**开源课程期末作业**

**分析一个 GitHub 2015-2021 任一年日志数量 TOP50 的开源项目**

**基本信息**

小组成员：秦长久51215903107，吴贤佑51215903108，刘添然51215903077

组内分工：

秦长久：数据挖掘与可视化

吴贤佑：信息汇总报告撰写

刘添然：项目基本信息收集

项目序号: ？？

项目id: 29028775

项目名: facebook/react-native

**一、项目的基本背景和发展历程介绍**

* **技术类型**

React Native是Facebook于2015年4月开源的跨平台移动应用开发框架，是Facebook早先开源的JS框架 React 在原生移动应用平台的衍生产物，目前支持iOS和安卓两大平台。RN使用Javascript语言，类似于HTML的JSX，以及CSS来开发移动应用，因此熟悉Web前端开发的技术人员只需很少的学习就可以进入移动应用开发领域。

React Native使开发者能够在Javascript和React的基础上获得完全一致的开发体验，构建世界一流的原生APP，并着力于提高多平台开发的开发效率 —— 仅需学习一次，编写任何平台。(Learn once, write anywhere)。Facebook已经在多项产品中使用了React Native，并且将持续地投入建设React Native。

* **版本发布历史**

共199次更新，其中a2（Alexsander Akers）在2015.03.27贡献了第一个版本（v0.1.0: Initial public release）。目前的最新稳定版本（v0.68.2）是由fortmarek（Marek Fořt）在2022.05.09贡献的。而cortinico（Nicola Corti）则在2022.06.01贡献了v0.69.0-rc.6。值得一提的是到目前为止该项目都没有跨越v0。

**2015.03.27**首次上线**v0.1.0**，来自a2：

1. 从FAQ中删除无关的问题。

**2018.05.15**由vjeux发布最早的带说明版本**v0.4.3**：

1. 新功能：

Websocket polyfill；

AlertIOS.prompt ()；

letterSpacing <Text>样式属性；

borderTopLeftRadius borderTopRightRadius…<View>样式属性；

2. 主要变化：

ref. gettnativehandle()和ref.getNodeHandle()将不复存在。取而代之的是React.findNodeHandle(ref)；

如果之前正在使用没有文档的createReactIOSNativeComponentClass或ReactIOSViewAttributes来创建iOS包装器，现在应该使用requireNativeComponent；

3. 升级：将flow升级至0.11，jest升级至0.4.2；

4. Bug修复：

修复了内部有空间的文件夹的软件包；

修复RCTImageLoader的线程安全问题；

正确支持<ScrollView>上的overflow: 'visible'；

如果组件已卸载，就不能再调用度量回调；

不再需要ruby依赖；

UIExplorer图标可显示在主屏幕上；

更改左侧图标现在可以在<NavigatorIOS>上工作；

修复<Navigator>的边界情况。

**2022.05.09**最新稳定版本**v0.68.2**，来自fortmarek：

1.变更：将react-native-codegen的版本升级为0.0.17 (by @cortinico)；

2.修复：在Android平台上解决Windows上的长路径限制(by @mganandraj)。

**2022.05.11-2022.06.01**由fortmarek和cortinico陆续发布开发版本**v0.69.0-rc1至rc6**。

* **主要贡献者的构成（国家、区域和组织等）**

贡献者一共有2302位。

shergin（Valentin Shergin），来自Austin, TX，就职于Coinbase；

JoshuaGross（Joshua Gross），来自Bay Area, CA，就职于Facebook；

RSNara（Ramanpreet Nara），就职于Facebook；

hramos（Héctor Ramos），来自Scottsdale, AZ，就职于Facebook；

rickhanlonii（Ricky），来自EST. AKRON，就职于Facebook；

cortinico（Nicola Corti），来自London, UK；

fortmarek（Marek Fořt），来自Berlin, Germany，就职于Shopify。

**CI/CD 的使用**

目前包含6条工作流程，共运行52505次且还在不断增加。

应用于issue的版本标记：1252次；

问题注意：34968次；

标识旧issue和PR：240次；

issue标记：6496次；

PR的危险运行：193次；

测试安卓容器镜像：2331次。

* **其他有价值的信息**

**主要功能**

* 原生的 iOS 组件

有了 ReactNative，你可使用标准平台组件，比如 iOS 平台上的 UITabBar 和 UINavigationController。这可以让你的应用程序拥有和原生平台一致的外观和体验，并保持较高的品质。使用相应的 React 组件，如 iOS 标签栏和 iOS 导航器，这些组件可以轻松并入你的应用程序中。

* 异步执行

JavaScript 应用代码和原生平台之间所有的操作都是异步执行，并且原生模块也可以使用额外线程。这意味着开发者可以解码主线程图像，并将其在后台保存至磁盘，在不阻塞 UI 的情况下进行文本和布局的估量计算，等等。因此，React Native 应用程序的流畅度和响应性都非常好。通信也是完全可序列化的，当运行完整的应用程序时，这允许我开发者使用 Chrome Developer Tools 来调试 JavaScript，或者在模拟器中，或者在真机上。

* 触摸处理

iOS 有一个非常强大的系统称为 Responder Chain，可以用来响应复杂视图层级中的事件，但是在 Web 中并没有类似功能的工具。React Native 可实现类似的响应系统并提供高水平的组件，比如 TouchableHighlight，无需额外配置即可与滚动视图和其他元素适度整合。

* 弹性框和样式

布局视图应该是简单的，所以 Web 平台上的弹性框模块也被引入了 React Native。弹性框可用来搭建最常用的 UI 布局，比如代用边缘和填充的堆叠和嵌入。React Native 还支持常见的 Web 样式，比如 fontWeight 和 StyleSheet 抽象，它们提供了一种优化机制来宣称你所有的样式和布局在组件中的应用是正确的，且组件把它们应用到了内网中。

* Polyfills

React Native 的重点是改变视图代码编写的方式。可以使用 npm 安装 JavaScript 库，这些库用于融入到 React Native 中的顶级功能，比如 XMLHttpRequest，window.requestAnimationFrame 及 navigator.geolocation。团队正在扩大可用的 API，并致力于为开源社区做出贡献。

* 可扩展性

使用 React Native 无需编写一行原生代码即可创建出一款不错的应用程序，并且 React Native 可通过自定义原生视图和模块来进行扩展--也就是说你可以重用你已经构建的任何内容，并且可导入和使用你最喜欢的原生库。为了在 iOS 中创建一个简单的模块，需要创建一个新的类来实现 RCTBridgeModule 协议，并将你想要在 RCT\_EXPORT\_METHOD 中对 JavaScript 可用的功能包装起来。另外，类本身必须可以用 RCT\_EXPORT\_MODULE() 显式导出；自定义的 iOS 视图可以通过子类化 RCTViewManager，实现 -(UIView \*)view 方法并用 RCT\_EXPORT\_VIEW\_PROPERTY 宏导出属性的办法来公开。然后一个简单的 JavaScript 文件会连接这些点。

**二、项目的历史轨迹分析**

1. **每月新增 Star 和 Frok 的个数**

表1 每月新增Star数量

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 时间 | 数量 | 时间 | 数量 | 时间 | 数量 | 时间 | 数量 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |



表2 每月新增Fork数量

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 时间 | 数量 | 时间 | 数量 | 时间 | 数量 | 时间 | 数量 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

