

软件过程

软件过程

软件过程

软件过程模型

敏捷开发过程

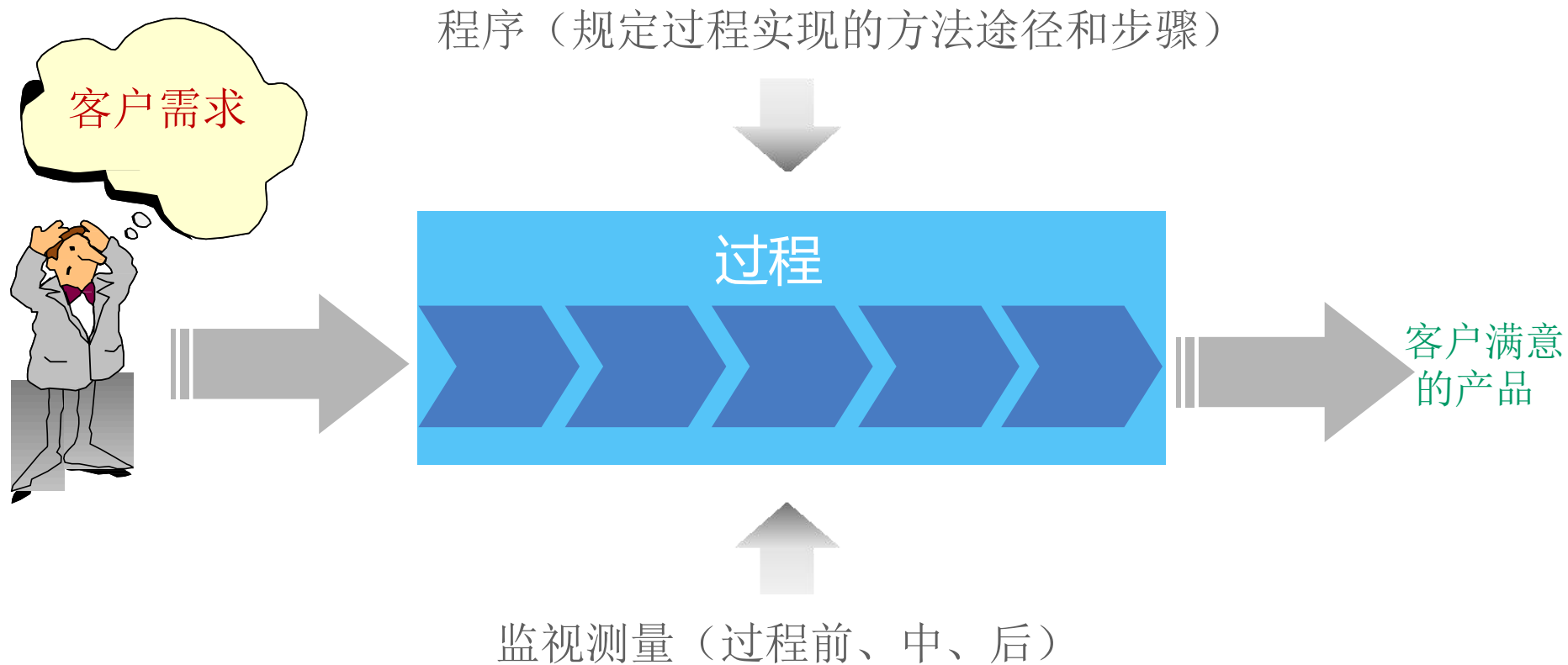
软件过程

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

过程是一组将输入转化为输出的相互关联或相互作用的活动。



举例：麦当劳的过程管理



举例：麦当劳的过程管理



- 面包直径为**17cm**，这种尺寸入口最佳
- 面包中的气泡为**0.5cm**，这种尺寸味道最佳
- 牛肉食品的品质检查有**40**多项内容
- 肉饼成分讲究，由**83%**肩肉与**17%**五花肉混制而成
- 牛肉饼重量在**45克**时其边际效益达到最大
- 汉堡包从制作到出炉时间严格控制在**5秒钟**
- 一个汉堡包净重**1.8盎司**，其中洋葱重量为**0.25盎司**

举例：麦当劳的过程管理



- 汉堡包出炉后不能超过**10分钟**
- 薯条炸好后不能超过**7分钟**
- 汉堡包饼面上不能有人工手压的痕迹
- 与汉堡包一起卖的可口可乐必须是 **4℃**
- 柜台高度**92cm**，绝大多数顾客付账取物最方便

