



# 项目介绍

## 课程内容：

---

- ▶ 一、项目规划
- ▶ 二、项目开发流程
- ▶ 三、项目文档编写规范
- ▶ 四、如何提高项目实施能力

# 一、项目规划

---

1、项目研发目标

2、项目研发涉及的过程

3、项目需求分析的重要性

4、项目立项的必要性

5、如何确定项目的总体方案

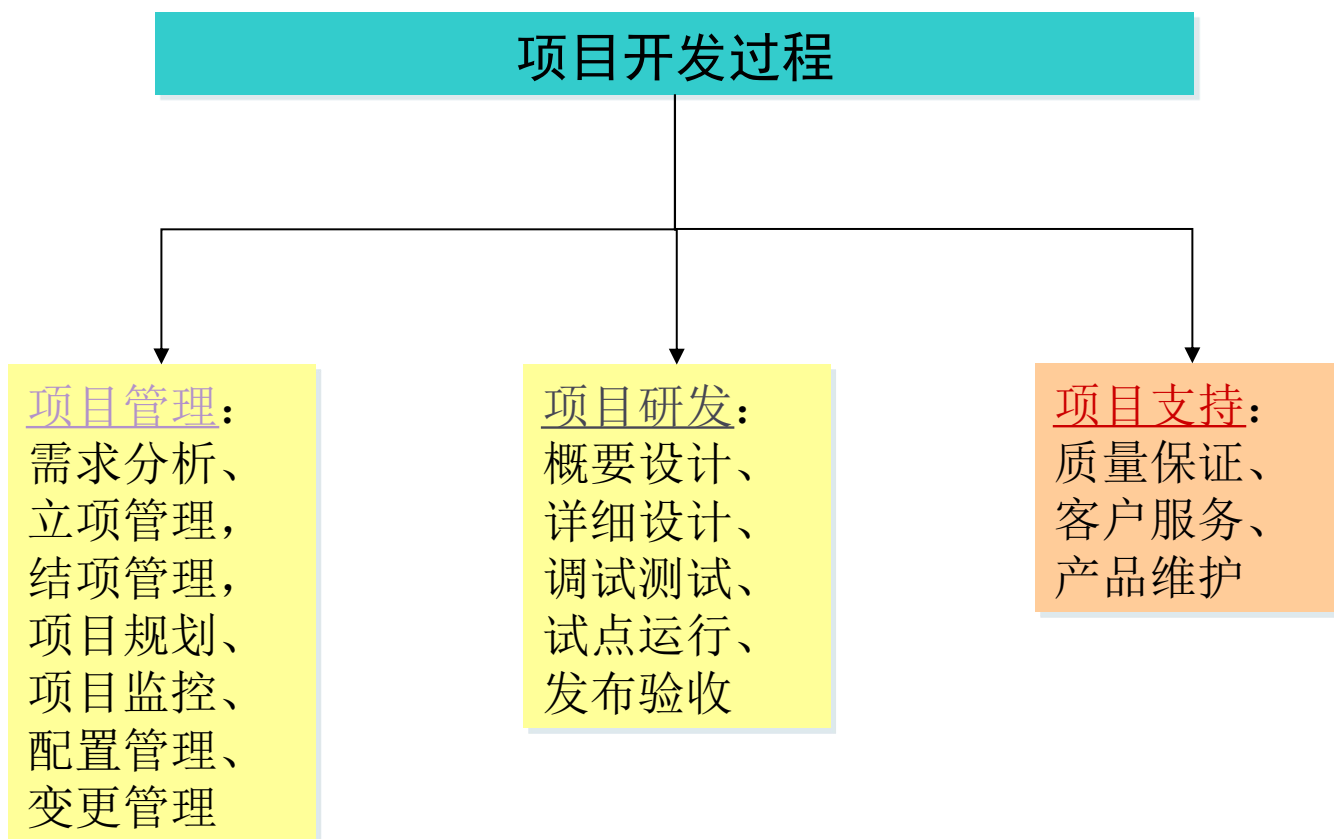
## 1.1 项目研发的目标

企业的根本目标是“合法地赚取尽可能多的利润，使企业利益最大化”

以追求商业利益最大化为总目标，将提高质量、提高效率、降低成本的方法融入到所有过程域中，形成适合于本企业的研发管理过程规范。

- 围绕企业目标展开
- 市场导向、用户第一
- 服从大局、服务企业
- 技术本身非第一要素

## 1.2 项目研发涉及的过程



## 1.3 项目需求分析的重要性

需求分析是要决定“做什么，不做什么”

- ▶ **需求是什么？**简单地讲，需求来源于用户的一些“需要”，这些“需要”被分析、确认后形成完整的文档，该文档详细地说明了产品“必须或应当”做什么。
- ▶ **需求的来源？**  
用户提出（合同项目）、市场调查（自主研发）、自己挖掘（自主研发）
- ▶ **需求开发过程**
  - ▶ **需求开发**的目的是通过调查与分析，获取用户需求并定义产品需求。
  - 需求调查**的目的是通过各种途径获取用户的需求信息（原始材料），产生《用户需求说明书》。
  - 需求分析**的目的是对各种需求信息进行分析，消除错误，刻画细节等。
  - 需求定义**的目的是根据需求调查和需求分析的结果，进一步定义准确无误的产品需求，产生《产品需求规格说明书》。系统设计人员将依据《产品需求规格说明书》开展系统设计工作。

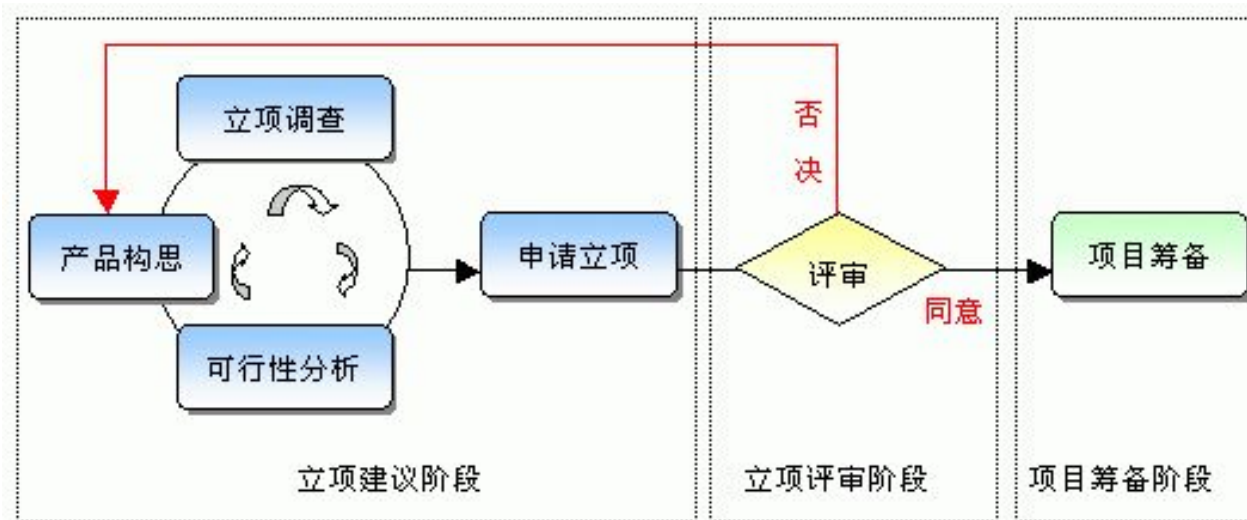
## 1.3 项目需求分析的重要性

---

- ▶ 需求开发的困难
- ▶ A、知识技能问题
- ▶ B、合作关系
- ▶ C、用户说不清需求
- ▶ D、双方误解需求
- ▶ E、需求文档写不好
- ▶ F、用户需求经常变更

