

A TEAM-BASED PROJECT QUALITY MANAGEMENT SYSTEM

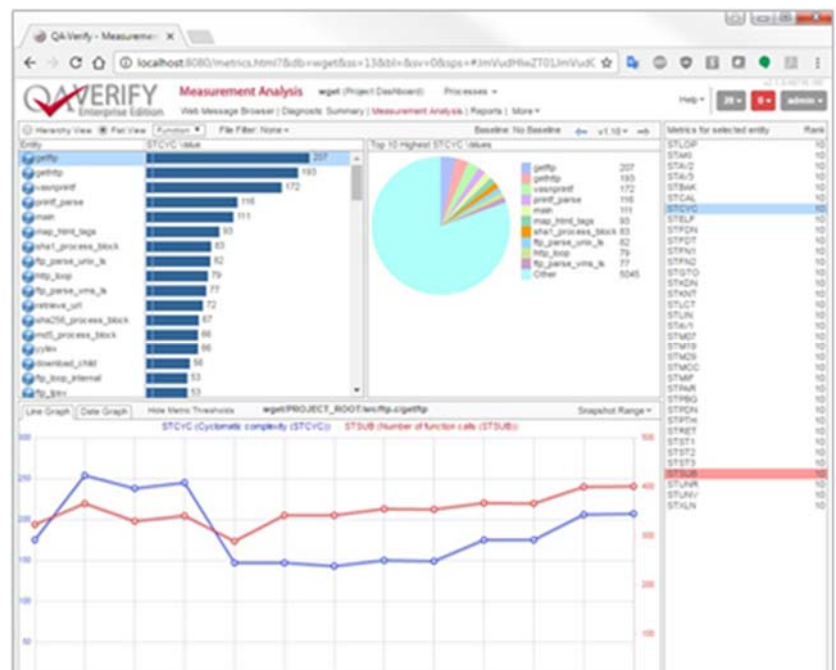
QA-Verify's client-server architecture and web-based interface combines the analysis strength and depth of our QA static analyzers (QA-C and QA-C++) with team-sharing collaboration and broader quality management concepts. This solution appeals to audiences beyond core traditional developers to encompass stakeholders such as development leads, architects, project managers and senior management.

With QA-Verify, team-based collaboration is fully embraced. Developers share a consistent view of source merged with detailed diagnostics for current or any past version of code. Additional comments and actions can be stored against any code artifact for forensic detailed code inspections. Controlled deviations from particular coding rules can be permitted through a sophisticated, flexible system of suppressions.

Baselining provides a mechanism to apply different coding rules to field-proven legacy code versus new code. Integration into the Version Control System gives access to historic snapshots, enabling diagnostic comparison and metric trending across project timelines.

KEY BENEFITS:

- > Share software quality metrics across key stakeholders
- > Perform collaborative code inspections with recorded actions and comments
- > Drill-down from project summary to detailed code diagnostics and metrics
- > Verify compliance to coding standards or best practices
- > Deviate from specific coding rules using a pragmatic, transparent and traceable mechanism
- > Differentiate between legacy and new code and apply different compliance criteria to each
- > Detect and prevent code defects



KEY COMPONENTS:

- > Client-server architecture
- > Detailed views of source code annotated with diagnostics
- > Flexible diagnostic suppression mechanisms coupled to controlled deviation justification
- > Trending and comparative analysis for metrics and diagnostics
- > Extensive & flexible range of language & user-defined metrics supplemented by user-defined compound metrics
- > Integration with all popular Version Control Systems user and project based access control privileges
- > Flexible, user-configurable report generation user-designed workflow integration for diagnostic and code inspection actions & comments



This is easily the best software I have seen to visualize, understand, and address complexity.

