Rational Dashboard

Automated, Web-based Metrics Collection & Analysis
September 1999
1 Introduction

1.1 Dashboard Overview

Rational’s Dashboard provides a graphical means to viewing large-scale software projects. The Dashboard is intended to provide critical information, i.e., metrics, regarding a project’s status. This includes providing graphical indicators, gauges, dials, graphs, as well as raw data to help identify slowdowns, breakdowns, and potential risk areas in a project.

The intended use of the Dashboard is to provide a view of metrics data for a software project. The CMM/SEI [1] stated four reasons for measuring software processes, products, and resources:

- To characterize – to gain understanding of processes, products, resources, and environments, and to establish baselines for comparisons with future assessments.
- To evaluate – to determine status with respect to plans; to determine if projects are drifting off track.
- To predict – to gain understanding so values that are observed can be used to predict others (e.g., cost, schedule, and quality).
- To improve – to help identify roadblocks, root causes, inefficiencies, and other opportunities for improving product quality and process performance.

A successful metrics collection program must also have the following characteristics:

- Metrics must be simple, object, easy to collect, easy to interpret, and hard to misinterpret.
- Metrics collection must be automated and unobtrusive, i.e., not interfere with the activities of developers.

The Dashboard consists of the following components (shown in Figure 1):

1. An Internet browser-based graphical user interface to display project status via gauges, dials, graphs, and indicators for portability across heterogeneous platforms.
2. An administrative tool to define and build metric hierarchical views, define metric artifacts, input metric data via CSV, and input metric data manually.
3. A database server which utilizes Microsoft SQL Server or Microsoft Access for storing historical data.
4. Agents that collect metrics data to populate the database from the following tools:
   a) Rational Rose®
   b) Rational RequisitePro
   c) Rational ClearQuest ™
   d) Rational TeamTest/SQA
1.2 User Role Description

1.2.1 Introduction

In a typical project, the users of the Dashboard may be interested in a set of metrics. These sets of metrics can be thought of as User Role Views.

1.2.2 User Role Profiles

Users of the Dashboard tool will typically be various levels of management:

- the Program Manager
- the Software Manager
- the Test Manager
- the Configuration Manager
- the QA Manager

The Program Manager may wish to track the following:

- Requirements progress (baseline percent complete versus plan)
- Action Item progress (open versus closed)
- Design progress (model percent complete versus plan, use case progress)
- Implementation progress (unit coding/standalone test percent complete versus plan)
Deployment progress (formal testing percent complete versus plan)

The Test Manager may wish to track the following:
- Number of Change Requests (CRs) opened
- Number of CRs or DRs closed
- CR closure rate trends
- CR profile by type

The Software Development Manager may wish to track the following:
- Number of CRs opened
- Number of CRs closed
- CR closure rate trends
- CR profile by type
- Requirements progress
- Design progress
- Implementation progress
- Deployment progress

In addition, individual software team members may also find the tool useful in managing personal tasks:
- Number of CRs assigned
- Number of CRs implemented
- Design progress
- Implementation progress

2 Dashboard Client

The Dashboard Client is a Java applet loaded from the Dashboard web page. This applet takes full advantage of many of the Java version 1.1 features. This requires that user's use a JDK 1.1.5 compliant web browser (e.g., Netscape 4.0.5 or above, AWT 1.1.5 release). To run the Dashboard client from your web browser, simply enter the URL of the Dashboard web page in your browser (you need to know the name of the WEB server hosting the Dashboard web server files).

The Dashboard client’s display consists of a toolbar along the top, with a tabbed panel area on the bottom. The tabbed panel area can be configured to contain multiple tabbed panels, each with multiple metric display widgets. Once configured, the tabbed panel configuration can be saved offline. When the Dashboard Applet initialized, it will automatically attempt to load the default panel configuration for the current user. If one does not exist, then a default panel configuration shared by the project will be loaded.

The Dashboard client currently supports the following display widgets:

| Artifact Table | The artifact table widget can be used to display a table of artifact values from a selected metric hierarchy. |
| Counter | The counter widget can be used to display the numeric value of an artifact from a selected metric hierarchy element. The counter can also display an arithmetic computation on an artifact value or multiple artifact values. |
| Chart | The chart widget can be used to display metric values and trends. Data can be displayed either by day, month, or year, and the chart can be configured as a line, bar, or pie chart. Multiple data sets can be overlaid on a single chart. These data sets can be either actual/plan/threshold data, or actual values selected from multiple hierarchy elements (useful for comparing trends of multiple artifacts). |
| Gauge | The gauge widget can be used to display a single metric value on a "speedometer" |

Rational Software ©1999
style gauge. If the metric being displayed has planned and threshold data entered into the database, the region between the high and low threshold will be colored green. The gauge can also display an arithmetic computation on an artifact value or multiple artifact values.

