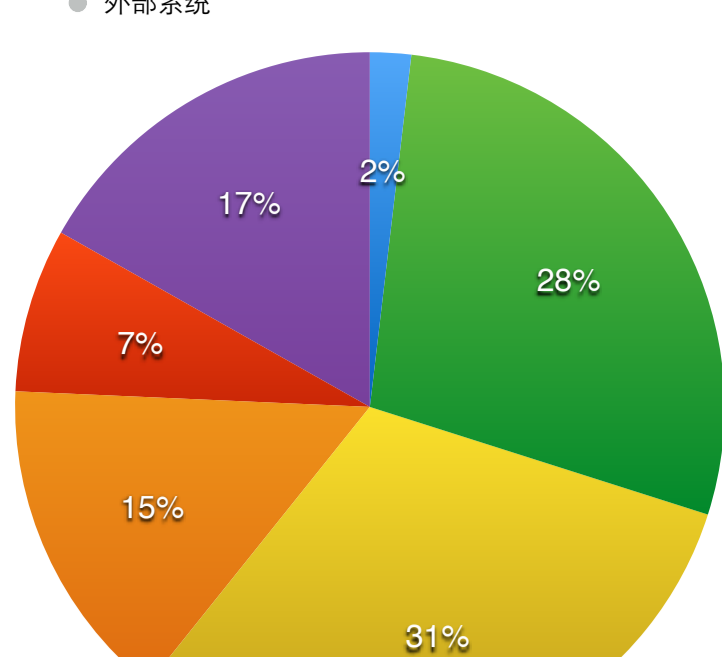


Bug类型

bug类型	数量 (106)	bug类型细分	数量
性能问题	2	数组越界	0
		数据类型	1
		逻辑问题	1
		版本兼容	0
代码逻辑	30	未知原因	0
		逻辑错误	26
接口相关	33	网络机制	4
		接口问题	33
		环境问题	0
完整性	16	请求参数问题	0
		考虑不全 (数据联动)	10
		考虑不全 (网络差)	0
		考虑不全 (多用户)	0
需求变更	8	考虑不全 (版本兼容)	2
		边界测试	4
		需求文档不明确	5
		需求文档变更	3
用户体验	18	新增需求	0
		交互处理	0
其他	0	样式	18
		兼容性	0
		未知原因	0
外部系统	0	数据调整	0

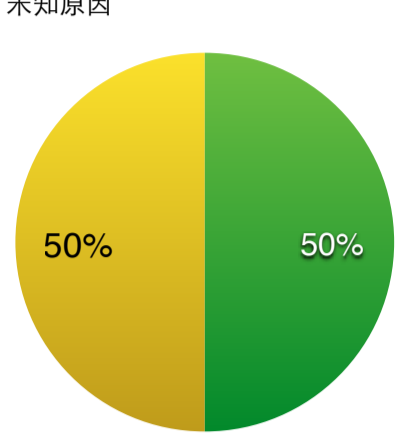
● 性能问题 ● 代码逻辑 ● 接口相关 ● 完整性 ● 需求变更 ● 用户体验
● 其他 ● 外部系统



