

敏捷开发-入门

The background features several decorative elements: a large blue circle at the top center, a medium purple circle at the top right, a large grey circle at the top right corner, a large dark grey circle on the left side, a large blue circle at the bottom center, a medium purple circle at the bottom right, and a large dark grey circle at the bottom left. Small triangles in blue, grey, and dark grey are scattered around the larger circles.



传统软件开发模型

- 边做边改型
- 瀑布开发模型
- 迭代式开发
- 螺旋开发
-



我要实现XXX功能

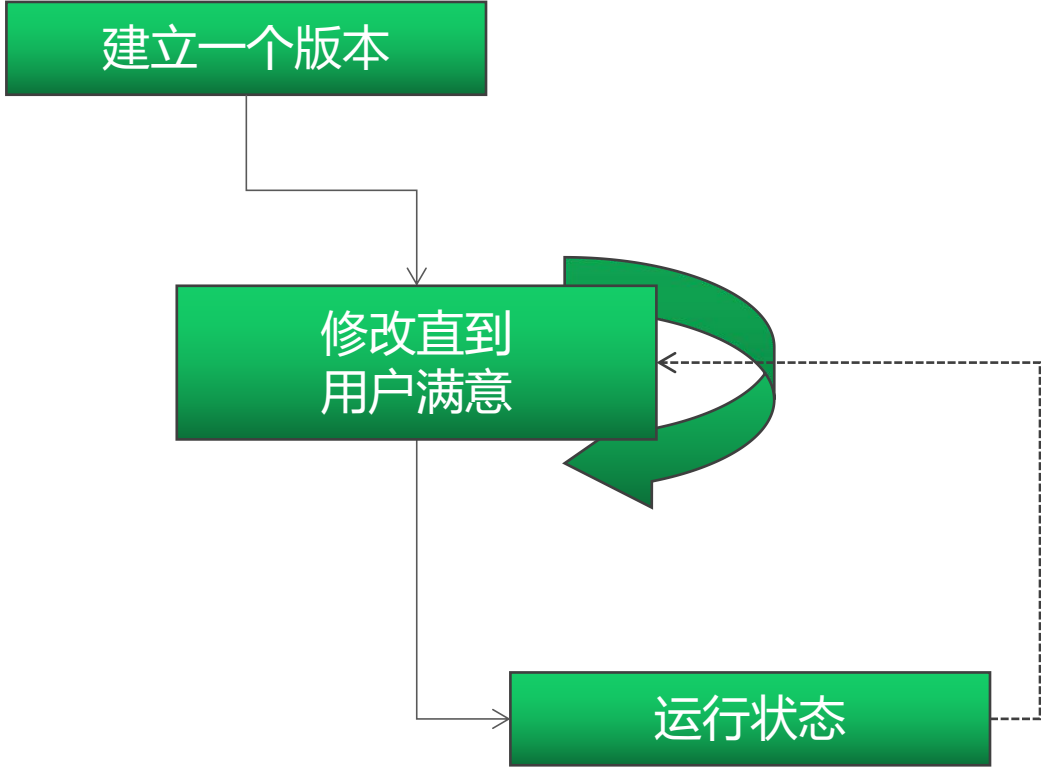
XXX功能还不够，我
需要增加其他功能

这个功能应该还好吧，
我现在就是需要它


这个需求很简单，我
分分钟搞定它！



em...这个，和之前有
冲突，不是很好实现



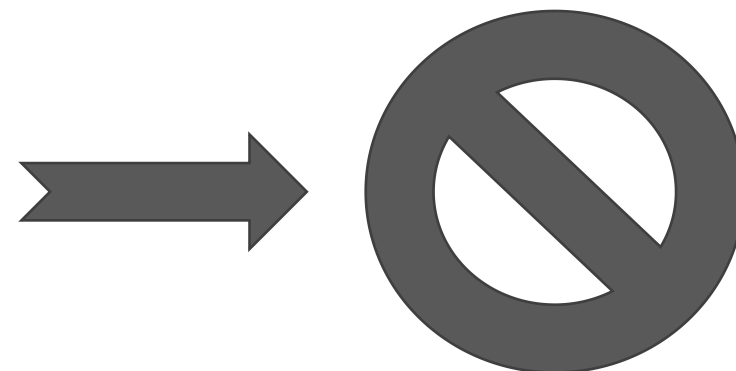
