Key Performance Indicator

Line Counting			
Lines of Code	481219	↗	✓
Source Lines Of Code	401326	↗	✓
Effective Lines Of Code	325196	↗	✓
Cyclomatic Complexity	12265	↗	✓
Comment Rate	16 %	↘	✗

Decision Making			
Business Value	1912	→	**i**
Technical Debt	3626	↗	**i**
Maturity Index	65 %	↘	**C**
Stability Index	84 %	↗	**B**
Reusability Index	44 %	↘	**D**

Squore 17.0.8
Eclipse Plugin Guide

Reference : SUM_Squore_ECL_PLG
Version : 17.0.8
Date : 30/11/2017

Eclipse Plugin Guide

Copyright © 2017 Squoring Technologies

Abstract

This edition of the Eclipse Plugin Guide applies to Squore 17.0.8 and to all subsequent releases and modifications until otherwise indicated in new editions.

Table of Contents

Typographical Conventions

The following conventions are used in this manual.

Typeface or Symbol	Meaning
Bold	Book titles, important items, or items that can be selected including buttons and menu choices. For example: Click the **Next** button to continue
Italic	A name of a user defined textual element. For example: **Username** : *admin*
`Courier New`	Files and directories; file extensions, computer output. For example: Edit the `config.xml` file
`Courier Bold`	Commands, screen messages requiring user action. For example: **Username** : *admin*
>	Menu choices. For example: Select **`File > Open`**. This means select the File menu, then select the Open command from it.
<...>	Generic terms. For example: `<SQUORE_HOME>` refers to the Squore installation directory.

Notes

Screenshots displayed in this manual may differ slightly from the ones in the actual product.

Acronyms and Abbreviations

The following acronyms and abbreviations are used in this manual.

CI	Continuous Integration
CLI	Command Line Interface
DP	Data Provider, a Squore module capable of handling input from various other systems and import information into Squore
RC	Repository Connector, a Squore module capable of extracting source code from source code management systems.

1. Introduction

1.1. Foreword

This document was released by Squoring Technologies.

It is part of the user documentation of the Squore software product edited and distributed by Squoring Technologies.

1.2. About This Document

This document is the Eclipse Plugin Guide for Squore.

Its objective is to guide you through various aspects of the Squore Eclipse Plugin:

→ Chapter 2, *Introducing the Squore Eclipse Plugin* positions the Squore Eclipse Plugin inside the Squore ecosystem, explaining its main features.

→ Chapter 3, *Installing and setting up the Squore Eclipse Plugin* describes the steps to follow in order to install the Squore Eclipse Plugin.

→ Chapter 4, *Getting Started With the Squore Eclipse Plugin* walks you through the features provided by the Squore Eclipse Plugin.

If you are already familiar with Squore, you can navigate this manual by looking for what has changed since the previous version. New functionality is tagged with **(new in 17.0)** throughout this manual. A summary of the new features described in this manual is available in the entry *** What's New in Squore 17.0?** of this manual's Index.

For information on how to use and configure Squore, the full suite of manuals includes:

→ Squore Installation Checklist

→ Squore Installation and Administration Guide

→ Squore Getting Started Guide

→ Squore Command Line Interface

→ Squore Configuration Guide

→ Squore Eclipse Plugin Guide

→ Squore Reference Manual

1.3. Contacting Squoring Technologies Product Support

If the information provided in this manual is erroneous or inaccurate, or if you encounter problems during your installation, contact Squoring Technologies Product Support: http://support.squoring.com/

You will need a valid Squore customer account to submit a support request. You can create an account on the support website if you do not have one already.

For any communication:

🖥 **support@squoring.com**

✉ **Squoring Technologies Product Support**
76, allées Jean Jaurès / 31000 Toulouse - FRANCE

1.4. Responsabilities

Approval of this version of the document and any further updates are the responsibility of Squoring Technologies.

1.5. Getting the Latest Version of this Manual

The version of this manual included in your Squore installation may have been updated. If you would like to check for updated user guides, consult the Squoring Technologies documentation site to consult or download the latest Squore manuals at http://support.squoring.com/documentation/17.0.8. Manuals are constantly updated and published as soon as they are available.

2. Introducing the Squore Eclipse Plugin

2.1. Squore Eclipse Plugin in the Squore ecosystem

Squore analyzes projects (in a broad sense) to provide high level indicators and produce personalised dashboards and tailored action lists for each kind of connected user. The whole process can be automated, and integrated in a continuous integration framework.

Information generated in this fashion can be refreshed with any desired rate, and the results will be available each time the Squore process concludes a new build. However, this scenario is best suited to a daily cycle at most, where all input files are the result of a daily commit, and results are generated along with the nightly build.

The Squore Eclipse Plugin is destined to operate at a faster rate, the objective being to deliver results during the development process, before files are committed and fed to the continuous integration process. Tightening the processing loop has several consequences for the Squore Eclipse Plugin behavior:

→ *It doesn't need to be connected to a central Squore server to rate a project, unless for project activation / licence validation purposes.*
By limiting communication with the server to a minimum, we make sure projects can evolve outside the global rating cycle, but still offer the rating service.