## 1.4 项目立项的必要性

项目立项是决策“做正确的事情”，而立项之后的研发活动和管理活动的目标是“正确地做事情”。“正确”就是指符合企业利益最大化这个根本目标。





## 1.4 项目立项的必要性

---

### 项目立项存在的问题

**自主研发项目：**缺乏“调研，可行性分析，立项建议，决策评审”，主要靠公司领导独断，团队只知道干活，却不了解产品的开发背景，不清楚用户期望的产品应该是什么样的。在开发过程中经常迷失方向，导致进度延误、费用超支等问题。

**合同项目：**需求不清晰、合同内容空洞；双方在签订合同的过程中给出了一些空头承诺（例如对进度、质量、费用的估计过于乐观），在实际执行时却难以兑现这些承诺。

## 1.4 项目立项的必要性

### 产品构思

在撰写正式的《立项建议书》之前，立项建议小组首先要在宏观层面上搞清楚“开发什么”、“怎样开发”、“怎样赚钱”等重大问题，即产品构思，主要包括以下内容：

- 待开发产品的主要功能；
- 待开发产品的技术方案；
- 哪些产品部件应当采购、外包开发或者自主研发）
- 开发计划；
- 市场营销计划（如果是合同项目，可能不必考虑市场营销问题）

## 1.4 项目立项的必要性

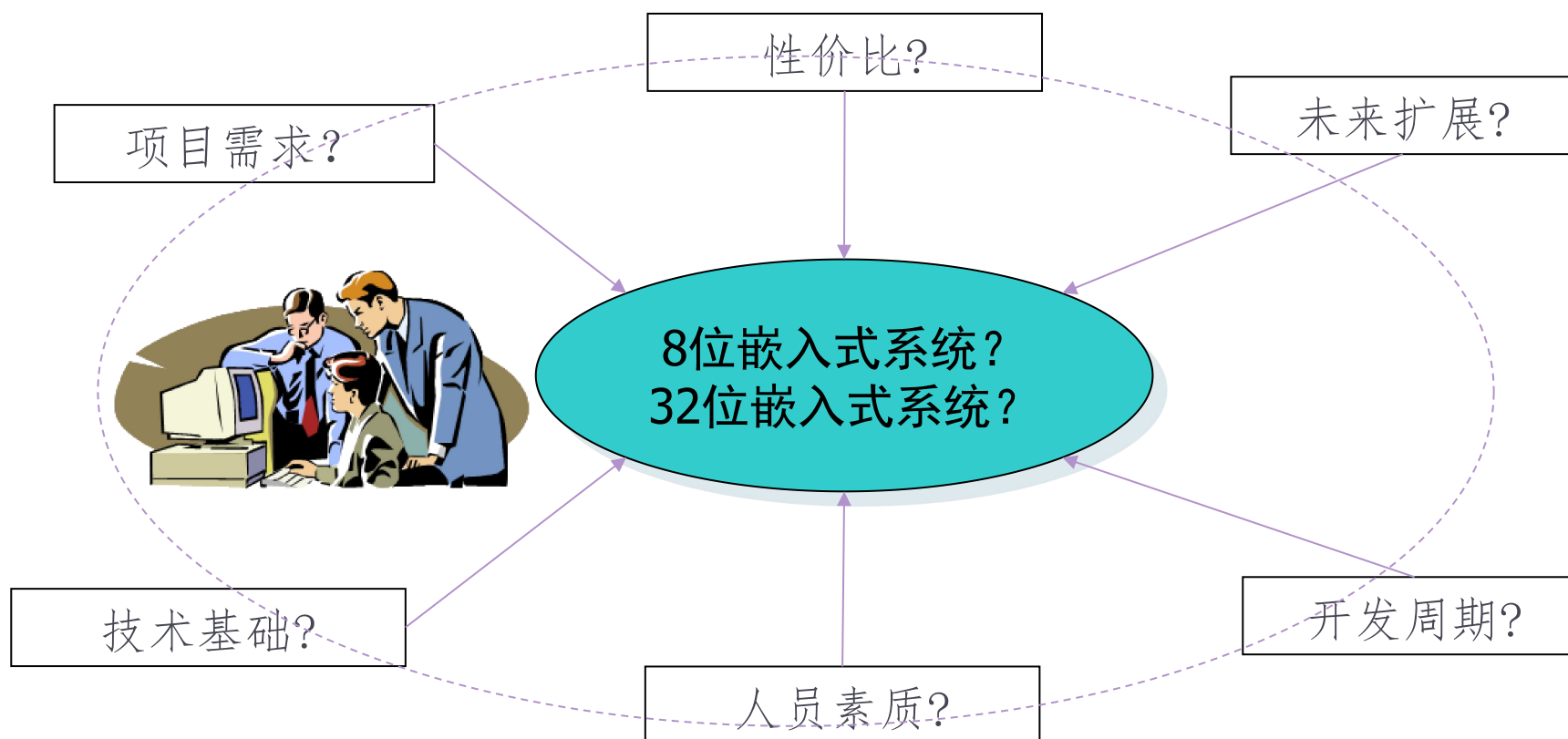
---

《立项建议书》主要论述“开发什么样的产品、如何开发、如何赚钱”

