

Echolot

Qualitätssicherung mit Sonar

Thomas Haug

thomas.haug@mathema.de

www.mathema.de

- Motivation
- Sonar Überblick
- Demo
- Fazit

- Motivation
- Sonar Überblick
- Demo
- Fazit

“Sometimes the developers manage to maintain this purity of design through the initial development and into the first release. More often something goes wrong. The software starts to rot like a piece of bad meat”
(Robert C. Martin)

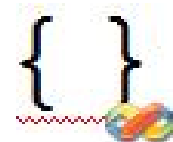


- Architektur-, Design- und Implementierungsvorgaben
 - Reviews

- Architektur-, Design- und Implementierungsvorgaben
 - Reviews
 - Statische und dynamische Analyse



JUnit

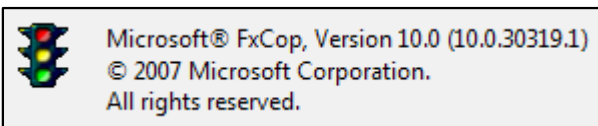


StyleCop

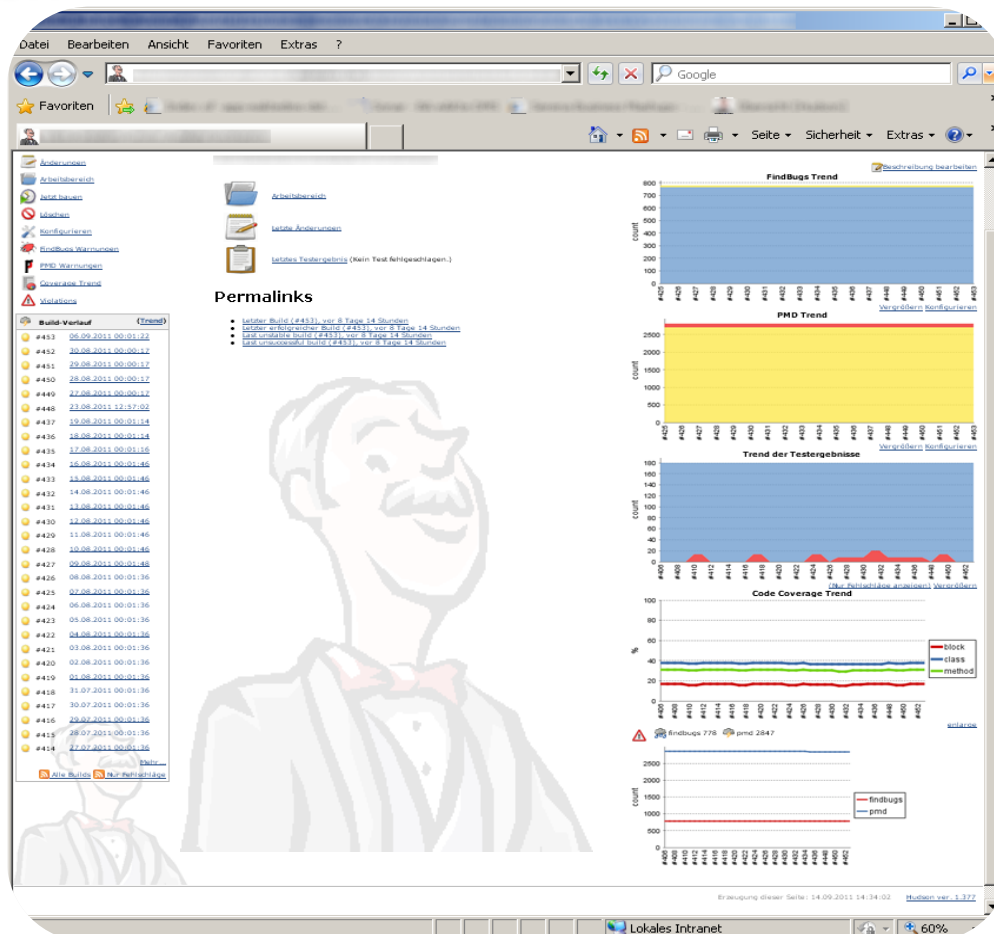


PartCover

OpenCover



- Typische Anwendung: Einsatz einer Vielzahl von Werkzeugen



JUnit



■ Herausforderungen:

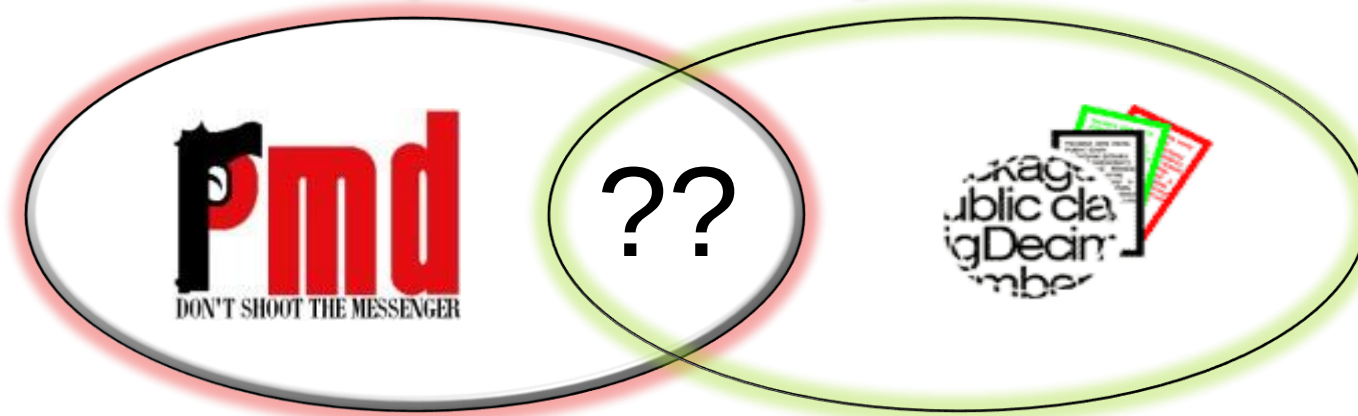
- Know-how und Konfiguration unterschiedlicher Werkzeuge



