



Application Focused Quality Assurance

QA Systems

The Software Assessment Company™

www.qa-systems.com

**November 2004
Version 1.1**

QA Systems - The Software Assessment Company™ - develops tools and services for application Focused quality assurance.

Introduction

How maintainable is your application? How do you measure it? If you don't measure it how do you know? How reliable is your application? How do you measure it? If you don't measure it how do you know? These are the kinds of questions that Application Focused Quality Assurance (AFQA) deals with.

AFQA focuses on measuring application quality itself through all phases of the application development process and using the results of these measures to steer the application development process itself. AFQA makes use of explicit application quality models (e.g. the ISO 9126 software quality model) to express and measure application quality

Process Focused Quality Assurance (PFQA) methodologies and frameworks such as CMM and ISO 9000 which generally form the largest portion of a companies QA implementation can (and often do) lead to applications with sub-optimal quality. The reason is that PFQA does not directly measure the quality of the application but the quality of the underlying development process. The underlying assumption around PFQA is that the quality of the process will somehow transfer to the quality of the application itself. The truth is that this form of "quality osmosis" is often a false hope. In any case it comes down to a question of common sense: if you are focused on the quality of the application then measure that and not something else (i.e. the quality of the development process).

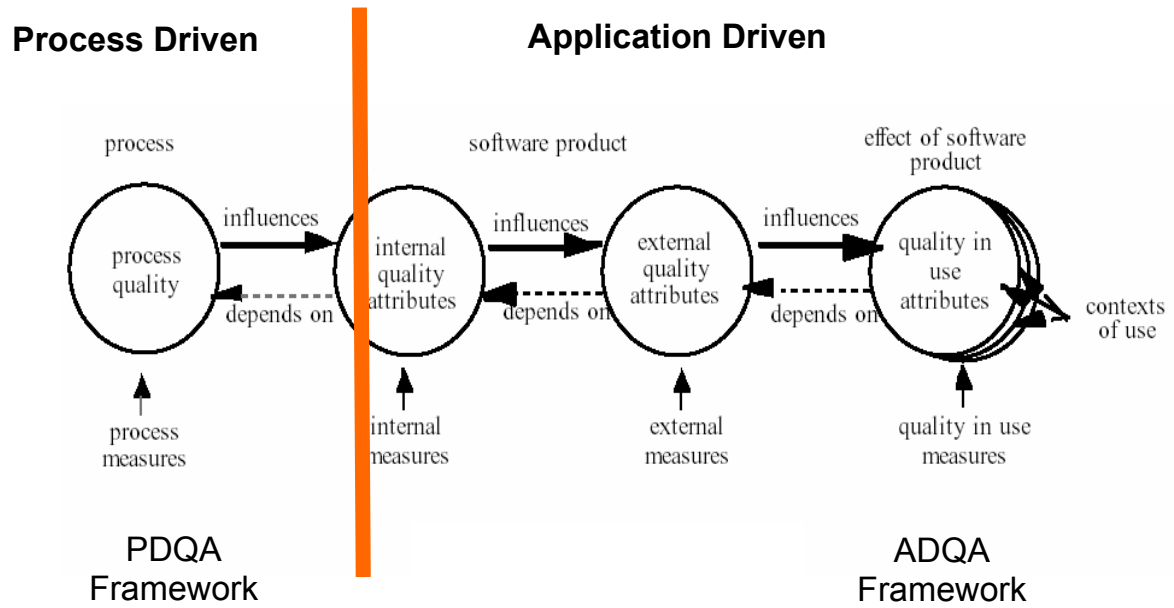
AFQA has advantages over PFQA in a number of areas. These are:

- A deeper and more quantifiable understanding of the required application quality
- A better measurement and assessment of achieved application quality
- Far earlier recognition of application quality problems
- more effective input into the development process

Application Focused QA does *not* replace traditional Process Focused QA but is *complementary* to it. AFQA is *output* oriented and focuses on the *effectivity* of application development (i.e. the right application quality). PFQA is *input* oriented and focuses on the *efficiency* of application development.

An AFQA strategy is particularly crucial in an outsourcing environment. The supplier will usually implement a PFQA approach and the client an AFQA approach. Both are necessary and neglecting either can be costly. The supplier is the natural owner of issues regarding efficiency of the development process. The client is the natural owner of issues regarding the effectivity of the developed application.