边做边改型



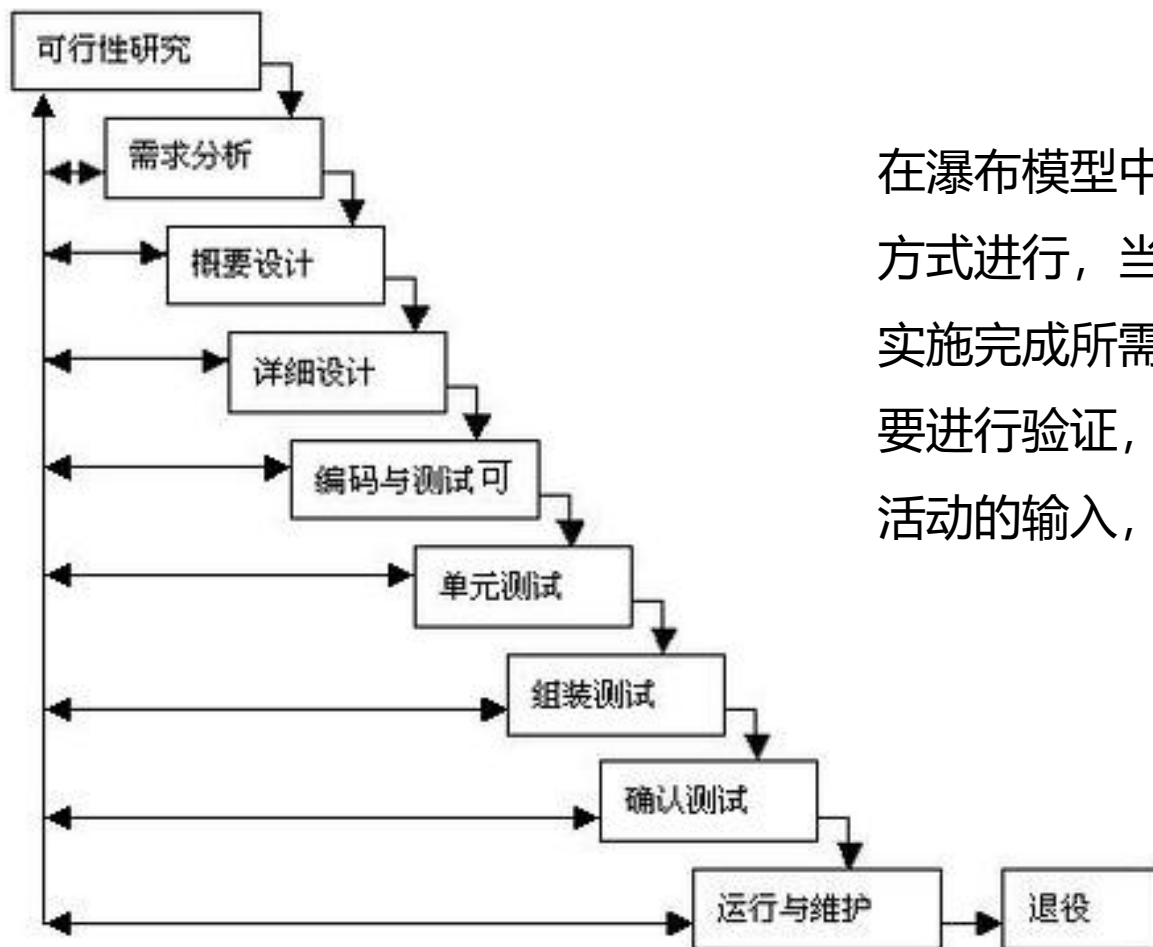
缺少规划和设计环节，软件的结构随着不断的修改越来越糟，导致无法继续修改

忽略需求环节，给软件开发带来很大的风险

没有考虑测试和程序的可维护性，也没有任何文档，软件的维护十分困难



瀑布开发模型



在瀑布模型中，软件开发的各项活动严格按照线性方式进行，当前活动接受上一项活动的工作结果，实施完成所需的工作内容。当前活动的工作结果需要进行验证，如果验证通过，则该结果作为下一项活动的输入，继续进行下一项活动，否则返回修改。





开始准备

配菜, 炒菜

十个菜都一起上桌

淡了加盐, 咸了
加糖, 不能退

先看看菜单呢

点了十个菜

半个小时了...

有些菜味道不
好, 多点了

拉黑





各个阶段的划分完全固定，阶段之间产生大量的文档，极大地增加了工作量

由于开发模型是线性的，用户只有等到整个过程的末期才能见到开发成果，从而增加了开发的风险

早期的错误可能要等到开发后期的测试阶段才能发现，进而带来严重的后果

严格分级导致的自由度降低，项目早期即作出承诺导致对后期需求的变化难以调整



迭代式开发

每次只设计和实现这个产品的一部分,逐步逐步完成的方法叫迭代开发,每次设计和实现一个阶段叫做一个迭代.在迭代式开发方法中,整个开发工作被组织为一系列的短小的、固定长度(如3周)的小项目,被称为一系列的迭代。