Should the server be unreachable, rating service is still available for a whole week, before being deactivated. Once a connection is available again, and the credentials are validated, the rating capabilities are restored.

→ *It must be able to process and rate a project locally*
The Squore Eclipse Plugin runs the Squan Sources static code analyser on the sources in your Eclipse project.

Note that no other data provider is run by the Squore Eclipse Plugin, so the overall rating of your project may vary from the one obtained when analysing the same for the same code base on Squore Server. Metrics and violations returned by Squan Sources will nevertheless be the same.

→ *It must use a configuration (in terms of Analysis Model and Data Providers) synchronised with the one installed on Squore Server*
Since the Squore Eclipse Plugin uses a Squore rating module, it needs a rating environment, which must be synchronised with the server's, to ensure results equivalence.

The Squore Eclipse Plugin will perform such synchronisation during the product activation, which is a feature available in the Squore Eclipse Plugin preferences page.

→ *It must be able to manage projects using other languages than Java (e.g. C, C++, C# Ada)*
The Squore Eclipse Plugin benefits from the Squore source code analyzer, which provides support for several languages (see Squore documentation for more details).

Analysis results will be available at folder and file level.

→ *It must be integrated into the Eclipse IDE, taking advantage of its extendability*
The Eclipse plugin development framework offers a number of mechanisms to enhance various parts of the IDE.The added features have been designed to be both unobtrusive and informative.

→ *It must enhance a project's available information by displaying ratings, Action Items, Findings and highlights for each compatible object*
The added information is assigned to each artefact, which means at project, file, module, class and function level. Each of these elements will have a rating, as well as a list of Action Items and Findings, depending on the model. A search view offering project artefacts sorted by rating is also available.

→ *It must be able to manage local versions, which are dissociated from the source code control versioning mechanism*

One of the differences between the Squore Eclipse Plugin and Squore is that Squore Server uses a database to store the project builds across multiple versions, whereas the Squore Eclipse Plugin uses a local file storage mechanism, to maintain a local history.

This allows functions such as `DELTA_VALUE` or `PREVIOUS_VALUE` to work as expected, and produce trend analysis indicators.

Trends are based on the evolution of indicators between a baseline analysis and the last-rated analysis.

2.2. Who is the Squore Eclipse Plugin destined to?

As hinted by its name, the Squore Eclipse Plugin is a developer tool. Its objective is to provide users with enough information to help them:

→ Process the generated list of Action Items

→ Check the evolution of the source code rating as it is being produced

→ Find what artefacts are responsible for a poor rating, and understand why

→ Anticipate and address problems that will appear in the project's daily Squore server build before the code is referenced in a SCM system

As the Squore Eclipse Plugin seamlessly integrates Squan Sources results in the IDE, and uses the same model as the one on Squore Server, developers will gain visibility over the evaluation process, and improve their development practices to match the standard defined by the model.

The integrated rating feedback also gives the developer the opportunity to actively participate to the global effort on quality improvement.

3. Installing and setting up the Squore Eclipse Plugin

3.1. Installation Prerequisites

3.1.1. Supported Operating Systems

The following is a list of the officially supported and tested operating systems:

→ CentOS 6
→ CentOS 7
→ Fedora 19
→ Ubuntu Server 14.04
→ Windows 7
→ Windows 8
→ Windows 10

> **Note**
> On Linux, a 64-bit version of the OS is required
>
> On Windows, a 64-bit version of the OS is required if you want Squore to run as a Windows service, but the installation is also supported on a 32-bit system (and is started and stopped using .bat files instead of a Windows Service)

The following is a list of the operating systems that are not regularly tested but are known to be working:

→ RedHat EL 6
→ RedHat EL 7
→ SuSe Linux 11.1
→ Ubuntu Server 10.04
→ Ubuntu Server 16.04
→ Windows Server 2008 R2
→ Windows Server 2012 R2

3.1.2. For All Systems

For a successful installation of Squore, you will need:

→ The latest version of the Squore Eclipse Plugin, which can be downloaded from http://localhost:8180/SQuORE_Server/eclipse
→ The Oracle Java Runtime Environment version 8 (other versions are not supported)
→ A supported Eclipse version (Luna SR2 and up)
→ Valid credentials to log into a Squore Server
→ At least 2 GB of space available on the disk for a full installation
→ At least 8 GB of RAM on the server machine
→ At least 4 GB of RAM on the client machine

3.1.3. Packages for Windows

A compatible version of Eclipse must be downloaded from http://www.eclipse.org/downloads/ .