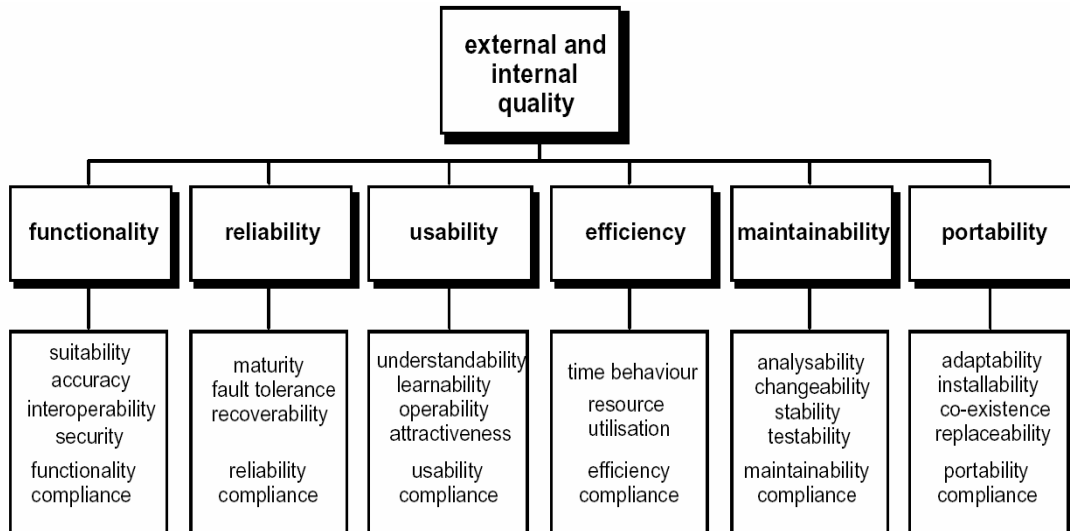
The figure below sketches the positioning of PFQA with respect to AFQA.



Application Quality Model

An essential element of an AFQA approach is the usage of explicit application quality models which provides a way of directly measuring application quality. ISO 9126 is an example of such a model.

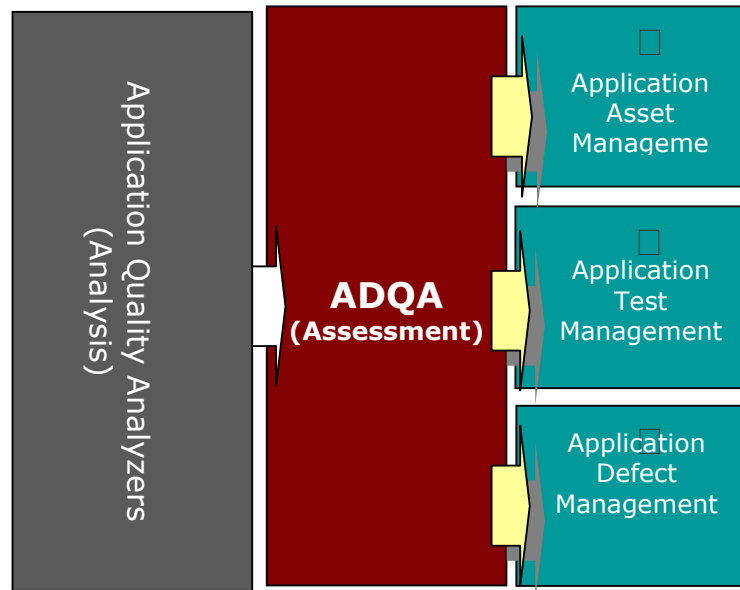
The model defines a stepwise refinement of the notion of application quality into a set of quality attributes and from these a further breakdown into quality sub-attributes.



Quality model for external and internal quality

Another essential aspect is the separation of application quality analysis (measurement) from application quality assessment (interpretation). See figure below. By separating out assessment from analysis and establishing an explicit assessment model several highly useful goals for the organization can be achieved:

- A single coherent assessment model to be used against various application quality analyzers. Each analyzer no longer needs to come with its own (generally incomplete or nonexistent) assessment model
- The assessment model can act as a platform for higher level reporting and management tools thereby providing a single coherent application quality data model. Today software management would need to create and interact with the various data models associated with each analyzer as well as define the mapping to a quality model
- it allows a best of breed approach to analyzer tool selection and reduces customer lock in to specific application quality analyzers



The application quality model stretches across all phases of the development. This allows a coherent assessment model to be put in place. Each phase of the process is assessed to what level it contributes (or detracts) from the required application quality.

