

# **QSM Reliability Model**

## **(Model Explanation & Behaviors)**

### Defect Creation Process in Software Development

The software development process is a continuous process where functionality is designed and then is expressed in some language which we refer to as source code. Defects are introduced as the source code is created. In this context it would be appropriate to model the defect creation, discovery and elimination process as a function of time.

From the time that we start to define, design, write, integrate and test source code, we have the capability of introducing defects in a product. In fact, at the beginning of a project there is nothing but a high level abstraction of what must be accomplished by this system. If we were to draw a graph of the defect discovery process at this point in time we would show no product and, of course, there would be no defects. As time progresses we complete some design and we start to generate some code. As the design and code are completed, problems that were introduced earlier are discovered and fixed. At some point in time the product reaches a maximum defect discovery rate. As work progresses the volume of remaining defects is reduced and the discovery rate gradually falls off.

### The Rayleigh Defect Model

The QSM defect estimation approach uses the Rayleigh function to forecast the discovery rate of defects as a function of time throughout the software development process. The Rayleigh function is a specific instance of one of the models in the Weibull family of reliability models. QSM believes there is a solid theoretical basis for its use as a software reliability modeling tool.

The Rayleigh function was discovered by the English physicist Lord Rayleigh in his work related to scattering of acoustic and electro-magnetic waves. In statistics it has been found that in processes with a large number of random sources of Gaussian noise, none of which are dominant, the Rayleigh function represents well the vector sum of all those Gaussian sources. We have empirically found that the Rayleigh model seems to represent well iterative design processes in which significant feedback is inherently part of the solution process. Further, we have found in our research that a Rayleigh reliability model closely approximates the actual profile of defect data collected from software development efforts.

In the QSM reliability modeling approach the Rayleigh equation is used to predict the number of defects discovered over time. The QSM application of the Rayleigh model has been formulated to cover the time period from Preliminary Design Review (PDR - High Level Design is Complete) until 99.9% of all the defects have been discovered. A sample Rayleigh defect estimate is shown in Figure 1.