举例：麦当劳的过程管理

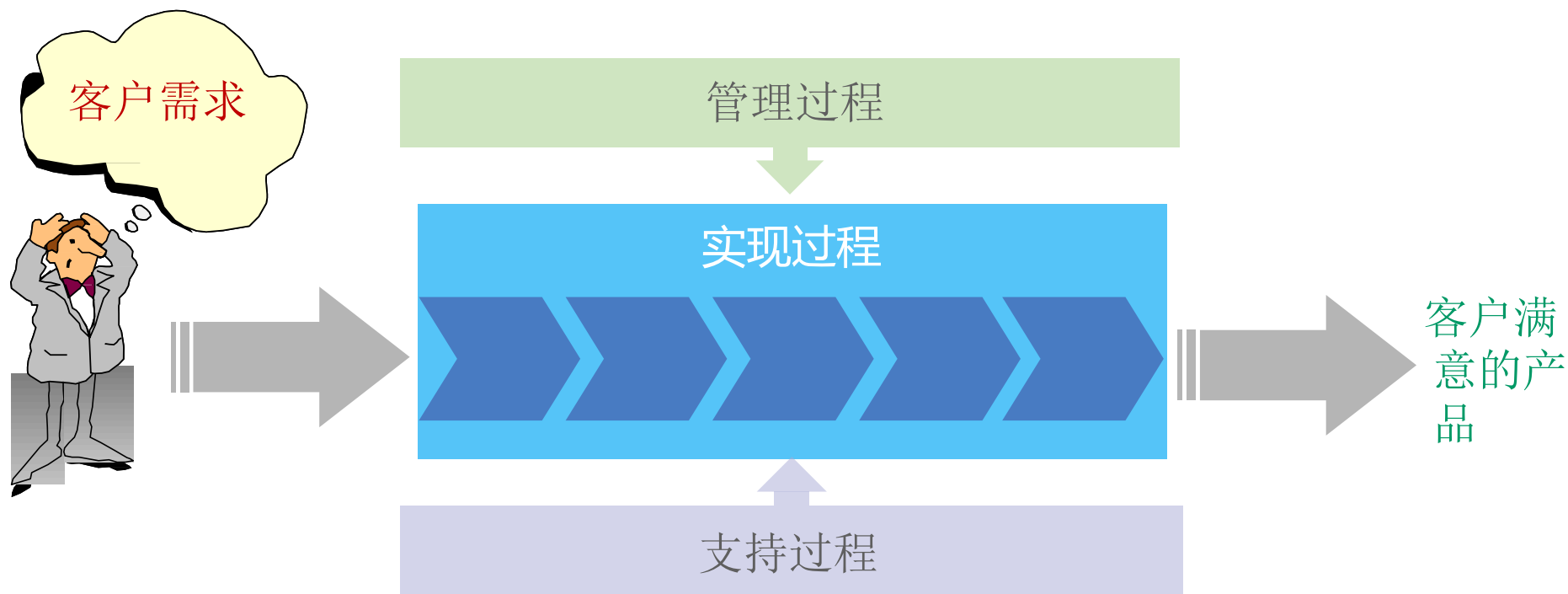


无论何人 无论何时
无论何地 无品质差异

质量控制：在过程细节上精益求精

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

软件开发活动

问题定义

- 构想文档
- 用户故事

需求开发

- 分析模型
- 软件需求规格说明

软件设计

- 设计模型
- 软件体系结构文档
- 软件详细设计文档

软件构造

- 源程序
- 目标代码
- 可执行构件

软件测试

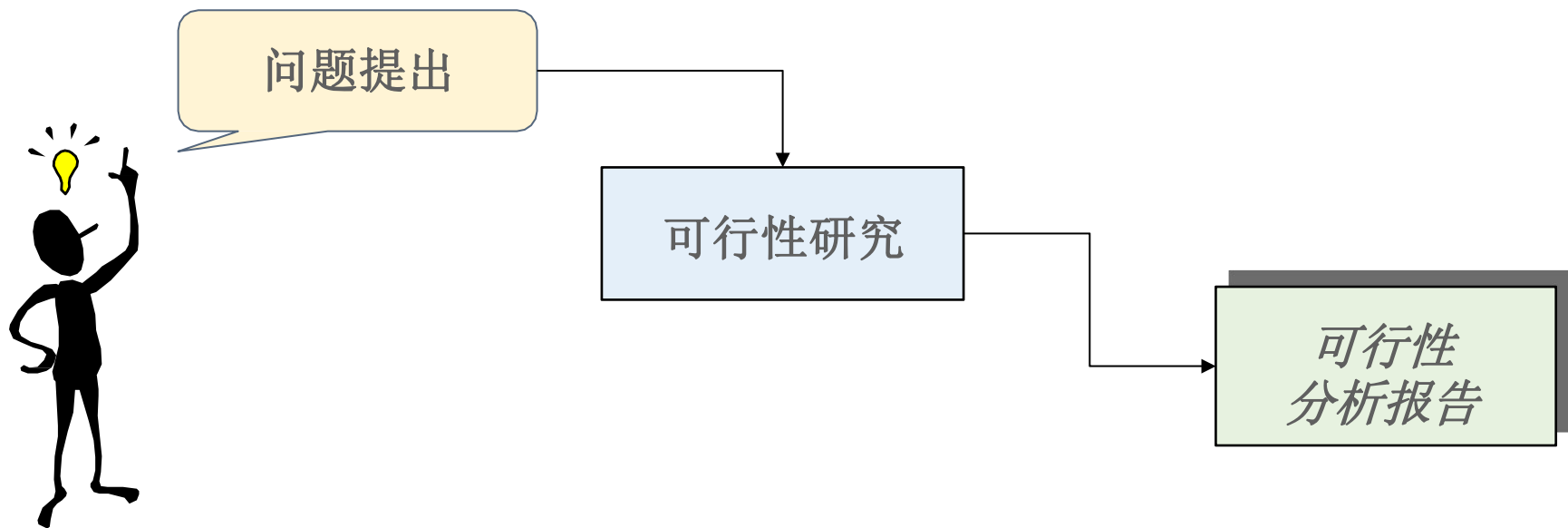
- 测试规程
- 测试用例
- 测试报告

软件开发管理与支持活动

（软件项目管理计划、软件配置管理计划、软件质量保证计划、评审记录.....）

软件开发活动

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。



软件开发活动



参加学校举办的活动和演出，
学生总是需要排队买票或领票

烦！



- WebApp
- 手机App
- 微信应用

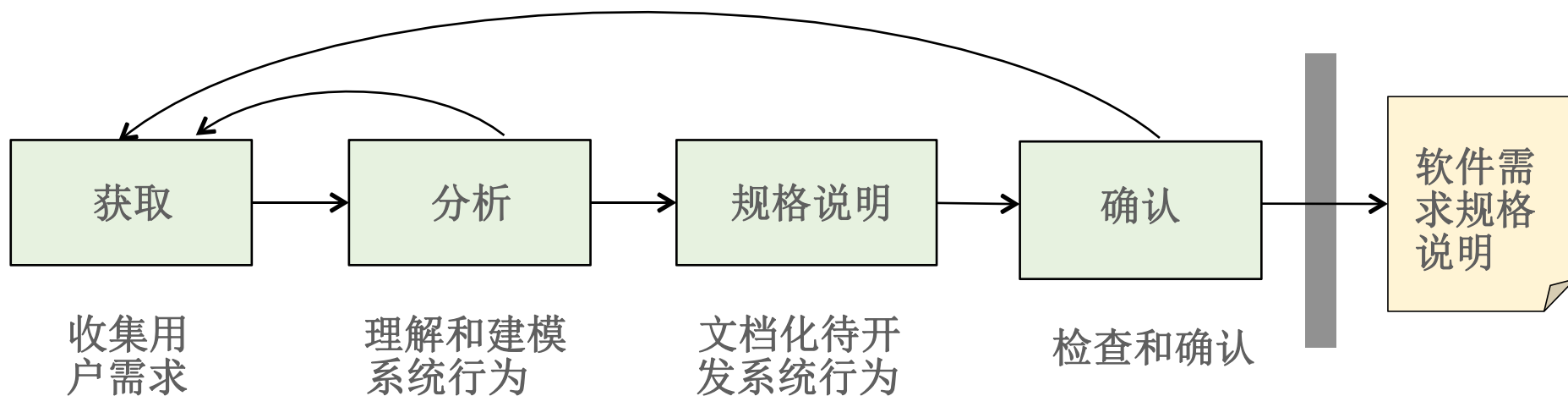


目标：微信抢票应用

爽！

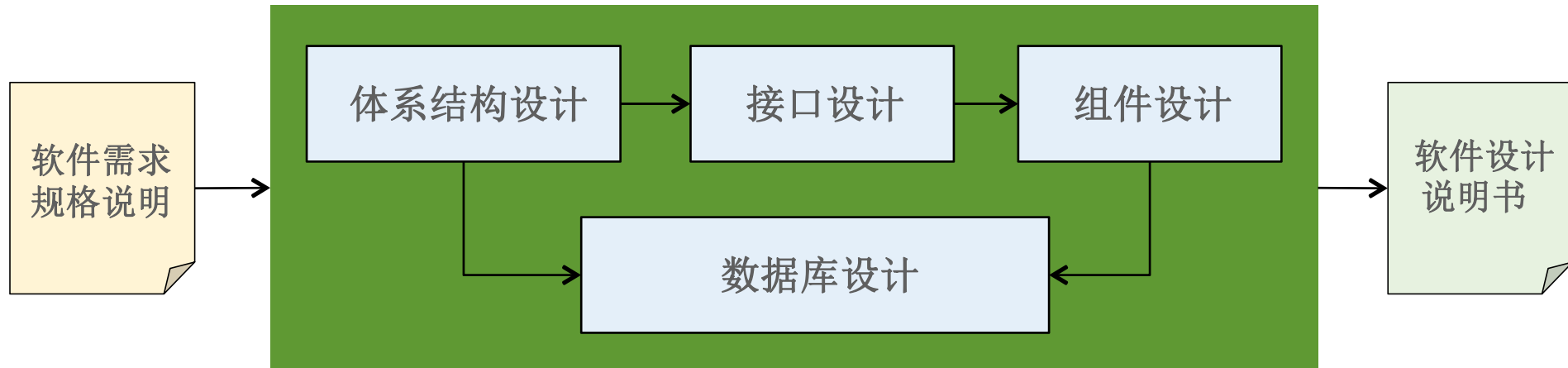
软件开发活动

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。



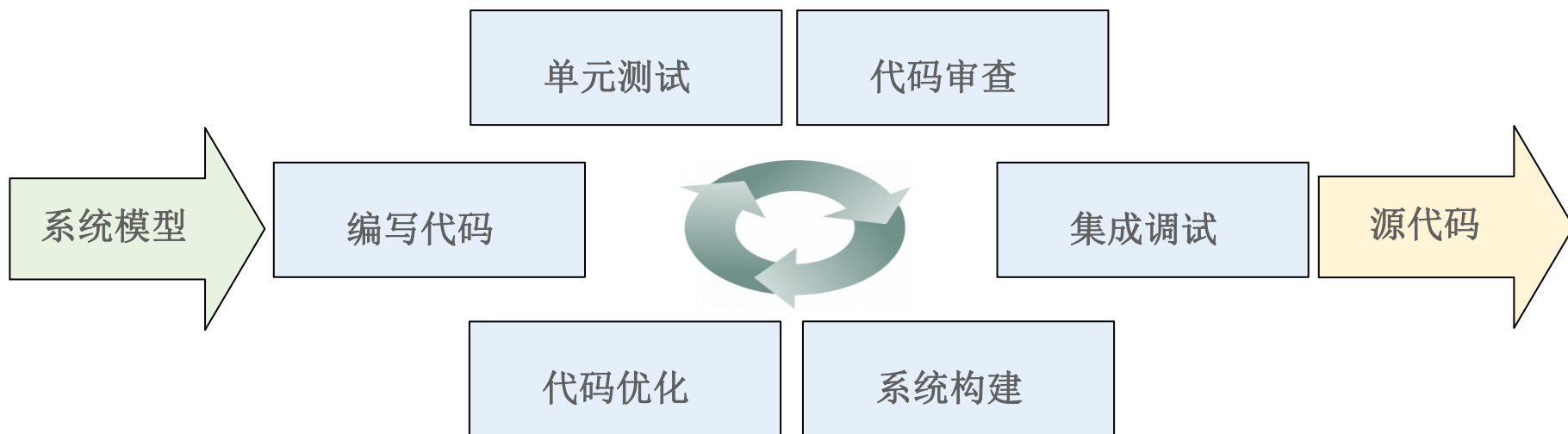
软件开发活动

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。



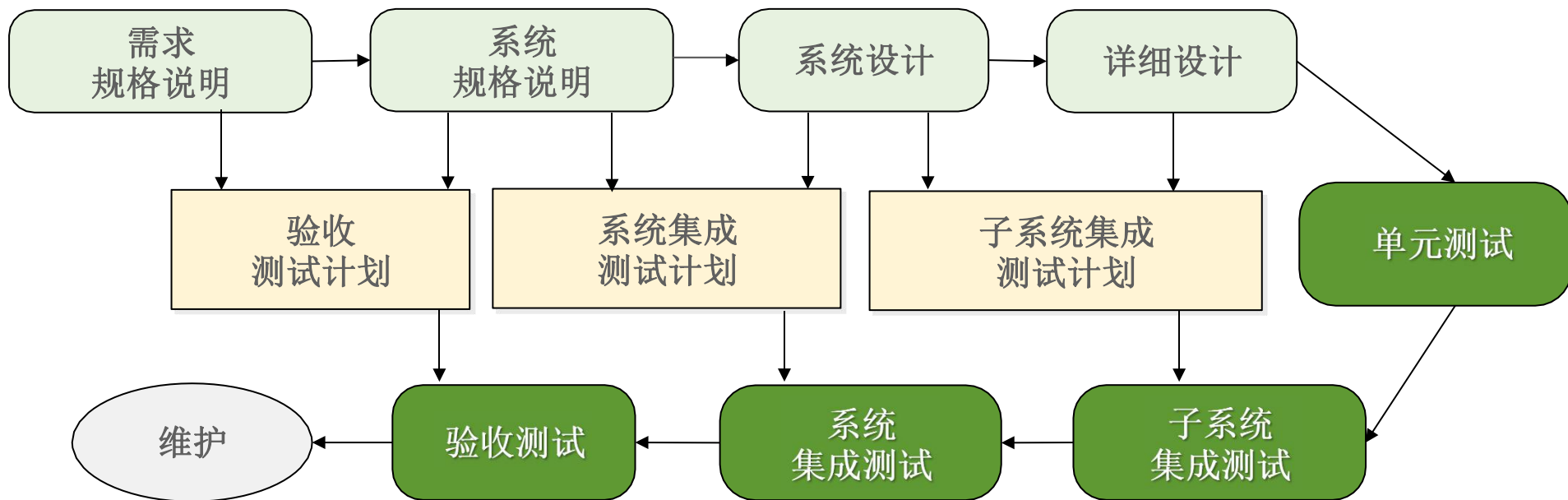
软件开发活动

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。



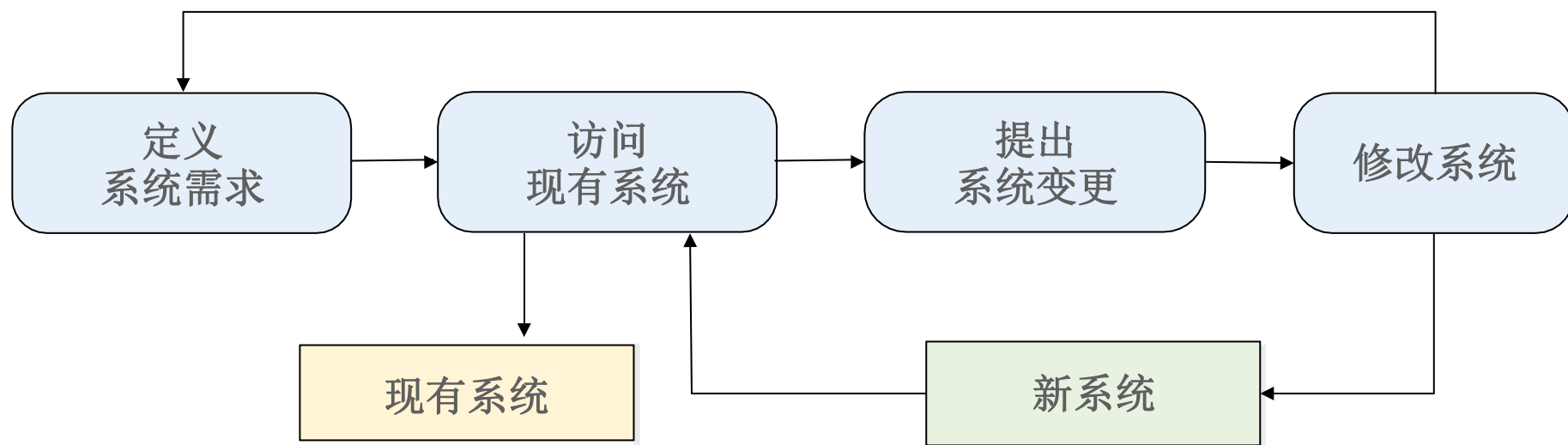
软件开发活动

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。



软件开发活动

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。



软件开发的基本活动

Requirements

Design

Implementation

Testing

Maintenance

Each phase requires different tools, knowledge, skill-set. A lot of ways to split responsibilities!

How are these related?
What is a good order?