每一次迭代都包括了需求分析、设计、实现与测试。采用这种方法,开发工作可以在需求被完整地确定之前启动,并在一次迭代中完成系统的一部分功能或业务逻辑的开发工作,再通过客户的反馈来细化需求,并开始新一轮的迭代。



降低风险

得到早期反馈

持续测试和集成

提高复用性



螺旋开发

“螺旋模型”刚开始规模很小，当项目被定义得更好、更稳定时，逐渐展开。

“螺旋模型”的核心就在于您不需要在刚开始的时候就把所有事情都定义的清清楚楚。您轻松上阵，定义最重要的功能，实现它，然后听取客户的意见，之后再进入到下一个阶段。如此不断轮回重复，直到得到您满意的最终产品。

- (1) 制定计划：确定软件目标，选定实施方案，弄清项目开发的限制条件；
- (2) 风险分析：分析评估所选方案，考虑如何识别和消除风险；
- (3) 实施工程：实施软件开发和验证；
- (4) 客户评估：评价开发工作，提出修正建议，制定下一步计划。


螺旋模型很大程度上是一种风险驱动的方法体系，因为在每个阶段之前及经常发生的循环之前，都必须首先进行风险评估。



敏捷开发模型

怎么理解呢？首先，我们要理解它不是一门技术，它是一种开发方法，也就是一种软件开发的流程，它会指导我们用规定的环节去一步一步完成项目的开发；而这种开发方式的主要驱动核心是人；它采用的是迭代式开发；

敏捷开发是一种以人为核心、迭代、循序渐进的开发方法。在敏捷开发中，软件项目的构建被切分成多个子项目，各个子项目的成果都经过测试，具备集成和可运行的特征。换言之，就是把一个大项目分为多个相互联系，但也可独立运行的小项目，并分别完成，在此过程中软件一直处于可使用状态。