Quality Measurement Model

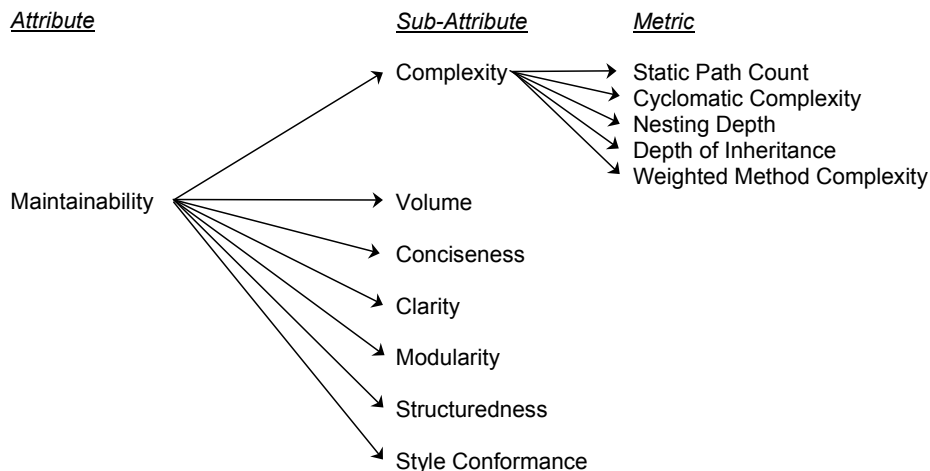
	Activity 1	Activity 2	Activity 3	Activity 4	Activity 5	Activity 6	Activity 7	Activity 8
Phase	Requirement analysis (Software and systems)	Architectural design (Software and systems)	Software detailed design	Software coding and testing	Software integration and software qualification testing	System integration and system qualification testing	Software installation	Software acceptance support
9126 series model reference	Required user quality, Required internal quality, Required external quality	Predicted quality in use, Predicted external quality, Measured internal quality	Predicted quality in use, Predicted external quality, Measured internal quality	Predicted quality in use, Measured external quality, Predicted external quality, Measured internal quality	Predicted quality in use, Measured external quality, Predicted external quality, Measured internal quality	Predicted quality in use, Measured external quality, Measured internal quality	Predicted quality in use, Measured external quality, Measured internal quality	Measured quality in use, Measured external quality, Measured internal quality

ISO 9126, for example, specifies the following application quality attributes:

- **Reliability:** The ability of the application 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 application to undergo repair and evolution. Observations generated on maintainability indicate that changes are hard to implement.

- **Testability:** The amount of test resources needed to reach acceptable test coverage. Observations generated on testability indicate that a great number of test cases might be needed to reach acceptable test coverage.
- **Re-usability:** The suitability of the application, its components and 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 re-use is limited or difficult.
- **Portability:** The ability of the application to be adapted to various user environments and development environments. Observations generated on Portability indicate risks that the application 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 application to perform its functions related to the amount of resources that are used by the application. Observations generated on efficiency indicate that resources can be more effectively used.

The figure below sketches how high level quality concepts are mapped into more detailed quality subattributes and from there on into application quality metrics (below is an example for code metrics). These metrics can be measured by automated tooling. In this way QStudio Enterprise integrates code quality analyzers for example.



Improving the code by following the advice given, it is clear how the code quality improves. It is also clear that by counting the number of (sub-) attribute hits, a detailed quality characteristic can be assessed quantitatively for (a batch of) source files.

Application quality is not just "Freedom of deficiencies" but also "Meeting customer needs" and in turn the customers are defined as "All that are affected by the quality characteristics of the product or service both internally (people who develop the product) as externally (people who use the product or service)".

Quality needs to be specified using objective quantitative indicators. As a result of multiple users having multiple needs, one single indicator cannot express the level of quality. For example, the manager of the software development department would like the software to be easily maintainable to keep down maintenance costs whilst the end user of the software product is interested in usability and performance.