软件生命周期 Software Lifecycle

Requirements

Design

Implementation

Testing

Maintenance

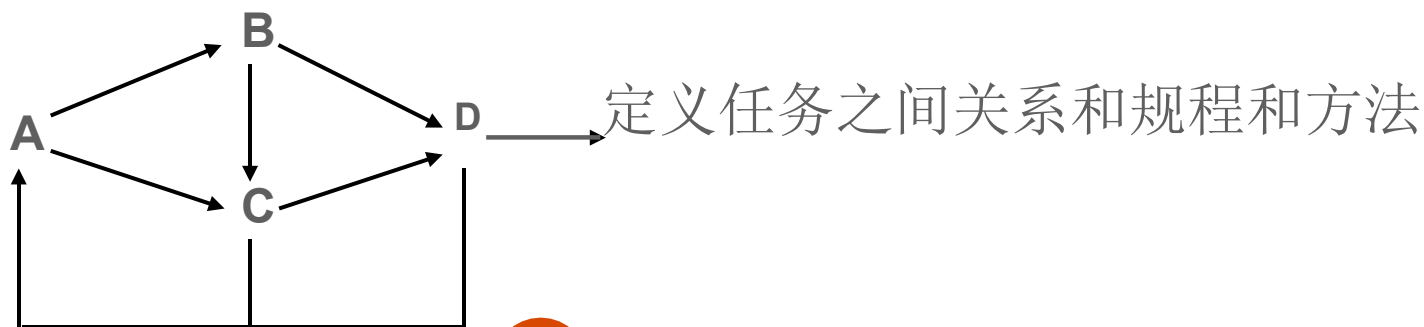
Software lifecycle is a series of phases through which software is produced:

- from conception to end-of-life
- can take months or years to complete

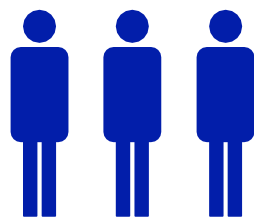
Goals of each phase:

- mark out a clear set of steps to perform
- produce a tangible item
- allow for review of work
- specify actions to perform in the next phase

