虚拟案例 - 超市管理系统

项目管理的大致过程

汪志成 (wangzhicheng@admaster.com.cn)

背景假设

时间	• 十年前 • 国内大多数超市都尚未电子化
我们	• 已有一个成功案例 • 想要拓展另外N个
客户	• 沃尔牛老板 • 传统经营方式 • 颇具规模
竞争者	• ICM，一家国外的大型ERP企业，沃尔玛系统提供商 • 超软，一个小工作室，想要接下这个项目作为第一单

复习：戴明环 - PDCA

计划阶段

产品经理主导的阶段

意向与调研

- 意向接触 - 沃尔牛老板找到了我们，但对是否合作举棋不定
- 制作原型 - 用于和客户对系统的未来达成共识
- 需求调研 - 首要做什么，次要做什么，不做什么
- 商务谈判 - 你的筹码与报价
- Spike - 我们可能涉及到的技术，无论能否拿下，都提前研究

执行阶段

项目经理和架构师主导的阶段

执行阶段（一）

产品经理

- 用户分析：知识水平、技能、工作习惯、好恶等
- 业务分析：未来这个行业会是什么样，电子化将如何改变这个行业，将如何改变这家企业
- 竞品分析：对手的产品定位与我们有什么不同，他们的优缺点是什么，要向他们学习什么
- 设计产品路线图：定义“在可预见的时间内做什么、不做什么”

项目经理

- 制定工作分解表：能细分为哪些工作，这些工作之间的依赖关系如何
- 识别关键路径：各条工作线是否平衡，有没有哪个人频繁出现在关键路径上
- 风险识别与预案：项目四要素中可能有哪些风险，你将如何应对，备胎在哪里
- 准备资源：你需要哪些资源，你现有哪些资源，资源需要在什么阶段到位，要为这些资源申请多少预算

架构师

- 有哪些非功能需求？
- 在几大架构指标中如何权衡：性能，可靠性，安全性，延展性，扩展性，定制性，可维护性，客户体验，需求响应速度
- 建立架构隐喻：让程序员、项目经理、产品经理甚至客户代表了解系统架构
- 设计架构路线图：定义“在可预见的时间内这片‘技术森林’将变成什么样”