【性能问题】类型细分

bug类型	数量
数组越界	0
逻辑问题	1
版本兼容	1
未知原因	0

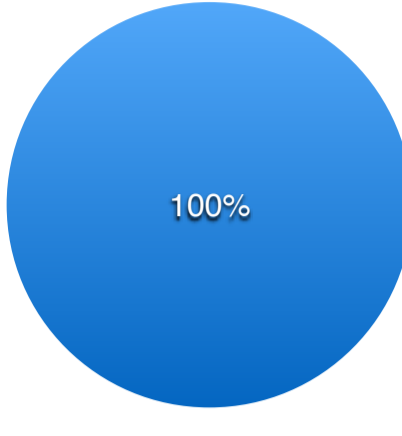
● 数组越界 ● 逻辑问题 ● 版本兼容 ● 未知原因



【接口相关】类型细分

bug类型	数量
接口问题	33
环境问题	0
请求参数问题	0

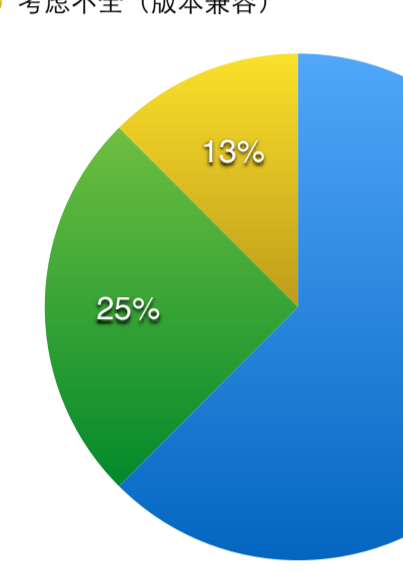
● 接口问题 ● 环境问题 ● 请求参数问题



【完整性】类型细分

bug类型	数量
考虑不全 (数据联动)	10
考虑不全 (边界测试)	4
考虑不全 (版本兼容)	2

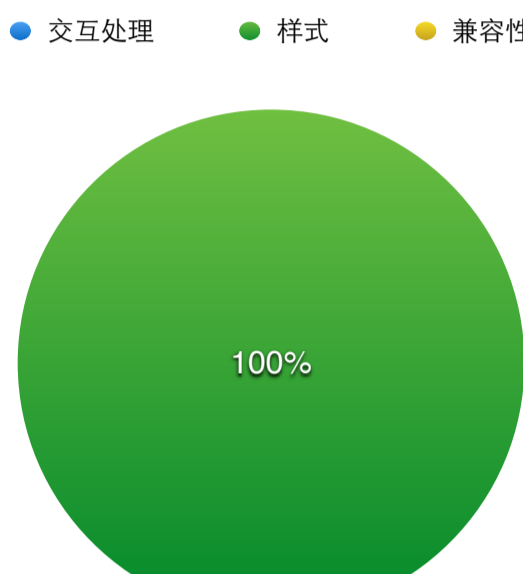
● 考虑不全 (数据联动) ● 考虑不全 (边界测试) ● 考虑不全 (版本兼容)



【用户体验】类型细分

bug类型	数量
交互处理	0
样式	18
兼容性	0

● 交互处理 ● 样式 ● 兼容性



组员日均bug产生率

组员	有效工作日 (天)	iOS开发的bug总数	bug数/工作日(产生率)
曾文杰	13	14	1.076
练积金	14	16	1.142
黄永坤	15	13	0.866
周鸿基	10	3	0.3
周健威	12	6	0.5
李冠淳	15	11	0.733

总日均bug产生率

工作日 (总时间)	iOS开发的bug总数	bug数/工作日
79	73	0.92

Bug激活情况

BugID	激活次数 (26)	类型	所属 (人)
51517	1	mrobot	黎国凯
52231	1	mrobot	兰振华
52303	1	iOS	练积金
52380	1	iOS	练积金
52637	3	mrobot	周儒轩
52665	1	iOS	曾文杰
52694	3	iOS	曾文杰
52709	2	iOS	练积金
52734	1	iOS	曾文杰
52776	1	iOS	黄永坤
52824	2	iOS	黄永坤
52895	2	iOS	李冠淳
53063	3	iOS	周健威
53213	1	iOS	周健威
53325	1	iOS	黄永坤
53406	1	iOS	李冠淳
53462	1	应用后端	陈卓林

各组员bug激活率

姓名	bug激活数	bug激活数/修复bug总数
曾文杰	5	0.357
练积金	4	0.25
黄永坤	4	0.307
周健威	4	0.666
李冠淳	3	0.273
周鸿基	0	0

