

عنوان: چطور باید یک پروژه سورس باز را خوب مدیریت کرد؟

نویسنده: وحید نصیری

تاریخ: ۱۳۹۲/۰۶/۲۸

آدرس: www.dotnettips.info

برچسب‌ها: Tips, Open Source, Project Management

اگر مایل هستید که پروژه خود را به صورت سورس باز ارائه دهید، نیاز است یک سری شرایط را رعایت کنید تا کاربران این پروژه بتوانند به سادگی از آن استفاده نمایند.

- فایل ReadMe را فراموش نکنید

حتی اگر پروژه شما از یک سایت اختصاصی استفاده می‌کند، اولین محلی که عموم کاربران برای دریافت اطلاعات کار با پروژه، به آن مراجعه می‌کنند، فایل ReadMe برنامه است. این فایل می‌تواند حاوی مشخصات ذیل باشد:

الف) وابستگی‌های پروژه را مشخص کنید

واقعیت این است که برخلاف شمای برنامه نویسی، عموم استفاده کنندگان، آشنایی چندانی با جزئیات محیط و شرایط تهیه برنامه شما ندارند. به این ترتیب بسیاری از مسایلی که برای شما بدیهی هستند، برای عموم اینگونه نخواهند بود. بنابراین مساله‌ای که به سرعت می‌تواند سبب خشم کاربران و صرفنظر از کار شما گردد، مشخص نبودن نحوه نصب و وابستگی‌های لازم برای اجرای برنامه است.

ب) وضعیت بلوغ پروژه خود را مشخص کنید

آیا از این برنامه، مدتی است که در محیط کاری استفاده می‌کنید؟ آیا به نظر شما هنوز ناتمام است؟ آیا API کتابخانه شما در نگارش بعدی کاملاً دگرگون خواهد شد؟ تمام این مسایل و سؤالات را به نحو واضحی توضیح دهید و مشخص کنید. همین توضیحات کوتاه می‌تواند ساعت‌های بسیاری از زندگی دیگران را صرفه جویی کند.

ج) اگر پروژه شما یک کتابخانه است، نوع زبان و Runtime‌های پشتیبانی شده را مشخص کنید

برای مثال اگر یک کتابخانه دات نتی را ارائه می‌دهید، مشخص کنید که از کدام نگارش دات نت به بعد را پشتیبانی می‌کنید.

د) مجوز استفاده از پروژه را مشخص کنید

مطلب [مقایسه مجوزهای سورس باز](#) را یکبار مطالعه نمائید و سپس مجوز صحیحی را برای کار خود انتخاب کنید. همچنین آن را به نحو واضحی در مستندات پروژه خود قید نمائید. به علاوه به‌خاطر داشته باشید که امکان ارائه مجوزهای دوگانه مانند AGPL نیز وجود دارند. در این حالت کاربر یا باید سورس محصول خودش را ارائه دهد، یا مجوز کتابخانه شما را خریداری کند. مانند RavenDB که از این نوع مجوز استفاده می‌کند.

- یک پروژه نیاز به مستندات دارد

مستند سازی کار، سخت و زمانبر است؛ اما بهترین لطفی است که می‌توانید به کاربران خود نمائید. مستندات نه تنها زمان جستجوی بسیاری را صرفه جویی خواهند کرد، همچنین حس اطمینان خاطر را به کاربر القاء می‌کنند. از این جهت که احساس می‌کنند شما برای کارتان ارزش قائل بوده‌اید و احتمال اینکه این برنامه در آینده نزدیک به یک abandonware تبدیل شود، کم است (منظور یک برنامه فراموش شده و خاتمه یافته).

- به روز رسانی را ساده کنید

بالاخره زمانی نیاز خواهد بود تا نگارش جدیدی از کار خود را ارائه دهید. در این حالت نیاز است یک سری از شرایط را مدنظر داشته باشید:

الف) سازگاری قبلی را مدنظر داشته باشید

یکی از بدترین حالات به روز رسانی یک کتابخانه زمانی است که کاربر آن با ده‌ها خطای کامپایل حاصل از به روز رسانی مواجه شود. اگر نیاز است قسمتی از کد خود را حذف کنید یا تغییر دهید، استفاده از ویژگی [Obsolete](#) را فراموش نکنید و اینکار باید مرحله به مرحله انجام شود. در یک نگارش، ویژگی Obsolete را معرفی کنید. در دو نگارش بعد، API را تغییر دهید.

ب) حتماً یک Change log را تکمیل کنید

پس از ارائه یک نگارش جدید، حداقل در چند سطر مشخص کنید که چه مواردی تغییر کرده‌اند، چه مواردی اضافه شده‌اند و چه مواردی را حذف کرده‌اید. همچنین اگر مواردی تغییر کرده‌اند، نحوه ارتقاء کدهای قدیمی را به نگارش جدید، شرح دهید. اگر مورد جدیدی اضافه شده‌است، لینکی را به مثالی درباره‌ی آن ارائه دهید.

- نگارش‌های جدید را اعلام کنید

برای مثال در طی ارائه یک مطلب جدید در وبلاگ خود، ارائه نگارش جدیدی از کتابخانه یا برنامه خود را به عموم اعلام کنید. در این حالت، حتماً لینکی را به `change log`، ارائه داده و مشخص کنید که وضعیت سازگاری آن با قبل چگونه است.

- محلی را برای دریافت بازخوردهای پروژه خود مشخص کنید

نیاز است بتوانید پروژه خود را پشتیبانی کنید یا به سؤالات مربوطه پاسخ دهید. اگر سورس کنترل یا برنامه مدیریت پروژه شما، امکان پرسش و پاسخ را دارد، که بسیار خوب. اگر خیر، می‌توانید مثلاً یک گروه گوگل جدید و امثال آن را برای دریافت بازخوردهای پروژه ایجاد کنید. همچنین نیاز است لینک به این محل را در فایل README پروژه به صراحت مشخص کنید.

- گذر از پروژه

بالاخره روزی فراخواهد رسید که دیگر علاقه‌ای به نگهداری پروژه نداشته باشید. این مساله را در مکان جمع آوری بازخوردهای خود اعلام کنید یا شخص دیگری را به نگهداری پروژه دعوت نمایید. اگر این کار را انجام ندهید، سبب خواهید شد `fork`های متعددی از این پروژه بی‌جهت ایجاد شده و در نهایت مشخص نباشد که کدامیک بهتر است و کدامیک مشکلات کمتری دارند.