



行为驱动开发

清华大学软件学院 刘强





1

再认识软件工程

2

项目二产品定位

3

行为驱动开发

4

软件界面与交互设计



1



- 项目一引发的思考
- 项目一经验总结
- 软件工程真正含义
- 软件开发基本策略

项目一引发的思考

为什么项目开发过程变得不可控？

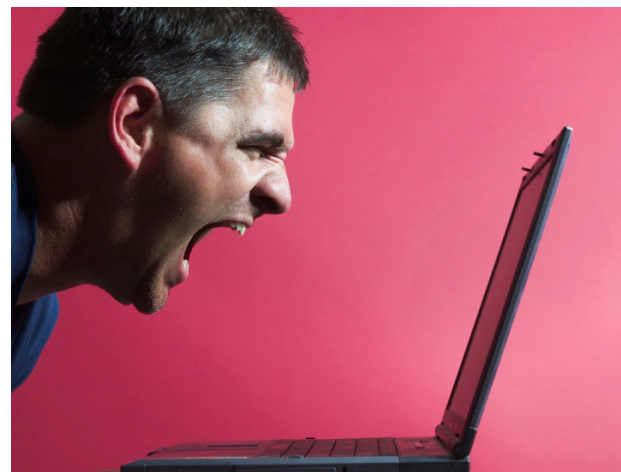
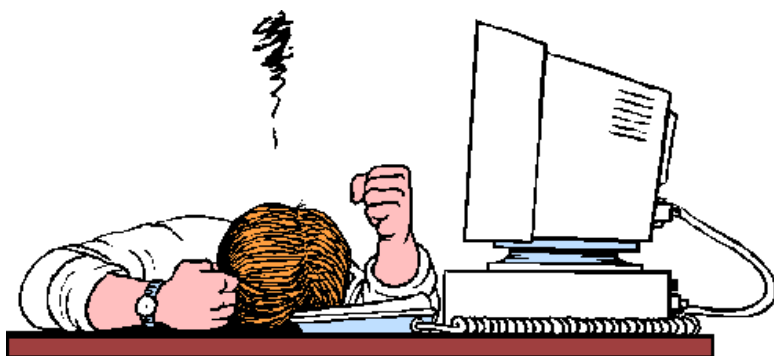


项目开始



项目交付

项目一引发的思考



项目一引发的思考



项目初期定义的目标是什么？

项目交付时是否实现了原定目标？

什么原因影响了项目目标的实现？

1

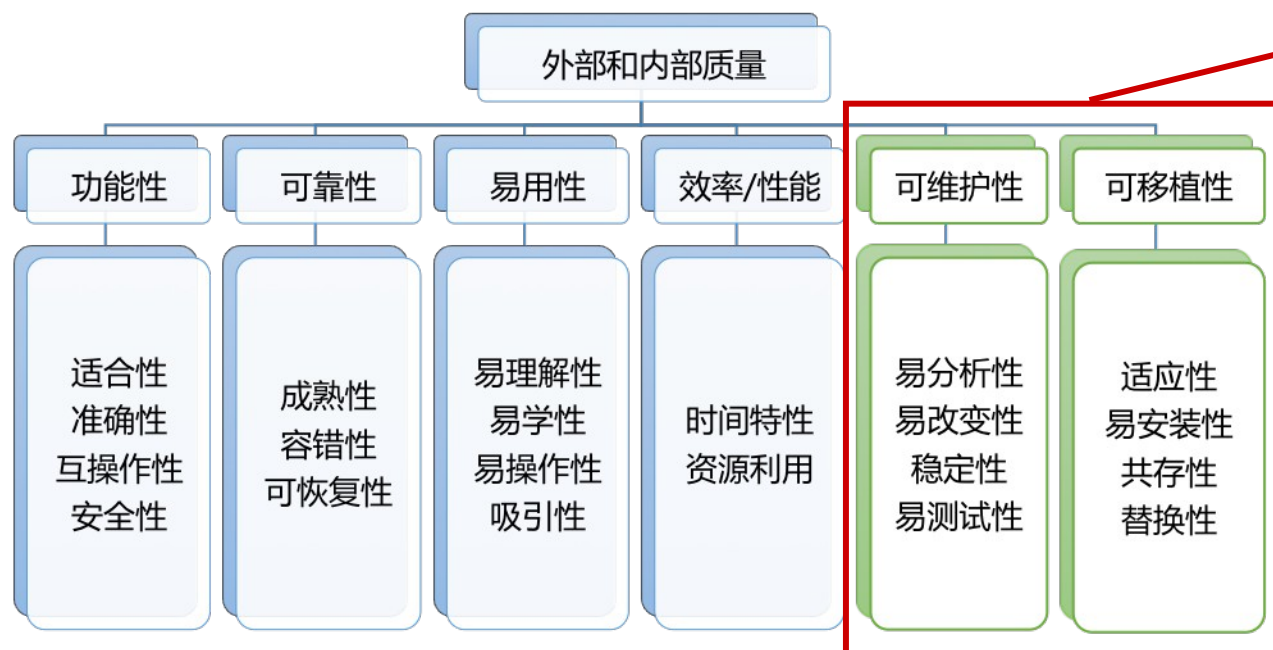
[illegible][illegible]

重构让人快乐让人苦

- 重构的代价惨重
- 重构要避免过度设计，应围绕一个适度的目标来进行，要考虑代价，在重构过程中时时考虑是否把简单的事情变复杂了
- 重构过程中应注意扩展性，一方面为后续功能添加预留好接口，另一方面要进行更进一步的抽象
- 重构是编程工作的一部分，同时也应该是程序员乐于实践的一项内容

项目一经验总结

质量就是效益，质量就是生命



编写优美的代码



[illegible]

项目一经验总结



联想集团
互联网产品经理

软件工程是一种手段，它通过一系列制度和管理的最佳实践，筛选磨合出来一支高效且有凝聚力的团队，进而齐心完成一件伟大的作品。能够和最优秀的游戏制作人以及一帮志同道合的追梦人，制作一款伟大的原创游戏，是人生之幸事。

• □ □ □ □ □ □ □

• □ □ □ □ □ □ □ □ □ □ □ □

• □ □ □ □ □ □ □ □

• □ □ □ □ □ □ □ □ □ □ □

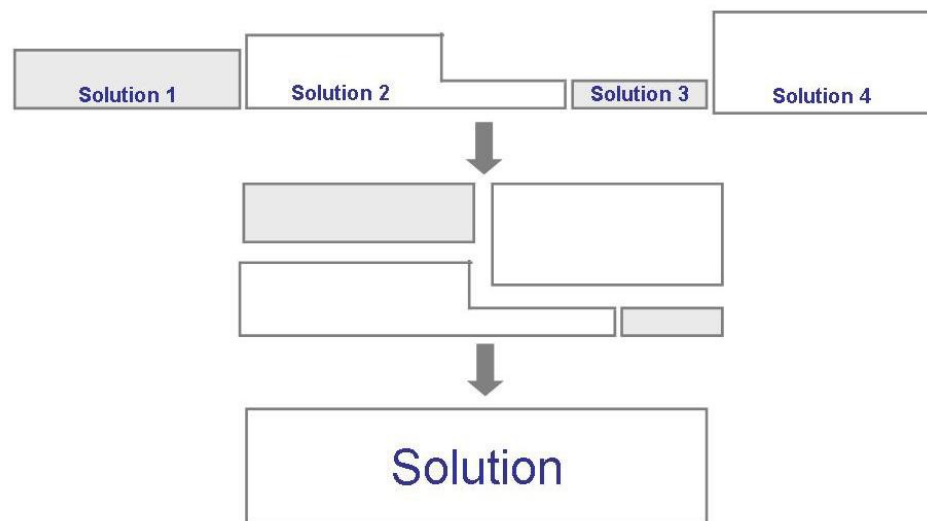
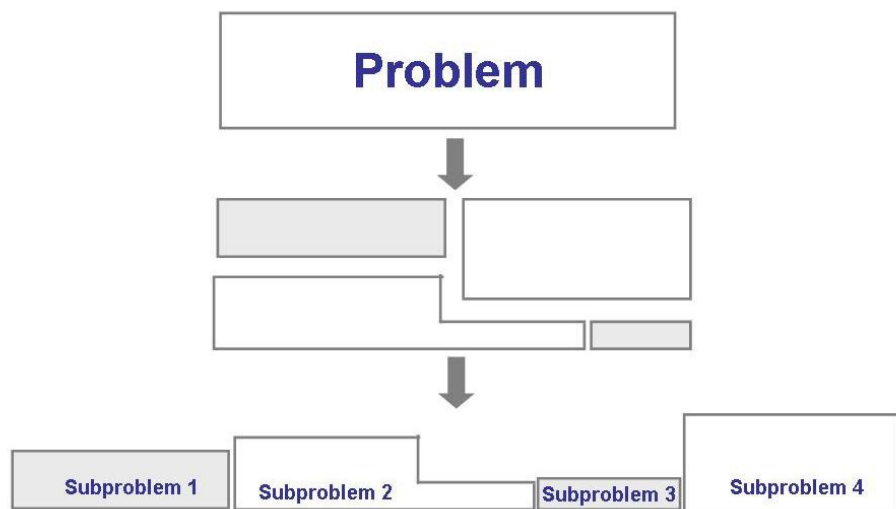
• □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