3.1.4. Packages for Linux

On Linux platforms, the following must be installed before installing Squore:

→ **Perl** version 5.10.1 or greater including the following extra-modules:
  → Mandatory packages:
    → **Algorithm::Diff** [module details] [http://search.cpan.org/~nedkonz/Algorithm-Diff/lib/Algorithm/Diff.pm]
    → **Archive::Zip** [module details] [http://search.cpan.org/~phred/Archive-Zip/lib/Archive/Zip.pm]
    → **Date::Calc** [module details] [http://search.cpan.org/~stbey/Date-Calc/lib/Date/Calc.pod]
    → **Digest::SHA** [module details] [http://search.cpan.org/~mshelor/Digest-SHA/lib/Digest/SHA.pm]
    → **HTTP::Request** [module details] [http://search.cpan.org/~gaas/HTTP-Message/lib/HTTP/Request.pm]
    → **JSON** [module details] [http://search.cpan.org/~makamaka/JSON/lib/JSON.pm]
    → **LWP** [module details] [http://search.cpan.org/~ether/libwww-perl/lib/LWP.pm]
    → **LWP::UserAgent** [module details] [http://search.cpan.org/~gaas/libwww-perl/lib/LWP/UserAgent.pm]
    → **Time::HiRes** [module details] [http://search.cpan.org/~zefram/Time-HiRes/HiRes.pm]
    → **XML::Parser** [module details] [http://search.cpan.org/~toddr/XML-Parser/Parser.pm]

    Tip
    If some of these modules are not available as packages on your operating system, use your perl installation's cpan to install the modules. Using the OS packages is recommended, as it avoids having to reinstall via cpan after upgrading your version of perl.

→ **Tcl** version 8.5 or greater,
→ A supported Eclipse version (Luna SR2 and up)

3.1.5. Packages for CentOS and Red Hat Enterprise Linux

On Red Hat Enterprise Linux and CentOS (6.5 and 7.1), the dependencies are satisfied by the following packages:

Mandatory packages:

→ **java-1.8.0-openjdk**
→ **perl**
→ **perl-Algorithm-Diff**
→ **perl-Archive-Zip**
→ **perl-Date-Calc**
→ **perl-Digest-SHA**
→ **perl-JSON**
→ **perl-libwww-perl**
→ **perl-Time-HiRes**
→ **perl-XML-Parser**
→ **tcl**
→ A compatible installation of Eclipse downloaded from http://eclipse.org/downloads .

For more information about how to install the Extra Packages for Enterprise Linux (EPEL) repository, consult https://fedoraproject.org/wiki/EPEL.

3.1.6. Packages for Ubuntu

On Ubuntu 14.04.3 LTS, the dependencies are satisfied by the following packages:

Mandatory packages:

→ **libalgorithm-diff-perl**

→ **libarchive-zip-perl**

→ **libdate-calc-perl**

→ **libhttp-message-perl**

→ **libjson-perl**

→ **libwww-perl**

→ **libxml-parser-perl**

→ **openjdk-8-jre**

→ **perl**

→ **tcl**

→ A compatible installation of Eclipse downloaded from http://eclipse.org/downloads .

3.2. Squore Eclipse Plugin installation

The Squore Eclipse Plugin complies with the Eclipse requirements, and is an OS-independent package to be deployed in the Eclipse plugins directory. Installation is done through an update site.

Tip

In order to launch analyses, the plugin needs to use an installation of perl and tclsh. If you are using a Windows operating system, the easiest way to install both dependencies is to install the Squore CLI package on your machine following the procedure outlined in the Command Line Interface manual.

→ Create a new update site within Eclipse
  → Select **Help > Install new software;**.
  → Click on **Available Software Sites;**.
  → Add a new site;
  → Give it a name (for example: "Squore Eclipse Plugin");
  → Set the location to **http://localhost:8180/SQuORE_Server/eclipse**
→ Go back to the **"Install"** window with the **"OK"** buttons;
→ Select the newly created update site;
→ Proceed with the plugin installation as with any Eclipse plugin.

Note

→ To speed-up installation procedure, it is possible to uncheck "Contact all update sites during install to find required software", as no external software is needed by the Squore Eclipse Plugin.

→ Should the plugin fail to deploy, Eclipse will notify the user of the problem cause, and will start its operations as usual. If necessary, copy and send the generated Eclipse log to support@squoring.com.

3.3. Squore Eclipse Plugin setup

To use the Squore Eclipse Plugin, it must be activated, the server must be reachable with proper credentials, and rating preferences must be provided.

