عنوان: **چطور باید یک پروژه سورس باز را خوب مدیریت کرد؟**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۶/۲۸ ۱۲:۰
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 نویس، 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 کاملا دگرگون خواهد شد؟ تمام این مسایل و سؤالات را به نحو واضحی توضیح دهید و مشخص کنید. همین توضیحات کوتاه می‌توانند 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 آن‌را 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) حتما 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، حتما 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 مثلا یک گروه گوگل جدید و امثال آن‌را 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 نمائید. اگر این کار را انجام ندهید، سبب خواهید شد forkهای متعددی از این پروژه بی‌جهت ایجاد شده و در نهایت مشخص نباشد که کدامیک بهتر است و کدامیک مشکلات کمتری دارند.

عنوان: **چگونه در یک پروژه سورس باز مشارکت کنیم؟**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۱۶ ۱۲:۰
آدرس: www.dotnettips.info
گروه‌ها: Open Source

مشارکت در پروژه‌های سورس باز الزاما به معنای هدیه کدهای جدیدی به آن پروژه یا حتی مشارکت مالی در آن نیست. در ادامه لیستی از مواردی را مرور خواهیم کرد که سبب زنده نگه داشته شدن یک پروژه سورس باز خواهند شد:

مشارکت در نگهداری پروژه

مشکلی را در این کتابخانه پیدا کرده‌اید؟ آن‌را در سیستم bug tracking پروژه گزارش کنید و بی‌تفاوت از کنار آن عبور نکنید.
مشکلی برطرف شده است؟ بررسی کنید، آیا واقعا این تغییرات مفید بوده است یا خیر و نتیجه را اعلام کنید.

بهبود کدهای موجود

در بهترین حالت، کدی را جهت رفع یک مشکل ارسال کنید. همچنین در این حالت سعی کنید یک مطلب جدید را ایجاد کرده و در مورد کدهای خود توضیح دهید. برای ارسال کدی جدید بهتر است تنها قسمت‌های تغییر کرده را ارسال کنید و اصطلاحا باید بتوانید توسط قابلیت‌های سورس کنترل‌ها یک patch را تهیه نمائید.
یک unit test جدید را به پروژه اضافه کنید. یک مثال جدید را برای قسمتی خاص تهیه نمائید.
ثوابت برنامه را به زبان‌های دیگر ترجمه کنید.
یک گزینه و قابلیت جدید را درخواست دهید.

بهبود مستندات پروژه

اگر توضیحات قسمت‌های مختلف و commentهای ارائه شده به نظر شما کافی نیست؛ آن‌ها را تکمیل کرده و یک patch برای آن ارائه دهید.
مستندات موجود را تکمیل کنید یا بهبود ببخشید.
یک مقاله‌ی جدید در مورد نحوه‌ی استفاده از آن پروژه بنویسید.
یک ویدیوی ساده آموزشی را در مورد آن تهیه کنید.

مشارکت در انجمن‌ها و شبکه‌های اجتماعی

به لیست سؤالات مطرح شده در یک پروژه مراجعه کرده و در آن مشارکت کنید. سعی کنید حضور مثبتی داشته باشید.
به دیگران در مورد وجود این پروژه اطلاع رسانی کنید.
اگر پروژه مفیدی است، به دیگران بگوئید این پروژه چقدر بر روی کار شما تاثیر داشته است و چه برنامه‌هایی را از طریق آن پیش برده‌اید.

نظرات خوانندگان

نویسنده: بهاره
تاریخ: ۱۳۹۲/۰۹/۱۶ ۱۵:۲۳

من خیلی وقت بود دلم می‌خواست تو یه همچین پروژه هایی شرکت کنم ..خواهش می‌کنم ادامه بدید..معرفی کنید چند تا از این پروژه‌ها ..عملی ...مخصوصا اونایی که با برنامه bazzar کار می‌کنند و اپن سورس اند.....سپاس

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۱۶ ۱۷:۴۱

تعدادی پروژه سورس باز در سایت جاری هستند: اینجا
می‌تونید مشارکت کنید، مقاله در موردشون بنویسید، مستندات درست کنید و موارد دیگری که در بحث عنوان شده.

نویسنده: پیام
تاریخ: ۱۳۹۲/۰۹/۲۶ ۱۸:۲۸

سلام؛ من دوست دارم یک پروژه اپن سورس رو در اینترنت قرار بدهم. علاقه شخصی ام این است که آن را در سایت خودم قرار بدم. آیا الزاما باید نرم افزارهای اپن سورس در سیستم‌های سورس کنترل مانند Git منتشر شوند یا الزامی خاصی در این موضوع وجود ندارد. مثلا من می‌خواهم فعلا پروژه را در یک وبلاگ قرار دهم و با کامنت گذاری علاقه مندان نظرات آن‌ها را هم در پروژه اعمال کنم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۲۶ ۲۱:۲۸

- یک پروژه خوب که «باید» از سورس کنترل استفاده کند. حداقل بهتر است روی سیستم خودتان نصب باشد. از این کار پشیمان نخواهید شد . اما حالت ارائه عمومی آن، نه؛ الزامی ندارد که حتما روی سورس کنترل‌های عمومی باشد، مگر اینکه مشارکت زیادی را پس از مدتی جلب کند.
- ضمنا مطلب « چطور باید یک پروژه سورس باز را خوب مدیریت کرد؟ » شاید برای حالت شما مفیدتر باشد.