开始准备

配菜, 炒菜

上了两个菜

继续上菜

根据口味调整


看看菜单

选了十个菜品

挺好吃的, 继续做

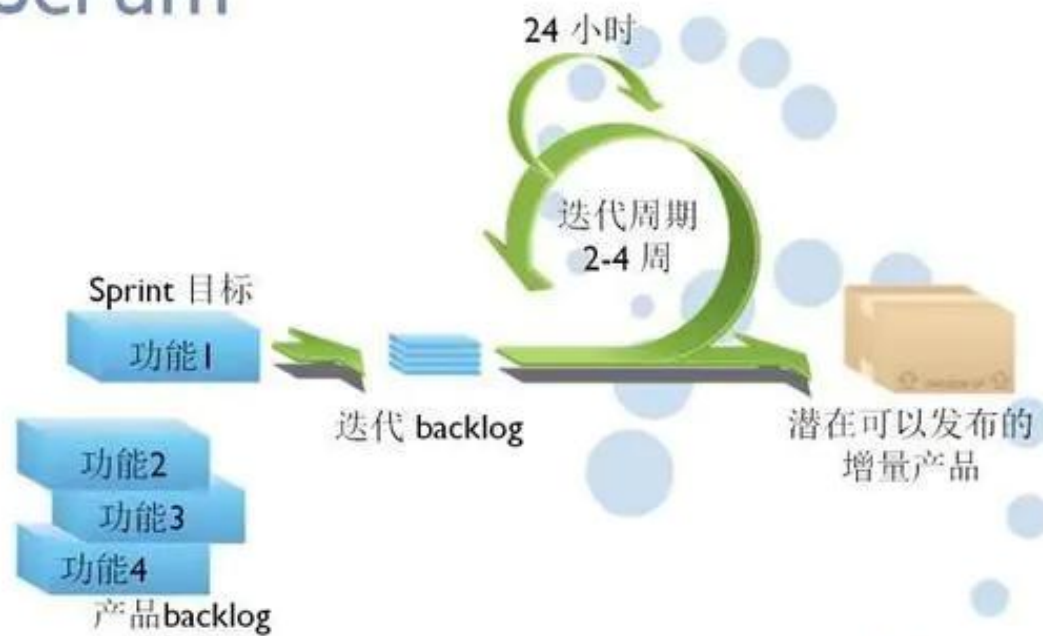
味道差点

吃饱了, 开心



敏捷开发模式