Defect Impact levels are established by distinguishing three stakeholders: the *end user* (the person using the application), the *application development process owner* (the person responsible for the – quality of the - software development process) and the *application developer*. Impact levels indicate to what extent the requirements of each of the stakeholders are satisfied.

QStudio Enterprise defines five impact levels with level 5 being the most severe:

- **Level 1:** This level relates to effectiveness of the application developer. Level 1 observations point to situations where best practices show that there is a better alternative. Modifying the application according to the advice given will make the application (code) more elegant.
- **Level 2:** This level relates to the effectiveness of teamwork. Level 2 observations point to a risk that the application is difficult to understand by peer developers because it is unclear, complex, sparsely documented etc. Modifying according to the advice given will make it easier to (re) use and maintain by others.
- **Level 3:** This level relates to the (cost) effectiveness of the application development process. Level 3 observations point to a risk that the development efforts take longer than expected due to unforeseen problems, mismatches, inconsistencies etc. Modifying according to the advice will lead to more predictable and efficient application development.
- **Level 4:** This level relates to the effectiveness of the user experience. Level 4 observations point to a risk that parts of functionality cannot be used or that basic application features like performance, resource behavior, user friendliness, accuracy, compliance are not within acceptable limits. Modifying according to the advice given will improve the quality of the end user's experience.
- **Level 5:** This relates to effectiveness of the application. Level 5 observations point to a risk that the application may fail. Modifying according to the advice given will improve the reliability of the application.

QStudio Enterprise

QStudio Enterprise is a powerful software quality assessment engine and AFQA framework which structures and automates portions of AFQA. For example, at the application code quality level QStudio Enterprise is capable of integrating with 3rd party code quality analyzers such as QJ-Pro for Java applications QA and Microsoft's FxCop for .NET applications.

QStudio Enterprise supports the ISO 9126 software quality model and an impact defect quality model and automatically generates management oriented software assessment reports. It has the ability to benchmark application quality as well as support trend analysis allowing comparison either with other applications or with previous versions of the same application.

QStudio Enterprise development support capabilities provide the application developer drill down capabilities to all identified issues within the software.

Target users of QStudio Enterprise include application development organizations, application operations organizations, outsourcing managers, software delivery managers and quality managers.

QStudio Enterprise significantly reduces the risk (and ultimately expense) associated with software development, software delivery or software acceptance during insourcing or outsourcing

QStudio Enterprise is based on several ground breaking concepts:

- QStudio Enterprise clearly separates automated analysis from automated assessment: this allows a best of breed approach to code analysis tool selection and hence reduces customer lock in.
- QStudio Enterprise implements the established ISO 9126 software quality standard as the basis for the assessment model. This means that software management can implement a coherent assessment model to be used against various analysis tools (and hence ultimately against all programming languages and application development).. Each tool no longer needs to provide its own (generally incomplete or nonexistent) assessment model
- By providing a single coherent data model based on an established standard. QStudio Enterprise provides a platform for higher level reporting and management utilities and a more consistent approach to application quality reporting. Today software management needs to create and interact with the (implicit) data model associated with each tool as well as determining the mapping to the higher level (more relevant) business risk concepts. A difficult and time consuming task at best.
- QStudio Enterprise is unique in that it can integrate with 3rd party code analyzers .

Use QStudio Enterprise if you need to:

- Software Quality Trend Analysis and Benchmarking: You maintain or develop source code. You need to understand the software quality risk of the source code and how it compares to previous versions of the code bases (trend analysis) or comparable code bases (benchmarking analysis).
- Software Assessment: You carry out software risk assessments for internal or external parties. You need detailed insight and reporting into the potential software problems within the assessed software.
- Software Outsourcing or Insourcing: 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 before accepting the code.
- Software Quality Demonstration: 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.

- **Software Quality Measurement:** You have (or need to have) a measurable software improvement program. QStudio Enterprise provides software quality measurement, software quality benchmarking and software quality trend analysis giving you the ability to institute a measurable software quality improvement program which can tie in with your overall quality program. You need to comply with internal or external software coding or quality standards.
- **Coding Standard Enforcement:** 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
- Integrates with the Microsoft .NET analyzer FxCop

QStudio® Enterprise 2.2 is available 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).