All these operations are accessible through the **Window > Preferences > Squore** page. Note that all provided information uses the Eclipse secure storage.
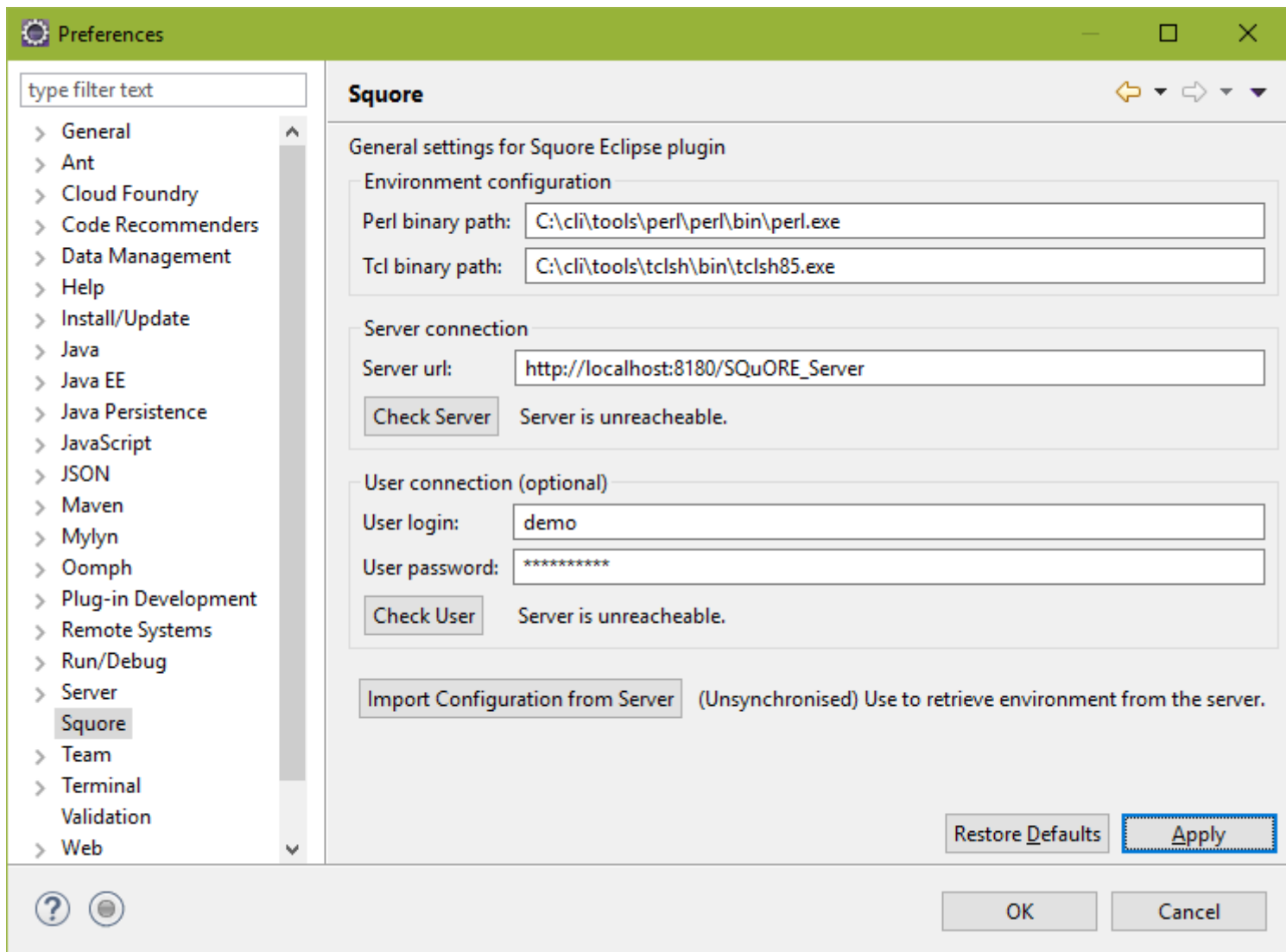
The Squore Eclipse Plugin preferences page

Fill in the paths to the perl and tclsh executables and the server details and click **Check Server** to ensure that your configuration is correct.

If you can successfully connect to the server, click **Import Configuration from Server** to synchronise your environment with the one from Squore Server.

When the synchronisation finishes, click **OK** to close the preferences window.

4. Getting Started With the Squore Eclipse Plugin

4.1. Launching Your First Analysis

Once the Squore Eclipse Plugin is installed in Eclipse, obtaining a rating for your project is rather straightforward.

First locate the Squore contextual menu entry for the project you want to rate.
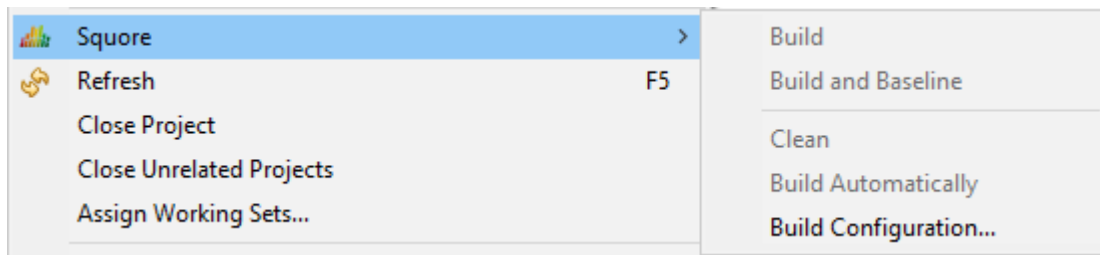
Contextual menu on an Eclipse project

Rating (or "building") a project with Squore Eclipse Plugin can either be performed as a baseline build or a draft build.

→ When choosing baseline build, the project rating will produce a new reference version, with which all subsequent ratings will be compared to. This will affect rating computations based on metrics previous values, as well as all trends displayed in the Squore Eclipse Plugin views.
The baseline mode is useful when a stable state of the project has been reached, and a development phase is about to begin.

