QStudio® Enterprise

The Software Assessment Tool

QStudio Enterprise 2.2

QStudio Enterprise automates software assessment, software quality trend analysis and software quality benchmarking. The powerful and unique software assessment engine is capable of *automatically* generating management oriented software assessment reports with benchmarking and trend analysis capabilities. The purpose of QStudio Enterprise is to significantly reduce the risk to the enterprise associated with software development, software delivery and software acceptance.

In addition, QStudio Enterprise supports drill down capabilities to all identified issues within the software and an enterprise-strength coding standards compliance system for a multi-user, multi-coding standard, multi-project environment The software engine database can be coupled to code management, enterprise reporting/data mining and software defect management systems.

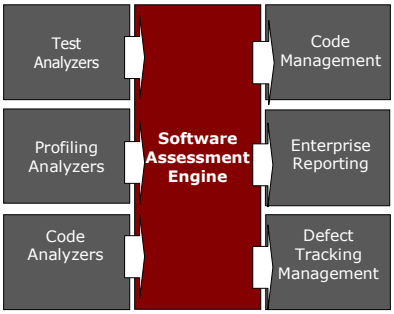
QStudio Enterprise 2.2 seamlessly integrates with the industry leading automated software inspection tool (code analyzer) for Java: QStudio for Java Pro. Later this year QA Systems will initiate activities for other domains including automated test analysis and profiling analysis.

QStudio Enterprise is unique in that it will integrate with 3rd party code analyzers for a variety of programming languages. QA Systems is working with industry leading code analyzer vendors and open source communities to provide the necessary integration. Initial analyzers targeted for QStudio Enterprise 3.0 include the Java open source analyzers CheckStyle, PMD and QJ Pro, Microsoft's FxCop for the .NET environment and Power Software's Krakatau analyzer for C/C++. Later this year QA Systems will be working to integrate test and profiling analyzers

QStudio Enterprise 2.2 is:

- a powerful software assessment engine based on the ISO 9126 software quality model that is capable of *automatically* generating management oriented software assessment reports (see portion of example report provided later in this document).
- an enterprise development support tool providing drill down capabilities to all identified issues within the software.
- an enterprise-strength coding standards compliance system for a multi-user, multi-coding standard, multi-project environment

These functions can be used either separately or in conjunction.



Use QStudio Enterprise if you need to:

Assess Your Software: You maintain or develop source code. You need to understand the software risk you have with respect to the quality of the source code.

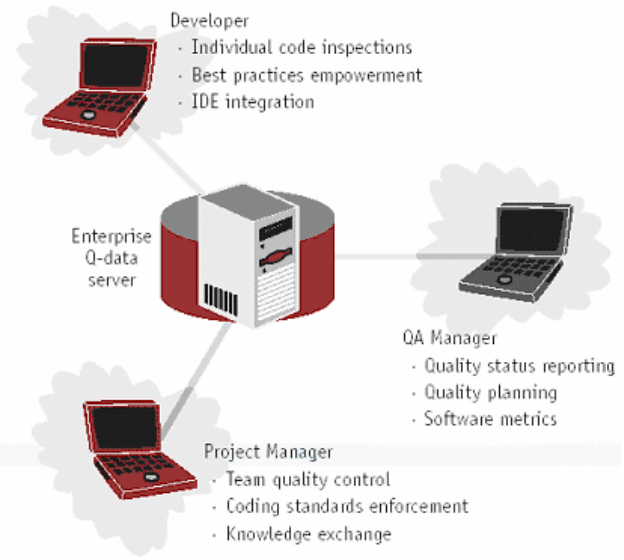
Assess 3rd Party Software: You carry out software risk assessments for internal or external parties. You need provide detailed insight and reporting into the potential software problems within the assessed software.

Accept Software: You outsource software development and you need to accept the software. Or you intend to insource software development for a customer You need to identify and understand the risks you are exposed to when accepting the code.

Deliver Software: You are delivering developed code to your internal or external customer. You need to demonstrate the delivered quality of the source code in terms that your customer's management can understand. QStudio Enterprise provides detailed visual overviews of the delivered code quality thereby giving you the ability to demonstrate the quality of your delivered source code.