《可行性分析报告》的宗旨是为决策提供有价值的证据和结论

市场可行性分析  
政策可行性分析  
技术可行性分析  
成本效益分析

## 1.5、产品方案如何确定？



## 二、项目开发流程

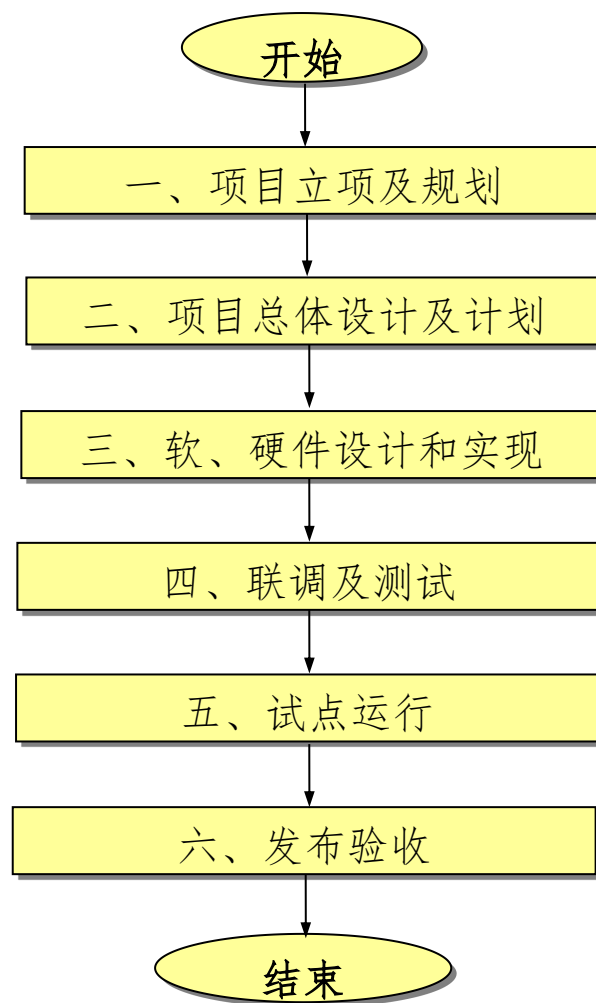
---

1、总体流程介绍

2、项目开发条件

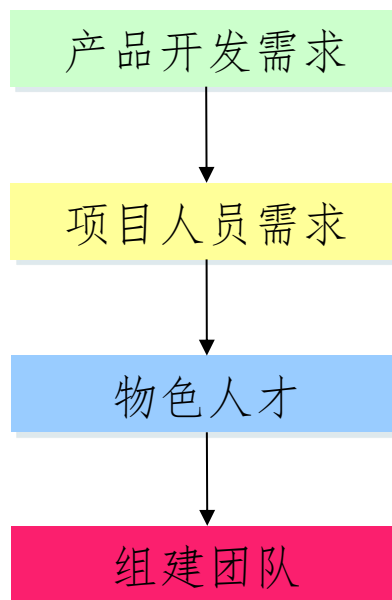
3、项目开发流程

