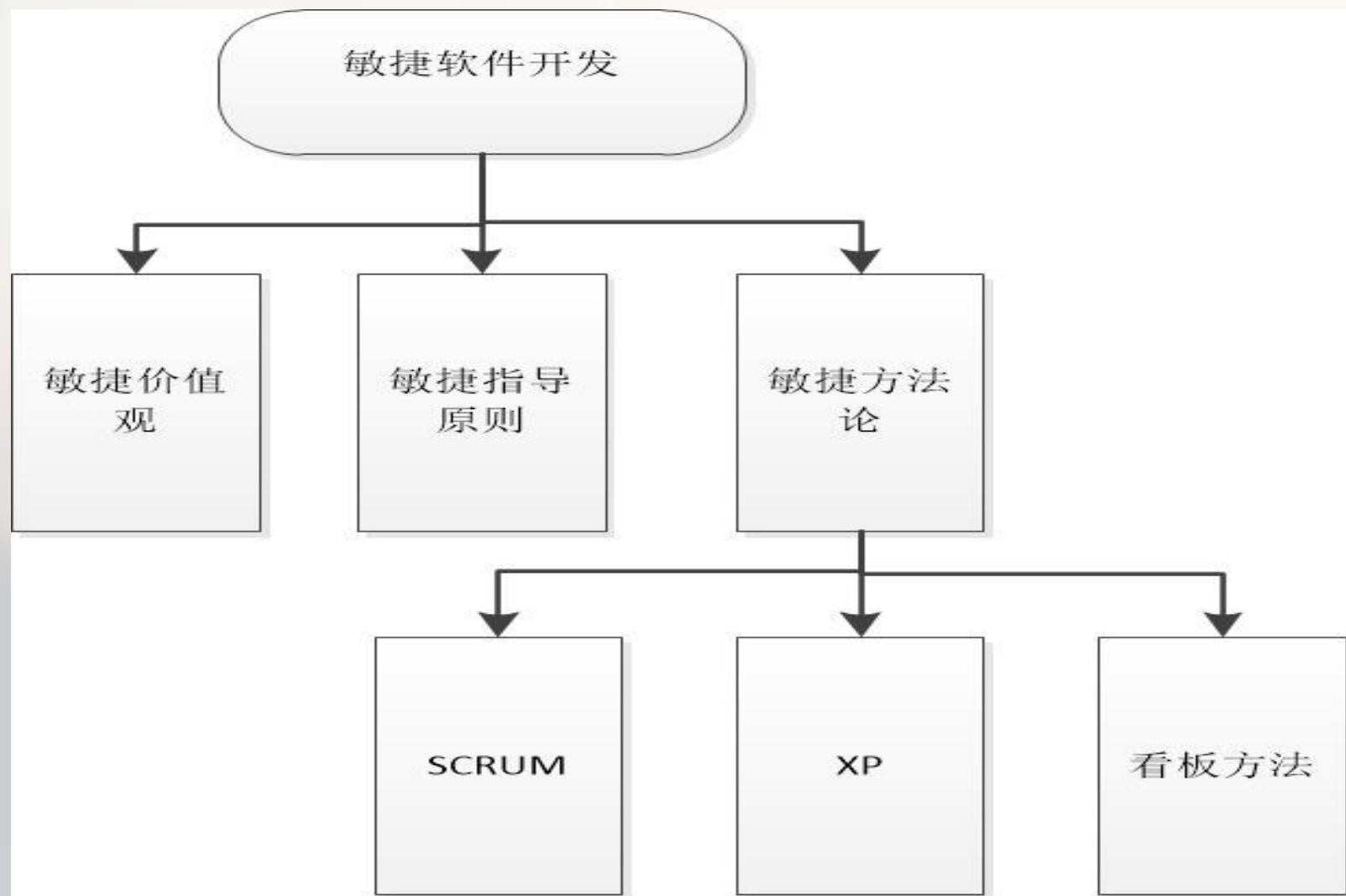