总激活率

是否包括应用相关的bug	bug总数	激活总次数	激活总次数/bug总数
是	106	26	0.245
否	73	20	0.273

项目总结

曾文杰	<p>学车宝典2.0.0版本, 总体开发符合开发计划, 按时按质提交迭交, 这个版本在吸取之前版本不足的基础上, 在质量上有了较大的进步, 主要体现在以下几个方面:</p> <ol style="list-style-type: none"> 一、产品需求不确定, 开发时间紧, 导致评估时间一改再改; 二、项目迭代频繁, 参与开发人员多的情况下, 项目代码冗余率高, 比如PCKit框架内已有分享模块、PCViewcontroller类、PCLabel等, 但在项目内却是另外重写, 不方便项目的后续维护与扩展, 同时这些新增的工作量在排期阶段未能评估到; 三、报价SDK模块。所谓的SDK是基于PCKit框架打包出来的, 而学车宝典本身又引用了PCKit, 这样会导致有冗余文件的存在, 无法嵌入SDK。这里采用了将报价模块的代码copy过来的方法, 引用后还需梳理报价的页面逻辑、魔方key、计数器、页面路径的配置等。最后SDK模块的实际工作量大于评估工作量, 而且由于短时间内难于对旧代码理解透彻, 在活动详情页面的统计中使用了idfa (虽然报价宝典就是这么处理的), 需要重新打迭交包; 四、临近QA阶段新增抽奖活动的需求, 而且活动页面的制作排期与客户端的开发排期没能同步, 并缺少统一跟进排期的人员, 增加了客户端与后端人员的沟通成本。 <p>在开发过程中也存在一些不足的地方, 在这里给大家一些建议:</p> <ol style="list-style-type: none"> 一、要培养自己独立思考问题的能力, 遇到问题, 要冷静思考, 发散思维, 寻求多种解决问题的办法。 二、当遇到接口, 设计等相关问题时, 要及时找相关的人员配合沟通, 一些重要的问题, 要拉上项目经理, 一起讨论解决。 三、提高自己的工作积极性, 始终保持对工作的热情。 <p>在今后的开发过程中, 大家要扬长避短, 不断的提升自己, 寻求更大的突破。</p>
练积金	<p>V2.0.0版本本人负责代码合并、消息推送、找车报价SDK、抽奖活动模块的开发, 开发过程中存在问题如下:</p> <ol style="list-style-type: none"> 1、产品需求不确定, 开发时间紧, 导致评估时间一改再改; 2、项目迭代频繁, 参与开发人员多的情况下, 项目代码冗余率高, 比如PCKit框架内已有分享模块、PCViewcontroller类、PCLabel等, 但在项目内却是另外重写, 不方便项目的后续维护与扩展, 同时这些新增的工作量在排期阶段未能评估到; 3、报价SDK模块。所谓的SDK是基于PCKit框架打包出来的, 而学车宝典本身又引用了PCKit, 这样会导致有冗余文件的存在, 无法嵌入SDK。这里采用了将报价模块的代码copy过来的方法, 引用后还需梳理报价的页面逻辑、魔方key、计数器、页面路径的配置等。最后SDK模块的实际工作量大于评估工作量, 而且由于短时间内难于对旧代码理解透彻, 在活动详情页面的统计中使用了idfa (虽然报价宝典就是这么处理的), 需要重新打迭交包; 四、临近QA阶段新增抽奖活动的需求, 而且活动页面的制作排期与客户端的开发排期没能同步, 并缺少统一跟进排期的人员, 增加了客户端与后端人员的沟通成本。
黄永坤	<p>在学车宝典2.0版本的开发过程中, 我负责开发模块主要有以下三部分: 学习历程、学习统计、问答模块的问题搜索和问题添加, 学习历程部分包括UI实现, 登录前后的个人信息刷新, 各个科目做题次数、考试次数、视频播放次数的统计; 学习统计部分包括UI的实现、练习统计信息的刷新; 问答部分包括常见问题列表、相似问题列表的加载和显示, 问答帖子的发布。</p> <p>这三个部分在开发的过程中或多或少存在一些问题, 现分别罗列各个部分存在的问题以及分析导致问题出现的原因:</p> <p>(一) 学习历程</p> <p>学习历程模块存在4个问题:</p> <ol style="list-style-type: none"> 1、使用第三方平台账号授权登录, 在没有修改过头像的情况下, 头像不对。 <p>这个问题是代码合并导致的, 将1.1和1.2版本的代码合并后, 代码存在以下问题: 首次授权登录时上传第三方平台用户头像到UPC的时候, 接口调用错了, 在1.1的最新代码中是调用上传头像URL的接口, 合并后是调用上传图片的接口。