Managing Product Quality By Integrating Operations Research And Artificial Intelligence Technologies

Charles S. Moon, Thomas M. Moore, and Suheil M. Nassar

IBM Industrial Sector Consulting Services
Applied Manufacturing Technologies / Quality Management Solutions
3200 Windy Hill Road, Marietta, GA 30067-5499
cmoon@vnet.ibm.com, mooretm@atlvm10.vnet.ibm.com, & smnassar@atlvm10.vnet.ibm.com

Abstract

This paper describes a model-based expert system called Quality and Reliability Expert System (QRES). QRES is a hybrid system for quality management applications. QRES implements quality and reliability prediction models of Operations Research (OR) technology in an Artificial Intelligence (AI) software technology area known as *Expert Systems (ES)* or *Knowledge Based Systems (KBS)*. QRES is presently used to:

- Identify problem areas in manufacturing production lines
- Identify where to make improvements which maximize quality levels at the least cost
- Predict production quality levels for new components and assemblies during design and development

QRES performs quality and reliability analysis using:

- Object data structures
- Defect cause/effect correlation
- Quality trend in process
- Process control
- Correlation between product types and their defect mechanisms
- Interactive graphics
- Detailed hypertext help with examples

Introduction

The job of the Quality Engineer (QE) is to manage the defect levels at each step of the manufacturing process to minimize defects in the final product. The number of defects that occur may be very small, measured in parts per million. Any early defect can potentially contribute to causing more defects later in the manufacturing process.

Each step of the manufacturing process is called a *sector*. The number of defects leaving a given sector is called the *Shipped Product Quality Level (SPQL)*.

For each sector, the components coming in bring their own SPQL (from leaving the preceding sector). The assembly that leaves the sector has an SPQL that is a combination of the incoming component SPQLs plus any defects added by the manufacturing process in this sector. There may be multiple components within each component type. Calculating SPQLs is difficult because there are other factors involved.

Each component that enters a sector may have one or more defect types. The defect types for different components will differ. During the manufacturing process of the sector, each defect type may:

- remain unaffected (stay the same defect)
- become a new defect (e.g. a "crack" may become an "open")
- actually be corrected (e.g. solder reflow seals a crack or open)

At the "end" of the sector, there will be one or more test facilities. Each test facility will detect (and reject) specific defects, yet others will pass through because they either were not being tested for or they were just not severe enough to be detected. The defects that are able to leave the sector are called *escapes*.

The capability of each tester to detect a particular defect mechanism is called the *test efficiency*. The test efficiency directly affects the SPQL of the defect escapes from the sector.

Sometimes, the QE may want to focus on:

- The SPQLs of a given component through the manufacturing process
- The SPQLs of an assembly as it progresses through the manufacturing process
- The SPQLs of multiple component types as they become part of a sub-assembly.
- The SPQLs of sub-assembly types as they become part of a final assembly.

Changes in the production process, production equipment, and test equipment all contribute to changes in SPQL:

- Manufacturing equipment ages, so possibly more defects are *induced* at a sector
- If the equipment is improved or replaced, some defect types may disappear while new defect types occur
- Test equipment ages, so test efficiencies can improve or deteriorate
- Test equipment is upgraded or replaced, so test efficiencies can be improved greatly and escapes minimized
- In addition to defects caused by the manufacturing process, test equipment can also induce defects.

The relationships of SPQLs and defect types between sectors is called *Defect Correlation* Changes to the manufacturing line which induce defect levels are dynamic, and their associated defect escapes are continuous throughout the process sectors. It is extremely difficult for the QE to identify what effect the changes are having on SPQL. It is also difficult for the QE to predict ahead of time what effects changes will have on SPQL.

When QRES was developed, there were no other tools available that the QE could use to determine final SPQL, driven by all the permutations of SPQLs (*bridging*) of the preceding sectors.

This is where QRES comes in. QRES allows the QE to provide a model of a manufacturing process, describing the sectors, the components, the defect types for each component, and the defect types that can be induced within each sector. The QE can then update test efficiency changes, defect type changes, escapes quantities, etc. as they occur on the true manufacturing line. Then the QE can use QRES to observe what effect the changes have had on final SPQL for components, assemblies, etc.

OR Models and Impact of AI

Traditionally, QEs used OR based tools to tackle quality and reliability problems. Their tools were mostly based on statistical formulas that provided a single value or a set of values as a solution assuming the input parameters are correct. A complex problem domain such as quality management cannot be solved with a single solution and a great deal of expertise is required to understand the input parameters. For these reasons, these tools were primarily used by experts to solve a problem in a narrowly focused area. Then they solved the larger quality management problem by combining and extrapolating the results from the narrowly focused area.

An AI solution was important because an OR based tool could only solve a small piece of the puzzle and could not put the pieces together without an expert. KBS technology provided an ideal solution by retaining the knowledge of the expert to drive OR models toward a complete solution.

The QE can use QRES to make "what if" changes to see what effect they could have. This allows the QE to identify the changes which will provide the greatest SPQL improvement for the least investment. QRES can extrapolate what SPQLs will be for new components and/or assemblies. To do this, the QE uses known values and alters them appropriately for the new process. The QE manipulates factors to simulate probable changes. The ability to predict final SPQLs for future (similar) production provides tremendous planning value for Production and Sales. In the spring of 1989, two experienced QEs decided to create a training class to train the new QEs with little or no experience in raw board manufacturing. While organizing class materials, the two QEs realized that the tasks they routinely performed are complex and time consuming and could be performed better using a computer. They decided to create a software tool to use as a part of the training class. This tool was developed using a traditional programming language.

The new training tool was well received by the new engineers. However, for this tool to benefit these new QEs, it had to do more than just perform daily calculations. It had to be easy to use with well-documented on-line help, easy to maintain, and most importantly of all, able to store expert knowledge of experienced QEs for later retrieval.

To increase the performance, the initial OR prediction models were hard-coded into the system along with the propagation rules which guided the process flow. This was fine when there were only handful of prediction models and only small number of rules to manage the propagation. However, as more models and functions were added to the system, it became too complex to maintain and upgrade. Every update would require a complete tear-down of a module requiring several weeks. The system became virtually impossible to maintain. And, without the maintenance and upgrade, its usage started to decline and started to become obsolete.

These limitations and requirements could not be easily be fulfilled by traditional programming methods. Expert systems (KBS) technology was used to create the next generation tool. The two QEs submitted a proposal in early 1990 to build a KBS similar to the initial training tool. They would use a commercial KBS shell which could provide a productive development environment for maintaining and updating the rule base. Their proposal was accepted and the corporate QRES project started in June 1990. There were several key checkpoints, including year-end 1990 functional prototype, year-end 1991 fully functional production system, and 1992 deployments to internal manufacturing sites.

