

主要内容

- 1. 需求获取概述
- 2. 确定项目的前景和范围
- 3. 涉众分析与硬数据采样
- 4. 面谈
- 5. 原型
- 本章小结

5.1. 应用原型的必要性

5.2. 原型的类别

5.3. 原型方法过程

5.4. 原型方法的风险

5.1.应用原型的必要性

——什么是原型

- “原型是一个系统，它内化了（capture）一个更迟系统（later system）的本质特征。原型系统通常被构造为不完整的系统，以在将来进行改进、补充或者替代。”
- 如果在最终的物件（final artifact）产生之前，一个中间物件（mediate artifact）被用来在一定广度和深度范围内表现这个最终物件，那么这个中间物件就被认为是最终物件在该广度和深度上的原型。

5.1.应用原型的必要性

——什么是原型

- 包括书面描绘、场景叙述、情节串联图板、幻灯演示、动画模拟、屏幕快照和程序代码等在内的各种被用来探索和论证软件系统功能的物件都是软件的原型
 - 一个组织会发展出自己独特的原型文化，这些原型文化仅仅将一些限定类型的事物看作是有效的原型

5.1.应用原型的必要性

——为什么要利用原型

■ 利用原型的好处有：

- 提高用户在软件开发中的参与程度；
- 及时、有力的响应用户需求的变化；
- 减少返工；
- 减少开发成本，提高经济效益；
- 有效的识别风险和解决风险，帮助进行风险管理；
- 帮助控制不完整需求所带来的风险；
- 可以将一个大的难以处理的开发过程细分成一些更小更容易处理的步骤；增加开发者之间的交流，帮助确定技术解决方案的可行性；

5.2. 原型的类别

——按照开发方法分类

■ 探索式 (exploratory)

- 以缺陷需求开始继而不断调整和修正需求的原型开发方式称为探索式
- 要尽可能的调整各种设计选项

■ 实验式 (experimental)

- 以清晰的用户需求和模糊的实现方法、实现效果、可行性开始，明确需求的可行性和技术实现方案
- 定义一个对原型的评估方法，确定评估的属性

■ 演化式 (evolutionary)

- 以清晰的原型化需求和项目积累下来的原型资产为开始
- 原型化的需求，也有项目积累下来的原型资产

5.2. 原型的类别

——按照开发方法分类

- 探索式和实验式方法产生的原型产品又被称为抛弃式原型
 - 花费最小的代价，争取最快的速度
 - 可能会使用简易的开发工具和不成熟的构造技术
 - 可能会忽略或简化处理原型目的不相关的功能特征
 - 要坚决的抛弃
- 演化式原型方法产生的原型产品被称为演化式原型（evolutionary prototype）
 - 质量要从一开始就能达到最终系统的要求
 - 要易于进行扩展和频繁改进，因此开发者必须重视演化式原型的设计
 - 仅应该被用于处理清晰的需求、规格说明和技术方案

