HELLO2MORROW

SONARJ

Brief Product Description

__SonarJ__ is a development tool for the efficient monitoring and improvement of the architecture and technical quality of medium-size and large Java projects. At the 2005 'Systems' trade fair, it was awarded a prize for the "most exciting innovation". The popular Spring Framework has adopted __SonarJ__ globally to assure a high level of technical quality and a clean software structure.

Which problems does SonarJ address?

During almost all major software projects, an increasing gap develops in the course of development work between the structure (architecture) that was originally planned and the structure realized in the source code. Although this gap is undesirable, it is extremely difficult to avoid. It is caused by the introduction of unwanted dependencies between different elements of the software. The outcome of these is that understanding and testing the system become increasingly difficult. Why are these unwanted dependencies so difficult to avoid? The reason is that they are not normally identified in the early stages, and exert their damaging effects only later – but in an all the more irritating way. The term used for this is "erosion" of the architecture, and its symptoms are only too well known to most people with IT responsibilities: delays occur in completing the software, and its cost rises well above the original budget. Worse still: the quality is still not up to the desired standard, and every attempt to extend the software is so expensive that the developers begin to suggest that a "redesign" would be preferable. In other words, the software is discarded and rewritten simply because this proves to be cheaper than eliminating the shortcomings from the existing software – a nightmare for anyone who has to answer for the costs incurred!

Other problem-areas are proving the quality of software from outside providers or ongoing checks on the technical quality of an internal project.When this work is commissioned, it is often limited to functions and technical requirements, whereas the internal technical quality of the software is ignored. In many cases, the software that is delivered satisfies the functional and technical formalities but proves very difficult to extend or modify, because of internal structural shortcomings. The client ends up paying a high price for these, and therefore has a fundamental interest in defining verifiable technical quality criteria for delivered software. This was the difficulty until now: establishing measurable criteria for technical quality and internal structures.

How has SonarJ solved these problems?

SonarJ allows software architects to define the structure of software that is being planned or extended at a very high level of abstraction, and to develop it further at this level. Since the abstraction level is so high, the structure becomes easy to understand and communicate. The defined structure is distributed to the development team as an XML file. SonarJ is installed at each development station, either as an Eclipse plug-in or as an independent product. SonarJ reads in the architecture definition and from then on checks every code change that is made for conformity with the planned structure. The developer is in fact unaware of the checking process, which takes place in the background at a high level of efficiency. However, if the specified structure is infringed, a warning is displayed immediately to assist the developer in eliminating the problem on the spot. The product also supplies a considerable number of additional aids to the developer; they make it easier to understand the software and the planned architecture, familiarize new employees with the necessary procedures more quickly and improve overall productivity. The combination of higher-level architecture definition and direct verification during the development process is currently unique on the market.

A further application for the product is monitoring the technical quality of software. SonarJ possesses powerful static code analysis capability and is therefore in a position to make internal structures visible. By the evaluation of certain core metrics computed by SonarJ, an initial verdict on the technical quality of the software under investigation can be obtained in a very short time (less than 2 hours). The client can now include verifiable technical quality criteria in tender invitations and order specifications, with the result that suppliers will no longer be able to rely on the acceptance of functionally correct but structurally defective systems.

What effect does SonarJ have on costs?

The "Software Technology Support Center" (STSC), which has close links with the Pentagon in the USA, has investigated the extent to which a satisfactory or defective inner structure influences the cost of further software development. The results clearly confirm what one has always instinctively felt to be true: structural errors can prove extremely costly! A system with 50,000 lines of code was investigated; the aim was to extend it by the addition of about 3,000 lines of code. Two teams were formed: team 1 was required to work on the basis of the mature system structure, which was in no way optimal, since it already exhibited typical symptoms of eroded architecture. Team 2 was supplied with a restructured system as a basis, after elimination of the previous structural defects.

Not surprisingly, team 1 needed over twice as long as team 2 to complete the task. And what is even more alarming: the results from team 1 contained more than eight times the number of errors that the work submitted by team 2.

No more studies of this kind seem to have been published as yet, but most experts in the software development sector are surely agreed that these results do not come as a surprise.

Let us assume that five people work for a year on a medium-size Java project. If we make a conservative forecast that SonarJ will make them only 30% percent more efficient and that only half as many errors will be produced (remembering that the STSC study found efficiency to be more than doubled and the number of errors to be eight times lower), we still arrive at a saving of 1.5 developer years as well as a result of higher technical quality. Without making any financial debit for the cost of additional errors, the saving already amounts to at least 150,000 euros. SonarJ licenses for the team cost under 10 percent of the sum saved in this way. The investment in SonarJ can therefore be regarded as commercially viable.

Are you interested?

We would welcome an opportunity to demonstrate our technology by applying it to one of your own projects. For further information, technical white papers and a chance to test the product free of charge, please visit our website:

www.hello2morrow.de (German site)
www.hello2morrow.com (English site)

"I was amazed to see how quick and easy we were able to adopt SonarJ for managing the architecture and technical quality of the Spring Framework family. For the Spring Web Flow project SonarJ helped us to almost cut in half the internal coupling of the code base with very little additional effort. Now we are using SonarJ in our daily development work and it helps us to keep the architecture and quality of Spring on the high level expected by our global user base."

Juergen Hoeller – VP & Distinguished Engineer, Interface21

hello2morrow GmbH · Alexander v. Zitzewitz · Maxstraße 9 · D-82335 Berg · Germany
Phone +49-89-25 55 71 44 · Fax +49-89-25 55 71 45 · Mobile +49-171-476 76 51
E-Mail: a.zitzewitz@hello2morrow.com · Skype: a.zitzewitz · www.hello2morrow.com