The checkpoints were met. The QEs enhanced QRES in 1992 and deployed QRES to new locations. The latest release, QRES Version 3.2, includes all the models and functions of the previous releases.

QRES Design

Figure 1, below, illustrates the approach taken developing QRES. The domain of solutions using AI and/or OR techniques is represented by the large circle. Within that domain, there exists a set of *feasible* solutions, represented by the next largest circle. Given an arbitrary starting point, OR techniques are used to drive the solution within the realm of feasible solutions. Once the domain has been diminished, AI techniques are used to search and focus in on an optimal (or near-optimal) solution set.

Quality and reliability algorithms and heuristics are used to establish an initial feasible solution. AI is then used to improve the solution to a more optimal solution. QRES does this by providing an intelligent front-end to guide the user to the optimum solution while the knowledge base (KB) component resolved the complex defect propagation, thus providing a path to the optimum solution. The integration of OR and AI techniques provides an efficient method to perform the complex tasks associated with the prediction of quality and reliability.

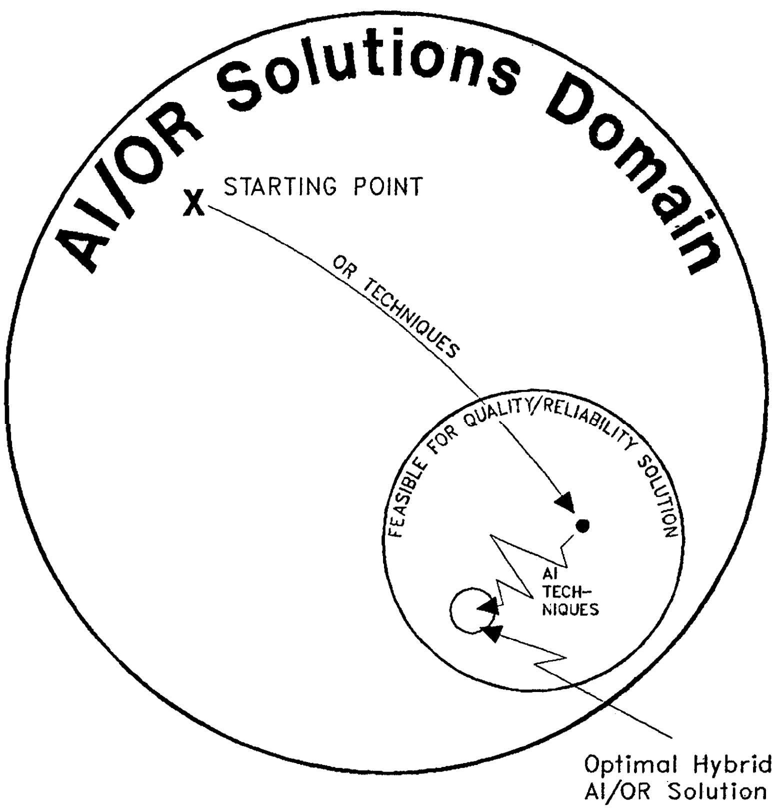


Figure 1. Use of OR and AI Technology. OR techniques are used to drive toward feasible solutions. AI techniques are used to search for and focus on optimum/near-optimum solutions.

Quality prediction is a bottoms-up approach suitable for a *forward-chaining* application since the production environment is known (described in models) and the object is to deduce the expected quality levels at each step...especially at the last production step. Reliability prediction is a tops-down approach suitable for a *backward-chaining* application since the results (product history in the field) is known and the object is to determine the defect levels at each step. QRES uses combination of these methods to solve the complex quality and reliability problems.

Figure 2, below, shows the the general concept of the linkage between the quality and reliability management in OR technology and KBS in AI technology.

The OR models used in QRES are Process Capability estimation, SPQL and Intrinsic Failure Rate (IFR), Simulation, and Cost vs. Quality trade-off analysis. These models were implemented in a data-driven, rule-based KBS using object-oriented data structures. See *Appendix A: Operations Research Models* for more detail.

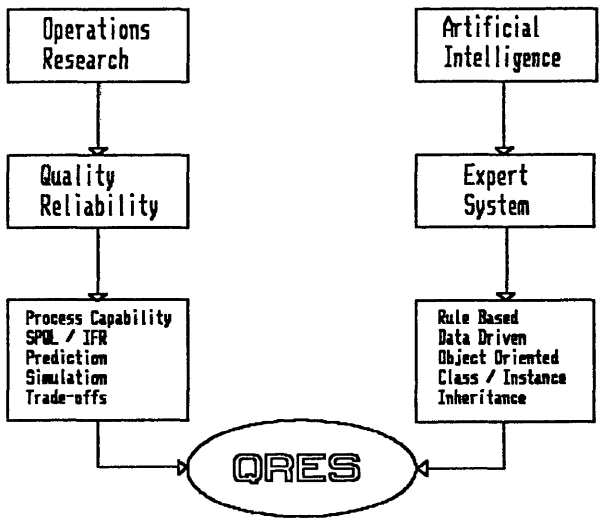


Figure 2. AI/OR Hybrid Technology. Merging of AI and OR technology can provide solutions that are more efficient and accurate by taking advantage of the strengths of both technologies.

System Overview

QRES is an *Application Enabler*. That is, by itself, it provides no value other than as a training tool for QEs to learn the terminology and functions associated with Quality and Reliability management. Once the QEs have entered data that describes their process, products, and test parameters, QRES is an extremely versatile and powerful tool. In a sense, it is a data-driven shell. Even much of the logic resides in the data.

All the analysis in QRES is menu-driven, including the interactive graphics screens. The entry screen displays the QRES version along with a menu bar that has the following menu options: Quality, Reliability, Utility, Analysis, Exit, and Help (see Table 1, on the next page). These menu item functions are accessed by simply placing the mouse cursor on one and clicking the left mouse button (mouse-driven). The entire analysis process can be navigated by using a mouse.

The latest version of QRES (Version 3.2), was developed and tested on an IBM Personal System/2 model 80 with 4 MB of internal memory, DOS 4.0 operating system, a Mouse, and a VGA color display. It was also tested successfully on other PS/2 and compatible systems with 386 and 386SX processors. A 486 processor is recommended for best performance. Although a similar configuration is strongly recommended, a mouse is required to fully utilize all point and select edit features.