5.2. 原型的类别

——按照构建技术分类

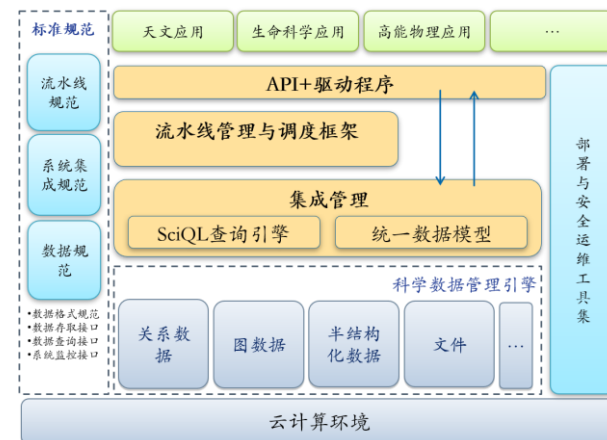
■ 水平原型方法（horizontal prototyping）

- 它仅仅实现选定功能的某些特定层次
- 建立的原型产品称为水平原型（horizontal prototype）
- 要把注意力集中在概括性需求和流程上



■ 垂直原型方法（vertical prototyping）

- 它会触及到选定功能实现的所有层次
- 建立的原型产品称为垂直原型（vertical prototype）
- 要保证真实实现它的各种功能



水平原型系统案例

案例1：校车座位预约系统

■ 水平原型系统

■ 潜在用户范围较大

- 全体北理工（中关村和良乡校区）在校生
 - 包括本科生、硕士研究生、博士研究生
- 全体教师

■ 一时难以确定完整稳定的功能需求

- 涉众范围大
- 类型多样
 - 学生年级分布情况不同
 - 学科不同
 - 教师类型不同
 - ...

案例1：校车座位预约系统

■ 水平原型系统

■ 实现部分概括性的功能

- 用户注册/登录界面
- 班次查询界面
- ...



案例2：校园二手交易平台

■ 水平原型系统

- 一时难以确定完整稳定的功能需求
 - 涉众范围大
 - 全体北理工学生
- 校内还没有类似的二手交易平台
 - 现有的二手平台缺乏管理、分类杂乱、买卖双方都难以找到合适的交易对象
 - 因此本项目定位是新型二手交易平台，卖家可以上传商品信息，买家通过“砍价”功能进行报价，由卖家决定是否出售，平台不参与交易，由买卖双方私下交易。
 - 由于是新型平台，因此项目的功能需求会在项目的推进中不断变化

案例2：校园二手交易平台

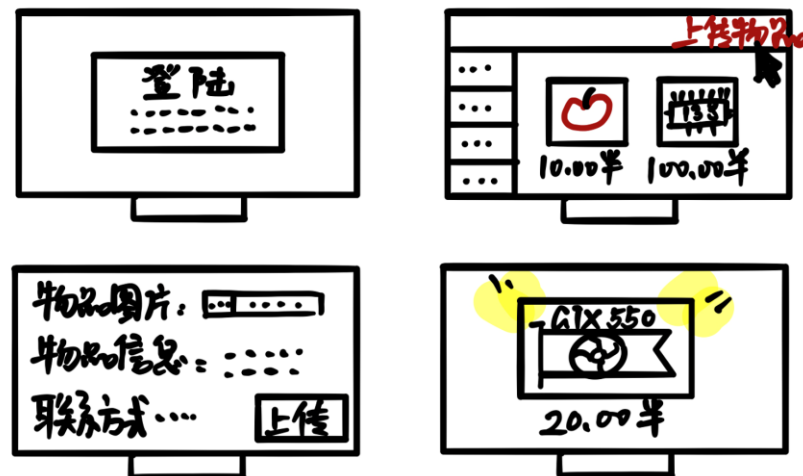
■ 水平原型系统

- 通过绘图的方式描述原型
- 主要关注了项目 workflow 问题的实现

买家用户：



卖家用户：



案例3：在线小导师

■ 水平原型系统

- 潜在用户范围较大
 - 全体在校生
- 功能需求不断修改完善
 - 预约功能
 - 游客浏览功能
 - 对答案/小导师的评价功能
 - 核心功能
 - 提问
 - 问题搜索
 - 回答
 - 私聊
 - ...

案例3：在线小导师

■ 水平原型系统

■ 实现部分核心功能

- 问题提出
- 问题解答
- 私聊
- ...