## 2.1 总体流程介绍

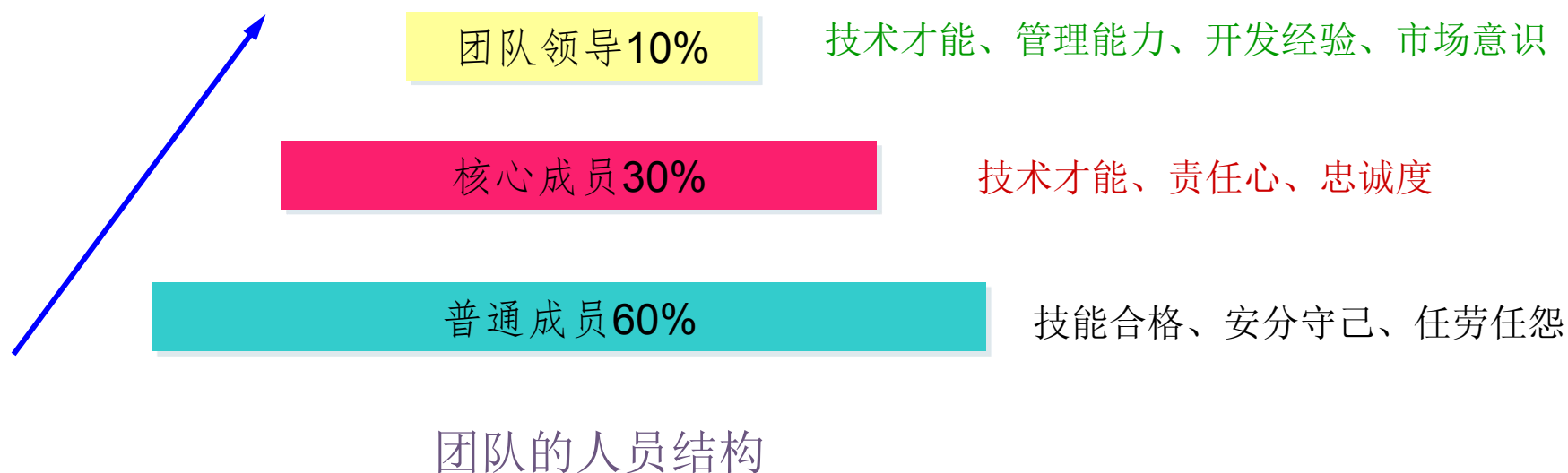


## 2.2 项目开发条件

### 2.2.1 开发团队组建

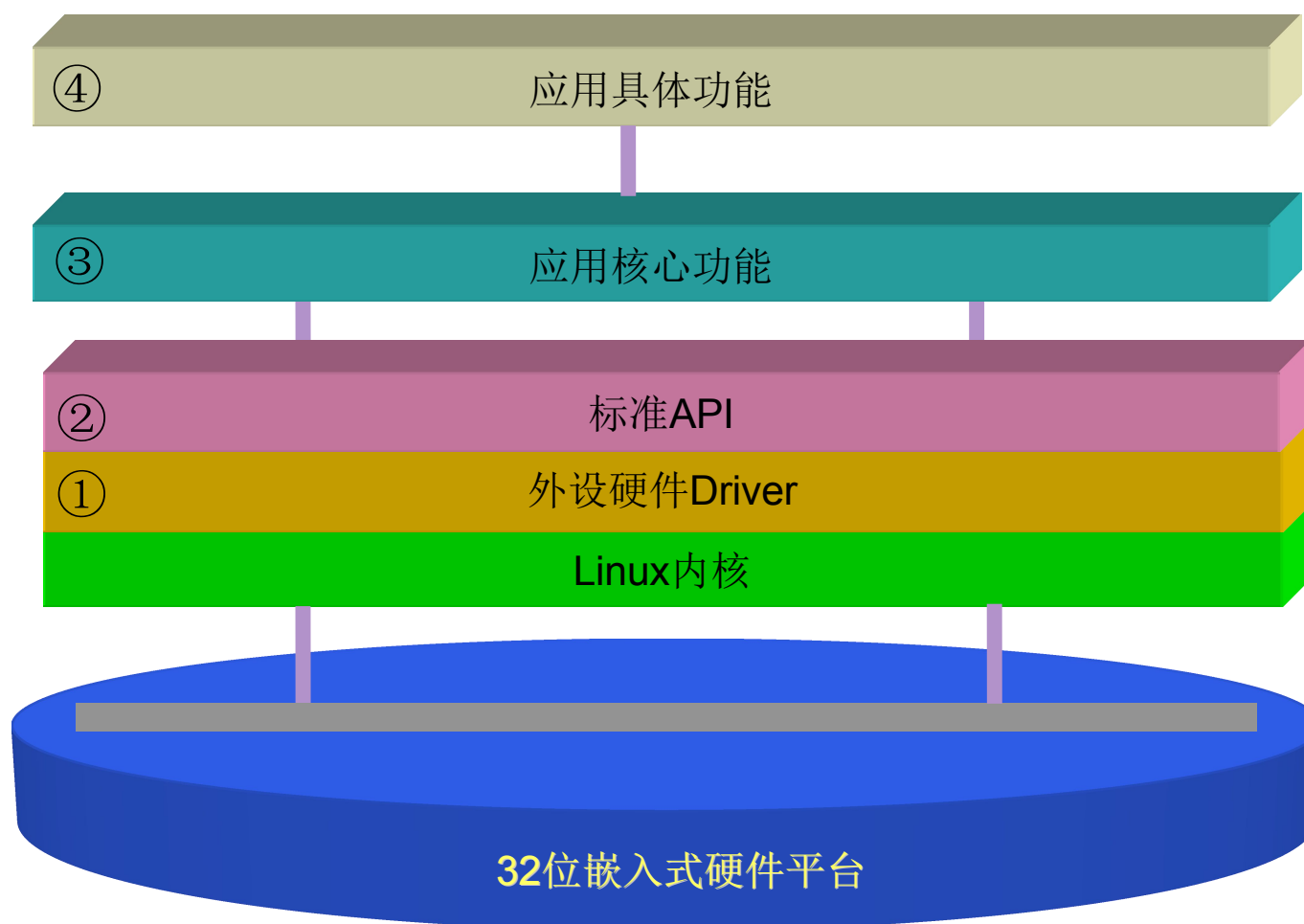


人分四类：人物、人才、人手、人渣

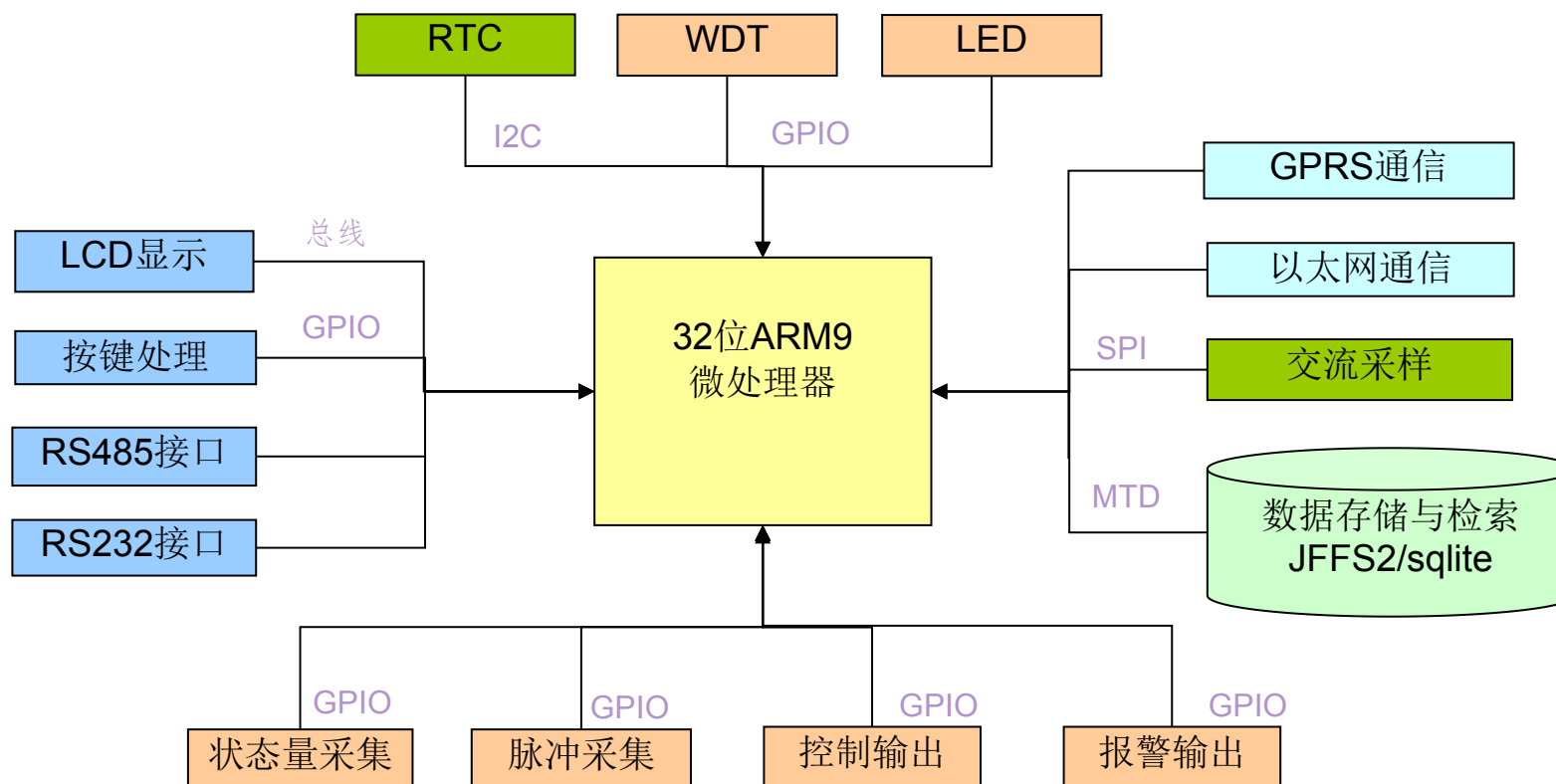




## 2.2.2 技术平台积累

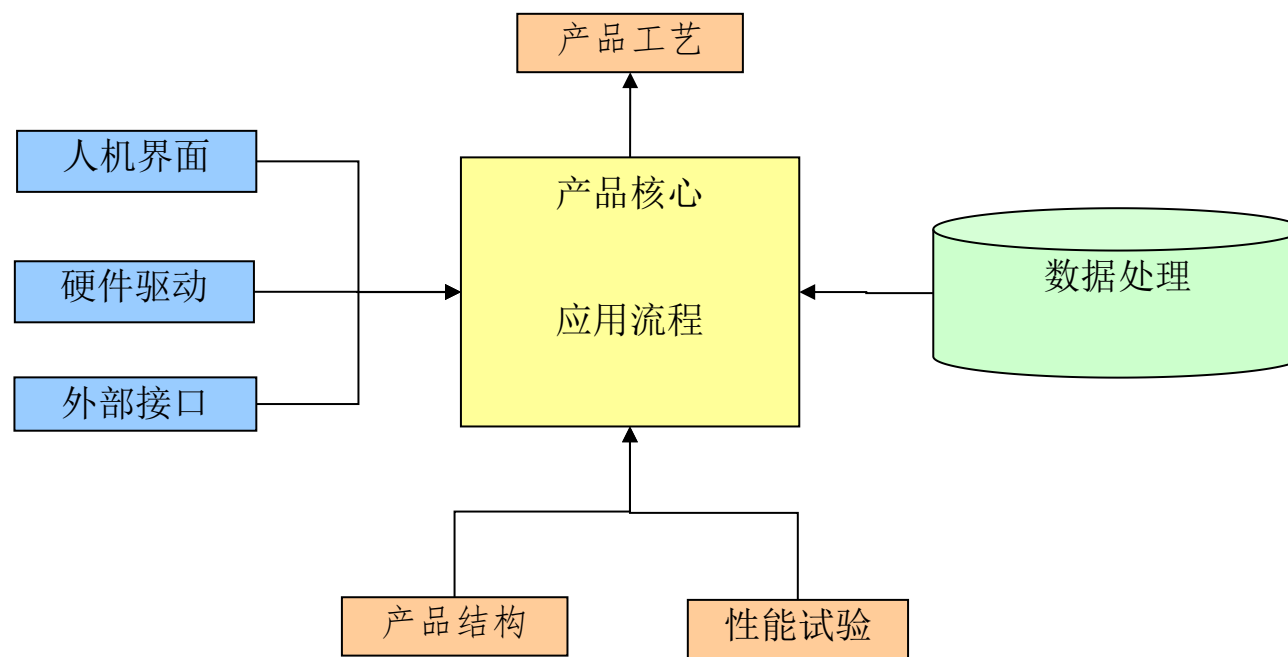