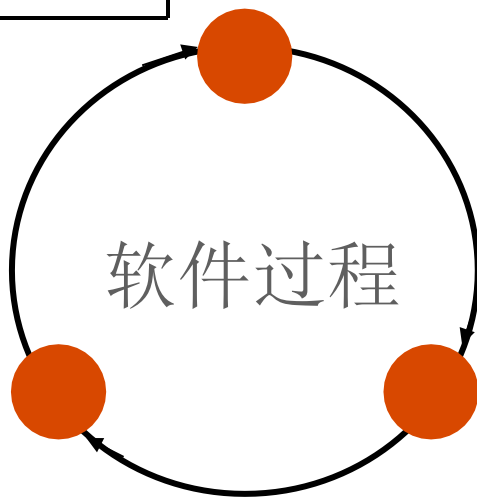
软件过程模型



有技能且受过
训练的人员



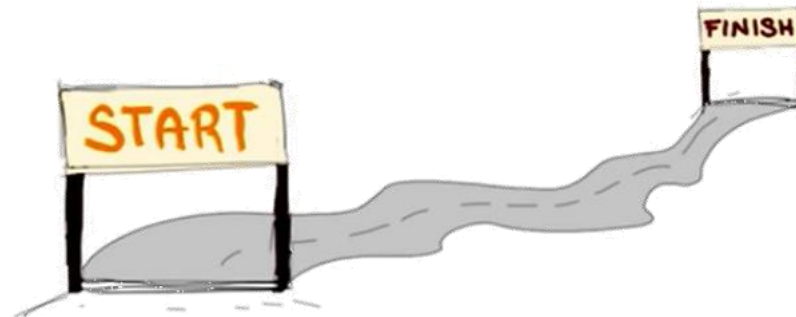
软件过程



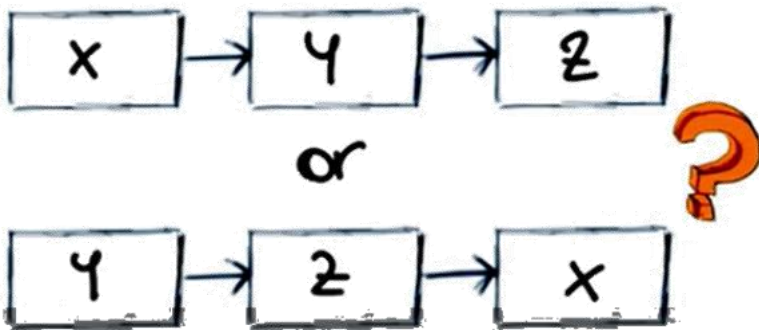
工具和设备

软件过程模型是对软件过程的抽象描述

软件过程模型



Determine the order



Establish The Transition criteria



软件过程模型

瀑布模型

将基本的开发活动看成是一系列界限分明的独立阶段，这是一种计划驱动的软件过程，有利于规范软件开发活动。

原型化模型

原型是一个部分开发的产品，用于加强对系统的理解，有助于明确需求和选择可行的设计策略。

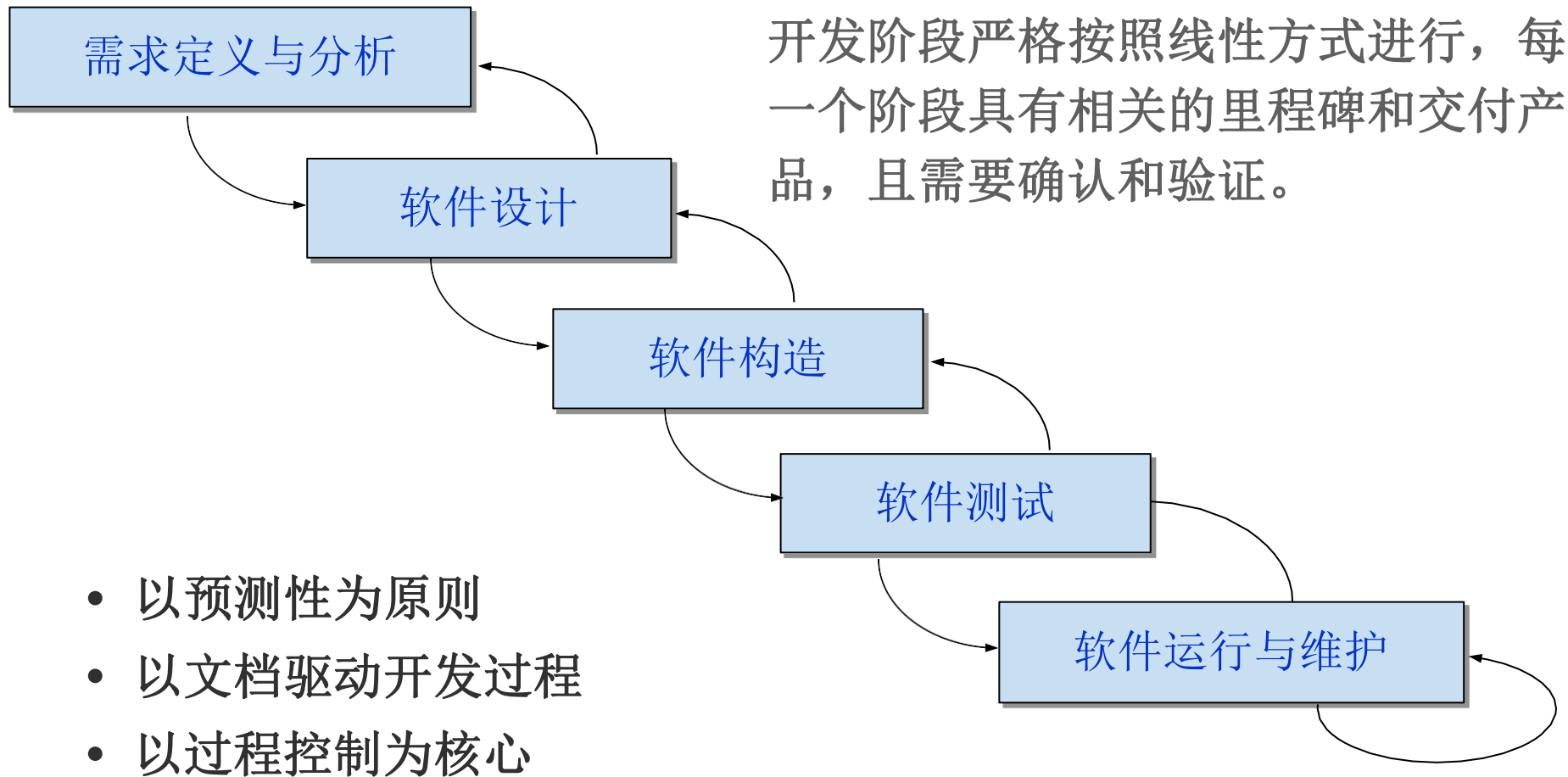
迭代式开发

将描述、开发和验证等不同活动交织在一起，在开发过程中建立一系列版本，将系统一部分一部分地逐步交付。

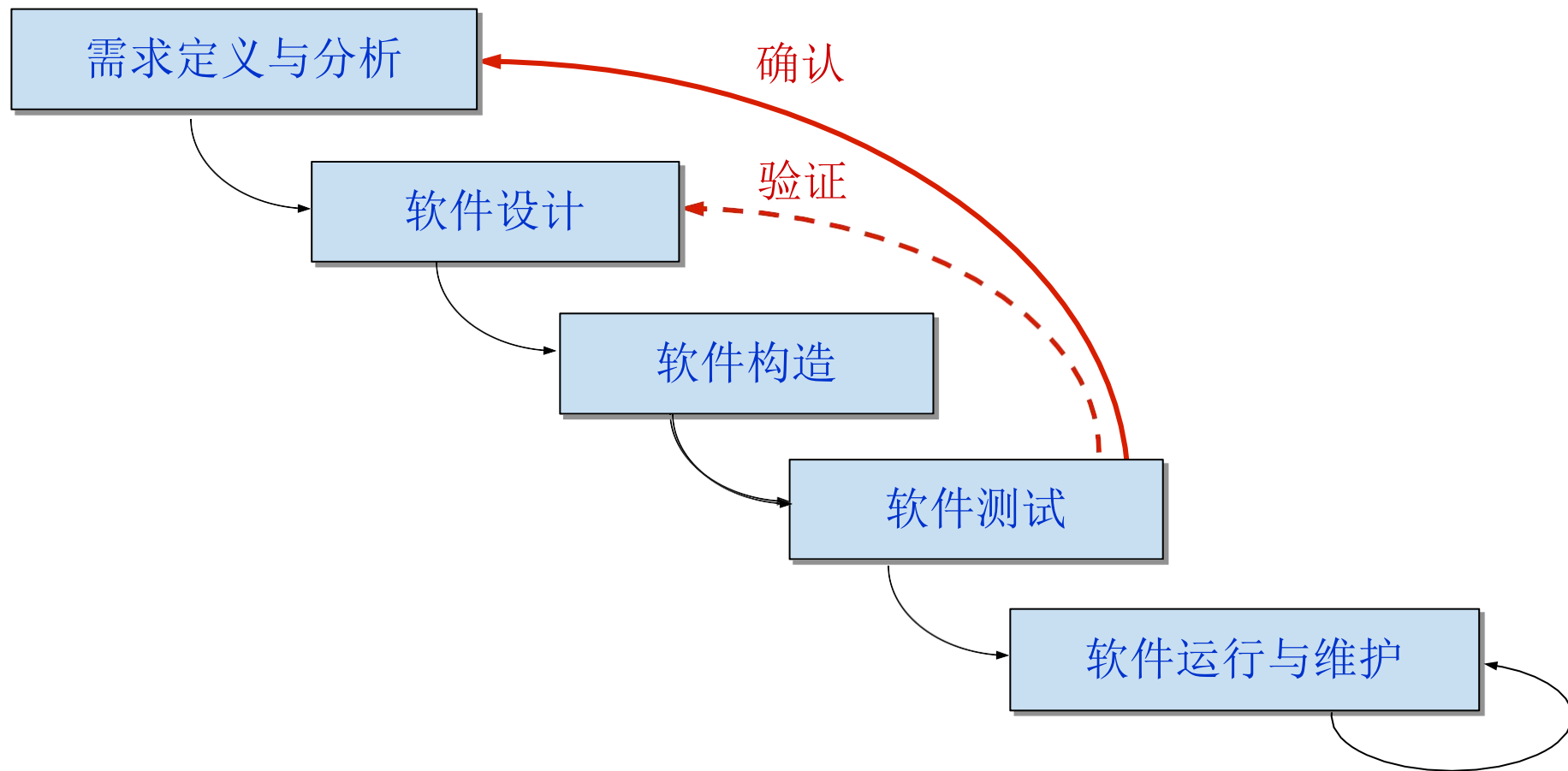
可转换模型

利用自动化的手段，通过一系列转换将需求规格说明转化为一个可交付使用的系统。

瀑布模型



瀑布模型



瀑布模型

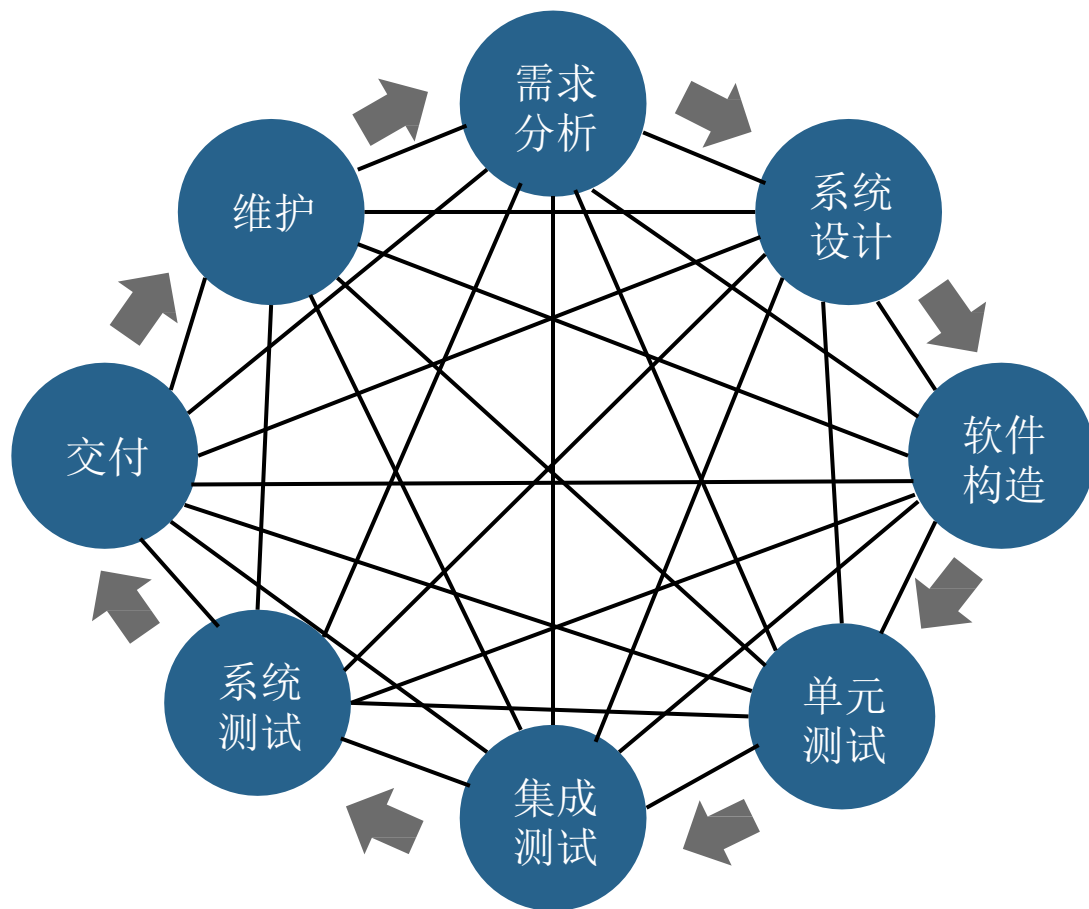
问题

- 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？



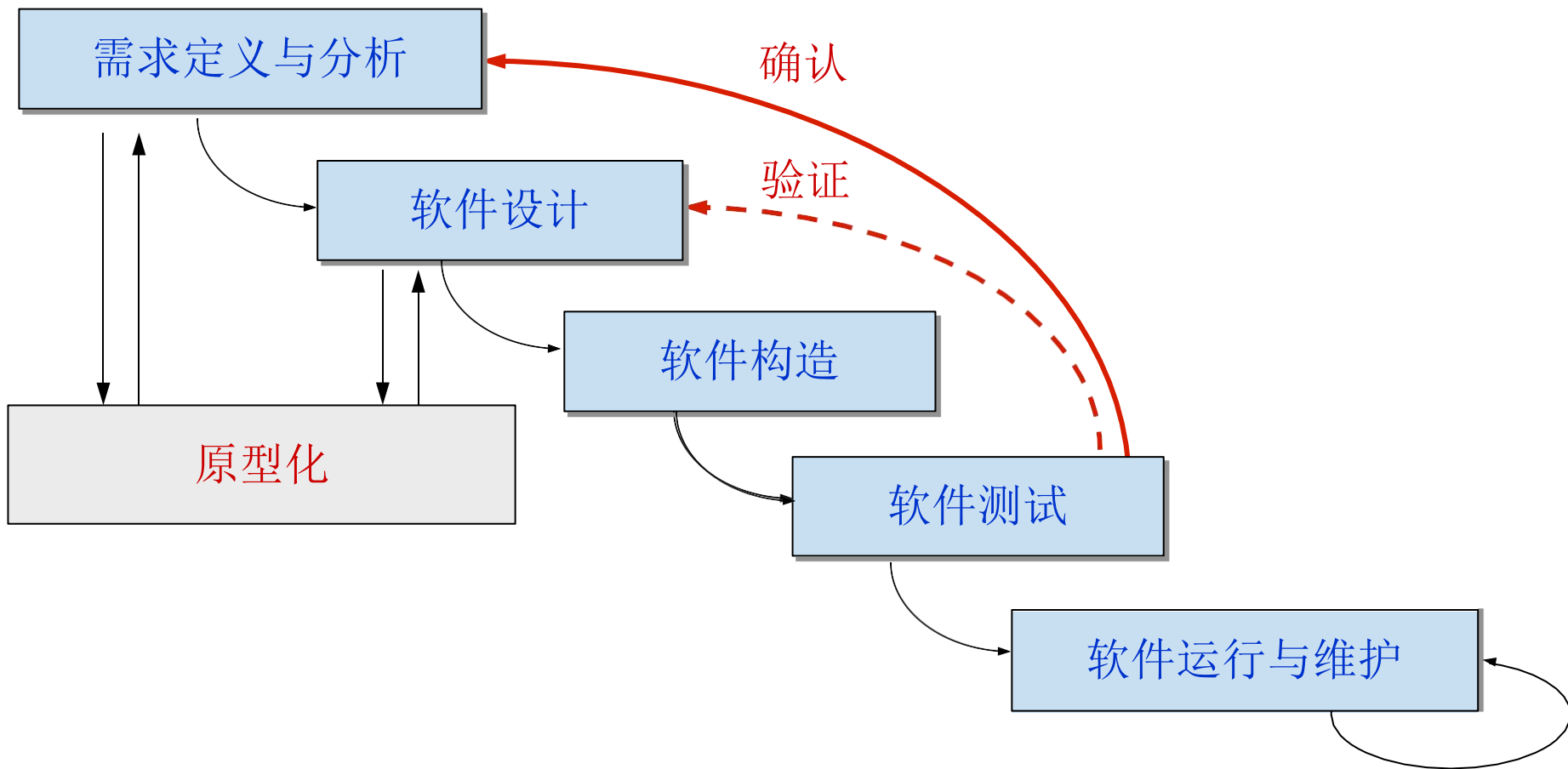
软件开发的迭代性

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。



软件开发是一个**创造**的过程，而不是一个**制造**的过程。

原型化模型



原型化模型



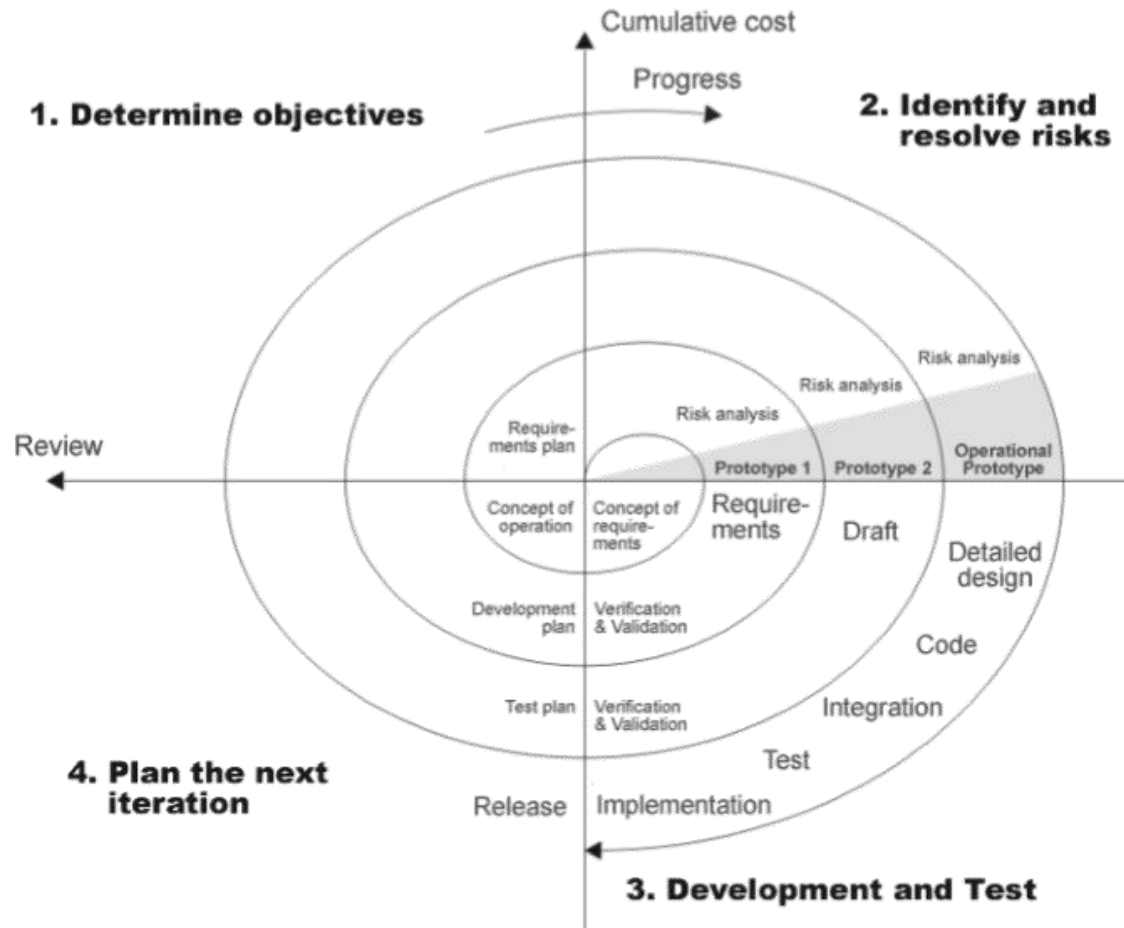
http://v.youku.com/v_show/id_XMjcyMTM1OTI=.html