1. **每月打开 Issue 和 关闭 Issue 的个数**

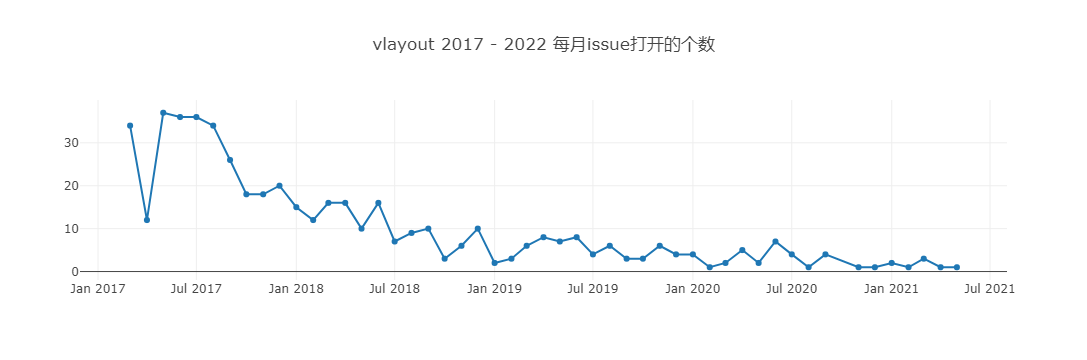


表3 每月打开Issue数量

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 时间 | 数量 | 时间 | 数量 | 时间 | 数量 | 时间 | 数量 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

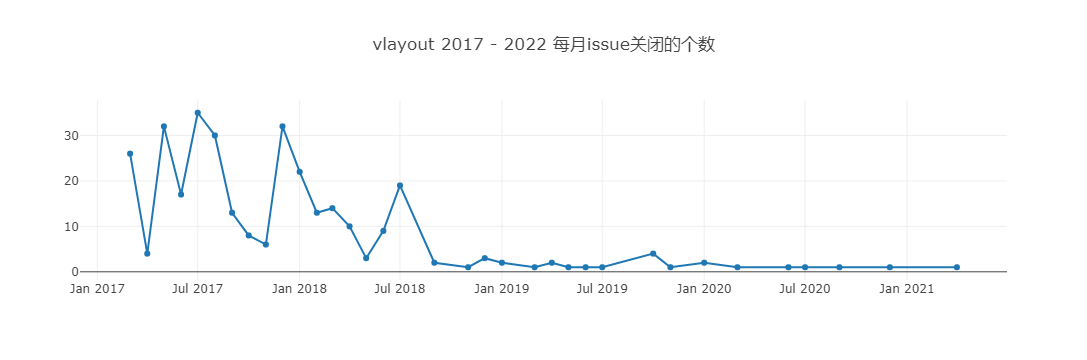


表4 每月关闭Issue数量

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 时间 | 数量 | 时间 | 数量 | 时间 | 数量 | 时间 | 数量 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

1. **每月打开 PR 和合入** **PR 的个数**



表5 每月打开PR的数量

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 时间 | 数量 | 时间 | 数量 | 时间 | 数量 | 时间 | 数量 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |



表6 每月合入PR的数量

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 时间 | 数量 | 时间 | 数量 | 时间 | 数量 | 时间 | 数量 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

1. **每月在仓库中活跃的不同开发者总数**



表7 每月在仓库中活跃的不同开发者的数量

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 时间 | 数量 | 时间 | 数量 | 时间 | 数量 | 时间 | 数量 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