→ When choosing draft build, the project is rated as usual, but the reference version is not modified. Each new draft build overwrites the previous analysis and displays trends and deltas relative to the last defined baseline.
The draft mode is useful during a development phase, when the project can undergo several back and forth changes, and the user wants to keep comparing the current version to the last stable one.

Before you can build a project, you first need to define its build configuration:
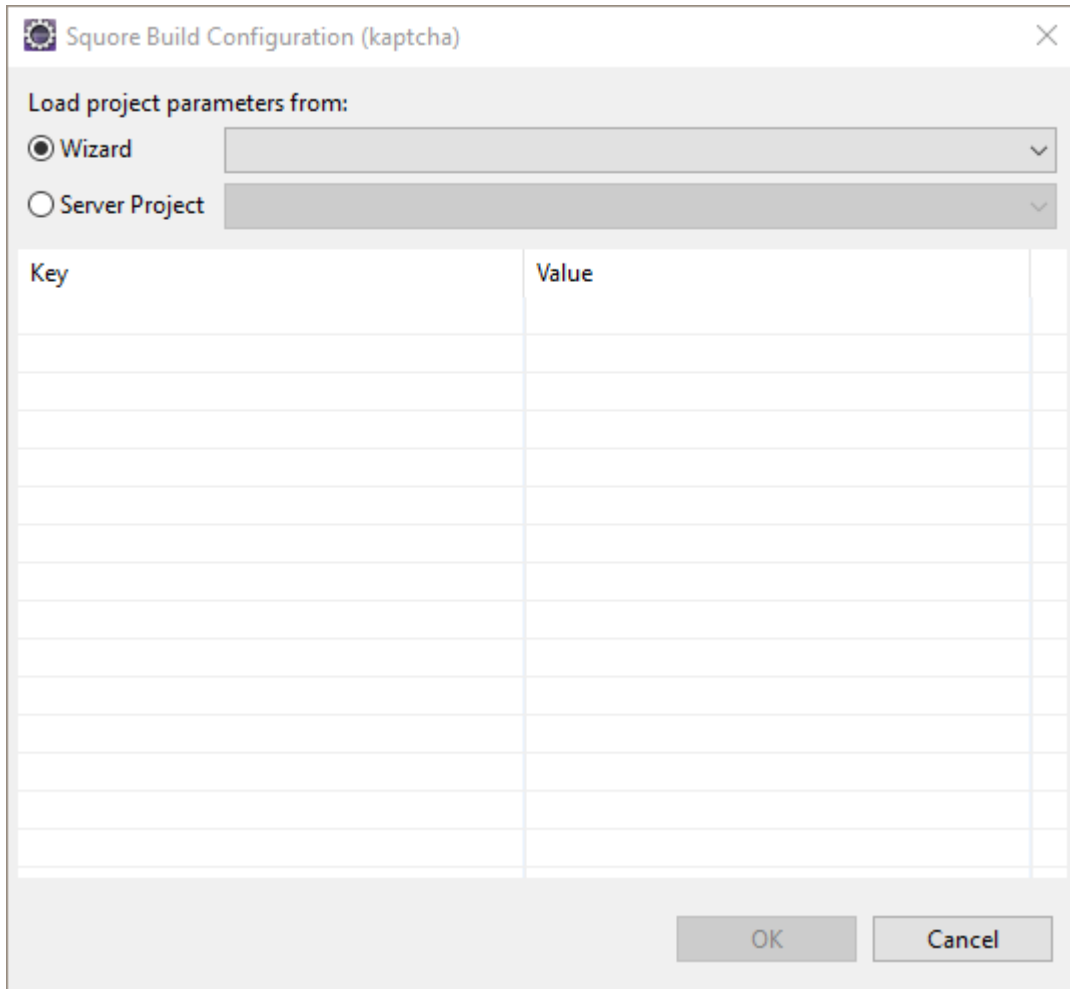
The Build Configuration screen

