

# 故事定位方法 - Story Positioning

大系统需求有效实践

张克强

下一代  
软件研发  
SOFTWARE  
DEVELOPMENT

# 讲师介绍



## 张克强

- 系统分析师,高级程序员, CSM
- 规模化敏捷框架认证项目群咨询师  
SAFe 4.0 Program Consultant -SPC4)
- Exin认证DevOps Master
- 首届中国UML认证专家委员会专家
- 在中国大陆发布了首份在CMMI环境下导入敏捷报告
- 在中国大陆发布了首个Scrumban案例分析报告

微博: [张克强-敏捷307](#)

博客: [从高效过程到卓越结果](#)

- 2017年4月,北京光环敏捷春季峰会,《跨国大银行大敏捷和DevOps实例分享》,提出“大敏捷”概念。
- 2016年12月,敏捷之旅上海站,《敏捷需求分解和定位》
- 2015年7月 在TiD 2015 质量竞争力大会上演讲《一种多产品线复杂系统的架构治理结构》
- 2014年12月15日 全球软件案例研究峰会,演讲《企业级效率提升实践》
- 2014年9月,上海敏捷大讲坛,演讲《论敏捷自组织团队》
- 2013年11月Top100软件案例研究峰会,演讲《一个大型跨国银行敏捷转型》。
- 2011年~2013年参与编写《敏捷开发一千零一夜》,2013年8月出版
- 2012~2013年作为副组长,参与编写《敏捷开发知识体系》,2013年6月出版