## 哪些技术可以积累？



技术的积累、沉淀可大大加速研发的进度

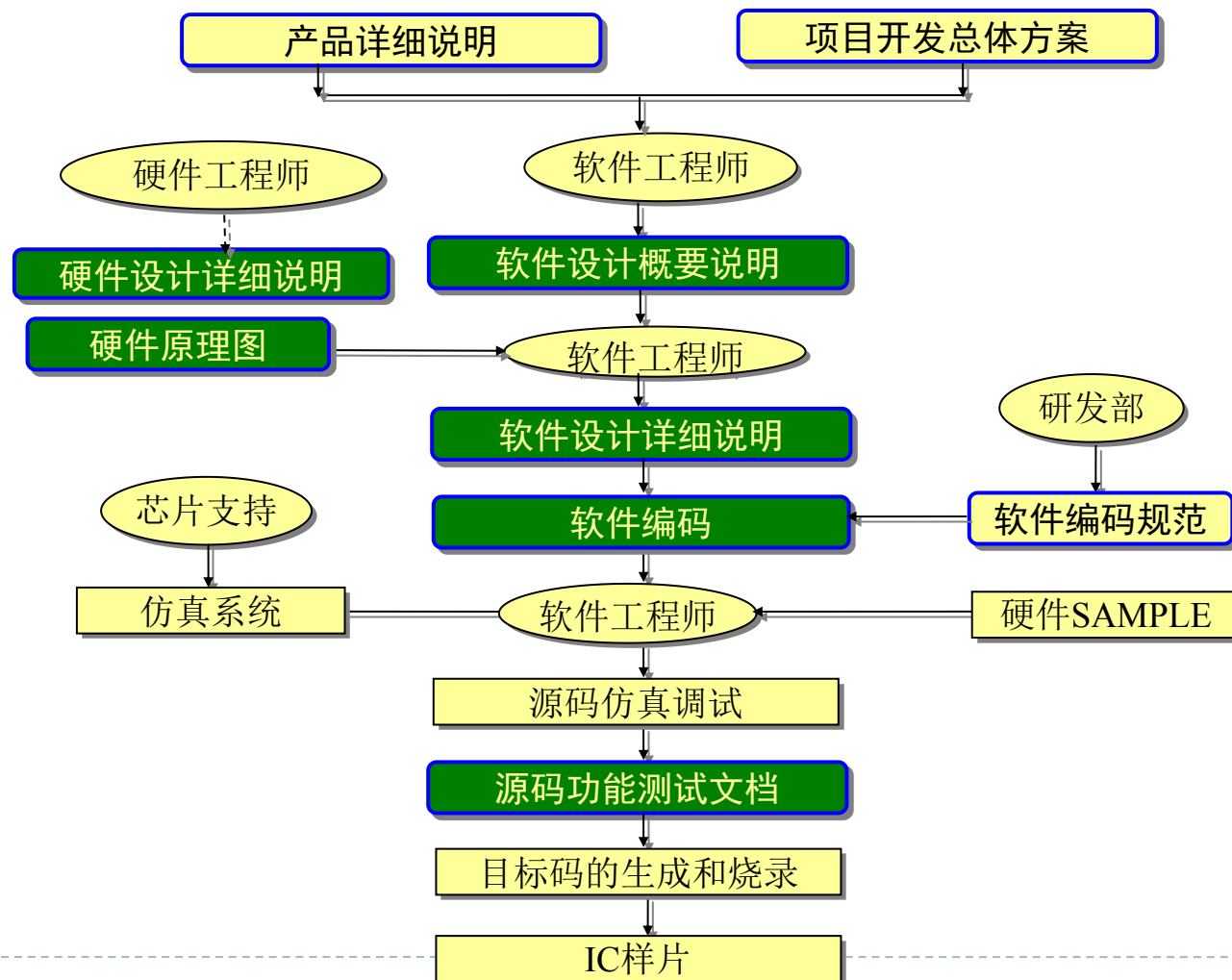
### 2.2.3 关键技术突破



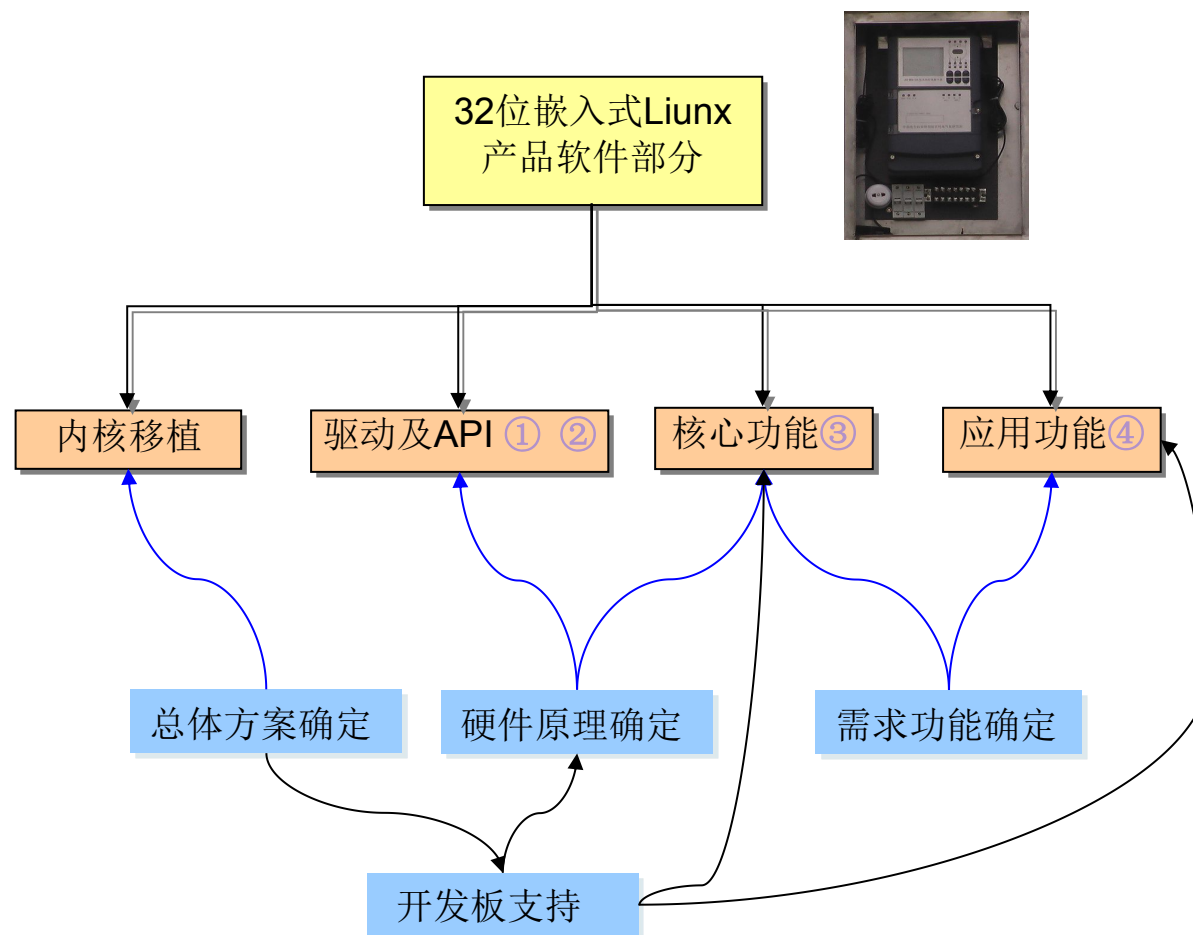
- ☐ 关键技术是否突破决定了项目的成败
- ☐ 关键技术的进展速度决定了项目的进展
- ☐ 关键技术是动态变化的
- ☐ 关键技术是可积累的
- ☐ 关键技术未必是产品的核心技术