On the Build configuration screen, you can choose an analysis model for your project, or you can select the project on the server that corresponds to the code you are working on locally. By selecting an existing project, you can retrieve the project attributes from the server, and therefore run your local analysis in the same conditions as it is run on the server.
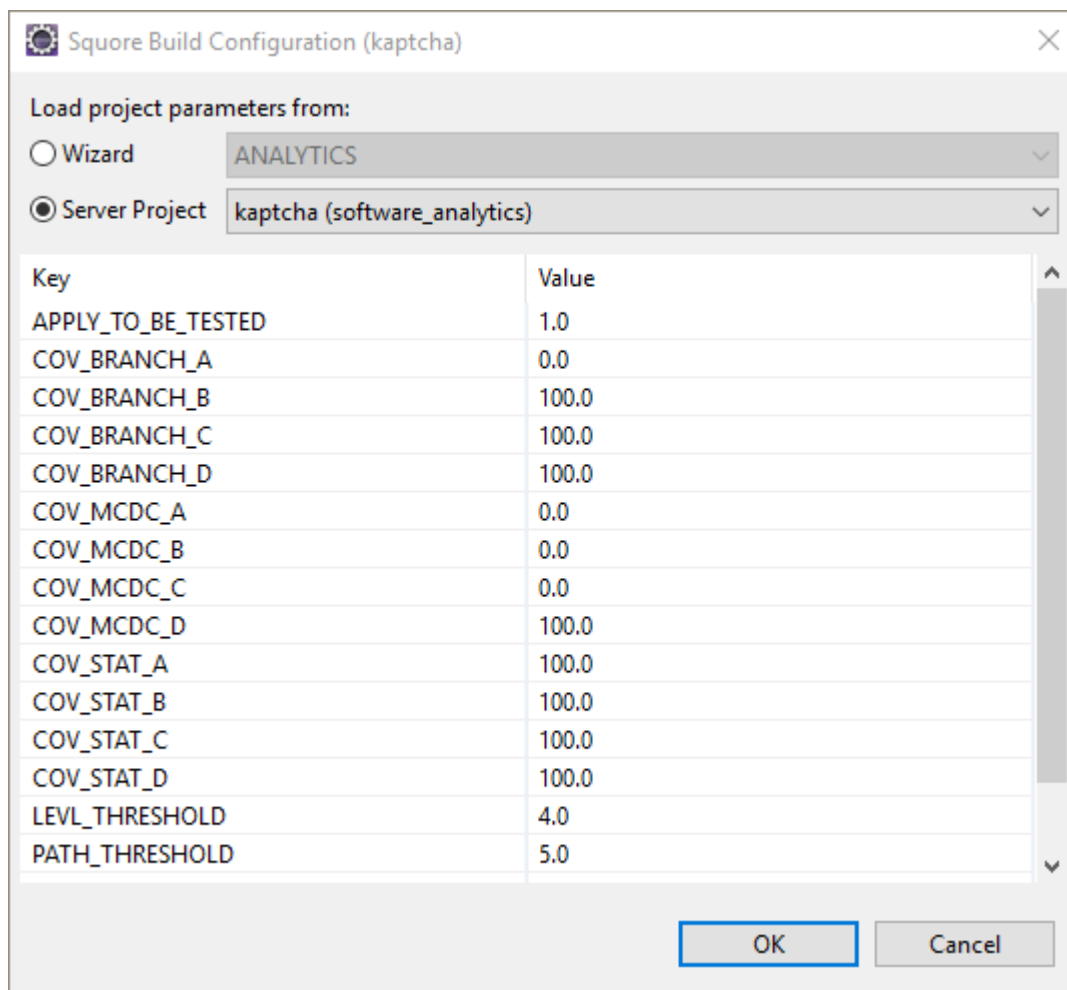
Project options for the kaptcha project on the server

Click **OK** to close the configuration dialogue. You are now ready to start building your project using the draft or baseline submenu in the Squore Eclipse Plugin context menu.

4.2. Visualising Analysis Results

After your analysis completes, switch to the Squore perspective to view the results.

Tip
Click **Window > Perspective > Open Perspective > Other** and search for **Squore** in the list of available perspectives if you are using it for the first time.

The Squore perspective displays various views with information relative to the file or folder you click in the Project Explorer.

The Squore Perspective with the Squore Card, Action items, Findings and Highlights views in the bottom panel

Click a file or folder in the Project Explorer and use one of the following views in the bottom panel to understand your rating:

→ The Squore Card view

displays the overall rating for the current artefact, and its trend. It also lists the tables of interests for this artefact, as defined in the Analysis Model used to rate the project.

