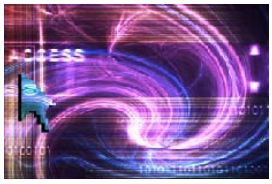
Software Quality Metrics and IB -ARM

The Need



Most organizations measure the quality of their core products and services. However, often no comparable measures of the organization's software quality are maintained. Some key quality metrics are maintained by most IT shops such as system availability, operational problems / failures and project completion statistics. However, IT organizations typically lack the resources to dig deeper into these metrics to find the root causes of the problems. Instead, fixes are installed, applications are patched and the immediate problem or emergency is averted. In some cases, procedures may be put in place to avoid the problem happening again.

But this is a very reactive approach to managing software. What is needed is a pro-active way to receive up-to-date software quality metrics and then to take action based on those metrics.

Software Quality Challenges

The measurement of quality requires the collection and analysis of quantitative information, usually stated in terms of metrics. The value of a metrics program is to measure the quality of software products being produced, to determine areas where improvements are required and ultimately to measure whether the improvement programs are having the desired effect.

Software quality metrics focus on the quality aspects of the product, process, and project. These metrics can be divided further into end-product quality metrics and in-process quality metrics. The essence of software quality engineering can be described in the following two stages:
To investigate the relationships among in-process metrics, project characteristics, and end-product quality.
Based on the findings, to engineer improvements in both process and product quality.

Moreover, quality should be viewed from the entire software life-cycle perspective, not simply as a check on the accuracy and performance of the developed code. Software quality metrics. should also include metrics that measure the quality of the maintenance process.

Benefits of using IB-ARM

IB-ARM provides a proactive method of collecting key software quality measures and distributing these measures throughout the IT organization via a web-based portal. In addition, these measures can be refreshed on a continuous basis as applications are modified and additional issues are encountered.

Problem Analysis

Traditionally, quality involves measuring the problems (also called bugs, defects, failures) encountered when using software. These problems can occur in several categories including:

- Internal user problems
- External client problems
- Operations problems.

Statistics for these problem categories described above are usually recorded in a variety of tools and methods. Client / user problems are typically gathered by Help Desks or customer services representatives. Operations problems are typically collected by operations staff and relate to systems failures and outages. These problems are recorded using both custom in-house tools and commercial tools.

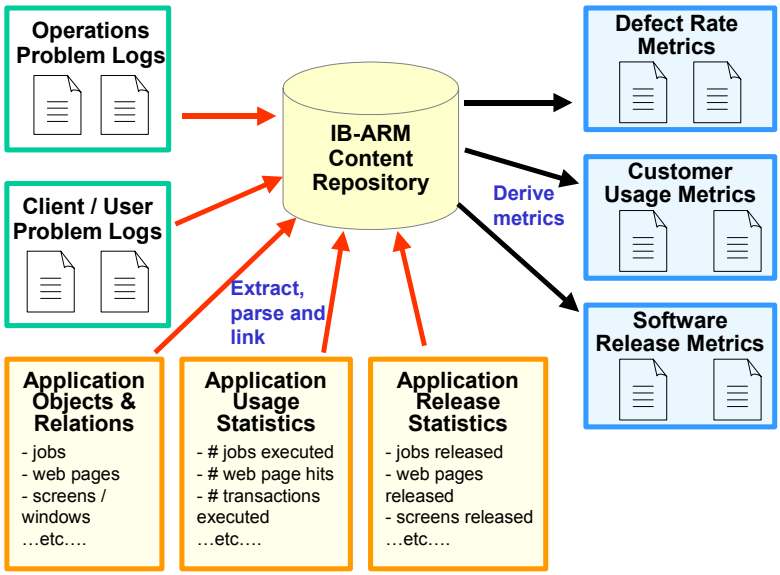
IB-ARM can parse these problem logs and relate the problems back to the underlying application and system objects.

Linking to Application Software Objects

While problem statistics often provide a base understanding of the number of problems that have occurred, deeper understanding of the causes of those problems requires that the problems be linked back to other information. Linking the problem logs to:

- Application objects such as jobs, web pages, screens, transactions, etc. can provide information about error-prone application components.
- Application releases can determine if problems are being introduced by software releases.
- Application usage statistics can determine the number of problems based on the amount of usage of the application objects.