# 需求工程与敏捷需求

## 软件工程的需求工程

### 需求开发过程

需求获取

需求分析

需求处理

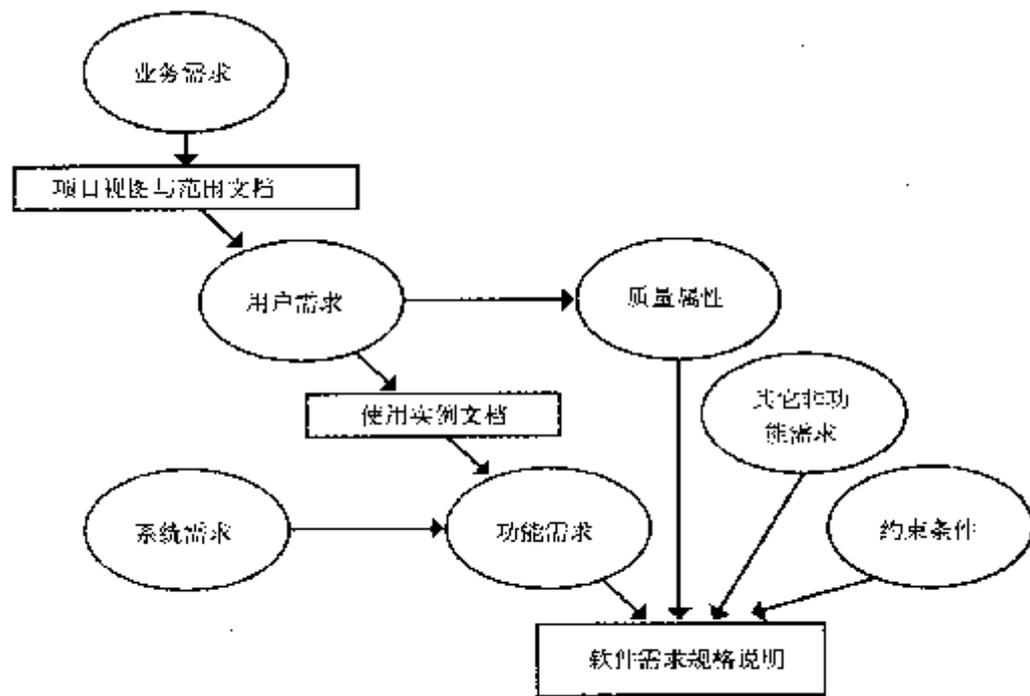
需求确认

### 需求管理过程

需求实现

需求跟踪

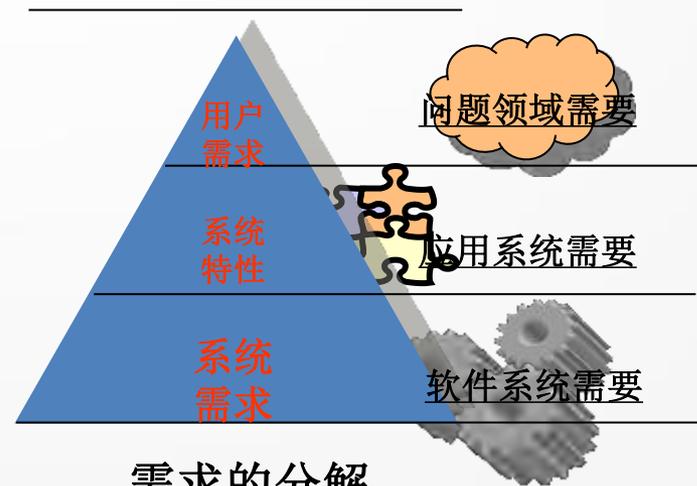
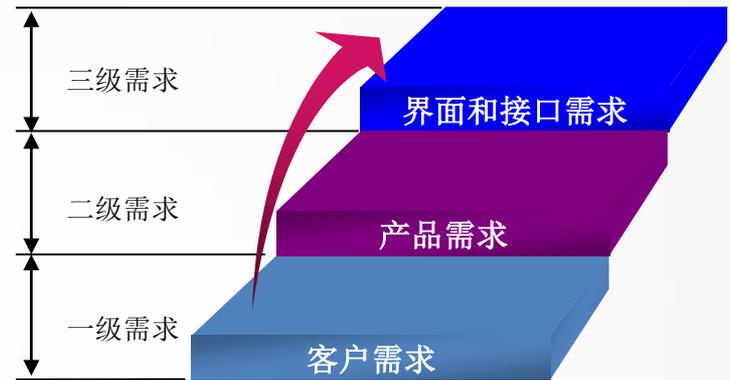
需求变更控制



软件需求各组成部分之间的关系

# 客观存在的需求层级

- 通俗常见划分
  - 业务需求，用户需求，功能需求
- 来自BABOK
  - 业务需求（Business Requirements）
  - 干系人需求（Stakeholder Requirements）
  - 方案需求（Solution Requirements）
  - 转换需求（Transition Requirements）
- 常见来自于软硬件综合开发类组织
  - BRD（Business Requirements Document）
  - MRD（Market Requirements Document）
  - PRD（Product Requirements Document）
  - FS（Function Specification）
- 来自CMMI-客户需求，产品需求，组件需求，接口需求
- 来自某些Agile做法- Epic, Theme, Story



