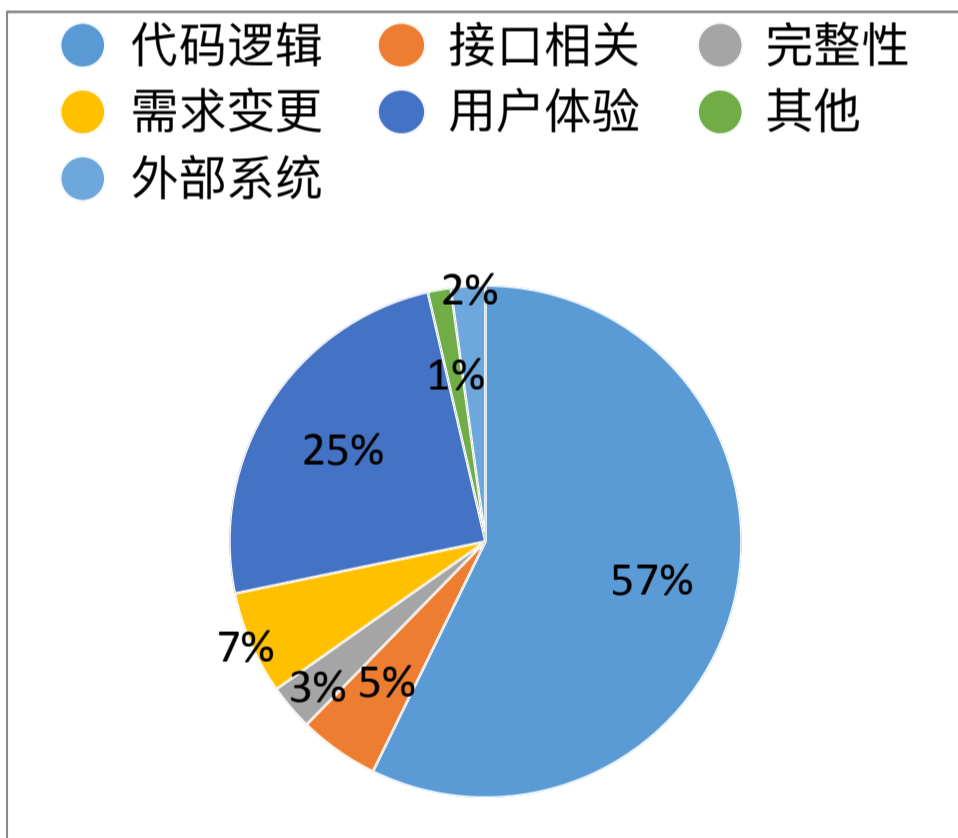


表格 1

bug类型	bug类型细分				
性能问题	数组越界				
	数据类型				
	逻辑问题				
	版本兼容				
	未知原因				
代码逻辑	79 逻辑错误		75		
	网络机制		4		
接口相关	7 接口问题				
	环境问题				
	请求参数问题				
完整性	4 考虑补全（数据联动）		1		
	考虑补全（网络差）				
	考虑补全（多用户）				
	考虑补全（版本兼容）		2		
	边界测试		1		
需求变更	9 需求文档没明确		3		
	需求文档变更		5		
	新增需求		1		
用户体验	34 交互处理		9		
	样式		25		
	兼容性				
其他	2 未知原因				
	数据调整		2		
历史遗留					
外部系统	3				



日均产生bug率

工作人日	iOS开发的bug总数	bug数/工作人日		
455	139	0.305494505494500		

组员日均bug产生率

组员	工作人日	开发的bug总数	bug数/工作人日
陈东贤	65	20	0.30769230769230
许伟杰	65	59	0.90769230769230
潘思嘉	65	27	0.41538461538461
林俊炳	65	33	0.50769230769230