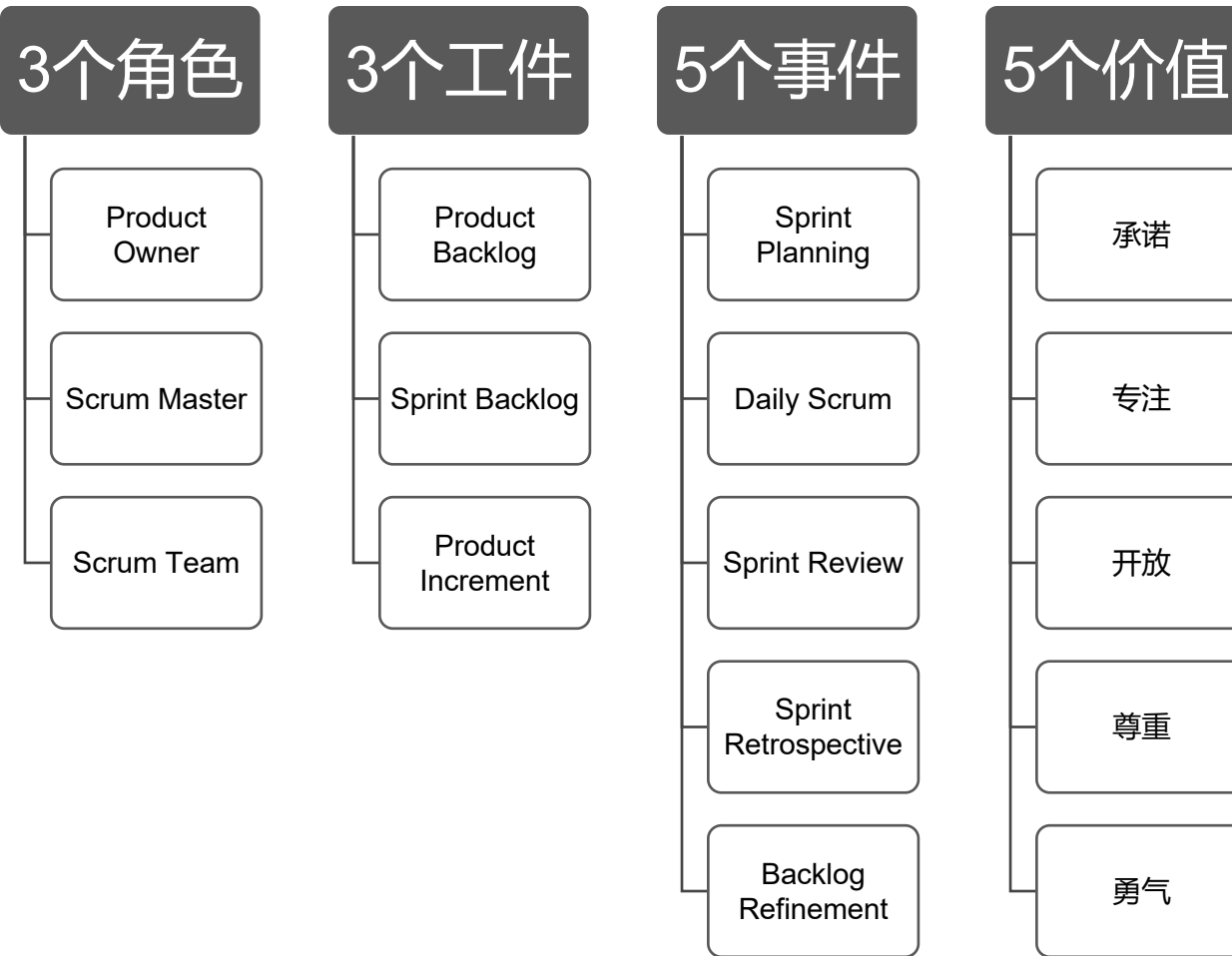
Scrum



实施条件

Scrum不是敏捷的全部，
它只是敏捷的一个落地方法之一。

Scrum就是3355。





Product Owner

- 主要负责确定产品的功能和达到要求的标准