• □ □ □ □ □ □ □



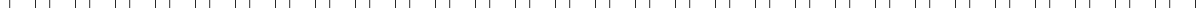
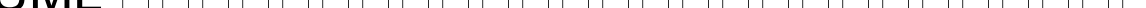


回顾：软件工程概述

软件工程是一项解决问题的工程活动，通过分析对问题进行研究，将问题分解成可以理解并能够处理的若干小部分。



软件工程是一项建模活动，通过抽象找到事物的重要特征而忽略非本质的细节，从不同侧面建立系统模型，有效地简化和处理复杂性。

一张图胜过千言万语！

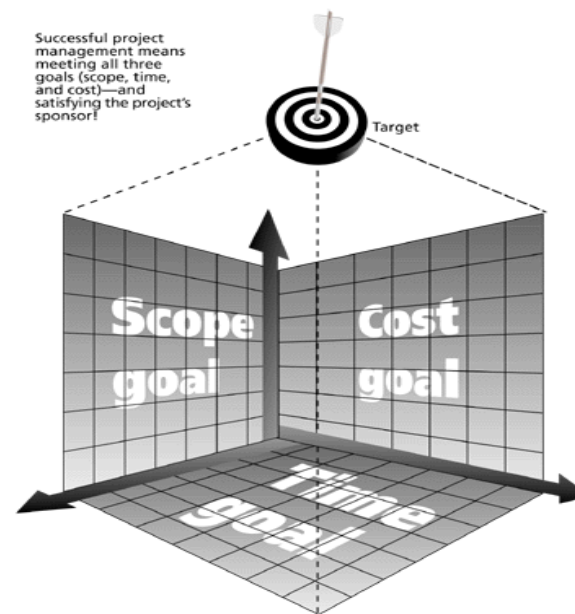
- UML  70%
- 
- ER  -  用于建立数据模型，它是关系数据库设计的一种工具

回顾：软件工程概述

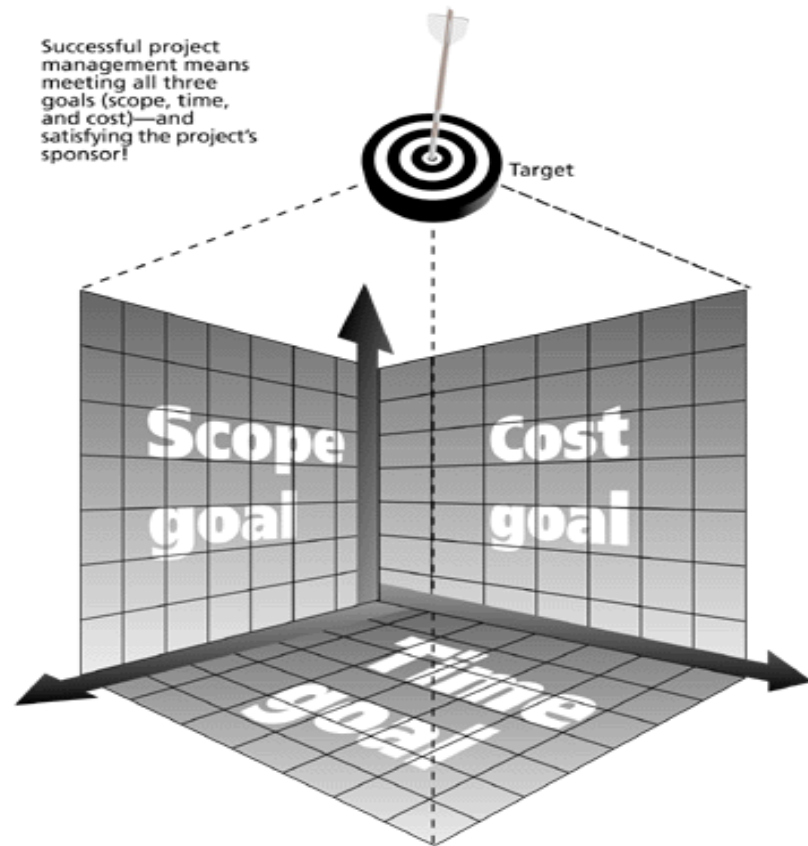
软件工程是一项受软件工程原理指导的活动，软件工程师需要捕捉和理解一个系统的基本原理模型，并根据评价标准提出合理的开发决策。

□□□□□□□□□□□□□□

- □□□□□□□□
- □□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□



项目成功的决定因素



项目成功的决定因素：

- 时间、成本、质量和客户满意等因素相互关联
- 保证四个因素的平衡是项目成功的关键

项目成功的衡量标准：

- 平衡四要素—多快好省（范围、时间、质量、成本）
- 平衡干系人—客户满意和团队成长
- 战略上平衡—打造可持续发展的学习型团队

回顾：软件的本质特性



软件开发的基本策略



分而治之

- 将一个复杂的问题分解成若干个简单的问题，然后逐个解决
- 来源于人们生活与工作的经验，完全适合于技术领域

逐步演进

- 软件开发是自底向上逐步有序的生长过程
- 小步快跑：每走完一步再调整并为下一步确定方向，直到终点

优化折中

- 软件开发管理是一个多目标最优化决策过程
- 折中策略：不能使某一方损失关键的职能，更不可以随意丢弃一方

创新设计

- 软件开发自始至终都是一种创造性的活动
- 创新思维：在已有知识基础上从某种事实中寻找新关系，找出新答案

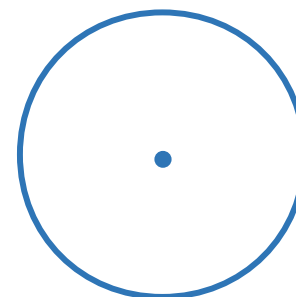
突破思维定势，扩展思维视角



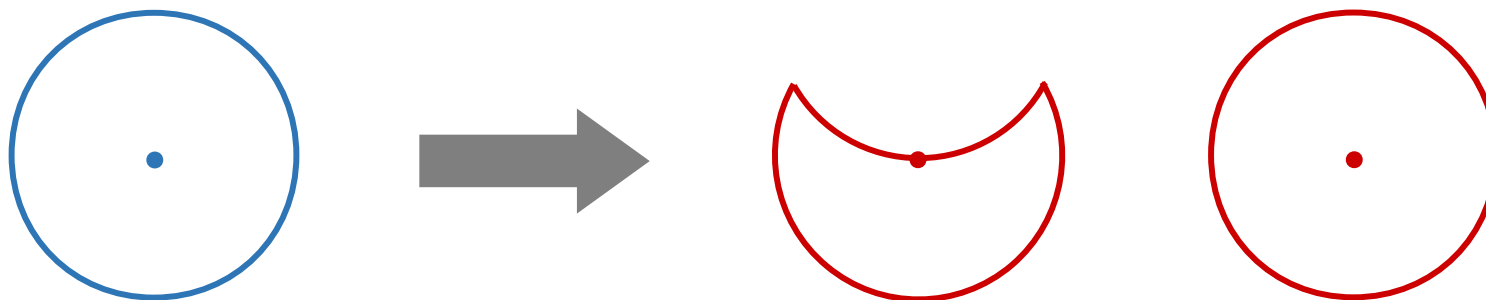
□ 1 □ 62-63=1 □

□ □ □ □ □ □ □ □ □ □

□ 2 □



$$62-63=1 \quad \longrightarrow \quad 2^6-63=1$$



坐完地铁出站时，对于楼梯跟电扶梯大家通常都选电梯，但走楼梯有益健康

。



怎么样可以让人自动
走楼梯呢？

创造性思维



在瑞典斯德哥尔摩一个地铁站，有人发挥创意，即将楼梯装饰成钢琴键盘并发出钢琴声音，让人爬楼梯变得就像在音乐演奏一样，相当有意思。



安装了钢琴楼梯之后，
走楼梯的人增加了 66%

2



- 校团委提出的要求
- 软件需求层次
- 干系人识别和分析
- 项目二产品定位

微信公共账号（自定账号名）

[illegible]