| ⯈ Squore Card ✕ | ⯈ Action Items | ⯈ Findings | ⯈ Highlights | ⯐ Console |

FishEyeGimpy.java

D
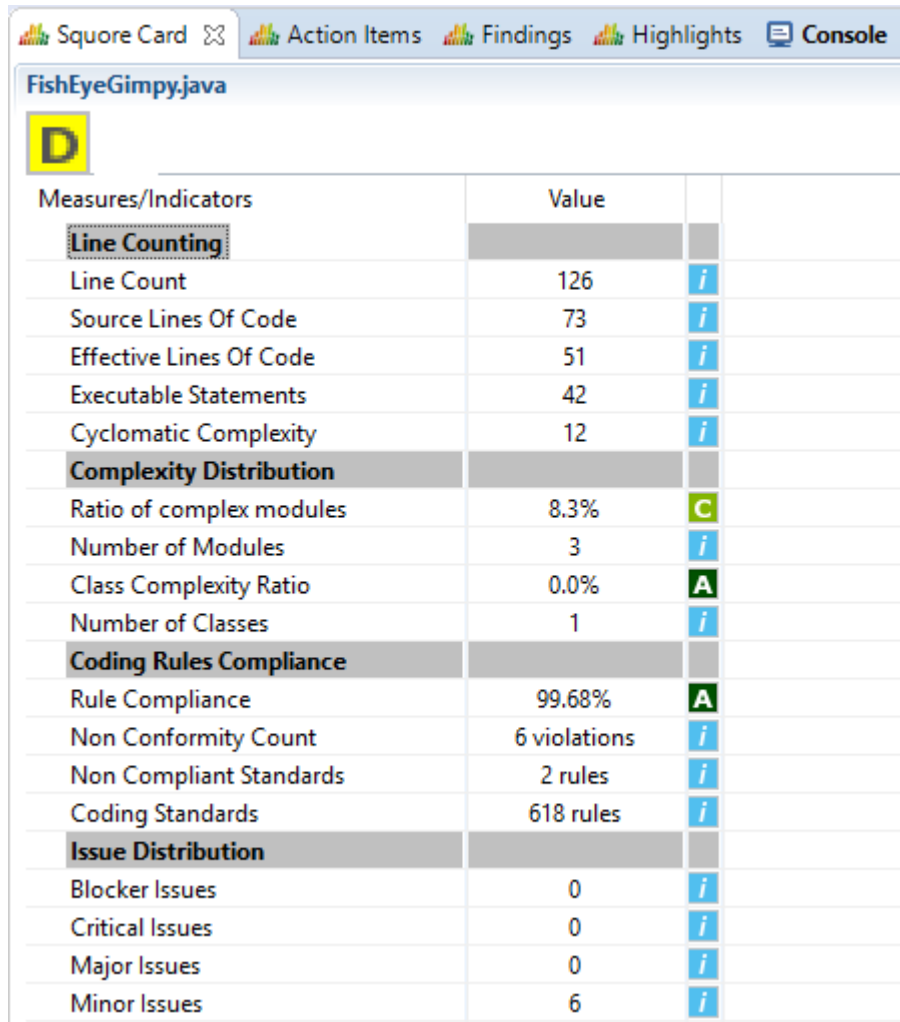
| Measures/Indicators	Value	
Line Counting		
Line Count	126	i
Source Lines Of Code	73	i
Effective Lines Of Code	51	i
Executable Statements	42	i
Cyclomatic Complexity	12	i
Complexity Distribution		
Ratio of complex modules	8.3%	C
Number of Modules	3	i
Class Complexity Ratio	0.0%	A
Number of Classes	1	i
Coding Rules Compliance		
Rule Compliance	99.68%	A
Non Conformity Count	6 violations	i
Non Compliant Standards	2 rules	i
Coding Standards	618 rules	i
Issue Distribution		
Blocker Issues	0	i
Critical Issues	0	i
Major Issues	0	i
Minor Issues	6	i

