Quality by Design

Ronald N. Goulden, PMP

Overview

There are countless philosophies regarding Quality Assurance and how to achieve ‘Zero Defects’ in a manufacturing or production environment. Less effort has been expended in defining the quality requirements for a software development environment. While quality standards have been developed for the more traditional environments and industries, corporate Information Services Departments lag behind.

Borrowing heavily from one of the most regulated industries in the world, American Food and Drugs, we can identify four key metrics and focal points of opportunity to improve the quality of Corporate Information Services efforts. These metrics are:

- **DQ - D**esign **Q**uality
- **OQ - O**perational **Q**uality
- **PQ - P**rocedural **Q**uality
- **IQ - I**mplementation **Q**uality

We also use a Quality Scorecard (see example at the end of this paper) to measure the ongoing quality lifecycle of the project or application. There is a column for each of the Development Quality metrics as well as columns for the QA function and the post-implementation cycle.

Please keep in mind that these first four design quality checkpoints MUST be performed before turning the application over to any formal QA function in order to have any level of significance.

DQ: Design Quality

As the name implies, this involves designing for quality before the programming even begins. This requires involvement of the user in an iterative process whereby the development team partners with the user community to identify:

- What the user and functional requirements are and
- What requests can be technologically achieved.

Normally, this function is performed by the project design team while working closely with Sponsors, Business Analysts, and other Subject Matter Experts (SME). This is an iterative process wherein the design of the product and deliverables is refined and agreed upon by all parties.

The process begins after a user request has been received by the Information Services department. Commonly refereed to as the fact-finding stage, it involves open discussion and cooperation to clearly identify what the user community requires. It then involves melding those requirements with the Information Services standards and corporate goals that will produce an application or system that functions reliably within the current and future Information Services environment.

Flexibility and adaptability are key criteria during this phase. In this phase, screen and report mock-ups are presented and iteratively modified until all requirements have been achieved, including Information Service design standards.

When a consensus has been reached among the user and developer community, the drafts are signed by the user sponsor and the Information Services representative and the project scope document can be drafted. The process is then documented, including project scope, process flow, Work Breakdown Structures (WBS). The mock-ups of screen and report layouts have already been approved and signed by the users and sponsors.

When the documentation package is complete, it is presented to a Design Review committee. The purpose of this committee is not to redesign the application requested and approved by the users, but to ensure that critical aspects of design characteristics have not been overlooked. This ensures that the design has the flexibility to support future needs of the business. This review must adopt a 'forward-looking' attitude and not base its decisions and recommendations on historical perspectives. To do otherwise is to force mediocrity into the application or system.

Any significant changes in the design that are made by the Review Committee must be approved by the user sponsor. These changes should be limited to standardization, flexibility, and utility, and not based on how things have been done in the past.

Once the Review Committee and the User Sponsor agree to the design, it can be turned over to the development team. At this point, the Design Quality (**DQ**) metric measurements begin. As the development team begins work, there may be design flaws or shortages that come to light as those with more detailed knowledge get involved. Any design changes discovered and suggested by the development team should be recorded on the scorecard by the Project Manager.

PQ: Procedural Quality

The core theme behind defining Procedural Quality is the enforcement of clearly defined programming standards. Modularity and code re-usability are critical success factors.

This level of quality is incorporated by the developers and their managers.

While the definition of the standards is incumbent upon the Information Services Management team, the actual implementation rests in the hands of the developers. There must be management oversight and involvement to ensure that the standards are being followed.

With the implementation of standards and pre-defined common modules, the foundation of every program will be established and the developer can concentrate on the business logic rather than continually spending time developing repetitious routines. The corollary to this is that the basic flow and navigation of every program will work exactly the same, reducing the effort required to test in the QA areas.