Functional Description

Quality Options. The QUALITY options (shown in Table 1, on the next page) allow SPQL prediction of a product currently under production or a product under development. It uses the SPQL model (discussed in detail in *Appendix A: Operations Research Models*) to simulate a manufacturing process in terms of defect origin and propagation based on an actual production line. This process information is stored in a knowledge base (KB), along with other product manufacturing-related data, in numerical representation of a particular product line. When it is retrieved properly, using the QRES product retrieval function, QRES rebuilds the conceptual manufacturing production line from previously stored expert product knowledge (see *Product Knowledge Representation*). QRES's main strength is it can be used to design a new product automatically, utilizing the Early Manufacturing Involvement (EMI) process. In other words, a designer can design a new product knowing exactly how much each enhancement or design parameter change will affect its SPQL during manufacturing. This allows the designer to utilize the strength of the manufacturing line.

Table 1. QRES Menu Functions. Functions available using the Menu Bar.

Menu	Sub-Functions Available
QUALITY (SPQL)	Existing Product New Product Confidence Interval Growth Simulation
RELIABILITY	Existing Product New Product Confidence Interval Growth Simulation
UTILITY	OS Shell Directory Utility OS Editor Clear Product Data Print Report Save Knowledge Base Display Product Class Customize User Profile
ANALYSIS	Display Sector & Defects Display Defect Correlation Display Subproduct Perform SPQL Analysis Display Base Calculation Display Field Fail History Process Improvement Matrix

The SPQL prediction for both the existing product and the new product uses the process capability, the inspection and test efficiency, and the design configuration as a basis for calculating the required SPQL. The *SPQL Confidence Interval* option calculates the confidence band around the predicted SPQL using the binomial distribution. The *SPQL Growth Simulation* uses the modified exponential distribution for assessing the impact of process learning on SPQL improvements. Once the process learning curve has been established, the model will simulate the SPQL of the product in the future.

Reliability Options. The RELIABILITY option for both the existing product and the new product, implements a top-down approach to predicting the reliability. Reliability is measured in the same terms as quality, such as defects per million. Unlike the bottom-up approach used in SPQL predictions, these models are established from the final field reliability performance with adjustments which include:

- Population: the total number of this specific type of component/assembly in use in the field.
- Power-on-hours (POH): the cumulative time that the population has been in use.
- Failures by defect mechanisms: the number of failures due to each defect type as reported from the field
- Planned process improvements: planned changes that will affect reliability
- Design configuration: impact of new design on reliability

QRES uses these adjustments and entries to predict reliability for products, including devices and large systems. The Confidence Interval and Growth Simulation are provided to enhance the prediction process.

Utility Options. The UTILITY options provide tools that can be used to increase the productivity while using QRES. They allow the user to access the operating system tools and editors, save/print the product data, and set up a customized user profile. Another very important function provided is *product classification*. This allows the user to classify products into categories such as boards, cards, substrates, DASD, etc. It allows the user to also predefine defect mechanisms associated with each class. These mechanisms are not linked to each product class. Later, they can be inherited by its class instance, which is the product itself. *Product Knowledge Representation* covers more on this subject.

Analysis Options. Although the Quality and Reliability menu options handle KB retrieval, the ANALYSIS options are where all the prediction calculations are performed. Once the product knowledge is loaded into the system, the appropriate menu options enable display, modification, and propagation of the product information. All navigation throughout the analysis can be mouse-driven, including the graphical analysis.

AI Implementation

Expert Systems Technology

QRES takes advantage of the KBS technology to do the following:

- Simulate the job and related tasks of quality and reliability engineers and managers. This provides a means for training new people. They can obtain hands-on experience of the task without physically being involved in the process.
- Guide engineers and technicians with less skill to perform a complex task of experienced engineers. This allows knowledge and skill learned over years to be available to people with much less skill or experience so that they can perform the job when an expert is not available. This is very important since much of the knowledge about the process and the product may not be available for the next generation. Typically, most new products being developed are an extension or enhancement of the current technology. Hence, the majority of the knowledge gained from the current technology is very valuable for the new technology.
- Allow a computer system to perform routine and tedious tasks. QRES performs the simple but lengthy tasks generally performed by experienced engineers. Now these expert have more time to concentrate on ways to improve performance or efficiency.

A characteristic of KBSs is that they improve with age. Applications written in the traditional manner deteriorate with age because changes and enhancements frequently result in program bugs because of tightly coupled logic and data. KBSs allow relationships between logic and data to be loosely coupled. That is, individual rules are only invoked ("fired") when appropriate and are not necessarily affected by the data they act on.

There were several key considerations while selecting a KBS shell for this implementation. The first, and perhaps the most important, was the speed of the inference engine. Typical model-based KBSs require complex mathematical algorithms and data. Although it is possible to create external functions and data sets to overcome this hurdle, it would not be feasible to maintain and update all the algorithms as external functions. Besides, that would defeat the purpose of using a KBS shell in the first place. Another key consideration was the inference engine itself. A model-based KBS such as QRES is data-driven and the inference engine should provide a strong forward-chaining capability. The third, and perhaps the most important consideration to the developers was the actual implementation time and cost. The developers needed a KBS shell that could provide a powerful development environment, such as, integrated graphical user interface capability, menuing capability, external data handing capability, and code debug capability. After reviewing several shells, the developers selected ART-IM 1.5 for DOS (ART-IM is a product of Inference Corporation, El Segundo California). This shell provided all the requirements listed above plus many other features such as hypertext help generation facility.

We were able to deliver a fully functional prototype in approximately six months including a functional hypertext help facility. This was critical for the year-end 1990 deliverable and for the continuation of the project in 1991. One of these six months was spent introducing a domain expert to the new shell so he could generate the code for the prototype. The prototype delivered contained 125 rules, 4 internal C functions, 2 external functions, and 28 menus. Resource used to build prototype: one part-time domain expert/KBS developer for 5 months. Resource used to build the production-level system: one part-time domain expert, one part-time KE/designer, and one part-time KE/developer for one year.

Product Knowledge Representation

The QRES product data is represented as ART-IM schemas, hence the organization of the product data can be conceptually perceived in the classical FRAME/INSTANCE sense as PRODUCT, SECTOR, and DEFECT classes of objects. Table 2, below, shows the attributes for these objects.

Table 2. Object Attributes

Object	Attributes
PRODUCT	Product Name Product-class Sector-list
SECTOR	Sector Name Number-of-components Defect-list
DEFECT	Defect Name Defect-level Test-efficiency Adjustment-factor Correlation-matrix

The Product-class is a bridge that joins this product to a group of products that are considered common. This link is vital in terms of representing a product by its technology classification and by the defect mechanisms contributed by each technology.