Indicator

The indicator widget can be used to display a graphical indication of the status of a metric artifact. The status is displayed as a graphical "traffic light". The red/green/yellow values are derived using the artifacts actual, planned, and threshold data. The percentage over/under threshold that causes the indicator to change color is configurable in the Indicator's property dialog.

Label

A label can be placed anywhere on a tab panel to add comments to the displayed widgets.

Container

The Container widget can be used to group and label a set of metric widgets.

Metric information can be summarized following a user-defined linear structure (for example, lines-of-code can be summarized by unit, subsystem, and project). The project is the top qualifier for all data belonging to a set (top-level context). This structure can be thought of as a hierarchical tree. The hierarchical trees can be dynamically constructed by the Dashboard agents or the CSV Tool, or they can be manually constructed using the Admin Tool. For example, the Dashboard agent for Rational Rose can automatically define a tree based on the model hierarchy. The project or model name can be defined as the top-level context. Under the project could be the Use Case View or the Logical View. Under the view could be either Categories (Use Case or Class), use cases, or classes.

When configuring one of the Dashboard widgets, the data source is defined by selecting node(s) from a metric hierarchy tree and an artifact(s) currently associated with the node(s).

3 Dashboard Agents

Dashboard agents are used to automatically and periodically collect metric data from various sources. Currently, the following agents are delivered "out of the box" with the Dashboard:

3.1 Dashboard SoDA Agent

The Rational SoDA tool provides a mechanism to generate documents from data extracted from one or more Rational products (e.g., Rose, ReqPro, etc.). SoDA also provides application specific "domains" that can be accessed via an API to programmatically extract data from the different Rational products. The Dashboard SoDA Agent utilizes these APIs to collect data from any of the SoDA supported Rational products. As new SoDA domains are developed, the Dashboard SoDA Agent can be easily updated to collect metrics for new domains. Currently, SoDA provides domains for the following Rational products:

- Rational Rose ® (versions 4.x and 98)
- Rational RequisitePro ® (version 3.1)
- Rational SQAITeamTest ® (version 5.1 or later)
- Rational ClearQuest ® (version 1.0/1.1)

The metrics collected by the Dashboard SoDA Agent are highly configurable through a standard "ini" file. In general, if an artifact can be collected using SoDA, the Dashboard SoDA Agent can extract it.

Rational Software ©1999
Appendix A contains an extensive list of metrics that could be collected from each supported Rational products.

### 3.2 Dashboard File Agent

The Dashboard File Agent collects metrics on file system elements. Currently, information on when a file has been modified as well as a count of source lines of code can be collected. The SLOC counter is not intended to be used as an out-of-the-box source code complexity analyzer. Rather it is intended to be a starting point for customers to create their own custom agents.

The Dashboard File Agent is configurable through a standard "ini" file. This file specifies information such as the path to the files being processed, as well as whether to collect SLOC counts or modification date information.

### 4 Dashboard CSV Tool

The Dashboard CSV Tool provides a command-line interface for importing and exporting information into/out of the Dashboard database based on comma-separated-value (CSV) format. The CSV Tool provides the following functionality:

- Import metric data
- Export metric data
- Import hierarchy information

#### 4.1 CSV File Format

The CSV format for an input file contains the following:

- The first line in the file must contain the column headers. Example:

  Project, Hierarchy, Threshold, Artifact, State, Index, Event, Date, Weight, State, Delta, Metric

- The rest of the file contains comma-separated values. Example:

  Package6, "Package6: ReqPro: NI NFR: Node UC: NI NFR Verify CPNS Parameter reception", Actual, Documents, 0, 9/10/98, 0, 0, 1
  Package6, "Package6: ReqPro: NI NFR: NIC Document: NI-NFR Intersystem Page Invoke Msg", Actual, Messages, 0, 9/24/98, 0, 0, 32
  Package6, "Package6: ReqPro: SME: Non-Func Req: SME: NFR 4", Actual, Requirements, 0, 9/1/98, 0, 0, 1
  Package6, "Package6: ReqPro: SME: Non-Func Req: SME: NFR 4", Actual, Review State: NodDoF TC Rev Ready, 1, 9/1/98, 0, 0, 1
  Package6, "Package6: ReqPro: SME: Non-Func Req: SME - New Priv Param in HANDMREQ", Actual, Documents, 0, 9/1/98, 0, 0, 1