## 2.3 项目开发流程

### 2.3.1 软件开发流程



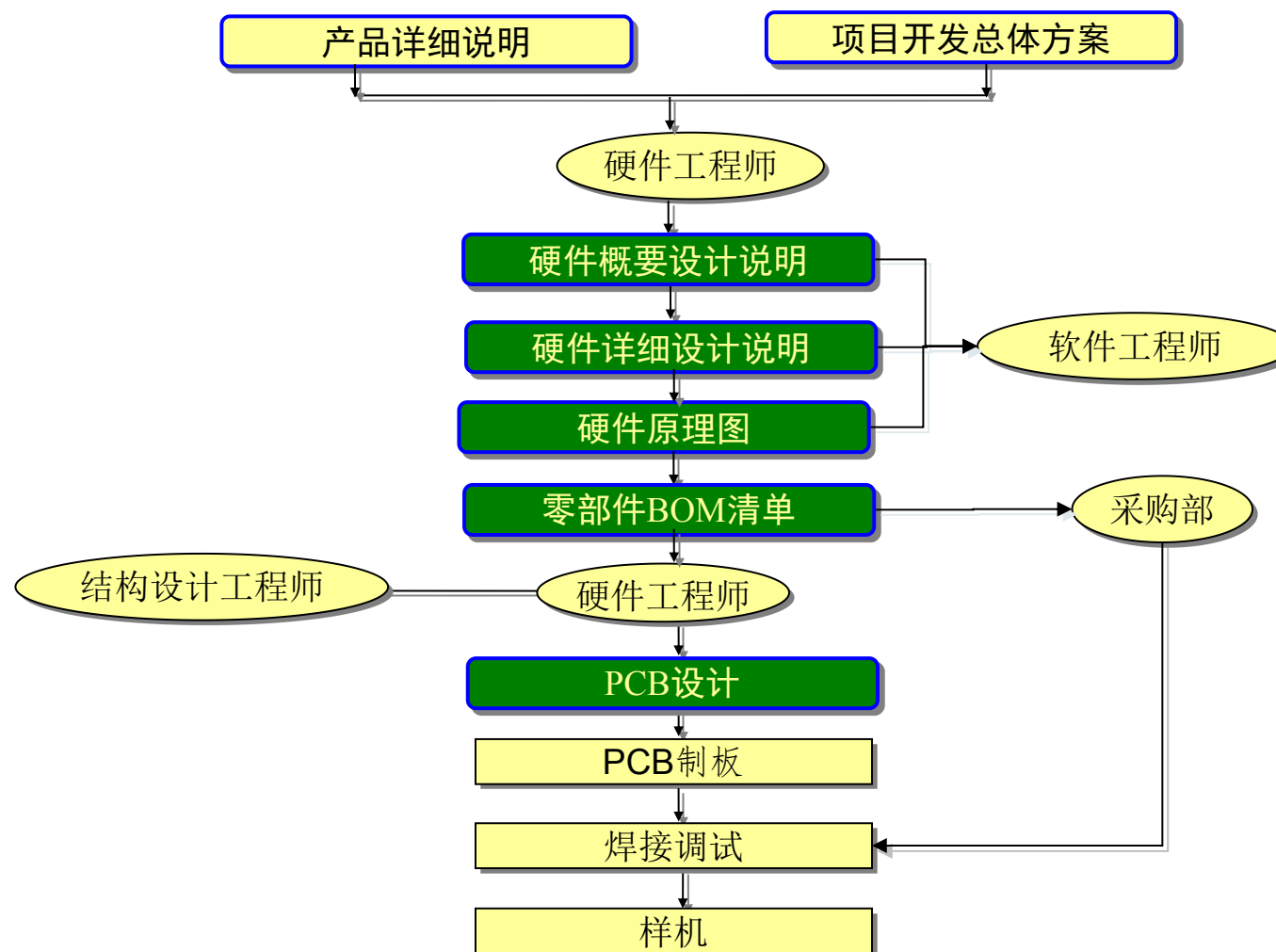
## 2.3.1 软件开发流程



## 自顶向下，逐步求精方法

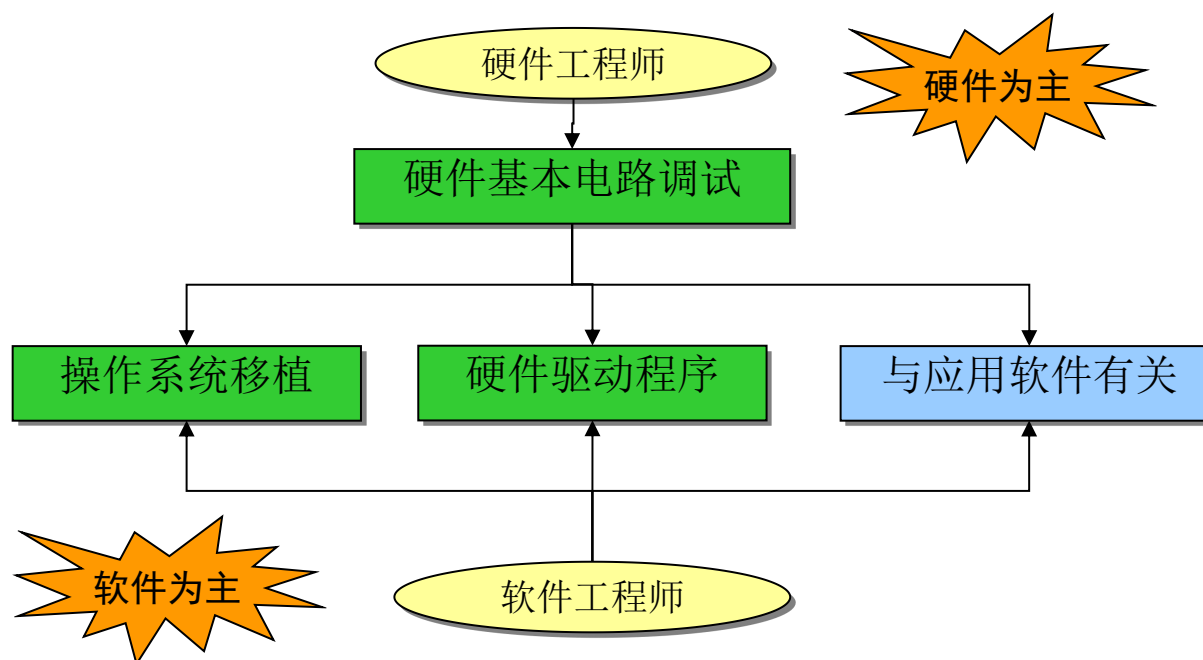
- ▶ 符合人们解决复杂问题的普遍规律。可提高软件开发的成功率和生产率
- ▶ 用先全局后局部，先整体后细节，先抽象后具体的逐步求精的过程开发出来的程序具有清晰的层次结构，程序容易阅读和理解
- ▶ 程序自顶向下，逐步细化，分解成一个树形结构。在同一层的节点上的细化工作相互独立。有利于编码、测试和集成
- ▶ 每一步工作仅在上层节点的基础上做不多的设计扩展，便于检查
- ▶ 有利于设计的分工和组织工作。