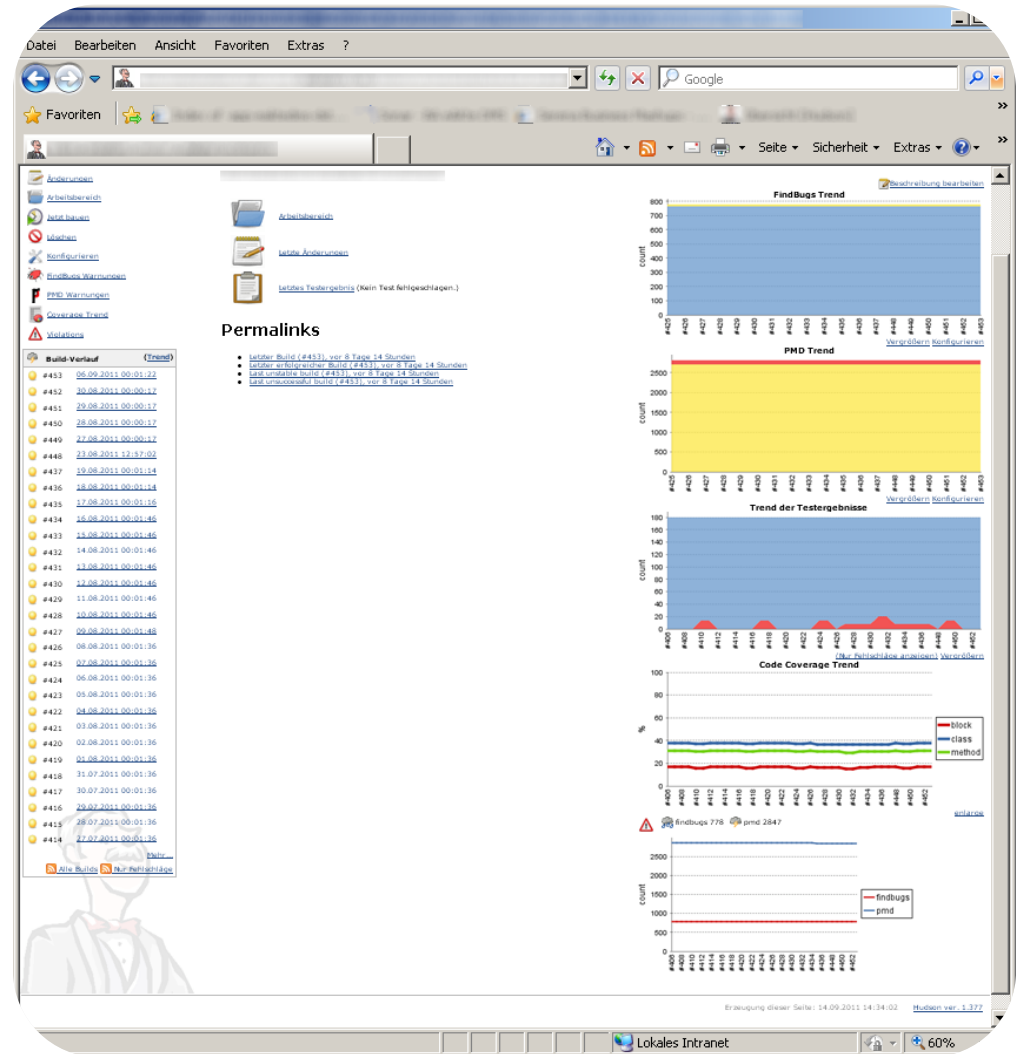
JUnit



- Überschneidungen der Werkzeuge



- Herausforderungen:
 - Einbetten in
(Continuous Integration)
Build
 - Trendanalyse