头像右边将来放搜索题目栏，实现检索功能。

下面每个粉色框对应一个题目，发起提问就是简单的发起提问，点私聊是切换到发起的聊天列表。



垂直原型系统案例

案例1：失物招领系统

■ 垂直原型系统

■ 功能需求明确

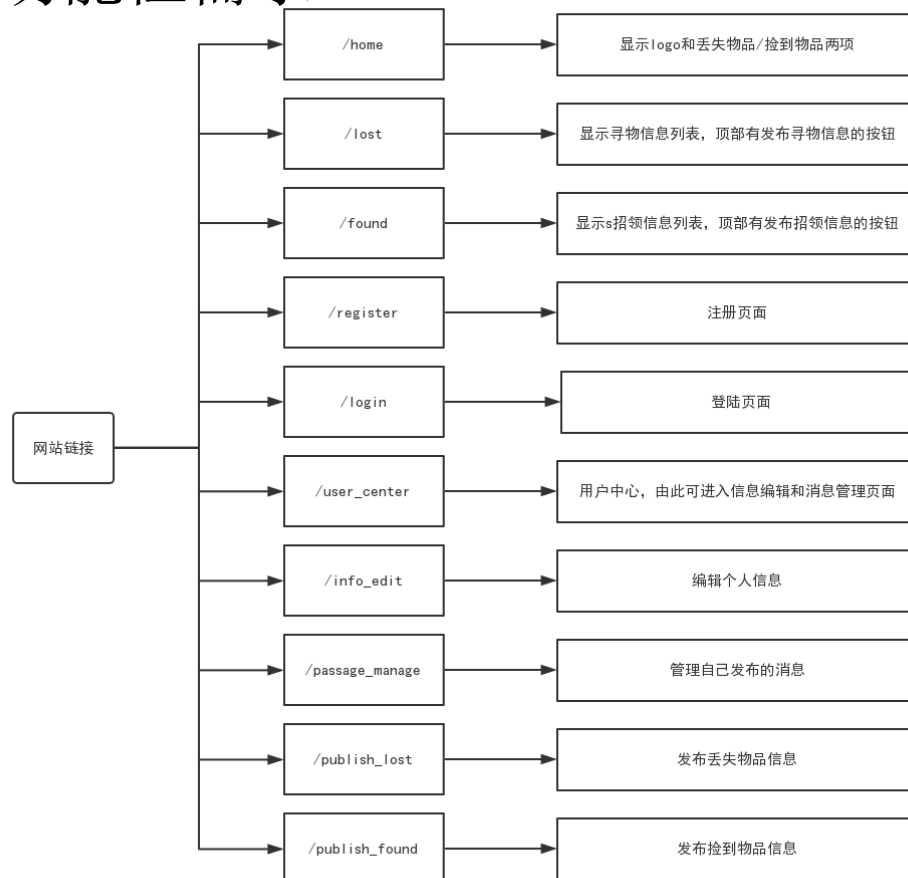
- 浏览他人发布的寻物/招领信息。（无需登录）
- 注册、登陆、编辑个人信息功能。
- 发布寻物/招领信息。
- 对他人发布的信息进行归还物品/认领物品。
- 查看和管理自己发布的消息。

案例1：失物招领系统

■ 垂直原型系统

■ 系统设计涵盖了所有功能性需求

项目总体框架



案例1：失物招领系统

■ 垂直原型系统

■ 项目的部分代码

- 该部分代码定义了项目中各个子页面的链接和名称

```
urls.py
1  # main中的urls.py处理网站的所有链接
2  from django.urls import path
3  from . import views
4
5
6  app_name = 'main'
7
8  # 处理链接跳转的函数写在views.py中
9  urlpatterns = [
10     path('', views.home, name='home'), # 主页
11     path('lost/', views.lost, name='lost'), # 发布寻物信息的页面
12     path('found/', views.found, name='found'), # 发布捡到物品信息的页面
13     path('register/', views.register, name='register'), # 注册页面
14     path('login/', views.login, name='login'), # 登录页面
15     path('user_center/', views.user_center, name='user_center'), # 个人中心页面
16     path('info_edit/', views.info_edit, name='info_edit'), # 个人信息编辑页面
17     path('passage_manage/', views.passage_manage, name='passage_manage'), # 个人消息管理页面
18     path('publish_lost/', views.publish_lost, name='publish_lost'), # 发布丢失物品信息的页面
19     path('publish_found/', views.publish_found, name='publish_found'), # 发布捡到物品信息的页面
20 ]
21
```

