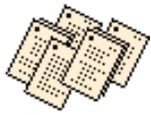
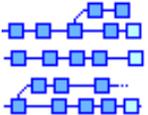


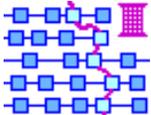
Razor Product Overview



Issues



Versions



Threads

Razor -- Smarter, Faster, and Better Software Development

Software productivity has suffered because demand for new software is increasing faster than our ability to supply it. Increased demand strains an organization's core technical resources and threatens software quality. In fact, the biggest contributors to software failure are lack of coordination among developers and inability to resolve team problems.

Whether your organization has hundreds of developers building millions of lines of code or only a handful of people developing new applications, you need to invest in a resource for effective management of the entire development process. Razor, through its ability to manage changes throughout the software lifecycle, is such an investment. Razor gives you the power to drive and track your software development process.

An integrated solution, Razor delivers problem and issue tracking, version control, and release management for customized software configuration management processes. Razor has helped many clients achieve within a year not only software project excellence by also CMM levels of software quality.

KEY FEATURES

Integrated Tool for Configuration Management

- Integrated Change Management
- Web-based/Remote Issues Tracking
- Integrated Source Code and Version Control
- Issues and Activity-based Release Management
- Parallel and Remote Team Development
- CMM and ISO 9000 Workflow Management Problem Tracking and Reporting
- Extensive Software Metrics Reporting
- Dynamic Baseline Management
- Platform Independence
- Multiple Servers Supported over Distributed Networks
- Scalable, Flexible, and Affordable

COMPETITIVE EDGE

Easy to Use

Razor can be installed, configured, and learned in a few simple steps. Unlike competitive products with complex and varied user interfaces, Razor has one clear and consistent user interface for all functions

across all platforms. Whether Razor is being run on Windows, NT, UNIX, Linux, or Motif, user access and operation are the same.

Configurable to Your Development Practices

Razor can be adapted to meet the needs of your development process, including SEI CMM, ISO 9000, and other business practices. Through a point and click setup, you can customize the user interface, as well as the entire workflow to accommodate your organization's way of doing business. Many competitive tools require users to adhere to rigorous, predefined methodologies that do not take into account your organization's particular workflow. Razor gives you the flexibility to make the tool your own.

Affordable with Full Functionality

Razor is designed to deliver a full set of features at half the price of competing products. A support team that is focused solely on the Razor product backs it. Support personnel are often well versed in the customers' day-to-day configuration management requirements.

Integrated Product

Razor was developed as an integrated product -- not a conglomeration of purchased products. Newest developments include efforts that support the Integrated Development Environments for IBM Visual Age, Microsoft Visual Studio, Microsoft .NET, Rational Rose and PowerBuilder.

Metrics Data

Razor can provide metrics data for project scheduling, requirements engineering, code analysis, defect tracking, and risk management. You can generate reports on status, portion of requirements completed, number of issues per release, number of bugs per component, lines of code, number of source files and so on. The metrics data resides in a repository instantiated in a common RDBMS, so that custom-made reports are possible. You can track which bugs have been fixed in any baseline and automatically compile release notes on a shippable release, including problem/resolution synopsis.

Using Razor Internally

Visible employees use Razor to develop and maintain each release of Razor as a configuration management tool. This results in an unparalleled level of efficiency and has aided the company to maintain product integrity, continuity and accumulated product history and knowledge.

Razor Key Features and Functionality



Issue/Problem Tracking

Through Razor's customizable interface, you define the type of information to be tracked and how it should be displayed. Razor features include:

- Checkboxes, text fields, X-of-many selectors lists, and timestamp fields
- Ability to filter and sort the display
- User-defined rules for signature and e-mail notifications
- Powerful e-mail interface for information entry or query
- Attached triggers to implement a process
- Intuitive graphical reporting

Version Control and Change Management

Razor easily handles both ASCII and binary files. Your team can check files in and out of the system for edits, while automatically tracking, the "who, what, where, and why" of individual efforts. Special features include:

- Drag-n-drop integration with problem-tracking system for bi-directional accountability
- Ability to quickly review the change history of any file in the system
- Graphical merge to support parallel development
- Full access control

Release Management

You can define named file sets for specific product builds, either through file by file commands or broad statements such as "include all C files at the version that passed release testing" or "update my definition based on changes related to a list of issues." These are file-controlled, living file sets that evolve along with the project. Special features include:

- Complete control and automation over the build process
- Comparison of any two releases
- Ability to easily recreate any known build.

Web Interface

Issue Weaver[®] allows you to access issues through a Web interface. You may permit authorized users all over the world to access project issues or restrict access only to users on a certain basis – be it on a geographic, project or priority basis. Issue Weaver offers options for configuring the display of an issue form that is not available in the issues tool and you can also add your own HTML to the form to change colors, display company logos, or add other elements.

Workflow Support

Razor is designed to manage changes in all phases of the software development lifecycle. It facilitates a component-based software engineering approach that accommodates many small incremental

revisions and multiple dependencies between software projects. By grouping project tasks you can manage projects by feature requests or bug reports. And, as your software engineering process changes, you can adapt the definition and customization of workflow through Razor's rule base and unique scripting capabilities.

Project Management & Reporting

Razor allows you to relate units of work to software engineering resources. With metrics software, you are able to more effectively measure and allocate your project resources – saving your company both time and money. All metrics data resides in an open repository so you can run customized reports. You can track which bugs have been fixed in any baseline and automatically compile release notes on a shippable release, including problem/resolution synopsis.

Collaborative Change Management

Razor's version control system enables parallel development and merging of source code. When parallel development (multiple engineers working on the same code base at the same time) is the default way of working, Razor will notify specified functionaries when check-in conflicts occur.

Requirements Tracking

Razor integrates requirements definition so all check-ins and projects can be related to requirements. You can easily track requirements in the Razor issues tool. Each requirement, bug report, or change request is entered in terms of an issue and assigned a unique issue number. This issue number can then be associated with files under version control.

Project Integrity

Changes to entities in Razor's environment (source code, designs, states, projects, test scripts, documentation) are reversible at any time.

Depending on the state of a unit of work, an entity can be reverted back to a previous state by a current owner (peer reviewer during coding, QA engineer during testing, build manager prior to build).

Extended Version Management

Razor supports multiple projects by providing support for multiple source trees. Razor not only enables source code version management, it also manages other documents, such as project plans, designs (both text and diagrams), test scripts, and related documentation.

Custom Development Environment

Razor does not restrict the development environment used by individual engineers. Engineers can pick their own compilers, debugging tools, and editors.