■**地点查询**：输入搜索地点，可以返回地图位置以及该地点的图片。比如搜索“二校门”，可以返回“二校门”的地图位置和相关图片。

■**智能问答：**对于初到清华的新生，肯定有些常见的问题，比如怎么选课、寝室怎么申请上网等，助手可以较智能地回答这些问题。

■**人工服务**：这个人工服务的参与者既是新生，也应该是热心的学长学姐。新生在遇到自动回答处理不了的问题时，可以选择人工求助。新生可以使用文字或语音提问，学长学姐在空闲时可以主动去访问平台，系统返回一个求助任务列表，学长学姐可以选择解答。之后，新生可以查看自己提交的问题并听取解答，对解答满意之后可以关闭该问题。这个类似很多网站上的在线客服，不过本项目利用微信平台来完成，而且客服人员变成了”学长学姐“。

■其他功能：可选项，各□□□自由发挥提出□□□□□□。

- 敢想、敢干
- 创造力
执行力

校团委提出的要求



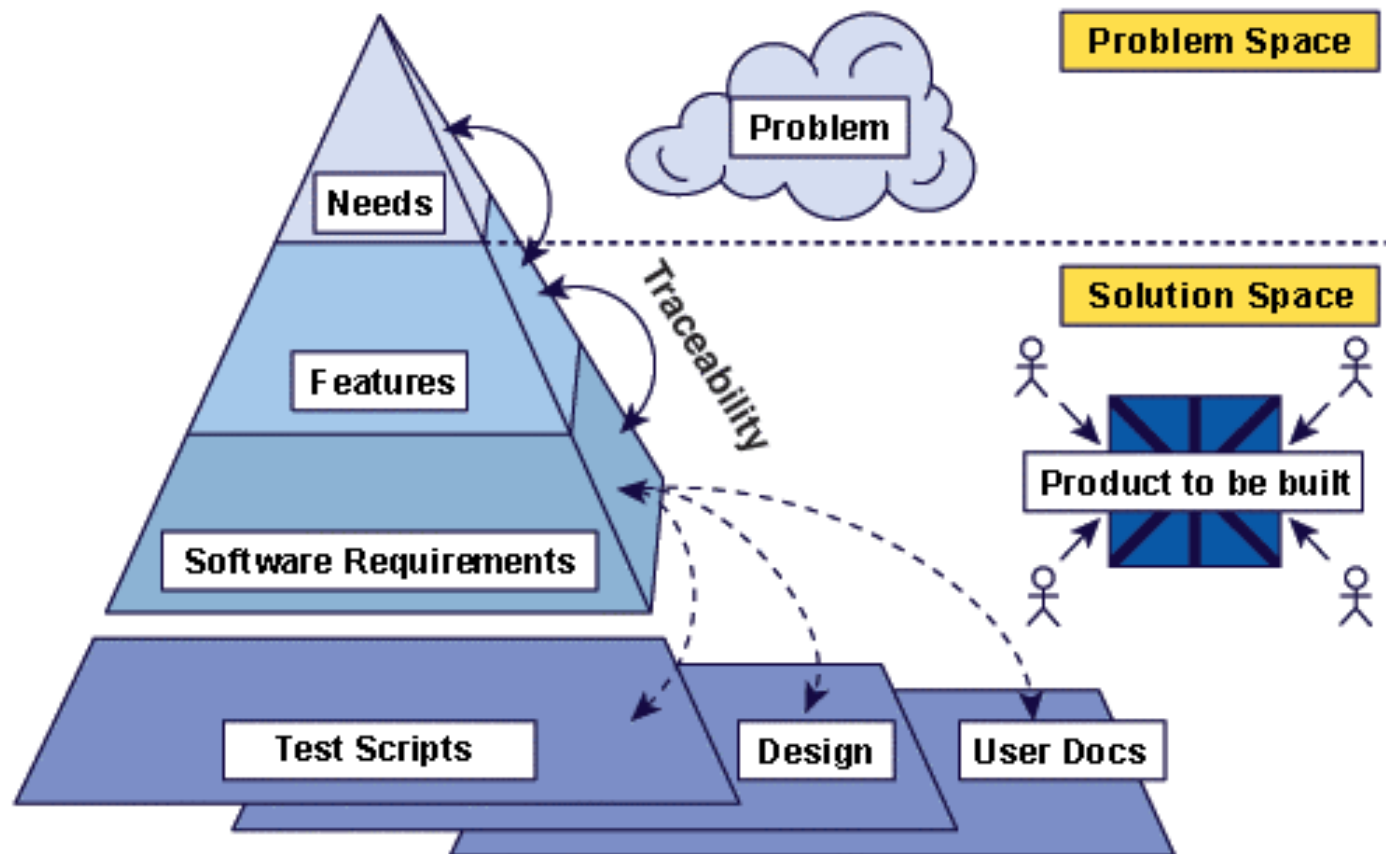
服务号可以实现的基本功能（包括但不限于此）：

- 搜索引擎式的校内资讯查询服务。例如：输入“社团部”，会返回社团部相关介绍；输入“吉他协会”，会返回吉他协会相关介绍；输入“时事大讲堂最新讲座信息”，会将时事大讲堂的最新活动信息返回给同学。
- 热门活动的抢票服务。例如：除排队领票外，可以开放一部分门票给线上抢票，用户根据学号和姓名等信息进行抢票，后台立即提示是否抢票成功，之后再开放一个集中时间段供同学凭学生证来领票。
- 大家可以录入自己的院系、年级、社团、兴趣爱好等信息，基于这些信息进行个性化的信息推送，并根据这些信息调整查询结果的内容顺序。
- 根据大家的点击行为、查询行为猜测大家的兴趣，并调整推送信息的内容和查询内容的排序。
- 了解不同兴趣点的相关性，了解不同活动参与人群的分布，为活动组织提供数据指导，为新的活动与资讯定点推送到热点人群。

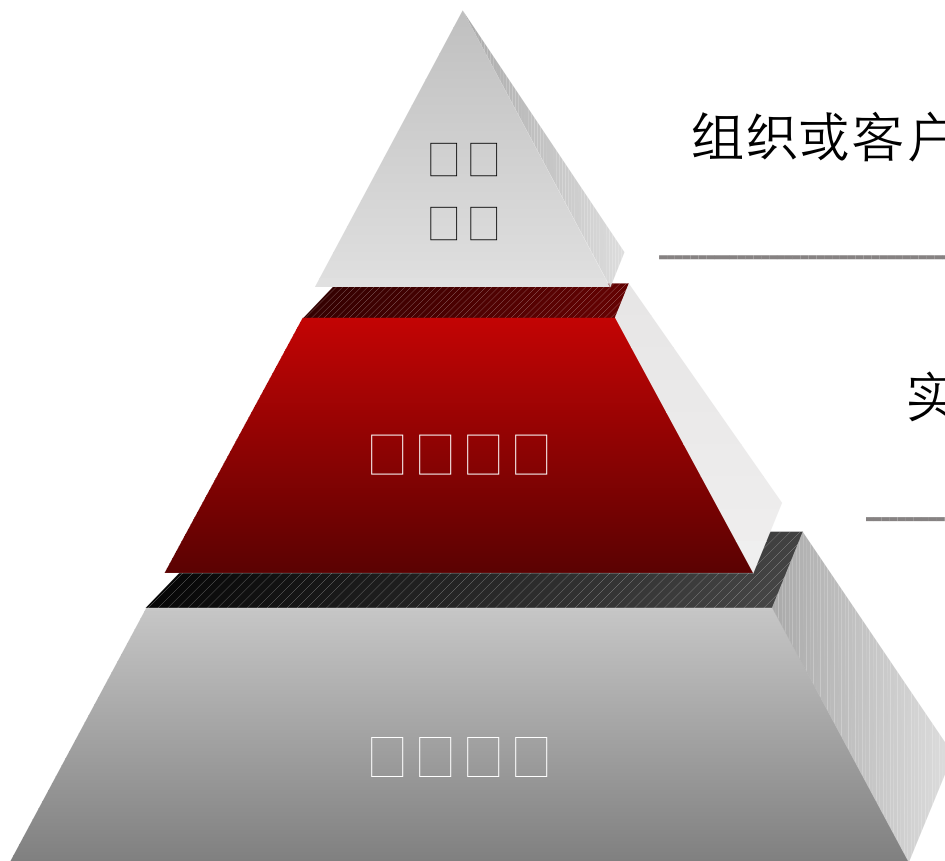
[illegible][illegible]

● □□□□□□□□□□□□□□□□ “□□□□□□□□□□□□□□□” □□□□□□□□□□□□□□□□□□□□□□□□
 □□□□□□□

软件需求层次



软件需求层次

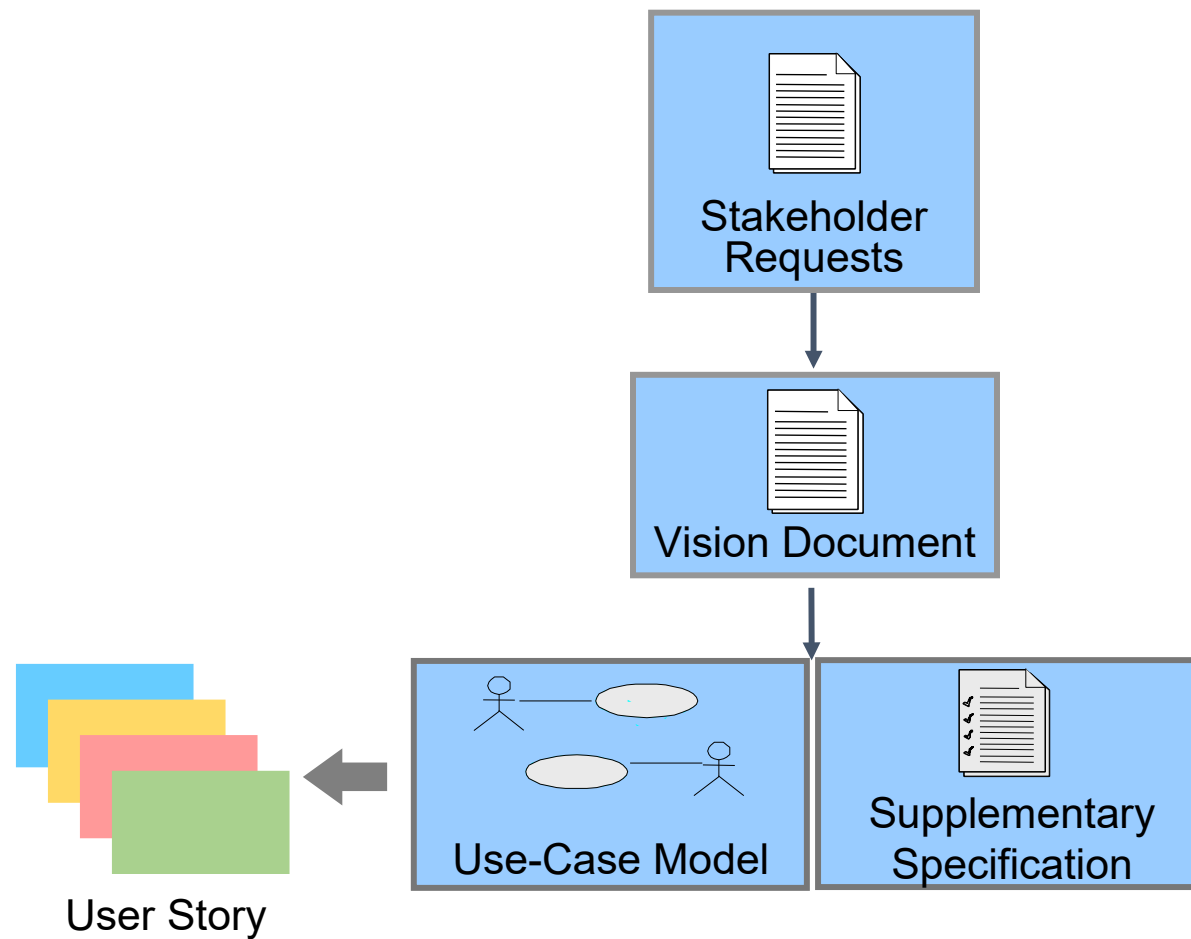
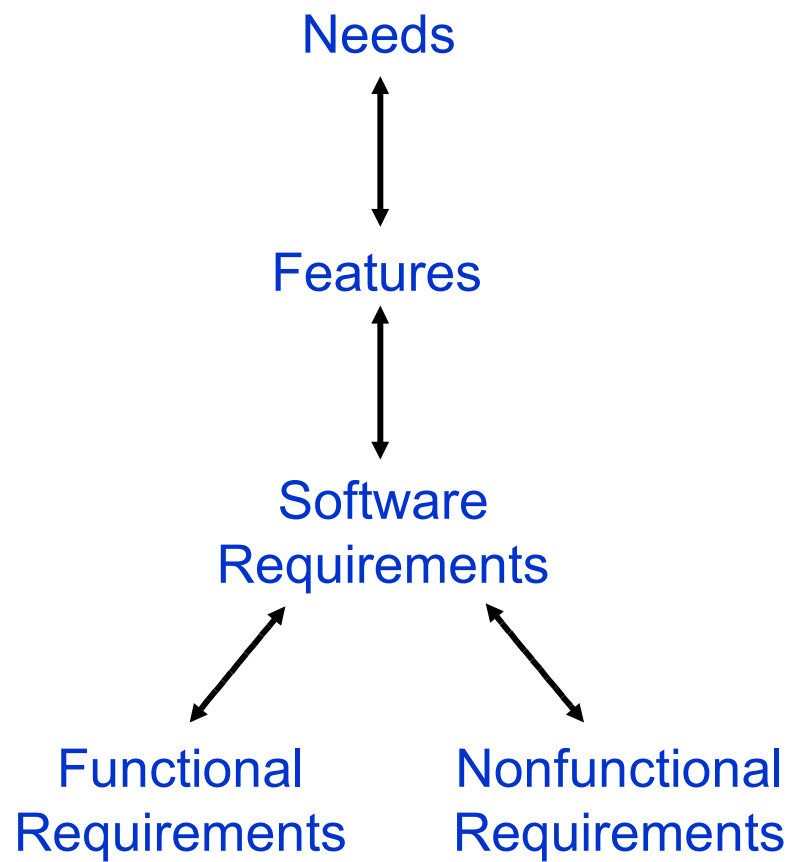


组织或客户对于系统的高层次目标要求，描述了组织为什么要开发系统

实际使用系统者对系统所能完成的具体任务的期望，描述了系统能够帮助用户做什么

系统将要提供的功能或服务以及系统所受到的约束和限制

软件需求层次





项目二有哪些重要的干系人？

这些干系人具备什么特征？

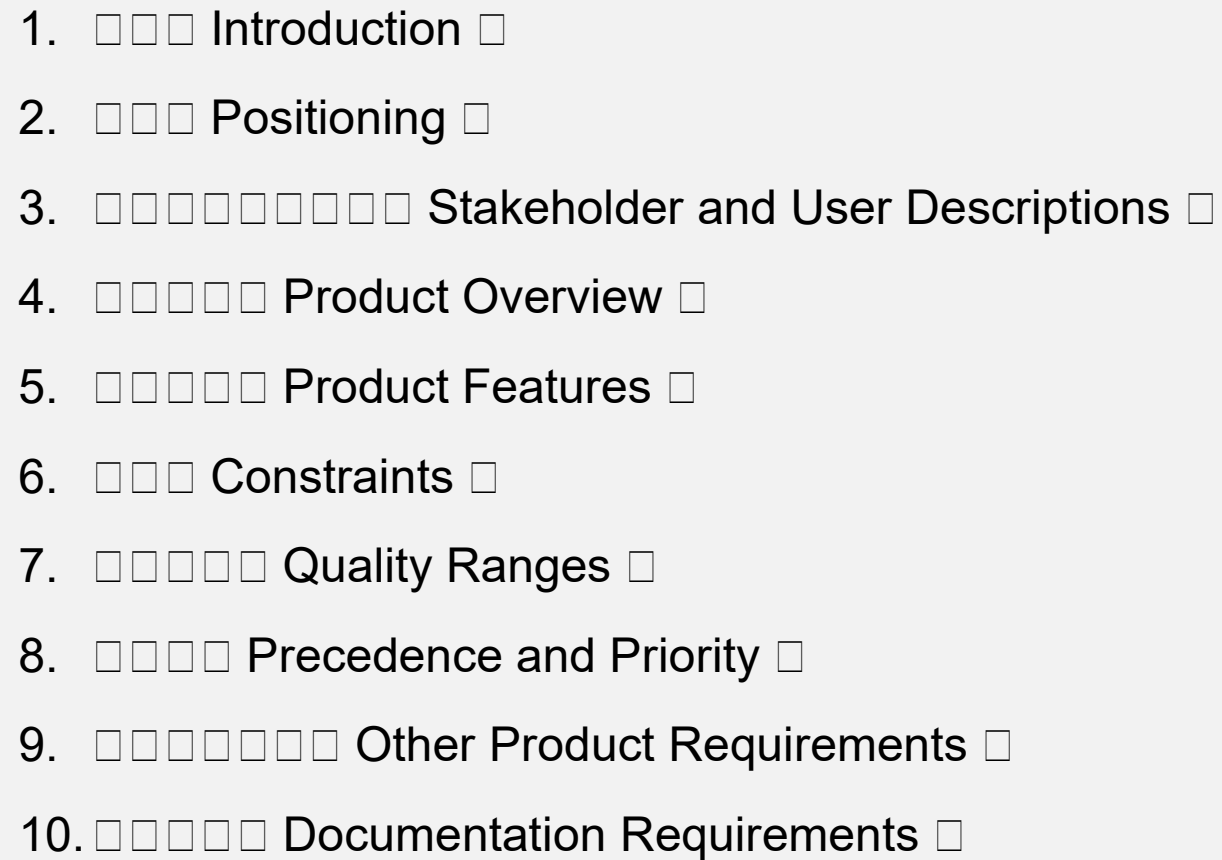
他们对系统的期望和要求是什么？

33

- Info

34

Vision □ □

- 
1. □ □ □ Introduction □
 2. □ □ □ Positioning □
 3. □ □ □ □ □ □ □ □ □ Stakeholder and User Descriptions □
 4. □ □ □ □ □ Product Overview □
 5. □ □ □ □ □ Product Features □
 6. □ □ □ Constraints □
 7. □ □ □ □ □ Quality Ranges □
 8. □ □ □ □ Precedence and Priority □
 9. □ □ □ □ □ □ □ Other Product Requirements □
 10. □ □ □ □ □ Documentation Requirements □

3

□ □ □ □ □ □

- □ □ □ □ □ □ □
- □ □ □ □ □ □ BDD □
- Cucumber □ □
- □ □ □ □ □ □

我们如何做事



英国某家报纸曾举办一项高额奖金有奖征答活动，题目如下：

在一个充气不足的热气球上，载着三位关系世界兴亡命运的科学家。第一位是环保专家，他的研究可拯救人类因环境污染而面临死亡的厄运；第二位是核子专家，他有能力防止全球性的核子战争，使地球免于遭受灭亡的绝境；第三位是粮食专家，他能在不毛之地，运用专业知识成功地种植食物，使几千万人脱离饥荒而灭亡的命运。

此刻热气球即将坠毁，必须丢出一个人以减轻载重，使其余的两人得以活存，请问该丢下哪一位科学家？



Doing The Right Things !
Doing The Things Right !

—— Peter F. Drucker

选择了错误的方向而又正确地执行它，将是一件可怕的事情！

您是想完成一个更有价值的交付产品呢，
还是只想完成作业要求？！

正确的
软件



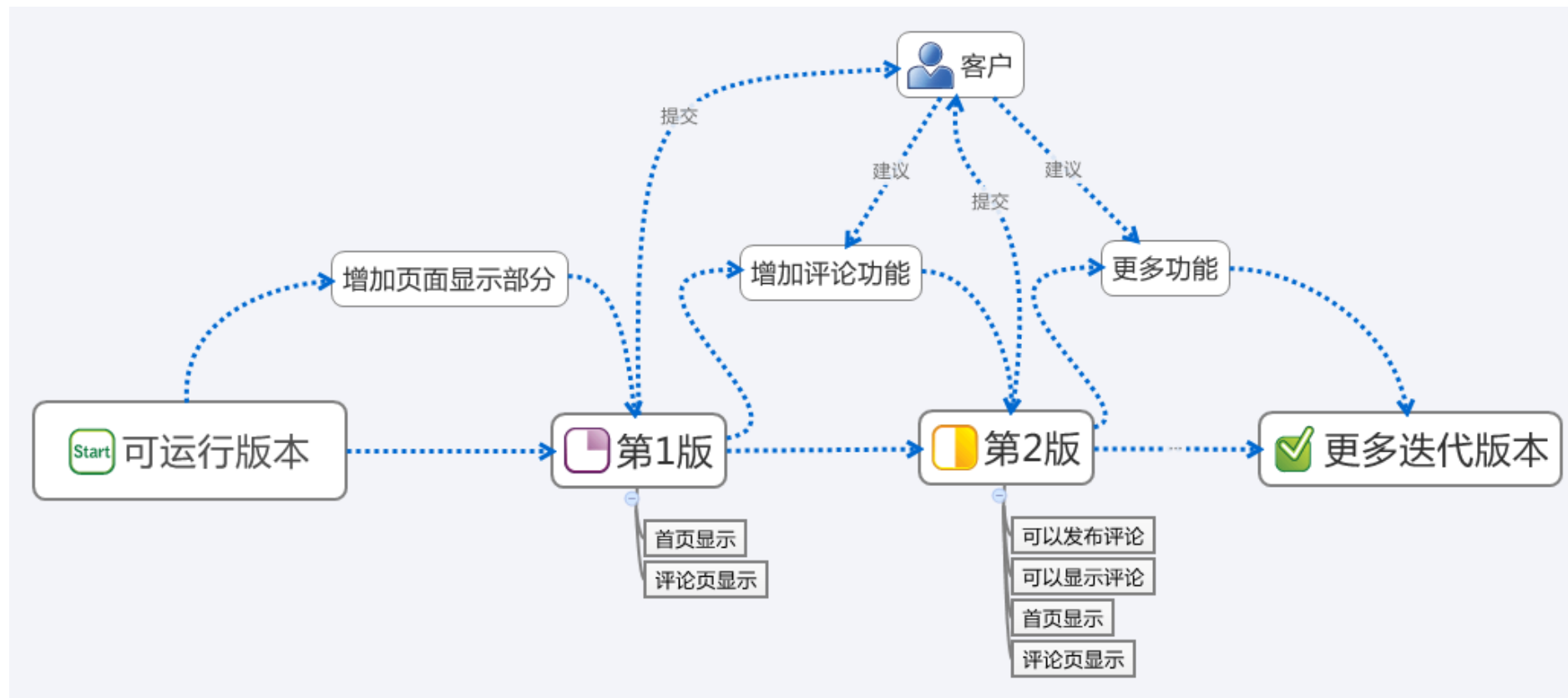
□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□。

软件运行
正确



□□□□□□□□□□□□□□□□□□□□□□□□□□□□

回顾：敏捷开发方法



有效交流

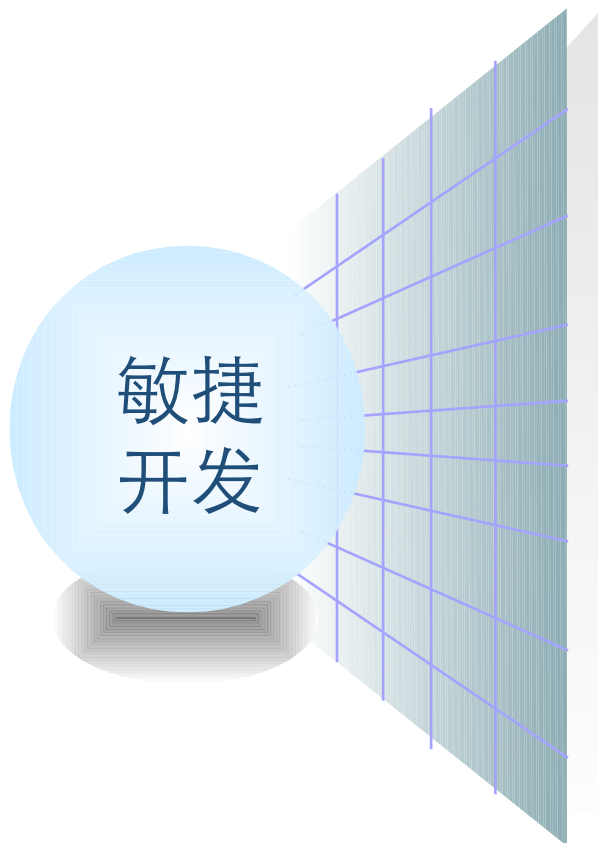
迭代开发

持续交付

简洁为本

自组织团队

回顾：敏捷开发方法



用户故事

结对编程

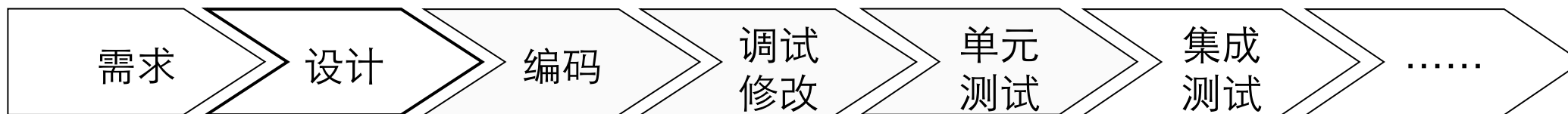
测试驱动开发

持续集成

[illegible]

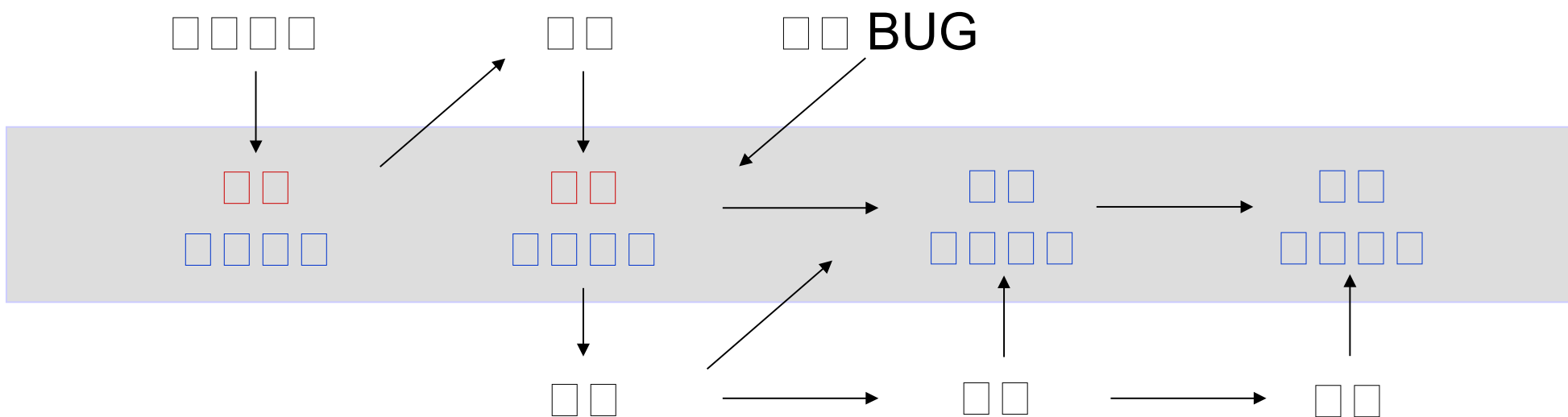
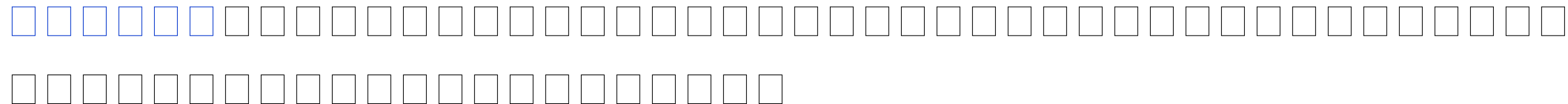
传统的开发方法

软件开发过程：先设计，再实现，最后测试



迫于时间压力，程序员往往会简化或者不执行单元测试





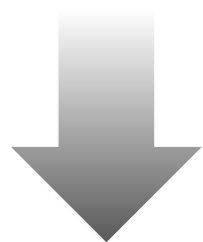
测试驱动开发过程

- 对要解决的问题建模，编写测试代码
- 编写解决问题代码，并使其通过测试
- 思考意外情况，编写额外的测试
- 解决意外情况，使测试通过
- 重复上述流程，直到想不到意外



- 46

定义系统的行为才是测试驱动开发的真正价值！



为什么不把测试与设计
更紧密地结合起来？

□□□□□□□ Behavior-Driven Development □ BDD □□测试驱动开发的进化，但关注的核心是设计，它鼓励软件项目中开发者、测试人员和非技术人员或商业参与者之间的协作。

行为驱动开发



□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

- □ □ □ □ □ □ □ □ □
- □ □ □ □ □ □ □ □ □ □ □ □ □ □
- □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □
- □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □
- □ □ □ □ □ □ □ □ □ □ □



□ □

通用语言



□ □ □



系统
行为

在行为驱动开发中，我们需要使用一种通用语言来定义系统行为。而通用语言实际上是一个最小化的词汇表，我们使用这些词汇来书写故事。

注意：词汇表的词汇必须具有准确无误的表达能力和一致的含义

例如：“系统”和“自动提款机”哪一个更符合要求？

故事是对系统某一方面的行为描述，它具有特定的格式，为接下来检验所实现的代码提供一个规范。

故事的书写格式

Story: □□□□□□□□□□□□

As a □□□□

I want □□□□

So that □□□□

Scenario 1: □□□□□□□□□□□□

Given □□□□□

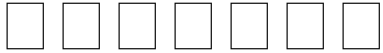
And □□□□□□□□.....

When □□□□

Then □□□□

And □□□□□□.....

用一系列场景
定义验证标准



Story: Customer withdraws cash

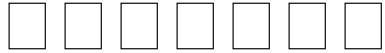
As a customer,

I want to withdraw cash from an ATM,

so that I don't have to wait in line at the bank.



顾客的取款行为可以用哪些
具体场景进行描述?