Enforce Software Coding Compliance: You need to comply with internal or external software coding or quality standards. QStudio Enterprise supports multiple different coding compliance standards within multiple projects over the enterprise and can integrate with code management systems ensuring that only code satisfying each project specific coding standard can be checked in into the projects code base.

QStudio Enterprise automates team/departmental based software coding compliance. Multiple coding standards and quality standards configurations can be managed and made available to the software development team. QStudio Enterprise identifies potential defects in the software and provides development powerful drill-down capabilities to the identified issues.



QStudio Enterprise 2.2 supports the following features:

- Automated high level management report generation of software risk assessment (see example below)
- Multiple Coding standard compliance to departmental and project related coding standards
- Software quality trend analysis and benchmarking analysis
- Automated Annotated Source Code Generation for easy code review
- Web based graphical administration and reporting capabilities
- Command line interfaces for batch processing of large projects
- Interface with code management systems to ensure that only code satisfying the defined quality standard is committed to the code base
- Integrates with the industry leading open source Java code analyzer: QJ Pro

Interfaces to multiple 3rd party code analyzers for C/C++/Java/Cobol/Fortran and .NET planned for version 3.0.

Availability

QStudio® Enterprise 2.2 is available in April 2004 for Windows (98/2000/NT/XP/ME), Linux (RedHat Linux 6.1 and higher, SuSE Linux 7.0 and higher) and Solaris (Solaris 6.1 and higher). Pricing is €9950 for the base server, €2950 per supported analyzer, €1950 per assessment user (can execute software assessments) and €295 per developer user (supported by automated coding standards).

QA SYSTEMS
Software Assessment

QA Systems - The Software Assessment Company™ - develops tools for software assessment and software quality compliance.

www.qa-systems.com

Open Source Algebra DB Software Risk Assessment Report

Report not complete. Excerpts only for the sake of brevity.

The intention of this assessment report is to provide management information with respect to quality risk of the open source Algebra DB code base (algebradb.sourceforge.net).

This analysis was carried out using the QJ Pro Java code analyzer. The rule set against which the code base was tested applied is listed later in this report.

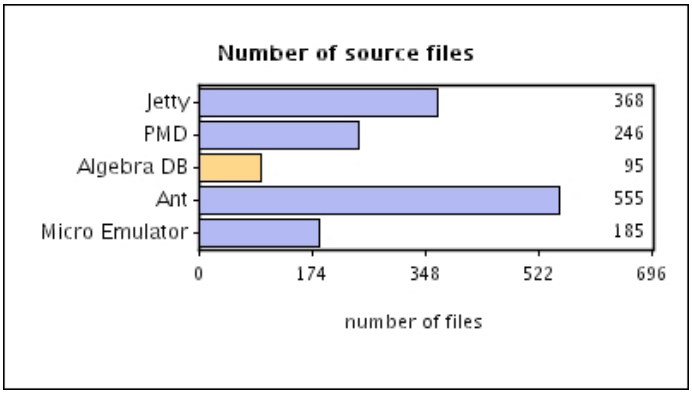
This assessment report indicates:

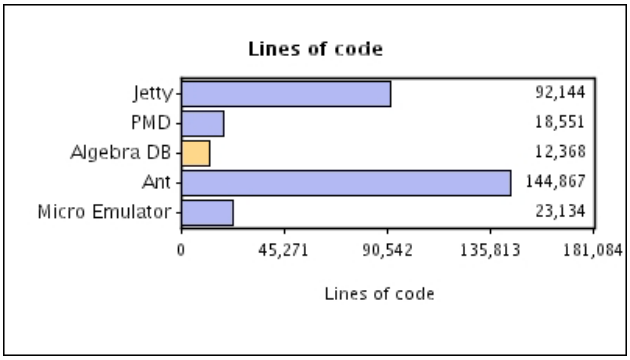
- How well the code base performs against the identified benchmark set for each particular metric
- The percentage of code satisfying a particular rating (derived from the benchmark set) for each particular metric
- The total number of observations for each given metric/risk

Users can build their own customized benchmark set using open source code bases and/or their own code base sets. By using consecutive versions of the same code base as the benchmark set, QStudio Enterprise automatically provides sophisticated code quality trend analysis capabilities.