执行阶段（二）

分析师

- 和产品经理共同建立需求模型：用例图
- 建立分析模型：活动图、状态图、（分析）类图
- 建立设计模型：序列图、合作图、（实现）类图

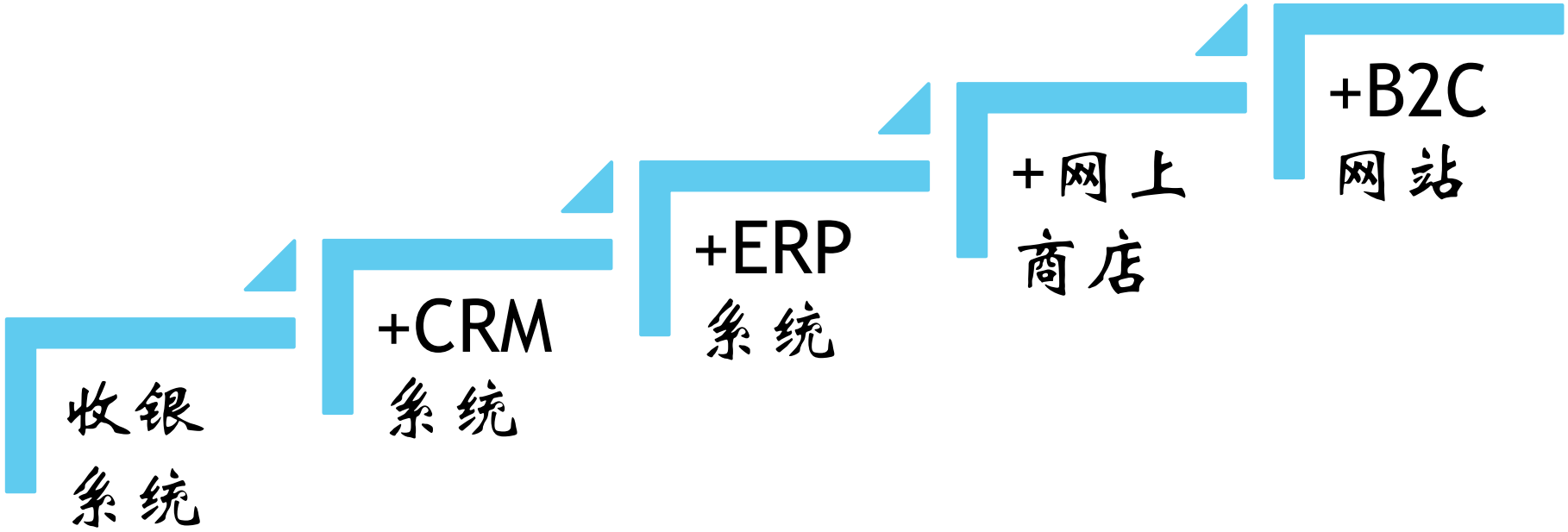
技术专家

- 对相关技术进行预先研究：完善基线中的“难度”部分
- 帮助项目经理评估风险：完善基线中的“风险”部分
- 识别、设计核心算法等模块

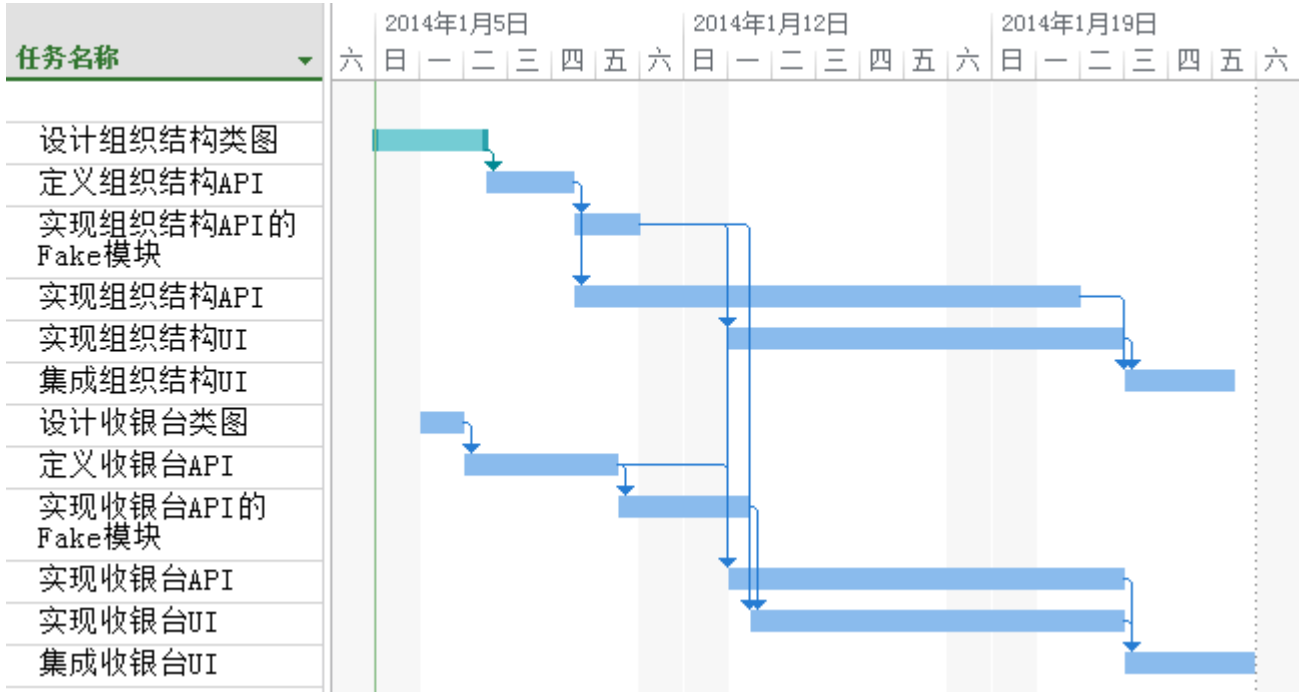
程序员

- 详细设计：对分析师的成果进一步细化，补充辅助类
- 技术评估：对具体实现时用到的技术进行选型和试验
- 编程实现：数据库设计、编码、单元测试、行进中代码复查

工件 - 产品路线图



工件 - 项目计划



Fake模块是指具有指定API接口的伪实现。
它不关心背后的处理逻辑，只负责接受预先定义的参数，并给出预先定义好的结果，
