Software product quality assurance

by JOHN R. RYAN
Texas Instruments, Inc.
Austin, Texas

ABSTRACT

Providing clear objectives, guidelines, and requirements in an environment conducive to high productivity is absolutely essential to designing and producing high-quality software. The Software Quality Branch of the Computer Systems Division of Texas Instruments is tasked with providing support functions that are vital to producing high-quality software.

This paper explains the role of the Software Quality Branch in administering the development methodology of the Computer Systems Division. The paper also describes our participation in a corporate effort to define and monitor quality indices and our use of a software quality circle to encourage commitment to quality goals and to develop solutions to quality problems.
INTRODUCTION

For a computer system to be competitive in today's marketplace, the manufacturer must commit significant resources to its support. For the product to remain viable, timely maintenance releases must be made to correct functional deficiencies, to provide the additional functionality needed to maintain a leadership position, or simply to meet competition. When functional enhancements are made, it is essential that upward compatibility with earlier releases be maintained in order to protect the users' investment in application software. The product must be scrutinized to assure that manufacturability, configurability, and ease of installation have been accounted for in the packaging.

Good software engineering practices must be employed in the analysis, design, and implementation of the product for it to be testable, maintainable, and easy to use. It is economically essential that the development and maintenance of a viable computer system product be carefully controlled in order to meet customer requirements while minimizing development, maintenance, and support costs.

It is the responsibility of the Software Quality Branch to provide all levels of management with the information necessary to orchestrate the development and maintenance of a computer system product without neglecting any of the critical quality attributes, including the following:

1. Correctness—The extent to which a program meets its specifications and fulfills the user's objectives.
2. Flexibility—The effort required to modify an operational program.
3. Interoperability—The effort required to couple one system to another.
4. Reliability—The extent to which a program can be expected to perform its intended function with the required precision.
5. Maintainability—The effort required to locate and fix an error in an operational program.
6. Usability—The effort required to learn, operate, input data and interpret output of a program.
7. Testability—The effort required to test a program to insure that it performs its intended function.

THE SOFTWARE QUALITY ORGANIZATION

Each section of the Software Quality Branch is responsible for assuring that one or more of the quality attributes are present in the product under development. The sections are as follows:

1. Software Quality Assurance
2. Software Audit
3. Software Engineering Support
4. Software Configuration Management

Software Quality Assurance

The Software Quality Assurance (SQA) Section verifies that the product meets all specifications (requirements, functional, and design). When authorization has been granted to develop a new product or enhance an existing one, staff begin immediately to develop an overall test plan.

The test plan describes both the strategy for testing the product and the hardware and software configurations in which it will be tested. A list of features added since the last version is complemented by a description of new tools and programs that will be developed to test them. All previously developed test tools and regression test software are described. A schedule is included, indicating when specifications, user documentation, unit test results, hardware, and software are to be delivered to SQA and when status reports will be made to the project manager.

An SQA representative attends the weekly project reviews to report on the status of the test effort. SQA verifies the technical content and clarity of the user documentation produced by the Technical Publications Branch.

Documentation and software defects are noted by submitting Software Trouble Reports (STRs). When SQA has received a new version and verified that the deficiency has been corrected, the STR is marked "fixed." Both the SQA and the project managers receive reports showing the status of all STRs against the product. Using these reports to track outstanding STRs reduces the chance that a problem will be accidentally overlooked.

When all tests have been passed satisfactorily and the product and documentation are deemed customer-ready, i.e., meet all specified requirements, they are turned over to the Software Audit Section for final verification.

Software Audit

The Software Audit (SWA) Section is responsible for verifying the installation and operation of the computer system product on actual production hardware, using only the documentation that will be shipped to the customer. A software auditor must simulate as closely as possible a "real" customer. He uses the system in a simulated production environment.

A minor defect that is encountered after SWA has begun final verification must be documented in the product Release Information document. If the problem is sufficiently serious that the product is no longer judged to be customer-ready, it is returned to the project and must undergo regression testing by the SQA Section before it can be resubmitted to SWA.
Some examples of serious problems that can cause a product to fail final verification are these:

1. It causes the operating system to crash or hang.
2. It causes data to be lost or destroyed.
3. It interferes with the operation of other programs.
4. It does not satisfy its specifications.