## 需求表达——传统的《需求规格说明书》

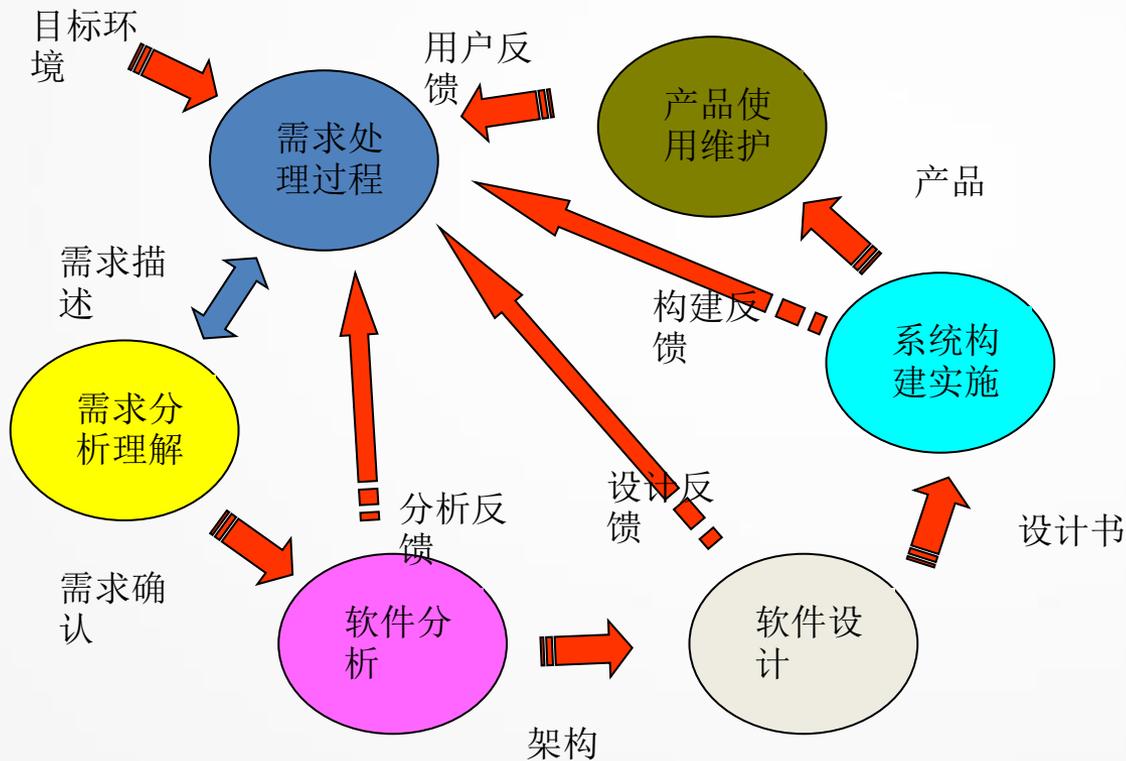
		1	2	3	4	5	6
A	引言	目的	文档约定	预期的和阅读建议	产品的范围	参考文献	
B	综合描述	产品前景	产品的功能	用户类和特征	运行环境	设计和实现上的限制	假设和依赖附录
C	外部接口需求附录	用户界面附录	硬件接口	软件接口	通信接口		
D	系统特性	说明和优先级	激励/响应序列	功能需求			
E	其他非功能需求	性能需求	完全设施需求	安全性需求	软件质量属性	业务规范	用户文档
F	其他需求						
G	附件	词汇表	分析模型	待确定清单			