案例2：校园广播点播平台

■ 垂直原型系统

■ 涉众范围较小

- 北理工良乡校区学生
- 北理工良乡广播站

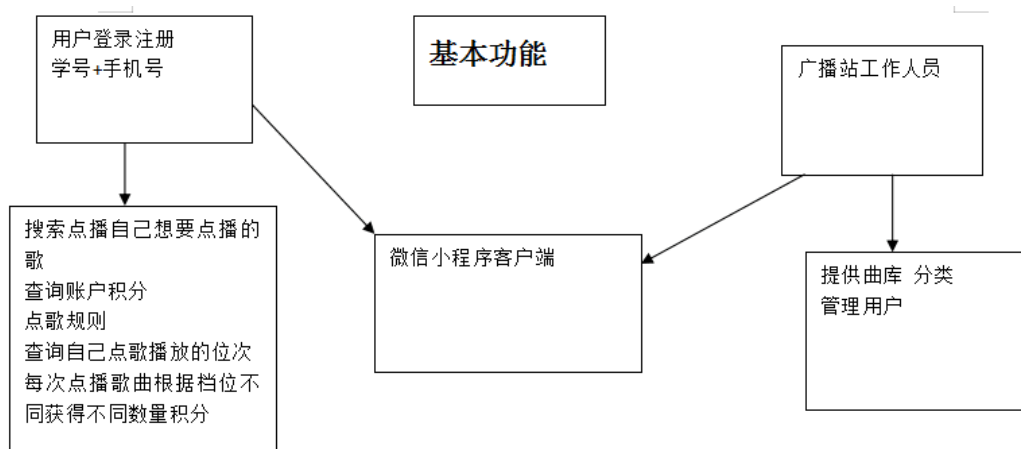
■ 功能需求明确

- 广播站
 - 选取学生提出的话题以及本期节目的外送歌曲
 - 有权利指定歌曲和话题内容
- 北理工良乡校区学生
 - 在曲库选择歌曲，进入播放队列
 - 或者私信管理员话题内容
 - 查看当前播放队列或者历史曲目
 - 查询账户积分和点歌规则（无需登录）