## 2.3.2 硬件开发流程



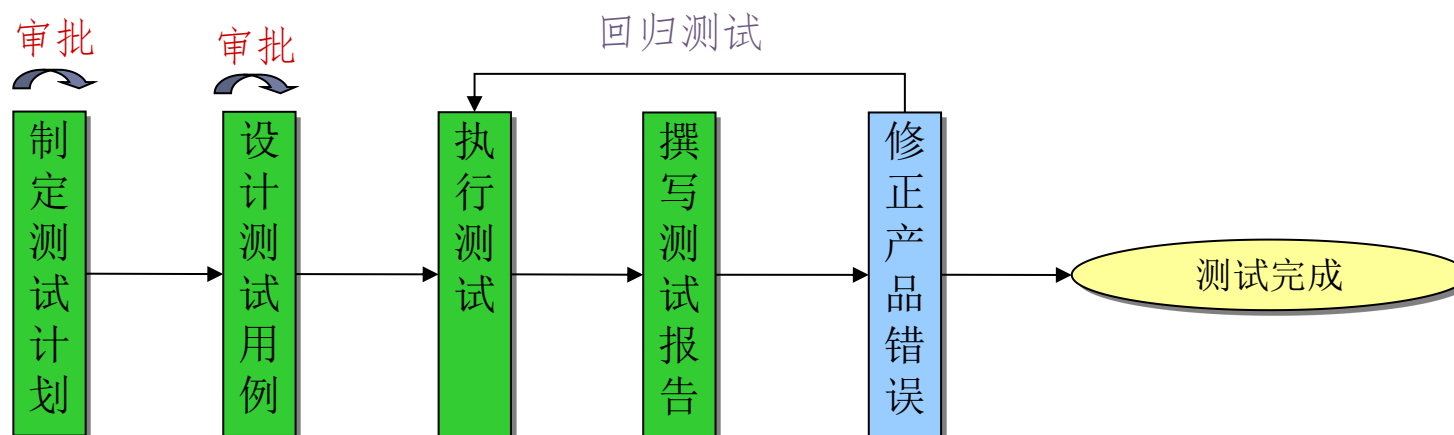


### 2.3.3 调试流程

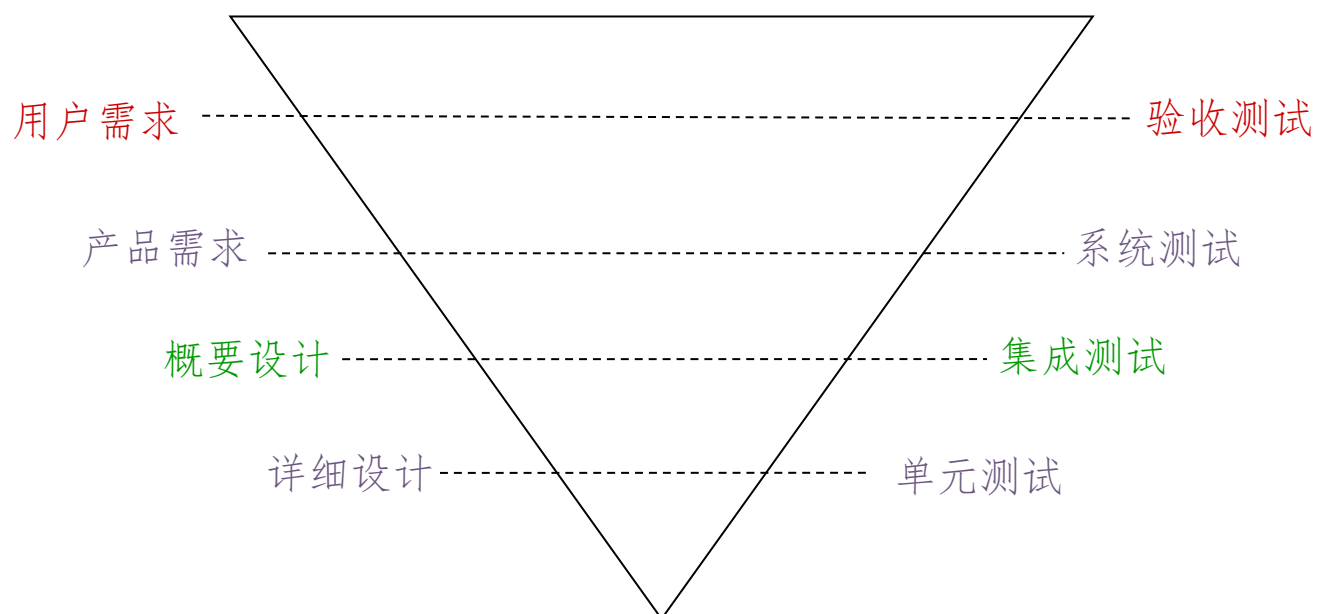


## 2.3.4 测试流程

测试流程图



## 开发与测试的关系



- (1) 白盒、黑盒测试
- (2) 功能测试
- (3) 压力测试**
- (4) 性能测试
- (5) 易用性测试

## 测试报告

---

(1) 纪录测试实况

(2) 对本次测试进行分析、提出建议

- 
- 1、基本信息（测试对象、测试环境、测试人员、测试时间）
  - 2、实况纪录（测试情况、测试用例、错误严重程度）
  - 3、分析与建议
  - 4、错误修改纪录
  - 5、附件

## 测试记录单

项目名称		版本号	XXX	
测试项	管理员管理	测试用例号	006	
初测结论： <input checked="" type="checkbox"/> 通过 <input type="checkbox"/> 不通过				
复测记录				
复测次数	复测情况	复测人	复测日期	结论
错误类型统计（[ ]内填写错误类型数量） <input type="checkbox"/> 可重复错误 <input type="checkbox"/> 不可重复错误 <input type="checkbox"/> 死机性错误 <input type="checkbox"/> 数据性错误 <input type="checkbox"/> 影响其它模块运行错误 <input type="checkbox"/> 导致测试停顿性错误 <input type="checkbox"/> 需求错误 <input type="checkbox"/> 设计错误 <input type="checkbox"/> 计算错误 <input type="checkbox"/> 帮助等文字错误				

### 2.3.5 试点运行、发布验收、推广应用

1、现场运行环境会复杂和不确定

2、修改、完善和改进是必然的

3、产品测试好再去试点运行

4、能在家做的事千万别在现场做

5、勇于承担，不要推卸责任

6、详细做好纪录，不能疏忽大意

7、尽可能满足用户的合理需求

## 三、项目文档编写规范

---

1、为什么要编写项目文档

2、需要编写哪些项目文档

3、如何编写项目文档？

4、代码编写规范

5、书面和口头表达的重要性

## 3.1 为什么要编写项目文档

### 什么是文档？

在项目工程中，文档用来表示对活动、需求、过程或结果进行描述、定义、规定、报告或认证的任何书面或图示的信息。

应说清楚下列问题：



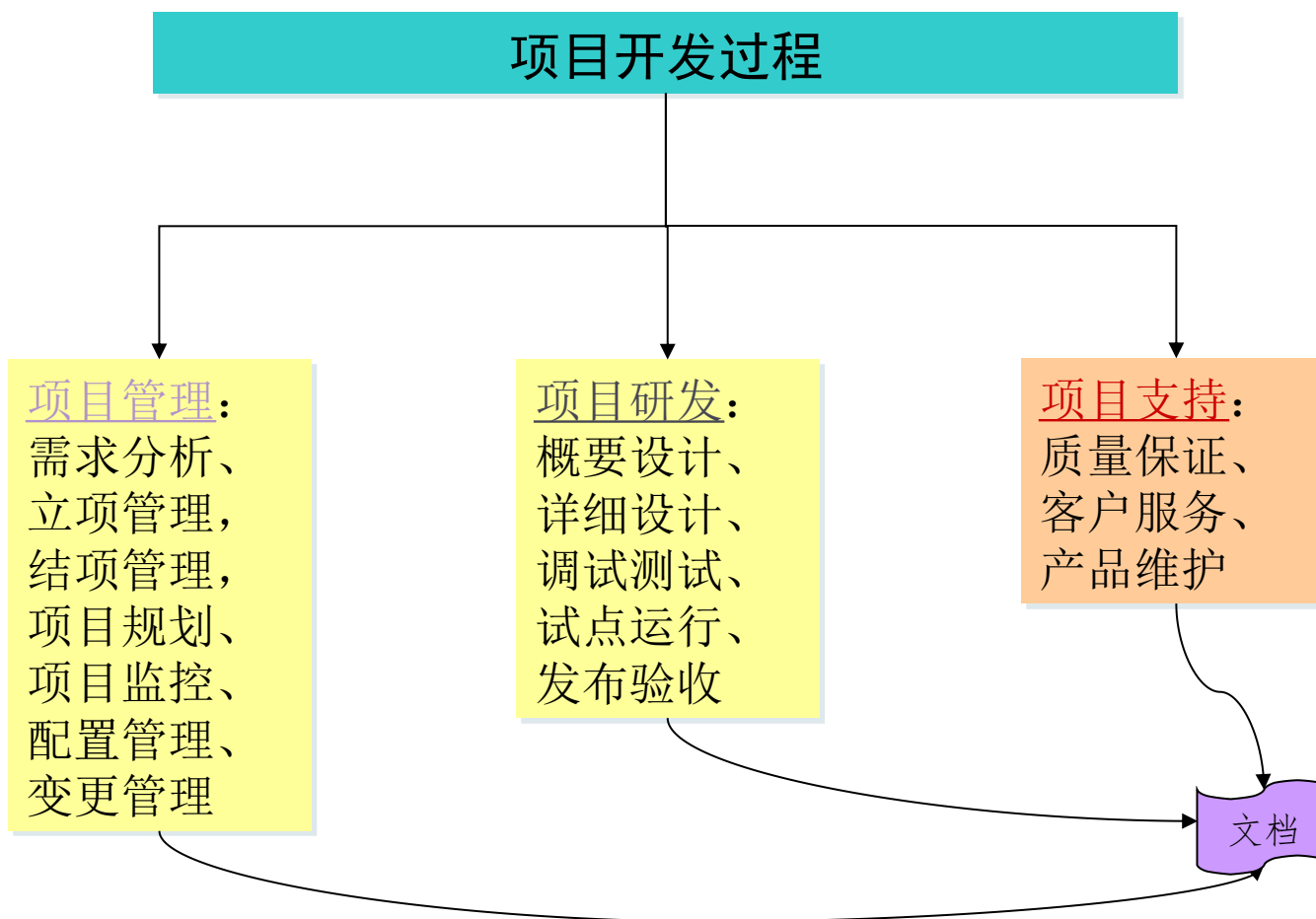
是什么？  
做什么？  
为什么做？  
谁来做？  
怎么做？



在项目过程中，会产生和使用大量的信息，文档在项目的开发过程中起着重要的作用，文档是开发人员思考和开发过程的纪录。

- 便于管理汇报。项目管理者可把这些记载下来的材料作为检查项目开发进度和开发质量的依据，实现对项目开发的工程管理。
- 提高开发效率。文档的编制，可使开发人员对各个阶段的工作都进行周密思考、全盘权衡、减少返工。并且可在开发早期发现错误和不一致性，便于及时加以纠正。
- 便于总结分析。作为开发人员在一定阶段的工作成果和结束标志，也可为基础后续工作奠定基础。
- 便于协调沟通。便于管理人员、开发人员、操作人员、用户之间的协作、交流和了解。使项目开发更科学、更有成效。