TRW Automotive

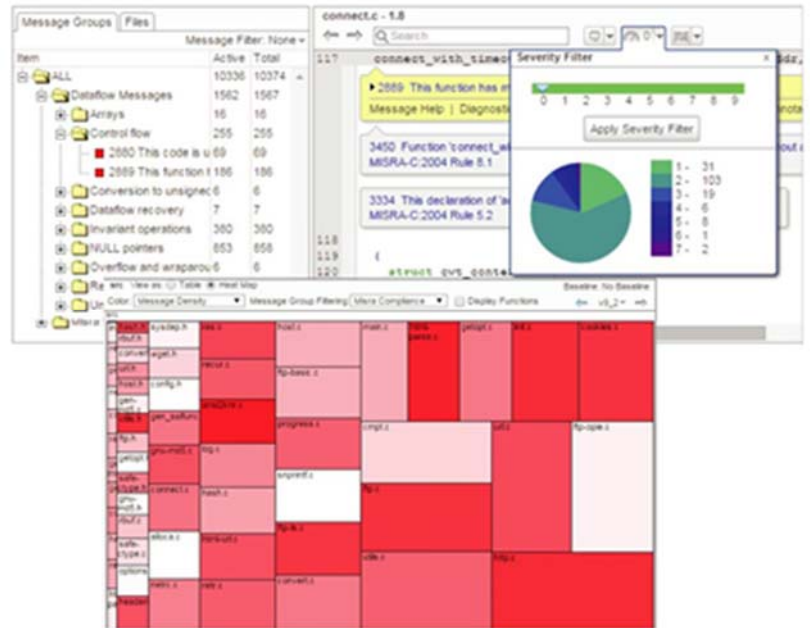


STAKEHOLDER ACCESS

QA·Verify broadens the availability of key quality information beyond the traditional core development team. It is important to recognize that these different stakeholders have very different perspectives and understanding of software quality.

Developers need detailed code-centric compliance focussed on their own subset of the code-base. Project managers and QA professionals need summaries and profiles of quality across their projects. Suppliers and customers often have a contractual need to share a de-tailed profile of code quality.

QA·Verify delivers all these views from the same source data, providing consistent and justifiable quality analytics and meaningful, detailed remediation tasks.



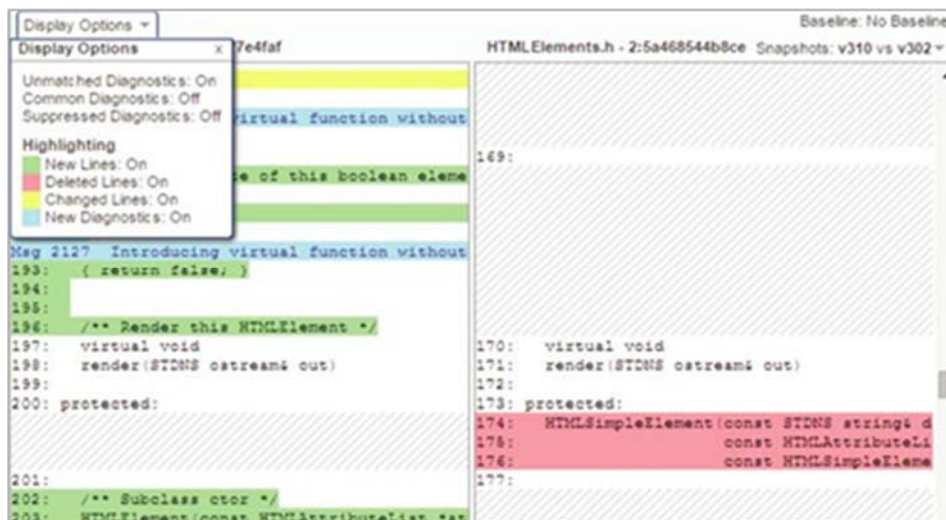
LEGACY CODE MANAGEMENT

Most software projects contain portions of legacy code. While striving for full compliance on new code, many organizations are understandably more cautious when it comes to field-proven legacy code.

QA·Verify offers a project-wide analysis baseline solution that suppresses benign legacy issues and displays only diagnostics in new or changed code. The shared team-centric view of this baseline diagnostic output is also supplemented by more intricate individual developer comparative views.

For immediate and timely attention to the latest code changes, a “New Diagnostic” mode shows only the newly created diagnostics compared to the immediate parent snapshot, providing a clear path to full compliance on each iteration.

One of the most powerful features of New Diagnostic mode is when instant notifications are sent to each developer if they check-in code that breaks a coding rule or raises any other critical issue.





PRAGMATIC COMPLIANCE TO CODING STANDARDS



Full compliance to the entire set of MISRA coding rules is often unachievable. Legitimate reasons for non-compliance include hardware constraints, legacy code rewrite restrictions and conflicting coding rules.

QA·Verify provides a sophisticated system of diagnostic suppressions to man- age this situation. These suppressions can be applied to entire coding rules, across specific files, or to suppress a single diagnostic.