案例2：校园广播点播平台

■ 垂直原型系统

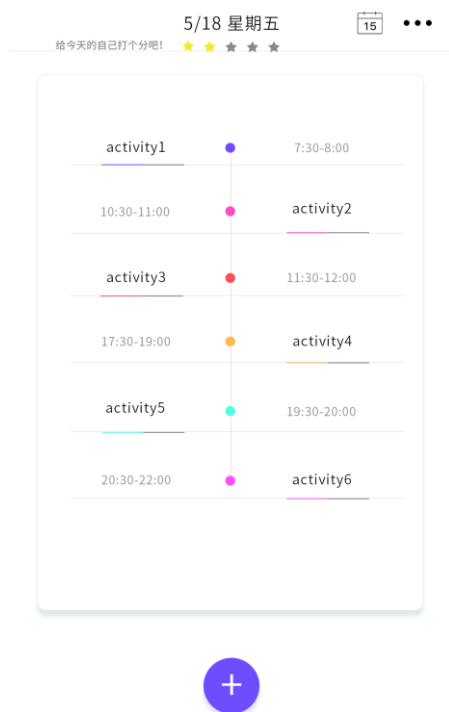
- 通过情节串联图板描述原型
 - 涵盖了系统前后端的功能需求



案例3：时间管理助手平台

■ 垂直原型系统

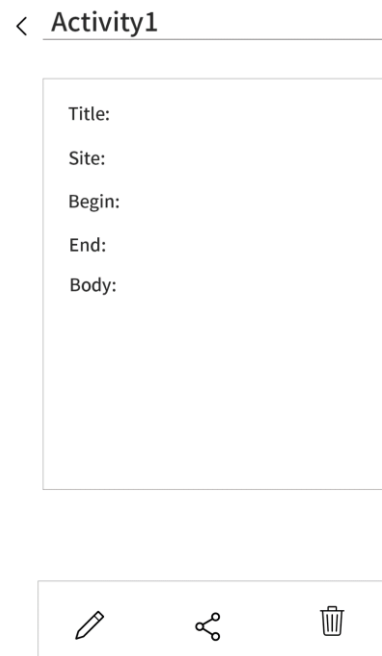
主界面



日历查询



活动详情



活动编辑



5.2. 原型的类别

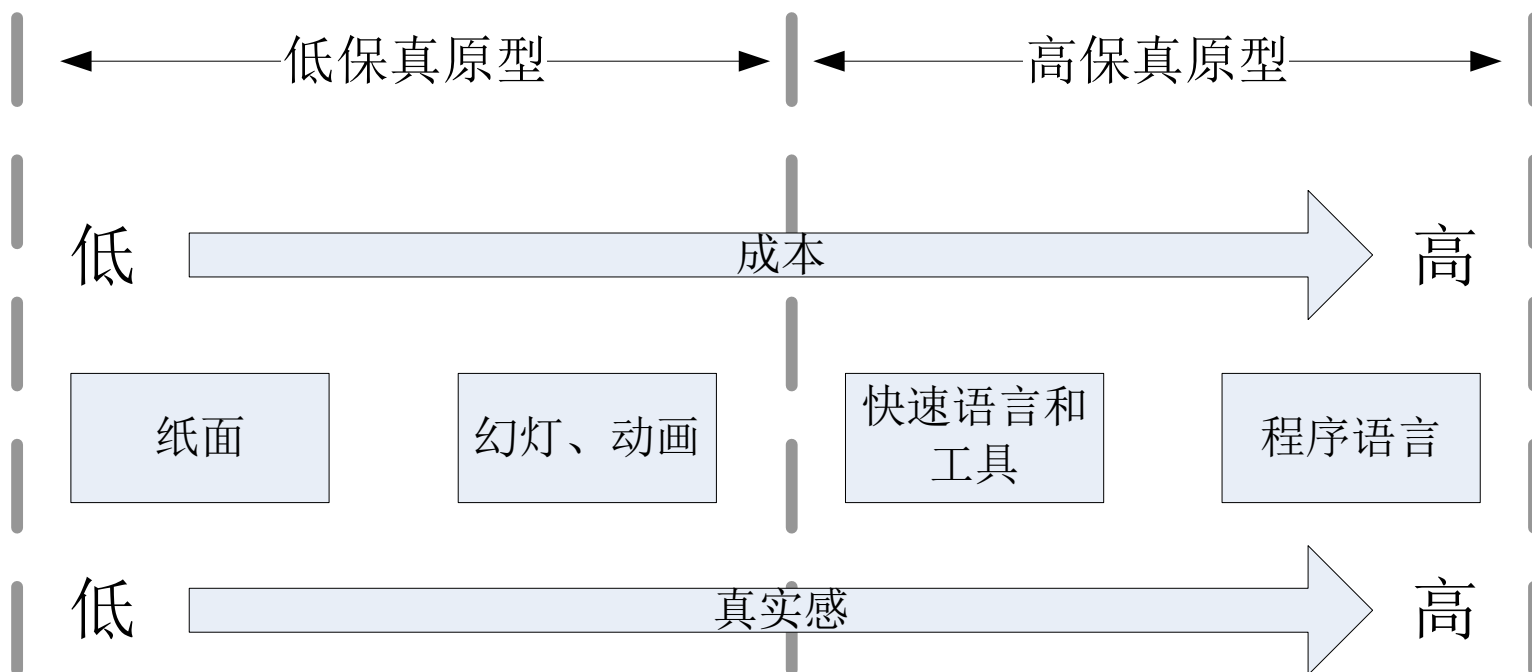
——按照构建技术分类

	废弃型	演化型
水平型	阐明并细化用例和功能性需求 识别遗漏功能 研究用户界面方法	实现核心用例 根据优先级实现其他用例 使得系统适应快速变化的需要
垂直型	演示系统可行性	实现并扩充核心功能 实现并扩充核心算法 测试并调整性能