Bug激活情况

BugID	激活次数	类型	所属	
99588	1		陈东贤	
99641	1		陈东贤	
99706	1		陈东贤	
99806	1		俊炳	
99974	1		俊炳	
100027	1		俊炳	
100567	1		俊炳	
100828	1		俊炳	
99148	2		潘思嘉	
100283	1		潘思嘉	
100773	1		潘思嘉	
100397	1		潘思嘉, 林俊炳	
99135	1		思嘉	
100069	1		许伟杰	
100140	1		许伟杰	
100384	1		许伟杰	
100396	1		许伟杰	
100423	1		许伟杰	
100441	1		许伟杰	
100456	2		许伟杰	
100619	1		许伟杰	
100843	1		许伟杰	

激活率

姓名	bug总数	激活总次数	激活总次数/bug总数	
陈东贤	20	3	0.15	
许伟杰	59	9	0.152542372881356	
潘思嘉	27	5	0.185185185185185	
林俊炳	33	6	0.181818181818182	

表格 1

许伟杰	<p>这次做全新的项目口袋蜜蜂1.0版本，我主要负责海报、文章列表、营销h5、IM消息、推送、分享、魔方计数等功能的开发。这次的bug还是很多的，首先是因为是新的项目，功能比较多，有些地方自己考虑不周全。其次是开发过程需求有变动，有时改一点需求后可能会影响原本正常的功能，如天气海报的需求由单张海报改为列表，影响到了其它海报的功能。再其次是自己有时没有及时沟通解决，例如推送的文案，没有及时反馈沟通，没有及时解决，到后面有些与需求有出入的问题就被自己遗忘了。今后要认真对待每一个问题，不关乎大小，都要积极及时地解决。当然，总体上自己还是认真地全身心地投入到项目中，最终项目在大家共同努力下按时完成开发。</p>			
陈东贤	<p>口袋蜜蜂是个新项目，从零开始去构建整个项目。模块也比较多，登录，素材首页，还有融云和rn的发现。我负责的主要是项目的框架和素材首页，以及一些“我的”模块的页面。这次是我第一次参与到有rn模块的项目，导致项目初期会有和负责rn的同事沟通不到位的情况。也由于项目周期比较长，我在其中还参与报价两个小版本的开发，导致专注度下降，没有及时发现到一些需求和接口的变动，出现了很多不必要的bug。不过最大原因还是本身不够专注的原因，像一些文章页、海报加载慢其实可以很早就发现并及时和mrobot沟通解决的。</p>			
林俊炳	<p>概况：经过电脑资讯客户端、家居杂志等App中部分模块的积累，RN在口袋蜜蜂中的应用算是比较顺利，这也应该提及一开始做功能评估时，原生与RN分工的情况。目前App中容易出现OOM的列表、功能偏复杂的IM基于原生实现，其他界面则由RN负责。项目涉及的数据，基本流转都是在原生层面，RN的专项工作是绘制UI。实际开发中，需要做好原生与RN的数据交互工作。</p> <p>相关总结： 组件产出（PCFetch、PCImage、PCToast、PCModule）：基于公司网络框架封装的PCFetch和PCImage，很好地处理了IPV6和HTTPDNS问题。实践下来Android端的PCFetch偶尔会出现回调多次导致crash的问题（请求频繁时，上次请求未响应时，发起了新的请求？），尚未定位到问题，仍需进一步完善。PCModule中有个别方法可抽取出来作为通用组件，在其他App使用，后续计划维护一个PCCommonModule，比如魔方的一些方法，皆可在这里进行封装。</p> <p>关于bug： 严重级别的问题中，有两个（100270、100820）是RN组件功能不完善，自己的处理方式不稳妥导致crash或是功能失效，后在issue中找到相应解决办法；其他的（100419、100828）则是代码逻辑不严谨导致； 需求文档的更新说明，偶尔会遗漏某个点，自己又没有去重新翻看具体页面需求，导致个别需求bug，这部分包括了需求变更和新增需求； 除以上两点，其他的基本是一些样式适配问题，也有部分是逻辑边界处理不严谨导致。 总的来说，原生与RN的混编已经有了一个基本的模式，后续需关注桥接组件的通用性和可靠性，使之经得起不同项目考验；另外即是扩大和维护组件库。</p>			