### 5 Dashboard Administration Tool

The Dashboard Admin Tool provides a graphical interface for administration type functions. The following functions can be performed using the Admin Tool:

- Manually enter data – User can specify metric values for a given node in a hierarchical view
- Import data – User can import data from a comma-separated-values file
- Export data – User can export data to a file in a comma-separated-values format
- Query data – User can query data
- Discard data – User can query and delete data from the Dashboard database
- Archive data – User and query and save data to a file in a comma-separated-values format
- Thin data – User can selectively query and delete data

6 Dashboard Server

The Dashboard Server provides the necessary database interface for the Dashboard Client, Dashboard Agents, Dashboard Administration Tool, and the Dashboard CSV Tool to store and retrieve metric information. The Dashboard Server must execute on the same physical machine where the Dashboard Client Applet is loaded. Currently, support is provided for the Microsoft Access 97 and SQL Server 6.5 databases.

7 References

Appendix A

The following sections contain an extensive list of “out-of-the-box” metrics that the Dashboard could possibly collect from each supported tool. Please note that these lists may not be the complete list of all possible metrics.

A list of “recommended” metrics is also included. Please note this list is very subjective and could change from project to project as well as from customer to customer.

1 RequisitePro Metrics

Extended list of metrics that could be collected from Rational RequisitePro:

- Number of Stakeholder Needs
  - Number of Stakeholder Needs for a specific “Priority” level
  - Number of Stakeholder Needs for a specific “Difficulty” level
  - Number of Stakeholder Needs for a specific “Risk” level
  - Number of Stakeholder Needs for a specific “Stability” level
  - Number of Stakeholder Needs for a specific “Origin”
- Number of Vision Requirements
  - Number of Vision Requirements for a specific “Priority” level
  - Number of Vision Requirements for a specific “Difficulty” level
  - Number of Vision Requirements for a specific “Risk” level
  - Number of Vision Requirements for a specific “Status” level
- Number of Product Requirements
  - Number of Product Requirements for a specific “Priority” level
  - Number of Product Requirements for a specific “Difficulty” level
  - Number of Product Requirements for a specific “Risk” level
  - Number of Product Requirements for a specific “Status” level
  - Number of Product Requirements for a specific “Stability” level
- Number of Glossary documents
  - Number of Glossary Terms for a specific Glossary
- Number of Use Case Actors
- Number of Use Cases
  - Number of Use Cases for a specific “Priority” level
  - Number of Use Cases for a specific “Difficulty” level
  - Number of Use Cases for a specific “Risk” level
  - Number of Use Cases for a specific “Stability” level
  - Number of Use Cases for a specific “Iteration Build”
  - Number of Use Cases for a specific “Status” level
  - Number of Use Cases for a specific “Assigned To” designator
- Number of steps in the Flow of Events for a specific Use Case
  - Number of steps in the Flow of Events for a specific Use Case, at a specific “Difficulty” level
  - Number of steps in the Flow of Events for a specific Use Case, at a specific “Status” level
- Number of Alternative Flows of Events for a specific Use Case
- Number of Pre-Conditions for a specific Use Case
- Number of Post-Conditions for a specific Use Case
- Number of Supplementary Requirements Specification documents
Number of Supplementary Requirements for a specific Requirement Type (e.g., Functionality, Usability, Reliability, Performance, Supportability, Design Constraints, etc.)

Number of Supplementary Requirements for a specific “Priority” level

Number of Supplementary Requirements for a specific “Difficulty” level

Number of Supplementary Requirements for a specific “Risk” level

Number of Supplementary Requirements for a specific “Safety/Criticality” level

Number of Supplementary Requirements for a specific “Status” level

Number of Supplementary Requirements for a specific “Stability” level

Number of Supplementary Requirements for a specific “Iteration Build”

Number of Supplementary Requirements for a specific “Assigned To” designator

Number of Test Specifications

Number of Test Requirements for a specific Test Specification

Number of Test Requirements for a specific Test Specification, at a specific “Priority” level

Number of Test Requirements for a specific Test Specification, for a specific “Iteration Build”

Number of Test Requirements for a specific Test Specification, at a specific “Status” level

Number of Test Requirements for a specific Test Specification, for a specific “Assigned To” designator