The Sector-list attribute for each product identifies the list of sectors involved in that product's model. Likewise, the Defect-list identifies the different defects which contribute to the failure mechanism within each sector. The product model calculations require access to the individual defect attributes to maintain the model integrity.

By using a data-driven KB, any change to a data object causes an immediate propagation through the system. This mechanism allows sophisticated data integrity checks to be run automatically whenever the data is edited. Also, because it's data-driven, all propagation occurs in the background without triggering every step.

QRES represents working data in two forms: schemas and facts. A schema definition follows the normal structure of frames found in other expert system shells. Class/subclass and class/instance relations were used to create a hierarchical product structure and to allow data inheritance in the hierarchy. QRES primarily uses schema objects to represent its required control, process, and product data.

Sector information is represented as a schema with slots containing defect names and their parameters. Therefore, each sector schema values can be used as a data pattern by a rule which is checking for data integrity. Each defect element of the sector contains information about that defect's characteristics in that sector, e.g. the defect test efficiency.

The rules in QRES are functionally grouped into Initialization, GUI, Propagation, External data, External function, and Clean-up. The initialization and clean-up rules are the typical administrative rules that initiates and maintains the KB. The GUI rules handles all interactive object-oriented graphics, and the propagations rules maintains and executes the process KB. The external data handing rules interfaces the external product KB are responsible for object conversion and structuring of in QRES, and the external function rules interfaces with the external OR algorithms.

QRES Application

Applications Spectrum

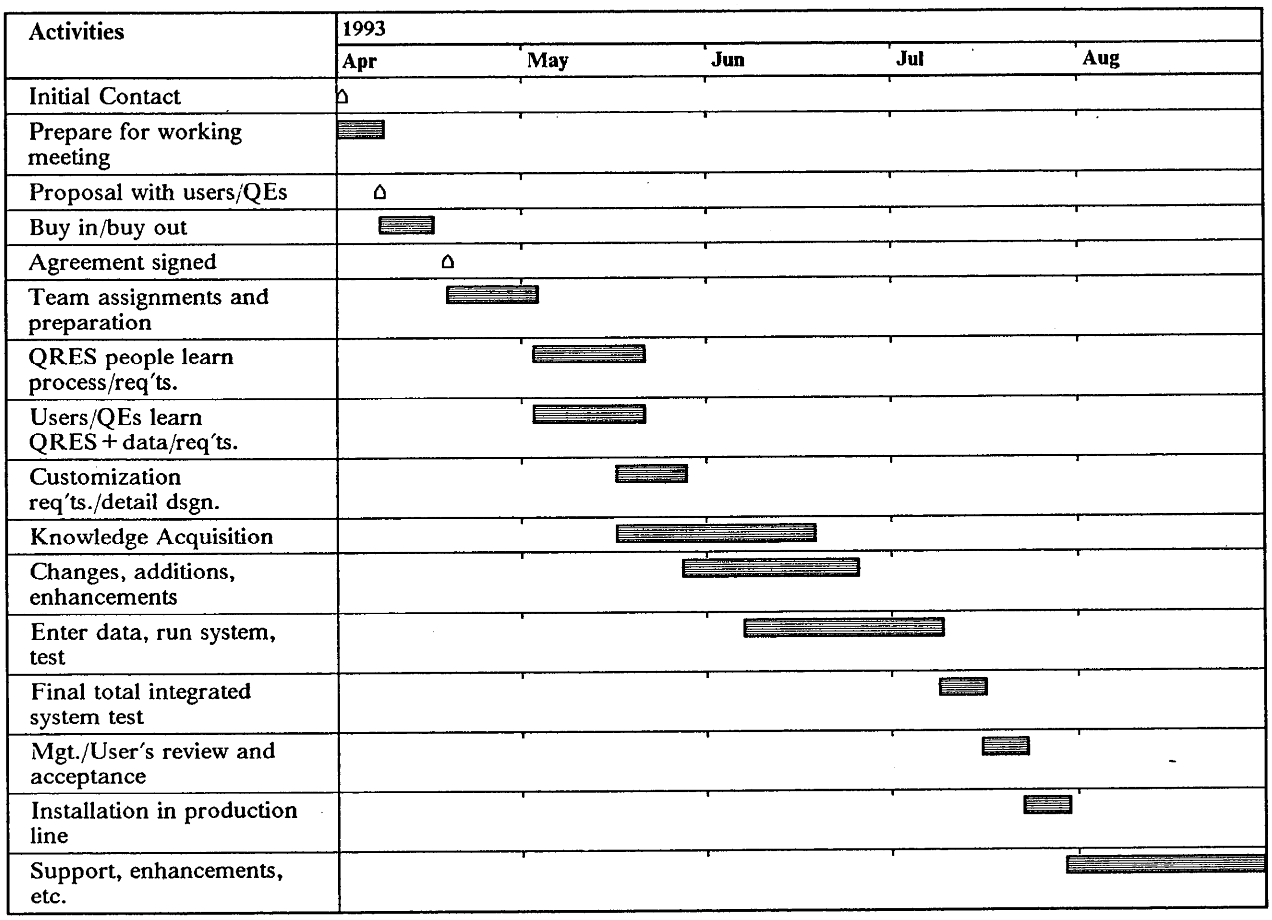
QRES has a wide range of uses across many organizations. Table 3, below, lists some key organizations and the tasks where QRES is being used in IBM Corporation.

Table 3. QRES Applications and Usage

User Organizations	**Tasks**
Quality Engineering	Prediction of process escapes Optimize clip levels
Test Engineering	Optimize test processes and parameters
Manufacturing	Constant monitor of quality performance Process improvement requirements
Reliability	Learning curve estimation Early life and intrinsic failure prediction Understanding all failure mechanisms
Development	Design based on manufacturing strength Availability prediction
Service	Spare parts inventory control Accurate service cost estimate
Central Source/CCP	Vendor quality tracking Commonality among vendors

Deployment

QRES has been deployed at several manufacturing sites that specialize in both component and subsystem manufacturing. The schedule following illustrates a typical deployment. It shows elapsed time (as opposed to actual work effort). Generally, the first month involves preliminary work: introductions, proposals, meeting the players, etc. You will notice a *Buy in/buy out* checkpoint. During these preliminary discovery activities either party should cancel out if it is discovered that there is not adequate justification to proceed. This actually creates a stronger bond between parties because they then respect each others judgements. When the time is right, it will happen. The next month is a knowledge acquisition and education process. It is through planning for the actual implementation. The next four to six weeks is spent developing the initial system and entering data. Any desired changes are made to customize the user interface, integrate with existing databases, etc. The last few weeks are spent fine-tuning the system. Once the system is in productive use, it continues to grow and be refined as more and more products are added to be tracked by the QE users.