</p> <ol style="list-style-type: none"> 2、在学习历程页面点击默认头像进行登录, 登录成功后用户头像还是显示默认头像, 需要退出学习历程页面再进入才能刷新。 <p>这个问题在iOS 7系统中才会出现, 解决方法是在给PCButton设置图片的时候改用setUiview:defaultImage:indicator:style:cachePolicy:priority:方法。</p> <ol style="list-style-type: none"> 3、分享时, 分享的问题出错。 <p>这个问题是由于输入文字, 直接复制学习历程页面中显示相伴多少天的label赋值代码, 在单元测试的时候也没有发现文案不正确。</p> <ol style="list-style-type: none"> 4、从1.2版本升级到2.0版本, 没有统计1.2版本考试次数的数据。 <p>这个问题是由于统计旧版本的数据信息时不对造成的, 在版本升级的时候, 数据库也会升级, 统计的时机必须在数据库升级完成以后。</p> <p>(二) 学习统计</p> <p>学习统计主要存在2个问题:</p> <ol style="list-style-type: none"> 1、从学习统计页面点击“马上去完成”按钮进入未做题练习页面, 做题后返回, 学习统计页面的数据没有刷新。 <p>这个问题是由于统计练习信息的方法调用放在了学习统计页面的viewDidLoad:方法中, 没有在viewWillAppear:方法中。</p> <ol style="list-style-type: none"> 2、一些UI的细节问题。 <p>主要是在完成了UI的实现后没有再次认真仔细对设计稿, 存在一些位置偏移的问题; 成绩预测的分数字体没有动态改变 (显示问号 and 显示分数的字体大小是不一样的)。</p> <p>(三) 问答模块</p> <p>问答模块主要存在以下问题:</p> <ol style="list-style-type: none"> 1、问答页面点击输入框, 切换到中文拼音输入法, 输入拼音但不选词, 直接点击问题列表的某一条记录, 问答界面crash。 <p>这个问题是由于参数值输入的时候出错了, 向一个空的数组发送objectAtIndex:方法。解决方法是优化程序中切换UITableView数据源的逻辑, 在向数组发送objectAtIndex:方法时, 做是否越界的判断。</p> <ol style="list-style-type: none"> 2、在问答页面中输入关键词后在1秒之内点击“提问”按钮、“添加问题”按钮或者点击列表的某一条数据离开问答页面, 然后返回, 问题列表有时候为空。 <p>这个问题是由于问题列表的数据刷新逻辑有错误导致的, 在问答页面的viewWillAppear:方法中调用请求相似问题列表数据的方法时, 没有做关键词是否为空、关键词是否变化判断, 在关键词有变化时, pageNo没有重新设为0。</p> <ol style="list-style-type: none"> 3、问答页面点击输入框, 弹出的键盘样式问题。 <p>在实现UI的过程中, 没有留意到设计稿中键盘回车键的样式。</p> <ol style="list-style-type: none"> 4、问答页面上拉列表加载更多时, 存在的样式问题。 <p>上拉加载更多时, 调用了父类的showLoading方法多显示了一个菊花; 在上拉常见问题列表加载更多时, 在加载分页数据时没有设置好列表的tableFooterView。</p> <ol style="list-style-type: none"> 5、在屏幕尺寸为3.5寸设备下, 问答模块的发帖页面, 输入法切换为搜狗输入法, 输入内容较多时键盘会挡住一部分内容。 <p>这个问题是因为在实现动态调整UITextView的高度时, 没有考虑到键盘的高度是变化的。解决办法是根据当前输入法的键盘高度来调整UITextView的高度。</p> <ol style="list-style-type: none"> 6、发帖成功页面, 按钮的标题不正确; 手抖返回没有禁用。 <p>这个问题是需求描述有歧义导致的, 需求中没有明确指出按钮的标题有“继续查看教程”、“返回科目二首页”、“返回科目三首页”这三种。</p> <p>总结:</p> <p>上述中绝大部分问题的出现, 是因为在设计功能时, 没有考虑到很小的细节上; 另外一些问题则是因为粘贴复制代码后, 修改了其中一处代码后, 另外一处代码没有做相应的修改, 代码提交SVN前的代码检查做的不够仔细。</p>
