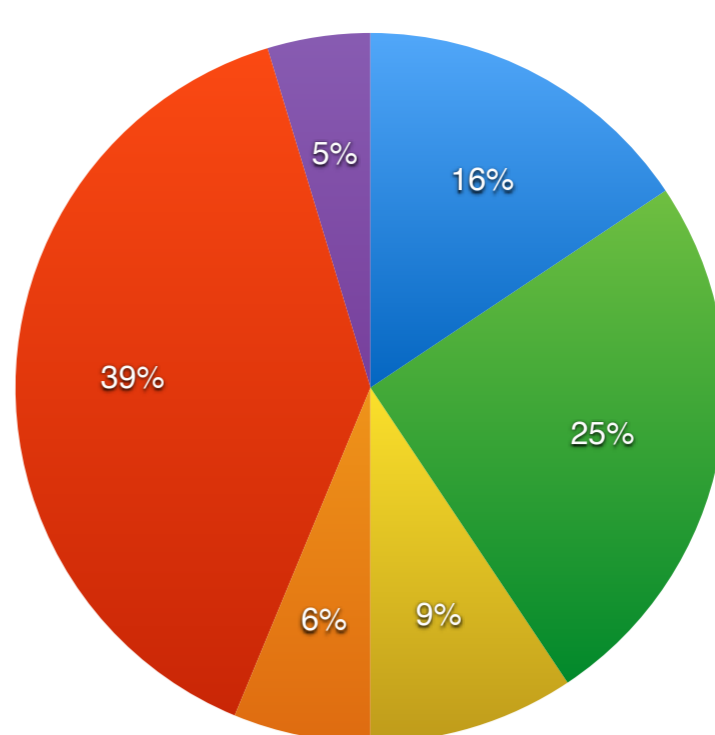


Bug类型

bug类型	数量 (64)	bug类型细分	数量
性能问题	10	数组越界	2
		数据类型	1
		逻辑问题	4
		版本兼容	1
代码逻辑	16	未知原因	1
		逻辑错误	14
		网络机制	2
完整性	6	考虑不全 (数据联动)	4
		考虑不全 (网络差)	1
		考虑不全 (版本兼容)	1
需求变更	4	需求文档不明确	1
		需求文档变更	2
		新增需求	1
用户体验	25	交互处理	9
		样式	15
		兼容性	1
		未知原因	1
其他	3	未知原因	1
		数据调整	2
外部系统	0		

● 性能问题 ● 代码逻辑 ● 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	2
数据类型	1
逻辑问题	4
版本兼容	1
未知原因	1

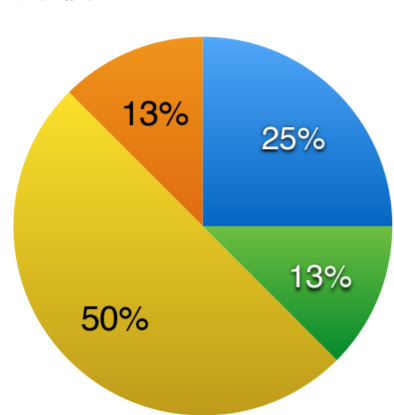
【代码逻辑】类型细分

bug类型	数量
逻辑错误	14
网络机制	2

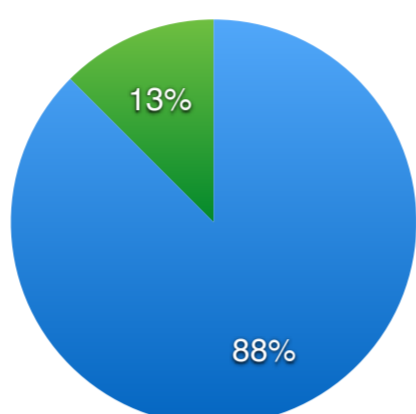
【完整性】类型细分

bug类型	数量
考虑不全 (数据联动)	4
考虑不全 (网络差)	1
考虑不全 (版本兼容)	1

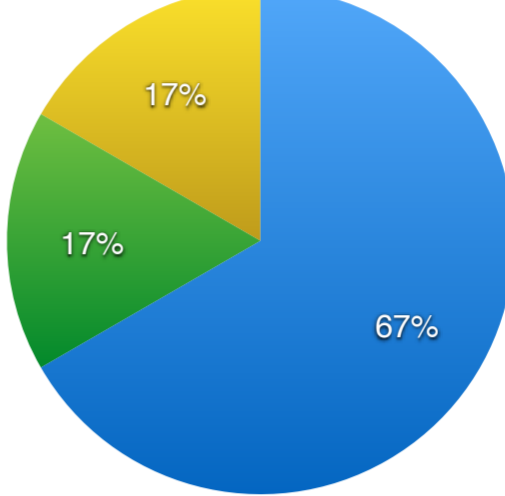
● 数组越界 ● 数据类型 ● 逻辑问题 ● 版本兼容



● 逻辑错误 ● 网络机制



● 考虑不全 (数据联动) ● 考虑不全 (网络差) ● 考虑不全 (版本兼容)



【需求变更】类型细分

bug类型	数量
需求文档不明确	1
需求文档变更	2
新增需求	1

【用户体验】类型细分

bug类型	数量
交互处理	9
样式	15
兼容性	1

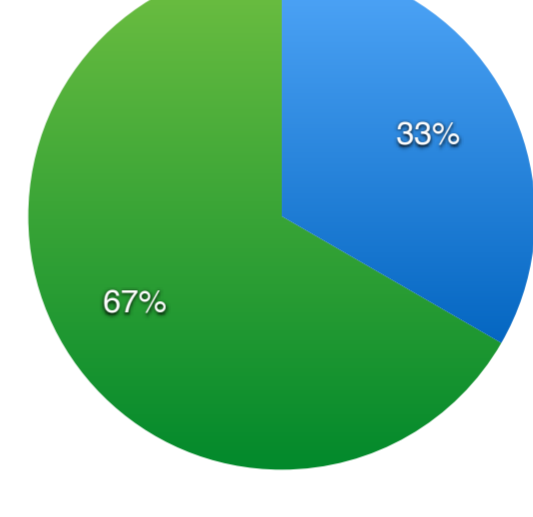
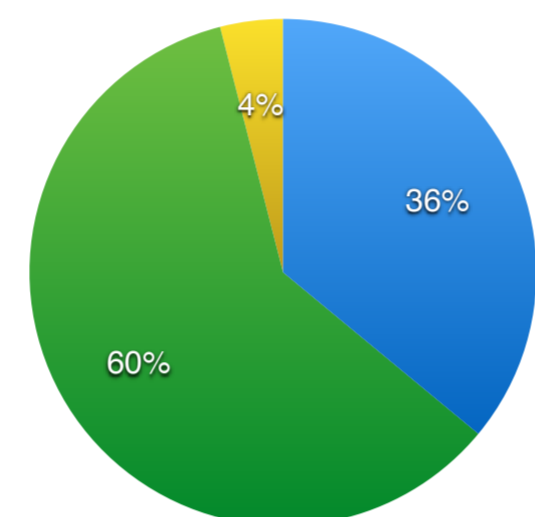
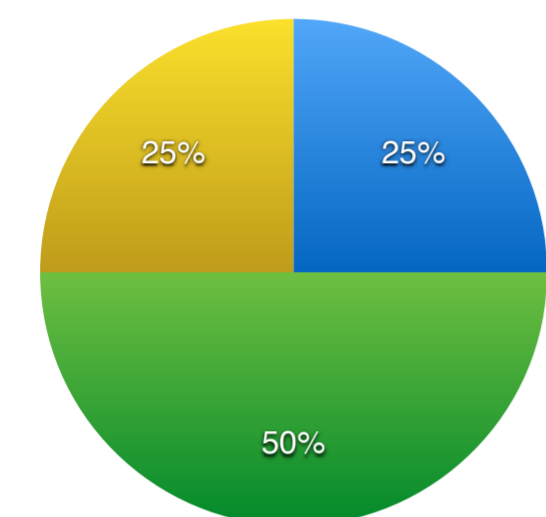
【其他】类型细分

bug类型	数量
未知原因	1
数据调整	2

● 需求文档不明确 ● 需求文档变更 ● 新增需求

● 交互处理 ● 样式 ● 兼容性

● 未知原因 ● 数据调整



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)
黄敏锋	13	7	0.538
黄永坤	13	9	0.692
李冠淳	13	3	0.230
陈东贤	13	5	0.384
历史遗留bug		28	
需求变动		4	
数据库文件导致		4	
何敬东灰度发布测试		4	
iOS开发总计	52	24	0.461

Bug激活情况

BugID	激活次数 (45)	类型	所属 (人)
61451	1	iOS	黄敏锋
61578	2	iOS	李冠淳
61631	1	iOS	陈东贤
61656	1	iOS	陈东贤
61669	1	iOS	黄永坤
61860	1	iOS	黄永坤
61949	1	iOS	黄敏锋
62012	1	iOS	陈东贤
62179	1	iOS	李冠淳
62311	1	iOS	李冠淳

各组员bug激活率

姓名	bug激活数	bug激活数/修复bug总数
黄敏锋	2	0.125
黄永坤	2	0.117
陈东贤	3	0.2
李冠淳	4	0.266

总激活率

bug总数	激活总次数	激活总次数/bug总数
64	11	0.171

项目总结

黄敏锋	<p>这次学车宝典2.4版本开发总体来说比较顺利, 这有赖于团队成员的鼎力合作。</p> <p>这次加入学车宝典项目, 作为项目负责人, 相对于项目的其他成员来说我算个新人, 对于旧版本的功能需求, 代码的熟悉程度都没有其他人透彻, 并且还有其他的研究项目的开发任务在身上, 所以这次项目的开发, 我把自己的工作重点定位为负责项目进度管理和项目质量把控, 而把关键技术的实施交给黄永坤来负责, 只是在出现比较棘手的问题时我才帮忙一起分析和解决问题, 从而能使时间分配得更合理。事实证明永坤表现得很好, 项目也能顺利的进行。</p> <p>这次项目的难点有两个, 一个是从报价宝典移植过来的销量排行榜模块, 一个是新增驾照类型。销量排行榜我安排给冠淳负责, 冠淳的效率很高, 比我预料的时间还要短就完成了移植, 但由于没有在真机上测试, 导致后来我发现内存飙升引发闪退的问题, 出现这个问题我先让冠淳处理, 一段时间没结果, 我让永坤处理, 永坤很快定位到框架, 然后我分析了框架相关代码, 发现是图片解压的方法会导致内存飙升, 去除该方法使得内存不会突然飙升。后来冠淳发现引发内存飙升的另一个原因是使用的图片尺寸太大, 然后修改了请求的图片尺寸, 问题得到解决。</p> <p>另一个难点是新增驾照类型, 数据库结构发生了很大变化, 而且数据库数据也发生了变化。我把数据库的难点工作安排给永坤负责, 总体来说比较顺利, 测试的前面阶段也没发现问题, 只是到公测前一天, qa测升级的时候才发现升级后数据库一个表的数据全错了, 然后永坤找到原因是数据库一个表的主键数据改变了, 在永坤一段时间没有找到解决方案后, 我把问题反映给对旧版本数据库比较熟悉的何敬东, 他找到安卓的同事, 原来是他们提供给我们的数据库, 改了表的字段属性, 还改了表的数据, 然后他们提供修改方法问题得到解决。在这个问题上, 如果我对学车数据库部分的结构了解更多, 会很快解决这个问题, 不必浪费太多时间, 作为项目负责人, 必须对关键技术部分的架构有所认识, 这个我没做好, 在下一个版本需要提高。然后学车部分就交给东贤完成了, 他能独立完成任务, 并且表现出对技术的强烈的探索欲望。</p> <p>除此之外还得感谢产品把很大一部分时间没发确切把握的功能去掉。产品一开始从汽车资讯app截几个图, 来问我开发相关功能模块需要多长的开发时间。我以没有详细需求无法确切回复具体开发时间为由, 建议把功能细分, 分几个版本来逐步实施, 多得产品这么做, 不然这次的开发时间真的不好把握。</p>