定义良好的API及其Fake模块可以有效提高开发的并行程度，缩短工期

汪志成 (wangzhicheng@admaster.com.cn)

工件 - 用户故事

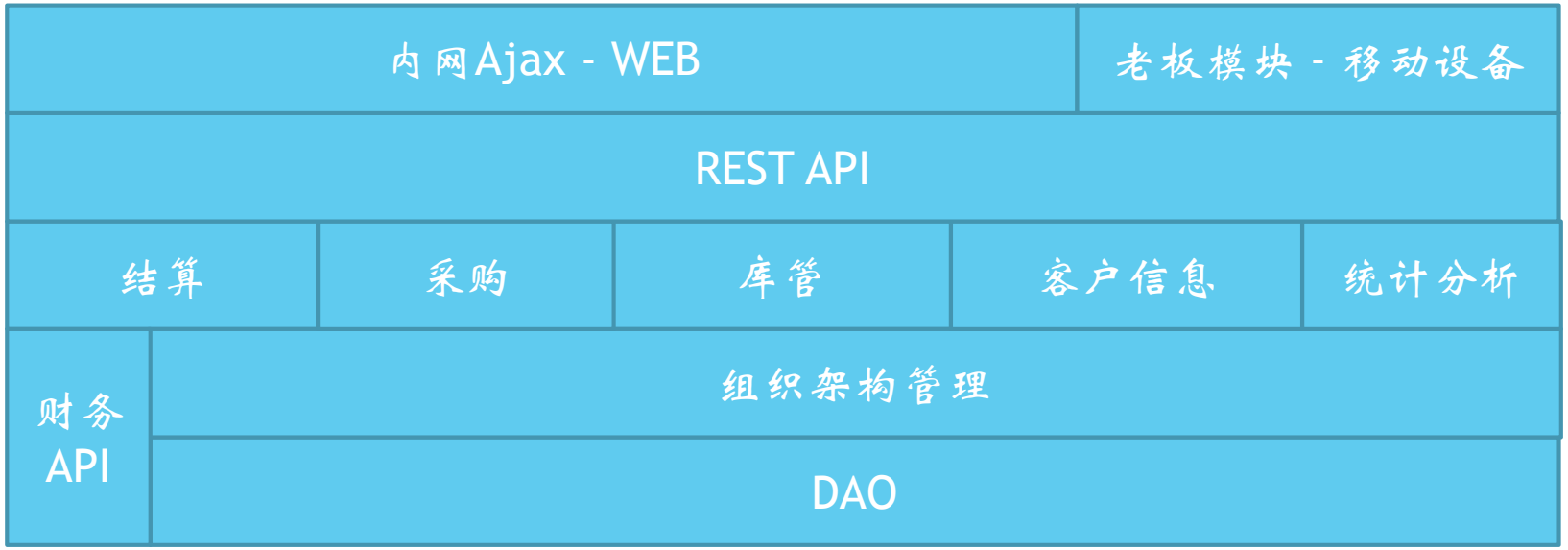
- 李女士是一位中年下岗职工，住在五公里外的红星小区
- 她初中文化，经简单培训，输入字母和拼音没问题，但是打字速度比较慢
- 她在沃尔牛超市担任库管，负责出货进货
- 她认真负责，手脚麻利，从未出错，是超市优秀职工
- 超市实行电子化管理之后，她每天来到库房就先打开电脑
- 这台电脑与系统中的其他电脑连成一个局域网
- 有人送货来，她逐件进行“条码扫描，输入一个数量（默认为1件），按回车”，再刷一下送货人的身份磁卡。就完成了一个入库。
- 有人来提货，她先把提货单号输入电脑，然后比对显示出的提货人身份，按照规定的数量取出货物，逐件扫描登记后，回车。就完成了一次出库。

