# 敏捷开发

**Scrum的工作方式**

1. 产品积压工作
2. 冲刺积压工作
3. 冲刺
4. 发布

**敏捷方法论**包括一组轻量级软件开发方法，它们都允许及时响应客户的需求，即使项目已经在进行过程中了。

**角色和职责**

**产品负责人**

1. 决定构建特性
2. 根据业务价值设定特性的优先级
3. 接受或者拒绝“已完成”的工作

对做出怎样的产品负责

**Scrum主管**

他主要负责在冲刺进行过程中为团队隔离所有外部影响，并且处理团队成员在每日标准会议上提到的各种影响开发的障碍。

对如何完成产品的流程负责。记录每日的会议内容，确保会议人员出席等。

**开发团队**

每个人都能胜任，同时又有特别擅长的领域，技能重叠防止出现筒仓现象。

软件测试人员负责保证开发软件质量。即核实软件产品是否按照要求工作。

**工件**

文档对于项目依然十分重要，但是它不可取代可工作的软件和沟通本身。即定要出现可执行的代码和实验成品，而不是仅依靠文档说明功能，不可“大设计优先”的错误理念。

**Scrum面板**

日常的核心工作区域。实物（图1）以及数字化面板。实物可以更加直观体现进度，因此实物是必须的，虽然数字化也有帮助，但其仍只是辅助工具（无法那么直观）。

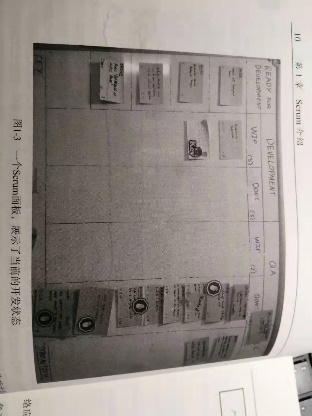


图.1 Scrum实物面板

1. **卡片**

* **组成的层次结构**
* **产品**

食物链顶端是要构建的软件产品。通常开发团队每次只完成一个产品，有时候则是交付多个产品

* **发布**

一个发布就是一个特定版本的软件。有的发布只是为了解决若干的缺陷，有的发布则是为关键用户提供有价值的新特性，还有的发布是为了提供测试版本让用户尝鲜和反馈。网络应用程序通常只会在所有发布前部署一次，版本的变更也不会很明显。Scrum在每个冲刺后能很自然地通过专注于交付可工作的软件来支持这种发布模式。

* + **最小可行发布**

第一个发布可以是一个最小可行发布，它包括了能够满足用户最基本需求的一组特性。自筹基金和获得反馈及时调整软件后续发展方向。软件不断进化的前提就是软件必须要变化。

* **特性**

**最小可售特性**用来界定一个发布需要的集合。制定特性需要分为三类：必须的、可选的和想要的。三个分类是互斥的，用来划分优先级别。开发团队必须首先完成所有必须的特性，然后才是可选的特性，只有在时间允许的情况下才可以触及想要的特性。这些分类和特性本身总是可变的，当团队要切换工作焦点的时候，可以在任意时间中止它们，对其重新排序，交换其顺序以及废弃他们。

* **用户故事**
* **任务**
* **技术债务**
* **缺陷**
* **让卡片意图更明确**

1. **泳道**
2. **技术债务**
3. **数字Scrum面板**
4. **完成的定义**