Benchmark Set Characteristics

The code base Algebra DB was benchmarked against the following code bases. The code base size characteristics are as follows (observations/kloc represents the number of observations per thousand lines of code).





Risk Rating Model

QStudio implements a software risk rating model (Good, Sufficient, Suspicious and Bad). The rating model provides a valuation for the acceptable number of observations for each category. The lower the value, the better the lower the related risk aspect.

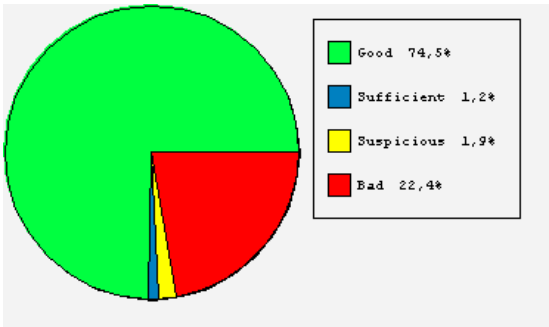
A value of 75% for “Good”, for example, indicates that 75% of the files in the source package analyzed satisfy the “Good” ratings criterion for the particular software risk under discussion.

The ratings transition values for each metric can be adjusted based on actual project experience and/or company guidelines giving the organization the ability to fine-tune these benchmarking capabilities.

Good	Measured value between 0 and X (value of ‘X’ depending on the metric)
Sufficient	X <= Sufficient < 2X
Suspicious	2X <= Suspicious < 3X
Bad	All values above 3X

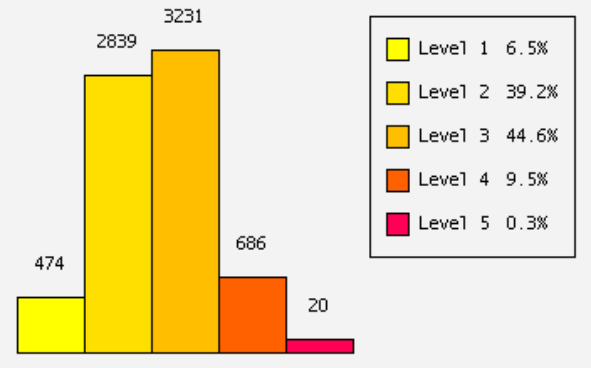
The actual ratings values are based on averaging the values from a default set of benchmark code bases. Users can build their own customized benchmark set using, for example, a combination of open source code bases and/or their own code base set.

Each pie chart gives an indication of the amount of source code satisfying a particular rating. So for the pie chart below approximately 74.5% of the code satisfies a rating of Good and 22.4% has a rating of Bad. We say approximately because in fact it is the percentage of files satisfying a particular rating that is measured. For large code bases this percentage will represent a reasonable approximation of the actual percentage of code satisfying the particular rating. The pie charts give a first order indication of the amount of effort that would be necessary to improve the code base. This represents a quantitative statement about code quality and risk.