软件需求规格说明书阐述一个软件系统必须提供的功能及其性能，以及他们的限制条件。这不仅是测试和用户使用的文档基础，而且是系统规划、设计和编码的基础。

# 在没有故事的时候

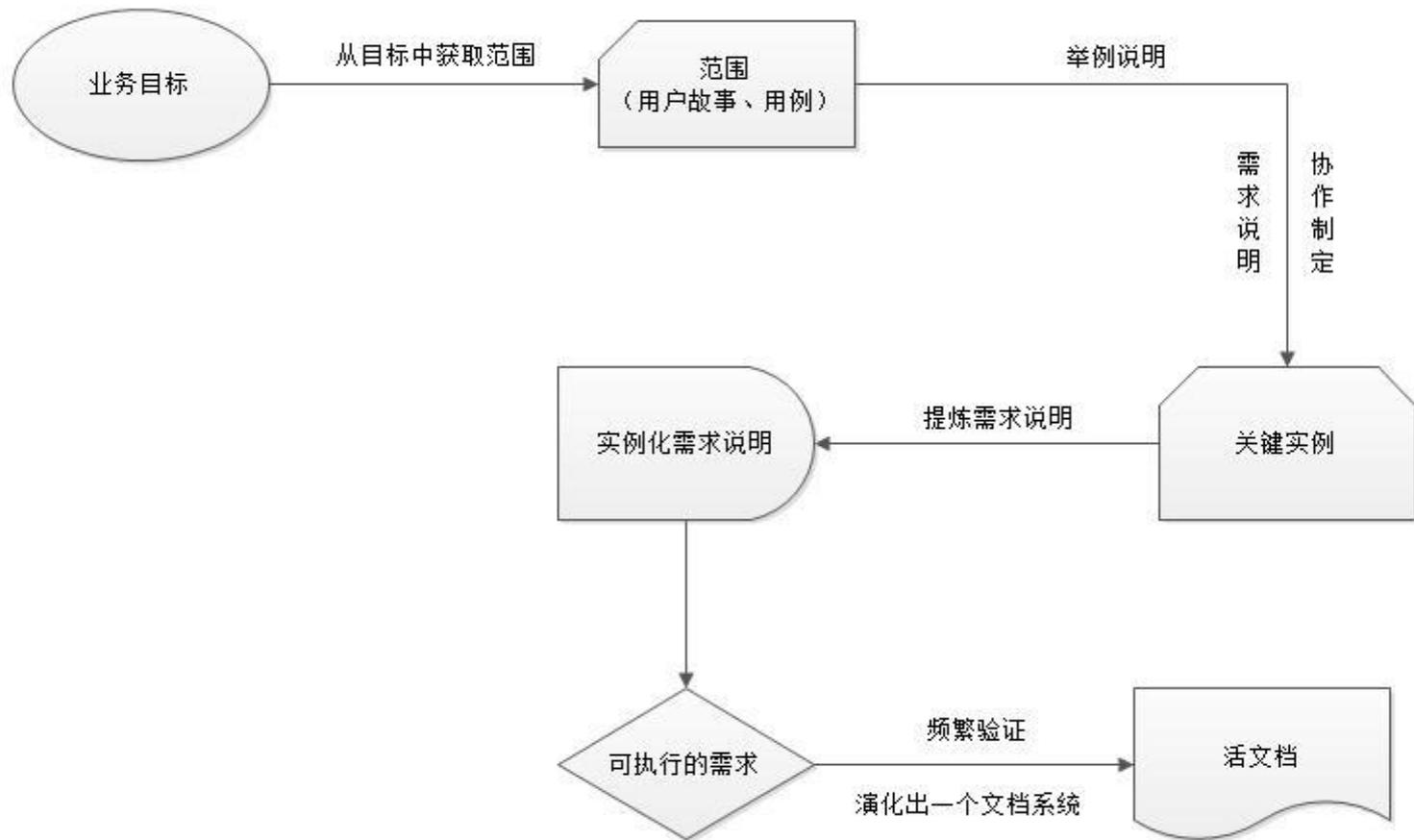
- 要做什么？
  - SRS
  - Use Case
- 如何安排？
  - 里程碑
  - WBS
- 如何度量
  - WBS
  - 功能点、用例点

# 敏捷需求 --- 拥抱变化



- 短迭代或者快速上线
- 全过程
- 多头并发

# BDD 实例化需求的主要流程





- 2001年，用户故事经典句型出世，As a role, I want to ..., so that ...
- 2001年，用户故事3C要点由Ron Jeffries提出，Card, Conversation, Confirmation
- 2002年，计划扑克发明，以故事点来估算故事的大小。
- 2003年，用户故事INVEST检查表提出，Independent, Negotiable, Valuable, Estimable, Small, Testable。
- 2003年，BDD由Dan North提出，它包括验收测试和客户测试驱动等的极限编程的实践。
- 2004年，User Stories Applied 出版，作者Mike Cohn

- 2005年，Mike Cohn发表“Agile Estimating and Planning”，planning poker开始流行。
- epic的使用，难以追查是哪年开始的，应当是在2003年以后。
- theme在用户故事的使用，同样难以追查何时开始，估计也是在2003年以后。
- 2006年，the Given-When-Then template 出现
- 2012年，《实例化需求》中文版出版
- 2014年，User Story Mapping 出版，作者 Jeff Patton
- 以Rally和Jira为代表的用户故事管理工具在2005年以后得到了巨大发展。