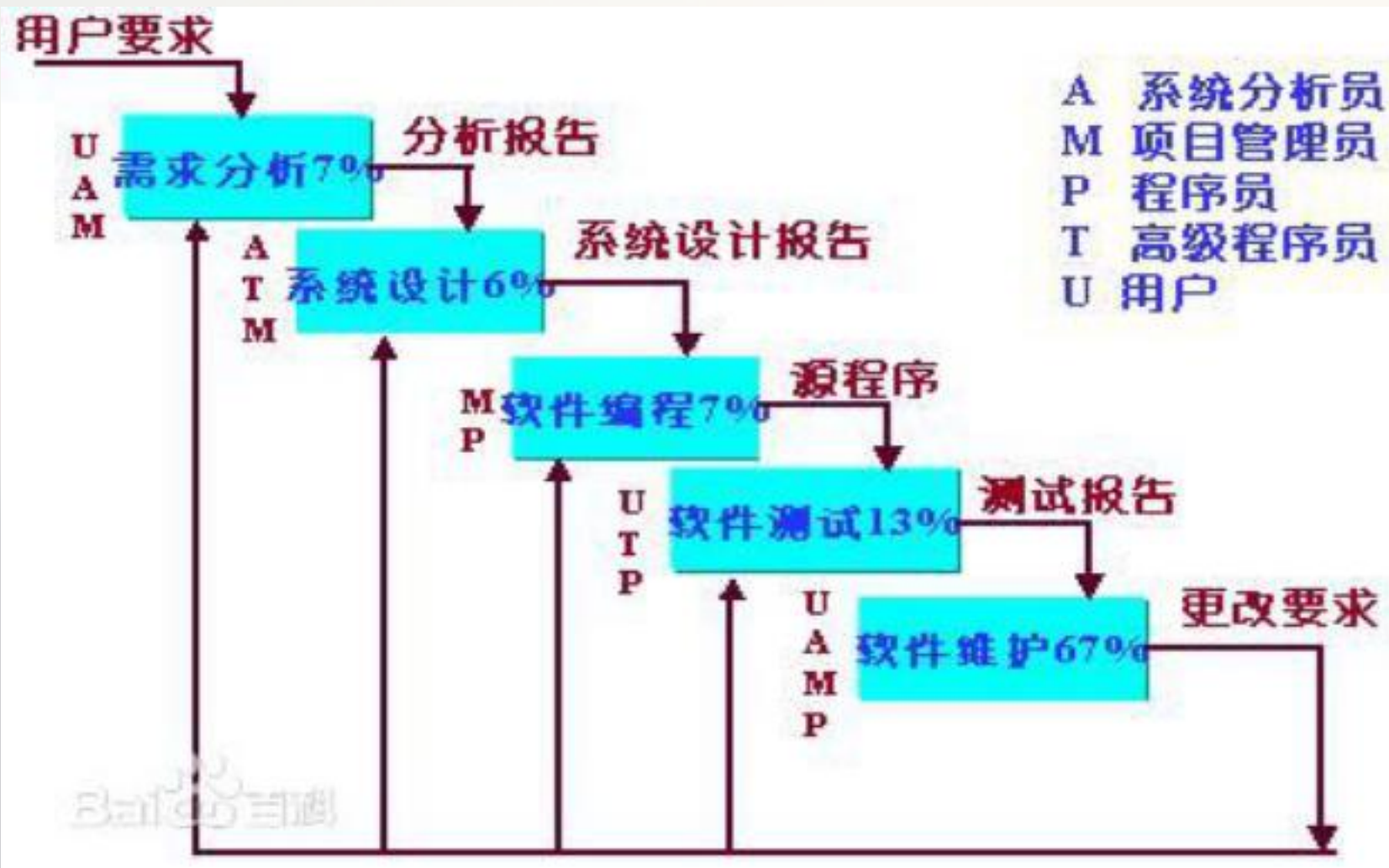