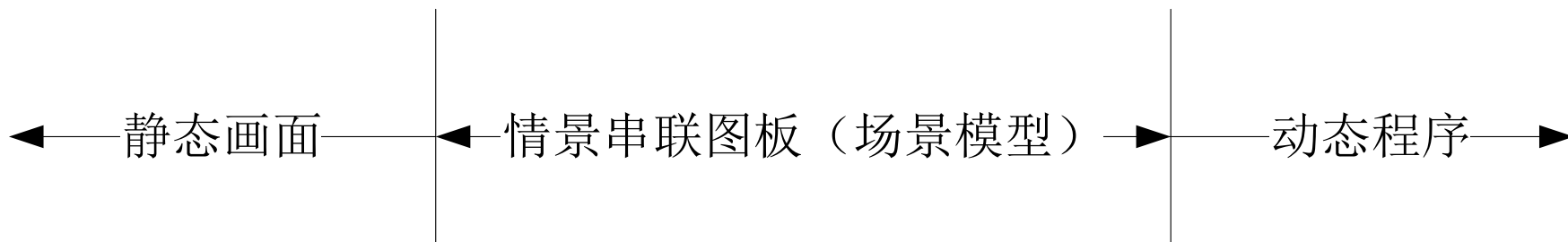
5.2. 原型的类别

——按照介质分类



5.2. 原型的类别

——按照表现分类





用户需要帮助



用户登录



系统提供候选专家



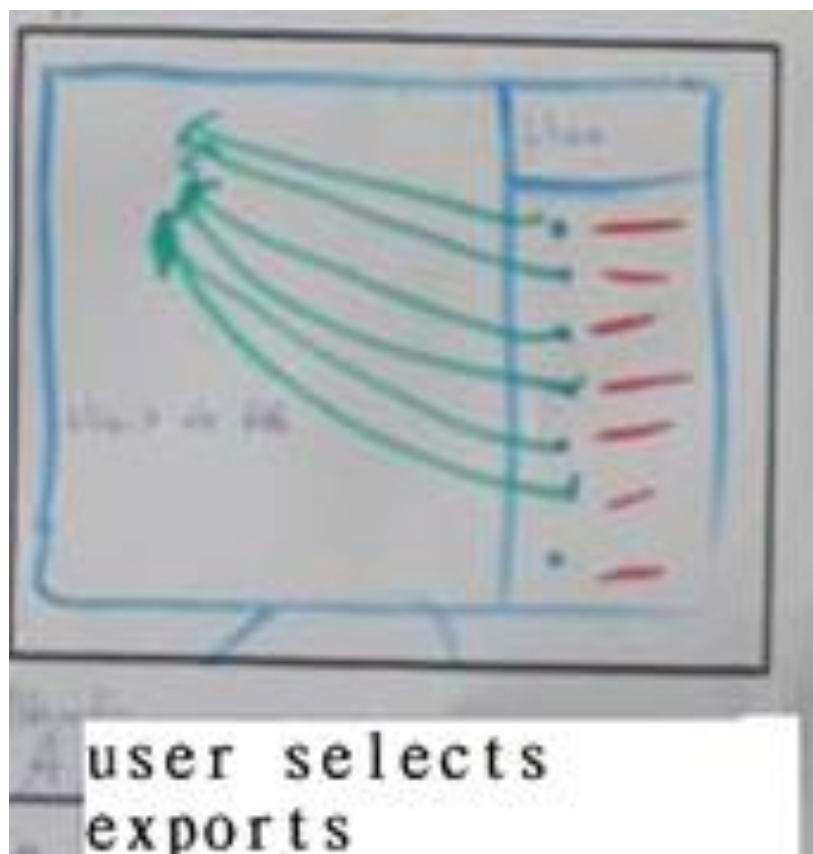
用户探索专家的信息



用户探索专家的因素



列出可选的专家



用户挑选专家



用户与专家合作



得到结果

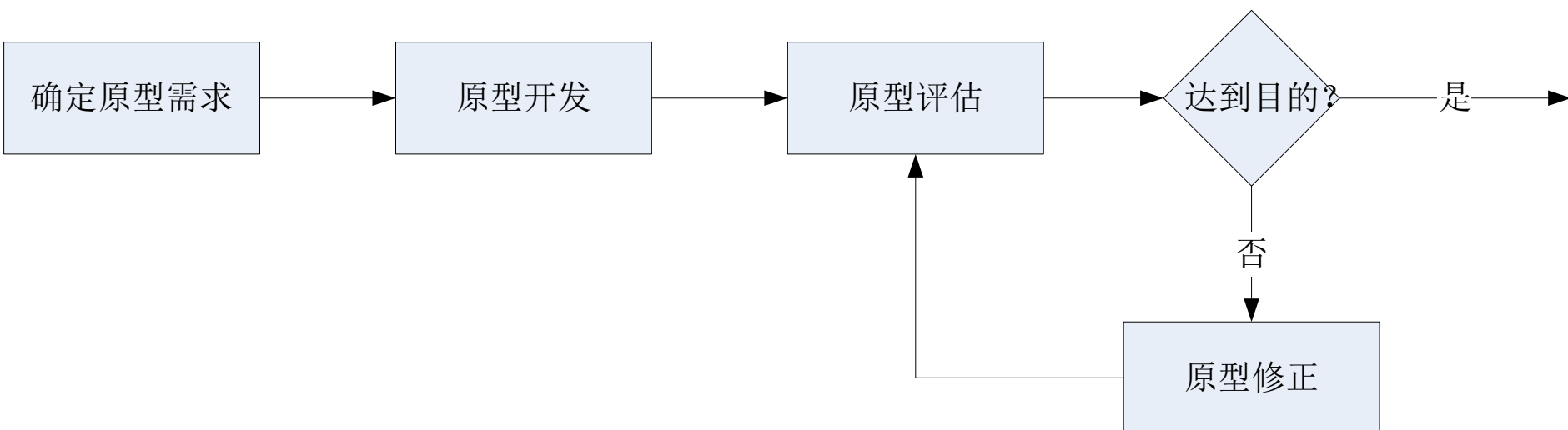
5.1. 应用原型的必要性

5.2. 原型的类别

5.3. 原型方法过程

5.4. 原型方法的风险

3.原型方法过程



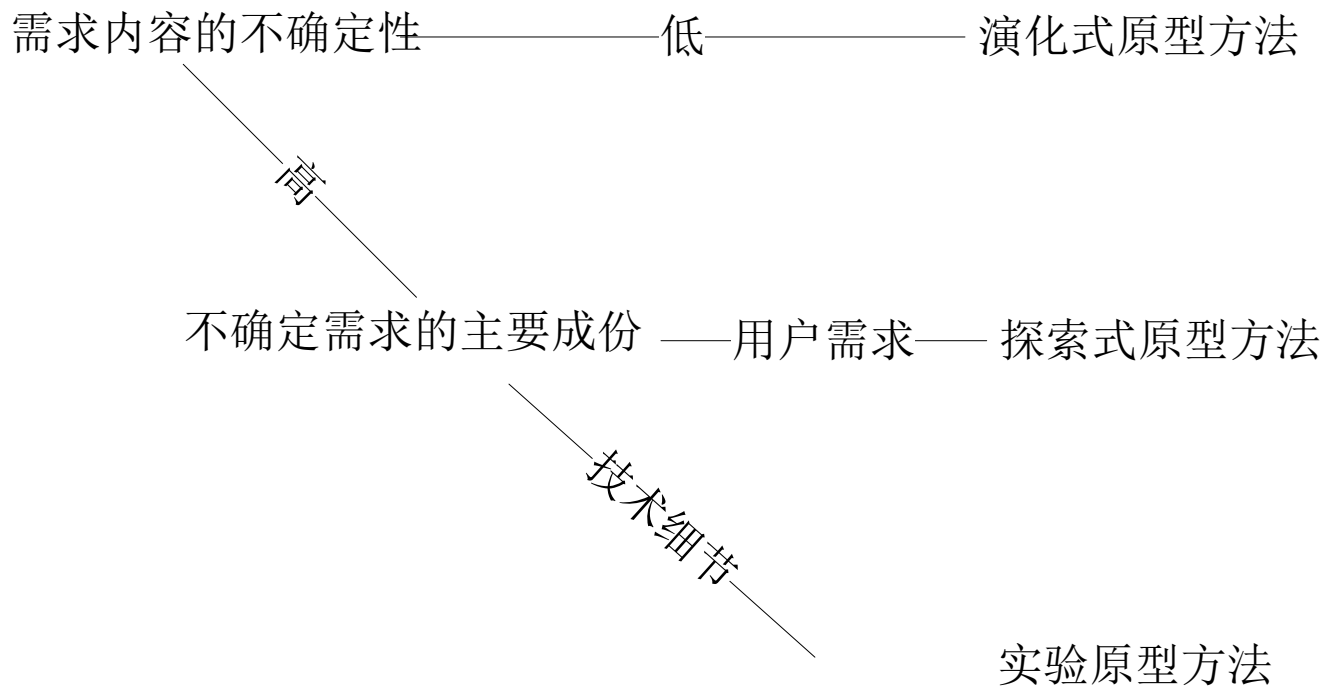
5.3.原型方法过程

——确定原型需求

- 原型的需求内容可以从三个维度上分析：角色（Role）、外观（Look and Feel）和实现（Implementation）
 - 角色是指原型物件在用户工作中的价值，也就是说它为什么是对用户有用的。
 - 原型物件到底能够帮助用户完成什么样的工作