# 用户故事的常见困难

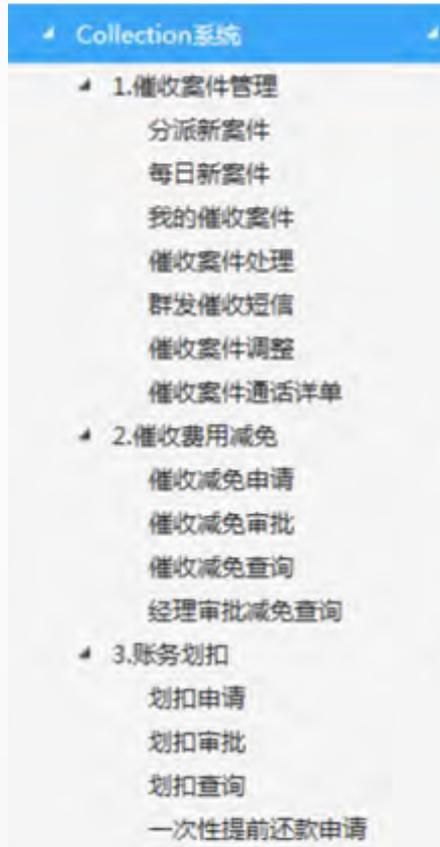
- 常见的困难1：如何分拆故事？
  - 往往故事来自于史诗，刚开始比较模糊，到后面发现有许多细节要处理，而一个迭代内来不及处理了
- 常见的困难2：如何处理关联到以前故事的修改升级？
  - 物理故事卡片难以长期保存，而故事完成后不定何时需要修改，有些故事上线后马上收到反馈，需要在下个迭代修改；
- 常见的困难3：如何估算故事的大小？
  - 全新的故事与修改的故事分别如何估算？故事的大小与故事所要花费的工作量是什么关系？
- 常见的困难4：如何整体理解故事？列表形式的backlog重点在于优先级排定，难以表达故事之间的关系

# 故事定位的原理

- 在识别故事的开始就试图把用户故事的内容存放到长期存放的地方，直接定位到表达系统功能合适的地方，减少复制和搬移。这就是定位的含义。
- 核心做法摘要
  - 使用Wiki来书写用户故事的内容，从故事树根部采用多层次文档结构，层层递进，采用思维导图方式，形成故事树，将故事放置到故事树的恰当位置
  - 如果史诗的初次定位不清晰，那么就从史诗开始，同样从史诗到故事采用多层文档，层层递进，到用户故事。
  - 使用条目化管理工具来管理用户故事的开发测试等活动
  - 使用泳道图来分析获得用户故事
  - 采用内置于用户故事内容中的迭代标识来识别迭代中用户故事，也称为故事切片，不因为用户故事在不同迭代开发而切分它。