迭代式开发



- 更快速地发布产品
- 追求产品创新
- 需求不确定性高
- 需要快速响应用户的变化
- 关注用户行为

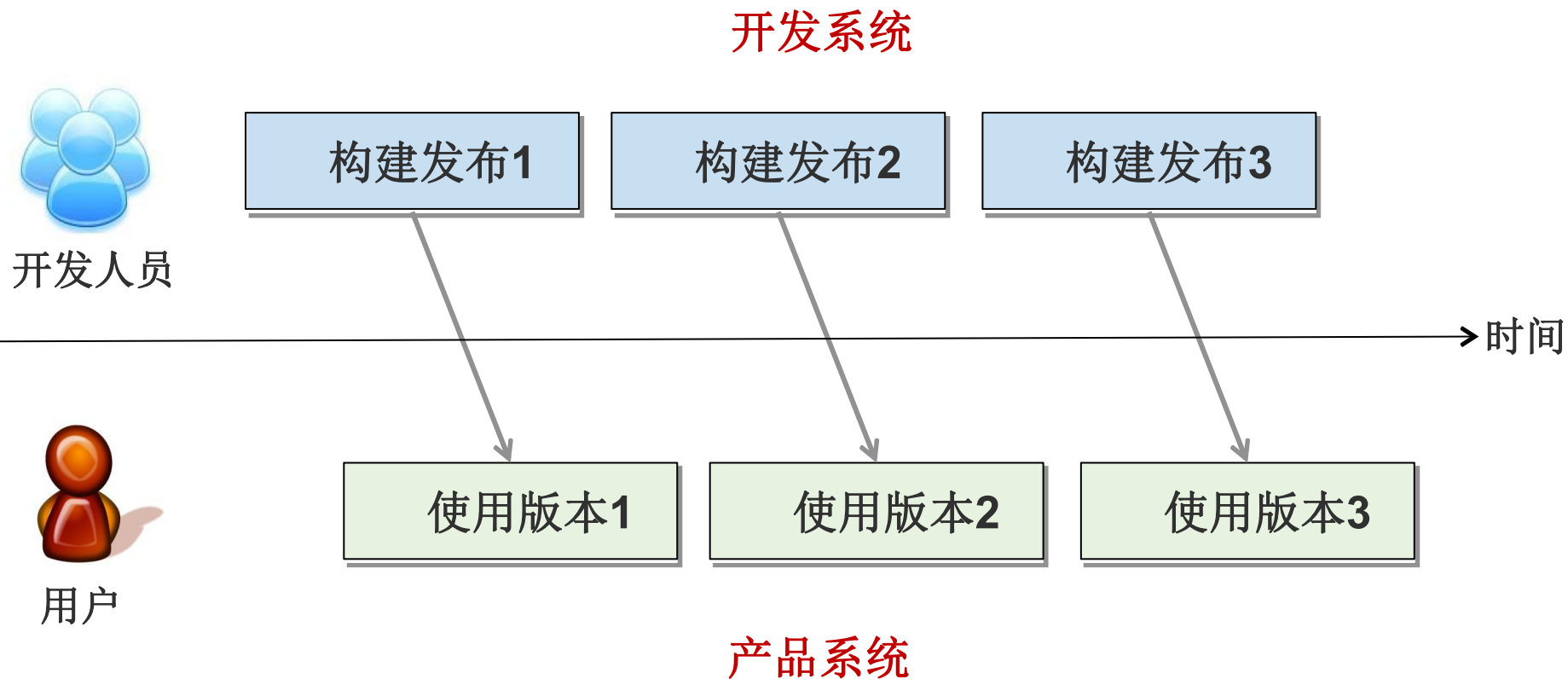
螺旋模型



Advantages & Disadvantages

- Especially appropriate at the beginning of the project, when the requirements are still fluid
- Provides early indication of unforeseen problems
- Accommodates change
- As costs increase, risks decrease!
- Always addresses the biggest risk first
- A lot of planning and management
- Frequent changes of task
 - But, get to stick with one product feature/goal
- Requires customer and contract flexibility
- Developers must be able to assess risk
 - Must address most important issues

迭代式开发

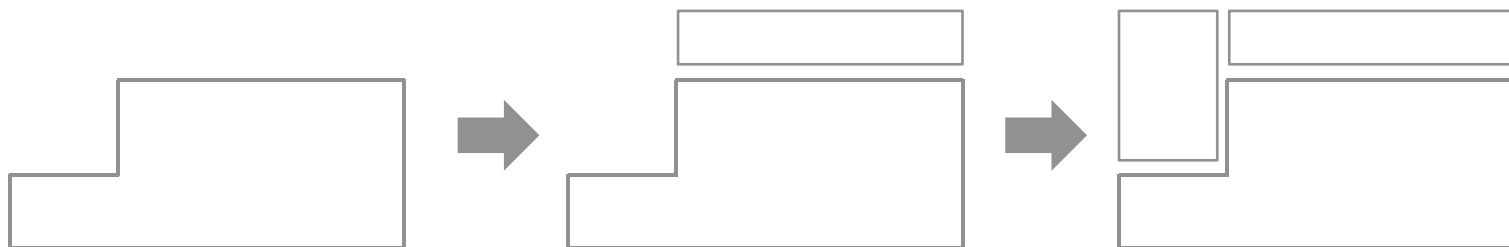


迭代式开发

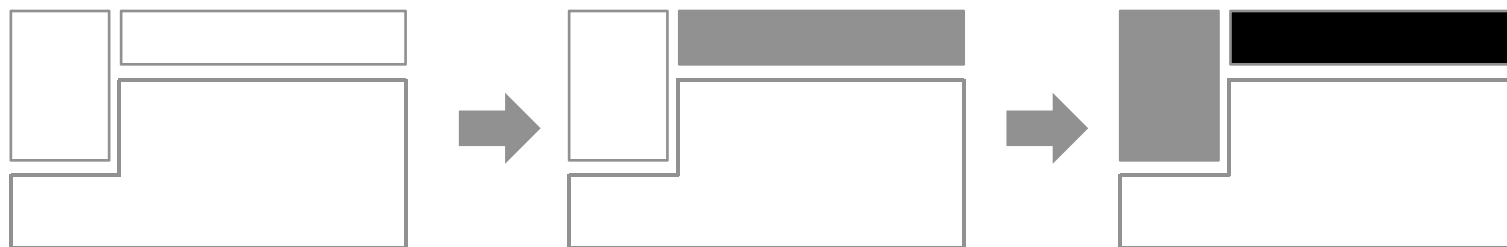
增量模型：在每一个新的发布中逐步增加功能直到构造全部功能。

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



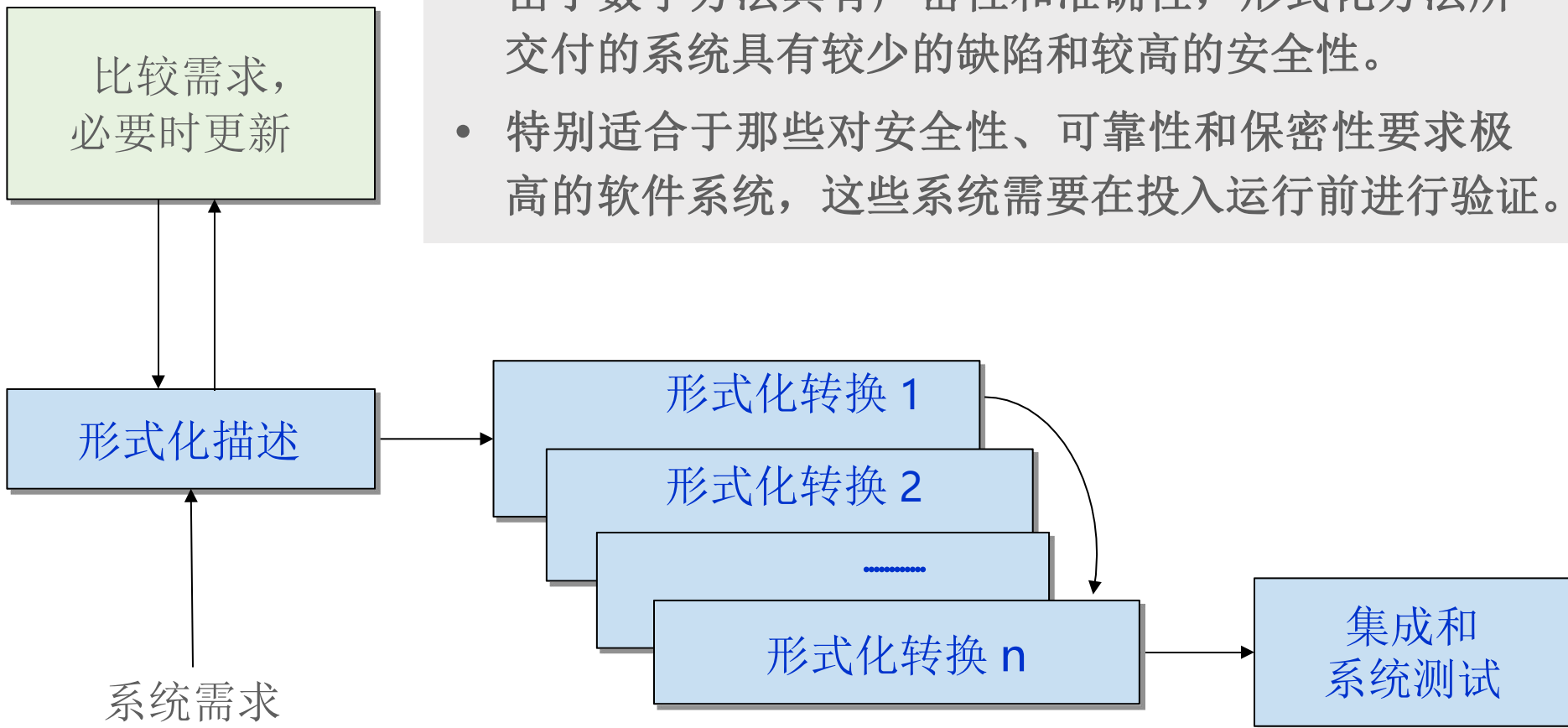
增量模型

迭代模型



可转换模型

- 由于数学方法具有严密性和准确性，形式化方法所交付的系统具有较少的缺陷和较高的安全性。
- 特别适合于那些对安全性、可靠性和保密性要求极高的软件系统，这些系统需要在投入运行前进行验证。