 - 外观是指用户对原型物件的具体感觉体验，即用户在使用原型物件时会看到什么、听到什么和感觉到什么
 - 实现是指原型物件完成功能的细节技术和方法

3.原型方法过程 ——原型开发

■ 选择原型开发的方法



5.3.原型方法过程

——原型开发

- 选择原型的构建技术

需求内容纬度	构建技术
外观	水平
角色	
实现	
集成	垂直



5.3.原型方法过程 ——原型评估

- 需要获取的评估者反馈
 - 评估者反应
 - 评估者建议
 - 创新思想
- 可以创建一些脚本来指导评估者的体验活动
- 务必要让合适的人从恰当的角度来评估原型
- 观察评估人员使用原型的过程
- 创造一个无偏见的环境，让评估人员畅所欲言

5.3.原型方法过程 ——原型修正

- 一方面要依据评估人员的反馈
- 另一方面也要考虑事先的原型调整计划

5.1. 应用原型的必要性

5.2. 原型的类别

5.3. 原型方法过程

5.4. 原型方法的风险

5.4. 原型方法的风险

- 涉众看到了一个正在运行的原型，得出产品几乎已经完成的结论，从而提出快速交付产品的不当要求
- 用户可能会被原型所表现出来的非功能特性遮蔽了眼睛，从而忽略了他们更应该重视的功能特性
 - 在澄清需求不确定性的同时也可能会掩盖一些用户的假设，这些假设将会无从发现