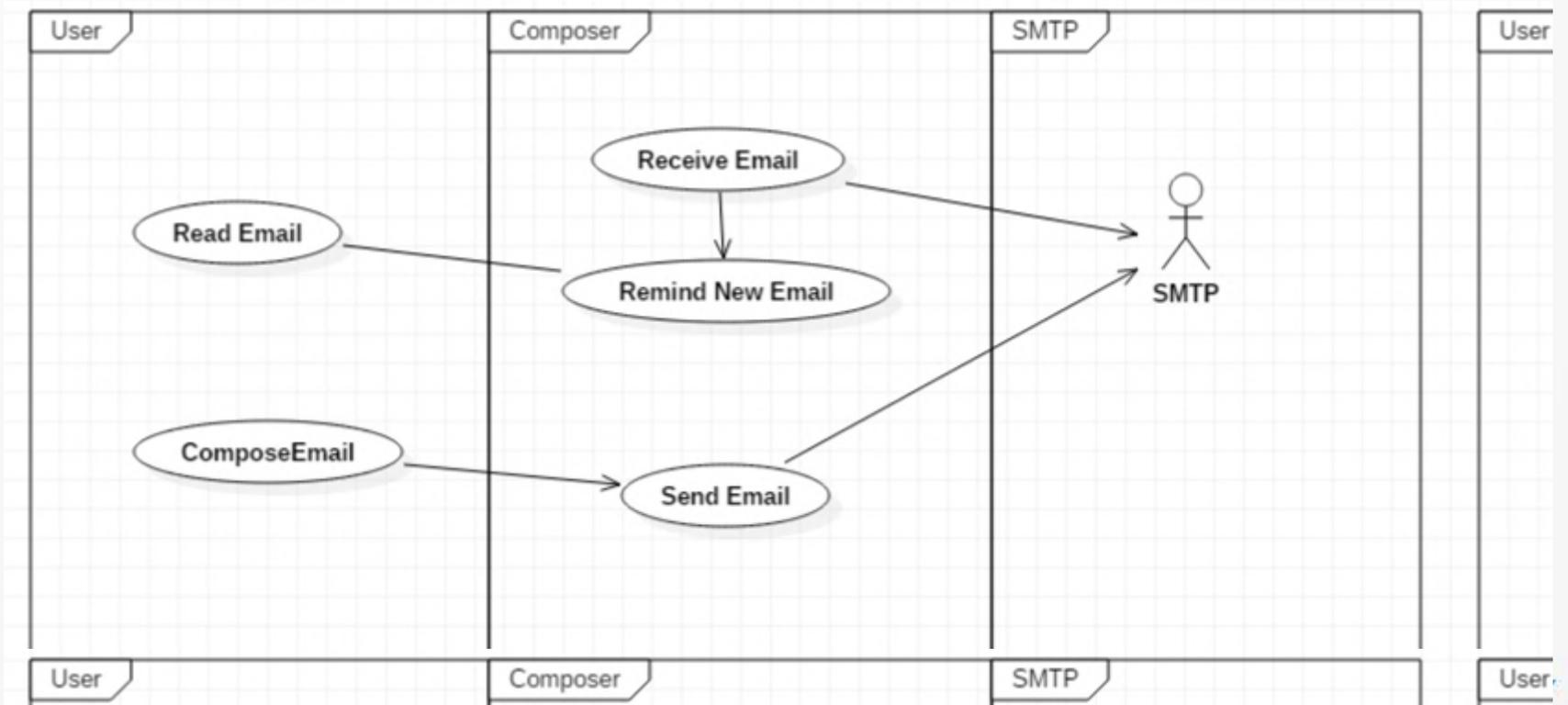
# 故事定位的实例

- 故事树

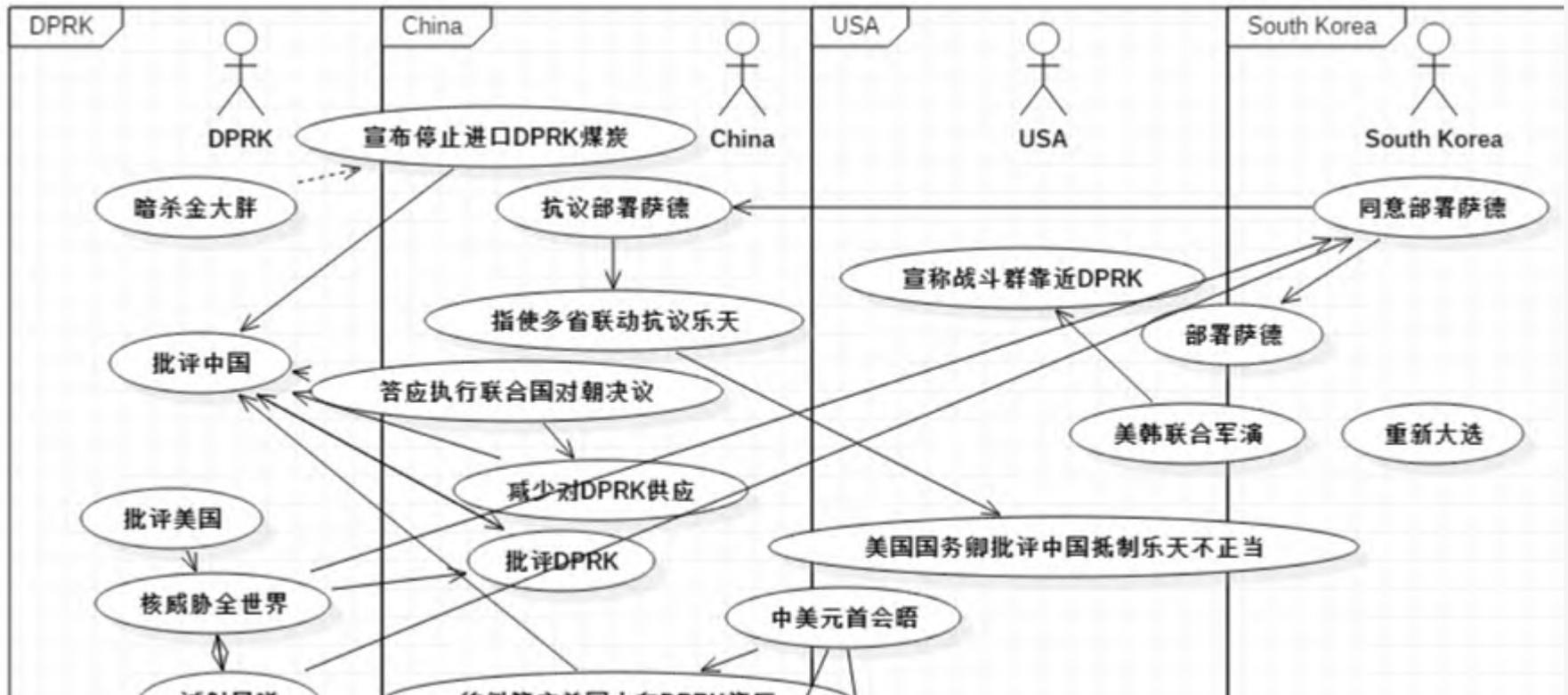


# 故事图

- 带角色泳道的用例图-示例1



# 故事图-示例2



# 故事图规则

- 一副故事图一般对应一个完整业务情景
- 将所有角色(人类,系统,自身系统的分解)以泳道列出
- 按用例图绘制业务场景的流程
- 角色的动作放在该角色的泳道里面
- 动作下游如果有多个可选动作,在连线上标明条件

# 描述故事+验收条件

- As an executive, I want to be able to filter the dashboard by department so that I can isolate data by a specific department.
- Acceptance Criteria:
  - **Given** the Executive Dashboard default view, **when** I select the department drop-down, **then** I have the ability to select a specific department to so only that data throughout the dashboard.
  - **Given** the department drop-down, **when** I select a specific department, **then** the entire dashboard filters to display only that department data.
- 3 question:
  - “Hey Product Owner, does the Executive need to be able to Multi-select several departments?”
  - “How about grouping by division?”
  - "who can access the Executive Dashboard"

# 讲故事 Story Telling

- As an executive, I want to be able to filter the dashboard by department so that I can isolate data by a specific department.
- Start Point: the dashboard is shown
- Normal Steps
  1. executive select the department drop-down
  2. system list all departments in drop-down
  3. executive choose a specific department
  4. the entire dashboard filters to display only that department data.
    - 4.1 department data is grouped by division(@future).
- Abnormal Steps
  - 3a executive choose 2+ departments by shift click or multi-selection, only first department will be chosen, or no response.

# 例子演练

- 为了防止欺诈，ATM非同名转账要延迟24小时到达
- 为了防范和打击电信网络诈骗犯罪，近日，最高人民法院等六部委联合发布《防范和打击电信网络诈骗犯罪的通告》（后文简称为《通告》），《通告》中指出：限定日期前，未投案自首的将依法严惩；下一步将继续严控电信网络诈骗犯罪；要求通信运营商今年年底前电话用户实名制，否则停机；要求银行自2016年12月1日起，个人通过银行自助柜员机向非同名账户转账的，资金24小时后到账。

