

Ch. 4 项目与项目进程

2017.4



4.0 从哪里开始？

- 你PC上的目录是否经常找不到？
- 从整理目录开始吧！

- 认识项目
- 组织和团队
- 项目的开发过程
- 需求分析

4.1 认识项目

- 什么是项目
 - 项目 → project → 项目,工程; 计划,规划;

Project

From Wikipedia, the free encyclopedia

For the urban low-income housing buildings called projects, see [Public housing](#). For other uses, see [Wikipedia Projects](#). For [:Wikipedia:WikiProject](#), see [Project \(disambiguation\)](#).

In contemporary [business](#) and [science](#), a **project** is an individual or collaborative enterprise, possibly involving research or design, that is carefully [planned](#), usually by the project assigned team, to achieve a particular aim.^[1]

One can also define a project as a set of interrelated tasks to be executed over a fixed period and within certain cost and other limitations.^[2]

One can view projects as temporary (rather than permanent) [social systems](#) or as [work systems](#) that are constituted by [teams](#) within or across organizations to accomplish particular [tasks](#) under time constraints.^[3] An ongoing project is usually^[*quantify*] called (or evolves into) a [program](#).

项目

联合国工业发展组织《工业项目评估手册》对项目的定义是：“一个项目是对一项投资的一个提案，用来创建、扩建或发展某些工厂企业，以便在一定周期内增加货物的生产或社会的服务。

项目管理协会（Project Management Institute, PMI)认为：项目是为完成某一独特的产品或服务所做的一次性努力。

中国项目管理知识体系纲要（2002版）中对项目的定义为：项目是创造独特产品、服务或其他成果的一次性工作任务。

《项目管理质量指南(ISO10006)》定义项目为：“具有独特的过程，有开始和结束日期，由一系列相互协调和受控的活动组成。过程的实施是为了达到规定的目标，包括满足时间、费用和资源等约束条件”。

- 项目是在一定约束条件下（主要是限定资源、限定时间、限定质量），具有特定目标的一次性任务。

项目的一些特点

- 一次性
- 有明确的目标
- 有限定条件
- 计划性

项目的目标

在限定的条件下完成一个产品：

一款商业保险

一款新的冰淇淋

一款软件

完成目标的途径

把大象关进冰箱里

- 整理出需求
- 找到解决方案
- 把方案完成

“解决方案”