- 有权力接受或拒绝开发团队的工作成果

Scrum Master

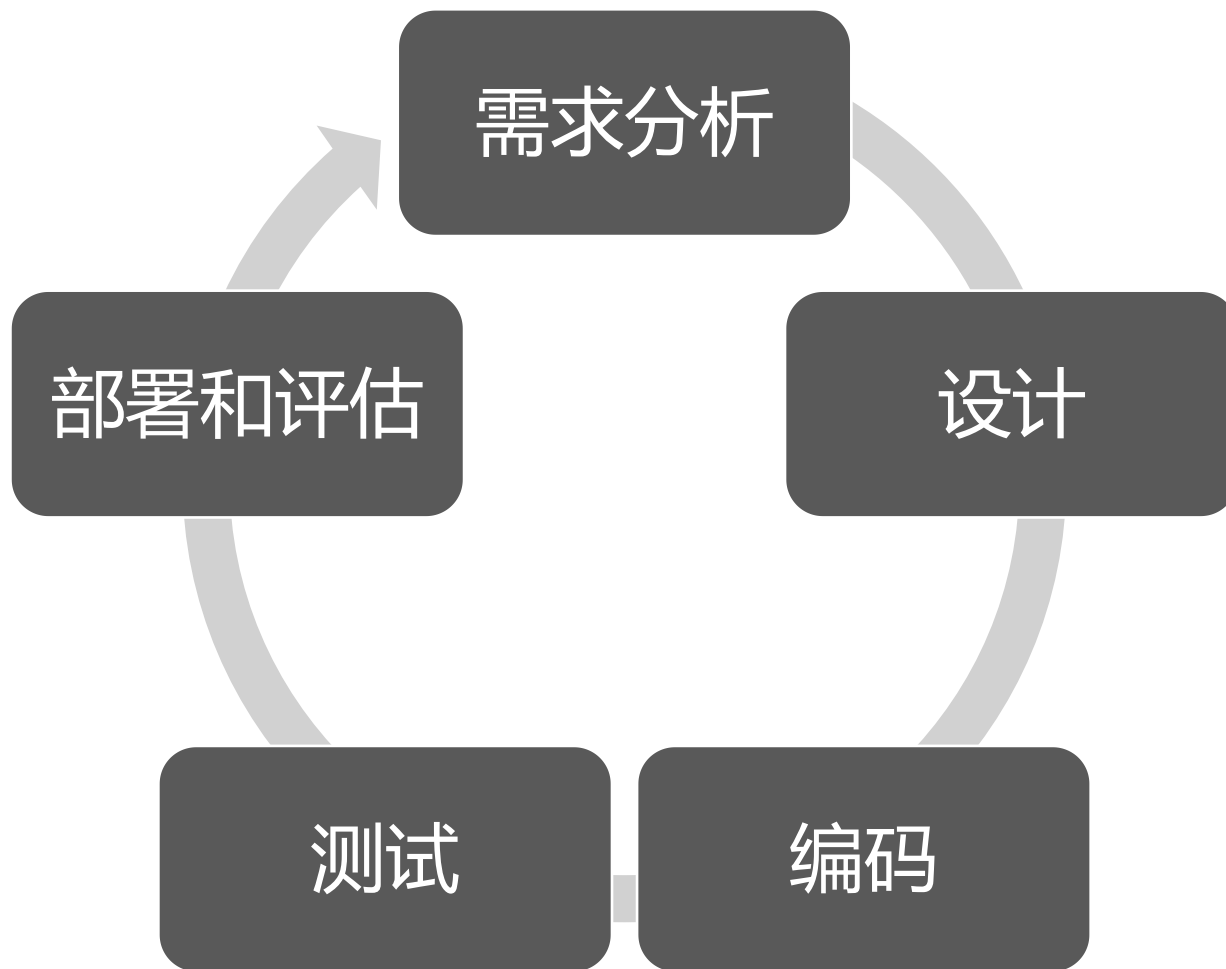
- 使得每一时刻的需求都能明确
- 管理每一次需求变动，变动原因，将变动落实到实处