案例分析



汽车制动防抱系统



网络公开课程网站

案例分析

实例一：汽车制动防抱系统

- 汽车制动防抱系统（**Anti-locked Braking System**，简称**ABS**）是一种具有防滑、防锁死等优点的汽车安全控制系统。
- 该系统通过汽车微电脑控制，以非常快的速度精密地控制制动液压力的收放，从而达到防止车轮抱死，确保轮胎的最大制动力以及制动过程中的转向能力，使车辆在紧急制动时具有躲避障碍的能力。



案例分析

实例一：汽车制动防抱系统

- 汽车制动防抱系统（**Anti-locked Braking System**，简称**ABS**）是一种具有防滑、防锁死等优点的汽车安全控制系统。
- 该系统通过汽车微电脑控制，以非常快的速度精密地控制制动液压力的收放，从而达到防止车轮抱死，确保轮胎的最大制动力以及制动过程中的转向能力，使车辆在紧急制动时具有躲避障碍的能力。

实例分析：

- 嵌入式控制系统
- 该系统对安全性和可靠性要求极高，需要在投入运行前进行验证
- **适合采用可转换模型**



案例分析

实例二：网络公开课程网站

- 某公司准备开发一个大规模在线公开课程网站，支持学校将自己的课程录像、课件及参考资料等公布在网上，学生可以进行自主学习。
- 该系统将教育、娱乐和社交网络结合在一起，创造了一种新型的网络教育模式，对传统的高等教育模式带有很大的冲击。



案例分析

实例二：网络公开课程网站

- 某公司准备开发一个大规模在线公开课程网站，支持学校将自己的课程录像、课件及参考资料等公布在网上，学生可以进行自主学习。
- 该系统将教育、娱乐和社交网络结合在一起，创造了一种新型的网络教育模式，对传统的高等教育模式带了很多的冲击。

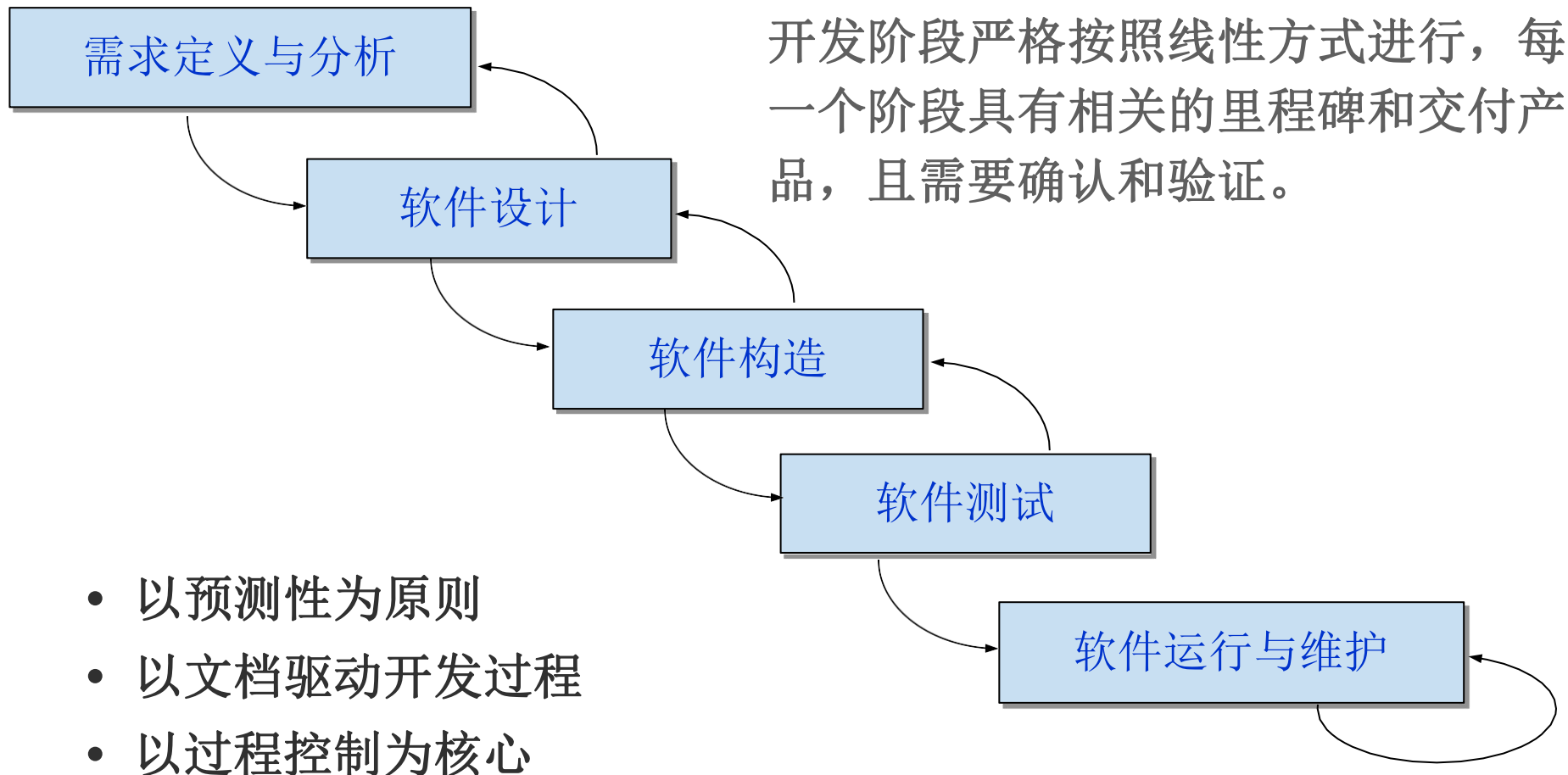
实例分析：

- 系统需求会经常发生变化，业务模式存在不确定性
- 系统应该易于维护和修改
- **适合采用迭代式模型**



敏捷开发过程

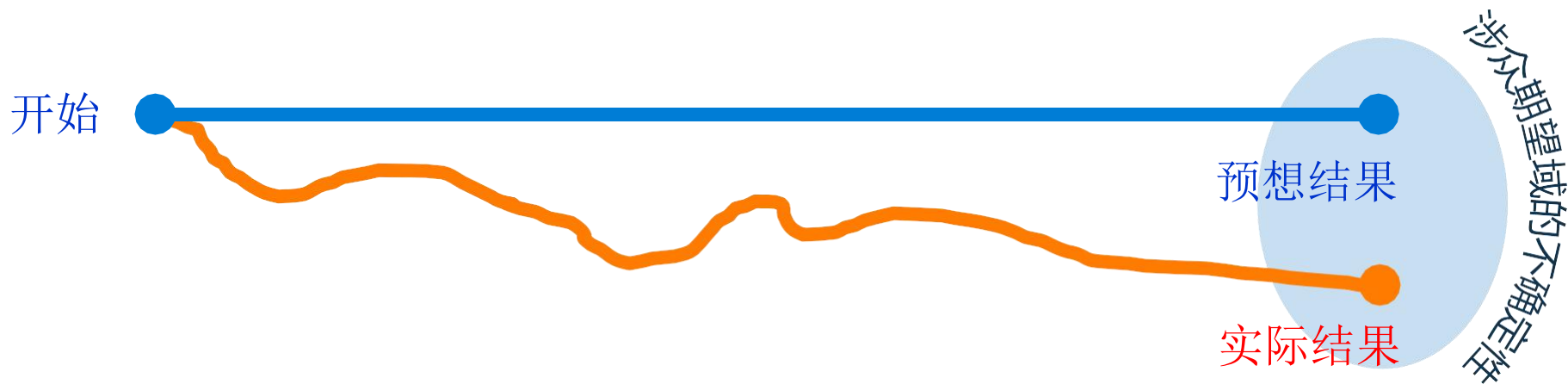
传统的瀑布模型



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！！！！



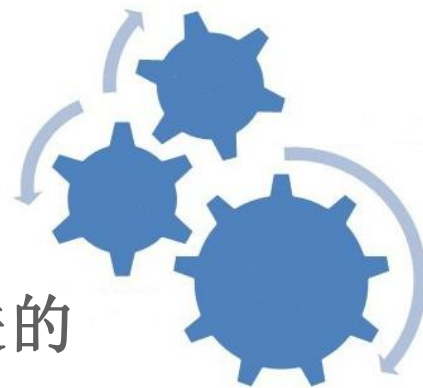
- ✓✓ 软件开发是一个逐步认知和明晰的活动
- ✓✓ 软件开发中的变化是实际存在和必然的

软件开发之道

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

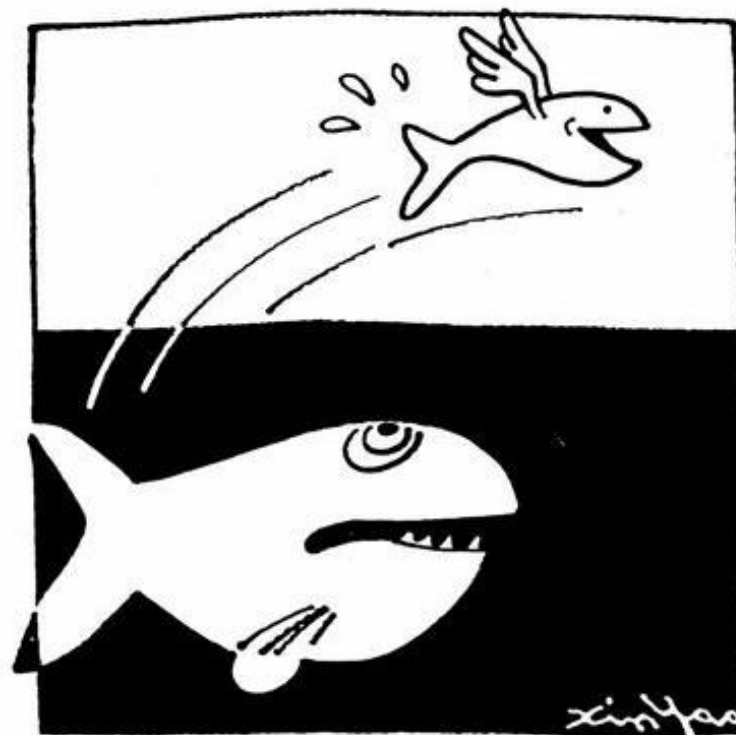
- ✓✓ 高质量的交付物是最重要的
- ✓✓ 系统不是一次构建而成，而是迭代演进的
- ✓✓ 基于完整的场景构建计划，并按优先级执行