# ATM功能树

- 查询
  - 查询余额
  - 查询流水
- 取款
  - 本行取款
  - 跨行取款
- 转账
  - 本行转账
  - 跨行转账
- 登录
- 密码修改

# ATM转账-原有故事

- 起始
  - 储户已经登录并选择转账
- 基本步骤
  1. 储户输入对方帐号和转账金额
  2. ATM显示掩盖对方姓的名字以及金额，请储户确认
  3. 储户确认转账
  4. ATM机提交转账请求到后台系统
  5. 后台系统进行转账处理，返回成功结果
  6. ATM机显示转账成功结果，询问储户是否打印凭条
  7. 储户选择打印，进行下一步，否则跳到第9步
  8. ATM打印凭条
  9. ATM机返回的储户主界面
- 异常步骤
  - 2a-对方帐号不存在
  - 2b-转账金额超过限额
  - 3a-储户否认转账
  - 5a-后台转账不成功

# 验收条件

- 各步骤响应时间不超过20秒
- 转账成功后，储户账户的余额减少转账金额
- 转账不成功，储户账户的余额不减少

# ATM转账-新故事1

- 起始
  - 储户已经登录并选择转账
- 基本步骤
  1. 储户输入对方帐号和转账金额
  2. ATM显示掩盖对方姓的名字以及金额，请储户确认
    - 2.1 如果非同名转账，提醒储户24小时后才能到账 @l17
  3. 储户确认转账
  4. ATM机提交转账请求到后台系统
  5. 后台系统进行转账处理，返回成功结果
  6. ATM机显示转账成功结果，询问储户是否打印凭条
  7. 储户选择打印，进行下一步，否则跳到第9步
  8. ATM打印凭条
    - 8.1如果非同名转账，说明仅作参考提示，不作为正式转账成功凭条
  9. ATM机返回的储户主界面
- 异常步骤
  - 2a-对方帐号不存在
  - 2b-转账金额超过限额
  - 3a-储户否认转账
  - 5a-后台转账不成功

# 新需求：储户可以选择24小时生效还是马上生效，如何修改？

- 起始

- 储户已经登录并选择转账

- 基本步骤

1. 储户输入对方帐号和转账金额
2. ATM显示掩盖对方姓的名字以及金额，请储户确认
  - 2.1 如果非同名转账，提醒储户24小时后才能到账 @l17
3. 储户确认转账
4. ATM机提交转账请求到后台系统
5. 后台系统进行转账处理，返回成功结果
6. ATM机显示转账成功结果，询问储户是否打印凭条

7. 储户选择打印，进行下一步，否则跳到第9步

8. ATM打印凭条

- 8.1 如果非同名转账，说明仅作参考提示，不作为正式转账成功凭条

9. ATM机返回的储户主界面

- 异常步骤

- 2a-对方帐号不存在
- 2b-转账金额超过限额
- 3a-储户否认转账
- 5a-后台转账不成功

# 好处

## 特征

- 成长Growing
  - 关注产品和系统变化,收集信息
- 分层Layering
  - 更深信息可以向下扩展
- 关联Relating
  - 故事与故事之间
  - 关联-嵌套-引用-等等
- 可视化Visualizing and 协作Collaborating
  - 所有团队成员随时能看能改
  - 鼓励业务人员随时能看能改
  - 尽可能让参与各方知道,在企业保密政策下

## 困难解决

- 困难1: 如何分拆故事?
  - 往往故事来自于史诗,刚开始比较模糊,到后面发现有许多细节要处理,而一个迭代内来不及处理了
- 困难2: 如何处理关联到以前故事的修改升级?
- 困难3: 如何估算故事的大小?
- 困难4: 如何整体理解故事

# 故事树的好处

- 树形结构符合人类大脑思维-脑图
- 长远上便于产品和系统修改升级
  - 支持全文检索
  - 便于新人学习
- 短期上在清晰定位的情况下准确理解
  - 方便查看全部上下文
  - 可以关联到任何地方
    - 条目
    - 代码
    - 测试
- 积累信息和知识, 将知识显性化
  - 方便补充更多信息

# 对比

**vs Backlog**

**vs Story Mapping**

# 故事树、故事地图与故事图

- 故事树 - 成长-活文档，与系统/产品共同成长
- 故事图 - 业务场景分析-便于快速理解
- 故事地图(对照) - 可视化迭代-发布增量，在迭代发布临时使用
- 可以融合在一起，按需使用

# Question?

- 联系张克强微信如右