As the development team continues work of the product or application, structural flaws may become apparent. Additional modules or programs may be affected that were not identified in earlier stages. These discoveries and observations may require changes to the scope that must be approved by the sponsors, since these changes may impact the Cost and Time Baselines. Procedural Quality (**PQ**) focuses on the individual components that make up the entire application or product.

OQ: Operational Quality

Operational Quality ensures that the components of the application or system will interact as the users expect when it is delivered. Typically, this is an iterative process performed by the members of the development team. Any issues requiring correction are coordinated among themselves for resolution. However, the Project Manager must be informed of any issues, which must be recorded in the Operational Quality **(OQ**) section of the Scorecard. Issues discovered at this stage in the Development Quality cycle could indicate serious design flaws, and should be documented for future efforts.

If any significant changes to the design or functionality of the system or application are identified and recommended by the OQ team, it should be reviewed and approved by the Sponsors, users, and the Review Committee to ensure that the changes will satisfy the needs of the user, adhere to standards, and maintain the flexibility to support future needs of the business. Close attention to the Cost and Time Baselines must be maintained.

IQ: Implementation Quality

The final component of the development quality definition is at the installation level. Normally, this is performed by a team that is responsible for documentation, training, packaging, and installation. By segregating this group from development and Quality Assurance, a fourth level of quality checking is applied to the overall development process. It is not uncommon for members of this team to discover problems and issues that were over-looked by the users, developers, and QA team.

By combining the different viewpoints from these distinctive team members, we can provide a closer view of what the actual end-user will see and experience. With some control and monitoring, this provides a final point to eliminate errors and flaws before going through a formal QA process.

The first step in the process is performed by the Project Manager, who must ensure that all documentation is correct and matches the application; that the application actually functions in accordance with the defined functional requirements; and that the supporting turnover documentation will allow other members of the IQ team to perform their assessment and functional responsibilities. Any flaws or omissions discovered by the Project Manager should be recorded on the scorecard in the Implementation Quality (**IQ**) column and returned to the development team to make the necessary corrections.

QA: Quality Assurance

This requires establishing standard test scripts to test each application or system by approved criteria.

Typically, this is an iterative process performed by the members of the Quality Assurance team. Any issues requiring correction are coordinated with the developers for resolution. If any significant change to the design or functionality of the system or application are identified and recommended by the QA team, it should be reviewed and approved by the Sponsors, users, and the Review Committee to ensure that the change will satisfy the needs of the user, adhere to standards, and maintain the flexibility to support future needs of the business.

In any event, all defects identified by the QA team must be recorded on the Scorecard by the Project Manager.

The defects identified by the QA team will illustrate areas of improvement for the design and development teams and areas where they need additional work on their testing procedures.

It is important to note that at the end of the fourth QA cycle, the results ONLY indicate the perception of the Project Quality from the standpoint of the development and QA teams. At this point, the user perception is not available.

Post-Implementation Evaluation

We take the stance that the first month after implementation of a project or application; it is still under the auspices of Project Management and the full ownership of the Project Manager. This provides a better framework to determine our overall project performance.

It is very common for an application or project to be delivered to the user community, thinking that the resulting product is feature-rich, easy to use, and meets the needs of the user. However, until the time the product is in the hands of the end-users, the Project Manager has absolutely no metric to determine the value of the product to the user community. That is why the Post-Implementation Evaluation is essential.

It is incumbent upon the Project Manager to track all project or application problems that are experienced by the users during this first four week period. This allows the Project Manager to identify design flaws or shortfalls as well as providing a mechanism to track the effectiveness of the Quality Assurance and Training & Documentation teams. By keeping the project ‘open’ during the first four weeks of implementation, we can gather valuable information to improve future efforts. While this extends the Cost and Time baselines a bit, it should be accounted for in the initial budgeting and scheduling processes.

The Project Manager should track each problem on a weekly basis. The completed Project Scorecard with graphic and supporting line items should be a fundamental document used in the Project Close-out meetings. It is not a ‘punitive’ tool, but a broad ‘how did we really do’ versus what we think we did.

