



**Measuring Progress over the  
Software Development Lifecycle**

## Reader ROI

- Traceability across system requirements and test cases can help determine the risk associated with deploying a system to production.
- When managing the schedule for large system implementations, metrics for test planning and execution are only meaningful if they can be related to the overall complexity of the system.
- Several automated tools have been created to maintain and report traceability of data.

# Measuring Progress over the Software Development Lifecycle

IT testing organizations are routinely asked to adjust schedules to meet changes in system requirements or changes to delivery schedules. Unfortunately many testing groups are ill-equipped to quantify their progress in either preparing for the start of testing or their progress in completing the successful execution of all test scripts. But implementing even a minimal level of traceability across the software development lifecycle produces metrics and data that support test reporting in the areas of test planning and preparation, test execution, system stability, system readiness, and overall deployment risk assessment.

## Structuring information to yield metrics

While care must be taken to avoid allowing metrics collection efforts to derail the progress of critical development initiatives, a minimal set of information must be collected from each phase of the software development lifecycle (SDLC) to make initiatives manageable and predictable.

- **Functional complexity metrics:** As the targeted functionality of a system is documented, some structure must be put in place to secure this information in a uniform fashion. This structure should support the capture of key information in system requirements documents and drive a uniform level of complexity across functional requirements documents. Metrics are simply derived from enumerating functional requirements and grouping them as appropriate. Additionally, it is necessary to collect meta data describing the business criticality associated with each requirement.
- **Technical complexity metrics:** When the functional requirements of the target system have sufficiently stabilized and technical design begins, the components of the system or platform should be individually labeled and described at a common level of granularity. Produced metrics are primarily a listing of the components that comprise the system architecture—often accompanied by meta data describing impact of component failure.
- **Test metrics:** Ideally, testing metrics will be derived from the lowest level of the test information architecture, usually the test scripts.

## Connecting metrics to achieve traceability

The above sets of metrics must be connected to produce information of any real value. Test cases are targeted for the validation of specific functions of a particular system. Test plans may be organized around common system functionality, a component of the platform architecture or a specific class of testing (performance, availability, etc.). Ultimately, traceability must be established between complexity metrics (functional requirements, architecture components, etc.) and validation metrics (test cases and scripts by function, module, or test class).

Typically, IT test organizations are focused on traceability from test cases to functional requirements. Additional traceability should be established between architecture components of the system and the associated test scripts. This allows for two dimensions of system stability reporting:

- **Business support:** As test cases are successfully executed, metrics can be reported on the percentage of key business functions that now have validated support by the system. This is critical in assessing the ability to promote a system to production, even if all testing has not been successfully completed—a typical situation when development cycles are extended without moving the final system deployment date.
- **Technical impact assessment:** In addition, by establishing traceability between test cases and components of the system architecture, testing groups can report on the stability of the system from an architectural perspective as well. If system components are described with meta data that includes information about dependencies between system components, then impact analysis can also be performed.

## Automated traceability through integrated tools

Although Microsoft Excel can be used to track and maintain associations required for traceability, the work quickly becomes labor-intensive. Fortunately, traceability and the associated reporting it enables have gained sufficient traction over the last five years that the IT tools industry now provides packages for automating the maintenance and reporting of traceability data.

- **All-in-one solutions:** Typically offered in the test tool space, traceability can be established by managing both system requirements and repositories of test cases in a single tool.
- **Integrated solutions:** Traceability has become sufficiently important so that tools from competing vendors now work together to report on traceability across artifact repositories.

Regardless of which approach is used to establish and manage traceability, it is critical that a sufficient level of update automation is supported to ensure that traceability data stays current and accurate.

## Leveraging traceability for advanced reporting

Once traceability has been established and its management is supported by a sufficiently robust infrastructure, reporting can be extended to support the decision-making required by complex IT initiatives. For example, progress-to-goal information can be reported across the software development lifecycle and can be described by standard measures (e.g., percentage of completion by business or system function or by architectural component).

The percentage of the system that has been validated through test execution can be stated as a percentage of the requirements that have been met to demonstrate the system's stability.

Finally, the readiness of a system to be deployed to production can now be quantified by functional completeness, component completeness, or by the number of defects associated with particular functions or components.



**EMC Corporation**  
Hopkinton  
Massachusetts  
01748-9103  
1-508-435-1000  
In North America 1-866-464-7381  
[www.EMC.com](http://www.EMC.com)