The better understanding provided by IB-ARM ensures that action plans addressing these problems are more focused and effective.



This diagram shows potential linkages between the reported problems and the other application-related information.

Production Metrics

End-product quality is usually measured by the number of "bugs" (functional defects) in the software or by how long the software can run before a critical failure (a "crash.") The previous diagram showed application-based software quality metrics being derived based on linkages between production problem logs and application object information. Several end-product or production software quality metrics can be derived as follows:

- Defects
  - Number of defects
  - Number of defects for each object, subsystem or system.
- Defect Rates
  - Number of defects / Size of application *
    * Size of the application can be measured by KLOC, use cases or function points.
- Customer Usage
  - Number of problems reported by customers for a time period / Total usage during the period **
    ** Usage can be measured in web page hits, online transactions, number of users or licenses.
- Software Releases
  - Number of defects reported for each software release.
- Software Maintenance Quality
  - Backlog management index: Number of problems closed during the month / number of problems added during the month
  - Fix responsiveness: Mean time, from open to closed, for all problems
  - Fix quality: Percentage of all fixes that were defective during a month

Test Coverage metrics

IB-ARM can also link to test cases stored in automated test tools, Excel spreadsheets or Word documents. These test cases can then be linked to application objects to understand the test coverage for objects such as web pages, windows, screens and batch jobs.

In-process metrics

One of the goals of a metrics program is to understand the development process and to learn to engineer quality into that process. In-process quality metrics play an important role. However, in-process quality metrics are usually less formally defined than end-product metrics, and their use varies greatly among software developers. Some of these metrics are listed below:

- Defect metrics (recorded in a bug tracking tool)
  - Defect Arrival Rate: Number of defects reported per week
  - Defect Fix Rate: Number of defects open (indicates quality and efficiency of bug fixing)
- In-Process Quality during Development (derived by linking bugs to application objects)
  - Defect Density: Number of defects / Size of application
  - Source of Defects: Number of Defects originating in each phase
  - Requirements Quality: Number of Defects originating in Requirements phase / # of Use Cases

These metrics can be correlated with the production or end-product metrics described above to understand how development practices affect the resulting production software.

The Process

IB-ARM forms the backbone of a proactive software quality metrics and quality improvement program. The following outlines the process for utilizing IB-ARM in a quality improvement program:

1) Planning the Metrics to track

The IT organization first determines the areas where improvements are required and then selects candidate metrics that will demonstrate the quality characteristics of those areas. The IT organization should identify and determine the level at which acceptable quality is attained for those areas.

2) Track the Metrics using IB-ARM

Determine the source data for the metrics and the relationships within the data that will be tracked. IB-ARM's collectors and parsers will extract the required information and linkages and place them in the IB-ARM repository for analysis purposes.

3) Check the Results

Using IB-ARM's web portal, metrics can be viewed and analysed by all the appropriate groups within the organization. Progress can then be assessed and compared to the acceptable quality levels.

4) Put improvement plans in place

Based on the results, action can now be taken to eliminate problems or to avoid problems that have not yet occurred. Improvement actions can include changing processes, installing additional tools or re-factoring error-prone software objects.

IB-ARM can form the heart of a continuous quality improvement program. Additional metrics can be defined within IB-ARM on a continuous basis as improvement concentrates on other aspects of the IT organization.

About Information Balance, Inc.

Information Balance, Inc. was founded in 1988 to provide consulting and training services to the IT industry. Over the years and through continuous growth, the company's field of expertise has significantly broadened and now includes all aspects of Systems Development and Integration, covering all platforms including mainframe, client/server and the Internet. Many of these areas are supported by formal training curricula.

Today, Information Balance, Inc. is a well-established firm with offices in Canada and Europe. Sustainable, controlled growth has been the mainstay of Information Balance's success, with over 30% growth year over year. Information Balance has been awarded the following accolades supporting its corporate excellence.

- Financial Post Fast 50 ('98,'99,'00)
- Profit Top 100 ('99,'00)
- Andersen 50 Best Privately Managed ('99, Regional Finalist)

Information Balance maintains on-going professional relationships with many large Fortune 500 corporations and government organizations.

For more information, please call 416-962-5235, e-mail infobal@infobal.com, or visit our website at www.infobal.com.