EMMA

JUnit



OpenCover



PartCover

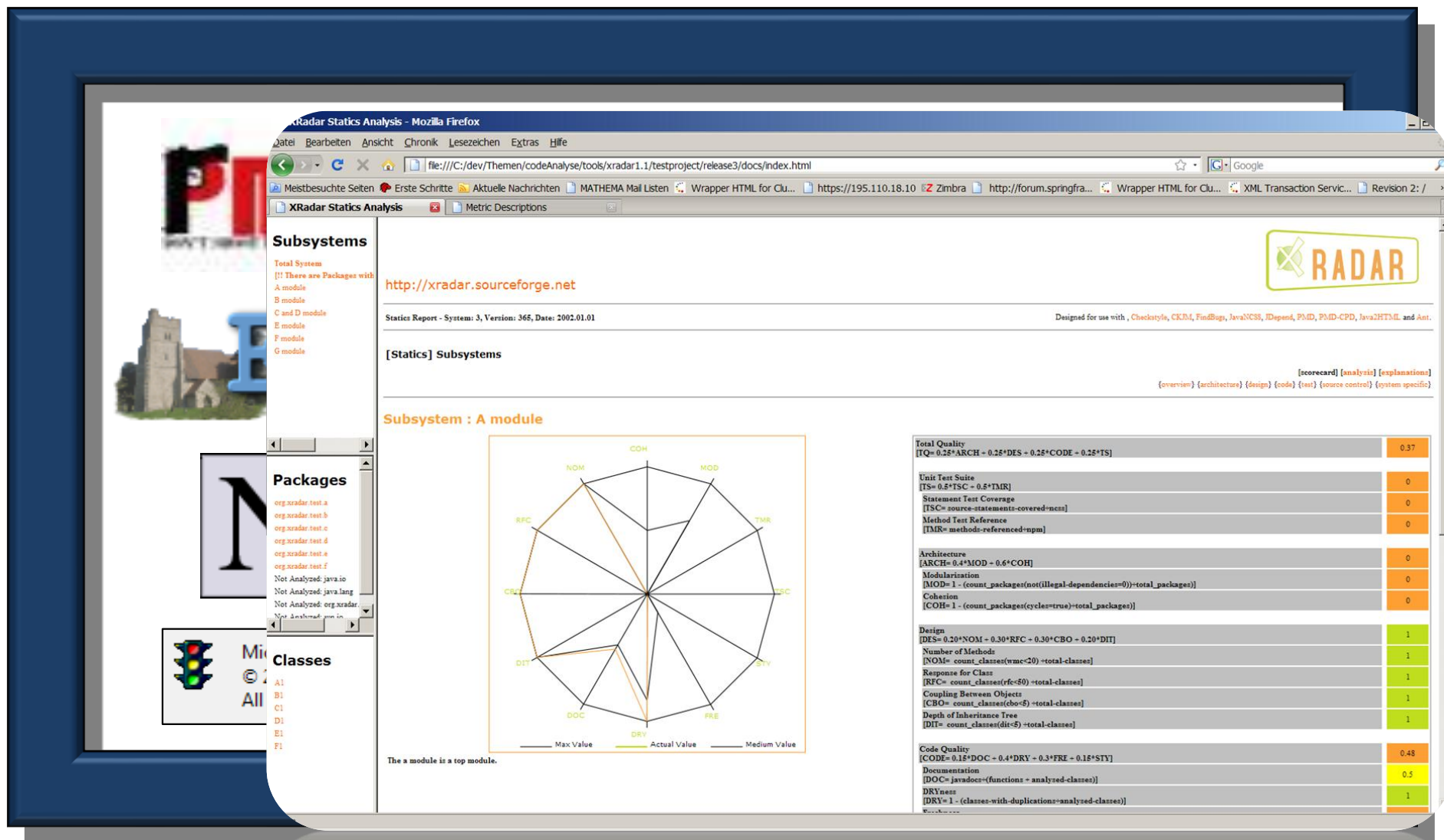


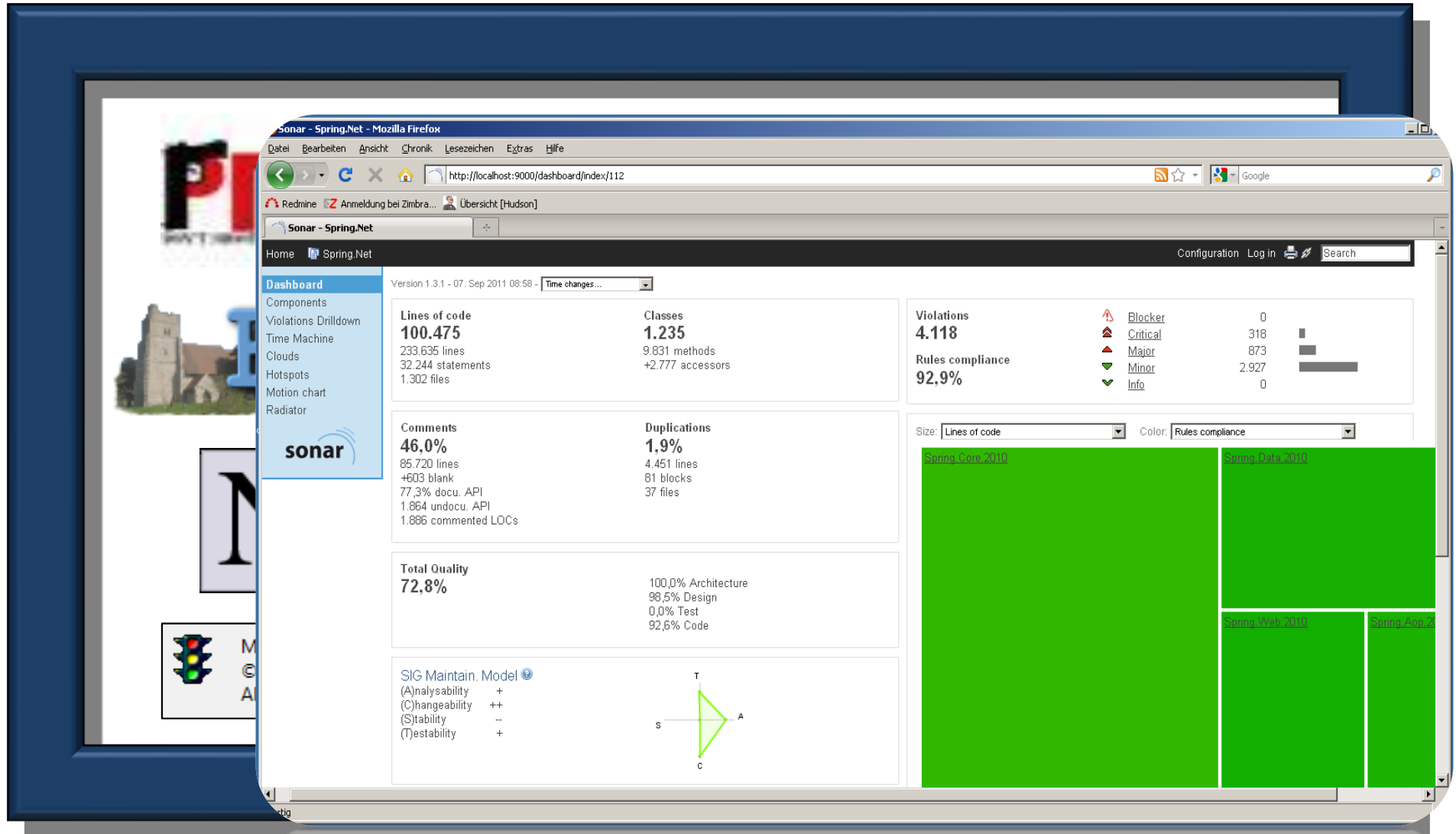
NCOVER



Microsoft® FxCop, Version 10.0 (10.0.30319.1)
© 2007 Microsoft Corporation.
All rights reserved.