Scrum Team

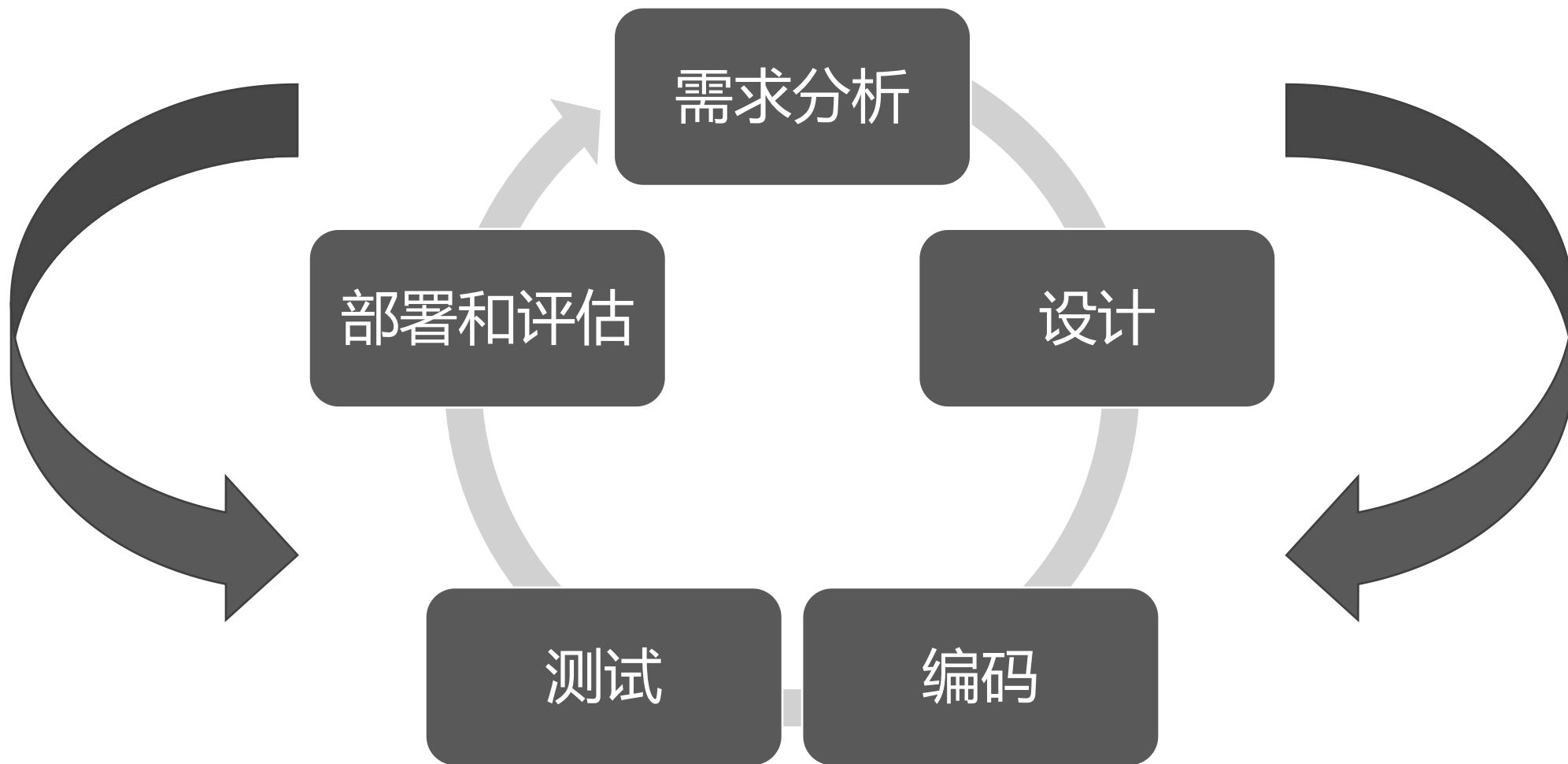
- 根据任务优先级编排任务

Scrum Tester

- 需求明确完之后，即可针对需求编写验收文档
- 测试过程，编写测试案例









实施条件

如果要想很好的实施scrum，通常有两点需要满足：一是团队有三名或以上的研发工程师，二是团队内有一名合适的Scrum Master。当团队内无法找到合适的Scrum Master时，不要轻易推行敏捷。如果你的团队是由新人组成，或者即使有资深员工但是他并不了解或认同敏捷开发的话，那么你需要等待合适的Scrum Master出现。

当你真正实行敏捷开发时，要注意量化衡量团队的执行力的指标：完成度、评估准确度、计划合理度。这是评定整个进度的很重要的指标，也是让迭代更好的进行下去的准则。



敏捷团队

敏捷开发小组主要的工作方式可以归纳为：作为一个整体工作；按短迭代周期工作；每次迭代交付一些成果，关注业务优先级，检查与调整。

敏捷软件开发要注意项目规模，规模增长，团队交流成本就上去了，因此敏捷软件开发暂时适合不是特别大的团队开发，比较适合一个组的团队使用。



敏捷误区

文档能省则省

描述类属性文档、接口说明文档（利用swagger自动生成）。而一些有价值的文档，如设计方案文档、架构体系文档等仍然是必须的。

敏捷的初心

敏捷的初心是建议我们通过一系列方法来让我们的研发工作更加高效、灵活和有序，所以它强调团队成员的能动性和相互之间的协作，也更重视应对变化。

敏捷开发最终定义

敏捷开发以用户的需求进化为核心，采用迭代、循序渐进的方法进行软件开发。

目的

管理好需求，提高开发效率。

The background features several decorative elements: a large pink circle on the left, a large blue circle at the bottom center, a large grey circle at the top right, and a large pink circle at the bottom right. Scattered throughout are smaller triangles in shades of pink, blue, and grey.

感谢观看