The final scorecard can be invaluable in identifying shortages in the quality lifecycle. In this instance, the developers discovered issues after the design approval, and had to go back to the review team and Sponsors to get those design shortages included in the scope.

After turning it over to QA, the four-week QA cycle found relatively few defects before handing it over to the Training, Documentation, and Implementation teams. Upon initial implementation, the number of flaws and problems escalated quickly. This could indicate to the Project Manager that better oversight is required on the QA, Training, and Documentation processes.

Finally, this graph clearly indicates the differing viewpoints of the Development and QA teams when contrasted with the views of the end-users. At the end of the QA cycle, the defect count was near zero. During the first week of implementation, the errors escalated to over forty, much higher than any point during the development/testing cycle.

With an unbiased and open dialogue at the Project Close-out meeting, each functional manager may see areas of improvement. Even if the Project Manager categorizes an error incorrectly, the counts will remain the same, illustrating the overall quality of the project from the viewpoint of the end-user.

Summary

In summary, Designing for Quality requires:

1. Use of four additional quality metrics beyond the typical Quality Assurance team;
    a. **D**esign **Q**uality –
    b. **O**perational **Q**uality -
    c. **P**rocedural **Q**uality -
    d. **I**mplementation **Q**uality -
2. Accurately track, categorize, and record every flaw, shortfall, defect, error, and bug.
3. Use the Quality Scorecard to monitor and report the Project Quality Progress.
4. Inclusion of the Post Implementation cycle as a Quality metric on all projects