- Motivation
- Sonar Überblick
- Demo
- Fazit

- www.sonarsource.org
- Beheimatet bei Codehaus
- Aktuelle Version 3.1.1 (Juni 2012)
- LGPL Lizenz-Modell, kommerzieller Support möglich



- Über 100 000 Downloads
- Knapp 20 Releases innerhalb von 2 Jahren
- Über 1000 Subscribers auf Sonar Mailing-Listen
- Erweiterbar über Plugin Mechanismus
 - Es existieren mehr als 30 Plugins



■ Unterstützte Sprachen

■ **Java (Build-in)**

■ C

■ C#

■ Flex

■ Natural

■ PHP

■ PL/SQL

■ Cobol

■ Visual Basic 6

■ XML



- Multi-Projekt fähig
 - Alle Projekte können über das Dashboard erreicht werden
- „Drill-down“ Analyse
 - Abtauchen in die Tiefen des Designs und der Implementierung über Dashboards möglich
- Prüfen von Kodierrichtlinien
 - Über 600 Regeln vorgefertigt (Java)
 - Java : PMD, CheckStyle, FindBugs,...
 - C# : FxCop, StyleCop, Gendarme



- Standard Metriken
 - Sonar misst Metriken wie McCabe, LOC, ...
- Dynamische Analyse
 - Messen von Code-Coverage
 - via Cobertura, Emma, JaCoCo für Java
 - via NCover, PartCover, OpenCover für C#
- Build-Integration
 - Maven
 - Ant, Ad hoc (sonar runner)



■ Plugins Metriken

Additional Metrics

Artifact Size - Reports on the size of the artifact generated by projects.

Build Stability - Reports on stability of project build using Continuous Integration engine data.

Clirr - Checks Java libraries for binary and source compatibility with older releases.

Clover - Get code coverage with [Atlassian Clover](#).

Emma - An alternative to Clover and Cobertura to measure coverage by unit tests in Java.

fb-contrib - Additional rules from fb-contrib project associated to the FindBugs plugin.

GreenPepper - Collects and reports tests results of executable specifications provided by GreenPepper.

JaCoCo - An alternative to Clover and Cobertura to measure coverage by unit tests in Java.

JIRA Issues - Retrieves and reports the number of project issues from JIRA.

JMeter - Retrieve JMeter test results in SONAR.

Mantis - Retrieves and reports the number of project issues from Mantis.

Rules Meter - Gives information on the level of activation of projects quality profiles.

Security Rules - Enables to zoom on security rules violations to keep them under control.

Sonargraph - Provides architecture governance features accompanied by metrics about cyclic dependencies and other structural aspects using Sonargraph.

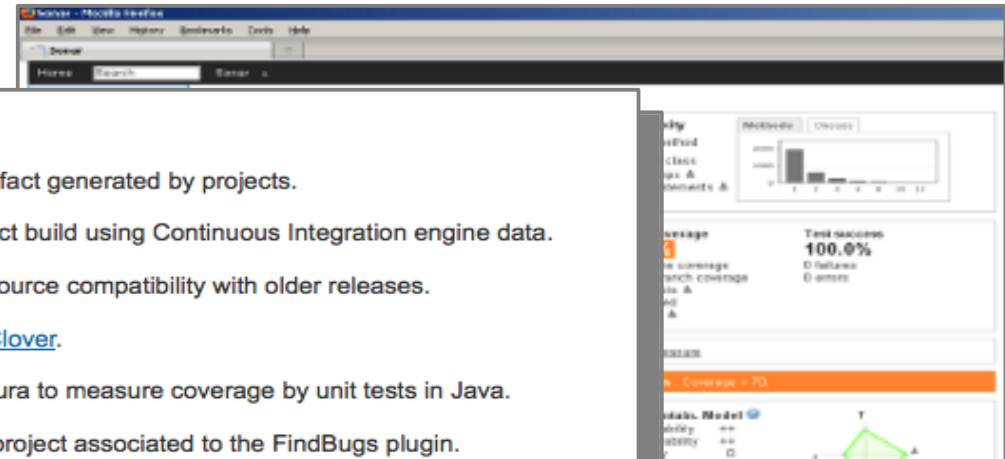
SonarJ - Provides architecture governance features accompanied by metrics about cyclic dependencies and other structural aspects using SonarJ.

Taglist - Generates a report on various tags found in the code, like @todo or //FIXME tags.

Trac - Retrieves and reports the number of project issues from a Trac instance.

Useless Code - Reports on the number of lines that can be reduced in an application.

Violation Density - Compute a new metric named violation density. This is kind of "opposite" for rule compliance metric.



■ Plugins Integration

Integration

[AnthillPro](#) - Enables to configure and launch Sonar analysis from AnthillPro

[Bamboo](#) - Enables to configure and launch Sonar analysis from [Bamboo](#), the Atlassian CI engine.

[Branding](#) - Allows to add your own logo to the Sonar UI.

[Build Breaker](#) - Makes the build fail if pre-defined alert thresholds are hit.

[Crowd](#) - Enables delegation of Sonar authentication to [Atlassian Crowd](#).

[Cutoff](#) - Exclude files from analysis based on a cutoff date threshold, to analyze the work done on an existing code base and measure the quality of new code/changes only.

[Google Calendar](#) - Posts an event, when project analysed by Sonar.

[Hudson / Jenkins](#) - Enables to configure and launch Sonar analysis from [Hudson](#) or [Jenkins](#) CI engines.

[LDAP](#) - Enables the delegation of Sonar authentication to LDAP and Microsoft Active Directory.

