Querying the History and Evolution of a Software Project.

Developers spend most of their time understanding how a software system works. In order to do this they need answers to a wide variety of questions. There have been extensive studies concerning the nature of these questions.

Using these studies, we identified three different categories of questions. The first one contains simple questions that can be answered using a proper IDE. An example of such a question is finding out the places where a method is called by the rest of the system.

The second category contains more complex questions, which can be answered using program query tools. Program query tools allow developers to describe the behaviour or characteristics the source code has to exhibit, and identifies source code that corresponds to the specified query. An example of such a question is finding out whether there is code that accesses the database outside of the persistence infrastructure.

The final category contains questions regarding the history and evolution of a software project. For example finding out how often a method was changed can only be answered by consulting the history of the project. A more complex example is finding who added an if-test in a method, and for what reason it was added. These sorts of questions cannot be easily answered using existing tools.

We propose a new program query tool that allows developers to answer these kinds of questions. This tool needs access to the history of a project, and provide this information to the developer. The history of a software project can be found in a version repository, which nowadays are an industry best practice.

Developers need to be able to specify the following characteristics:
1) The characteristics of the source code in one version.
2) The evolution of the source code across different versions.
3) The characteristics of a version in which the source has to be found.
4) The temporal relations between versions in which the evolution of the source code has to be found.

There are several challenges for such a query tool. We list some of them:

Code analyses are computation intensive, and doing these for each version will not scale. There is need for an incremental analysis that reuses information from a previous version.

A complete representation of each version consumes too much memory. The number of changed entities across versions is limited, allowing

reusing these entities in different versions.

In this presentation we show the current state of this tool. We explain the workings by several illustrative examples.