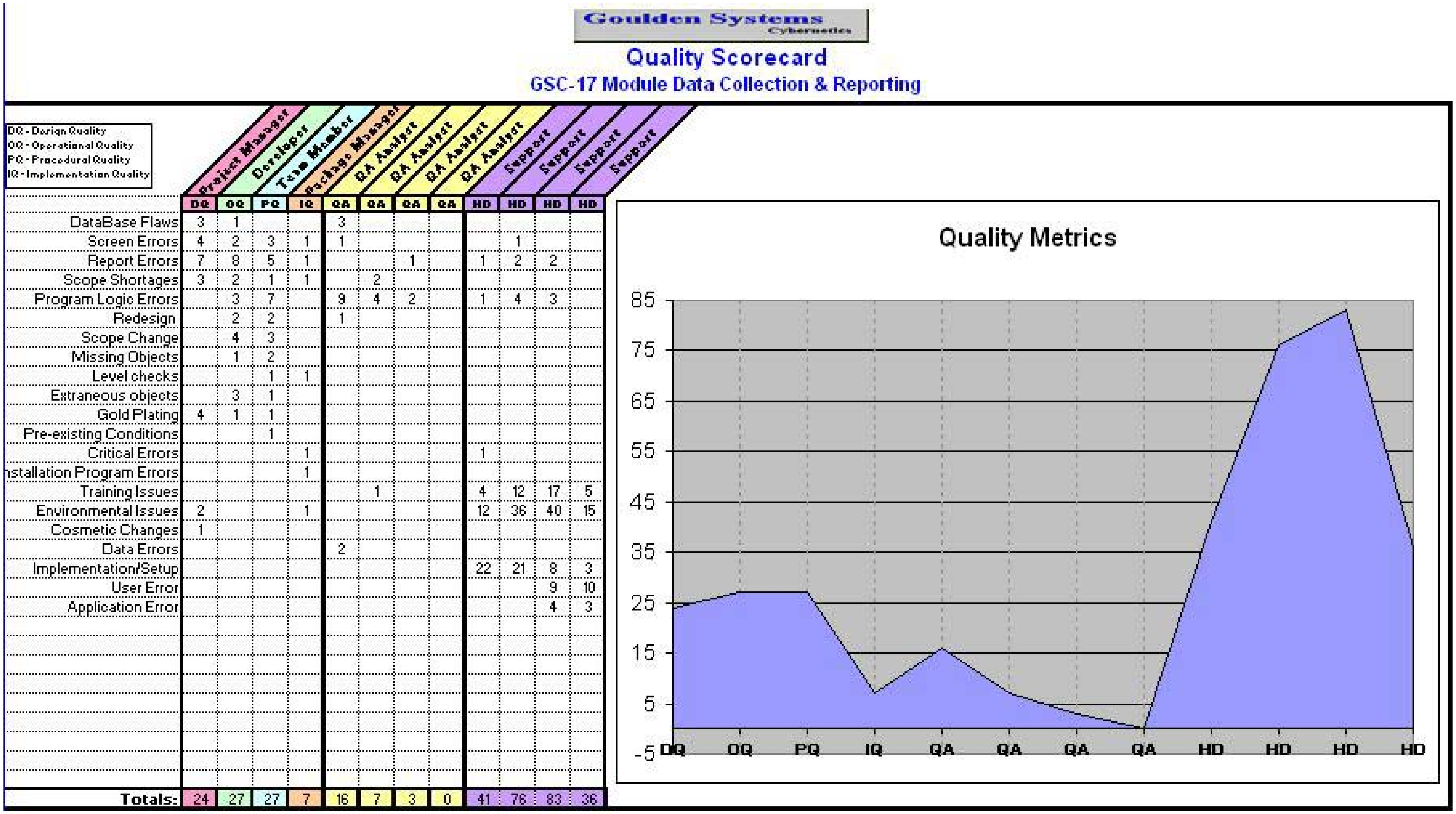
The Author

Ronald Goulden has worked in the Information Technology field for thirty years and has predicated his career on salvaging or rebuilding dysfunctional IT departments, always with an eye toward process improvements and overall enrichment of the corporate structure. He is President of Goulden Systems Cybernetics, which has two divisions, one devoted to research and development of Artificial Intelligence and Robotics; and a second division that can provide Project Management and process improvement consultations to companies searching for relief. He has found that the lack of quality standards and control are at the root of many IT issues and failures.

Project Perfect

Project Perfect is a project management software consulting and training organisation based in Sydney Australia. Their focus is to provide creative yet pragmatic solutions to Project Management issues.

Project Perfect sell "Project Administrator" software, which is a tool to assist organisations better manage project risks, issues, budgets, scope, documentation planning and scheduling. They also created a technique for gathering requirements called "Method H"™, and sell software to support the technique. For more information on Project tools or Project Management visit www.projectperfect.com.au



Gouldren Systems
Cybernetics

Quality Scorecard

GSC-17 Module Data Collection & Reporting

DQ - Design Quality
OQ - Operational Quality
PQ - Procedural Quality
IQ - Implementation Quality

	Project Manager	Developer	Team Member	Package Manager	QA Analyst	QA Analyst	QA Analyst	QA Analyst	Support	Support	Support	Support
	DQ	OQ	PQ	IQ	QA	QA	QA	QA	HD	HD	HD	HD
DataBase Flaws	3	1			3							
Screen Errors	4	2	3	1	1					1		
Report Errors	7	8	5	1			1		1	2	2	
Scope Shortages	3	2	1	1		2						
Program Logic Errors		3	7		9	4	2		1	4	3	
Redesign		2	2		1							
Scope Change		4	3									
Missing Objects		1	2									
Level checks			1	1								
Extraneous objects		3	1									
Gold Plating	4	1	1									
Pre-existing Conditions			1									
Critical Errors				1					1			
nstallation Program Errors				1								
Training Issues						1			4	12	17	5
Environmental Issues	2			1					12	36	40	15
Cosmetic Changes	1											
Data Errors					2							
Implementation/Setup									22	21	8	3
User Error											9	10
Application Error											4	3
Totals:	24	27	27	7	16	7	3	0	41	76	83	36