- 是不是软件项目一定要写代码呢？
- 用最低的成本解决问题吗？
- 有现成的“解决方案”，为什么还要做项目呢？

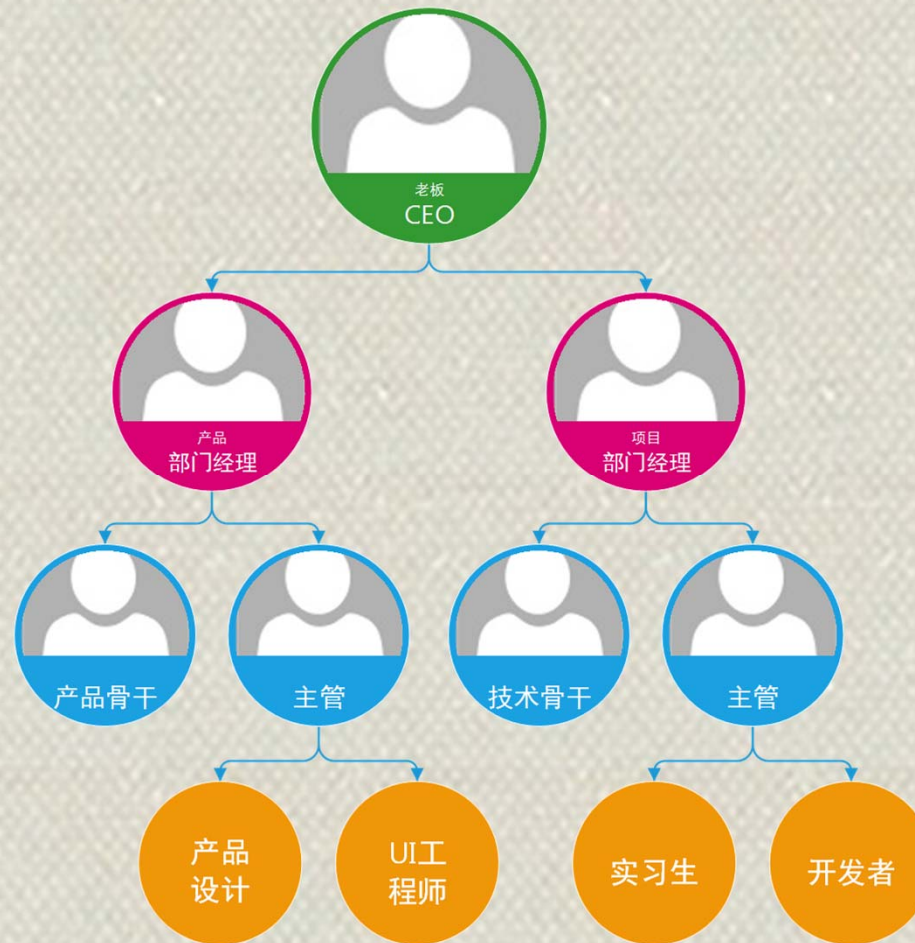
项目来源

- 客户需求
- 专业产品驱动
- 竞争产品刺激
- 作业和毕业设计

项目中的角色

- 组织结构角度
- 职能角度

组织结构



职责

- CEO，全面的管理工作、调动各种资源（人力资源、资本、社会资源）

- 产品经理，产品设计，产品相关的管理和资源协调（成本控制、产品预期），对产品负责
- 项目经理，项目进度控制，整体计划的制定者，项目成本控制，是项目内资源协调者，工作内容的分配者，对项目负责

- 项目骨干，核心工作内容或核心技术的主导者
- 主管，工作内容的划分和分配，重要工作内容的设计和实现，工作执行和反馈

员工

- 执行和反馈，具体工作（功能）的实现者，对工作内容加以反馈

职能角度

软件项目过程中会涉及到哪些角色？

软件产品都有哪些元素

- 可执行程序
- 美术效果、音效
- 数据库
- Localization /Internationalization
- 使用说明 Instructions

- 程序开发人员
- 数据库工程师和数据库管理员
- 美工、音乐家
- 翻译
- 文案

程序开发人员

- 开发者
 - 核心业务开发人员（C++/JAVA/C#）：前/后端、服务器/客户端
 - 运维人员（脚本开发，自动化操作，自动化流程处理）
 - 平台支持业务（技术支持网站、客服网站，微信接入）
 - 数据库设计和数据库管理员（DBA）

沟通

- 为什么要沟通（内部、外部）
 - 团队管理需要（认同感、工作方式、团队文化）
 - 项目共识需要（反馈和上情下达）

项目中都有哪些沟通方式

- 会议
- 邮件
- 流程单
- 协作工具
- 工作交接
-

会议










- 立项会
- 评审会
- 项目节点会（里程碑会议）

最有效的沟通方式

- 文档化 立字据
- 流程化 流水线

一些流程化的方法

- 分工记表
- 用流程软件

 WBS 表.xlsx	2012/8/13 16:48
 项目变更管理表.xlsx	2012/8/13 16:50
 项目风险管理表.xlsx	2012/8/13 16:50
 项目沟通计划表.xlsx	2012/8/13 16:50
 项目会议纪要.xlsx	2012/8/13 16:50
 项目进度计划表.xlsx	2012/8/13 16:49
 项目任务书.xlsx	2012/8/13 16:49
 项目总结表.xlsx	2012/8/13 16:50
 项目组成员表.xlsx	2012/8/13 16:48

一些流程软件

- Bugzilla: 开源的缺陷跟踪系统（Bug-Tracking System），它可以管理软件开发中缺陷的提交（new），修复（resolve），关闭（close）等整个生命周期。