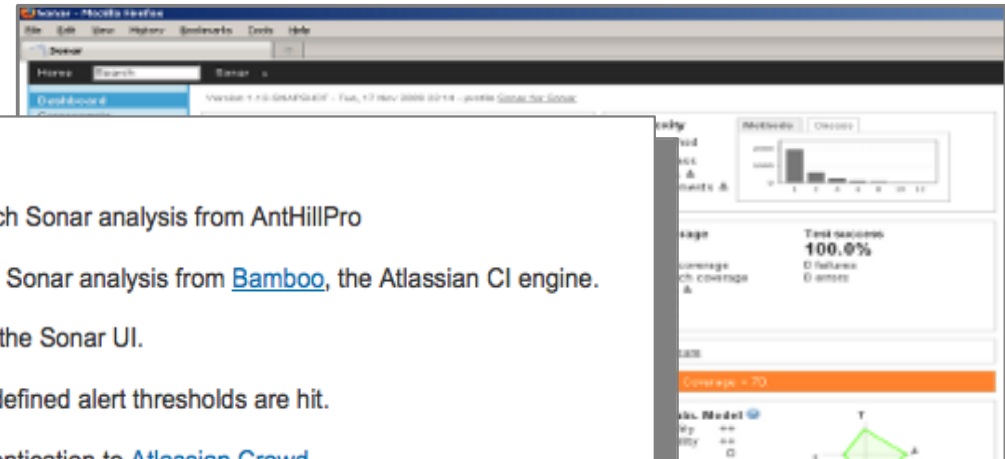
[PAM](#) - Enables the delegation of Sonar authentication to underlying PAM subsystem.

[Piwik](#) - Submits usage of a Sonar instance to a [Piwik](#) server.

[SCM Activity](#) - Collects and reports information on commits using SCM data.

[Switch off violations](#) - Exclude some violations in a fine-grained way.

[Twitter](#) - Creates tweet, when project analysed by Sonar.



■ Plugins Darstellung

Visualization / Reporting

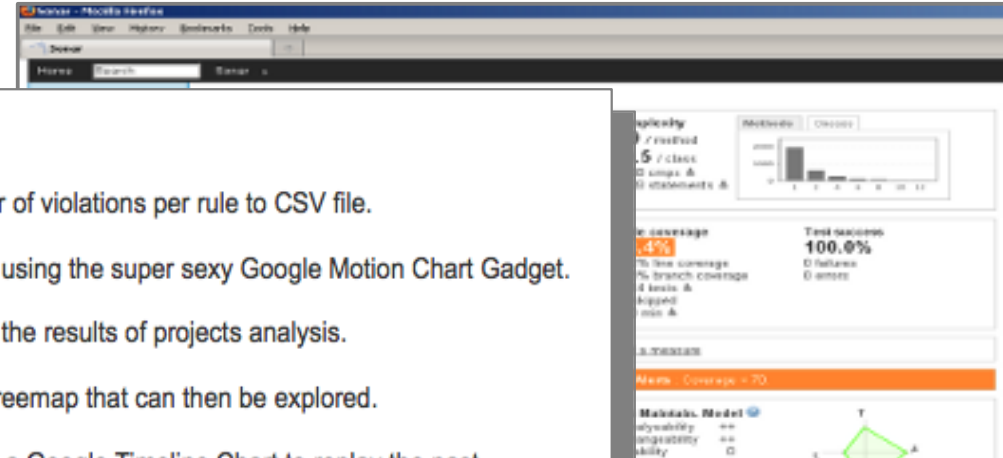
[CSV Export](#) - Export measures and number of violations per rule to CSV file.

[Motion chart](#) - Displays projects measures using the super sexy Google Motion Chart Gadget.

[PDF Report](#) - Generates a PDF report with the results of projects analysis.

[Radiator](#) - Displays measures using a big treemap that can then be explored.

[Timeline](#) - Displays measures history using a Google Timeline Chart to replay the past.



■ Plugins Steuerung

Governance

[Quality Index](#) - Calculates a global Quality Index based on coding rules, Style, Complexity and Coverage by unit tests.

[SIG Maintainability Model](#) - An implementation of the SIG MM to evaluate the maintainability of an application.

[SQALE - Quality Model](#) (Commercial) - An implementation of the SQALE Methodology, which supports the evaluation of a software application's source code in the most objective, accurate, reproducible and automated way possible.

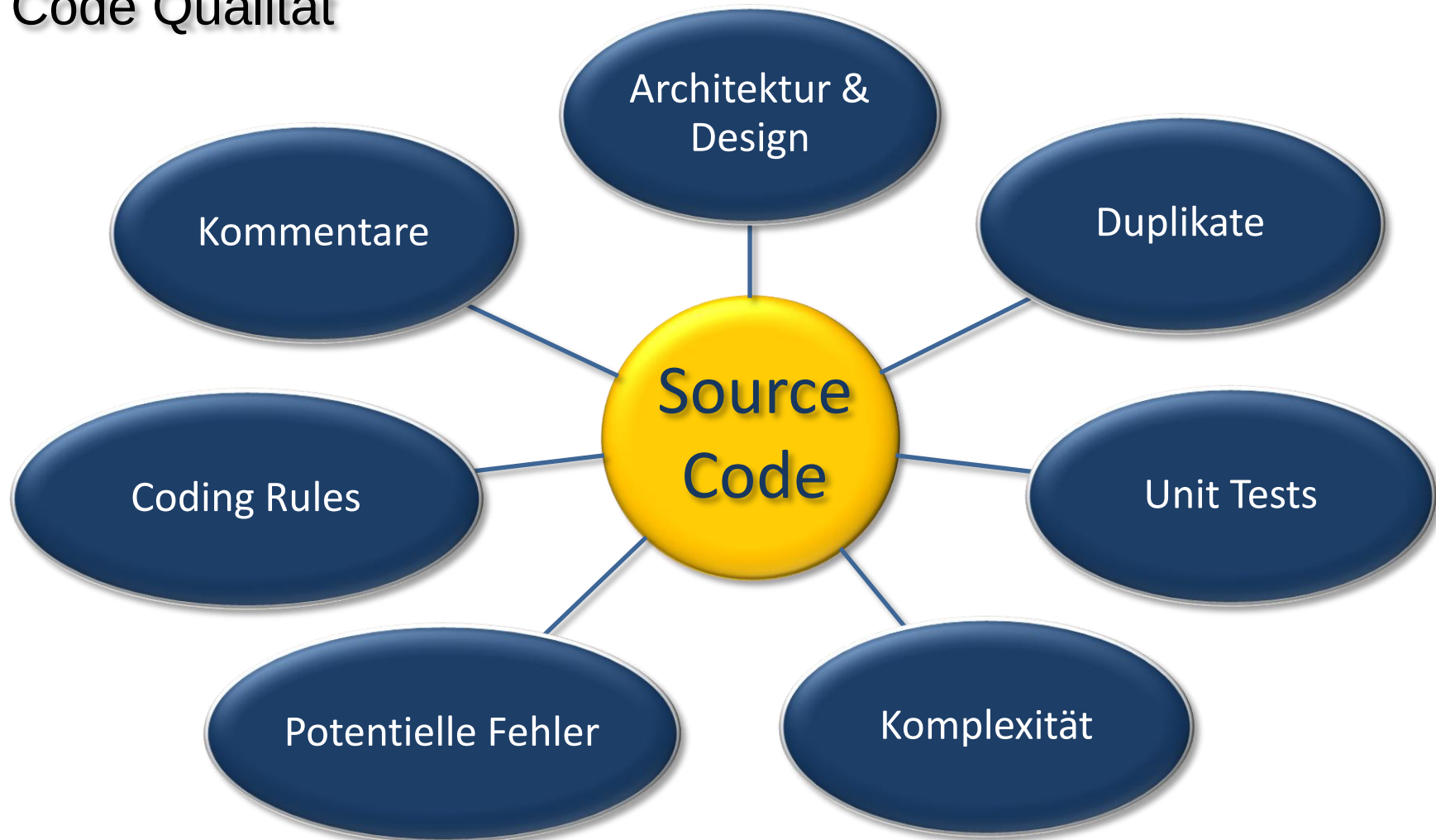
[Technical debt](#) - Calculates the technical debt on every component of projects with a breakdown by duplications, documentation, coverage, complexity...

