

0.1 版流程图设计分析

2016年1月3日 星期日

17:36

索引:

- [为什么要做流程图:](#)
 - [关于开发流程:](#)
 - [为什么要做流程图:](#)
 - [最后扯点理想](#)
- [流程图设计:](#)
 - [一个APP所拥有的所有元素:](#)
 - [界面元素:](#)
 - [界面元素特性:](#)
 - [界面的存在性:](#)
 - [界面的事件相关:](#)
 - [界面元素涉及内容](#)
 - [怎么在流程图描述](#)
 - [事件:](#)
 - [事件的一个重要的概念](#)
 - [事件特性:](#)
 - [流程图如何描述事件:](#)
 - [状态:](#)
 - [状态特性](#)
- [怎么用流程图描述APP的这三种元素](#)

为什么要做流程图:

- [关于开发流程:](#)
 - 从PM说起:
 - 首先有一个项目
 - 这个项目最初的时候呢基本上是一片空白，其只有一个中心主题比如叫“这是一个2B的洗衣机APP”
 - 然后这个项目从天而降，砸到了某一个倒霉产品经理身上
 - 项目在开始的时候其本身只有一些核心，没有太多的具体内容。既然项目被砸到了产品身上，产品就要负责把项目抓换成实际的产品。但是只有核心理念的东西是没法拿来实现的，所以产品要从核心拓展，继续往里面加内容。
 - 加内容的方式多种多样，无论是市场调研也好，借鉴同类也行，这方面并不是讨论重点
 - 总之通过了种种方式，产品完成了需求挖掘，成功的将本身只有一个核心的产品加上了条条框框，罗列了一二三四等等模块。每个模块下面又列举出了四五六项等等功能，然后这个项目终于完善到可以拿去实现了。
 - ☆ 把上面这些东西整合起来，包括核心和所有从核心拓展的内容，将这些东西统称为需求。然后上面整个过程可以称之为“定义需求”。PM所负责的就是定义需求的工作。这样一个纸面上的产品就出炉了。
 - PM外其他人是做什么的
 - 毕竟产品不是万能的，其本身通常不负责产品的实现，那么需求有了以后，这一份需求就要被提交给测试进行校验，交给设计制作界面，交给开发完成程序。
 - 如果把产品经理称为上层，那么开发，测试等等人员可以称为下层（好像这个概念没什么卵用）
 - 实际工作中，PM可能为多层，不止一个人，并且上层的任务PM是负责人，但不一定要自己实现。比如原型图，通常是专人负责。
 - ？ • 这里表达上可能不太好
 - 而开发在其中要做的，就是将纸面的需求转换为正式的产品。
 - ☆ 下层的工作统称为“实现需求”
- [为什么要做流程图:](#)
 - 从“需求定义”到“需求实现”这样的两步过来以后，似乎一个产品已经出来了。那么为什么作为开发，我要多此一举的想要弄一份流程图呢？
 - ☆ 因为漏了一步！特别需要注意的是，定义需求的人，和实现需求的人并不是同一个。所以从需求定义完到对需求开始实现之间还有很重要的一步叫“需求传达”
 - 而一个项目的逻辑本身是非常复杂的，这样的内容即使存在一个人脑中理清逻辑也是个较大的工作量。何况一个项目组通常有很多人。如何的准确，完整，并且直观的向所有人传达所有的需求是一个很高技术含量的事情。
 - 准确:
 - 栗子:
 - windowPhone应用要多少有多少
 - iOS应用要多少有多少
 - 这样传达就叫不准确
 - 准确还应该包含逻辑间不能自相矛盾
 - 完整:
 - 例如:

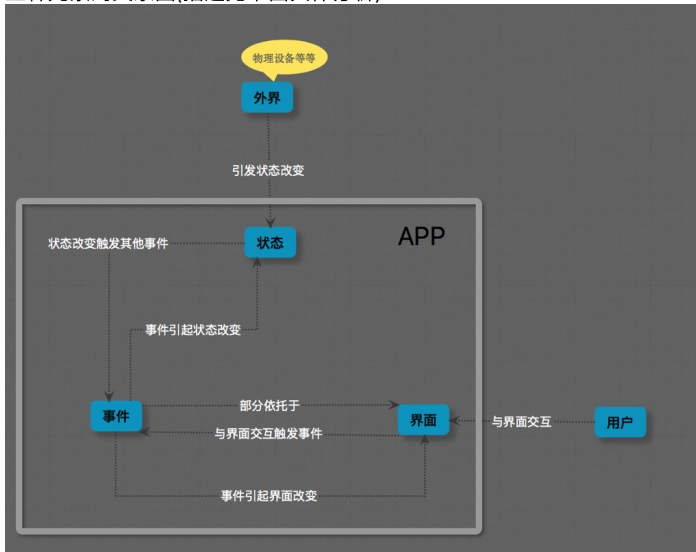
- 给了文本框没给输入规则
 - 给了登录没给登出（实际开发中重灾区，同类状态没闭合是经常会出现的）
- 实际原型图通常过于简陋
- 直观：
 - 栗子：
 - 原型图一份
 - 需求文档一份
 - 协议书一份
 - 口头n份，会议n份
 - 需求表达在多个地方，或者给的东西很难看都叫不直观
- 通常开发程序的时候需要有一份统称为“需求说明文档”的东西作为指南，程序则照着需求说明文档给定的内容进行开发。需求说明文档就是一个文本形式的产品
 - 此处一份是虚指，实际上需求文档不单止数量不定，表现形式也是任意的。而具体的表现效果如何，需要看设计人员的功力。
 - 表现形式：原型图，协议书，开会等等的都可以做需求说明文档（因为暂时想不到合适的名词）
- ☆ 需求说明文档就是用于“需求传达”的东西
- 实际工作中需求文档从来没满足过上面三条标准，不，需求定义阶段通常就开始存在问题了
 - 真实情况也不可能完全满足，只能说最大限度，毕竟需求是会变的，理解是有误差的，表述是有转换率的。。总之种种原因
 - 这些问题从以前的几个版本流程图第一设计目的是给原型图找茬就知道了
- 有空要做具体描述，把常见的文档或原型图存在的问题理出来[常见的需求传达问题](#)
- 所以为什么要自己做一份流程图
 - 其实就是要自己复述一次需求说明书，因为需求说明书直接用效率低有问题
 - 在复述的过程中，找到需求遗漏弥补需求定义不完全
 - 找到需求模糊的地方，和PM协商解决
 - 提高开发效率，整合多份文档
- 最后扯点理想
 - 理论的分工
 - 开发重点关心的是对产品的技术实现过程
 - 如何实现产品的设计
 - 如何保证产品稳健与高效
 - 如何使产品易于维护与拓展
 - 而PM面向开发的两项职能应该是
 - 定义产品的需求
 - 准确传达产品需求
 - 实际的上分工
 - 开发
 - 产品程序角度上看不难，实现问题大部分直接搞定，不会的谷歌搞定
 - bug不少（要做一份常见BUG总结，然后从设计上既要根除），不过app不复杂，性能够用，谈不上需要处理运行效率
 - 维护性。。。拓展性。。。反正需要加的不多
 - PM
 -
 -
 -
 - 理论不适用于实际标准，但程序的优秀是追求，还是希望能专注在自己领域

流程图设计：

- 基本目的：
 - 上面提到，流程图其实就是对说明文档的复述，因为直接使用需求说明书不方便[为什么要做流程图：](#)
 - 所以流程图设计时要做的事就是，把APP中所有的需求都罗列出来。
 - 这显然不是一个容易的事，并且在需求说明文档可能不完善，只能自己补充的情况下
 - 一个APP产品需要的所有需求这个内容太泛，想到再补充
- 希望达到的效果：
 - 首先要有所有的需求这点是不必细说的
 - 另一个必要条件是表达直观
 - 最好能够把能够展现的内容都展现在一个文件内
 - 制作快捷
- 一个APP所拥有的所有元素：
 - ○ 这里最好加个论述过程
 - ☆ ○ 要完整的描述一个APP的逻辑，需要以下三个元素（并且只需要这三个？）
 - 界面，事件（Action / Function），状态
 - 具体为什么是这三个见[历史版本](#)

