Analyzing Metrics

A single metric rarely gives enough information to understand the trend, so it is important to analyze and cross-correlate metric results to help understand trends. You should use both raw data and combinations of data that you collect. Raw data includes requirements changes. Combination data includes the average time to fix defects, by severity, by subsystem, by time of defect insertion (to help define defect correction activity).

Analysis tools such as Pareto Analysis help determine root causes. A common practice to get to the root cause is to ask "why" five times.

Be sure to provide feedback to management and developers.

The software project manager has primary responsibility to:

- Develop plans for metrics collection and analysis,
- Document these plans in the software development plan, and
- Ensure that project metric data is consolidated and analyzed, and metrics reports are prepared.

The software project manager may designate a project metrics coordinator responsible for collecting, analyzing, and reporting metrics. This is not generally a full-time position and requires an engineer who is well-versed in the process model. He/she should consider using a member of the Software Engineering Process Group (SEPG) on several projects as the metrics coordinator.

Other involved project personnel are development and test engineers who update estimates, report when tasks are complete, and generate Software Problem Reports (SP) and Change Reports (CR); inspection moderators who report data from inspections; and Configuration Management (CM) personnel who manage the SP/CR system.

Implementing a Metrics Program

When implementing a metrics program, it is important to define the link between the project and the metrics that will be collected. There are five parts to ensuring a successful link:

- Project Identification
- Software Development Process
- Software Application Description
- Software Management Process
- Software Development Environment

Software Application Description

• Development type (new system, enhancement, etc.)

• Expected complexity

• Architecture

• Application type (Real-time, embedded, Web I/F)
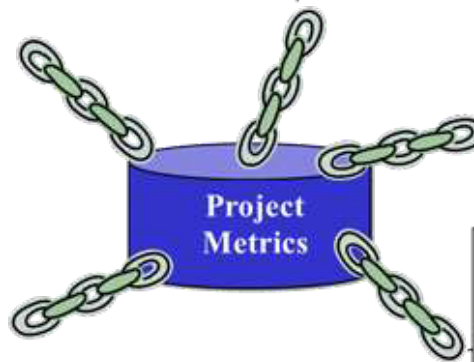
• Security requirements

Project identification

• Project name

• Software project manager

• Procuring agency

• Contract type (e.g., CPFF, FFP)

• Contract start date

• Contract scheduled & actual completion date

Software Development Process

• Extent of software engineering participation in systems analysis & design

• Methods/tools used for requirements analysis

• Methods/tools used for design

• Methods/tools used for testing

Project Metrics

Software Management Process

• Extent of software manager participation in proposal

• Percent of subcontracted software

• Methods/tools for size estimation

• Methods/tools for cost estimation

Software Development Environment

• Development HW/OS

• Training provided

• Team co-located?

• Performance characteristics of computers and networks