[Total Quality](#) - Provides an overall measure of the quality of projects linking code quality, design, architecture, and unit testing.

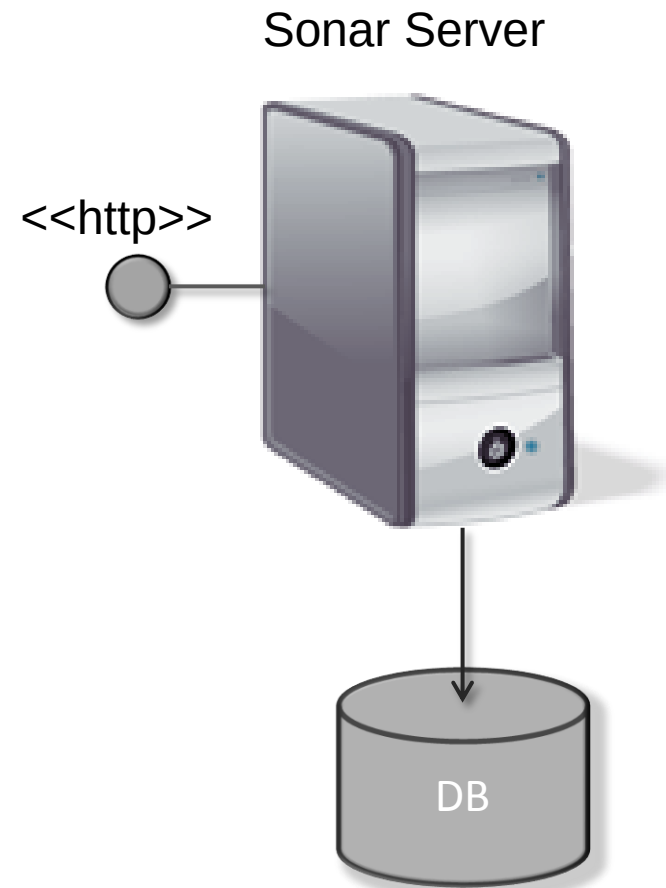
[Views - Portfolio Management](#) (Commercial) - Enables aggregation of projects. Projects can be grouped into applications, applications into teams, teams into departments...



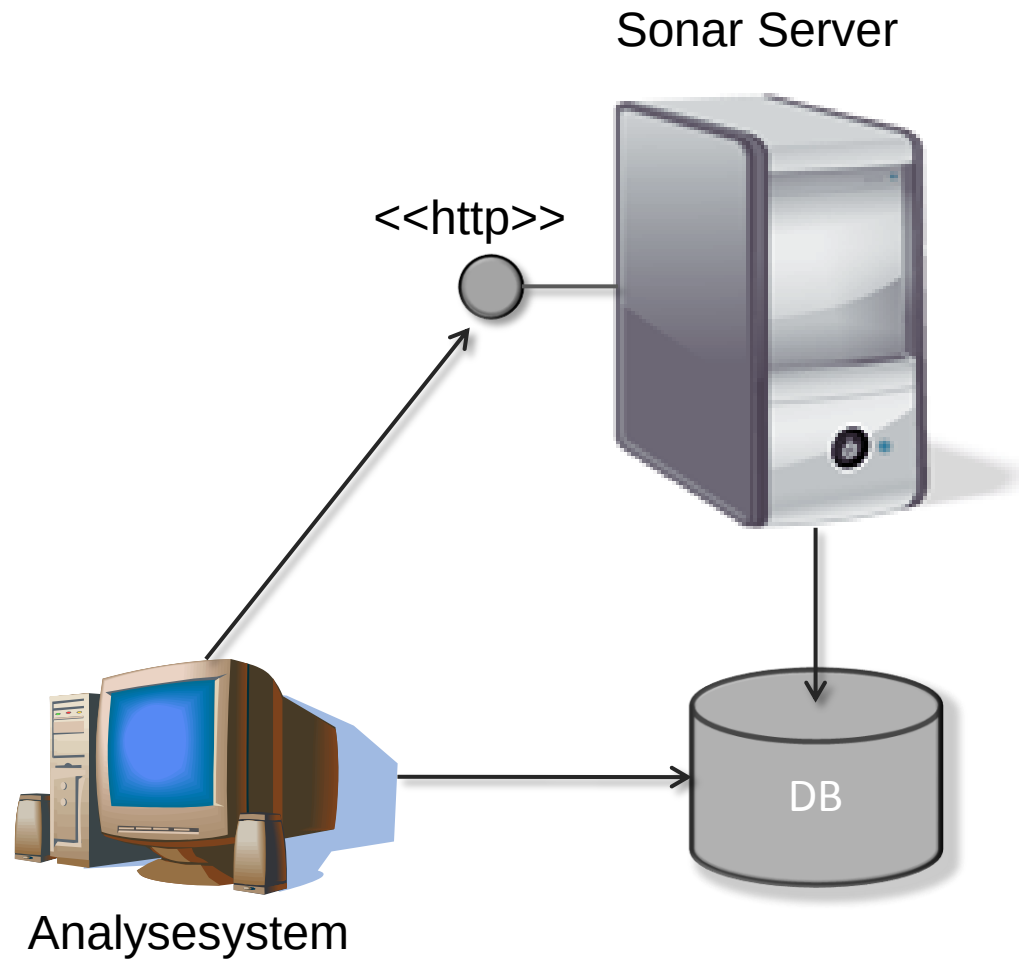
- Sonar bietet eine Sicht auf sieben Achsen/Dimensionen der Code Qualität