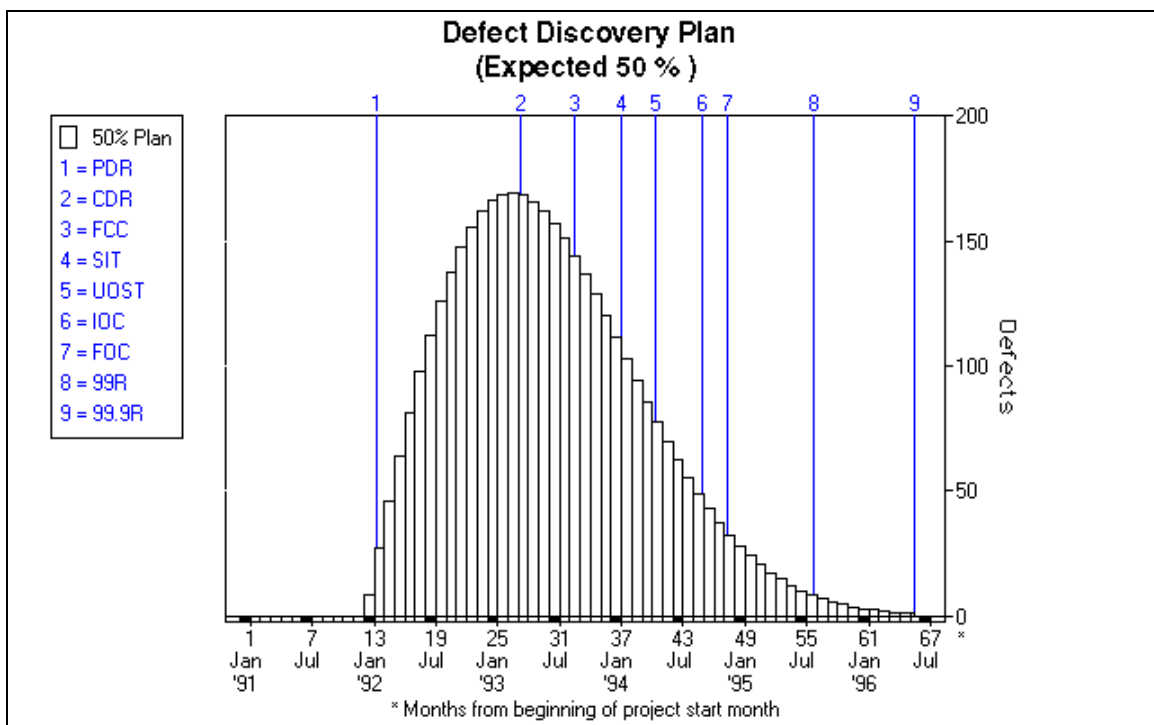


Figure 1. Sample Rayleigh Defect Estimate. This defect estimate is based on project of 350,000 SLOC, a PI of 12 and a peak staffing of 85 people.

Why does the QSM implementation of the Rayleigh model start at PDR rather than some point in time prior to PDR? There are several practical reasons. First, it is not unusual to see gaps in time at key decision points in a project. For example, there may be a 90 day delay at PDR to evaluate risk and decide on whether to proceed. This time gap would be difficult to model with a continuous time model. Second, there could be two organizations participating in the design. One does the high level design and then hands the product to a second organization that does the detail design and construction. Two organizations operating at different level of efficiency introduce a source of uncertainty. Finally, it is important to be able to establish a well defined starting point whenever one uses a time-based model. PDR is a reasonably well defined point in a project.

Note that the peak of the curve occurs at milestone # 2 (Critical Design Review). This means that a large number of the total defects are created and discovered early in the project. These defects are mostly requirements, design and unit coding defects. If they are not found they will surface later in the project resulting in extensive rework.

Milestone 9 is declared to be the point in time when 99.9% of the defects have been discovered. Most organizations do not have formal processes to capture and eliminate defects in the early design phases. Similarly, most organizations don't formally analyze the sources of the defects in their development process. Less than 5% of the organizations that QSM has worked with have kept defect data during the detailed design phase. The organizations that do keep and use the early defect data say it is worth the effort. The Fagan inspection method (IBM) is the most commonly adopted process for finding defects early in the development process. Industry researchers claim that it can cost 3-10 times more to fix a defect found during system test rather than during design or coding. From milestone 2 onward the curve tapers off.

Simple extensions of the model provide other useful information. For example, defect priority classes can be specified as percentages of the total curve. This allows the model to predict defects by severity categories over time. This is illustrated in Figure 2.

A defect estimate could be thought of as a plan. For a particular set of conditions (size, complexity, efficiency, staffing, etc.) a planned curve could be generated. A manager could use this as a rough gauge of performance to see if his project is performing consistent with the plan and by association with comparable historic projects. If there are significant deviations, this would probably cause the manager to investigate and, if justified, take remedial action.

