Maintaining Coherent Architectural Models with Sonargraph

Alexander von Zitzewitz
hello2morrow GmbH

Most nontrivial software systems suffer from significant levels of technical and architectural debt. This leads to exponentially increasing cost of change, which is not sustainable for a longer period of time. The single best thing you can do to counter this problem is to carefully manage and control the dependencies among the different elements and components of a software system. This is only possible with the support of tools,

Here we will look at the way Sonargraph solves this problem. The tool comes with a domain specific language to describe architectural blueprints. This is a very powerful and scalable approach that even works for very large projects. Session participants will learn the basics of Sonargraph's DSL which will enable them to create their own models with very little effort. Once a model has been created it can be automatically enforced in your CI build or directly in your IDE using IDE plugins.

Moreover Sonargraph is very strong in visualizing the structure of any software system written in Java, C# or C/C++. The new session view allows you to take an existing legacy code base and transform it into a well architected system by organizing the code into a tree structure of subsystems without actually touching the code. All transformations are simulated and will result in a list of actions to be performed in your IDE.

Sonargraph also computes hundreds of software metrics which can be used to identify issues like overly complex code or high coupling. A Groovy based scripting engine allows you to add your own metrics and code checkers.