Typical deployment activities also include:

1. Initial Contact. Our customers find us through shows, references from satisfied customers, papers, etc. They find out we understand Quality and Reliability issues and problems. They find out basically what QRES does. Once they determine they could use QRES, we meet for detailed discussions.
2. Build Credibility and Confidence. We meet with appropriate levels of management. We take QRES to them and show them how quickly we can customize QRES to fit their application. We learn their processes and can identify probable areas of greatest benefit. This is actually a consulting role. We ensure that they understand the work involved in collecting the information to model their processes in QRES. We advise, but they do the knowledge acquisition work.
3. Commitment. The customer commits people and money to the project. We jointly come up with a reasonable schedule and checkpoints.
4. Knowledge Acquisition. We work with the QEs to learn their processes as we help them identify a reasonable scope for their first application. We show them how to document their processes: the sectors, the defects, the test efficiencies, relationships, etc. Knowledge Acquisition is also discussed in *Customization* on the following page.
5. Customization (also see *Customization* on the following page). Customization involves working very closely with the QEs. We determine what customization is required to make QRES easiest to use at their site. Customization may be only a matter of gathering, refining, and selecting the

information to be entered into QRES. Or, it may involve adding rules and making changes to code so that QRES handles their processes in a special manner. We might customize QRES while they proceed to define and enter the information about their processes: the sectors, the defects, the test efficiencies, relationships, etc.

6. Data Entry. This is more involved than it sounds. The QE enters the information that has been gathered. As structures are developed, new information is required. QRES helps the QE throughout the Data Entry process. QRES has *hypertext* features that can help the QE determine which data belongs in which structures. QRES helps the QE model the production process.
7. Initial Production System. We show the QE how to manipulate QRES for many useful purposes. Actually, it is best if there is more than one QE involved because different views of the process reflected in the system make it even more useful. The QEs really begin to understand QRES's value when it is applied to their real production process. They begin to see new innovative uses for it.
8. Refinement. The Initial Production System must be refined to gain full benefit. The QEs add the information that they missed when originally developing their process model.
9. Enhancement. The QEs then can then develop new applications using QRES. They have been involved in every step. They frequently call for advice and guidance while designing and implementing QRES in related areas. Once they have used QRES for several processes, they may find totally unrelated areas of the plant that could be helped using QRES.
10. Workshops. We sometimes have our users get together for workshops and technical vitality. We learn from them and it gives them an opportunity to learn new techniques from us and others.

Application Installations

QRES is currently installed at:

- IBM Rochester: installed 2 years, 3 users
- IBM Endicott - Electronic Card Assembly Technology Development (ECAT): installed 6 months, 2 users
- IBM Endicott (High Performance Packaging): installed 1 year, 2 users
- IBM Austin: installed 1 1/2 years, 3 users
- IBM Sindelfingen (Germany): installed 1 1/2 years, 4 users

The QEs at these sites use QRES at least weekly. QRES output is used for reports to management so they can see that SPQL levels are within tolerance.

QRES is currently planned on being installed in several other internal IBM and external customer locations:

- IBM: Tucson, Boca Raton, Sumare (Brazil), and Charlotte
- Automotive assembly plants (2)
- Discrete manufacturing (3)
- Process industry (4)

Customization

QRES is a customizable application enabler which contains the generic functions, rules, heuristics, algorithms, GUI and a structure.

There are two types of customization:

1. Providing the information about your processes for the QRES shell to operate on. This involves a lot of work gathering information and data and then organizing it to enter it for the QRES system. This does **not** require program changes to QRES. Many installations would only need this type of customization.
2. Changing QRES to better suit the user. This could be minimal: changing screens, menus, field locations, etc. Or it could require the addition of rules and code so that QRES handles specific unique situations. Such changes are fairly easy to accommodate because of the use of the ART-IM shell.

Customization is part of the Deployment process (see *Deployment*). Deploying the system for a specific application domain requires customization which typically involves the following steps:

1. Requirements gathering of both the functional and system requirements
2. Design/Modeling of the specific application environment. This includes the identification and documentation of:
   - Product types
   - Product components
   - Manufacturing (production) sectors
   - Defect types within sectors. Lists all the defect types which can be caused at each step.
   - Test types. Lists the tests that are conducted at each sector.
   - Test types associated with each defect. Lists the tests that are conducted to find each defect type.

Design/Modeling also covers the correlation and propagation of defects from sector to sector, and the actual structure and flow of the parts through the manufacturing line.

3. Data and knowledge acquisition. This step is oriented towards quantifying parameters identified in the design/modeling step. For example, what is the test efficiency at each tester? It encompasses the analysis of actual design and manufacturing performance data, and the knowledge acquisition, typically in the form of rules.
4. System customization. This step uses the information gathered in steps 1, 2, and 3. QRES panels guide the QE through the process of entering the required data. QRES panels can be customized to meet specific requirements of the application.
5. Training. Typically, we offer initial training on the systems and QRES functions to new users. It is very important for new QEs to understand **why** they are using specific QRES functions as well as **how** Once they learn the fundamentals, they are able to apply QRES in many different situations.

From our experience, it is best to get our customers fully engaged in all of the above steps. This assures better success and end-user buy-in. It greatly reduces the issues typically associated with the transfer of technology and ownership.

As you can imagine, the concept of quality prediction/management is very complex. Otherwise, everyone could do it. We provide QRES software with a complete training package. Our previous IBM internal installations include initial training on QRES models, customer requirements gathering, knowledge engineering (KE), QRES screen customization, and final installation in customer sites with training. Typically, all this takes anywhere between 6 to 10 weeks with 2 to 3 weeks of customer users involvement such as training and KE. We could reduce the length of training and eliminate the re-training if customer experts and KEs could get involved. That way, we can train their experts along with initial set of users and they can train future users.

Maintenance

QRES requires two types of maintenance: product KB maintenance and process model KB maintenance. The first is considered an application maintenance, and the second is maintenance of QRES itself.

Maintenance of the Application. QRES is application-independent. So most maintenance activity is devoted, by the QEs, to refining and adding data. The QEs typically find new and creative ways to manipulate the data using QRES to get a better handle on how their process operates and where improvements can be made to gain the greatest return on investment. The application knowledge does change over time, but it is reflected in data changes, not necessarily in changes to QRES. Therefore, the KB does change. Customized QRES data entry (information entry) panels make it very easy for the QE to update the KB.