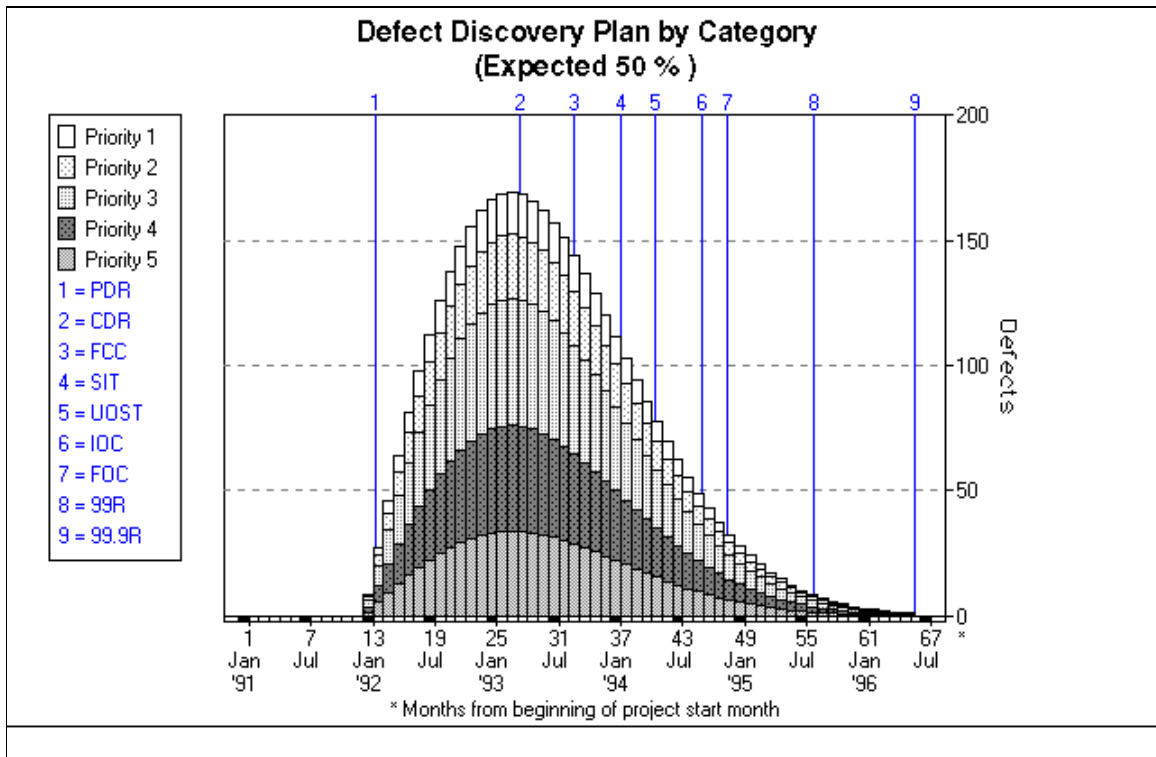


Figure 2. Rayleigh defect model broken out by defect severity classes.

## Examples of Real Project Defect Data

All of the defect data shown in this section come from real projects. Figure 3 is a project that implemented a complete life cycle defect measurement program. The reported monthly defect rates are plotted on top of the corresponding theoretical Rayleigh model. The defect data was measured from the start of detailed design (completion of high level design) and continued for 16 calendar months.

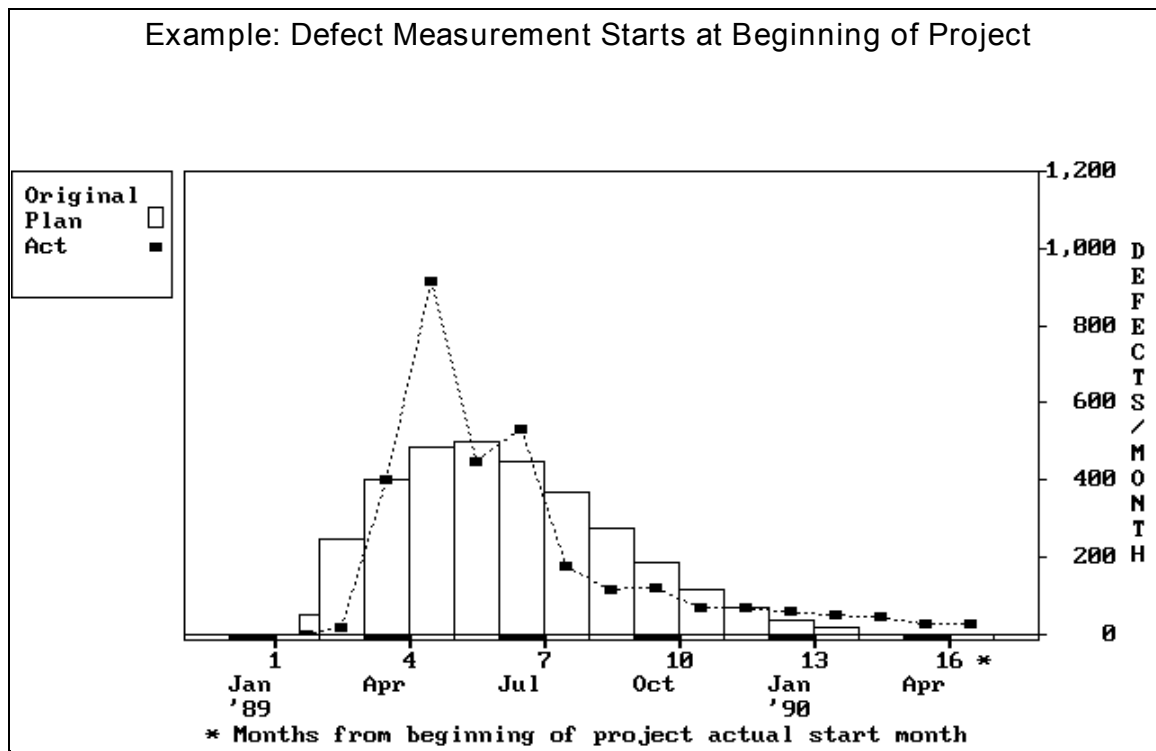


Figure 3. Example that compares the Rayleigh defect model to a project that collected defect data from the start of the project.