工件 - 需求基线

编号	说明	重要性	难度	风险	优先级
1	计算应收款项与找零	5	4	3	5
2	管理库存	5	4	2	5
3	会员卡	5	5	4	5
4	打印小票	4	4	1	4
5	刷卡消费	4	5	5	3
6	客服接入	3	4	4	2
7	考勤系统	1	1	1	1

重要性：客户对于这项需求的要求有多么急切，如果不实现对客户影响如何
难　度：这项需求涉及到的技术对当前团队来说难度有多大
风　险：实现这项需求时的不可控因素有多少，有没有备胎
优先级：综合考虑上述三项而确定出的一个数字，用于划分不同的迭代周期

工件 - 系统架构



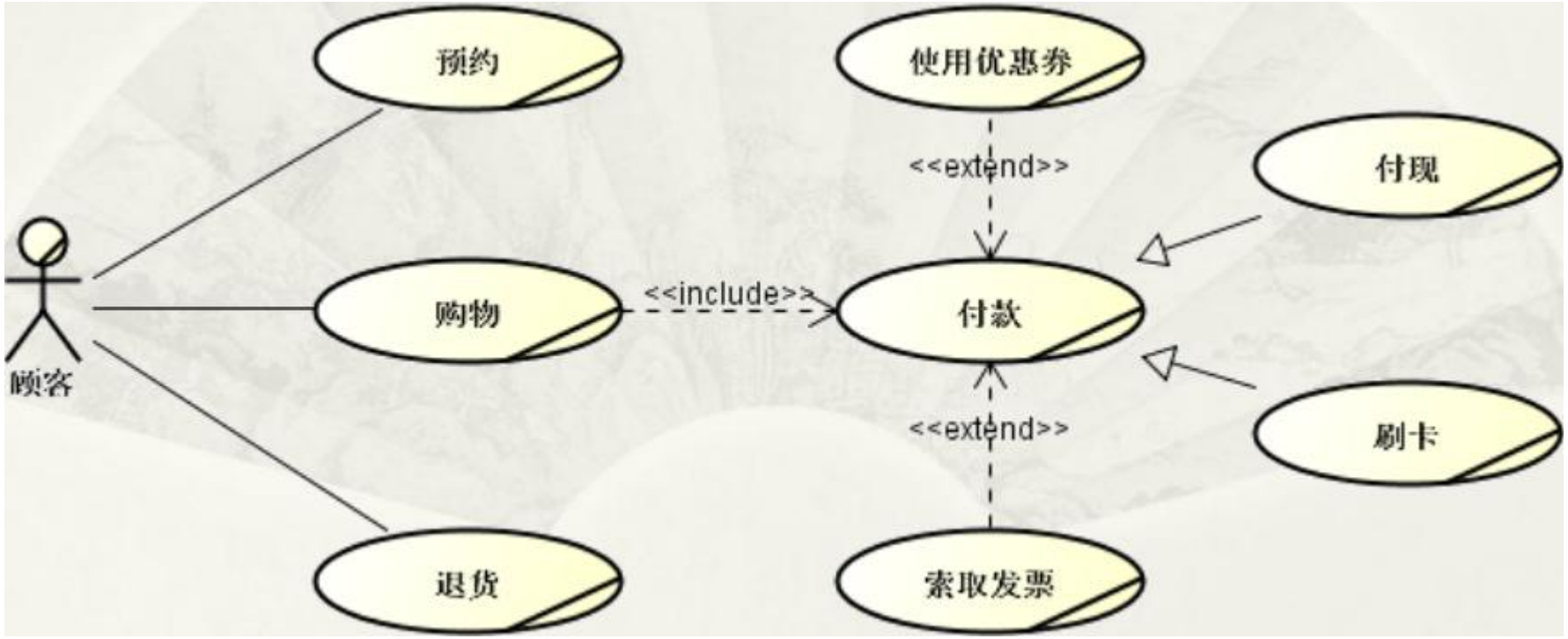
- 系统架构用于体现顶级的模块划分和依赖关系
- 上级模块必须单向依赖于下级模块，防止循环依赖
- 这种图是最浅显的部分，实际上还包括一系列组件图和部署图

工件 - 技术评估报告

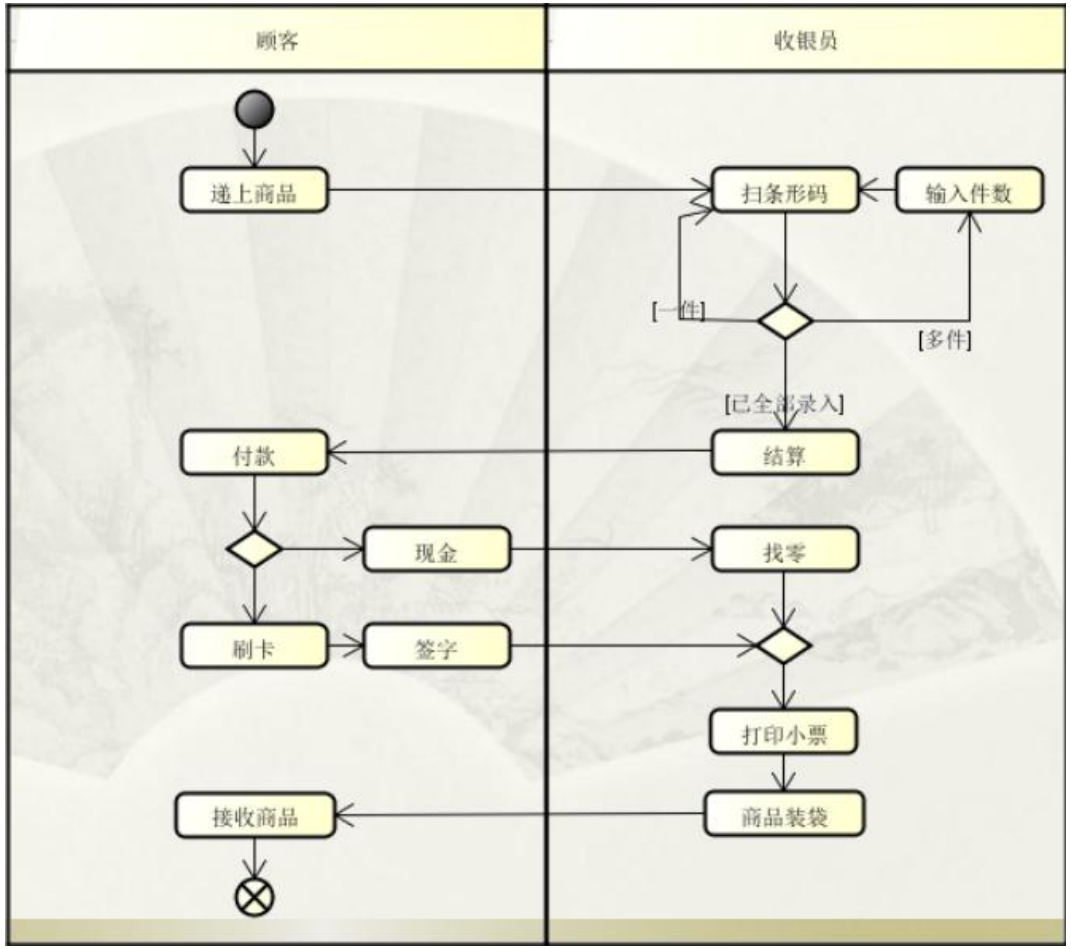
对比指标	jQuery	AngularJS
对系统长期演化的适应能力	一般	很好
社区支持力度	很强	英文很强，中文一般
本团队掌握程度	四人精通，其余熟练	两人熟练，其余陌生
技术长期前景	平稳，渐趋没落	爆发阶段，候选标准
……		

需要评估的指标很多，可以由多人从不同角度评估，最后由项目经理拍板。
不但要通过公开的资料调查，还要进行必要的技术实验来得出一些量化指标。

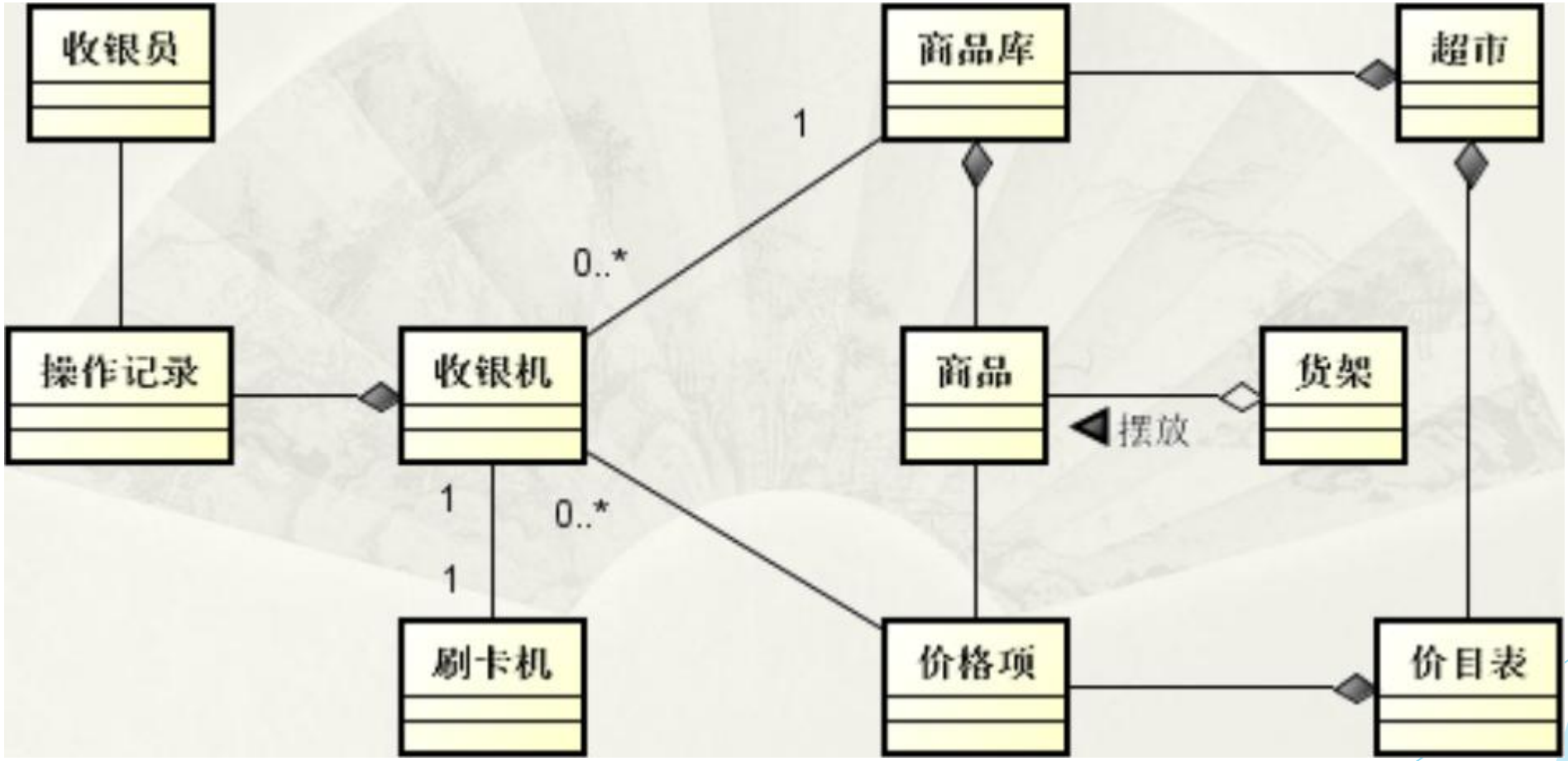
工件 - 用例图



工件 - 活动图


顾客
收银员
递上商品
扫条形码
输入件数
[一件]
[多件]
[已全部录入]
付款
结算
现金
找零
刷卡
签字
打印小票
接收商品
商品装袋

工件 - 类图



工件 - 开发日志

[2010-10-20 15:30:10]

- 一个Ajax错误死活解决不了，本页改用静态网页暂时替代，项目结束后再研究
- 发现momentjs真心强大，对各类日期处理支持很好，有空了给伙计们分享下
- 今天丢人了，竟然在条件语句里写了句if (a = 1).....

[2010-10-22 9:30:10]

- 晨会上迎来一个需求变更，以前误解了客户需求，我的模块看来得重写了
- 对当前项目打了一个版本标签“准备重写，注意：这个版本尚有很多bug”

工件 - 测试用例 - 收银台

- 用户概况
  - 张三(e10001)：收银员，8点到11点操作收银机1001
  - 李四(e10002)：收银员，11点到16点操作收银机1001
- 测试目的：收银流程，收银机交接流程
- 测试计划
  - 张三打开收银机，刷员工卡完成登录
  - 现金支付（有找零），详细步骤如下.....（根据活动图设计一系列具体操作）
  - 现金+打折卡，详细步骤如下.....
  - 刷卡支付，详细步骤如下.....
  - 刷卡支付+打折卡，详细步骤如下.....
  - 张三登出本收银机，一分钟后，李四登入
  - .....

检查阶段

测试和技术专家主导的阶段

集成测试

- 人工测试（与执行有一定的重叠）
  - 需求完成度（往往是major级别的bug）
  - 用户友好度
  - 界面瑕疵
  - 综合感受
- 自动化测试
  - API测试（从设计阶段一直持续到发布）
  - 脚本化的场景测试
  - 压力测试
  - 安全检测

代码复查

- 检查典型的设计与编码错误
  - 与执行阶段的“行进间代码复查”相比，它更注重共性
  - 程序员A犯过这个错误，程序员B有没有犯？
  - 模块A的接口设计非常不友好，模块B的接口怎么样？
  - 程序员A使用了一个10表大join，程序员B有没有这样的问题？
  - 某个关键算法没有单元测试，其他的关键算法怎么样？
- 查找优化与复用的契机
  - A模块和B模块都有这项功能，能否提取为公共模块？
  - 是现在提取还是等将来再提取？
  - 压力测试的结果是否具有典型性，是否值得现在就冒险优化？

总结提高阶段

QA主导的阶段

流程复查

有哪些问题？
- 源于流程？
- 源于人？
- 源于技术？

遗漏了什么？
- 没有提前申请资源？
- 没有预见到风险？

多余了什么？
- 正确的无用功？
- 不必要的与会者？

新的流程！
- 实用
- 贴身
- 敏捷化！

研发规范

有哪些问题？

- 源于规范？
- 源于人？
- 源于技术？

遗漏了什么？

- 未能防范典型错误？
- 内容表述不够清晰？

多余了什么？

- 有没有官僚条款？
- 可否通过工具保证？

新的规范！

- 准确
- 实用
- 简短
- 敏捷化！

可复用清单

管理	项目组织结构
	迭代周期安排
	可共享资源
需求	调研技巧
	分析技巧
	基线决策技巧
设计	架构原则
	公共类体系
	技术评估结果
实现	公共库
	第三方框架
	自有框架

分享完毕，谢谢观赏！