这些教科书式的内容不要太细究。先着重怎么弄，现在想这个太虚了

- 还是把具体位置提取出来说明为什么是这三种吧
- 三种元素的关系图(描述见下面具体分析)



- 界面元素:
 - 界面元素不必多说, app中看到的所有东西都叫界面元素
 - 界面元素特性:
 - 界面元素不是生来就有的, 其由某一个事件, 在某一时刻创建出来
 - 并且界面的内容是不固定的, 界面会随着某个状态的改变而发生转变
 - 界面的存在性:
 - 界面元素的存在是和页面的当前状态直接相关的
 - app当前展示的页面固定的, 在这个页面中某一刻展示的界面元素却是不定的
 - 所以界面元素的描述应该依据状态进行
 - 界面的事件相关:
 - 界面的改变是依托于事件的
 - 但是部分事件又是由用户与界面的交互触发的
 - 界面元素涉及内容
 - 位置
 - 大小
 - 颜色
 - 字体
 - 上述这些内容应该是有一个统一的标准
 - 一个项目应该提供一个公用"仓库"
 - 用于存放统一的颜色, 字体等基础属性
 - 那么怎么在流程图描述
 - 大部分界面无论在什么状态下都是不变的
 - 所以描述的时候应该以界面元素为基础, 对其不同状态下得变化进行描述
 - 部分界面是可以进行交互的, 而部分事件是由用户与界面的交互引发
 - 所以在描述界面的时候需要把依托在该元素之上的事件展现出来
- ☆ 描述界面最好的方法永远是设计图! 设计图! 但是现在的设计图对界面没有描述的那么详细, 很遗憾
 - ? • 对于界面的描述, 是否可以简化
- 事件:
 - 事件指的就是程序的内部逻辑处理, 用程序的名词表述就是函数, 方法
 - ☆ 事件有一个很重要的概念是
 - 整个APP其实可以说为就是一个在系统执行的事件 (进程其实就是一个程序主函数在跑)
 - 所以APP是事件驱动的, 程序从出生开始就在执行事件, 程序拥有的所有东西, 都是在事件处理的时候创造的
 - ? 其实事件是一个很模糊的概念, 怎么规划好是个很大的问题详见: [程序中容易模糊的地方](#)
 - 事件特性:
 - 事件生来就有
 - 一个页面的实现中必定包含了所有这个页面要处理的事件
 - 由于事件生来就有, 当状态改变时, 状态对事件的影响不是让事件消失
 - ☆ 而是让事件走向另一条分支 (这是状态对事件和界面的不同之处, 描述方法应该会因此不同)
 - 流程图如何描述事件:
 - 所有事件是有触发源的, 要把能够触发事件的地方写出来
 - 在页面的描述中只需要把事件做什么写出来就好了, 不要写流程
 - 事件的处理应该需要专门开一页, 因为程序流程图的表述和层次图有本质的不同: [程序流程图与我们说的逻辑图的不同](#) [关于流程图](#)

○ 状态：

- 状态指的是程序运行中由于事件或者外界引起的各种不同的状态

■ • 这个概念不好解释

• 状态特性

☆ • 状态无处不在，程序当前必定出在一个状态中

- 部分状态具有全局性（登录登出），全局性状态改变可以引发特定页面事件发生
- 界面元素在不同状态下有不同效果
- 事件元素在状态下有不同分支

■ • 流程图如何描述状态：

- 状态的描述可能会造成流程图非常复杂
 - 一个页面有多个状态，是不是要把这所有状态的界面都枚举一遍
 - 事件倒是不用特殊处理，本来就囊括不同状态不同处理
- [如何最方便的描述不同状态下的界面或者事件](#)

■ • 待补全

- 少很多很多东西

• 怎么用流程图描述APP的这三种元素

■ ○ 区分页面与页面之间叫流程，页面内部叫页面描述

- 一个是用流程图描述，有先后顺序
- 一个是用层次结构图描述，有包含与被包含关系