Maintenance of QRES. QRES users frequently call on us to discuss how to use QRES in new ways, since QRES is essentially application-independent. Sometimes, new uses are easily accommodated by changing data (the KB). Other times, changes require the addition of new rules and logic to QRES. These new rules and logic frequently add new capabilities to QRES. The developers make the rule and logic changes when they see a real benefit for QRES users.

This technique causes QRES to change based on *applied* Quality and Reliability in the plant, rather than from ideas gleaned from periodicals and publications (although these are not ruled out). That is, it is *market-driven* and quickly reflects the needs of today's users.

At periodic intervals, we produce a new QRES release that incorporates all the changes up to that point (typically six months). We update the QRES documentation and provide the new release to our customers. Some users will simply keep the new release "on the shelf" until such time that they need the new capability. We conduct periodic Quality and Reliability workshops that illustrate the QRES functions. QEs come to these workshops and can actually bring their own KBs to try out using the new version. When they leave, they have an operable QRES KBS to take back. We also have the capability of transmitting changes, and even new releases, over IBM's internal network.

Benefits

The utilization of AI/OR hybrid technology has benefited QRES in several ways. The most obvious is the opportunity for a system to house the strengths of both technologies as discussed earlier. By combining the two, it was possible to develop a powerful manufacturing modeling, quality managing, and process optimizing OR tool in an easy to use, easy to maintain, knowledge-based AI software platform. In other words, the OR techniques in QRES save time during quality management analysis and the AI techniques save time during development and maintenance of the KB. As discussed earlier in the introduction section, it was not possible to maintain the initial release because the KB was hard-coded into the system. The loose coupling of logic and data in KBSs allow rule-base management without affecting the data which directly impacts the time it takes to maintain and update the KB.

QRES benefits user organizations in many ways. The tangible benefits listed below were actual experiences by our customers. There were also many intangible benefits that are considered long-term improvements.

- Reduced the cycle time involved in performing the quality and reliability prediction. For each new product, QEs must determine the probable quality and/or reliability. This is critical strategic information needed by management. In one case, a customer realized a cycle time reduction from three months to one month. This not only saved two months of engineering time (3200 engineering hours), it also allowed a new product to be announced two months ahead of schedule.
- Improved the link between design and manufacturing engineering. This resulted in fewer engineering changes and more robust designs. QRES, in this case, was used as a Design For Manufacturability (DFM) tool. It allowed the developers to design a new circuit board for a new generation of computer using the their current manufacturing process capability.
- Optimized the performance of the process as a whole. Avoided the pitfall of sub-optimization. This is greatly due to the fact that QRES integrates all the data from design, manufacturing, and test into one system. QRES helps identify which sub-process can be improved with the greatest payback. The QE can perform "what if" functions with QRES. QRES quickly shows the effect of changes on the overall process quality throughput.
- Tool for performing test flow analysis, manufacturing flow analysis, and "what-if" analysis in a very robust and comprehensive manner. It is an important tool for optimizing design changes and their impact on yields, quality, and reliability.

Although there are many advantages for this system across many organizations, one key advantage that should be pointed out is the commonality it provides in the area of technology transfer. This becomes particularly important when the development of a new product involves many functions and business units, including manufacturing. QRES can help integrate the reliability related issues, and in turn, eliminate the conflict that typically occurs at the time of technology/product transfer.

QRES unifies all the data, rules, and processes associated with the quality of the product into a common platform that helps in eliminating organizational barriers.

Future Plans

Future releases will include enhancements which exploit the data representation, user interface, database utilities, and Case-Based Reasoning (CBR). Using CBR techniques, QRES will provide further intelligence to the user during analysis by making available a "best case" (analysis process) previously developed and proven effective. Hence, it will learn the best process like an expert.

Future releases will allow the user to configure a system-level product in QRES for SPQL analysis using previously configured subsystems. This object-oriented defect analysis method treats a product as a set of imaginary defects and propagates defect escapes through the parent product as if the whole product is one huge group of cascading defects. It will then be possible to estimate the SPQL of a product purely based on its system integration process and previously defined subsystems. On the other end of the spectrum, it will be possible for a designer to design and optimize the *cost versus quality* of a product simply by selecting a set of pre-analyzed components from the base to design a higher level assembly and its process that can meet the required SPQL.

The current QRES system is implemented on an IBM PS/2 and the DOS operating system. Future releases of QRES will support multiple user hardware platforms such as OS/2 2.0 and AIX. In these releases, we will be able to segregate the User Interface, Database, and Communications functions from the QRES application. This will permit:

- Real-time updates of Quality information from the plant floor
- Relational database capability for sharing information across systems
- Multiple user access
- Sophisticated interactive graphics

Other enhancements planned for the future releases include, but are not limited to, IBM Computer Integrated Manufacturing (CIM) compliance using IBM's Distributed Application Environment (DAE), Quality vs. Cost analysis option, and Defect/test Sensitivity analysis.

Summary

QRES is a KBS implementation of OR models providing a quality management shell for manufactured products. The quality management component KB resides in the rule base where the product-specific knowledge such as defects, test efficiencies, defect correlations, failure history, and process data, are stored. It is stored externally and later imported. The rule base allows easy maintenance of the quality management (prediction and estimation) models by a simple modification to the propagation rules, and also allows the user to effectively handle the large volume of product specific data.

QRES offers an environment for performing SPQL and reliability calculations based on a KB that contains all the relevant process, test, design, and field history data. This KB forms the basis from which all the parameters that are in the reliability and SPQL models get their values. QRES helps in retaining a permanent KB that would be helpful in the process of designing, developing and manufacturing a new product. Following are some key advantages of using QRES as discussed in earlier sections:

- Engineering time reduction
- Improved accuracy and efficiency
- Standardizing the prediction models
- Product/process cost reduction
- Enhanced inspection/test strategy

Appendices

Appendix A: Operations Research Models

In this section, we give the reader an overview of the QRES OR functions as they apply to quality and reliability. These functions are shown in Table 1.

Quality Prediction

SPQL Prediction for Existing Product. Each step in the overall process is referred to as a *sector*. Each sector has a list of defects which are either generated by the operation at that step or are passed to it from previous steps. Each process step also has some test mechanism for eliminating defects. Because tests are usually less than perfect, some defects will *escape* to appear as defects in the following sectors.

The SPQL Prediction for Existing Product option predicts the SPQL of a product by using the manufacturing defect types, the defect levels, and the tester efficiencies by defect types for each sector. A correlation matrix maps the defect escape from one sector to the next. The user can divide the manufacturing process into 20 sectors and have up to 20 defect types per sector.