Each suppressed diagnostic or rule is matched to a deviation which documents the rationale and justification for the action. QA·Verify tracks all sup- pressions and supports two sophisticated methods to automatically “carry forward” suppressions into future versions of the code. Additional deviation- suppression coupling comprehensively addresses industry adoption of tightly controlled deviations.

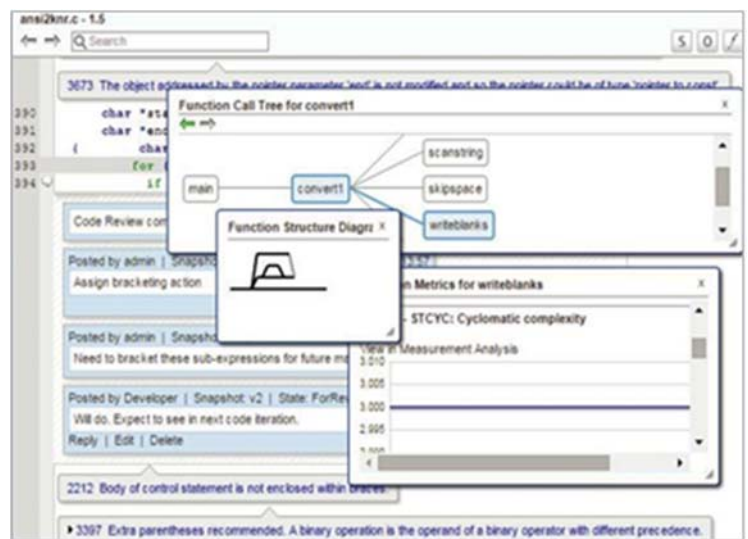
COLLABORATIVE CODE INSPECTIONS

Code inspection, a thorough code-centric examination incorporating automated detection of coding issues and manual review of structure and intent, is known to be one of the most successful ways to eliminate defects early in the development process. QA·Verify supports this critical task with important information and a new annotation capability.

Key functionality to support this activity includes:

- > Display of comprehensive, accurate diagnostics from coding compliance, defect detection, and other code quality objectives
- > File inclusion and function calling trees with definition & call-site information
- > Function structure visual display
- > Metric trend displays for each entity (function, class, file) under observation

QA·Verify’s annotation system is a sequence of user-supplied commentary and action text, which can be applied against a code line or diagnostic and propagated into future code snapshots. The annotation system’s state definition and transition can be separately defined for each project, incorporating user- assignment for actioning of code fixes or as part of a code review.



FLEXIBLE REPORTING

QA·Verify is designed to accommodate a diverse range of user-specific reports providing, for example, evidence of compliance to a coding standard along with a detailed log of deviations; profiles of code quality according to wide-ranging attributes; and overviews of historic metrics trending and outlier values down to file, function and class level.

The key to this flexibility is the availability of a large collection of report components which can be selected, combined and configured to create user-defined composite reports. And these reports can also be run against any snapshot version of a project.



TECHNICAL SPECIFICATIONS

INSTALLATION CONTENTS:

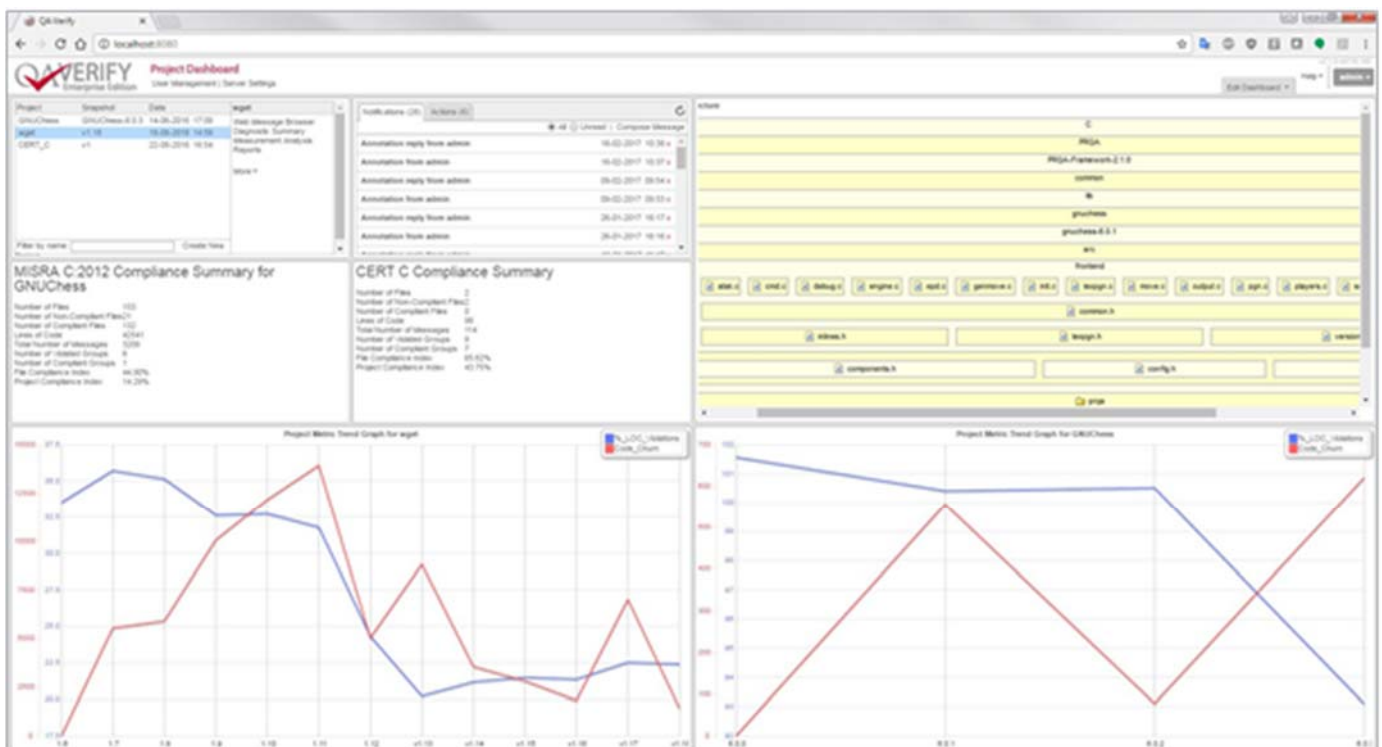
- > Self-contained Web Server with internal
- > SQL database
- > HTTP and HTTPS connections supported
- > Graphical User Interfaces for Project
- > Configuration and Project Upload
- > Command-level integration for automated build environment project upload
- > Windows and Linux 64-bit server platforms supported (compatible to any build platform)
- > Full installation requires 135MB excluding project databases

WEB BROWSER CLIENT:

- > Windows Internet Explorer 11 (all service updates)
- > Mozilla Firefox (recommend latest release)
- > Google Chrome (recommend latest release)

INTEGRATIONS:

- > Version Control Systems: CVS, Sub version, Perforce, Clearcase, PVCS/ Serena, MKS, Synergy, Accurev, Git, Mercurial, Team Foundation Server, (others by request)
- > Build environments: Make Targets, Jenkins, (others by request)
- > Issue Tracking Systems: JIRA for annotation coupling, (others by request)



SGS-TÜV SAAR CERTIFIED

SGS-TÜV Saar has certified QA·C and QA·C++ as “usable in the development of safety related software” for the key safety critical standards, IEC 61508, ISO 26262, EN 50128, IEC 60880 and IEC 62304.





QA Systems and Programming Research Ltd

QA Systems is an authorised reseller of the QA·C / QA·C++, QA·Verify static testing tools and their compliance module add-ons, which are owned by Programming Research Ltd.

QA·C ®, QA·C++ ® and QA·Verify ® are registered trademarks of Programming Research Ltd. These tools and this document are the copyright © 2016 of Programming Research Ltd.

Third party trademarks, logos and trade names appearing in this document are the trademarks and property of their respective owners.

QA·C, QA·C++ and QA·Verify, offer the closest possible examination of C and C++ code. All contain powerful, proprietary parsing engines combined with deep accurate dataflow which deliver high fidelity language analysis and comprehension. They identify problems caused by language usage that is dangerous, overly complex, non-portable or difficult to maintain. Plus, they provide a mechanism for coding standard enforcement.

Contact Us

For further information regarding QA·C, QA·C++ and QA·Verify and compliance module add-ons, please contact QA Systems at info@qa-systems.de.