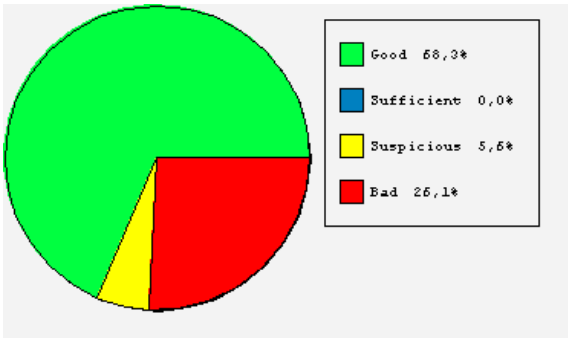
Impact Analysis

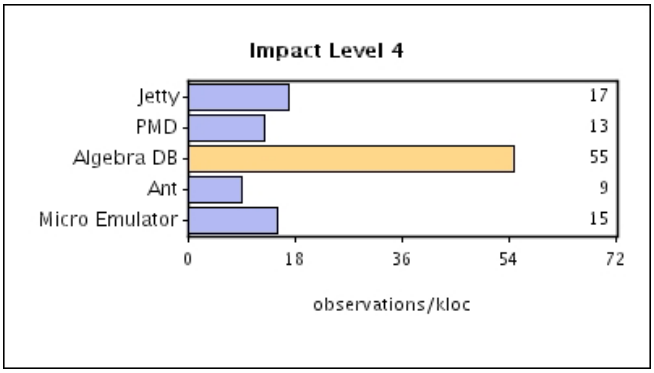
QStudio Enterprise diagnoses 5 impact levels indicating the risk severity of a potential software defect. Level 1 representing the least risk severity and level 5 the highest. The source code base displayed the following risk severity profile: [so for example 20 observations were detected with severity level 5]



Example Impact Level - Level 4

Level 4 (one but highest severity level) points to a risk that parts of functionality cannot be used or that basic product features such as performance, resource behavior, user friendliness or accuracy are not within acceptable limits. Approximately 69% of the code satisfies Level 4 compliance. The source code base displayed the following risk profile for level 4 observations:





Risk Assessment Characteristics

The risk assessment model defines a stepwise refinement of the notion of software risk assessment into a set of ISO defined software attributes and from there on into language specific constructs. QStudio Enterprise maps analyzer output into an extended version of the ISO 9126 software risk assessment standard.

The following higher level attributes are defined according to the ISO 9126 Software Risk Assessment Model.

Reliability: The ability of a software product to keep operating over time without failures that renders the system unusable. Observations generated on reliability indicate a risk that the application will fail during actual use.

Maintainability: The aptitude of the source code to undergo repair and evolution. Observations generated on maintainability indicate that there is a risk that changes are hard to implement.

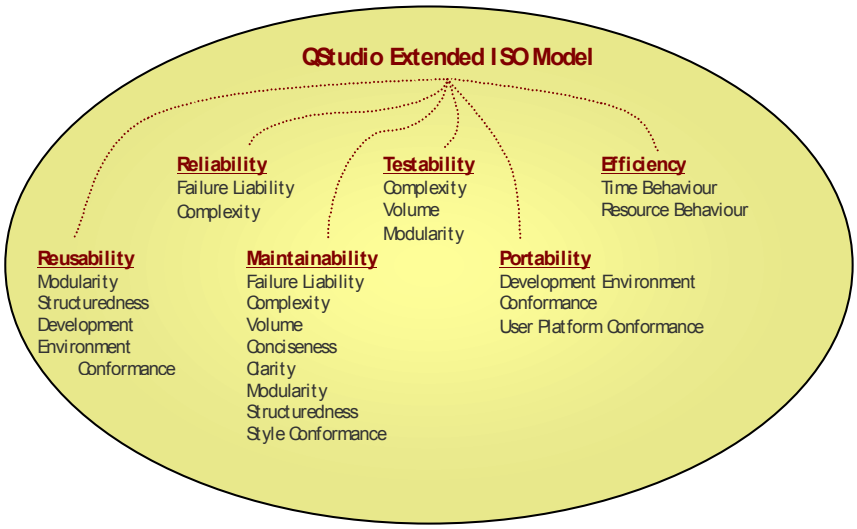
Testability: The amount of test resources needed to reach acceptable test coverage. Observations generated on testability indicate that there is a risk that a great number of test cases might be needed to reach acceptable test coverage.

Re-usability: The suitableness of the source code to be used by a variety of users. Re-use is the practice of using parts of source code that already have been developed. Observations generated on re-usability indicate that there is a risk that re-use is limited or difficult.

Portability: The ability of the source code to be used on various user environments and development environments. Observations generated on Portability indicate risks that the software product cannot be used on specific user platforms or that the source code can't be used "as is" in specific development environments.

