



Interview with Olivier Gaudin

“Agile is Really Part of SonarSource’s DNA”

We talk to SonarSource’s CEO about Continuous Inspection, technical debt, and Sonar, an open platform to manage Code quality.

Java Tech Journal: Continuous Integration has been around for more than a decade. But what exactly is Continuous Inspection, in a nutshell, and how is it related to the former?

Olivier Gaudin: Continuous Inspection is the ability to continuously manage the technical debt of an application, to maximize its level of maintainability. Doing Continuous Inspection will support the resolution of issues such as “any change or addition to source code should be covered by unit tests”, “code should not be duplicated” or “no new cycle dependencies should be introduced”... and thus at any point in time.

Although they focus on two different areas of the Application Lifecycle Management, the two concepts share a common objective, which is to deliver quickly high quality software increments and ultimately to be in a position to do Continuous Delivery.

JTJ: How and when did the practice and the term originate and why?

Gaudin: Agile is really part of SonarSource’s DNA, and we are convinced that feedback loops must be as short as possible. In the code quality management world, before Sonar, all products supported only a traditional approach to code quality which we call Quality Gate. Quality Gate consists of running a formal inspection at the end of the development cycle and using the results of this inspection to decide whether to ship or not to production.

We believe this approach will not have a significant impact on quality as it comes (too long) after the fact and is often initiated by an external entity. And real life actually shows that it does not work: Very often and regardless of the inspection results, the application will ship to production “as is” because the immediate loss of not shipping looks more threatening than the medium-term savings look appealing.

This is why we have built with Sonar a tool that can provide feedback on code quality to the developer in a continuous way, i. e. as soon as a quality defect is introduced. The development team therefore has the opportunity to take an

action when the change that introduced the issue is still fresh in the minds. This is a change of paradigm and this is why we introduced this new term.

JTJ: The practice (Continuous Inspection) is strongly associated with Sonar; Sonar in turn is maintained by SonarSource, the company you co-founded. How did you get the idea of starting a business around the Sonar platform?

Gaudin: We started SonarSource four years ago because we found that none of the tools in the market at the time responded well to the needs of development teams and more specifically to agile teams. Also, we felt that the maturity of development processes had increased strongly and that code quality management was going to be one of the next big things organisations would want to look at if the right product is available.

JTJ: In a blog post of yours [1], you refer to Martin Fowler’s concept of “technical debt” [2]. Can you explain how this metaphor relates to the problems addressed by Sonar?

Gaudin: The concept of technical debt consists of comparing any shortcut you take while developing to a financial debt

Portrait

Olivier Gaudin is co-founder and CEO at SonarSource, the company that develops and promotes the open source platform Sonar to manage source code quality. Olivier started his career in London, working on trading systems in the banking industry. He then managed IT for a startup, ran big projects such as off-shoring, CMMi and ran support and development departments. Olivier has developed a strong sensibility to robustness of processes and quality of delivery. In 2007, Olivier started contributing to Sonar and decided with Simon Brandhof and Freddy Mallet to create SonarSource to accelerate development and adoption of the platform.

you would contract with a bank. The technical debt represents the sum of quality flaws of an application. If you are not careful, you will continue to borrow at each iteration and then increase your debt. The more debt in your application, the more difficult and costly it is to make changes to it. If you do not start managing the debt, you will eventually get bankrupt, which means a necessary rewrite of the application. Sonar enables to manage the part of this technical debt, which relates to source code.

JTJ: In that essay, Fowler states that technical debt cannot be measured because a team's productivity cannot be measured. What's your take on this?

Gaudin: I don't think this is what Martin Fowler meant! As I explained earlier, technical debt is strongly correlated to the level of maintainability of an application. Therefore, if you pay back some of the technical debt, you expect that the level of maintainability of the application increases. Measuring the cost for reimbursing the debt is possible in most cases (although not always, for example measuring the quality of an architecture for example is not really possible). What we *cannot* measure is the benefit you get by paying back the debt, i. e. answer questions such as: How much time am I going to save now that I do not have this quality issue? How much more productive is my team going to be?

Indeed, as stated by Martin Fowler, we cannot measure productivity of a team because too many variables are involved, including very subjective ones.

JTJ: As you put it, Sonar helps resist the temptation of the "seven deadly sins" of developers: bad distribution of complexity, duplications, lack of comments, coding rules violations, potential bugs, no unit tests or useless ones and bad design. Can you give us a few examples of how these things are addressed technically?

Gaudin: In terms of approach, very little has changed since the first steps of Thomas McCabe in the field of code quality in the 70's. We are still using the same techniques of lexing, parsing, building tables of symbols, walking ASTs, and using trees of dependencies. However, although we started Sonar by integrating existing open source modules, we have, for the last two years, strongly invested in those techniques to be able to tackle any new language with reduced effort of development and maintenance. For example, we have developed our own engine of Copy-Paste Detection (CPD) last year, enabling to now offer to Sonar users the ability to detect Copy and Paste inside projects, but also across projects to detect them on their entire portfolio. We have also developed our grammar for JavaScript, Flex, Python, Cobol, PL/SQL, C, C++, C#, and so forth.

JTJ: What else in the pipeline for Sonar – what's on the road-map?

Gaudin: We have many ideas and projects to build on the existing platform, such as adding new programming languages (PL/I, VB.NET, etc.) or providing extra SCM information to add the frequency of changes on a piece of code. But our next big project is called Cartography and consists of adding a new dimension to the Sonar platform with intra- and inter-technology dependency management.

Literature & Additional Resources

- [1] <http://www.sonarsource.org/evaluate-your-technical-debt-with-sonar>
 [2] <http://martinfowler.com/bliki/TechnicalDebt.html>

Imprint

Publisher
Software & Support Media GmbH

Editorial Office Address
Darmstädter Landstraße 108
60598 Frankfurt am Main
Germany
www.jaxenter.com

Editor in Chief: Sebastian Meyen
Editors: Chris Mayer, Claudia Fröhling, Diana Kupfer
Authors: Scott Mattocks, Roman Pichler, Jessica Thornsby
Copy Editor: Nicole Bechtel, Lisa Pychlau
Creative Director: Jens Mainz
Layout: Petra Rüh

Sales:
Ellen May
+44 (0)20 7401 4837
ellenm@sandsmedia.com

Entire contents copyright © 2012 Software & Support Media GmbH. All rights reserved. No part of this publication may be reproduced, redistributed, posted online, or reused by any means in any form, including print, electronic, photocopy, internal network, Web or any other method, without prior written permission of Software & Support Media GmbH

The views expressed are solely those of the authors and do not reflect the views or position of their firm, any of their clients, or Publisher. Regarding the information, Publisher disclaims all warranties as to the accuracy, completeness, or adequacy of any information, and is not responsible for any errors, omissions, inadequacies, misuse, or the consequences of using any information provided by Publisher. Rights of disposal of rewarded articles belong to Publisher. All mentioned trademarks and service marks are copyrighted by their respective owners.