The Squore card

→ The Findings view

lists the Findings for the current artefact as well as underlying artefacts, with their location and definition details. This list can be sorted, and allows navigation to artefacts associated to each finding by double-clicking on their name.
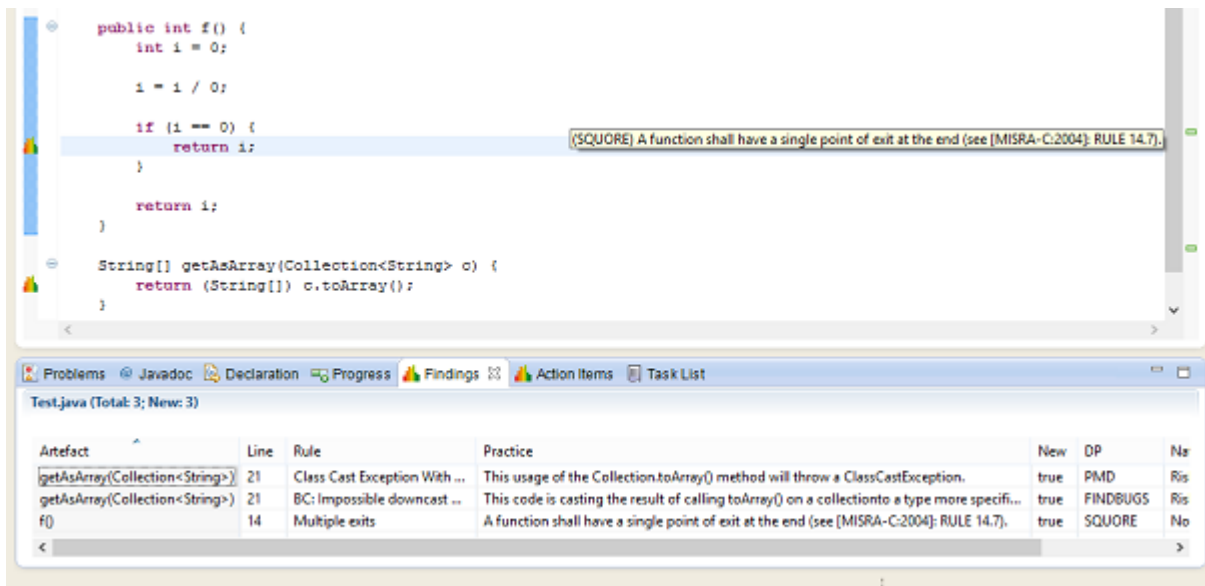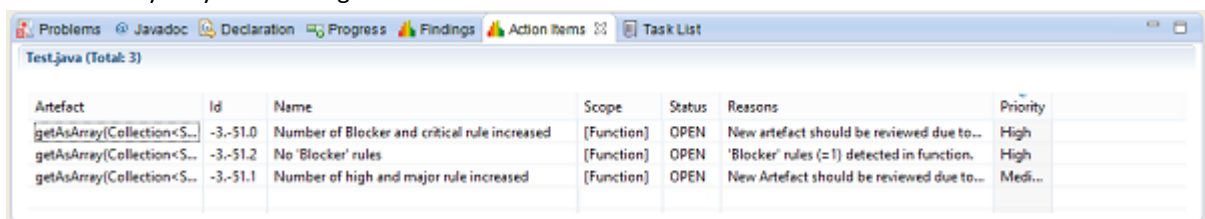
A finding and its location in the source code
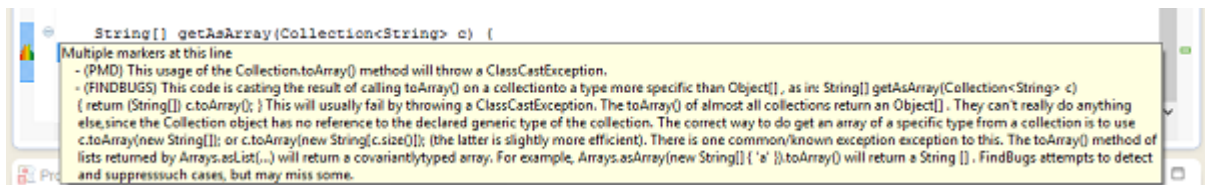
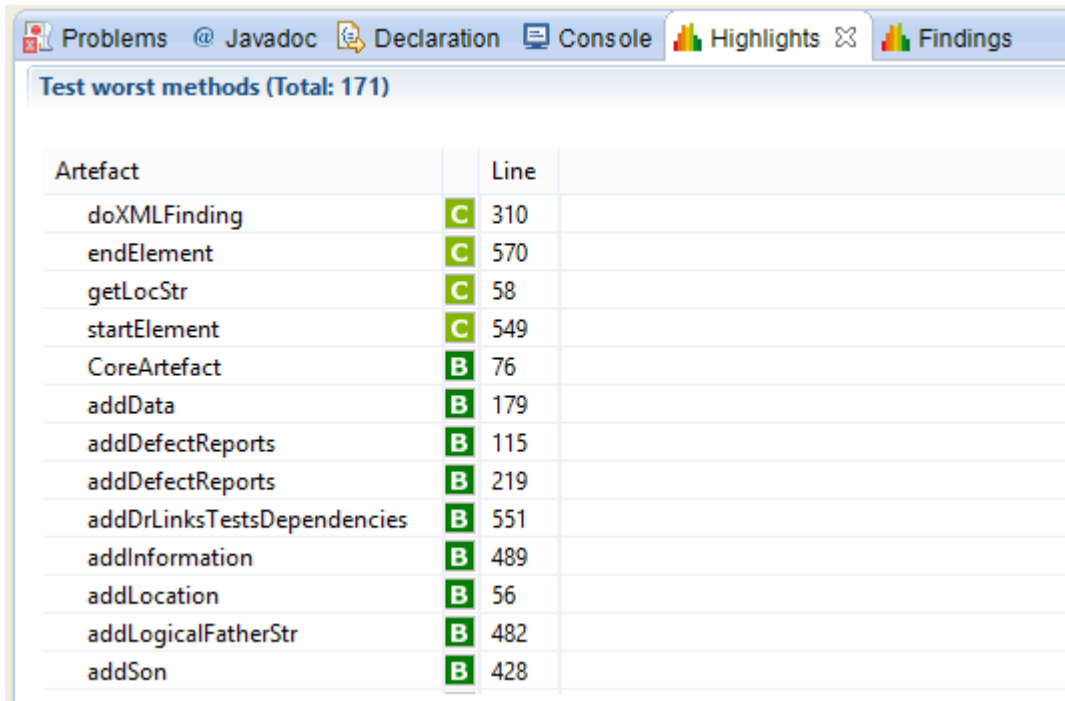A finding full description

→ **The Action Items view**
works much like the Findings one, except it lists the Action Items computed by the Decision Model, and the reason why they have been generated.

Action Items

→ **The Highlights view**
works much like the Findings one, except it provides a list of the current project's Files, Classes or Functions, sorted by decreasing rating. Selecting the type of object to sort (File, Class or Function) is performed through the top-left menu icons.

Highlights

Tip

Only the highlights created for the role **PLUGIN** are displayed in Eclipse. For more information about configuring highlights in your model, refer to the Configuration Guide.

Index

A

Action Items view, 14

B

Baseline build, 9

D

Disk Space, 5
Draft build, 9

F

Findings view, 13

H

Highlights view, 14

J

Java, 5

M

Memory, 5, 5

P

Prerequisites, 5

S

Squore Card view, 13