By insuring that all known problems with the final version of the product are either fixed or documented, the SWA Section protects the customer from unexpected difficulties when installing and using the product.

**Software Engineering Support**

The Software Engineering Support (SES) Section reviews specifications, evaluates software prototypes from a human-factor point of view, and coordinates the development and distribution of appropriate software tools that are used internally for development activities.

SES representatives attend all design reviews and comment on the consistency and compatibility of the proposed design with respect to other related products. In addition, they coordinate the proposed changes in development standards or methodology.

Because of the involvement of the SES Section, many compatibility and usability problems are identified and corrected early in the development cycle. Their efforts in promoting standards, methodology improvements, and the use of software tools also contribute to the efficiency of the development effort.

**Software Configuration Management**

The Software Configuration Management (SCM) Section is responsible for controlling source changes for products under development. They verify that product installation kits (the collection of programs that the customer purchases to install on a system) can be manufactured from the corresponding library and produce each intermediate test version for the collection of programs that the customer purchases to install.

SCM integrates new modules to be tested into the master library and produces each intermediate test version for the project and SQA.

One of the key elements of configuration management is the control of reports of software failures and requests for design changes. For this reason the SCM Section manages the Software Trouble Report (STR) system. This system tracks functional deficiencies and requests for enhancements from customers, field analysts, and factory personnel.

**THE DEVELOPMENT CYCLE**

The Software Quality Branch participates actively in each phase of software development, assisting management in verifying that all milestones have been met and all quality requirements are being satisfied. The major phases of product development are as follows:

1. Initiation
2. Definition
3. Design
4. Programming
5. System test
6. Acceptance

**Initiation Phase**

During the initiation phase of product development, all market requirements are analyzed by the Product Planning Branch. The product planners work with systems analysts from one or more of the software development branches to produce the marketability requirements specification. At this time, the SQA Section becomes familiar with the system requirements in order to later evaluate the functional specification, prepare the product test plan, and develop any test tools or environments needed.

The SES Section uses the marketability specification to establish a user profile for the proposed product. This user profile, which assesses the background, capabilities, and preferences of the projected user, forms a basis for evaluating the applicability of the user interface and documentation.

**Definition Phase**

The definition phase of product development includes the analysis of functional requirements necessary to satisfy market demands. It culminates in the review and approval of the functional specification. SQA verifies that the proposed product meets all of the marketability requirements and develops a preliminary test plan. The SES group projects whether the system will be easy to use and whether it will be operationally consistent with related software products.

**Design Phase**

The SQA Section reviews the system design during this phase to insure that it is complete, consistent with the approved functional requirements, and compatible with related system products. SQA also completes the product test plan and puts it into final form. The test plan is reviewed and approved by the project manager. As draft copies of the user's guide and other manuals become available, they are reviewed by SQA for accuracy of technical content and conformity to system specifications.

**Programming Phase**

During the programming phase of product development, the Software Configuration Management (SCM) Section assists the development programmers in controlling source code changes; archiving copies of source, object, and listings as required; and maintaining the unit test library.

The software prototype of the user interface is evaluated by the SES Section for usability, user friendliness, and simplicity.
SQA participates in project code-reading sessions to see that the code is being developed according to standards and to help identify problems early. They evaluate the unit test results to determine when the product is ready to proceed to the system test phase.

System Test Phase

Most of the system test phase is executed in the factory by the SQA Section. An internal alpha test is conducted as soon as the product is sufficiently functionally complete and operationally stable to be used in a limited production environment. When the product has satisfactorily completed the alpha test, it is distributed to selected external customers for beta test. The beta test period is concurrent with the final weeks of system test.

Internal alpha test

As integration is completed on major subsystems, they are turned over to the SQA Section for system testing. Problems encountered are reported to the project manager so that the modules affected may be identified, corrected, and resubmitted to SCM. SCM integrates the changes into a new test version for SQA. The test history and current problem status are used by the project manager and the SQA manager to determine when the product is ready to proceed to beta test and finally to the acceptance phase.

External beta test

The SQA Section coordinates the planning and execution of the beta test. With the assistance of Product Marketing, customers are identified to use the new software for its application in a controlled, quasi-production environment. Each beta test site is contacted weekly for a report of confirmed or suspected bugs. This information is used to identify areas within the system where intensive testing may be needed. The beta test arrangement gives the selected customer the advantage of advance information about the new product and a head start in developing applications to use the new functions. It also enables SQA to identify problems in environments that are extremely difficult to stage or even simulate in the factory. The beta test sites provide an important evaluation of the usability of the software and user documentation.