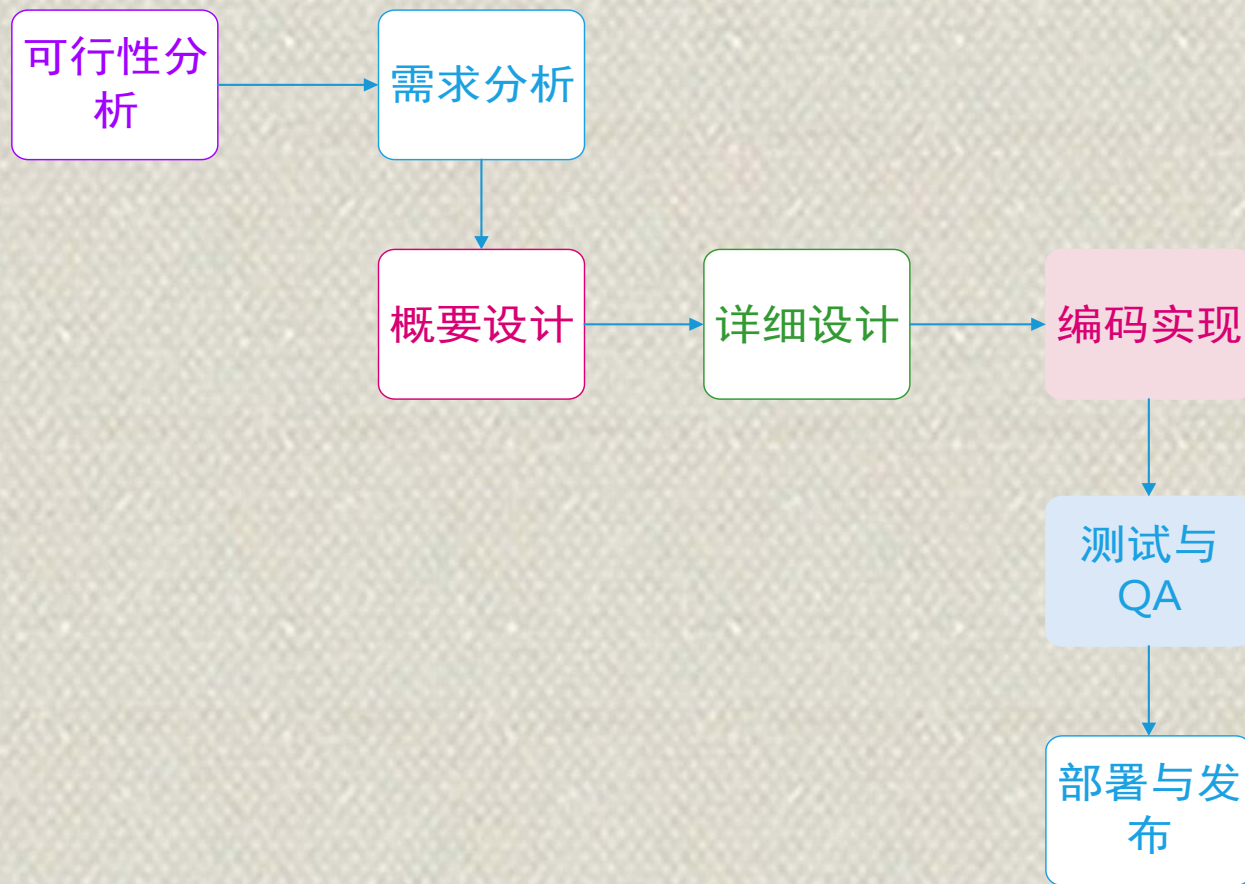
一些流程软件

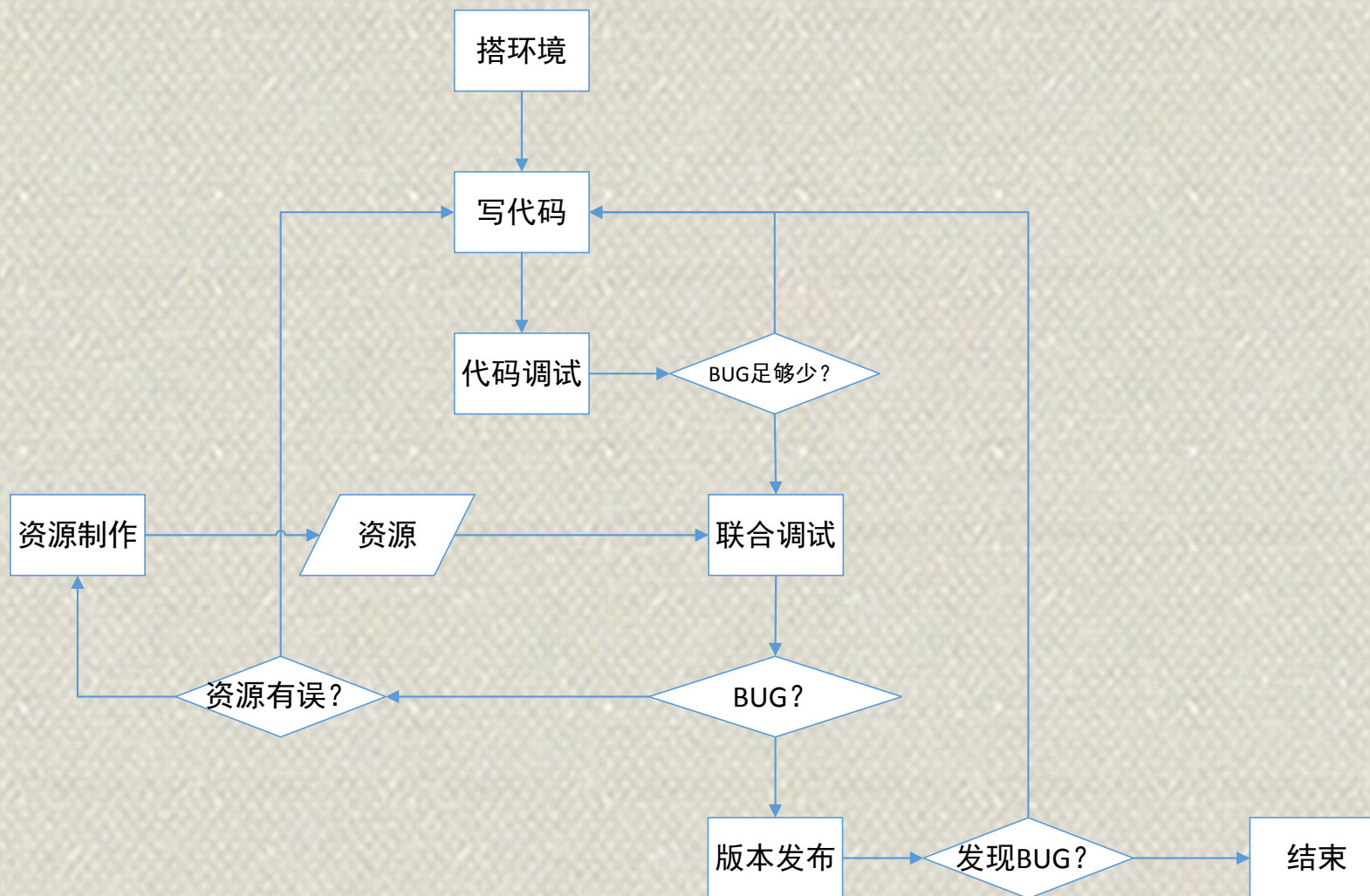
- **Jira** :被广泛应用于缺陷跟踪、客户服务、需求收集、流程审批、任务跟踪、项目跟踪和敏捷管理等工作领域。