1. **Issue 从打开到关闭的平均时长和中位数（单位：天）**

项目创建至归档期间Issue 从打开到关闭的时长中位数：天。





1. **PR 从打开到合入的平均时长和中位数（单位：天）**

项目创建至归档期间PR从打开到合入的时长中位数：天。

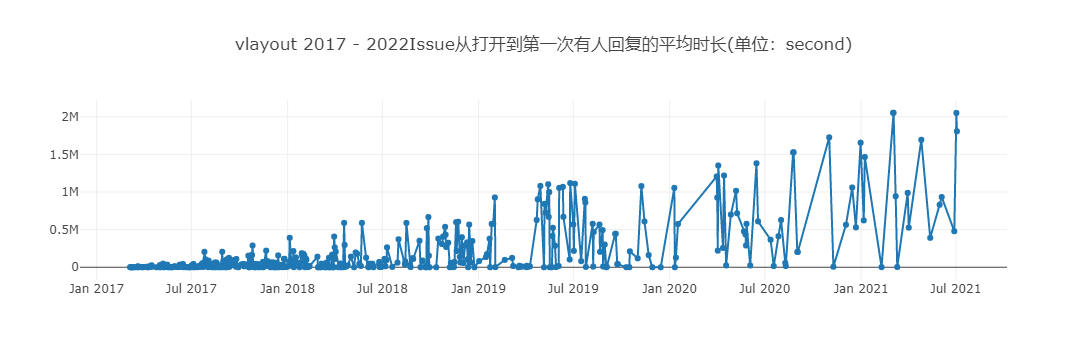


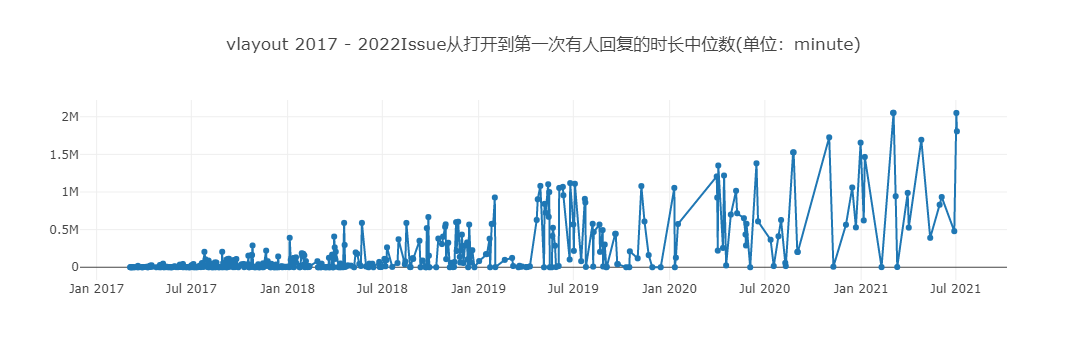


1. **Issue和PR从打开到第一次有人回复（非本人回复）的平均时长和中位数（单位：天）**

项目创建至归档期间Issue 从打开到第一次有人回复的时长中位数：天。

项目创建至归档期间PR从打开到第一次有人回复的时长中位数：天。









**三、结合期中分析的归档项目，对比分析活跃/归档项目**

* **项目数据对比分析**
  + **项目基础数据的变化趋势**

* + **开发者数量变化趋势**

* + **其他你感兴趣的对比方向**

* **项目发展到活跃/归档的主要影响因素及原因**

**部分可能资料参考：**

Airbnb起初是活跃的使用者，但之后放弃了使用

中间有重构过，寻找临界点，把简要说明放到版本发布等地方

从提交统计来看，贯穿全周期开发的贡献者较少，大部分贡献者只参与了一段时间的开发，后来可能由于不再使用而离开团队

Google Flutter的竞争：

由于Facebook在其平台上仅拥有大量受众，因此编码人员可以确保他们能够覆盖更广泛的受众。Instagram还将React Native与其他大公司一起使用，如Skype，特斯拉，Airbnb和沃尔玛。

由于Flutter是如此新颖，它没有任何真正展示给企业主或公司进行转换的坚实理由。此外，一些开发人员担心Dart很容易被淘汰，就像谷歌通过各种升级消除AngularJS一样，名称改为简单的Angular。

这阻碍了广泛的采用，因为许多开发人员可以轻松地浪费时间开发应用程序，如果Google停止支持Dart或消除Flutter，这些应用程序很容易变得无效或需要彻底检修。

尽管Google Flutter为希望定制跨平台应用程序的开发人员带来了希望，但其限制和覆盖大量受众的能力将使开发人员难以切换。许多IDE与Flutter不兼容的事实也使得程序员更难以接受它，因为他们必须适应新的平台。

因此，Facebook的React Native并不难以超越Flutter。通过利用这个功能强大且广泛采用的框架，您可以确保您的开发团队使用的工具可以为跨平台应用程序的开发和部署提供长期结果。