- 原型开发工作投入太多的工作，使得开发团队消耗了过多的时间和过大的成本

实例分析

■ 问题

- 项目使用到了原型。该原型主要用于探索与试验目的，开发速度快，采用了一些技术不成熟的构造技术，原本打算在项目开发时抛弃。但是公司高层领导认为该原型花费了大量人力、物力以及时间，所以不允许抛弃。要求开发团队在该原型基础上继续开发，以达到节约开发成本的目的
- 然而，事实上事情并非之前所预料的那样。由于该原型的质量没有达到最终系统的要求，部分实现细节存在缺陷，所以开发时不得不重新整理该原型，反而增加了项目的时间成本与风险。

第5节小结

- 原型是软件开发当中消除不确定性风险的有效工具，是一种有效的需求获取方法
- 原型的体系是复杂的，不同类型的原型具有不同的作用和创建要求，实践当中应该综合考虑各种应用因素选择合适的类别
- 一个完整的原型方法过程可以帮助更有效的应用原型方法
- 原型方法的应用可能会给项目带来相应的风险，需要妥善的加以解决

本章小结

- 需求获取是一个困难和复杂的任务
- 实践调查情况表明，需求获取活动是一个具有挑战性的任务
- 需求获取的成功执行需要有效组织子活动过程
 - 项目前景和范围
 - 涉众分析和硬数据采集
 - 面谈
 - 原型