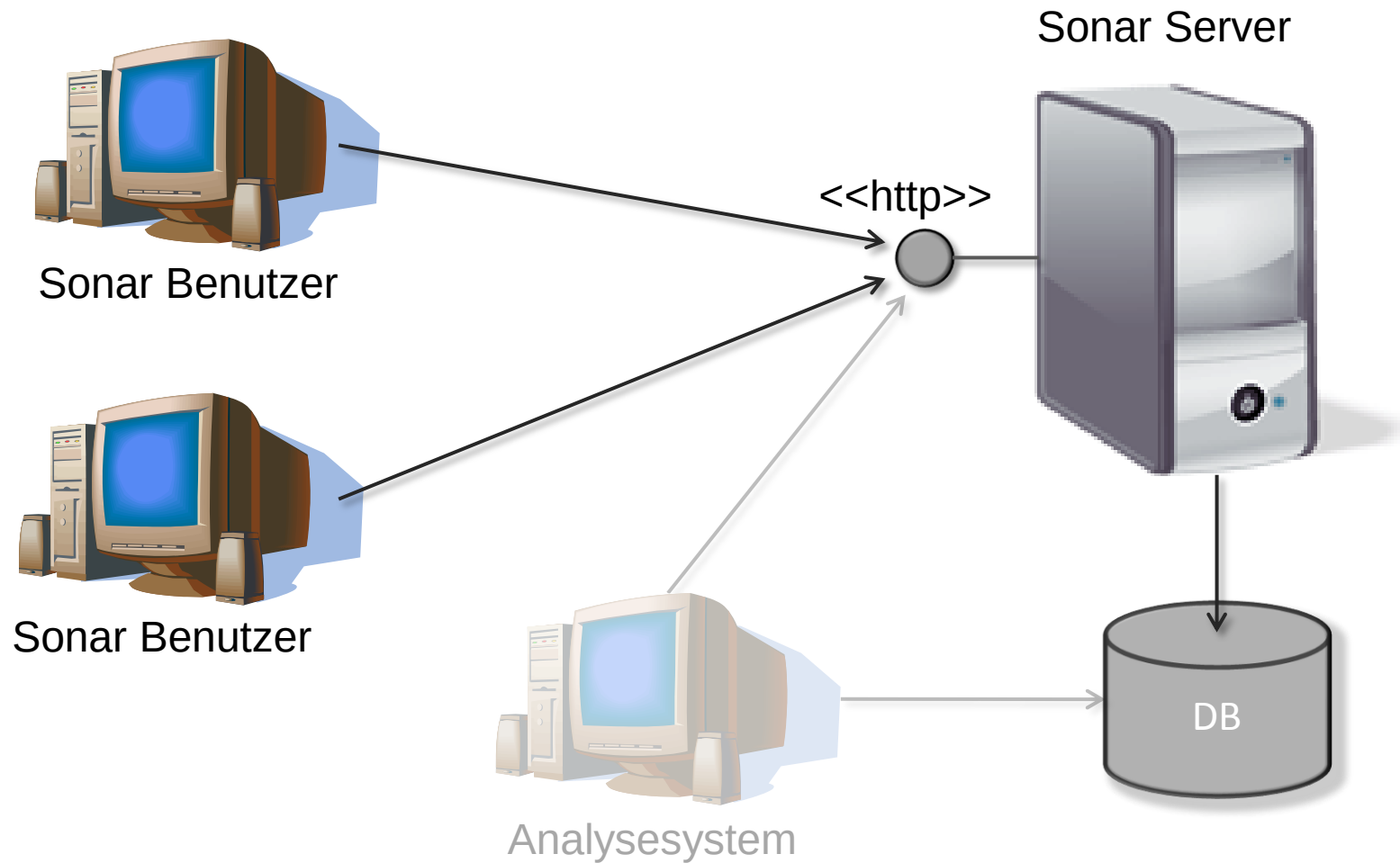
■ Sonar Setup



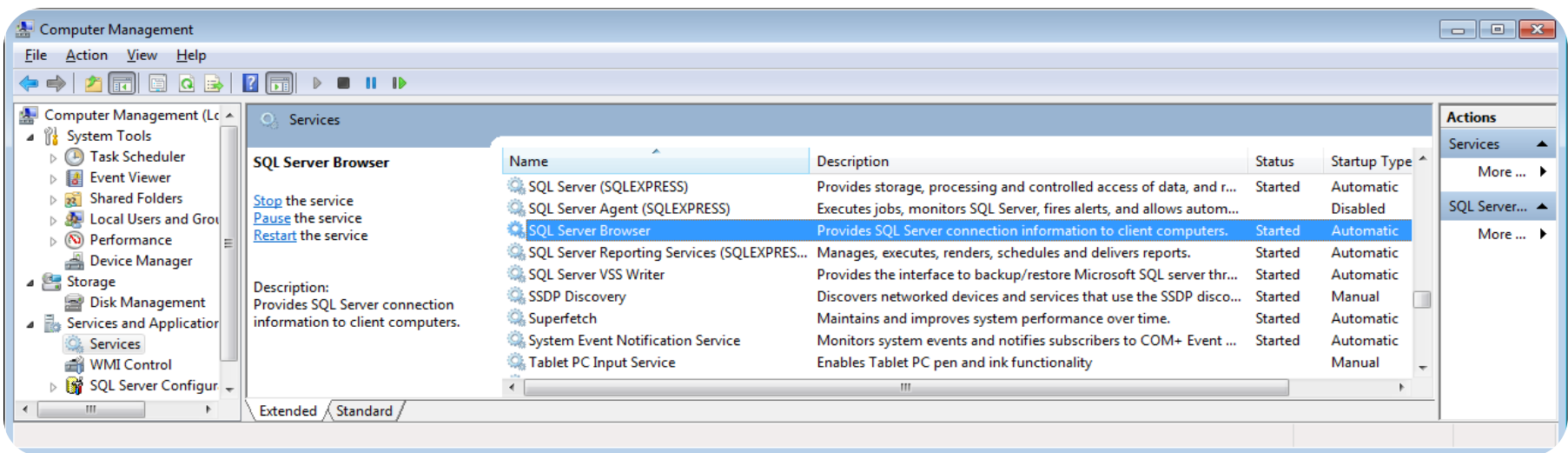
■ Sonar Setup



■ Sonar Setup



- Vorbereitung SQL Server (Express)
 - Benutzer anlegen (z.B sonar/sonar mit SQL Server Authentifizierung)
 - Datenbank in SQL Server anlegen
 - SQL Server Browser aktivieren
 - Wird für jDTS JDBC Treiber benötigt



- Vorbereitung Sonar Konfigurieren
 - JAVA_HOME
 - Sonar.config anpassen (Datenbank Konfiguration)

- ... starten



Los geht's

- Motivation
- Sonar Überblick
- Demo
- Fazit

- Fazit
 - Sonar lässt sich sehr schnell aufsetzen
 - Konfiguration der „Violation rules“ einfach
 - Definition der Sichten (Dashboards) erleichtert Adressaten gerechte Informationen
 - Sicherheitskonzept mit Benutzern und Gruppen
 - Sowohl Java als auch C#

Viel Spaß mit **sonar** 

The sonar logo consists of three concentric blue arcs of increasing size, suggesting a sonar wave or signal.

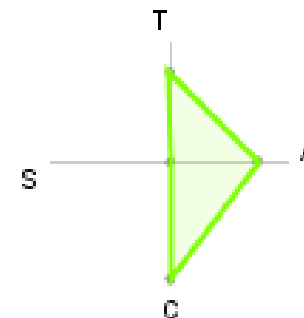
thomas.haug@mathema.de

- Backup

- Definiert von Software Improvement Group
- Einfaches Wartbarkeitsmodell
- basiert auf 4 Dimension
 - Analysierbarkeit
 - Änderbarkeit
 - Stabilität
 - Testbarkeit

SIG Maintain. Model ?

(A)nalsability	+
(C)hangeability	++
(S)tability	--
(T)estability	+



- (Projekt/System) Volumen