Recommended short-list of metrics collected from Rational RequisitePro

- Number of Requirements (Product, S/W, Test) per Status level
- Number of Documents (Use Case, others) per Status level

2 Rose Metrics

Extended list of metrics that can be collected from Rational Rose

- Number of Packages or Subsystems
- Number of Use Case Diagrams
- Number of Use Cases
- Number of Concrete Use Cases
- Number of Abstract Use Cases
- Number of Actors
- Number of Class Diagrams
- Number of Classes
- Number of Abstract Classes
- Number of Classes based on Export Control (Public, Private, Protected, Implementation)
- Number of Classes based on Persistence (Persistent or Transient)
- Number of Classes based on Concurrency
- Number of Sequence Diagrams
- Number of Collaboration Diagrams
- Number of State Diagrams
- Number of Deployment Diagrams
- Number of Component Diagrams
- Number of Associations
- Number of Relationships
- Number of Stereotypes
- Number of Messages

- Number of Use Cases per Package
- Number of Actors per Package

Rational Software ©1999
- Number of Classes per Package
- Number of Class Diagrams per Package
- Number of Sub Packages per Package
- Number of Scenarios per Package
- Number of Associations per Package
- Number of Sequence Diagrams per Use Case
- Number of Collaboration Diagrams per Use Case
- Number of State Diagrams per Use Case
- Number of Use Case Diagrams per Use Case
- Number of Associations per Use Case
- Number of Attributes per Actor
- Number of Operations per Actor
- Number of Attributes per Class
- Number of Operations per Class
- Number of Subclasses per Class
- Number of Classes based on User-defined or Predefined Model Property setting
- Number of Classes based on Language specific Model Property setting
- Number of Derived Attributes per Class
- Number of Attributes per Class
- Number of Attributes based on Export Control (Public, Protected, Private, Implementation)
- Number of Attributes based on Containment (Value, Reference, Unspecified)
- Number of Static Attributes
- Number of Derived Attributes
- Number of Attributes based on User-defined or Predefined Model Property setting
- Number of Attributes based on Language specific Model Property setting
- Number of Operations based on Export Control (Public, Protected, Private, Implementation)
- Number of Operations based on Concurrency (Sequential, Guarded, Synchronous)
- Number of Operations based on User-defined or Predefined Model Property setting
- Number of Operations based on Language specific Model Property setting
- Number of Parameters per Operation
- Number of Messages per Interaction Diagram
- Number of Objects per Interaction Diagram
- Number of Packages per Component Diagram
- Number of Components per Component Package
- Number of Component Diagrams per Component Package
- Number of Processors per Deployment Diagram
- Number of Devices per Deployment Diagram

**Recommended short-list of metrics collected from Rational Rose**

- Number of Classes (per package)
- Number of Attributes per class (per package)
- Number of Operations per class (per package)
- Number of Use Cases (per package)
3 ClearQuest Metrics

Extended list of metrics that can be collected from Rational ClearQuest:

- Number of defects per Assignee
- Number of defects per Build
- Number of defects per Call Tracking Id
- Number of defects per Component
- Number of defects per Detection Method
- Number of defects per Detection Phase
- Number of defects per Failure Type
- Number of defects per Kind
- Number of defects per OS (Server and Client)
- Number of defects per Release Fixed (Actual and Planned)
- Number of defects per Priority
- Number of defects per Problem Type
- Number of defects per Project
- Number of defects per Resolution Type
- Number of defects per Severity
- Number of defects per Site
- Number of defects per Submitter
- Number of defects per State
- Number of defects per Verification Method
- Attachments per Defect
- Detection Effort per Defect
- Estimated Fix Effort per Defect
- Actual Resolution Effort per Defect

Recommended short-list of metrics collected from Rational ClearQuest:

- Defects per State by Severity by Release by Component
- Comparison of States (e.g., Open versus Closed) by Severity by Release by Component
- Estimated Fix Effort per Defect by Severity by Release by Component
- Actual Fix Effort per Defect by Severity by Release by Component

4 TeamTest Metrics

Extended list of metrics that can be collected from TeamTest:

- Number of Builds
  - Number of Builds per Owner
  - Number of Builds per Project
  - Number of Builds based on Status
- Number of Scripts
  - Number of Scripts per Owner
  - Number of Scripts per Project

Rational Software ©1999
- Number of Scripts per Build
- Number of Test Documents
- Number of Test Requirements
- Number of Users
- Number of Computers

**Recommended short-list of metrics collected from Rational TeamTest:**

- Number of Test Requirements
- Number of Scripts per Build