Dear Mr. Kovalenko,

We’d like to start by saying we’re excited. As software developers, as FOSS enthusiasts, as students approaching a business environment, and as daily users of your products, we are excited about applying for these two projects. For the past few days we’ve been working on this team task that started out as another deadline to meet and became a challenging, and honestly quite fun, dive into an open source project that we unanimously agree is remarkable.

As we are applying to two different projects: “Tracking Changes in Links to Code”, and “Detecting Refactorings in Code Changes”, we will explain our vision for each.

***Tracking Changes in Links to Code***

We believe that software’s purpose should be to make people’s lives easier, by automating mundane tasks and giving people more time to focus on the activities that matter. An IDE should do the same for software developers. Documentation and tracking of a software project is vital for healthy teamwork and quality software production, but the maintenance of it is time that could be better spent. Following the development and growth of a software project is a challenging enough task in itself. Not having to worry about keeping track of how the documentation refers to an ever evolving codebase, we believe, would help achieve a smoother workflow, while keeping the benefits of a clear documentation.

***Detecting Refactorings in Links to Code***

Git is a powerful tool. An *extremely* powerful tool. This said, it is definitely not as easy to use as it is powerful. We think it’s important that a tool as powerful as this should be accessible by as many developers as possible, regardless of their level of experience. This is why Git management tools are important, especially if integrated into an IDE.

As stated before, teamworking on a large software project is no doubt a challenging task. It oftentime requires a developer to refactor their code to adapt to the evolving structure of the project. It happened multiple time to us while working on the project we’re submitting to you now. This constant state of evolution makes reviewing code changes, in simple words: a pain in the ass. But it doesn’t have to be. It is one of those tasks that can definitely be made less time consuming (and boring) by a well designed tool, such as RefactoringMiner. Given its usefulness, it is in everyone’s best interest that RefactoringMiner be available for as many programming languages as possible, and with an interface as intuitive as possible.

Wrapping up: as software developers we benefit by a wide range of software tools that make our everyday workflow easier. Without all of those, we would probably still be trying to fix NullPointerExceptions in our first Hello World app. We would love to, and we would be honored to give back to the development community that provided us with all these tools, by creating useful ones ourselves. We’d be honored to make something By Developers and For Developers.

We hope you will appreciate our demo project, we put all our love into it. Cheers 😊