 - Größe des gemessenen System

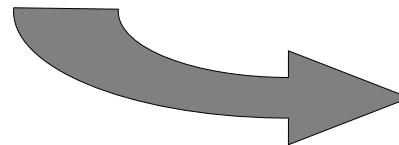
Rank	LOC
--	> 1310000
-	> 655000
0	> 246000
+	> 66000
++	> 0

- Duplikate
 - Anzahl duplizierte Zeilen im System

Rank	Duplication
--	> 20%
-	> 10%
0	> 5%
+	> 3%
++	> 0%

- (Projekt/System) Komplexität
 - Nutzt die McCabe zyklomatische Komplexität
 - Prozentualer Anteil der LOC, die zu Methoden gehören, die in einem Komplexitäts-Bereich liegen

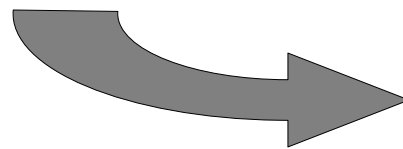
Eval	Complexity
Very high	> 50
High	> 20
Medium	> 10
Low	> 0



Rank	Medium	High	Very High
++	< 25%	< 0%	< 0%
+	< 30%	< 5%	< 0%
0	< 40%	< 10%	< 0%
-	< 50%	< 15%	< 5%

- Unit (= Methoden) Größe
 - Prozentualer Anteil von LOC, die zu Methoden eines gewissen Größenbereichs (LOCs) gehören

Eval	LOCs
Very high	> 100
High	> 50
Medium	> 10
Low	> 0



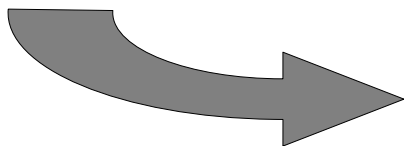
Rank	Medium	High	Very High
++	< 25%	< 0%	< 0%
+	< 30%	< 5%	< 0%
0	< 40%	< 10%	< 0%
-	< 50%	< 15%	< 5%

- Unit-Tests
 - Code Coverage (Zeilenabdeckung)

Rank	Coverage
++	> 95%
+	> 80%
0	> 60%
-	> 20%
--	> 0%

■ Sonar Erweitern

	Volume	Complexity	Duplications	Unit size	Unit tests
analysability	✓		✓	✓	✓
changeability		✓	✓		
stability					✓
testability		✓		✓	✓



SIG Maintain. Model ?

(A)nalysability +
 (C)hangeability ++
 (S)tability --
 (T)estability +