This comparison shows the real data often has some statistical variation when compared to the theoretical Rayleigh model. In some months the data points will be higher or lower than the model would predict. Over the duration of a project the high points will tend to compensate for the low ones so the average behavior predicted by the Rayleigh defect model will be a good approximation.

In Figure 4 we examine a real project where measurement did not start until critical design review occurred. In this case the defects found are primarily in unit coding, unit testing, integration and system test.

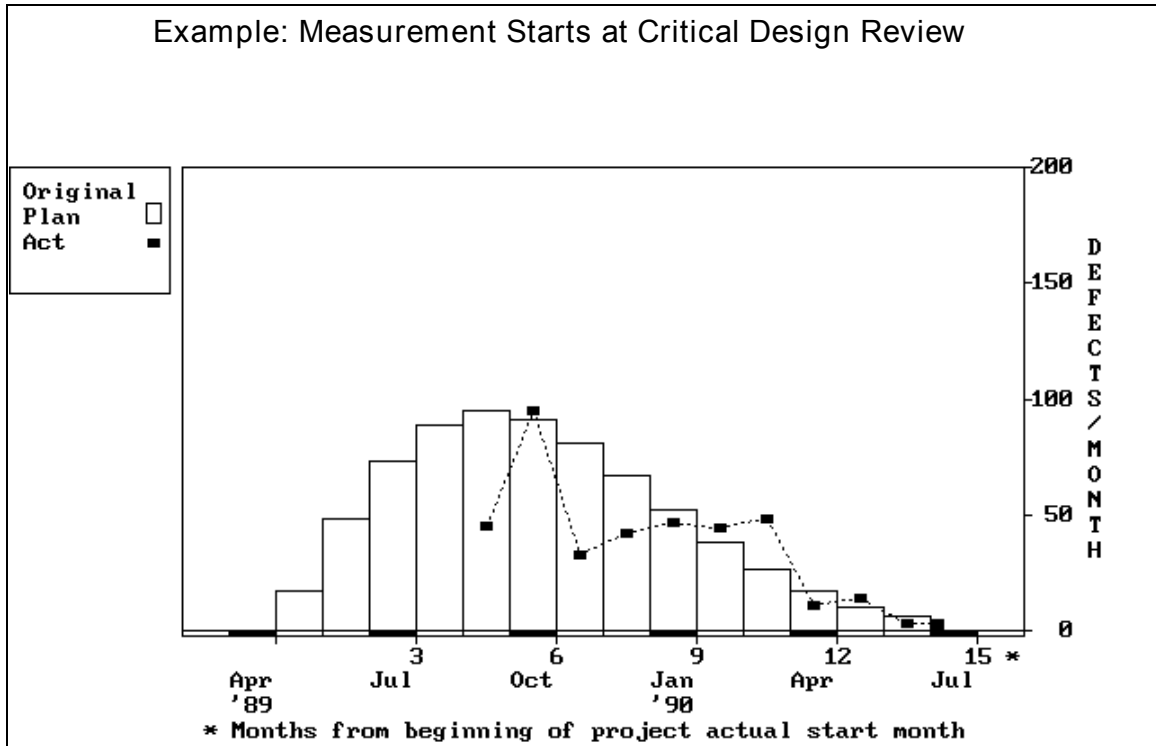


Figure 4. Example that compares the Rayleigh defect model to a project that collected defect data from Critical Design Review onward.

Figure 5 shows a project where the collection of defect data did not occur until the start of system test. Initially the discovery rate is low but quickly approaches the profile and tracks the Rayleigh model reasonably well for the duration of the project.