敏捷软件开发

主要内容



传统瀑布模型



敏捷历史

1. 迭代式开发。
2. 增量交付
3. 开发团队和用户反馈推动产品开发
4. 持续集成。
5. 开发团队自我管理。

敏捷价值观

我们通过身体力行和帮助他人来揭示更好的软件开发方式。
经由这项工作，我们形成了如下价值观：

个体与交互 胜过 过程和工具

可以工作的软件 胜过 面面俱到的文档

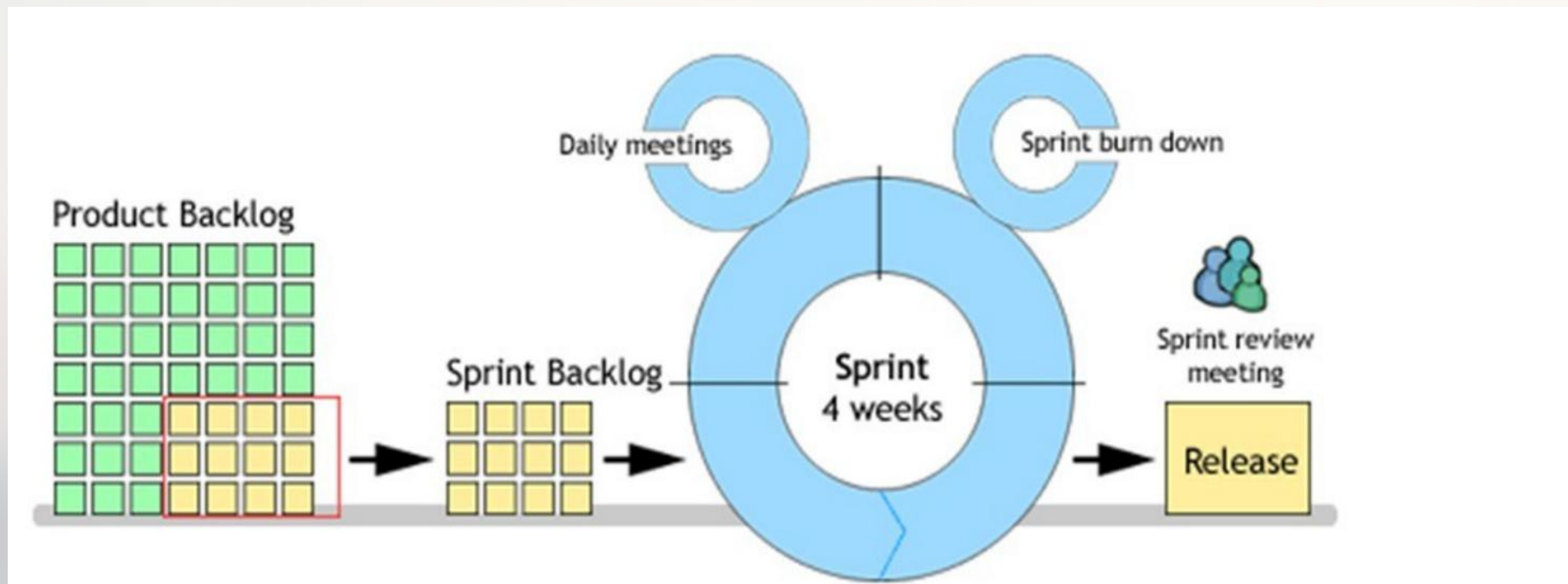
客户协作 胜过 合同谈判

响应变化 胜过 遵循计划

敏捷指导原则

1. 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。
2. 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。
3. 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。
4. 项目过程中，业务人员与开发人员必须在一起工作。
5. 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。
6. 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。
7. 可用的软件是衡量进度的主要指标。
8. 敏捷过程提倡可持续的开发。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。
9. 对技术的精益求精以及对设计的不断完善将提升敏捷性。
10. 要做到简洁，即尽最大可能减少不必要的工作。这是一门艺术。
11. 最佳的架构、需求和设计出自于自组织的团队。
12. 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。

SCRUM管理实践



SCRUM管理实践

迭代周期: Sprint 1-4周

三个角色: 产品负责人、开发团队、Scrum Master

三个工件: Product Backlog、SPRINT BACKLOG、燃尽图

五个活动: 产品待办事项列表梳理、Sprint计划会议、每日Scrum会议、Sprint评审会议、Sprint回顾会议

制定完成的定义

SCRUM管理实践-四大支柱

迭代开发

增量交付

自组织团队

高优先级的需求驱动

SCRUM之用户故事--是什么？



用户故事描述了对用户，系统或者软件购买者有价值的功能。

目的：提示开发人员要与需求方进行沟通。

使用方式：卡片（Card） 交谈（Conversation） 确认（Confirmation）

SCRUM之用户故事编写格式

1. 角色：谁要使用这个功能。
2. 活动：需要完成什么样的功能。
3. 商业价值：为什么需要这个功能，这个功能带来什么样的价值。

作为一个“网站管理员”，我想要“统计每天有多少人访问了我的网站”，以便于“我的赞助商了解我的网站会给他们带来什么收益”