QRES provides a Menu bar with pull-down menus and dialog boxes to guide the user through the defect lists that contain the defect levels and tester efficiencies. Mouse-selectable and content-modifiable windows allow the user to easily identify defect-correlation elements and modify their values. These windows greatly simplify the process of interactive sensitivity analysis around critical parameters while performing reliability and cost trade-off analysis. Finally, all variable data used under this option can be saved under a specific product name for later retrieval. The user can load specific product data from a previously saved KB, perform the modifications to reflect a process changes (including tester efficiency changes or process flow modifications), and then save this revised data under another name for future reference. Figure 3, on the next page, illustrates the existing product SPQL prediction model. Figure 4, also on the next page, shows an example of a method used by QRES to estimate the raw board SPQL escape.

The system must track the defects which appear in each sector, the amount of each defect which is induced by the process at each sector, and the test efficiency for each sector. The number of defect escapes from each sector is calculated based on the test efficiency, the number of induced defects, and the effect of defect escapes from previous sectors. Because one defect may have an effect on multiple defects in subsequent sectors, a correlation matrix is used to track the effect (which defects have) between consecutive sectors (bridging). The overall system reliability is determined by calculating the reliability of the product at each step and propagating the results throughout the remainder of the system.

SPQL Prediction for a New Product. This option allows the user to predict the SPQL of a product under development. In addition to the features available in *SPQL Prediction for Existing Product*, above, this option has an Adjustment Factor Matrix (AFM), which consists of the complexity factors for the new product in relation to a referenced existing product. The AFM is unique to a new product. Hence, it will be stored at the end of each application along with the remaining new product information. The AFM can either contain an overall design complexity factor or a set of complexity factors by defect mechanisms for each of the sectors.

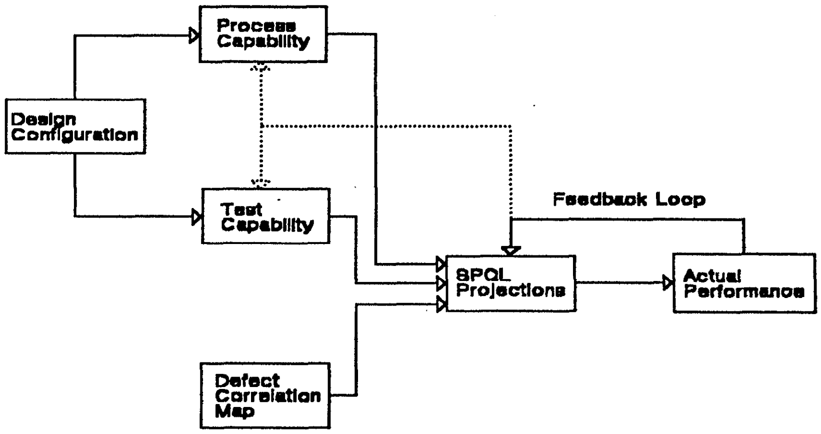


Figure 3. Existing Product SPQL Prediction Process

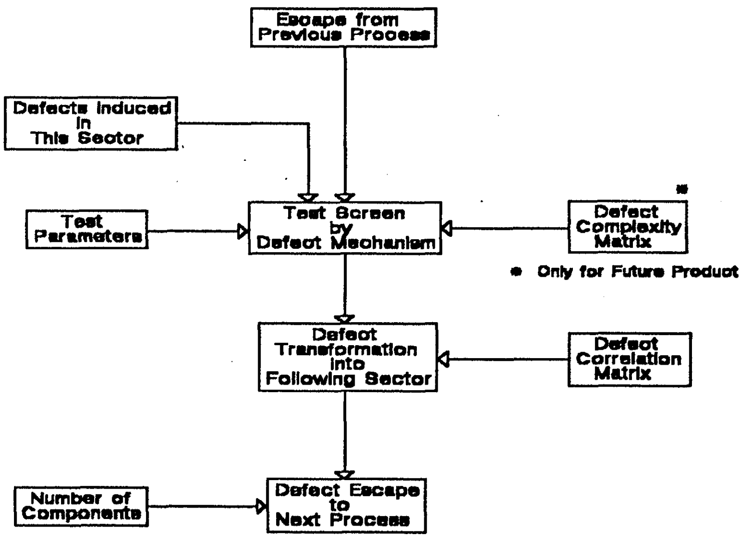


Figure 4. SPQL Estimation Model Per Process Sector

This option allows the user to simulate different design configurations, test strategies or process flows, and measure their effects on SPQL. Figure 5, on the next page, demonstrates the new product model used in this option.

SPQL Growth Simulation. This option simulates the SPQL growth of an existing product. The system quantifies the growth factor from the existing product's SPQL performance after the removal of the effects of the major design changes. This growth factor (BETA) is calculated to estimate the SPQL projections for the desired period. The growth simulation is all performed in a interactive "select and edit" dialog box format where the user can modify any of the input parameters during the simulation. The user is also provided with a graphics option during this analysis to display the growth it has estimated.

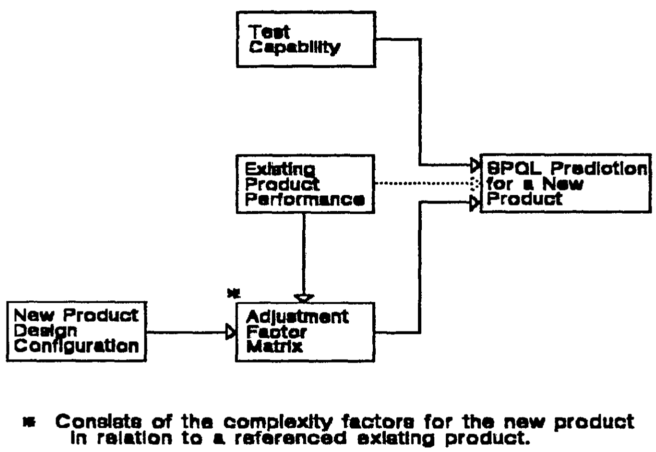


Figure 5. Future Product SPQL Prediction Model

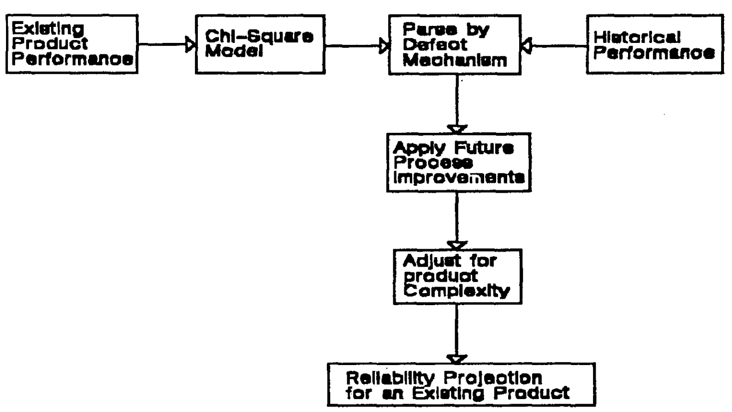


Figure 6. Reliability Prediction for Existing Product

This simulation is useful for making SPQL projections that would incorporate the following:

1. Correcting the weaknesses or errors in the manufacturing methods and designs.
2. Operating each newly developed product to identify and eliminate bad components.
3. "Settling down" in product servicing, use, and manufacturing.
4. Increasing operator skill and familiarization.