软件项目的开发过程

- 需求分析
- 概要设计
- 详细设计
- 编码实现
- 测试/QA
- 部署
- 维护





可行性分析

- 更多市场层面和产品层面的分析
- 产品的定位（卖出去，还是实训项目？）
- 技术可行性与团队的技术实力考量
- 工期是否可控，投入是否可量化
- 竞争产品当前的状况（抄，还是改进？）
- 核心竞争力（技术上，还是产品设计上？）

Let us go

- 自行设计一款计算器



- 统计计算器
- 汇率换算计算器
- 税率（年终奖计算器）

需求分析

- 需求有多重要？
- 没有需求你甚至不知道要做什么

一个段子

G20上午开会休息了，马老板对秘书说：中午帮我买肯德基，30分钟后，秘书回来说，买好了，一共4.6亿美元，咱是支付宝还是现金？马云眉头一皱，马上说：赶紧把小王追回来！我刚让他去买中南海了...告诉他：那是烟！是烟！！是烟！！！！



- 根据Standish Group对23000个项目进行的研究结果表明，28%的项目彻底失败，46%的项目超出经费预算或者超出工期，只有约26%的项目获得成功。
- 而在于这些高达74%的不成功项目中，有约60%的失败是源于需求问题。
- 也就是说，有近45%的项目最终因为需求的问题最终导致失败

怎样做需求分析

- 面对面访谈(face-to-face interviewing)
- 专题讨论会(workshop)
- 现场观察(observing on the scene)
- 头脑风暴(brainstorming)

要做什么样的需求分析

- 明确，明确要做什么，过程中会不会更改？
- 无歧义，不同行业背景的人，对相同的词汇的理解可能是不同的
- 具体，越具体越好
- 把“用户的需求”变成“产品需求”再转成“项目的需求”

需求分析的要点

- 功能性需求(Functional Requirements)
- 非功能性需求(Non-Functional Requirements)

功能性需求

- 完备性Completeness：软件能够支持用户所需求的全部功能的能力；
- 正确性Correctness：软件按照需求正确执行任务的能力；
- 健壮性Robustness：在异常情况下，软件能够正常运行的能力

容错能力

恢复能力

- 可靠性Reliability：在一定的环境下，在给定的时间内，系统不发生故障的概率，或者是快速从错误状态恢复到正确状态的能力

非功能性需求

- 性能Performance Quality：时间/空间效率
- 易用性Usability：学习成本，傻瓜式，习惯的用户体验
- 清晰性Clarity：易读、易理解，可以提高团队开发效率，降低维护代价；
- 安全性Safety：授权，密码，是否会被注入恶意代码
- 可扩展性Expandability：添加一种新的运算是否方便？
- 兼容性Compatibility：你的计算器运行在什么环境下？你的计算器开了，是不是别的软件就不能用了？如果是页面版的，使用哪种浏览器打开？
- 可移植性Portability：能方便地进行移植吗？网页版的可能更好一些

需求发生变化怎么办？

- Catastrophic ? new project : Update all;