How the Umweltbundesamt (Environment Agency Austria) Achieved Sustainable Software Development

About Environment Agency Austria (EAA):

With more than 450 members of staff, the Environment Agency Austria is the largest environmental expert organization in Austria. It is a leading provider of environmental solutions. As independent partner, the Environment Agency Austria builds bridges between business, science and politics at national and international level. As leading IT provider in the environment sector, the Environment Agency Austria develops eGovernment-conforming environmental information systems, databases and applications.

The experience of more than 25 years makes the Environment Agency Austria one of the leading providers of environmental consulting services to national and international clients. Apart from big enterprises, clients and partners are scientific and political institutions in Austria, the EU and beyond. EAA performs evaluations, sets standards and develops methods and recommendations. EAA provides the basics for decision-making and creates ecological and economic added value for customers. EAA is a member of more than 200 national, European and international partner networks, bodies and working groups. The Environment Agency Austria collaborates with international organizations such as UNEP and the OECD.

EAA is striving to improve the state of the environment and to preserve livelihoods and natural resources. This is how EAA contributes to sustainable development, by providing a vision for society and aiming to improve the quality of life of both current and future generations.

Environment Agency Austria IT Services and Solutions offers a wide range of information technology services, from the development of complete information systems to maintenance, operation, and web hosting, as well as a help desk with user support and user training. Experts cover the entire software development cycle including system design, implementation, testing, operation, training, help desk and documentation.

The Environment Agency Austria is the key service provider for EDM, Electronic Data Management for environment and waste management. It is an efficient, user-friendly platform for administrative processes. EAA provides analysis, software development and operation.

About hello2morrow

hello2morrow is a leading provider of static analysis tools for the validation and enforcement of rules related to architecture, structure and technical quality of software systems. Our products support Java, C#, C/C++ and other languages and are used by more than 300 customers all over the world.

The Challenges:

In 2005 EAA had 5 Java applications and 3 Java components partially shared between them. At that time, it was already clear that maintaining those applications and components was tedious and expensive because of the accumulation of technical debt and problems with

code quality. IT Services and Solutions therefore initiated policies that would put the achievement of code quality standards at the center of all development work.

The following challenges had to be met:

- Developing sustainable applications, software and modules for multiple users and usage scenarios
- Avoiding structural and technical debt when starting projects from scratch
- Dealing with structural erosion and technical debt when working with legacy applications
- External software development service providers would have to work with the same standards
- Growing demand
- A growing team

The Solution:

First, the acceptable code quality standards had to be defined. It became obvious that it would only make sense to enforce standards that can be checked efficiently, ideally in a fully automated way. It was also considered important that developers would not be overwhelmed with too many rules and guidelines. The goal was to improve the quality of project outcomes without slowing down the development process in a significant way.
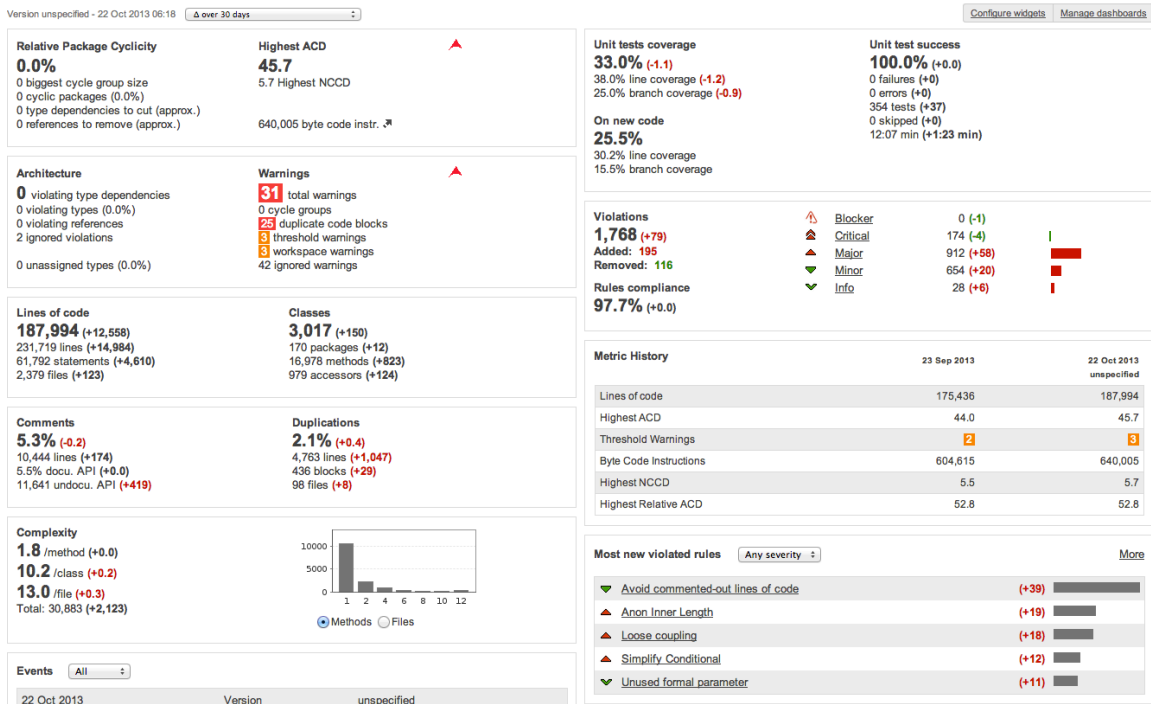
 The standard had to cover the following aspects:

- Coding and formatting rules
- Testing and test coverage
- Software metrics
- Architecture rules and dependency structure

Each of those aspects needed tool support to ensure the automated enforcement of rules. The first two aspects could be covered easily with the help of Open Source tools like PMD, CheckStyle, FindBugs and Cobertura. For the last two aspects there was no viable open source option, so EAA looked for commercial tools to cover the gap. Finding a solution for the architectural aspect was considered to be especially critical, because, in most cases, it is prohibitively expensive to repair an application that is structurally broken.

In early 2007 the decision was made to purchase SonarJ (now SonarGraph), from hello2morrow, to support the automated enforcement of metric and architecture based rules. In 2008 all tools were integrated under the open source umbrella tool "Sonar" (now "SonarQube"), which would combine the findings of all the different tools in a unified quality dashboard (see screenshot below).

While the SonarQube collects data daily during the nightly build, Hudson (now often replaced by "Jenkins") is used to check coding and architecture rules with every commit. Rule violations will break the build. Developers will receive an email notification from the build server so that the problem can be fixed while it is still easy to fix.

Relative Package Cyclicity
0.0%
0 biggest cycle group size
0 cyclic packages (0.0%)
0 type dependencies to cut (approx.)
0 references to remove (approx.)

Highest ACD
45.7
5.7 Highest NCCD

640,005 byte code instr.

Architecture
0 violating type dependencies
0 violating types (0.0%)
0 violating references
2 ignored violations

0 unassigned types (0.0%)

Warnings
31 total warnings
0 cycle groups
25 duplicate code blocks
3 threshold warnings
3 workspace warnings
42 ignored warnings

Lines of code
187,994 (+12,558)
231,719 lines (+14,984)
61,792 statements (+4,610)
2,379 files (+123)

Classes
3,017 (+150)
170 packages (+12)
16,978 methods (+823)
979 accessors (+124)

Comments
5.3% (-0.2)
10,444 lines (+174)
5.5% docu. API (+0.0)
11,641 undocu. API (+419)

Duplications
2.1% (+0.4)
4,763 lines (+1,047)
436 blocks (+29)
98 files (+8)

Complexity
1.8 /method (+0.0)
10.2 /class (+0.2)
13.0 /file (+0.3)
Total: 30,883 (+2,123)