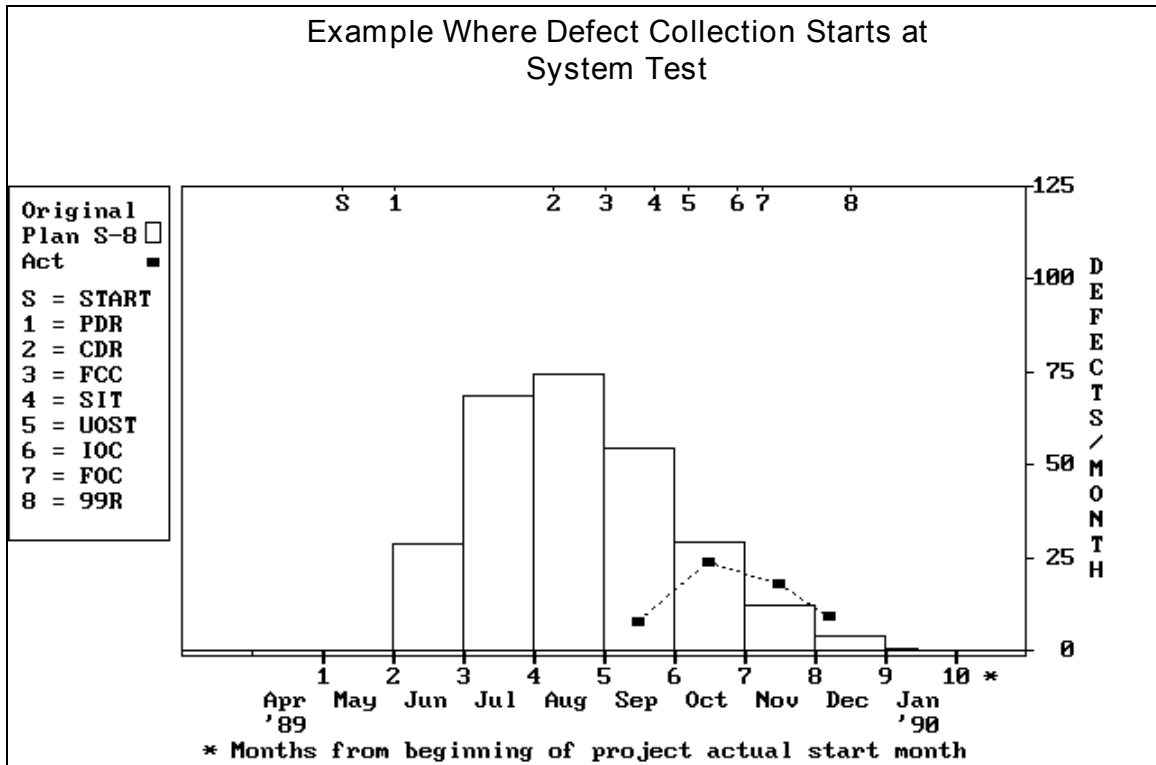


Figure 5. Example that compares the Rayleigh defect model to a project that collected defect data from the start of systems test onward.

In figures 4 and 5 we are looking at a defect measurement program that covers only part of the life cycle. One might ask what happened to the early defects? Most likely, they were discovered and fixed. When measurement is initiated it is possible to track the actuals from this point onward on a monthly basis. In partial life cycle measurement situations it is not unusual to see a short period of time where the actuals build to the Rayleigh model forecast. For example, if formal measurement starts at integration time some portion of the product will be in integration and subject to measurement. Some will still be in unit coding. The measurement data only represent a portion of the product. When measurement starts at systems test the ramp up of actuals should be more rapid. Factors that may slow the closure of the actuals to the forecast might be some delay on the ramp up of testing manpower, or test case availability.

## Rayleigh Defect Model Inputs

There are specific inputs that determine the duration and magnitude of the Rayleigh defect model. The inputs enable the model to provide an accurate forecast for a given situation. There are three macro parameters that the QSM model uses. They are:

- ❑ Source lines of code (new and modified) - A measure of the size required to build the specified functionality in the system.
- ❑ Productivity Index - A measure of process efficiency and product complexity.
- ❑ Peak Staffing - A measure of the human effort required to construct and test the system.

## Model Behavior Patterns

**Size:** Historic data have shown that as the code size increases the number of defects increases. The rate of defect increase is close to linear.

**Productivity Index:** The historic data have shown that the number of defects decreases as the Productivity Index increases. The decrease is exponential.