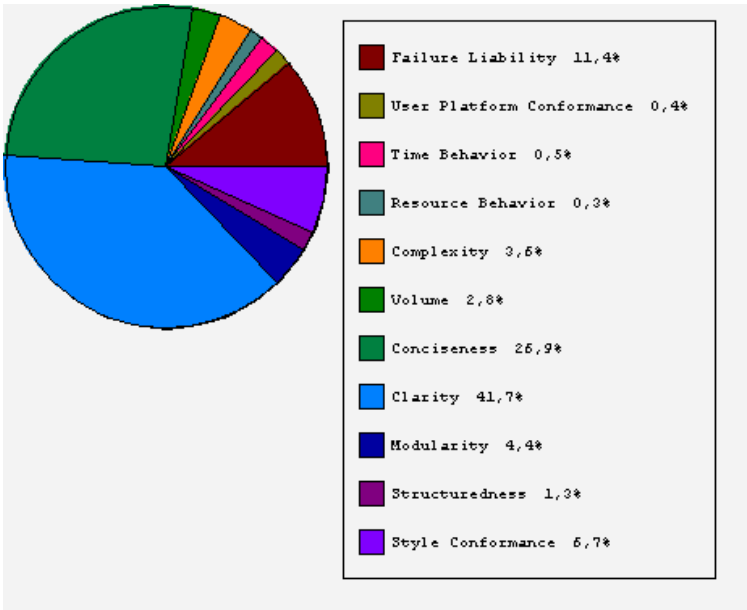
Efficiency: The ability of the software product to perform its functions related to the amount of resources that are used by the application. Observations generated on efficiency indicate that there is a risk that resources are not being used as effectively as possible.

The above attributes are mapped into lower level risk characteristics as displayed below:

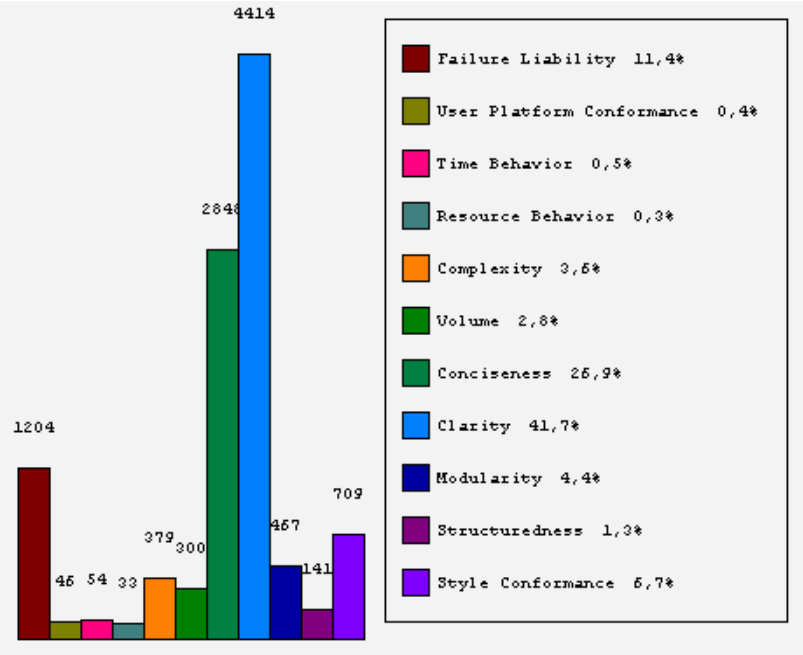


.

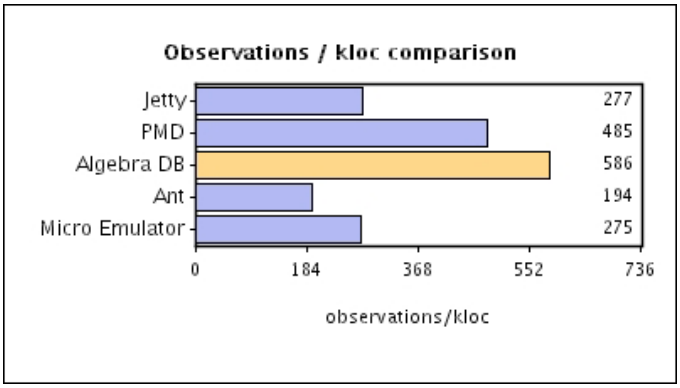
The relative distribution in the source code base of the risk characteristics is shown below. The percentage numbers refers to the number of observations as a percentage of the total observations.