软件开发之道

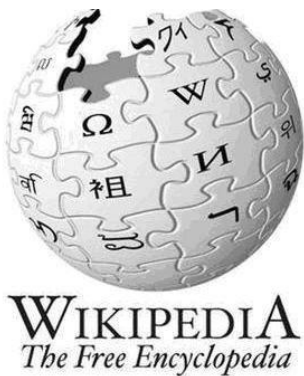
互联网产品的开发特点

- 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



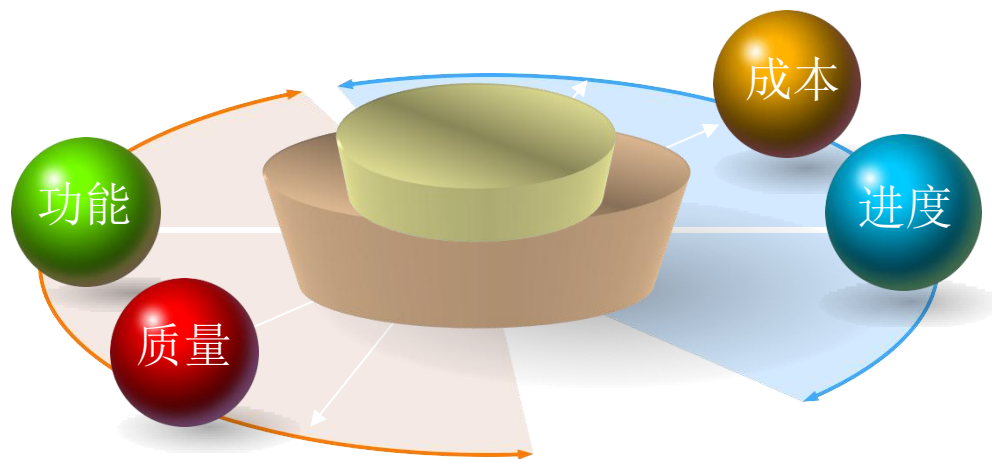
如今不是大鱼吃小鱼,而是快鱼吃慢鱼。

敏捷开发方法

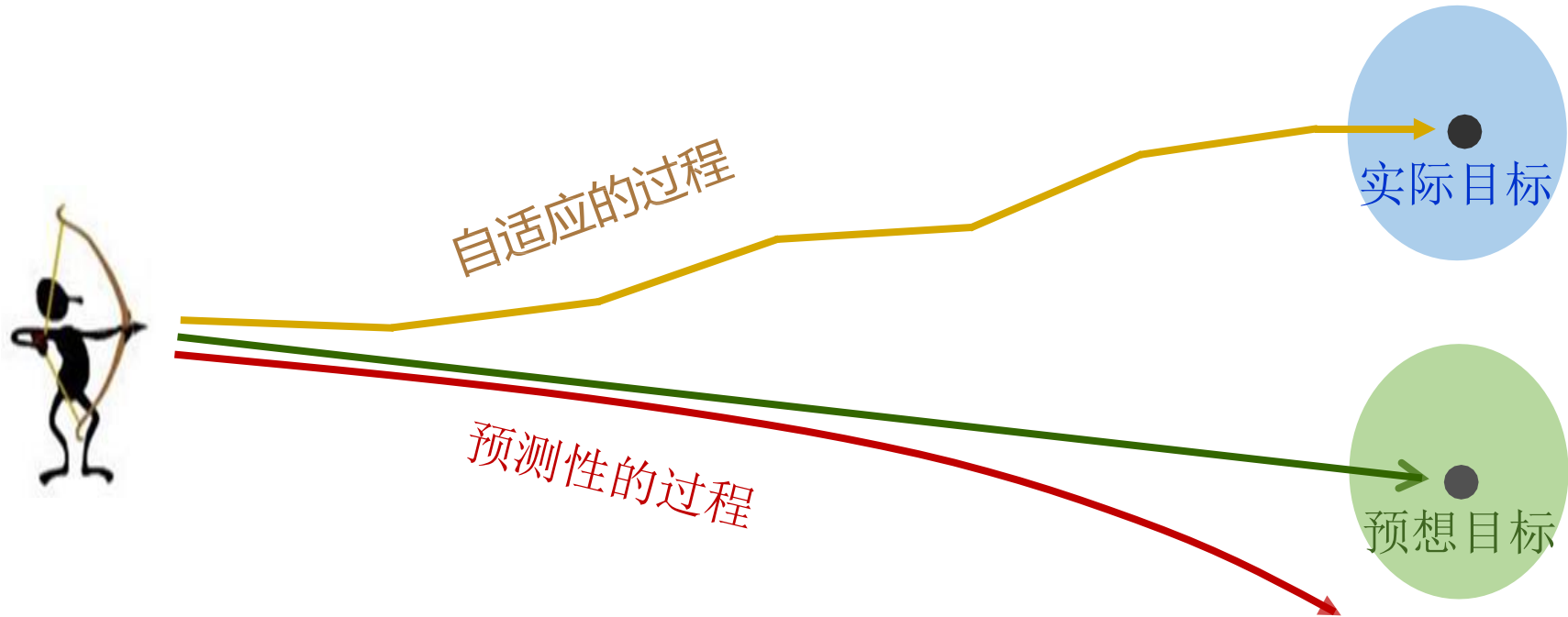


敏捷开发是一种基于更紧密的团队协作、能够有效应对快速变化需求、快速交付高质量软件的迭代和增量的新型软件开发方法。

- 更关注协作
- 更关注质量
- 更关注可工作的产品
- 更关注全才化的专才
- 基于实践而非基于理论



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

敏捷方法的由来

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler

James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

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。



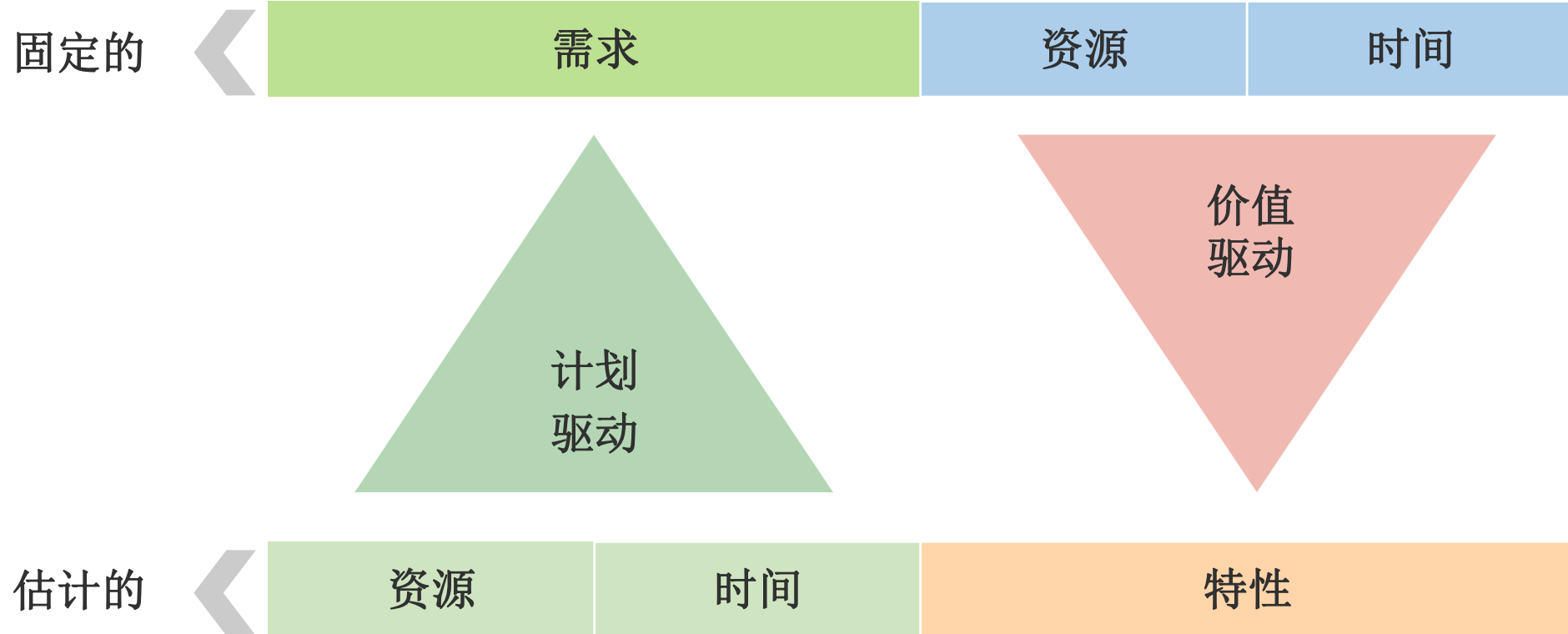
敏捷宣言

- 我们的最高目标是，通过**尽早和持续地交付有价值的软件**来满足客户。
- **欢迎对需求提出变更**——即使是在项目开发后期。要善于利用需求变更，帮助客户获得竞争优势。
- 要不断**交付**可用的软件，**周期**从几周到几个月不等，且**越短越好**。
- 项目过程中，**用户与开发人员**必须**在一起工作**。
- **要善于激励项目人员**，给他们以所需要的环境和支持，并相信他们能够完成任务。
- 无论是团队内还是团队间，最有效的沟通方法是**面对面的交谈**。
- **可用的软件是衡量进度的主要指标**。
- 敏捷过程**提倡可持续的开发速度**，项目方、开发人员和用户应该能够保持恒久稳定的进展速度。
- 坚持不懈地**追求技术卓越和良好设计**，这将提升敏捷能力。
- **要做到简单**，即尽最大可能减少不必要的工作，这是一门艺术。
- 最佳的架构、需求和设计出自于**自组织的团队**。
- 团队要**定期反省**如何能够做到更有效，并**相应地调整**团队的行为。