**Peak Staffing:** The historic data have shown that adding people increases the defect creation process at a rapidly accelerating rate. For example, a project of 350,000 SLOC , operating at a PI of 12 , using 40 people at peak staff would create approximately 2125 total defects. If 60 people were used, 3 months of schedule compression would be gained but 3010 defects would be created.

## Model Normalization

If we assume that the Rayleigh model is a reasonable representation of defect discovery over time then one needs to determine the appropriate area under the curve. To do that and have it reflect reality one needs data. QSM's initial data collection efforts showed that the only consistent time period where defect data was collected across organizations was from the start of systems test until the initial delivery of the system. The area under the Rayleigh curve during this time period is 17%. The model was normalized on data collected in this time increment. Knowing the area under the curve for this time increment it is possible to determine the total area under the curve.



Historically, the reliability of delivered software products at initial delivery has averaged 95% defect free. This level of reliability appears to be a "Minimum Acceptable Quality" level. Anything lower than this is generally not accepted by the customer. It is simply to "buggy".

## Model Calibration

As more organizations have become concerned with software reliability they have collected data. QSM has been actively reviewing this data and comparing it to what our model would predict. In some of these "play back's" we have found that magnitude of the curve was different. This causes one to speculate "why?" The historic data used in the model normalization process comes from a variety of organizations and environments. The testing characteristics in these environments is variable. Some of the projects experienced a normal testing profile of 8 hour cycles on a single platform. Other projects experienced more rigorous testing profiles. The variability of the historic data introduces some uncertainty in the normalization process.

To cope with this uncertainty we have allowed the model to be scaled up or down as is appropriate for a situation when the data analysis shows there is reason to do so.

A large data services company will serve as an example of when it is appropriate to scale the Rayleigh model. The company in this example has been performing software measurement since 1981. By 1985 they had a good body of cost, schedule and reliability statistics. As part of their software improvement process they implemented a formal inspection method developed by IBM corporation. This process was rigorous and included defects categories that had not been counted previously. As they measured their products under development they consistently found that the Rayleigh model under predicted the number of defects they were actually finding. The shape was still appropriate but the magnitude was consistently low. It took a multiplier close to 2 to bring the curve into good correspondence with their experience. They have continued to use this successfully in their environment. For the past four years they has been increasing their Productivity Index by .75 a year. Today they are a Software Engineering Institute (SEI) maturity level 3 organization.