عنوان: آشنایی با ساختار یک Pull Request خوب
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۲/۱۳ ۱۶:۵۰
آدرس: www.dotnettips.info
گروه‌ها: Git, Open Source, GitHub

در مطلب « نحوه‌ی مشارکت در پروژه‌های GitHub به کمک Visual Studio » با مفهوم pull request آشنا شدیم. اما ... یک pull request خوب چه خصوصیاتی دارد و فرهنگ ارسال یک PR خوب چیست؟

اخلاق مشارکت در یک پروژه‌ی سورس باز

بعضی از توسعه دهنده‌ها در حین مشارکت در یک پروژه‌ی سورس باز، برای مثال جهت افزودن قابلیتی جدید و یا رفع مشکلی، ابتدا سعی می‌کنند تا کدهای فعلی را برای خودشان «قابل فهم‌تر» کنند. این قابل فهم‌تر کردن پروژه، شامل تغییر نام متغیرها و متدهای فعلی، انتقال کدهای موجود به فایل‌هایی دیگر یا حتی یکی کردن چندین فایل با هم، مرتب سازی متدهای یک کلاس بر اساس حروف الفباء و امثال آن می‌شود.
این کارها را **نباید** در حین مشارکت و توسعه‌ی پروژه‌های سورس باز دیگران انجام دهید! اگر هدفتان رفع مشکلی است یا افزودن قابلیتی جدید، باید نحوه‌ی کدنویسی فعلی را حفظ کنید. از این جهت که نگهدارنده‌ی اصلی پروژه، پیش از شما این‌کار را شروع کرده‌است و زمانیکه شما به پروژه‌ای دیگر رجوع خواهید کرد، باز نیز باید همین کار را ادامه دهد.
اگر refactoring گسترده‌ی شما به هر نحوی سبب بهبود پروژه‌ی اصلی می‌شود، ابتدا این مورد را با مسئول اصلی پروژه مطرح کنید. اگر او قبول کرد، سپس اقدام به چنین کاری نمائید.

بحث در مورد تغییرات پیش از ارسال PR

قبل از اینکه PR ایی را ارسال کنید، بهتر است یک issue یا ticket جدید را باز کرده و در مورد آن بحث کنید یا توضیح دهید. در این حالت ممکن است توضیحات بهتری را در مورد سازگار سازی تغییرات خود با کدهای فعلی دریافت کنید.

Pull requestها را کوچک نگه‌دارید

برای اینکه شانس قبول شدن PR خود را بالا ببرید، حجم و تمرکز آن‌را کوچک نگه دارید. بسیاری از توسعه دهنده‌های سورس باز اگر با یک PR حجیم روبرو شوند، آن‌را رد می‌کنند چون مشکل اصلی، مدت زمان بالایی است که باید جهت بررسی این PR اختصاص داد. هرچقدر حجم آن بیشتر باشد، زمان بیشتری را خواهد برد.

فقط یک کار را انجام دهید

شبیه به اصل تک مسئولیتی کلاس‌ها، یک PR نیز باید تنها یک کار را انجام دهد و بر روی یک موضوع خاص تمرکز داشته باشد. فرض کنید PR ایی را ارسال کرده‌اید که سه مشکل A، B و C را برطرف می‌کند. از دیدگاه مسئول اصلی پروژه، موارد A و C قابل قبول هستند؛ اما نه مورد C مطرح شده. در این حالت کل PR شما برگشت خواهد خورد. به همین جهت بهتر است بجای یک PR، سه PR مختلف و مجزا را جهت رفع مشکلات A، B و C ارسال کنید.

سازگاری تغییرات ارسالی را بررسی کنید

حداقل کاری را که پیش از ارسال PR باید انجام دهید این است که بررسی کنید آیا این تغییرات قابل Build هستند یا خیر. همچنین اگر پروژه دارای یک سری Unit tests است، حتما آن‌ها را یکبار بررسی کنید تا مطمئن شوید جای دیگری را به هم نریخته‌اید. ضمنا وجود این تست‌ها به صورت ضمنی به این معنا است که تغییرات جدید شما نیز باید به همراه تست‌های مرتبطی باشند تا پذیرفته شوند.

PR ایی را بر روی شاخه‌ی master ارسال نکنید

پس از اینکه یک fork از پروژه‌ای سورس باز را ایجاد کردید و سپس آن‌را clone نمودید تا به صورت Local بتوانید با آن کار کنید، فراموش نکنید که در همینجا باید یک branch و انشعاب جدید را جهت کار بر روی ویژگی مدنظر خود ایجاد کنید (برای مثال feature-X, fix-Y). بسیاری از پروژه‌های سورس باز به هیچ عنوان PRهای کار شده‌ی بر روی انشعاب master را قبول نمی‌کنند.

برای مطالعه بیشتر

Open Source Contribution Etiquette

ten tips for better Pull Requests

Getting a Pull Request Accepted

Optimize Your Pull-request

نظرات خوانندگان

نویسنده: جلال
تاریخ: ۱۳۹۳/۱۲/۱۴ ۱۹:۱۱

کاش زودتر خونده بودمش :P