The quantifiable risk in the source code base is shown below. The numbers above the columns refer to the number of times an observation of the relevant type was made in the source code base.

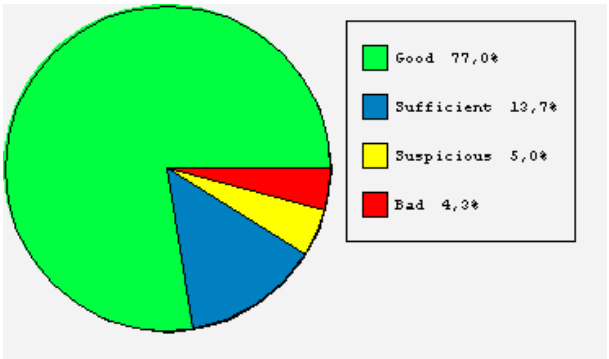


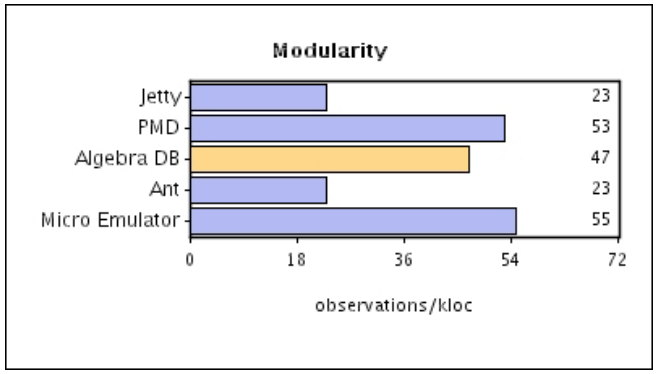
Observations per kloc comparison:



Example Metric - Modularity

The property of source code to be constructed with standardized units for flexibility and variety in use. Modularization of source code is established to facilitate re-use and to assign clearly defined interfaces to software parts. Observations on modularity indicate that the used code construct may reduce the modularity of the source code. The source code base displayed the following risk profile for this characteristic: [77% of the code satisfies the modularity criterion]





Other metrics include:

- Failure Liability/Complexity/Volume/Conciseness
- Clarity/Structuredness/Style Conformance
- Development Environment Conformance
- User Platform Conformance/Time Behaviour
- Resource Behaviour
- Methods per Class/Inheritance depth
- Average over all Method Related Metrics
- Cyclomatic complexity/Nesting depth/Path Count
- Number of lines per method/Returns per method

QStudio Enterprise 2.2 Features	
Automated High Level Management Report Generation	✓
ISO 9126 based Software Risk Assessment Model	✓
Defect Impact Level Assessment	✓
Software Quality Trend Analysis	✓
Software Quality Benchmarking Analysis	✓
Drill down overviews of type Table, Pie Chart and Bar Chart	✓
Automated Annotated Source Code Generation	✓
Application Usage reporting	✓
Per Project Configurable Coding Standard Compliance	✓
Enterprise Reporting Integration	✓
Integratable with Code Management Systems	✓
Command Line Interface	✓
Role dependent and customizable user profiles	✓
Windows 98,2000,NT,XP,ME, Solaris, Linux	✓
Integrates with QStudio for Java Pro Java Code Analyzer (Java)	✓
Integration with FxCop .NET Code Analyzer (Microsoft)	Version 3.0:
Integration with QJ Pro (Java – Open Source)	Version 3.0:
Integration with PMD Code Analyzer (Java – Open Source)	Version 3.0:
Integration with CheckStyle Code Analyzer (Java – Open Source)	Version 3.0:
Integration with Power Software Krakatau Analyzer (C/C++)	Version 3.0:
Integration With Other Analyzers Possible On Request	Version 3.0: