# 需求阶段

## 要做什么(关注点，好的问题定义)

1. 动员大会
2. 获取客户需求
3. 将得到的客户需求记录成文
4. 评审需求文档，客户签字确认

## 怎么做(会用到哪些工具、方法、原则)

1. 如何获取客户需求
2. 首先找客户 确定项目的类型(进销存|OA|CRM|…)、规模与范围
3. 自己分析 将接收到的需求，根据项目的类型，分割成若干个独立模块(尽可能的断耦合)，每个模块之间有什么约束关系(因为每个项目类型都有其固定的内容)
4. 自己分析 得出客户术语，要名正才能言顺，而且最好确认术语唯一性，可避免以后沟通所产生的问题，后期要不断的补充
5. 找客户 确定第一步、第二步、第三步中分析出来的结果是否合理，以及确认模块之间的关系、每个模块的意义，得出每个模块要完成的功能。
6. 自己分析 在确定每个模块之间的关系后，再针对每个模块独立分析，每个模块应该要完成的功能
7. 找客户 再次确定每个模块要完成的功能
8. 然后针对这个模块，在细分成若干个子模块，再从第一步开始，直到细化到页面功能时，才继续下面的步骤
9. 在具体实现上需诱导客户向我们希望的方向进行，比如说在这个页面上要完成这个功能的时候，在具体实现上可诱导客户。

需要注意事项：

1. 每次与客户沟通时，要让客户知道什么时间、地点、人员、开会的议题(要排好优先级，并标注)。
2. 会议结束前确认本次会议中已确认的内容(主要是看记录是否有误或遗漏)，以及下次开会的事宜。
3. 记录应该用纸质需求卡来记录
4. 每次记录完毕后，需要与需求核对表进行比对，看目标是否达到
5. 整理需求卡，将需求录入到网页版的需求卡上，同时补充需求文档
6. 客户在这个过程中会提供若干文件，要仔细分析，整理，根据情况，有必要的要在需求文档中描述
7. 如何将得到的客户需求记录成文
8. 要以客户角色来书写该文档，要求：客户能够读懂
9. 如何评审需求文档，客户确认签字
10. 拿出以往的纸质需求卡&网页版需求卡和需求文档比对，确认需求的准确性
11. 如何做动员大会
12. 项目各方面成员均要参与
13. 明确告知各方面成员如何配合
14. 获得高层的支持与资源
15. 用到了哪些工具
16. 纸质需求卡(Volere)
17. 网页版需求卡
18. 客户术语表
19. 版本管理
20. 软件需求规格说明书模版
21. 用到了哪些方法
22. 隐喻，找个客户熟知的东西，来举例自己想要达到的一个效果，可提升生沟通效率
23. 好的问题定义，只有对的问题才会得到对的答案
24. 在需求阶段，确保项目每个成员(包含客户)知道需求的重要性，以及需求变更的代价
25. 剥洋葱方法
26. 用到了哪些原则
27. 在设计到具体实现某功能是，才能诱导客户选择某种方式实现
28. 开会准备议题时，需安排优先级
29. 需求调研时，只能以产品经理的角色进行
30. 名正言顺

## 结果是什么

1. 软件需求规格说明书文档

## 如何验证结果

1. 用户评审通过

## 会遇到哪些严重的问题

1. 沟通中产生了歧义，牛头不对马嘴(客户讲的程序员不理解，程序员讲的客户不理解)
2. 程序员理解错误
3. 客户描述不清晰，并不知道他想要个什么东西
4. 如何拒绝客户的异想天开的要求
5. 要学会坚持自己的立场，拒绝他人

# 设计阶段

## 要做什么(关注点，好的问题定义)

1. 架构设计
   1. 类设计(UML相关文件)
   2. 数据库设计
   3. 界面设计
   4. 业务规则
   5. 选择开发语言、开发框架
   6. 安全性、耦合性、扩展性、性能、互用性、国际化/本地化、错误处理、容错性、架构的可行性、健壮性
   7. 变更策略
   8. 关于是“买”还是“造”的决策
   9. 确定开发模式

## 怎么做(会用到哪些工具、方法、原则)

1. 如何进行类设计？
2. 根据客户提供的实体单据为依据
3. 结合业务规则(可变|不可变 的因素)
4. 综合各种设计模式
5. 类设计原则(开闭原则、单一职责原则、接口分隔原则、依赖倒置原则、里氏替换原则)
6. 相关约定(命名规则…)
7. 如何进行数据库设计？
8. 以类设计为依据
9. 结合业务规则(可变|不可变|特殊要求 的因素)
10. 综合三范式
11. 相关约定(命名规则…)
12. 如何进行界面设计？
13. 以软件的类型为依据，参考同类软件的设计
14. 收集网上好的设计方案，与用户体验
15. 资料 <http://www.cnblogs.com/Wayou/p/goodui.html>
16. 相关约定(控件命名规则、控件的特殊用法…)
17. 如何选择开发语言、开发框架
18. 基于项目的类型
19. 开发团队的水平
20. 如何设计安全性
21. 如何设计耦合性
22. 如何设计扩展性
23. 用到了哪些工具
24. 类设计|数据库设计：PD
25. 界面设计工具：axure
26. 架构核对卡
27. 常用术语字典
28. 用到了哪些方法

参考他人的设计方案

1. 用到了哪些原则
2. 8/2原则
3. 类设计原则
4. 数据库设计原则
5. 用最简单的设计解决复杂的关系

## 结果是什么

1. 类设计文档(类草图、类优化、时序图、状态图、流程图…)
2. 数据库设计文档
3. UI设计文档
4. 架构思路设计文档

## 如何验证结果

1. 架构能够满足需求文档中的业务要求
2. 验证架构核对卡

## 会遇到哪些严重的问题

1. 不能满足对业务的扩展

# 编码阶段

## 要做什么(关注点，好的问题定义)

1. 按照需求文档要求实现功能
2. 开发前约定(角色定位、人员配合、命名规则、开发模版)
3. 代码框架准备
4. 项目阶段分解，需设定各阶段不同的里程碑
5. 动员大会

## 怎么做(会用到哪些工具、方法、原则)

1. 如何做动员大会
2. 告知团队所有成员，项目正式进入开发阶段
3. 明确各成员的角色、职责，如何配合
4. 公布项目计划，重点提出里程碑的设计
5. 详细描述开发时应该注意的事项
6. 各里程碑完成需开一次会议，以示庆祝，激励士气
7. 用到了哪些工具
8. CodeSmith代码模版工具
9. DEVEXPRESS 第三方的控件
10. 源代码管理工具
11. 程序自动更新程序
12. 企业库
13. 测试用例(防御式编程)
14. 用到了哪些方法
15. 代码分解

全员参与，按功能类别进行分类，每种类别取一个，从客户度到服务器，全部写完，然后进行代码优化，完成后花1-2天进行个方面的考虑。确认无误后，这个就是模版。如果遇到新的功能同样这样处理，这样可确保代码的质量与一致性，代码后期维护比较方便。

1. 伪代码编程
2. 用到了哪些原则
3. 最新的技术不一定造出好的软件，但好的软件一定是最新的技术创造出来的
4. 各司其职，明确角色(工作范围)

## 结果是什么

1. 各里程碑完成

## 如何验证结果

1. 根据项目验收文档&需求文档验收软件
2. 软件基础测试通过

## 会遇到哪些严重的问题

1. 需求不断的变更
2. 框架适应不了新需求
3. 团队士气低落，没有战斗力，没有凝聚力

# 测试阶段

## 要做什么(关注点，好的问题定义)

1. 以上每个阶段都会有测试的参与，需求阶段，验证需求的合理性等，设计阶段测试架构的稳定性等。编码阶段测试软件各方面是否有BUG等
2. 写测试用例文档
3. 测试系统

## 怎么做(会用到哪些工具、方法、原则)

1. 根据测试用例，录入数据进行测试，看结果是否与测试用例中的结果一致
2. 记录调试出的BUG，出现BUG后要能够还原操作步骤
3. 用到了哪些工具
4. 各种测试工具
5. 测试用例文档模版
6. BUG记录文档模版
7. 用到了哪些方法
8. 白盒、黑盒测试等
9. 用到了哪些原则

## 结果是什么

1. 测试用例文档
2. BUG记录文档

## 如何验证结果

1. 所有测试用例全部通过

## 会遇到哪些严重的问题

# 上线运维阶段

## 要做什么(关注点，好的问题定义)

1. 写用户培训手册
2. 用户培训
3. 正式环境配置与发布

## 怎么做(会用到哪些工具、方法、原则)

1. 用到了哪些工具
2. FTP 上传下载工具
3. 文档编写工具

## 结果是什么

1. 软件能够正常使用，自动升级等
2. 用户手册

## 如何验证结果

1. 用户能熟练使用软件

## 会遇到哪些严重的问题

1. 用户高度频繁提出一些细小的问题以及用户体验方面的问题
2. 用户骂娘

# 总结

## 贯穿整个项目的原则

1. 最新的技术不一定造出好的软件，但好的软件一定是最新的技术创造出来的
2. 各司其职，明确角色(工作范围)
3. 8/2原则
4. 用最简单的设计解决复杂的关系
5. 以人为本

## 应该注意最基本的事项

1. 团队成员士气、内部矛盾
2. 互帮互助