格式：角色+实现的功能 例：用户输入用户名密码后可看到主页信息。

程序可用java语言编写。
前端必须用js编写。

SCRUM之用户故事---怎么拆分？

故事拆分原则：

独立的

可讨论的

对用户有价值的

可估计的

尽量小的

可测试的

其实很简单，把要做的功能拆分成一个个的小功能，小功能再拆成小功能，知道符合以上原则

SCRUM之用户故事---怎么拆分？

摇一摇例子：

用户可以在后台添加摇一摇活动

- 用户输入用户名密码可以跳转到摇一摇台主页

- 当用户输错三次用户名密码时应该填写验证码

- 当用户登录到主页时需要根据用户权限查看对应的链接信息

- 用户需要选择活动类型的模板

- 用户可以添加摇奖的时间段

- 用户可以添加活动的提示信息

- 用户可以添加奖品信息

一个详细的用户故事例子

当用户输入手机号等相关信息完成注册。

Task: 1.注册页面前端开发。

2.数据库表结构设计。

3.后台逻辑开发。

Note: 1.用户名必须是英文字母开头。

2.密码不能超过36位置

估点：3点

优先级：高

SCRUM与精益创业的结合

没钱没人不能创业？

先市场中投入一个极简的原型产品，然后通过不断的学习和有价值的用户反馈，对产品进行快速迭代优化，以期适应市场（小步试错）

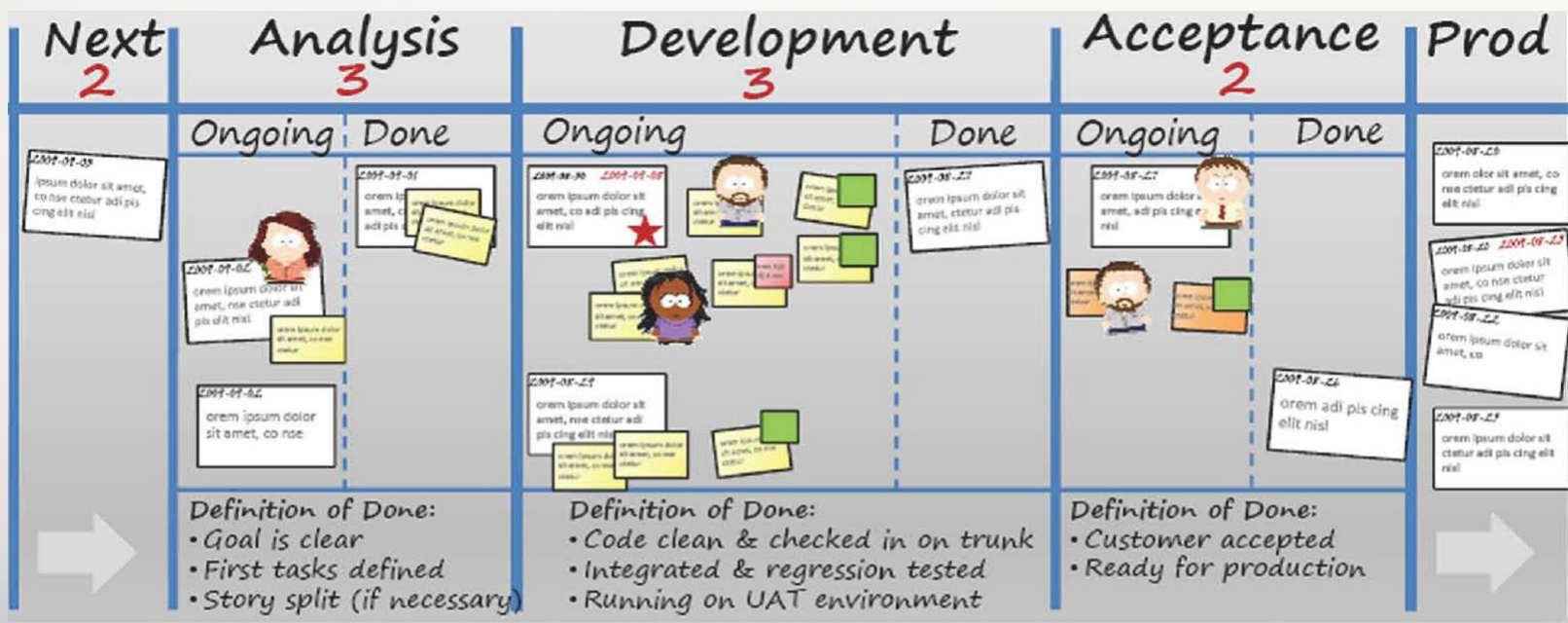
三大法宝：“最小可用品”、“客户反馈”、“快速迭代”

具体步骤：开发-测量-认知

XP工程实践

- *结对编程
- *可持续的节奏
- *代码集体所有
- *编码规范
- *简单设计
- *测试驱动开发
- *重构
- *持续集成
- *现场客户
- *计划游戏

精益软件开发--看板方法



Feature / story



Task / defect



What to pull first

1. Panic features ★★ (should be swarmed and kept moving. Interrupt other work and break WIP limits as necessary)
2. Priority features ★
3. Hard deadline features (only if deadline is at risk)
4. Oldest features

疑惑

- 1.敏捷能加快软件开发进度？
- 2.敏捷不考虑未来需求？
- 3.敏捷不适合大型软件开发？
- 4.敏捷不适合大型组织使用？
- 5.什么时候不应该使用敏捷开发？哪些项目不适合敏捷软件开发？

Ending