Methods / Files

Events | All

22 Oct 2013 | Version | unspecified

Unit tests coverage
33.0% (-1.1)
38.0% line coverage (-1.2)
25.0% branch coverage (-0.9)

On new code
25.5%
30.2% line coverage
15.5% branch coverage

Unit test success
100.0% (+0.0)
0 failures (+0)
0 errors (+0)
354 tests (+37)
0 skipped (+0)
12:07 min (+1:23 min)

Violations
1,768 (+79)
Added: 195
Removed: 116

Blocker	0	(-1)
Critical	174	(-4)
Major	912	(+58)
Minor	654	(+20)
Info	28	(+6)

Rules compliance
97.7% (+0.0)

Metric History

	23 Sep 2013	22 Oct 2013 unspecified
Lines of code	175,436	187,994
Highest ACD	44.0	45.7
Threshold Warnings	2	3
Byte Code Instructions	604,615	640,005
Highest NCCD	5.5	5.7
Highest Relative ACD	52.8	52.8

Most new violated rules | Any severity | More

Avoid commented-out lines of code	(+39)
Anon Inner Length	(+19)
Loose coupling	(+18)
Simplify Conditional	(+12)
Unused formal parameter	(+11)

Typical SonarCube Dashboard featuring findings from Sonargraph and other tools

Here are the code quality standards used by EAA to enforce a high level of code quality:

- No architecture violations (Sonargraph)
- No Package cycles, layer cycles (Sonargraph)
- Test coverage (JUnit): at least 80% (business logic)
- Documentation (JavaDoc): at least 30% (95% for public API)
- Duplicate Code: at most 4%
- No Blocker and critical rule violations (PMD, FindBugs, Checkstyle)

Once the technical infrastructure was put in place to check and enforce rules, projects were reviewed frequently for compliance. This usually took place once every two weeks, and was done by the Quality Assurance Manager, a role that was created to ensure the enforcement of the above standards. The review process is driven by the quality data derived from the SonarQube dashboard. If standards are not met the issue is escalated to the project manager. Additionally, any application that fails to meet the quality standard will not go into production.

An important aspect of successfully implementing software development based on quality standards is to make sure that all stakeholders are on board. This starts with the selection of capable developers and architects who are willing to follow the established rules, and who understand the benefits of a development approach centered on quality. Business stakeholders must understand that while meeting quality standards can initially slow down project progress, it will pay off nicely in the medium and long term. Moreover, there must be a general acceptance that sacrificing quality in order to pack in more features, or to meet an overly ambitious deadline, is usually not a very good strategy. Of course, sometimes situations will occur that make such a sacrifice necessary for reason outside of the technical

scope of the project. In that case it is important to plan for a repair phase after the initial goal has been met. A clean foundation and stable structure also makes repairs easier.

The Results:

All stakeholders consider the introduction of a quality centered development process to be great success.

In 2013, EAA has 42 Java applications with 28 components. While this represents an expansion of 8 times as many as in 2005, the number of developers has only increased by about 50%. Integrating code quality standards into the development process has increased development team efficiency by a minimum of 30%. The dramatic improvement in code quality has had a significant impact on maintenance cost, reducing them by more than 50%. User satisfaction has increased greatly and, applications are both more flexible, and have a longer potential lifetime than before.

Since a clean software architecture is the precondition for many other aspects of code quality, Sonargraph is considered to be a mission critical part of the tool chain used to enforce quality standards at the EAA.

In addition to he positive outcome from implementing this new strategy other agencies have learned about this success and now entrust the EAA to provide software QA services for some of their projects.

Contacts:

Umweltbundesamt (Environment Agency Austria)

Thomas Baldauf

Head of Java Development

EMail: thomas.baldauf@umweltbundesamt.at
Web: http://www.umweltbundesamt.at/en/

hello2morrow, Inc.

Alexander von Zitzewitz

EMail: a.zitzewitz@hello2morrow.com
Web: http://www.hello2morrow.com