黄永坤	<p>这个版本负责了驾照类型切换之后科目一、科目四获取题目信息的逻辑功能修改, 在开发与修复问题的过程中, 意识到:</p> <p>有些会用到的知识不要等到要用的时候再去学习, 虽然不一定要领会得多么透彻, 但至少确保会使用。临时抱佛脚或者只懂些皮毛就去实现与这部分知识有关的功能时, 会力不从心、时间消耗会比预期的多。</p> <p>在修复QA发现的问题时, 查看其所期望的效果是否已经列在需求中, 避免在修复存在问题的过程中去实现“新增”需求。在修复问题的过程中去实现新功能, 会引出新问题。因为没有足够时间去理解需求、梳理清楚这部分功能和已实现功能之间的关系, 很容易逻辑设计的不够全面、测试的不够全面、没有预测到会遇到的问题等等。</p> <p>除此之外, 还要学会调节个人情绪, 这次版本开发的过程中, 将个人情绪带入到工作中的次数比较多。</p>
陈东贤	<p>我在这个版本中主要是负责选车页的头部视图、部分的iOS优化。对于我负责模块产生的bug原因有:</p> <ol style="list-style-type: none"> 1.选车页的实现机制而导致的样式问题,在其他页面播放全屏视频导致选车页的宽高颠倒,让样式出现问题,虽然很难会想到这全屏播放的这个bug,但归根到底还是自己在代码上没做好预防预防措施以及没有彻底理解autolayout的实现原理. 2.选车页的轮播图实现方式,这个也是上面的bug修改后导致的,当时改的时候,没有检查清楚,把页面实现放在了viewDidAppear上而导致没次请求都会从新创建,从而会出现闪一下的情况. 3.模拟考的弹框实现的逻辑问题,没了解好旧版本类似的需求,出现了获得学分弹框和好评弹框的冲突. 4.以及在修复优惠页遗留bug的时候没理清逻辑需求,草率修复而导致出现了新的bug. <p>经过了这个版本,我对autolayout和继承以及有了跟深一层的理解,而且也用了很多以前不喜欢用的block,发现其实也挺便利的.</p>
李冠淳	<p>这次学车 v2.4 项目, 我主要负责选车页面销售排行榜的移植、小部分的题库适配, 还有部分的优化点。这次项目, 由于对汽车报价宝典的代码研究不深, 吃了很多亏。</p> <p>销售排行榜在汽车报价宝典是有完整的一个功能的, 而且进出逻辑基本一样, 一开始我觉得保证代码统一性 (2.0的时候报价 SDK 是基本复用当时的版本的), 以便日后维护, 所以将一块代码逻辑复用了过来。然而由于对报价这边代码理解不深, 迭测之后才发现有不少坑。汽车报价宝典这边复用率很高, 导致有一些问题会涉及到很多的功能。不仅仅是样式问题, 复用率高导致的回调也相对复杂。</p> <p>经过这次项目, 我清楚的体会到我基本功还是不够扎实。销售排行榜这边的那个 crash, 我找了1天还不能成功定位把能修的, 最后让测试的时候积极尝试, 尽量降低迭测时候产生的bug数量。另外, 一些有争议的 bug 需要更积极的拉上相关负责人 (QA、产品、设计等) 一起讨论确认最终效果, 这样不仅提高了 bug 的修复效率, 还有效减少 bug 因为被误解修复后被激活的次数。期望在之后的项目中能不断的提升自己理解代码的能力, 提高项目效率。</p>