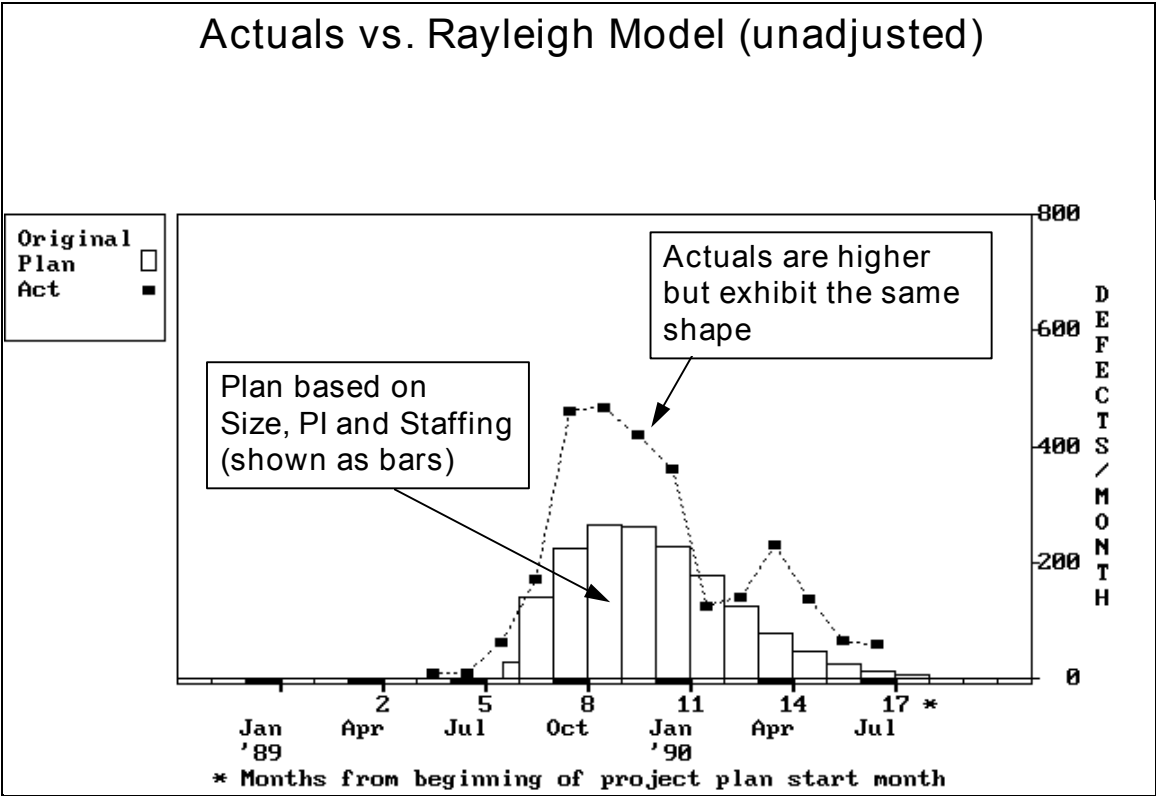


Figure 6. Rayleigh model vs. actual data for a large data services company.

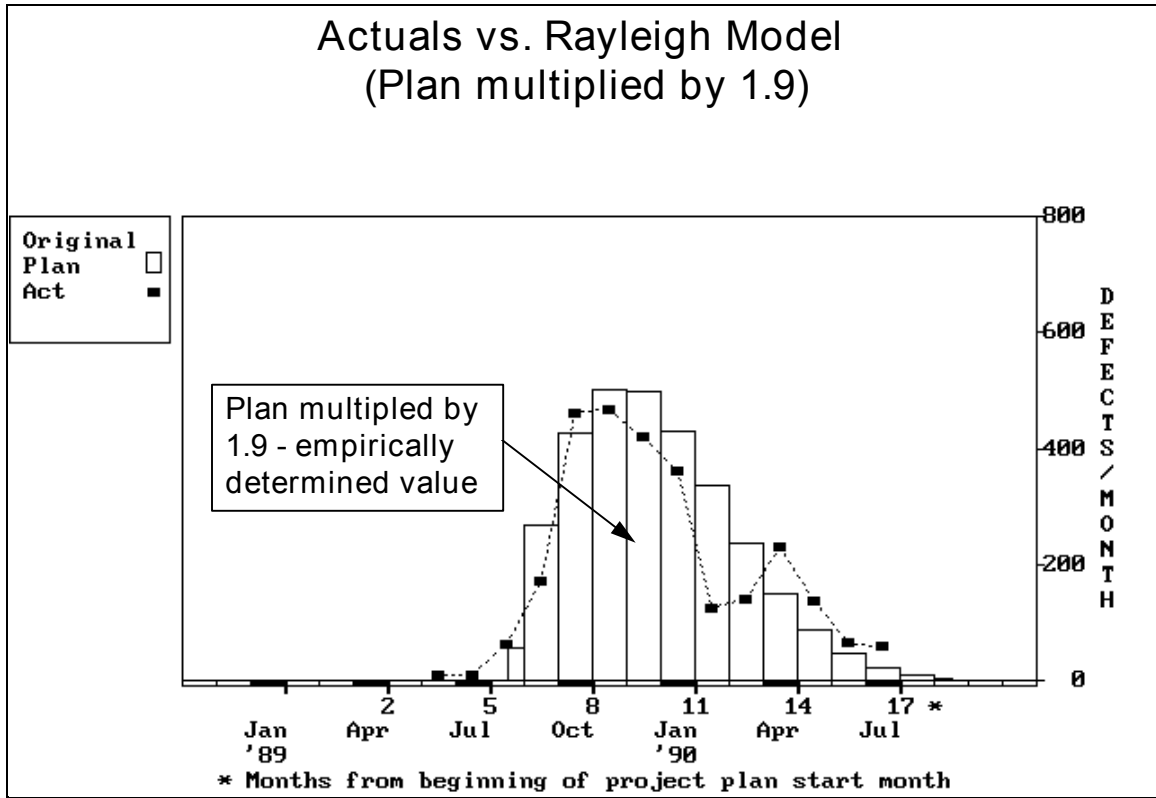


Figure 7. Rayleigh model multiplied by 1.9 to make actuals match experience.