Acceptance Phase

The SWA Section performs the final verification for the product during the acceptance phase. SWA receives the product on the same media as will be shipped to customers, including the released documentation. It is installed and executed according to the instructions in the manuals and verified to be functionally complete prior to shipment.

THE SOFTWARE QUALITY CIRCLE

The Software Quality Circle has been established within the Computer Systems Division to provide representatives of each development group with a forum where they can express their concerns about quality and propose solutions to any perceived problems. This not only promotes cooperation but also gives the Software Quality Branch access to the combined experience, wisdom, and innovative creativity of the development branches through their representatives on the circle.

Typical Issues

One of the first issues raised before the circle was the critical dependence of all other development groups on the basic functions provided by the operating systems. In order to better coordinate incremental changes to the operating systems, an OS control board was established. Composed of representatives from the various development areas, the OS control board evaluates the potential effect of each proposed change on other products.

After synthesizing, prioritizing, and carefully studying a list of problems that were judged to undermine the quality effort, the circle focused on several key areas of development methodology that were not being consistently applied. Various subcommittees were formed to formulate standards and procedures. They operate under the guidance of the Software Quality Branch and periodically report on their progress to the Software Quality Circle.

Assessment of Results

The Software Quality Circle has made several important contributions to the overall software quality effort during its first year of operation. The major benefits have been promoting better understanding of the root causes of quality problems and promoting community commitment to the proposed solutions.

The members come to the monthly meeting with symptomatic observations; group discussion and analysis normally lead to the identification of the underlying causes. Finally, the Software Quality Circle members have been of enormous help in promoting awareness of quality issues in the development community.

Future Plans

It is our intention to continue using the circle as a source of both new ideas and feedback with respect to quality standards and procedures. If a problem is identified that will require management attention to resolve, appropriate action will be taken by the Software Quality Branch to send the problem to the level of management necessary to solve the problem. The circle will also play an important role in coordinating our training effort to maximize the quality awareness of the development community through the use of structured methods.

QUALITY MEASURES

The project managers are required to forecast the quality of all computer system products. The Software Quality Branch
Leading, concurrent, and lagging quality indices have been established to indicate the measurable quality of the product during the programming phase, during the system test phase, and after the release of the product.

**Leading Index**

We have found that program complexity as measured by the Halstead Effort metric\(^3\) correlates well to the number of times that a module must be reworked to correct errors. We are currently establishing complexity guidelines for our products. Any module that exceeds the guideline will be reviewed and, if possible, reworked or decomposed into multiple smaller, simpler modules.

**Concurrent and Lagging Indices**

The concurrent index of quality is based on the number of problems documented by STRs while the software product is in the system test phase. At the time development is begun, a maximum number of acceptable outstanding problems is established for the product. If the number of STRs ever exceeds the maximum, corrective action is mandatory.

The lagging index is established in an analogous manner, but it is computed by using STRs that are submitted after the product is released. If the number of STRs for a product exceeds the established maximum, a new version of the product will be released to correct the reported problems.

**CONCLUSIONS**

Throughout the development cycle, Software Quality Branch personnel review specifications, plan and conduct various tests, and verify completion of each of the development phases. The project manager is regularly provided with an objective assessment of the status of the product relative to approved specifications. When conflicts arise, they are sent to higher management.

One can conclude from the specification, coordination, assessment, and verification activities described that software quality is fundamentally a management problem. This fact sometimes becomes lost in the myriad of very real technical issues and business decisions that project and quality management are faced with. The fact that computer products must be suited to a variety of applications can provide further complications for the computer system vendor.

To deal in an objective way with the complexity of modern systems having diverse requirements, it is essential to agree formally on the system requirements and a quality plan that insures that they will be met. Objective quality measures are essential to the avoidance of conflicting assessments of the true state of project completion.

A software quality circle is an invaluable forum for gaining community acceptance and support for methodology changes.

Although the responsibility for product quality must rest squarely with the project manager, the successful execution of both development and quality plans depends on the level of cooperation that the project manager and the quality manager are able to achieve.

**REFERENCES**