周鸿基	<p>在学车宝典2.0版本中, 我主要负责清空答题卡, 问答贴以及我的收藏, 我的错题的优化。</p> <ol style="list-style-type: none"> 1、对需求做需求分析, 列出需求点, 单元测试表 2、对相关知识点列出思路, 实现逻辑, 流程图 3、按着所列的实现逻辑写一个需求点实现 4、对实现的的需求点进行自测 5、所有功能点实现之后, 进行集成测试 <p>接着这个开发模式, 在动手写代码之前花多点时间去思考, 在开发的过程中思路比较清晰, 便开发能够顺利进行。后期的bug也会比较少, 可以减少改bug的时间。</p>
周健威	<p>在这项项目中, 我主要负责的模块包括红包点模块, 活动模块, 专项练习界面和我的圈子样式调整。</p> <ol style="list-style-type: none"> 1.红包点模块涉及的范围比较广, 涉及题库更新, 活动模块, 汽车资讯等等, 而且涉及的情况比较多, 因此在做这个模块的时候逻辑需要谨慎点, 否则容易出bug, 这次这个模块不足的地方主要是对接口考虑不全, 没有考虑到不返回字段的情况。 2.活动模块主要就是加载数据回来显示, 难度不算大, 主要问题出现在一些细节交互上, 比如说看花的显示等。 3.专项练习界面, 主要是UI方面的大整改, 使用的是collectionView来显示数据, 遇到的主要问题是关于性能的问题, 由于界面上涉及很多人的圆角视图, 圆角视图是很消耗性能的, 我在做这个界面的过程中搜索了一些资料, 学习了一些关于渲染的知识, 发现其中一些 [我个人感觉] 平时比较少用到的属性, 可以减轻圆角带来的性能损耗。 4.我的后台主要是UI上的一段小整改, 在做的时候没有考虑到版本兼容问题, 直接根据文案来判断, 后来才发现这种方式不合理, 于是就要求后台返回一个字段来校验。虽然是一些UI上的小整改, 但是它背后却考查了一个人对问题的熟悉程度以及发现问题和解决问题的全局观。
李冠淳	<p>这次学车宝典2.0的开发, 我主要是负责UI这边的处理, 主要包括个人中心页面的UI调整等, 虽然逻辑方面处理的学习也很重要, 但开发逻辑方面的处理比较复杂, 对于作为新手的我而言, 还是由改UI, 稳扎稳打的练习比较好。</p> <p>对于个人中心页面的改版, 因为是我第一次接触全用xib构建的VC, 一开始还是有点不习惯, 因为每个需要修改和引用的都要添加IBOutlet, 一上来满满的@property也是看的眼花缭乱, 这里就体现了命名的科学性和注释的有效性了, 处理过两个专门用xib链接的多元素页面, 一个的命名十分规范, 直接就能找到名称和引用, 节省了不短时间; 而另一个需要通过用关键字查找才能找出来, 本来只需要5s就能找到的时候硬生生花了3min, 这点需要多人协作的时候统一一下名称, 毕竟也有不少一个模块但是名称不同的情况。</p> <p>就控件样式上的改版, 充分体会到与设计充分沟通的重要性, 前后修改了不知道多少次, 而且因为赶工, 很多细节的调整都是在QA之后才进行的, 这点其实不好, 过于追求速度会导致质量无可避免的下降, 以后还是得加效率更多的时间用于Debug, 同时, 因为先做的个人中心页, 而且新改版的科目一四页元素也是比较多, 故直接通过xib进行开发, 经过比较, 对于较少重复元素或模块而言, 无疑还是xib编写编写比较直观而且快速, 然而如果多元素、模块、重复率比较高而言, xib还是累得够呛 (因为测试多设备适配的时候需要考虑更多页面布局的内容, 处理的AutoResizing活着AutoLayout会比较繁琐, 这有部分可能是因为我经验不足导致的)。</p> <p>其余一些工作有个人中心页登录和退出登录的处理, 汽车资讯页的需求等, 由于没什么遇到的bug, 而且整体比较顺利, 就不一一多说了。</p> <p>总体而言基本都是涉及UI的, 接下来项目如果可以的话, 想稍微做一点偏向逻辑处理方面的, 毕竟全面一点好。当然, 关于UI我只了解皮毛中的皮毛, 肯定还有很多东西需要学习和掌握的, 期待下次项目自己经手的模块bug量会吸取这次的教训越来越少。</p>