Scenario 1: Account is in credit

Given the account is in credit

And the card is valid

And the dispenser contains cash

When the customer requests cash

Then ensure the account is debited

And ensure cash is dispensed

And ensure the card is returned

取款金额比实际存款少

Scenario 2: Account is overdrawn past the overdraft limit

Given the account is overdrawn

And the card is valid

When the customer requests cash

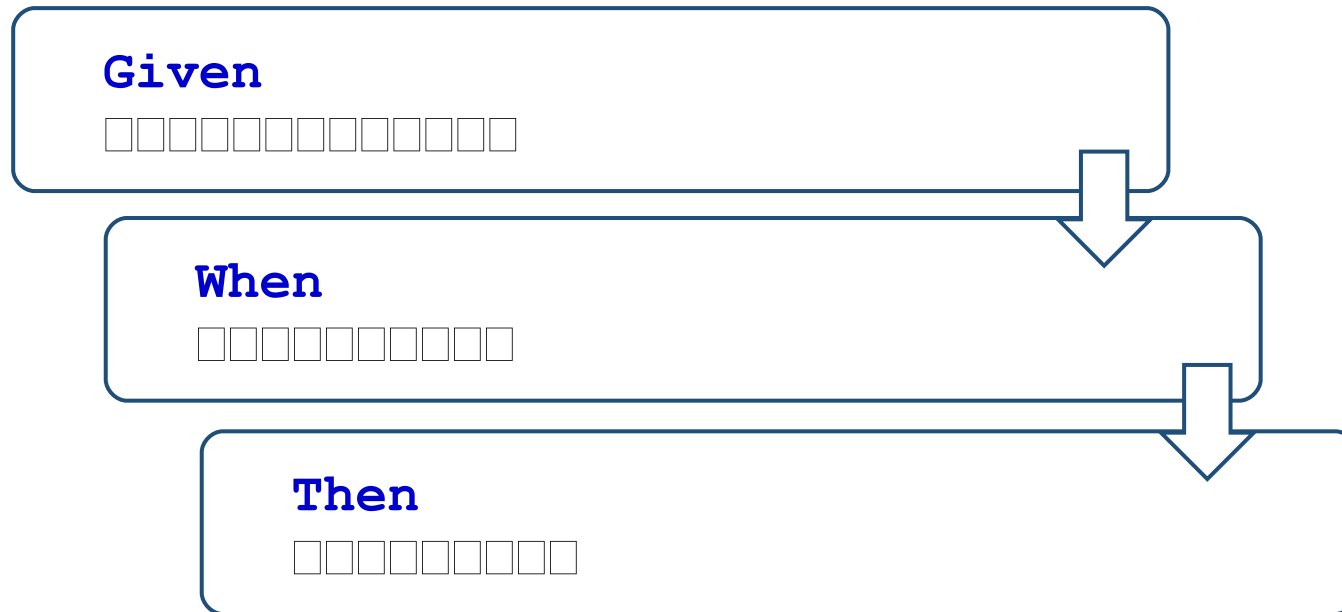
Then ensure a rejection message is displayed

And ensure cash is not dispensed

And ensure the card is returned

取款金额比实际存款多

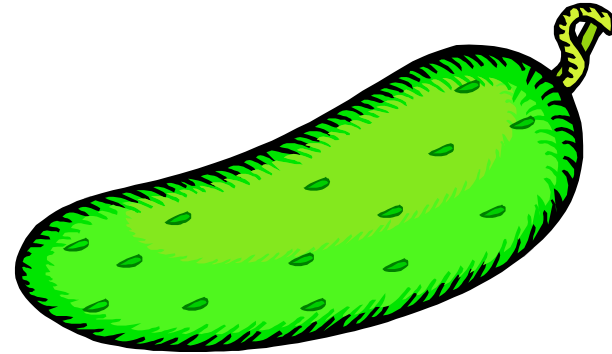
Given ... When ... Then □ □ □ □ □ □ □ □ □ □



□ □ □ □ □ □ □ □ □ □ **BDD** □ □
□ □ □ □ □ □ □ □ □ □ □ □ □ □
□ □ □ □ □ □ □ □ □ □ □ □ □ □
□

Cucumber

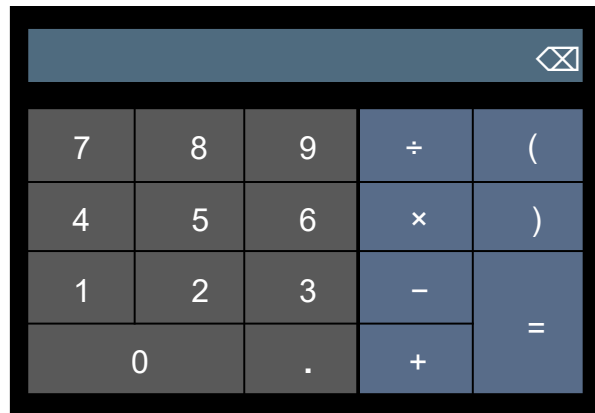
Cucumber □ 一个基于行为驱动开发的测试框架，它能够帮助团队使用业务上的语言对软件系统的行为进行测试。



Cucumber



Feature ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ User Story



Feature: Adding the numbers by calculator
As a customer,
I want to calculate the two numbers,
so that I can get the right results.

☐ ☐ ☐ ☐ Feature ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ Scenario

Cucumber



Scenario

Scenario: Add two numbers
Given the input "2+2"
When the calculator is run
Then the output should be "4"

- (Given)
- (When)
- (Then)

Scenario Step Step

Step Definition

```
Given /^the input("[^"]*)"$/ do |input_number|
  //get the input expression by the regexp match
  input = input_number
end

When /^the calculator is run$/ do
  //run the calculator and trigger it to do calculation by input

  result = calculator(input)
end

Then /^the output should be ([^"]*)"$/ do |expected_result|
  //get the expected result by the regexp match
  //verify whether the result is as we expected
  result == expected_result
end
```

Cucumber



Cucumber ☐

- Webrat
- Selenium
- Capybara
- WebDriver
- Watir
- Celerity ☐

NAME	TOTAL LINES	LINES OF CODE	TOTAL COVERAGE	CODE COVERAGE
app/controllers/activities_controller.rb	99	80	51.52%	46.25%
app/controllers/admin/administrators_controller.rb	7	4	28.57%	0.00%
app/controllers/admin/application_controller.rb	5	4	80.00%	75.00%
app/controllers/application_controller.rb	36	26	52.78%	50.00%
app/controllers/assets_controller.rb	9	8	44.44%	37.50%
app/controllers/attachments.rb	48	40	22.92%	22.50%
app/controllers/base_controller.rb	29	23	89.66%	86.96%
app/controllers/code_repositories_controller.rb	69	57	73.91%	70.18%
app/controllers/intros_controller.rb	3	3	66.67%	66.67%
app/controllers/sessions_controller.rb	43	31	48.84%	45.16%
app/controllers/site_controller.rb	13	11	53.85%	54.55%
app/controllers/topics_controller.rb	96	78	68.75%	64.10%
app/controllers/users_controller.rb	132	112	63.64%	62.50%
app/helpers/activities_helper.rb	32	26	75.00%	73.08%
app/helpers/admin/administrators_helper.rb	2	2	50.00%	50.00%
app/helpers/admin/application_helper.rb	2	2	50.00%	50.00%
app/helpers/application_helper.rb	9	6	66.67%	66.67%
app/helpers/audiences_helper.rb	2	2	50.00%	50.00%