This analysis quantifies the effect that these growth factors had on SPQL growth, and then performs the future projections utilizing the modified exponential growth model. An adjustable contingency factor incorporated in the algorithm can be used to account for the uncertainty associated with the projection. It may also be used if these estimates are to be used for cost or planning reasons.

Confidence Interval on SPQL. This option calculates the confidence intervals using the binomial distribution. The user is given the option of a *one-sided confidence interval* or a *two-sided confidence interval*, and the level of confidence required. The total number of units tested and the total number of fails are required inputs. The total operating time is not required, however

The confidence interval estimation is useful in determining if the process has shifted significantly in any direction. It is significant if the SPQL shifts above the upper confidence level (UCL) or below the lower confidence level (LCL). This is helpful in determining when to initiate corrective actions for improving SPQL in the case where a degradation had occurred, or to commit to lower SPQL targets in the case where improvements have occurred. Also, this information could be helpful when calculating the process clip levels.

Reliability Prediction

Reliability Prediction for an Existing Product. This option facilitates the prediction of the field reliability performance based on planned process improvements. The base failure rate and the first 90-day reliability performance are calculated based on the field performance data. The fallout rate is then allocated to the separate failure mechanisms in accordance with the field failure breakdown. Since each process improvement is targeted to lower a particular set of failure mechanisms, the fallout associated with these mechanisms can be reduced by the assessed impact of the reliability improvement program. The chi-square method, with 50% confidence, is used to calculate the projected reliability performance. The degrees of freedom used are $(2r+2)$, such that r is the number of failures. See Figure 6, on the preceding page.

This option does not consider the improvements that typically occur due to learning. This is addressed separately in QRES under the *Reliability Growth Simulation* option.

Reliability Prediction for a New Product. This option facilitates the reliability prediction of a new product. The user is prompted for:

1. The reliability of an existing product with a similar manufacturing process
2. The reliability impact of the major design changes
3. The T-0 test results for the new product
4. The intrinsic reliability
5. The number of risk sites

QRES then assesses the *reliability growth factor* of the existing product after removing the effects of the major design changes. This growth factor is then used to predict the new product's reliability performance based on the T-0 test results. The 10% and 90% reliability confidence limits are also calculated. The user may choose a point-wise calculation of the confidence intervals based on the projected volume requirements, or may perform a projection of the UCL and the LCL as per the growth factor and the modified exponential distribution. The user is may provide the new product's projected volume and the targeted reliability for each of the stages.

This data is used to assess the confidence interval around the projected reliability values to determine if a design change or if a reliability improvement program is required for any of the stages. See Figure 7, on the next page.

This option does consider the improvements that typically occur due to learning. This is addressed separately in QRES under the *Reliability Growth Simulation* option.

Reliability Growth Simulation. This option simulates the reliability growth of an existing product. QRES quantifies the growth from the existing product's reliability performance after the removing the effects of the major design changes. This growth factor is then used to calculate the reliability projections for the future stages. The modified exponential reliability growth model is used to perform the reliability projections. Also, the user is given the option to assign reliability improvement programs or design changes to future programs, and to study their aggregate effect on the overall reliability projection.

The user is given the option to add a contingency factor to the reliability projections. This is typically used to account for the uncertainty associated with the projection, or if these estimates are to be used for product costing or planning reasons.

Confidence Interval on Reliability. This option calculates the confidence intervals on reliability using the chi-square technique. The total operating time, the total number of units tested, and the total number of fails are required inputs. Similar to *Confidence Interval on SPQL*, the user is given the option of a one-sided confidence interval or a two-sided confidence interval. The confidence interval on reliability helps in evaluating if a significant degradation has occurred to the product, or if a significant improvement has resulted from the implementation of a reliability improvement program or a design change.

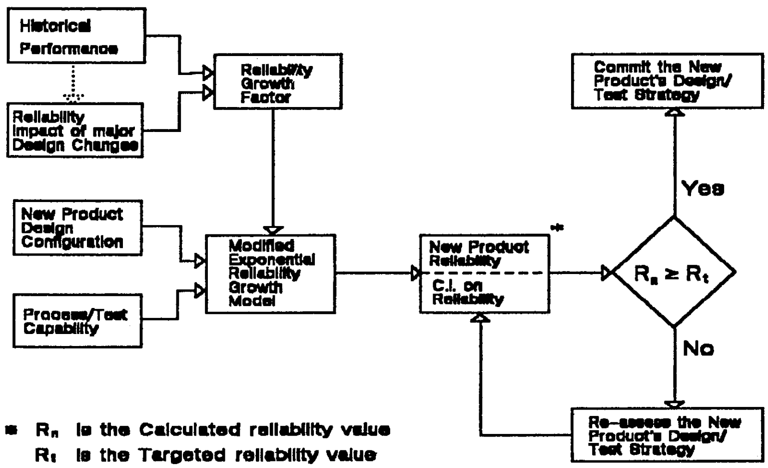


Figure 7. Reliability Prediction for New Product

This could be easily done by comparing the achieved or projected reliability value to the lower confidence level or to the upper confidence level.

In the case where the total operating time is not available, then the binomial distribution is used to calculate the confidence interval on reliability. This is similar to the method used tò calculate the SPQL confidence intervals.

References

Nassar, S. M., and Moon, C. S., *Reliability Expert System*
IBM Technical Report (TR 01.B121), July 1989

Tobias, P. A., and Trindade, D., *Applied Reliability*
Van Nostrand Reinhold Company, 1986

Winston, P. H., *Artificial Intelligence*
Addison-Wesley Publishing Company, 1992

Prerau, D. S., *Developing and managing Expert Systems*
Addison-Wesley Publishing Company, 1990