## 3.2 需要编写哪些项目文档



## 3.3 如何编写项目文档

文档编写存在的问题：

(1) **文档编写不够规范**。主要表现在项目实施过程中，没有对每一个阶段的文档进行整理，分层次管理，各类文档资料缺少一致性。

(2) **文档内容描写不够完善**。在编写各种文档过程中，虽然大家都按事先规定的模式进行了编写，但编写的内容经常不够完善。要么文档极其简单，相当于没有文档。要么文档流于形式，没有什么实际的价值。

(3) **文档没有统一入库管理**。随着系统的不断优化、升级，新需求的不断产生，各种文档越来越多，没有建立一个文档资料库。不同时期的各种文档零散存在，造成查询文档时非常困难。

(4) **只重视文档的形式，实用性不强**。文档的制定要实用，以减少繁文缛节的文字工作。

(5) **文档的安全性无法保障，未能有效管制机密文档的外漏问题**。项目一些文档必定是关键文档，起到非常重要的作用。对于这类文档没有设定优先级别特别关注。



不应该只为写文档而写文档，良好的文档是思想交流、沟通的基础，也是整理和理清思路的基础。

## 3.4 代码编写规范

代码实际上也是一种供人阅读的文章，有一个文章的风格问题。应该使程序具有良好的风格。

1、命名规范

2、注释

3、结构化、接口化、非耦合

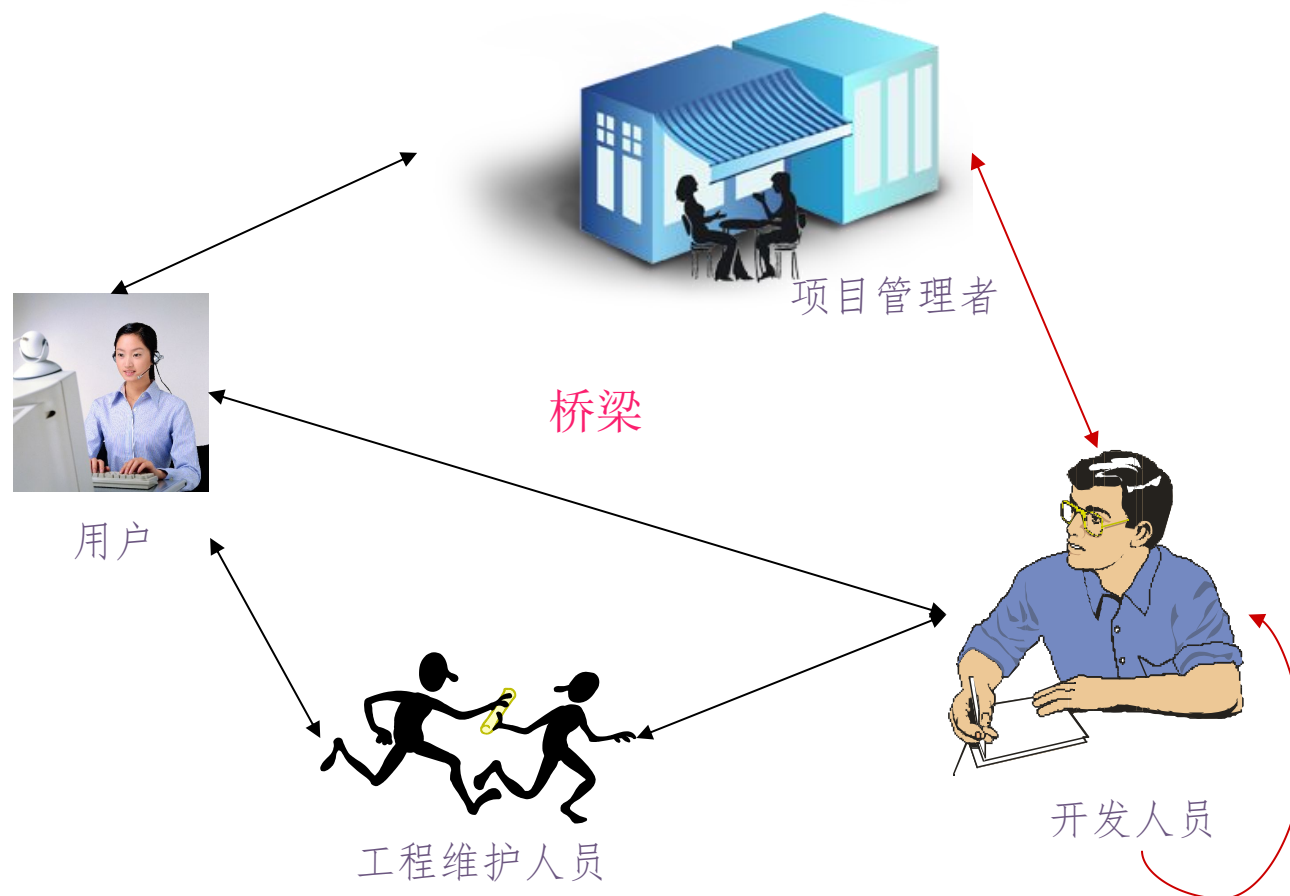
```
/*
*****
功能描述:链路层控制域组建
入口参数:buf:包头数据结构,AFN:应用层功能码,ACK_Flag确认标志,DIR传送方向
出口参数:无
返回值:无
时 间:XXXX-XX-XX
设 计 人:xxxxxx
*****
*/
void TLinkLayer_Build_Control(APDU_Head* buf,
                               BYTE AFN_Code, //应用层功能码
                               BYTE ACK_Flag, //确认标志
                               BYTE DIR)      //传送方向
```



嵌入式学院

华清远见旗下品牌

## 3.5 书面和口头表达的重要性—讲清楚思路



## 四、如何提高项目实施能力？

---

1、开发人员的职业规划

2、提高项目实施能力的前提

3、提高项目实施能力的方法

## 4.1 开发人员的职业规划



最低标准：社会责任/家庭责任

## 几点建议：

规划好发展方向，方向别偏太多

珍惜机会、在专业上要下功夫

**25至35岁**是人的创造黄金期

上天公平，付出就会有回报

多做少抱怨，能承受委屈

具备能力才能抓住机会





## 4.2 提高项目实施能力的前提

1、敬业、责任、追求

2、谦虚做人、勤恳做事

3、认真对待主管交给的任务

4、正确的思考、研究方法



## 4.3 提高项目实施能力的方法

---

### 4.3.1 以实践入门，在项目中成长

- 多练，多实践，切忌纸上谈兵
- 多参与项目开发，在项目开发中积累技巧、经验，切忌得过且过
- 在项目实践中多思考，达到举一反三的作用，切忌就题论题
- 在项目实践中要善于总结，领悟原理，不放过任何疑点，切忌一知半解

### 4.3.2、学习、实践、再学习

---

- 多学习，学习可以站得高一点
- 学习后实践，实践必须脚踏实地
- 除非专门学习，否则学习不可能面面俱到，用到什么学什么
- 实践中遇到问题，则继续学习，解决问题
- 螺旋上升，提高能力，不可能一步到位

### 4.3.3、因人而异，术有所专

---

- 任何人不可能对Linux技术面面俱到，全面掌握
- 依据自己的基础、兴趣和工作需要，选择自己的方向
- 大概念要有，做到面宽技专

#### 4.3.4、认识本质，注意积累

---

- 分析和解决问题，要先搞清楚原理
- 多条思路寻找解决问题的方法
- 不能一知半解，留有隐患
- 要深入认识事物本质，抓住根本

### 4.3.5、搜索+模仿

---

- 善于利用已有的Linux资源，多搜索资源
- 不是什么都自己从头干就是最好
- 多参考已有的成果、资源，消化吸收，为自己所用
- 模仿也算实践，学习的入门起始阶段模仿尤其重要

### 4.3.6、利剑磨成，专业结合

---

- 除非专业搞嵌入式Linux，否则其在一定时候只是你手上一把利剑
- 嵌入式Linux技术要和从事的专业紧密相结合
- 不能纯粹为了学习Linux而只懂Linux
- 以嵌入式Linux为平台，创造更多的应用产品，服务企业