需求的定义包括两个方面：

- 从用户角度（系统的外部行为）
- 从开发者角度（一些内部特性）来阐述需求



到底应该写什么样的需求文档？

你如何评价自己写的 SRS 作业？





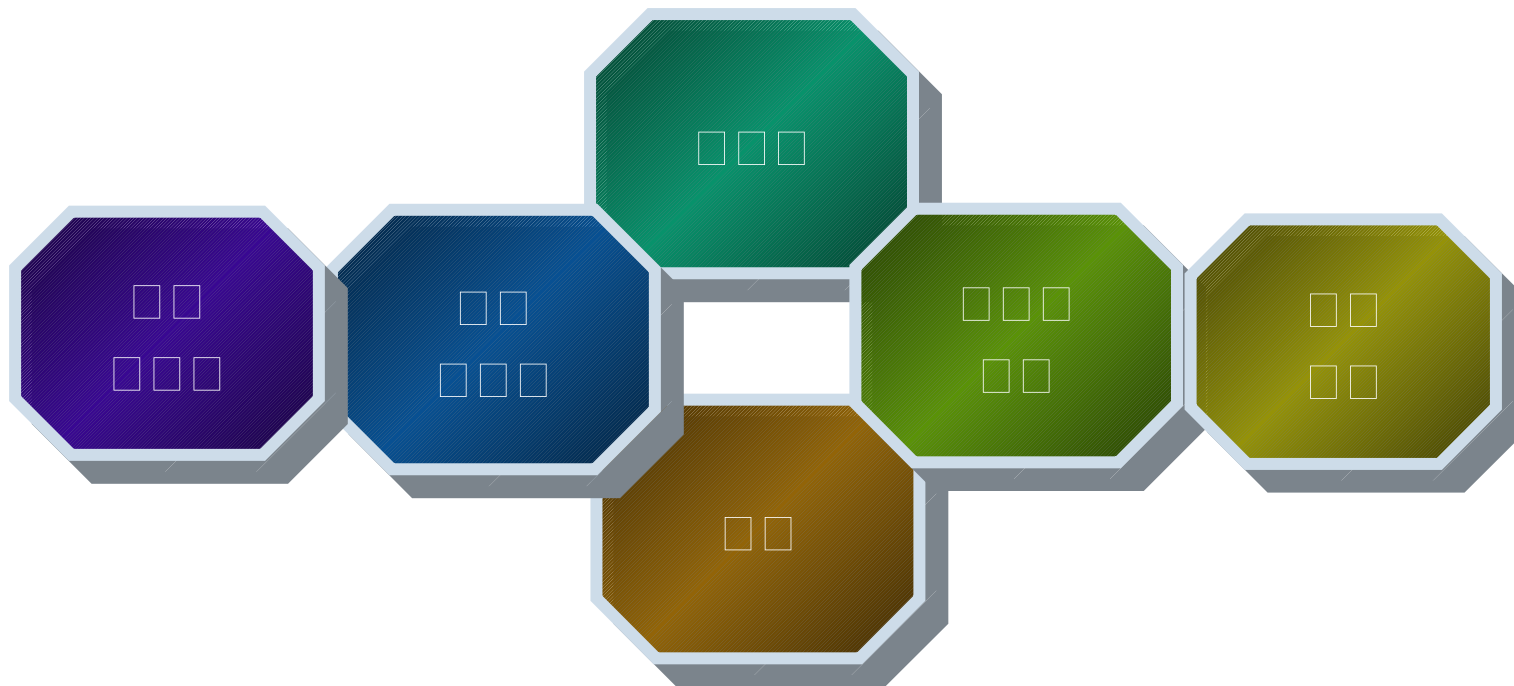
4



- 软件交互设计目标
- 软件交互设计师
- 软件交互设计过程
- 用户界面评价

软件交互设计目标

软件界面与交互设计的目标：优化人机界面，让软件界面更具可用性，让用户有更好的体验以及更方便地完成任务，从而提高产品商业价值。



界面布局
设计图像或标志
艺术的外观设计
形象地表达功能
视觉上的一致性
... ..

美术设计师



任务分析、概念设计
提供实时响应的标准
制定一致性的标准
可用性测试的评估
... ..



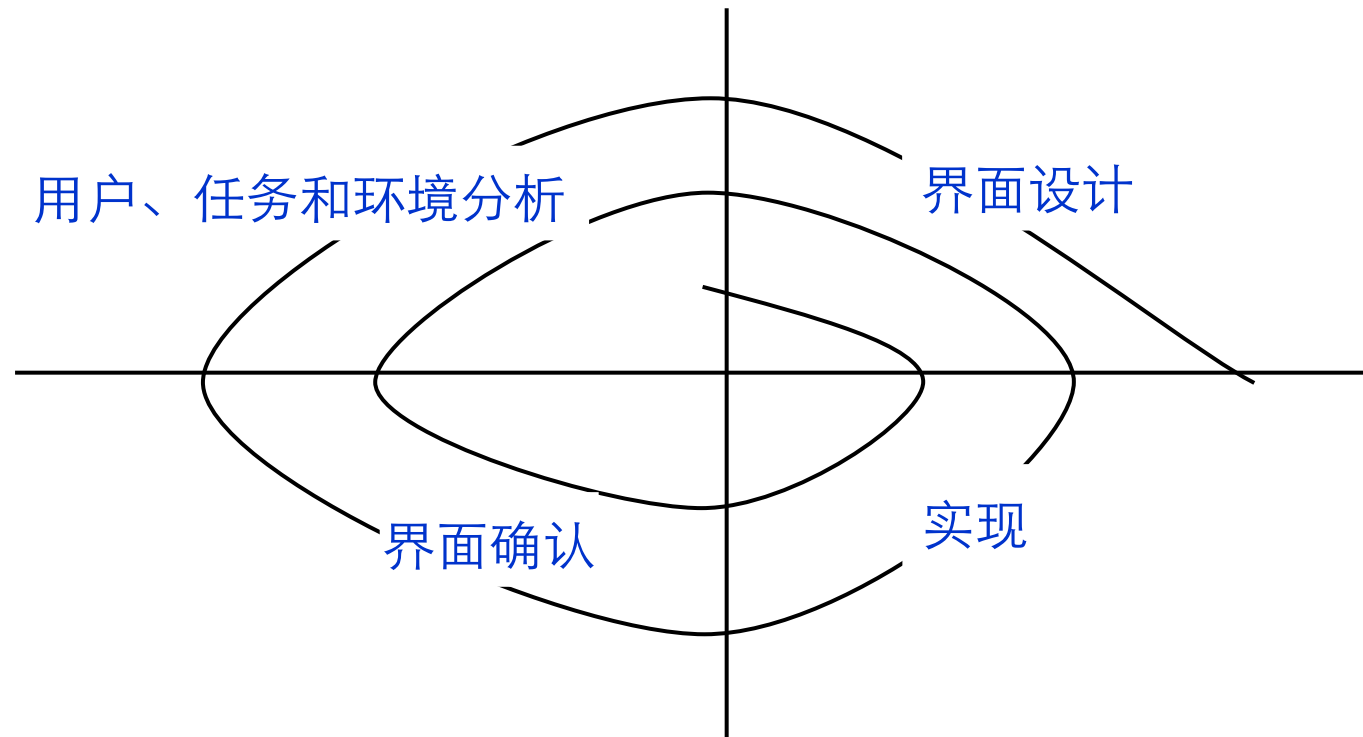
交互设计师

编程、内部结构
使用界面实现工具
重视功能的实现
理解技术局限
... ..



软件设计师

软件界面与交互设计是一个迭代化的过程





概念设计是通过对用户、任务和环境分析，制定出用户界面的整体架构。



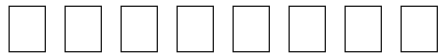
- [illegible]



- ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐
- ☐ ☐ ☐ ☐ ☐ ☐

□ □ □ □ □ □ □ □ □ □

- [illegible]



交互设计是使产品让用户能简单地使用，人的因素应作为核心被体现出来。

- 用户误操作后，系统提供有针对性的提示
- 让用户控制界面，给不同层次的用户提供多种可能性
- 允许兼用鼠标和键盘等，提供多种可能性
- 允许工作中断，如写短信时接到电话，通话完回到刚才处
- 使用用户的语言，而非技术的语言
- 提供快速反馈，给用户心理上的暗示，避免用户焦急
- 方便退出，如可手机上一个按键完全退出
- 导航帮助随时转移功能，很容易从一个功能跳到另外一个功能
- 让用户知道自己当前的位置，使其做出下一步行动的决定

视觉设计在概念设计基础上，参照目标群体心理模型和任务进行设计（色彩、字体、页面等）

- - □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □
 - □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □
 - □
 - □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □
 - □ □ □ □ □ □ □ □ □ □ □ □ □ □
 - □ □ □ □ □ □ □ □ □ □
 - □
 - □
 - □
 - □ □ □ □ □ □ □ □ □ □ □ □
 - □ □ □ □ □ □ □ □ 5 □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

可学习性

一个新用户需要多长时间才能成为熟练用户？

操作速度

系统响应用户工作情况的匹配速度如何？

鲁棒性

系统对用户错误的容忍程度如何？

可恢复性

系统从用户错误中的恢复能力如何？

适应性

系统与单一工作模式的紧密程度如何？



谢谢大家！

THANKS

