队贡献维度		
		前20%	中间70%	后10%
完成任务维度	好			
	中			
	差			



谢谢大家!

THANKS