传统开发 vs 敏捷开发

瀑布模型

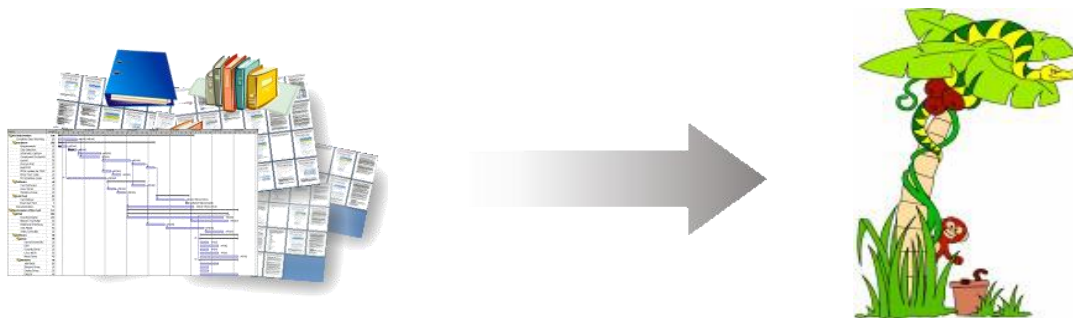
敏捷方法



传统开发 vs 敏捷开发

好的架构（产品）是**长出来的**，而不是设计出来的

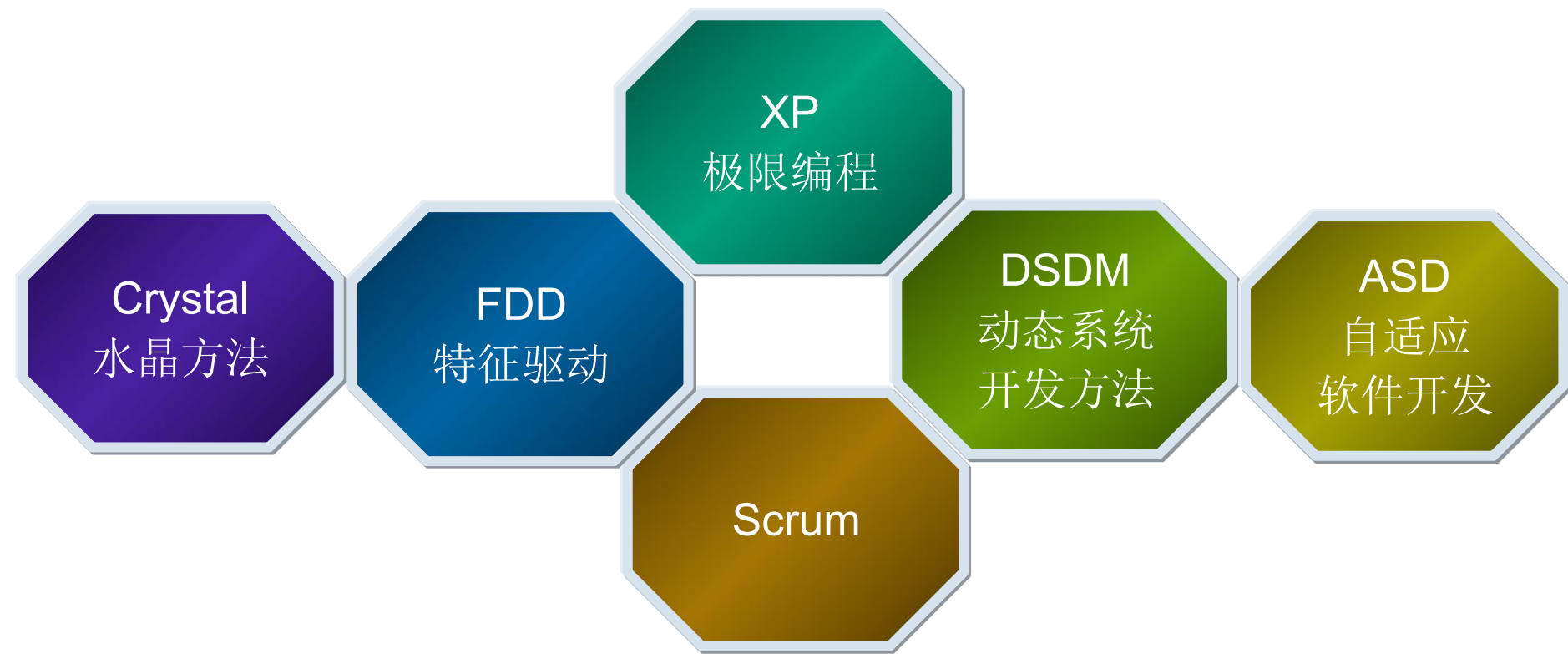
传统开发方法



敏捷开发方法



敏捷开发方法



敏捷开发方法是一组轻量级开发方法的总称，包含很多具体的开发过程和方法，最有影响的两个方法是极限编程（**XP**）和**Scrum**开发方法。

敏捷开发方法



Scrum偏重项目管理

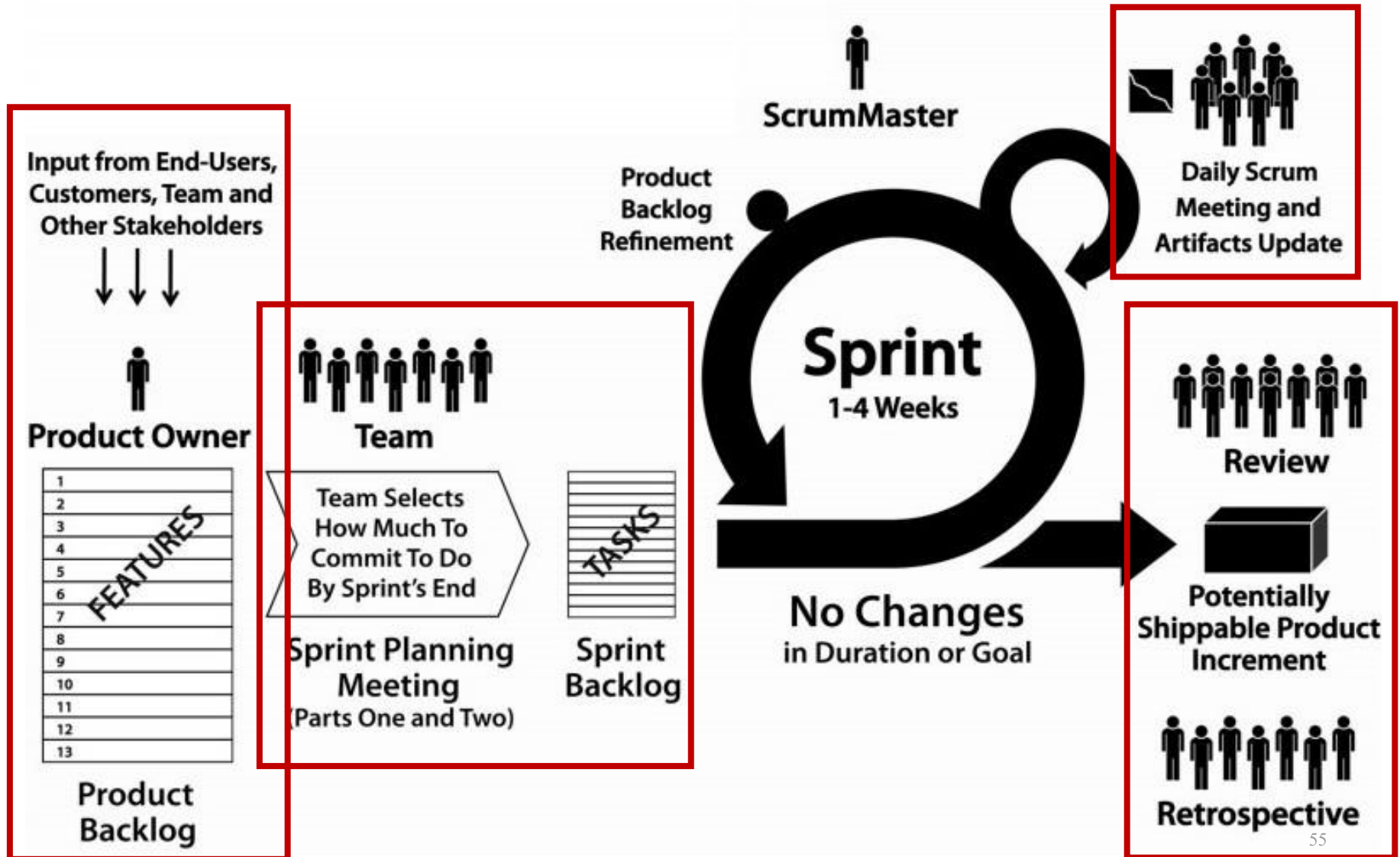
XP偏重编程实践

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。

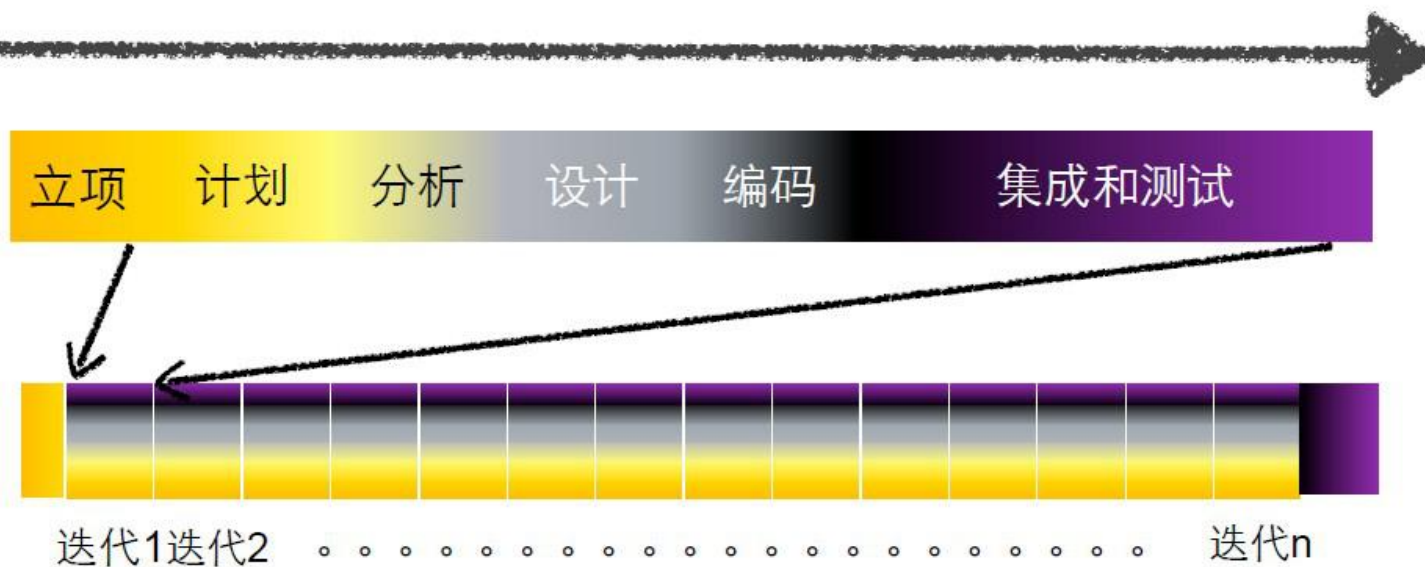


Scrum框架



Scrum迭代开发

迭代开发将整个软件生命周期分成多个小的迭代（一般**2~4**周），每一次迭代就是一个小的瀑布模型，包括需求分析、设计、实现和测试等活动，结束时都要生成一个稳定和被验证过的软件版本。



Scrum迭代开发

迭代开发的关键要点：

- 每一次迭代都建立在稳定的质量基础上，并做为下一轮迭代的基线，整个系统的功能随着迭代稳定地增长和不断完善。
- 每次迭代要邀请用户代表验收，提供需求是否满足的反馈。
- 在一次迭代中，一旦团队作出承诺，就不允许变更交付件和交付日期；如果发生重大变化，产品负责人可以中止当次迭代。
- 在迭代中可能会出现“分解”和“澄清”，但是不允许添加新工作或者对现有的工作进行“实质变更”。
- 对于“分解”和“澄清”，如果存在争议，那么将其认定为变更，放到产品订单中下一次迭代再考虑。

敏捷开发的应用



- **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）